Differential-Linear Attacks against the Stream Cipher Phelix

Hongjun Wu and Bart Preneel

Katholieke Universiteit Leuven
ESAT/COSIC

Overview

1. Introduction to Helix and Phelix
2. Description of Phelix
3. Differential propagation of addition
4. A basic attack on Phelix
5. Improving the attack on Phelix
6. Improving the security of Phelix
7. Open problems
8. Conclusion

1 Background (1)

Stream Cipher Helix (FSE 2003)

- stream cipher + message authentication
- message is applied to update the internal state
- encryption: message is XORed with the keystream
- MAC: generated from internal state after finishing encryption

- gain – no separate MAC
- cost – error propagation + security concern

1 Background (2)

Attacks against Helix

Differential key recovery attack (Muller, 2004):

nonce reuse;

2^{12} adaptively chosen plaintext words, 2^{88} operations

Reducing the number of plaintext words (Paul-Preneel, 2005)

about 2^{10} adaptively chosen plaintext words;

or $2^{35.6}$ chosen plaintext words

1. Background (3)

Stream Cipher Phelix (2005)

Phelix: the strengthened version of Helix

1) message passing through more operations before affecting the keystream: half block in Helix, one full block in Phelix
2) more internal state words in generating a keystream word: one internal state word in Helix, two in Phelix

Is Phelix secure? Still vulnerable to the differential key recovery attack, effective key size being reduced to 41.5 bits

2. Stream Cipher Phelix (1)

Stream Cipher Phelix

stream cipher + message authentication code

256-bit key, 128-bit IV

eSTREAM Phase II software and hardware focus cipher

Fast in software: 6.6 cycles/byte on Pentium M processor

Hardware: twelve 32-bit additions are required for one 32-bit keystream word: efficient ?

2. Stream Cipher Phelix (2)

Stream Cipher Phelix

160-bit internal state: updated by message

512-bit internal state: simply related to the key and IV
incremented during the encryption

Phelix: one block

Z_0, Z_1, Z_2, Z_3, Z_4 :

160-bit internal state

updated by message

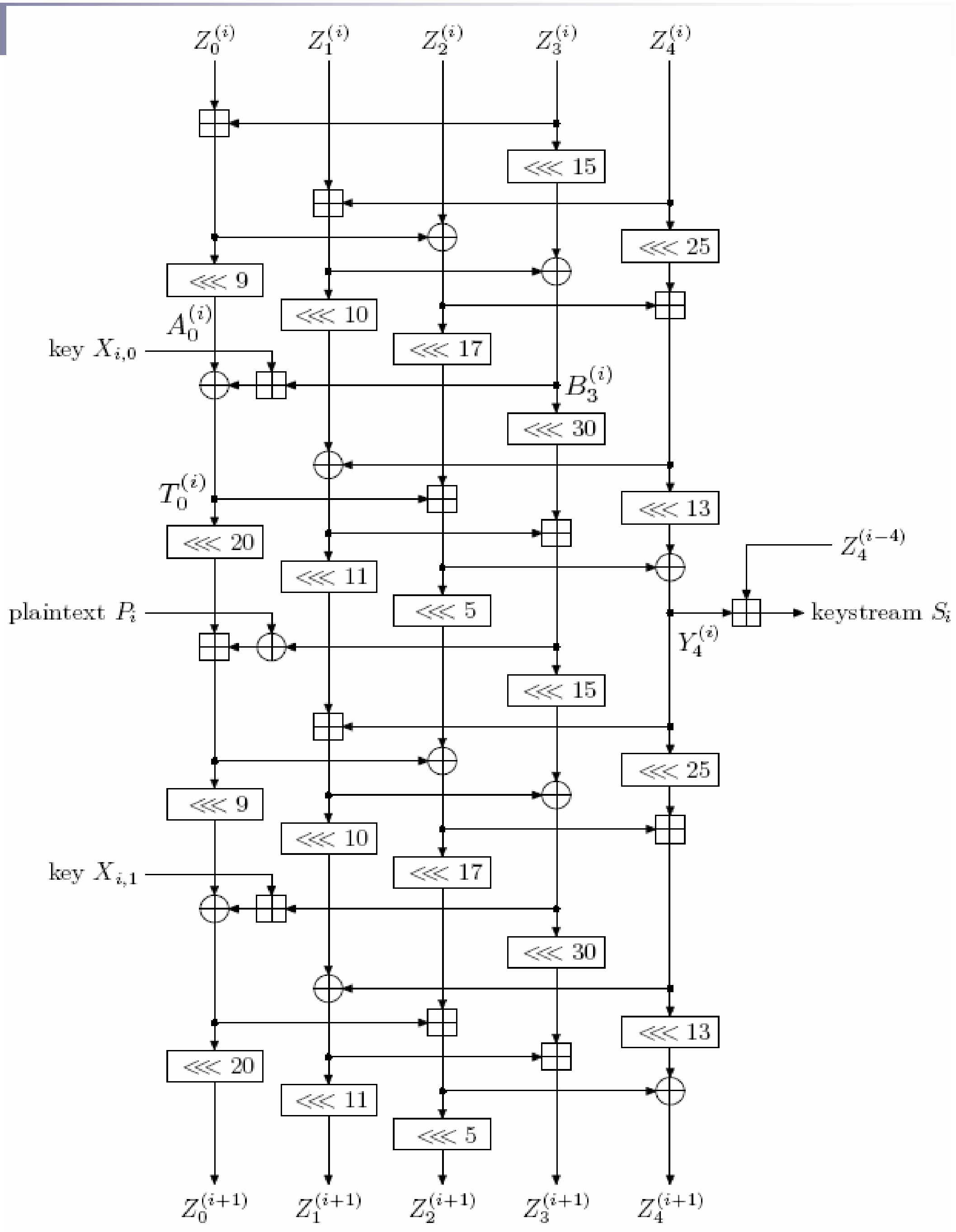
$X_{i,0},\ \ X_{i,1}$:

