Differential-Linear Attacks against the Stream Cipher Phelix*

Hongjun Wu and Bart Preneel

Katholieke Universiteit Leuven, ESAT/SCD-COSIC
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
{wu.hongjun,bart.preneel}@esat.kuleuven.be

Abstract. The previous key recovery attacks against Helix obtain the key with about 2^{88} operations using chosen nonces (reusing nonce) and about 1000 adaptively chosen plaintext words (or $2^{35.6}$ chosen plaintext words). The stream cipher Phelix is the strengthened version of Helix. In this paper we apply the differential-linear cryptanalysis to recover the key of Phelix. With 2^{34} chosen nonces and 2^{37} chosen plaintext words, the key of Phelix can be recovered with about $2^{41.5}$ operations.

1 Introduction

Phelix [5] is a fast stream cipher with an embedded authentication mechanism. It is one of the focus ciphers (both software and hardware) of the ECRYPT eSTREAM project. Phelix is a strengthened version of the stream cipher Helix [1].

Muller has applied differential attack to Helix [2]. He showed that the key of Helix can be recovered faster than by brute force if the attacker can force the initialization vectors to be used more than once. The attack requires about 2^{12} adaptively chosen plaintext words and 2^{88} operations. Paul and Preneel reduced the number of adaptively chosen plaintext words by a factor of at least 3 [4]. Later Paul and Preneel showed that $2^{35.6}$ chosen plaintext words can be used instead of adaptively chosen plaintexts [3]. All these key recovery attacks against Helix require about 2^{88} operations.

Phelix was designed and submitted to the ECRYPT eSTREAM project in 2005. The output function of Helix has been changed so that a larger plaintext diffusion can be achieved in Phelix. The Phelix designers claimed that Phelix is able to resist a differential key recovery attack even if the nonce is reused: "We claim, however, that even in such a case (referring to nonce reuse) it remains infeasible to recover the key" [5].

In this paper, we apply differential-linear cryptanalysis to Phelix assuming nonce reuse (this corresponds to a chosen nonce attack). We show that the key of Phelix can be recovered with a low complexity: 2^{37} chosen plaintext words and $2^{41.5}$ operations. Although the Phelix designers did expect that Phelix would

* This work was supported in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government and in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

loose most of its security properties when the nonce is reused, this paper shows that Phelix is completely insecure in such a setting.

This paper is organized as follows. In Sect. 2, we illustrate the operations of Phelix. Section 3 analyzes how the addend bits affect the differential distribution. Section 4 describes a basic differential key recovery attack on Phelix. The improved attack is given in Sect. 5. We discuss how to strengthen Phelix in Sect. 6. Section 7 concludes this paper.

2 The Stream Cipher Phelix

In this section, we only consider the encryption algorithm of Phelix. The full description of Phelix is given in [5]. The key size and nonce size of Phelix are 256 bits and 128 bits, respectively. The designers claim that there is no attack against Phelix with less than 2^{128} operations.

Phelix updates fives 32-bit words: Z_0, Z_1, Z_2, Z_3 and Z_4. At the ith step, two secret 32-bit words $X_{i,0}$, $X_{i,1}$ and one 32-bit plaintext word P_i are applied to update the internal states. One 32-bit keystream word S_i is generated and is used to encrypt the plaintext P_i. Note that the plaintext is used to update the internal state so that the authentication can be performed. The word $X_{i,0}$ is related to the key, and the word $X_{i,1}$ is related to the key and nonce in a very simple way. Recovering any $X_{i,0}$ and $X_{i,1}$ implies recovering part of the key. One step of Phelix is given in Fig. 1.

3 The Differential Propagation of Addition

In this section, we study how the addend bits affect the differential propagation. The importance of this study is that it shows that the values of the addend bits can be determined by observing the differential distribution of the sum.

Theorem 1. Denote ϕ_i as the ith least significant bit of ϕ. Suppose two positive m-bit integers ϕ and ϕ' differ only at the nth least significant bit position ($\phi \oplus \phi' = 2^n$). Let β be an m-bit random integer (m is much larger than n). Let $\psi = \phi + \beta$ and $\psi' = \phi' + \beta$. For $\beta_n = 0$, denote the probability that $\psi_{n+i} = \psi'_{n+i}$ as $p_{n+i,0}$. For $\beta_n = 1$, denote the probability that $\psi_{n+i} = \psi'_{n+i}$ as $p_{n+i,1}$. Then the difference $\Delta p_{n+i} = p_{n+i,0} - p_{n+i,1} = 2^{-n-i+1}$ ($i > 0$).

Theorem 1 can be proved easily if we consider the bias in the carry bits. We omit the proof here. In Theorem 1, the bias of the differential distribution decreases quickly as the value of n increases. We need another differential property that produces difference with a large bias even for large n. Before introducing that property, we give the following lemma from [6].

Lemma 1. Denote u and v as two random and independent n-bit integers. Let $c_n = (u + v) \gg n$, where c_n denotes the carry bit at the nth least significant

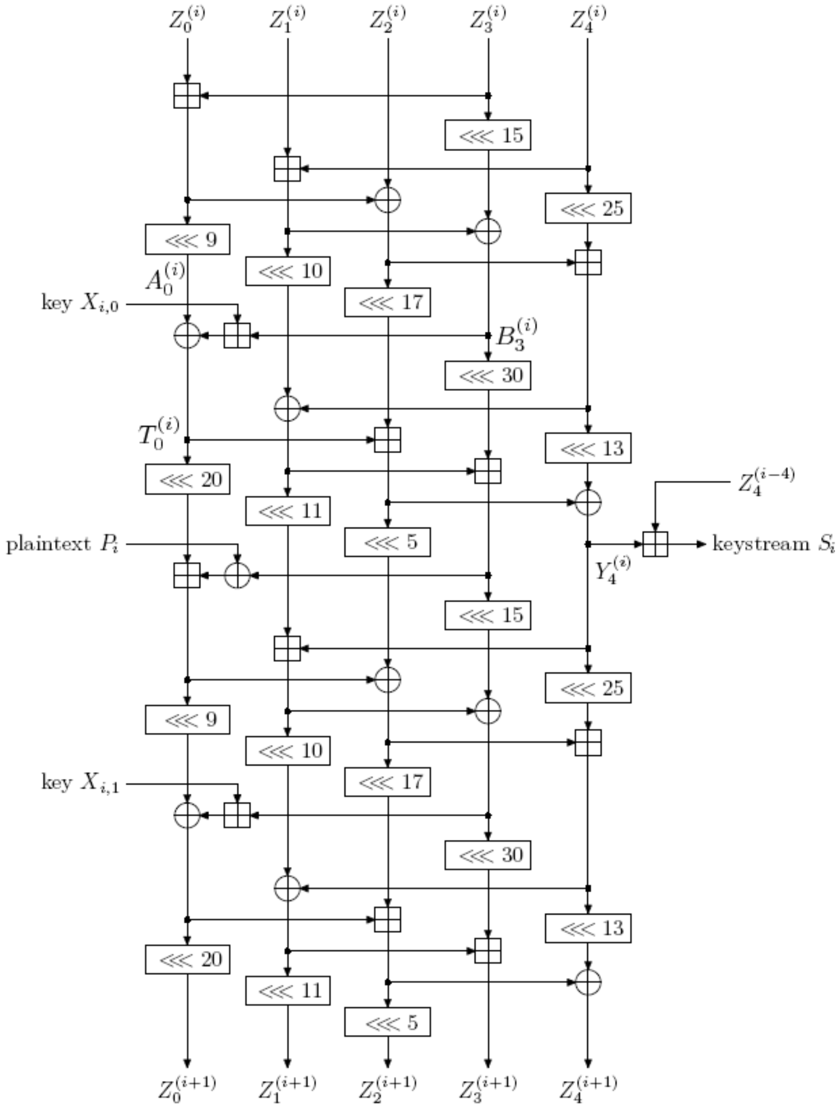

Fig. 1. One block of Phelix [5]

bit position. Denote the most significant bit of u as u_{n-1}. Then $\Pr(c_n \oplus u_{n-1} = 0) = \frac{3}{4}$.

The large bias of the differential distribution for large n is given below.

Theorem 2. Denote ϕ_i as the ith least significant bit of ϕ. Suppose two positive m-bit integers ϕ and ϕ' differ only at the nth least significant bit position ($\phi \oplus \phi' = 2^n$). Let β be an m-bit random integer (m is much larger than n). Let $\psi = \phi + \beta$ and $\psi' = \phi' + \beta$. For $\beta_n \oplus \beta_{n-1} = 0$, denote the probability that $\psi_{n+i} = \psi'_{n+i}$ as $\bar{p}_{n+i,0}$. For $\beta_n \oplus \beta_{n-1} = 1$, denote the probability that $\psi_{n+i} = \psi'_{n+i}$ as $\bar{p}_{n+i,1}$. Then the difference $\Delta\bar{p}_{n+i} = \bar{p}_{n+i,0} - \bar{p}_{n+i,1} = 2^{-i}$ ($i > 0$).

Proof. Denote the carry bit at the ith least significant bit position in $\psi = \phi + \beta$ as c_i, and that in $\psi' = \phi' + \beta$ as c'_i. Note that $c'_n = c_n$, thus $c'_n \oplus \beta_n = c_n \oplus \beta_n$. When $c'_n \oplus \beta_n = c_n \oplus \beta_n = 0$, we know that $\psi \oplus \psi' = 2^n$ with probability 1, i.e., $\psi_{n+i} = \psi'_{n+i}$ with probability 1 for $i > 0$. When $c'_n \oplus \beta_n = c_n \oplus \beta_n = 1$, by induction we obtain that $\psi_{n+i} = \psi'_{n+i}$ with probability $1 - 2^{-i+1}$ for $i > 0$. According to Lemma 1, we know that $c_n \oplus \beta_{n-1} = 0$ with probability $\frac{3}{4}$. If $\beta_n \oplus \beta_{n-1} = 0$, then $c_n \oplus \beta_n = 0$ with probability $\frac{3}{4}$, thus $\bar{p}_{n+i,0} = \frac{3}{4} \times 1 + \frac{1}{4} \times (1 - 2^{-i+1}) = 1 - \frac{1}{4} \times 2^{-i+1}$. If $\beta_n \oplus \beta_{n-1} = 1$, then $c_n \oplus \beta_n = 0$ with probability $\frac{1}{4}$, thus $\bar{p}_{n+i,1} = \frac{1}{4} \times 1 + \frac{3}{4} \times (1 - 2^{-i+1}) = 1 - \frac{3}{4} \times 2^{-i+1}$. Then the difference $\Delta\bar{p}_{n+i} = \bar{p}_{n+i,0} - \bar{p}_{n+i,1} = 2^{-i}$ for $i > 0$.

The above two theorems provide the guidelines to recover the key of Phelix. However, these two theorems deal with the ideal cases in which there is only one bit difference between ϕ and ϕ', and β is assumed to be random. In the attacks, we deal with the complicated situation where each bit of $\phi \oplus \phi'$ is biased, and β is a fixed integer. The value of each bit of β will affect the distribution of the higher order bits of $(\phi + \beta) \oplus (\phi' + \beta)$ in a complicated way. In order to simplify the analysis, we will use simulations to obtain these relations in the attacks.

4 A Basic Key Recovery Attack on Phelix

We will first investigate the differential propagation in Phelix. Then we show how to recover the key of Phelix by observing the differential distribution of the keystream.

4.1 The bias in the differential distribution of the keystream

Assume an attacker can choose an arbitrary value for the nonce, then a nonce can be used more than once. We introduce one-bit difference into the plaintext at the ith step, i.e., $P_i \neq P'_i$, and $P_i \oplus P'_i = 2^n$ ($0 \leq n \leq 31$). Then we analyze the difference between $B_3^{(i+1)}$ and $B_3'^{(i+1)}$ (as indicated in Fig. 1). If all the carry bits are 0 (replacing all the additions with XORs), then the differences only appear at the 9th, 11th, 13th, 15th and 17th least significant bits between $B_3^{(i+1)}$ and $B_3'^{(i+1)}$. Because of the carry bits, the differential distribution becomes complicated. We run the simulation and use the randomly generated $Y_k^{(i)}$ ($4 \geq k \geq 0$), P_i, $X_{i,1}$ in the simulation. With 2^{30} plaintext pairs, we obtain the distribution of $B_3^{(i+1)} \oplus B_3'^{(i+1)}$ in Table 1.