512-bit internal state,

determined by key, IV;

Encryption:

$$C_i = P_i \oplus S_i$$



3. Differential Propagation of Addition

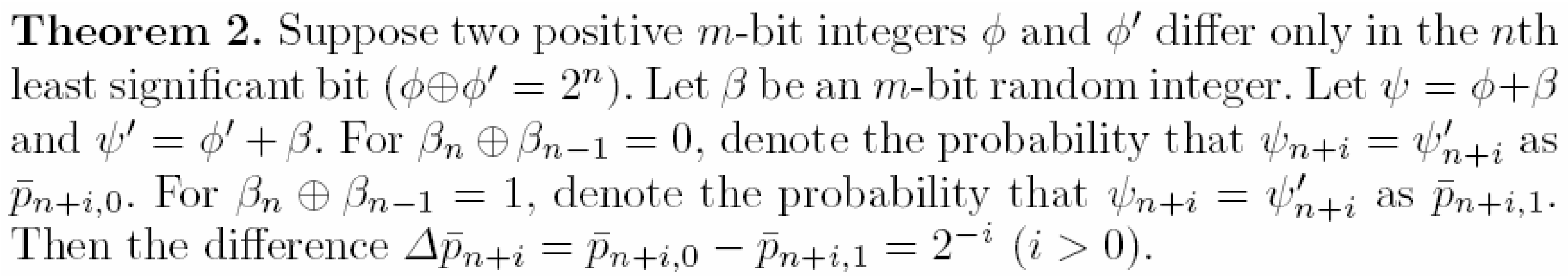
Observation:

addend bits strongly correlated with the difference of the sums

=> By observing the distribution of the difference of the sums, the value of addend bits can be determined with the linear attack technique

3. Differential Propagation of Addition

The following theorem shows that the check sum of two adjacent addend bits does affect significantly the distribution of the difference of the sums



Theorem 2. Suppose two positive m-bit integers ϕ and ϕ' differ only in the nth least significant bit ($\phi \oplus \phi' = 2^n$). Let β be an m-bit random integer. 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 $\Delta\bar{p}_{n+i} = \bar{p}_{n+i,0} - \bar{p}_{n+i,1} = 2^{-i}$ $(i > 0)$.