Table 1. The probability that $B_3^{(i+1),j} \oplus B_3'^{(i+1),j} = 0$ for $P_i \oplus P_i' = 1$

j	p	j	p	j	p	j	p
0	0.9997	8	1.0000	16	0.5001	24	0.9161
1	0.9998	9	0.0000	17	0.4348	25	0.9470
2	0.9999	10	0.5000	18	0.5000	26	0.9673
3	0.9999	11	0.4375	19	0.5486	27	0.9803
4	1.0000	12	0.5000	20	0.6366	28	0.9883
5	1.0000	13	0.4492	21	0.7283	29	0.9931
6	1.0000	14	0.5000	22	0.8083	30	0.9960
7	1.0000	15	0.4273	23	0.8708	31	0.9977

From Table 1, we see that the distribution of $B_3^{(i+1)} \oplus B_3'^{(i+1)}$ is heavily biased. For example, $B_3^{(i+1),8} = B_3'^{(i+1),8}$ with probability close to 1, while $B_3^{(i+1),9} = B_3'^{(i+1),9}$ with probability close to 0. Note that $T_0^{(i+1)} = A_0^{(i+1)} \oplus (B_3^{(i+1)} + X_{i+1,0})$, according to Theorem 2, the distribution of $T_0^{(i+1)} \oplus T_0'^{(i+1)}$ will be affected by the value of $X_{i+1,0}^8 \oplus X_{i+1,0}^9$, thus the distribution of $B_{i+1} \oplus B_{i+1}'$ will be affected by the value of $X_{i,0}^8 \oplus X_{i,0}^9$. By observing the distribution of $S_{i+1} \oplus S_{i+1}'$, it may be possible to determine the value of $X_{i,0}^8 \oplus X_{i,0}^9$. Shifting the one-bit difference between P_i and P_i', we may determine other values of $X_{i,0}^{j+1} \oplus X_{i,0}^j$ for $0 \le j \le 30$, and recover in this way the key $X_{i,0}$. After recovering eight consecutive $X_{i,0}$ values, the 256-bit key can be found immediately.

The above analysis gives a brief idea of the attack. However, the actual attacks are quite complicated due to the interference of many differences. It is very tedious to derive exactly how the distribution of $S_{i+1} \oplus S_{i+1}'$ is affected by the value of $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^j$. On the other hand, it is easy to search for the relation with simulations. In the following, we carried out a simulation to find the relation between the value of $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^j$ and the distribution of $S_{i+1} \oplus S_{i+1}'$.

Let two plaintexts differ only in the ith word, and $P_i \oplus P_i' = 1$. We use the randomly generated $Y_k^{(i)}$ $(4 \ge k \ge 0)$, P_i, $X_{i,1}$, $X_{i+1,0}$, $Z_4^{(i-3)}$ in the simulation. Denote with $p_{j,0}^n$ the probability that $S_{i+1}^n \oplus S_{i+1}'^n = 0$ when $X_{i,0}^{j+1} \oplus X_{i,0}^j = 0$. And denote $p_{j,1}^n$ as the probability that $S_{i+1}^n \oplus S_{i+1}'^n = 0$ when $X_{i,0}^{j+1} \oplus X_{i,0}^j = 1$. Let $\Delta\tilde{p}_j^n = (p_{j,0}^n - p_{j,1}^n) \times \frac{N}{\sigma}$, where N denotes the number of plaintext pairs, and $\sigma = \frac{\sqrt{N}}{2}$. Assume that the values of $p_{j,0}^n$ and $p_{j,1}^n$ are close to $\frac{1}{2}$. If $\Delta\tilde{p}_j^n > 4$, the difference between $p_{j,0}^n$ and $p_{j,1}^n$ is larger than 4σ, hence the value of $X_{i,0}^{j+1} \oplus X_{i,0}^j$ can be determined correctly with high probability. For every value of the two bits $X_{i,0}^{j+1}$ and $X_{i,0}^j$, we use 2^{28} pairs to generate $S_{i+1} \oplus S_{i+1}'$, then compute $p_{j,0}^n$ and $p_{j,1}^n$. Thus $N = 2^{29}$, and $\sigma = 2^{13.5}$. We list the large values of $\Delta\tilde{p}_j^n$ below:

For $j = 9$, $\Delta\tilde{p}_9^{13} = 55.7$.
For $j = 10$, $\Delta\tilde{p}_{10}^{13} = 133.9$.

For $j = 14$, $\Delta\tilde{p}_{14}^{17} = 51.5$.
For $j = 15$, $\Delta\tilde{p}_{15}^{19} = -9.1$, $\Delta\tilde{p}_{15}^{22} = 14.9$, $\Delta\tilde{p}_{15}^{23} = -15.7$.
For $j = 16$, $\Delta\tilde{p}_{16}^{19} = -50.8$, $\Delta\tilde{p}_{16}^{21} = 62.0$, $\Delta\tilde{p}_{16}^{22} = 97.7$, $\Delta\tilde{p}_{16}^{23} = -106.6$, $\Delta\tilde{p}_{16}^{25} = 11.8$, $\Delta\tilde{p}_{16}^{26} = 16.0$, $\Delta\tilde{p}_{16}^{27} = -17.4$.
For $j = 17$, $\Delta\tilde{p}_{17}^{21} = 77.4$, $\Delta\tilde{p}_{17}^{22} = 145.3$, $\Delta\tilde{p}_{17}^{23} = -171.6$, $\Delta\tilde{p}_{17}^{25} = 12.3$, $\Delta\tilde{p}_{17}^{26} = 28.5$, $\Delta\tilde{p}_{17}^{27} = -30.4$.
For $j = 18$, $\Delta\tilde{p}_{18}^{21} = 80.2$, $\Delta\tilde{p}_{18}^{22} = 179.7$, $\Delta\tilde{p}_{18}^{23} = -241.7$, $\Delta\tilde{p}_{18}^{26} = 32.8$, $\Delta\tilde{p}_{18}^{27} = -43.7$.
For $j = 19$, $\Delta\tilde{p}_{19}^{22} = 139.6$, $\Delta\tilde{p}_{19}^{23} = -220.6$, $\Delta\tilde{p}_{19}^{26} = 19.0$, $\Delta\tilde{p}_{19}^{27} = -46.5$.
For $j = 20$, $\Delta\tilde{p}_{20}^{23} = -156.7$, $\Delta\tilde{p}_{20}^{25} = -5.7$, $\Delta\tilde{p}_{20}^{26} = 18.3$, $\Delta\tilde{p}_{20}^{27} = -30.6$.
For $j = 21$, $\Delta\tilde{p}_{21}^{25} = -6.8$, $\Delta\tilde{p}_{21}^{26} = 9.5$, $\Delta\tilde{p}_{20}^{27} = -28.5$.

The data given above show that the distribution of $S_{i+1} \oplus S'_{i+1}$ is strongly affected by the value of $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^{j}$.

4.2 Recovering the key

Note that in the above analysis, when we deal with a particular $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^{j}$, the other bits of $X_{i+1,0}$ are random. In the key recovery attack, the value of $X_{i+1,0}$ is fixed, so we need to consider the interference between the bits $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^{j}$.

We notice that there are many large biases related to $S_{i+1}^{23} \oplus S_{i+1}'^{23}$. However, the values of $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^{j}$ $(15 \leq j \leq 20)$ all have a significant effect on the distribution of $S_{i+1}^{23} \oplus S_{i+1}'^{23}$. It is thus a bit complicated to determine the values of $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^{j}$ $(15 \leq j \leq 20)$.

In the following, we consider the bit $S_{i+1}^{17} \oplus S_{i+1}'^{17}$. Its distribution is dominated by the value of $X_{i+1,0}^{15} \oplus X_{i+1,0}^{14}$. For every value of the two bits $X_{i+1,0}^{15}$ and $X_{i+1,0}^{14}$, we use 2^{30} pairs to generate $S_{i+1} \oplus S'_{i+1}$, then compute $p_{14,0}^{17}$ and $p_{14,1}^{17}$. From the simulation, we found that $p_{14,0}^{17} = 0.50227$ and $p_{14,1}^{17} = 0.50117$. We denote the average of $p_{14,0}^{17}$ and $p_{14,1}^{17}$ as $\bar{p}_{14}^{17}$, i.e., $\bar{p}_{14}^{17} = \frac{p_{14,0}^{17} + p_{14,1}^{17}}{2} = 0.50172$. Running a similar simulation, we found that $p_{13,0}^{17} = 0.50175$ and $p_{13,1}^{17} = 0.50169$. For all the $j \neq 13, j \neq 14$, we found that $p_{j,0}^{17} \approx \bar{p}_{14}^{17}$ and $p_{j,1}^{17} \approx \bar{p}_{14}^{17}$. The value of $X_{i+1,0}^{15} \oplus X_{i+1,0}^{14}$ is recovered as follows: from the keystreams, we compute the fraction for which $S_{i+1}^{17} \oplus S_{i+1}'^{17} = 0$. If it is larger than $\bar{p}_{14}^{17}$, then the value of $X_{i+1,0}^{15} \oplus X_{i+1,0}^{14}$ is considered to be 0; otherwise the value of $X_{i+1,0}^{15} \oplus X_{i+1,0}^{14}$ is considered to be 1.

We now compute the number of plaintext pairs required to determine the value of $X_{i+1,0}^{15} \oplus X_{i+1,0}^{14}$. Suppose that N pairs of plaintexts are used. The standard deviation is $\sigma = \sqrt{N \times \bar{p}_{14}^{17} \times (1 - \bar{p}_{14}^{17})}$. To determine the value of $X_{i+1,0}^{15} \oplus X_{i+1,0}^{14}$ with success rate 0.99, we require that $N \times ((p_{14,0}^{17} - p_{14,1}^{17}) - (p_{13,0}^{17} - p_{13,1}^{17})) > 4.66 \times \sigma$ (The cumulative distribution function of the normal distribution gives value 0.99 at the point 2.33σ). Thus we require that $N > 2^{22.27}$.

We used the Phelix C source code submitted to eSTREAM in the experiments. [1]

Experiment 1. The goal of this experiment is to recover the value of $X_{1,0}^{15} \oplus X_{1,0}^{14}$. Each plaintext has two words P_0 and P_1. For each plaintext pair, the two words differ only in the least significant bit of P_0. N plaintext pairs are used for each key to determine the value of $X_{1,0}^{15} \oplus X_{1,0}^{14}$ as follows: if the fraction of cases for which $S_1^{17} \oplus S_1'^{17} = 0$ is larger than $\bar{p}_{14}^{17} = 0.50172$, then the value of $X_{1,0}^{15} \oplus X_{1,0}^{14}$ is considered to be 0; otherwise the value of $X_{1,0}^{15} \oplus X_{1,0}^{14}$ is considered to be 1. A random nonce was used for each plaintext pair. We tested 200 keys in the experiment. For $N = 2^{22.3}$, the values of $X_{1,0}^{15} \oplus X_{1,0}^{14}$ of 183 keys are determined correctly. For $N = 2^{25}$, the values of $X_{1,0}^{15} \oplus X_{1,0}^{14}$ of 192 keys are determined correctly.

Experiment 1 shows that the value of $X_{1,0}^{15} \oplus X_{1,0}^{14}$ can be determined successfully by introducing a difference in the least significant bit of P_0, but with a higher error rate. The reason is that other bits of $X_{1,0}$ affects the determination of $X_{1,0}^{15} \oplus X_{1,0}^{14}$ in a subtle way.

We now proceed to recover the other bits of $X_{1,0}$. By rotating the one-bit difference between P_0 and P_0', and using the same threshold value, we can determine the value of $X_{1,0}^{j+1} \oplus X_{1,0}^{j}$ for $2 \leq j \leq 3$, $5 \leq j \leq 10$ and $14 \leq j \leq 28$.

Thus we are able to recover 23 bits of information on each $X_{i,0}$. Note that the 256-bit key is recovered from eight consecutive $X_{i,0}$. Thus we are able to recover $23 \times 8 = 184$ bits of the key with success rate about $\frac{192}{200} = 0.96$. The number of plaintext pairs required in the attack is about $2^{25} \times 32 \times 8 = 2^{33}$.

We need to improve the above attack in two approaches: recovering more key bits and improving the success rate. The direct approach is to adjust the threshold value for each key bit position. In the following, we illustrate a more advanced approach which recovers the values of $Z_4^{(i)}$ before recovering the key.

5 Improving the Attack on Phelix

In the above attack, we use a random nonce for each plaintext pair, i.e., every nonce is used twice with the same key. When the nonce is used many times with the same key, we can introduce the difference at P_i and recover the value of Z_4^{i-3} by observing the distribution of $S_{i+1} \oplus S'_{i+1}$. Then we proceed to recover $X_{i+1,0}$.

5.1 Recovering $Z_4^{(i)}$

We introduce the difference to the least significant bit of P_i ($P_i \oplus P_i' = 1$). A simulation is carried out to determine the distribution of $Y_4^{(i+1)} \oplus Y_4'^{(i+1)}$. We use the randomly generated $Y_k^{(i)}$ ($4 \geq k \geq 0$), P_i, $X_{i,1}$, $X_{i+1,0}$ in the simulation.

[1] Note that we have found and corrected a small bug in this source code. The output should be computed as $S_i = Y_4^{(i)} + Z_4^{(i-4)}$ in stead of $S_i = Y_4^{(i)} + Z_4^{(i-3)}$.

Denote $\dot{p}_n$ as the probability that $Y_4^{(i+1),n} \oplus Y_4'^{(i+1),n} = 0$. With 2^{30} pairs, we obtain the values of $\dot{p}_n$ in Table 2.