4. A Basic Attack on Phelix (1)

1) Introducing one bit difference in P_i

2) $B_3^{(i+1)} \oplus B_3'^{(i+1)}$ heavily biased

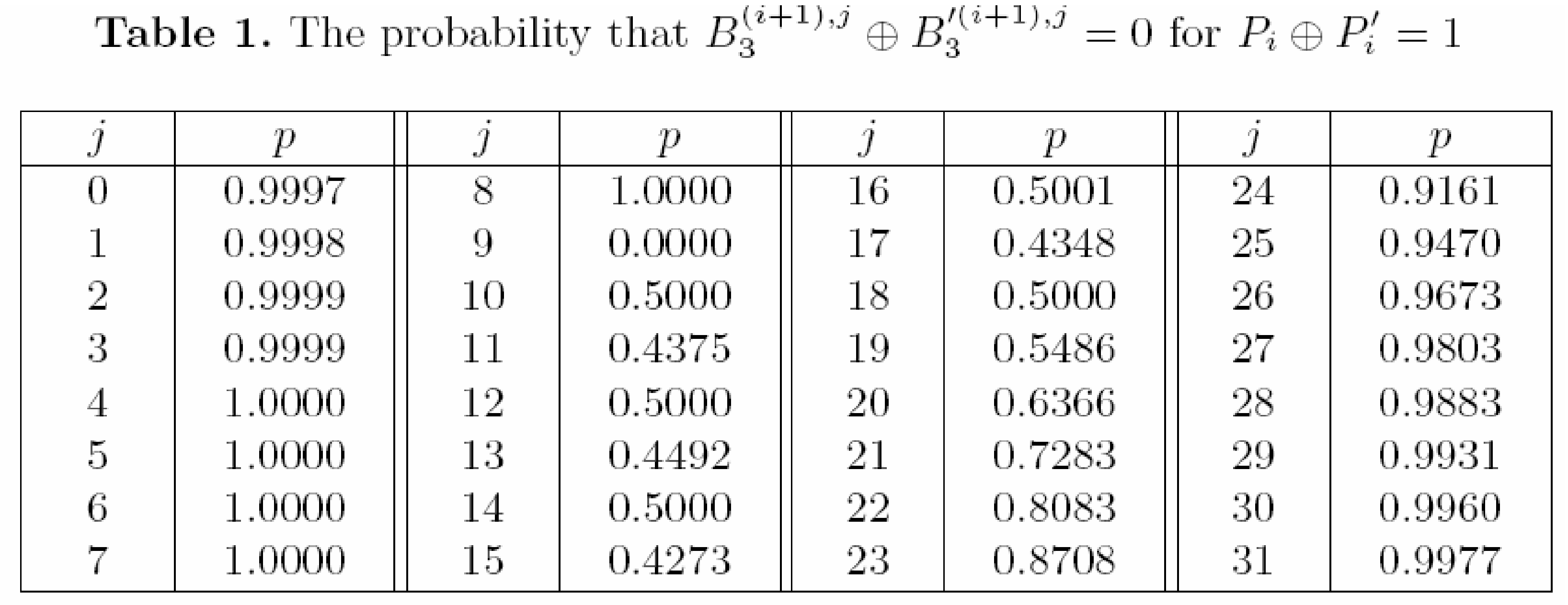


Table 1. The probability that $B_3^{(i+1),j} \oplus B_3'^{(i+1),j} = 0$ for $P_i \oplus P_i' = 1$

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

4. A Basic Attack on Phelix (2)

3) Since $T_0^{(i+1)} = A_0^{(i+1)} \oplus (B_3^{(i+1)} + X_{i+1,0})$ and that $B_3^{(i+1)} \oplus B_3'^{(i+1)}$ is heavily biased, we can predict which bits of $X_{i+1,0}$ may have significant effect on the distribution of the difference of the keystream according to Theorem 2.

4. A Basic Attack on Phelix (3)

4) When the one-bit difference is in the least significant bit of P_i, for $X^{15}_{i+1,0} \oplus X^{14}_{i+1,0} = 0$, the 17th least significant bit of $S_{i+1} \oplus S'_{i+1}$ is 0 with probability 0.50227; for $X^{15}_{i+1,0} \oplus X^{14}_{i+1,0} = 1$, the probability is 0.50117

=> The value of $X^{15}_{i+1,0} \oplus X^{14}_{i+1,0}$ is highly correlated to the distribution of $S^{17}_{i+1} \oplus S'^{17}_{i+1}$.

=> Recovering $X^{15}_{i+1,0} \oplus X^{14}_{i+1,0}$ with $2^{22.3}$ plaintext pairs

4. A Basic Attack on Phelix (4)

Experiment 1. With 2^{25} chosen plaintext pairs with difference in the least significant bit of P_i, the values of $X_{i+1,0}^{15} \oplus X_{i+1,0}^{14}$ of 192 keys among 200 keys are determined correctly.

The success rate is about 0.96. Lower than expected.

Reason: the other bits of $X_{i+1,0}$ interfere with $X_{i+1,0}^{15} \oplus X_{i+1,0}^{14}$

Shifting the one-bit difference, 23 bits of $X_{i+1,0}$ are recovered.

5. Improving the Attack on Phelix (1)

Aims:

- Recovering more key bits and improving the success rate
- Reducing the number of chosen plaintext pairs

Methods:

- Recovering $Z_4^{(i-3)}$ before recovering $X_{i+1,0}$
- Fine tuning of the threshold values in the attack

5. Improving the Attack on Phelix (2)

Recovering $Z_4^{(i-3)}$

1) Introducing difference in the least significant bit of P_i
2) $Y_4^{(i+1)} \oplus Y_4'^{(i+1)}$ is heavily biased

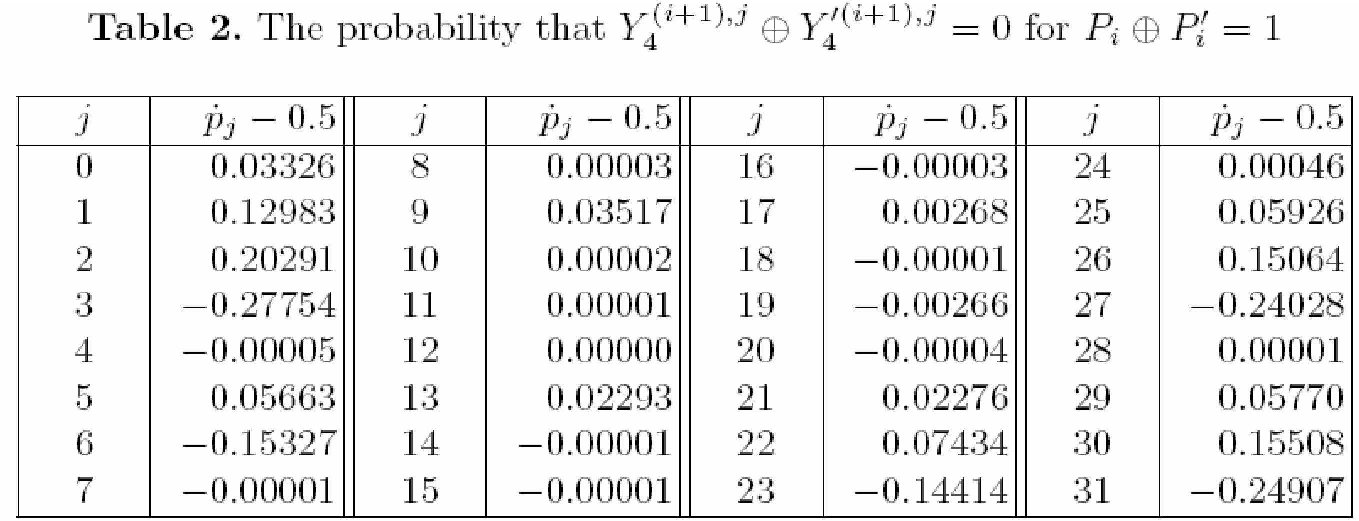


Table 2. The probability that $Y_4^{(i+1),j} \oplus Y_4'^{(i+1),j} = 0$ for $P_i \oplus P_i' = 1$

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

5. Improving the Attack on Phelix (3)

3) Since $Y_4^{(i+1)} \oplus Y_4'^{(i+1)}$ is heavily biased and $S_{i+1} = Y_4^{(i+1)} \oplus Z_4^{(i-3)}$, the value of the bits of $Z_4^{(i-3)}$ affects the distribution of $S_{i+1} \oplus S'_{i+1}$

4) When $P_i \oplus P_i' = 1$, for $Y_4^{(i+1),3} \oplus Y_4^{(i+1),2} = 0$, the 5th least significant bit of $S_{i-3} \oplus S'_{i-3}$ is 0 with probability 0.5461; for $Y_4^{(i+1),3} \oplus Y_4^{(i+1),2} = 1$, this probability is 0.5193
=> Recovering $Y_4^{(i+1),3} \oplus Y_4^{(i+1),2}$ requires 2^{14} plaintext pairs

5. Improving the Attack on Phelix (4)

5) In the attack, we determine the least significant bit of $Z_4^{(i-3)}$ first, then proceed to determine the more significant bits of $Z_4^{(i-3)}$ by shifting the one-bit difference.

6) When $Z_4^{(i-3),j}$ is analyzed, $Z_4^{(i-3),j-1} Z_4^{(i-3),j-2} \cdots Z_4^{(i-3),0}$ is subtracted from S_i and S_i' so that $Z_4^{(i-3),j-1} Z_4^{(i-3),j-2} \cdots Z_4^{(i-3),0}$ does not interfere with $Z_4^{(i-3),j}$. The success rate becomes very close to 1 with small number of plaintext pairs.