Table 2. The probability that $Y_4^{(i+1),j} \oplus Y_4'^{(i+1),j} = 0$ for $P_i \oplus P_i' = 1$

j	$\dot{p}_j - 0.5$	j	$\dot{p}_j - 0.5$	j	$\dot{p}_j - 0.5$	j	$\dot{p}_j - 0.5$
0	0.03326	8	0.00003	16	−0.00003	24	0.00046
1	0.12983	9	0.03517	17	0.00268	25	0.05926
2	0.20291	10	0.00002	18	−0.00001	26	0.15064
3	−0.27754	11	0.00001	19	−0.00266	27	−0.24028
4	−0.00005	12	0.00000	20	−0.00004	28	0.00001
5	0.05663	13	0.02293	21	0.02276	29	0.05770
6	−0.15327	14	−0.00001	22	0.07434	30	0.15508
7	−0.00001	15	−0.00001	23	−0.14414	31	−0.24907

From Table 2, we notice that $Y_4^{(i+1)} \oplus Y_4'^{(i+1)}$ is heavily biased. For example, $Y_4^{(i+1),2} = Y_4'^{(i+1),2}$ with probability about 0.70291, while $Y_4^{(i+1),3} = Y_4'^{(i+1),3}$ with probability about 0.22246. Note that $S_{i+1} = Y_4^{(i+1)} \oplus Z_4^{(i-3)}$, according to Theorem 2, the distribution of $S_{i+1} \oplus S'_{i+1}$ is affected by the value of $Z_4^{(i-3),3} \oplus Z_4^{(i-3),2}$. Next we carry out simulations to characterize this relation.

We use the randomly generated $Y_k^{(i)}$ $(4 \geq k \geq 0)$, P_i, $X_{i,1}$, $X_{i+1,0}$, $Z_4^{(i-3)}$ in the simulation. The one-bit difference is introduced to P_i, i.e., $P_i \oplus P_i' = 2^j$. Denote $\ddot{p}_{j,0}^n$ as the probability that $S_{i+1}^n \oplus S_{i+1}'^n = 0$ when $Z_4^{(i-3),j+1} \oplus Z_4^{(i-3),j} = 0$. And denote $\ddot{p}_{j,1}^n$ as the probability that $S_{i+1}^n \oplus S_{i+1}'^n = 0$ when $Z_4^{(i-3),j+1} \oplus Z_4^{(i-3),j} = 1$. For each value of $Z_4^{(i-3),3}$ and $Z_4^{(i-3),2}$, we use 2^{28} plaintext pairs. We find that $\ddot{p}_{2,0}^5 = 0.5461$ and $\ddot{p}_{2,1}^5 = 0.5193$. The large difference between $\ddot{p}_{2,0}^5$ and $\ddot{p}_{2,1}^5$ shows that the value of $Z_4^{(i-3),3} \oplus Z_4^{(i-3),2}$ can be determined with success rate 0.999 with about $2^{13.9}$ plaintext pairs (The cumulative distribution function of the normal distribution gives value 0.999 at the point 3.1σ).

The above approach is able to recover $Z_4^{(0)}$, but the success rate is not that high according to our experiment. In the following, we use a new approach to determine $Z_4^{(0)}$. To reduce the interference between the bits of $Z_4^{(i-3)}$, we recover the least significant bit of $Z_4^{(i-3)}$ first, then proceed to recover the more significant bits bit-by-bit.

We start with determining the value of $Z_4^{(i-3),0}$. Let $P_i \oplus P_i' = 1$. Running the simulation with2^{28} plaintext pairs, we found that $\ddot{p}_{-1,0}^2 = 0.70296$, and $\ddot{p}_{-1,1}^2 = 0.65422$ (let $Z_4^{(i-3),-1} = 0$). To determine the value of $Z_4^{(i-3),0}$ with success rate 0.999, we need about $2^{12.0}$ plaintext pairs.

Experiment 2. The goal of this experiment is to determine the value of $Z_4^{(0),0}$. Each plaintext has five random words P_i $(0 \leq i \leq 4)$. For each plaintext pair, the difference is only in the least significant bit of P_3. N plaintext pairs are used for each key/nonce pair to determine the value of $Z^{(0),0}$ as follows: if the rate that $S_4^2 \oplus S_4'^2 = 0$ is larger than $\frac{\ddot{p}_{-1,0}^2 + \ddot{p}_{-1,1}^2}{2} = \frac{0.70296+0.65422}{2} = 0.6786$, then the value of $Z_4^{(0),0}$ is considered to be 0; otherwise the value of $Z_4^{(0),0}$ is considered to be 1. We tested 1000 key/nonce pairs in the experiment. For $N = 2^{12}$, the values of $Z_4^{(0),0}$ of 998 key/nonce pairs are determined correctly. For $N = 2^{13}$, the values of $Z_4^{(0),0}$ of all the key/nonce pairs are determined correctly.

After recovering the value of $Z_4^{(i-3),0}$, we proceed to recover the values of the other bits of $Z_4^{(i-3)}$. Let $Z_4^{(i-3),(n-1\cdots0)}$ denote the n least significant bits of $Z^{(i-3)}$, i.e., $Z_4^{(i-3),(n-1\cdots0)} = Z^{(i-3)} \bmod 2^n$. Let the difference be introduced to the kth least significant bit of P_i, i.e., $P_i \oplus P_i' = 2^k$. Denote $\grave{p}^{k,j,0}_{Z_4^{(i-3),(n-1\cdots0)}}$ as the probability that the value of the jth bit of $(S_{i+1} - Z_4^{(i-3),(n-1\cdots0)}) \oplus (S_{i+1}' - Z_4^{(i-3),(n-1\cdots0)})$ is 0 when $Z_4^{(i-3),n} = 0$. Denote $\grave{p}^{k,j,1}_{Z_4^{(i-3),(n-1\cdots0)}}$ as the probability that the value of the jth bit of $(S_{i+1} - Z_4^{(i-3),(n-1\cdots0)}) \oplus (S_{i+1}' - Z_4^{(i-3),(n-1\cdots0)})$ is 0 when $Z_4^{(i-3),n} = 1$. If the value of $Z_4^{(i-3),(n-1\cdots0)}$ is determined correctly, then $\grave{p}^{k,j,0}_{Z_4^{(i-3),(n-1\cdots0)}} = \grave{p}_0^{k,j,0}$, and $\grave{p}^{k,j,1}_{Z_4^{(i-3),(n-1\cdots0)}} = \grave{p}_0^{k,j,1}$. This property is important for recovering $Z_4^{(i-3)}$.