5. Improving the Attack on Phelix (5)

7) With 2^{17} plaintext pairs, 30 bits of $Z_4^{(i-3)}$ (except the two most significant bits of $Z_4^{(i-3)}$) can be determined with success rate about 0.999.

After recovering $Z_4^{(i-3)}$, we recover $X_{i+1,0}$ from the distribution of $Y_4^{(i+1)} \oplus Y'^{(i+1)}_4$ instead of $S_{i+1} \oplus S'_{i+1}$.

5. Improving the Attack on Phelix (6)

Recovering $X_{i+1,0}$ from $Y_4^{(i+1)} \oplus Y_4'^{(i+1)}$

Due to the interference between the bits of $X_{i+1,0}$ on the distribution of $Y_4^{i+1} \oplus Y_4'^{i+1}$, we need the fine tuning of the threshold values in the attack.

For example, when $P_i \oplus P_i' = 1$, if $X_{i+1,0}^9 = 0$ and the value of $X_{i+1,0}^{11} \| X_{i+1,0}^{10}$ is 00, 11, 01, 10, then $Y_4^{i+1,13} \oplus Y_4'^{i+1,13} = 0$ with prob. 0.53033, 0.52334, 0.51946, 0.51864; if $X_{i+1,0}^9 = 1$, the prob. becomes 0.52334, 0.53030, 0.51861, 0.51948.

=> $X_{i+1,0}^9$ and $X_{i+1,0}^{11} \| X_{i+1,0}^{10}$ affect the distribution of $Y_4^{i+1,13} \oplus Y_4'^{i+1,13}$

5. Improving the Attack on Phelix (7)

Other bits of $X_{i+1,0}$ also affect the distribution of $Y_4^{i+1,13} \oplus Y_4'^{i+1,13}$

In the attack, we need to tune the threshold value to 0.52035, so that the value of $X_{i+1,0}^{11} \oplus X_{i+1,0}^{10}$ can be recovered with success rate 0.99 with 2^{21} chosen plaintext pairs.

The values of $X_{i+1,0}^{j+1} \oplus X_{i+1,0}^{j}$ $(2 \le j \le 28)$ can be determined in a similar way

5. Improving the Attack on Phelix (8)

The lsb $X_{i+1,0}^{0}$ is recovered in a different way:

2^{16} chosen plaintext pairs with $P_i \oplus P_i' = 2^{21}$

observing the distribution of $Y_4^{(i+1),2} \oplus Y_4'^{(i+1),2}$

The second lsb $X_{i+1,0}^{1}$ can be recovered if $X_{i+1,0}^{0} = 0$

$2^{16.4}$ chosen plaintext pairs with $P_i \oplus P_i' = 2^{22}$

observing the distribution of $Y_4^{(i+1),3} \oplus Y_4'^{(i+1),3}$

The value of $X_{i+1,0}^{2}$ can be recovered if $X_{i+1,0}^{0} = 0$ and $X_{i+1,0}^{1} = 0$

5. Improving the Attack on Phelix (9)

The above attack recovers 28.75 bits of $X_{i+1,0}$

After recovering eight consecutive $X_{i+1,0}$, 230 key bits are recovered. Considering the error rate of about 0.01, the effective key size is reduced to 41.5 bits.

The attack requires $2^{32.7}$ chosen plaintext pairs.

6. Improving the Security of Phelix

Problem in Phelix: The plaintext affects the keystream before passing through enough confusion and diffusion operations

Solution 1: plaintext passing through more operations
=> resulting in slow cipher

Solution 2: using strong one-way function to generate the
initial internal state from key and IV
=> secure against key recovery attack
but the leaked internal state allows message forgery

7. Open problems (1)

Open Problem 1.

How to design an efficient stream cipher with embedded MAC, secure against the key recovery attack in the applications where an attacker has the ability to control the nonce generation for a while?

Helix and Phelix are insecure in these applications

7. Open problems (2)

Open Problem 2.

How to design an efficient stream cipher with embedded MAC, secure against the key recovery attack only in the applications where the nonce generation is secure

Helix and Phelix are secure in these applications.

But there may be dedicated and more efficient designs

8. Conclusion

1. The computational complexity of the attack against Phelix is $2^{41.5}$, less than the 2^{88} operations required to break Helix

   => Phelix fails to strengthen Helix in this respect

2. Open problems: Efficient embedded MACs for stream cipher

Thank you!

Q & A