Let $P_i \oplus P_i' = 2$. We use 2^{28} plaintext pairs in the simulation. We found that $\grave{p}_0^{1,3,0} = 0.66469$ and $\grave{p}_0^{1,3,1} = 0.60220$. It shows that when $Z_4^{(i-3),0} = 0$, if the rate that $S_{i+1}^3 \oplus S_{i+1}'^3 = 0$ is larger than $\frac{0.66469+0.60220}{2} = 0.63345$, then the value of $Z_4^{(i-3),1}$ is determined to be 0; otherwise the value of $Z_4^{(i-3),1}$ is determined to be 1. We need about $2^{11.3}$ plaintext pairs to determine the value of $Z_4^{(i-3),1}$ correctly with success rate 0.999. Using the Phelix code in the experiment, we tested 1000 random key/nonce pairs satisfying $Z_4^{(0),0} = 0$, and 2^{12} plaintext pairs are used for each key/nonce pair with the difference $P_3 \oplus P_3' = 2$. We found that all the 1000 values of $Z_4^{(0),1}$ are determined correctly. If $Z_4^{(i-3),0} = 1$, we observe the third least significant bit of $(S_{i+1} - 1) \oplus (S_{i+1}' - 1)$, and we can determine the value of $Z_4^{(0),1} = 0$ with success rate 0.999 with about $2^{11.3}$ plaintext pairs.

Let $P_i \oplus P_i' = 2^2$, we are able to determine the value of $Z_4^{(i-3),2}$ by observing the fourth least significant bit of $(S_{i+1} - Z_4^{(i-3),(1\cdots0)}) \oplus (S_{i+1}' - Z_4^{(i-3),(1\cdots0)})$. In general, let $P_i \oplus P_i' = 2^j$, then we are able to determine the value of $Z_4^{(i-3),j}$ by observing the $(j+2)$th least significant bit of $(S_{i+1} - Z_4^{(i-3),(j-1\cdots0)}) \oplus (S_{i+1}' - Z_4^{(i-3),(j-1\cdots0)})$ with about 2^{12} plaintext pairs. Thus we are able to recover $Z_4^{(i-3)}$ (except the values of $Z_4^{(i-3),30}$ and $Z_4^{(i-3),31}$) with success rate very close to 1. The number of plaintext pairs required in the above attack is about $2^{12} \times 30 \approx 2^{17}$.

5.2 Recovering $X_{i+1,0}$

After recovering $Z_4^{(i-3)}$ (except $Z_4^{(i-3),31}$ and $Z_4^{(i-3),30}$), we know the value of $(S_{i+1} - Z_4^{(i-3),(29\cdots0)}) \oplus (S'_{i+1} - Z_4^{(i-3),(29\cdots0)})$. Thus we know the value of $Y^{(i+1),j} \oplus Y'^{(i+1),j}$ $(0 \leq j \leq 30)$. Then we are able to recover $X_{i+1,0}$ more efficiently.

Let two plaintexts differ only in the ith word. And let $P_i \oplus P'_i = 1$. We use the randomly generated $Y_k^{(i)}$ $(4 \geq k \geq 0)$, P_i, $X_{i,1}$, $X_{i+1,0}$, $Z_4^{(i-3)}$ in the simulation. For every value of the two bits $X_{i,0}^{j+1}$ and $X_{i,0}^{j}$, we use 2^{28} plaintext pairs to generate $Y_4^{(i+1)} \oplus Y_4'^{(i+1)}$, then compute $p_{j,0}^n$ and $p_{j,1}^n$ (suppose that $S_{i+1} = Y_4^{(i+1)}$ since $Z_4^{(i-3)}$ is known). Thus $N = 2^{29}$, and $\sigma = 2^{13.5}$. We list the following two large biases $\Delta\tilde{p}_j^n$:

For $j = 9$, $\Delta\tilde{p}_9^{13} = 144.1$
For $j = 10$, $\Delta\tilde{p}_{10}^{13} = 362.12$

We use $\Delta\tilde{p}_9^{13}$ and $\Delta\tilde{p}_{10}^{13}$ in the attack. Note that the values of $X_{i+1,0}^{10} \oplus X_{i+1,0}^{9}$ and $X_{i+1,0}^{11} \oplus X_{i+1,0}^{10}$ both affect the distribution of $Y_4^{(i+1),13} \oplus Y_4'^{(i+1),13}$. We carried out a simulation with 2^{30} chosen plaintext pairs to determine how the value of $X_{i+1,0}^{9}$ affects the value of $Y_4^{(i+1),13}$. If $X_{i+1,0}^{9} = 0$, and the values of $X_{i+1,0}^{11}||X_{i+1,0}^{0}$ are 00, 11, 01 and 11 (in binary format), we obtain that $Y_4^{(i+1),13} \oplus Y_4'^{(i+1),13} = 0$ with probability 0.53033, 0.52334, 0.51946 and 0.51864, respectively; if $X_{i+1,0}^{9} = 1$, we obtain that $Y_4^{(i+1),13} \oplus Y_4'^{(i+1),13} = 0$ with probability 0.52334, 0.53030, 0.51861 and 0.51948, respectively. We thus let $p_{0,0}^{13} = 0.52334$, and $p_{0,1}^{13} = \frac{0.51946+0.51948}{2} = 0.51947$. About $2^{19.3}$ plaintext pairs are required to determine the value of $X_{i+1,0}^{11} \oplus X_{i+1,0}^{10}$ with success rate 0.999.

Experiment 3. Suppose that the value of $Z_4^{(0)}$ is known. The goal of this experiment is to determine the value of $X_{4,0}^{11} \oplus X_{4,0}^{10}$. Each plaintext has five random words P_i $(0 \leq i \leq 4)$. For each plaintext pair, those five words differ only in the least significant bit of P_3. N plaintext pairs are used for each key/nonce pair to determine the value of $X_{4,0}^{11} \oplus X_{4,0}^{10}$ as follows: if the rate that $Y_4^{13} \oplus Y_4'^{13} = 0$ is larger than $\frac{0.52334+0.51947}{2} = 0.52140$, then the value of $X_{4,0}^{11} \oplus X_{4,0}^{10}$ is considered to be 0; otherwise the value of $X_{4,0}^{11} \oplus X_{4,0}^{10}$ is considered to be 1. We tested 1000 key/nonce pairs in the experiment. For $N = 2^{19.3}$, 948 values of 1000 $X_{4,0}^{11} \oplus X_{4,0}^{10}$ are determined correctly. We change the threshold value 0.52140 to 0.52035, then 970 values of 1000 $X_{4,0}^{11} \oplus X_{4,0}^{10}$ are determined correctly for $N = 2^{20}$, 976 values are determined correctly for $N = 2^{21}$, 990 values are determined correctly for $N = 2^{22}$.

Experiment 3 shows that the value of $X_{4,0}^{11} \oplus X_{4,0}^{10}$ can be determined successfully by introducing a difference in the least significant bit of P_3. With 2^{22}

chosen pairs, we are able to determine the value of $X_{4,0}^{11} \oplus X_{4,0}^{10}$ with success rate about 0.99.

Then we shift the one-bit difference to recover the values of $X_{1,0}^{j+1} \oplus X_{1,0}^{j}$ for $2 \leq j \leq 28$. The threshold value needs to be modified for different values of j. The results are given in Table 3 in Appendix A. Note that according to Experiment 3, the threshold values should be slightly adjusted to achieve high success rate.

The reason that the values of $X_{4,0}^{j+1} \oplus X_{4,0}^{j}$ cannot be recovered for $j \geq 29$ is that the value of $X_{1,0}^{j+1} \oplus X_{1,0}^{j}$ cannot affect the distribution of $S_1^{j+3} \oplus S_1'^{j+3}$ since $S_1 \oplus S_1'$ is a 32-bit word. The reason that the number of plaintext required for $j = 9$ is relatively small is that the difference for $j = 13$ is introduced to the most significant bit of the word P_3, thus it causes less difference propagation, and results in a larger bias in the keystream.

Note that the most significant bit of $Y_4^{(i+1)} \oplus Y_4'^{(i+1)}$ is not known since $Z_4^{i-1,31}$ and $Z_4^{i-1,30}$ are not recovered. Thus to determine the value of $X_{1,0}^{29} \oplus X_{1,0}^{28}$, we need to consider the most significant bit of $(S_{i+1} - Z_4^{(i-3),(29\cdots0)}) \oplus (S'_{i+1} - Z_4^{(i-3),(29\cdots0)})$. The threshold value needs to be changed to 0.51128; and the number of plaintext pairs required is $2^{22.1}$.

After recovering the values of $X_{1,0}^{j+1} \oplus X_{1,0}^{j}$ for $2 \leq j \leq 28$, we proceed to determine the value of $X_{i+1,0}^{0}$, $X_{i+1,0}^{1}$ and $X_{i+1,0}^{2}$.

We start with recovering $X_{i+1,0}^{0}$. Let $P_i \oplus P_i' = 2^{21}$. Running the simulation with 2^{28} plaintext pairs, we found that $p_{21,0}^2 = 0.51596$, $p_{21,1}^2 = 0.50355$. Thus $2^{15.93}$ plaintext pairs are needed to determine the value of $X_{i+1,0}^{0}$ with success rate 0.999. Using the Phelix code in the experiment, we introduce the difference $P_3 \oplus P_3' = 2^{21}$, and set the threshold value as $\frac{0.51596+0.50355}{2} = 0.50975$. We tested 1000 key/nonce pairs in the experiment. With 2^{16} plaintext pairs, all the values of the 1000 $X_{4,0}^{0}$ are determined correctly.

After determine the value of $X_{i+1,0}^{0}$, we determine the value of $X_{i+1,0}^{1}$ as follows. The simulation shows that the value of $X_{i+1,0}^{1}$ can be determined only when $X_{i+1,0}^{0} = 0$. For $X_{i+1,0}^{0} = 0$, we set the difference as $P_i \oplus P_i' = 2^{22}$. With 2^{28} chosen plaintext pairs, we found that if $X_{i+1,0}^{1} = 0$, then $Y_4^{(i+1),3} = 0$ with rate 0.51528; otherwise $Y_4^{(i+1),3} = 0$ with rate 0.50459. With $2^{16.4}$ plaintext pairs, the value of $X_{i+1,0}^{1}$ can be determined with success rate 0.999. Using the Phelix code in the experiement, we introduce the difference $P_3 \oplus P_3' = 2^{22}$, and set the threshold value as $\frac{0.51528+0.50459}{2} = 0.50994$. We tested 1000 key/nonce pairs with $X_{4,0}^{0} = 0$ in the experiment. With $2^{16.4}$ plaintext pairs, all the values of the 1000 $X_{4,0}^{1}$ are determined correctly. It shows that the value of $X_{i+1,0}^{1}$ can be determined successfully if $X_{4,0}^{0} = 0$.

We continue to recover the value of $X_{i+1,0}^{2}$. We introduce a difference to the 15th least significant bit of P_i, and observe the distribution of $Y_4^{(i-3),4}$. We carry out a simulation with 2^{31} plaintext pairs with $P_3 \oplus P_3' = 2^{15}$. 2^{31} plaintext pairs are used for each value of $X_{i+1,0}^{1}X_{i+1,0}^{0}$. When $X_{i+1,0}^{0} = 0$, if $X_{i+1,0}^{1} = 0$, the fraction of values for which $Y_4^{(i-3),4} = 0$ if $X_{i+1,0}^{2} = 0$ and $X_{i+1,0}^{2} = 1$ are

0.53106 and 0.52613, respectively; if $X^1_{i+1,0} = 1$, the franction of values for which $Y_4^{(i-3),4} = 0$ if $X^2_{i+1,0} = 0$ and $X^2_{i+1,0} = 1$ are 0.52318 and 0.52315, respectively. It shows that the value $X^2_{i+1,0}$ can only be determined if the values of $X^1_{i+1,0}$ and $X^1_{i+1,0}$ are both zero, and $2^{18.6}$ plaintext pairs are required to achieve the success rate 0.999.

In the above attacks, we recovered 28.75 bits of $X_{i+1,0}$: $X^{j+1}_{i+1,0} \oplus X^j_{i+1,0}$ for $2 \leq j \leq 28$, $X^0_{i+1,0}$, $X^1_{i+1,0}$ (only if $X^0_{i+1,0} = 0$), and $X^1_{i+1,0}$ (only if $X^0_{i+1,0} = 0$ and $X^1_{i+1,0} = 0$). 27 bits of $X^{j+1}_{i+1,0} \oplus X^j_{i+1,0}$ $(2 \leq j \leq 28)$ can be determined according to Table 3. From Experiment 3, we know that if we slightly adjust the threshold value, and use about $2^{2.7}$ times the number of plaintext pairs compared to Table 3, the success rate is about 0.99. The number of plaintext pairs required to determine these 27 bits is thus about $27 \times 2^{22.2} \times 2^{2.7} = 2^{29.7}$. The number of plaintext pairs to determine $X^0_{i+1,0}$, $X^1_{i+1,0}$ and $X^2_{i+1,0}$ is small compared to $2^{29.7}$. The attack to recover 28.75 bits of $X_{i+1,0}$ requires thus about $2^{32.7}$ chosen plaintext pairs.

After recovering eight consecutive $X_{i+1,0}$, we recovered $28.75 \times 8 = 230$ key bits. To recover the 256-bit key, the amount of operations required is about $2^{256-230+\binom{27\times 8}{0.01\times 27\times 8}} = 2^{41.5}$.

The number of chosen plaintext pairs required in the attack is about $2^{29.7} \times 8 = 2^{32.7}$. The length of each plaintext ranges from 5 to 13 words. Thus the total amount of chosen plaintext required is about $2 \times 2^{32.7} \times \frac{5+13}{2} \approx 2^{37}$ words. (The number of plaintext pairs needed to recover 8 consecutive $Z^{(i)}_4$ is about $2^{17} \times 8 = 2^{25}$. It is small compared to $2^{32.7}$).

6 An Approach to Strengthen Helix and Phelix

In Helix and Phelix, the plaintext is used to affect the internal state of the cipher. In order to achieve a high encryption speed, each plaintext word affects the keystream without passing through sufficient confusion and diffusion layers. This is the intrinsic weakness in the structure of Helix and Phelix. In the following, we provide a method to reduce the effect of such weakness.

The security of the encryption of Helix and Phelix can be improved significantly if a secure one-way function is used to generate the initial state of the cipher from the key and nonce. Then even if the internal state of one particular nonce is recovered, the impact on the security of the encryption is very limited since the key of the cipher is not affected. We believe that such an approach can be applied to improve the security of all the ciphers that use the plaintext to affect the internal state.

However, we must point out that such an approach does not substantially improve the security of the MAC in Helix and Phelix. Once an internal state is recovered, the attacker can forge many messages related to that particular nonce.

7 Conclusion

Phelix is vulnerable to a key recovery attack when chosen nonces and chosen plaintexts are used. The computational complexity of the attack is much less than that of the attack against Helix. Our attack shows that Phelix fails to strengthen Helix in this respect.

We believe that one necessary requirement for a secure general-purpose stream cipher is that the key of the cipher should not be recoverable even if the attacker can control the generation of the nonce. In practice an attacker may gain access to a Phelix encryption device for a while, reuse a nonce and recover the key. We thus consider Phelix as insecure. (When the integrity checking mechanism is not enforced, an attacker can even modify the nonces in ciphertext and obtain repeated nonces.) Muller has pointed out the practical impact of the key recovery attack with reused nonces on the security of Helix in detail [2]. Muller stated clearly the difference between the nonce reusing attack against Helix and that against a synchronous stream cipher since the attack against Helix results in key recovery. The same comments apply to the attacks against Phelix.

Acknowledgements

The authors would like to thank the anonymous reviewers of FSE 2007 for their helpful comments.

References

1. N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, S. Lucks, and T. Kohno. "Helix, Fast Encryption and Authentication in a Single Cryptographic Primitive." In *Fast Software Encryption – FSE 2003*, LNCS 2887, T. Johansson (Ed.), pp. 330-346. Springer-Verlag, 2003.
2. F. Muller. "Differential Attacks against the Helix Stream Cipher." In *Fast Software Encryption – FSE 2004*, LNCS 3017, B.K. Roy and W. Meier (Eds.), pp. 94-108, Springer-Verlag, 2004.
3. S. Paul, and B. Preneel, "Solving Systems of Differential Equations of Addition." *Australasian Conference on Information Security and Privacy – ACISP 2005*, LNCS 3574, C. Boyd and J. Gonzalez (Eds.), pp. 75-88, Springer-Verlag, 2005.
4. S. Paul, and B. Preneel, "Near Optimal Algorithms for Solving Differential Equations of Addition with Batch Queries." In *Progress in Cryptology – Indocrypt 2005*, LNCS 3797, S. Maitra, C.E. Venimadhavan, R. Venkatesan (Eds.), pp. 90-103, Springer-Verlag, 2003.
5. D. Whiting, B. Schneier, S. Lucks, and F. Muller, "Phelix: Fast Encryption and Authentication in a Single Cryptographic Primitive." eSTREAM, ECRYPT Stream Cipher Project Report 2005/027.
6. H. Wu, and B. Preneel, "Cryptanalysis of the Stream Cipher ABC v2." In *Selected Areas in Cryptography – SAC 2006*, Lecture Notes in Computer Science, to appear.

A The complexity to recover $X_{i+1,0}$ with $Z_4^{(i-3)}$ known

The number of plaintext pairs and the threshold value required to recover the value of each $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^{j}$ $(2 \leq j \leq 28)$ are given in Table 3. Each value n in the second column indicates that the difference is introduced in the nth least significant bit of P_i. Each value n in the third column shows that the nth least significant bit of $Y_4^{(i+1)}$ is used in the attack.

Table 3. The number of plaintext pairs for recovering $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^{j}$

j	Difference position in P_i	Bit position in $Y_4^{(i+1)}$	Threshold value	Plaintext Pairs
2	24	5	0.51101	$2^{24.4}$
3	25	6	0.51110	$2^{23.1}$
4	26	7	0.51120	$2^{22.5}$
5	27	8	0.51125	$2^{22.3}$
6	28	9	0.51091	$2^{22.4}$
7	29	10	0.51116	$2^{23.6}$
8	30	11	0.51562	$2^{20.7}$
9	31	12	0.54353	$2^{18.4}$
10	0	13	0.52141	$2^{19.3}$
11	1	14	0.52099	$2^{19.3}$
12	2	15	0.51850	$2^{19.5}$
13	3	16	0.50998	$2^{21.3}$
14	4	17	0.51107	$2^{21.9}$
15	5	18	0.51128	$2^{22.2}$
16	6	19	0.51129	$2^{22.2}$
17	7	20	0.51131	$2^{22.2}$
18	8	21	0.51128	$2^{22.1}$
19	9	22	0.51117	$2^{21.7}$
20	10	23	0.51149	$2^{22.2}$
21	11	24	0.51172	$2^{22.0}$
22	12	25	0.51187	$2^{22.0}$
23	13	26	0.51191	$2^{22.0}$
24	14	27	0.51185	$2^{22.1}$
25	15	28	0.51129	$2^{22.2}$
26	16	29	0.51129	$2^{22.1}$
27	17	30	0.51131	$2^{22.2}$
28	18	31	0.51130	$2^{22.1}$