Linear Cryptanalysis of FEAL 8X – Winning the FEAL 25 Years Challenge

Yaniv Carmeli

Joint work with Prof. Eli Biham

CRYPTODAY 2014

FEAL

FEAL

- Published in 1987, designed by Miyaguchi and Shimizu (NTT).
- 64-bit block cipher family with the Feistel structure.
- Key size was initially 64 bits, later extended to 128 bits as FEAL-NX.
- Had major contributions to the history of block ciphers.
- Inspired many new ideas, including differential and linear cryptanalysis.

Previous Attacks on FEAL-8

- 1000 Chosen-Plaintexts – Differential Cryptanalysis [Biham Shamir 91]
- 2^{24} Known Plaintexts – Linear Cryptanalysis [Biham 94]
- 2^{15}-2^{28} Known-Plaintexts with high time complexity [Matsui Yamagishi 92].
  - But the time complexity is 2^{50} or higher.

THE FEAL-8X CHALLENGE

Celebrating the 25th year of FEAL

- A New Prize Problem -

August 21 2012
Mitsuru Matsui
Mitsubishi Electric Corporation

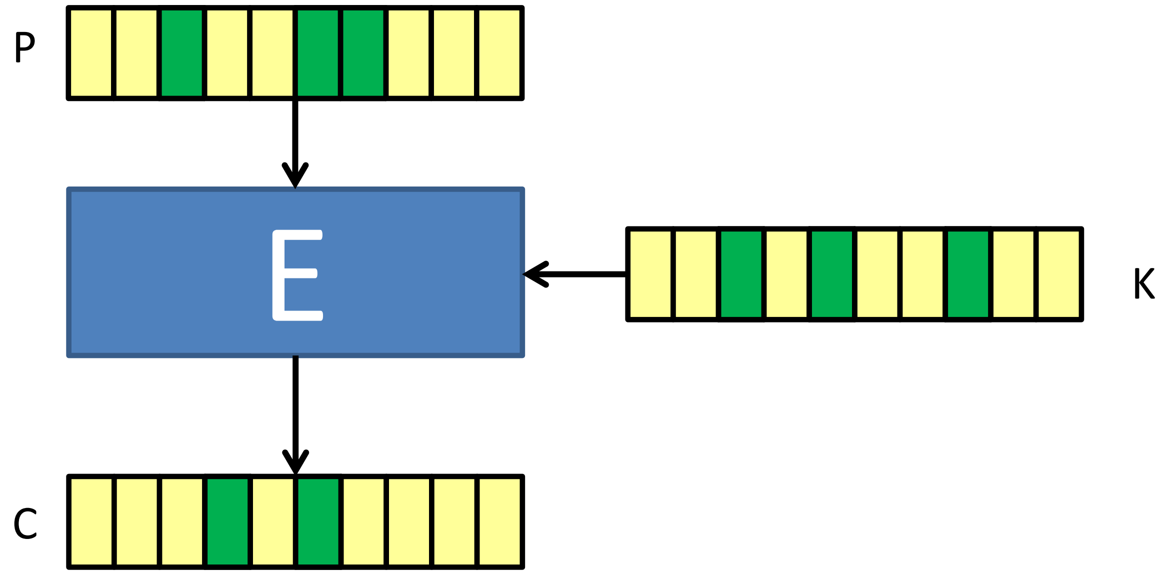
The New Prize Problem

- The target cipher: FEAL-8X
  - FEAL cipher with 8 rounds and 128-bit key
  - Same as FEAL-8 except its key scheduling part
- 2^b plaintext-ciphertext pairs are given ($b \leq 20$).
- Good news: winner (min b, first) receives $1500.
- Bad news: brute force is infeasible (128-bit key)
- Deadline: CRYPTO 2013
- For more details, see

  https://docs.google.com/open?id=0B3xMqN36HCf2eDVzb191R1VHY0k

LINEAR CRYPTANALYSIS

Linear Biases



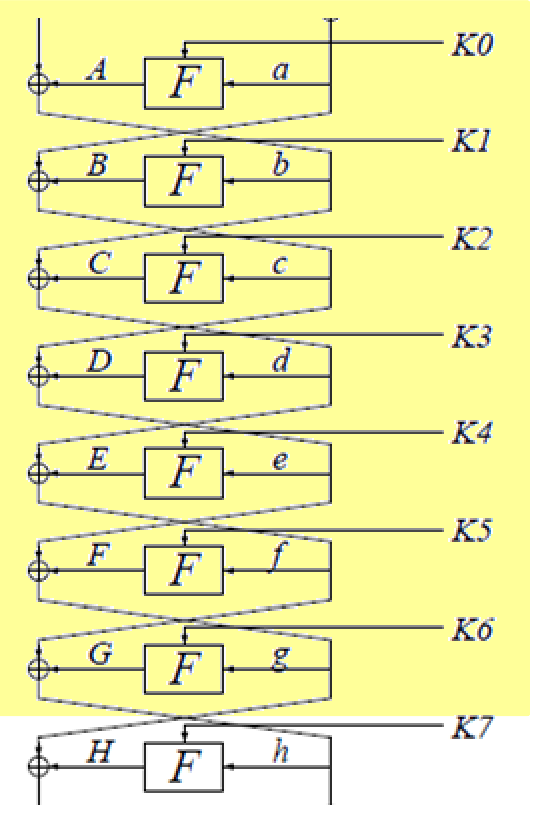
$$p\left(P_3 \oplus P_4 \oplus P_7 \oplus C_4 \oplus C_6 \oplus K_2 \oplus K_5 \oplus K_7 = 0\right) = \frac{1}{2} \pm b$$

Linear Attacks

- What can we do with a known linear bias of the cipher?
  - Learn one bit of information about the key
  - Build a distinguisher
  - Using a distinguisher for key-recovery (last round attack)

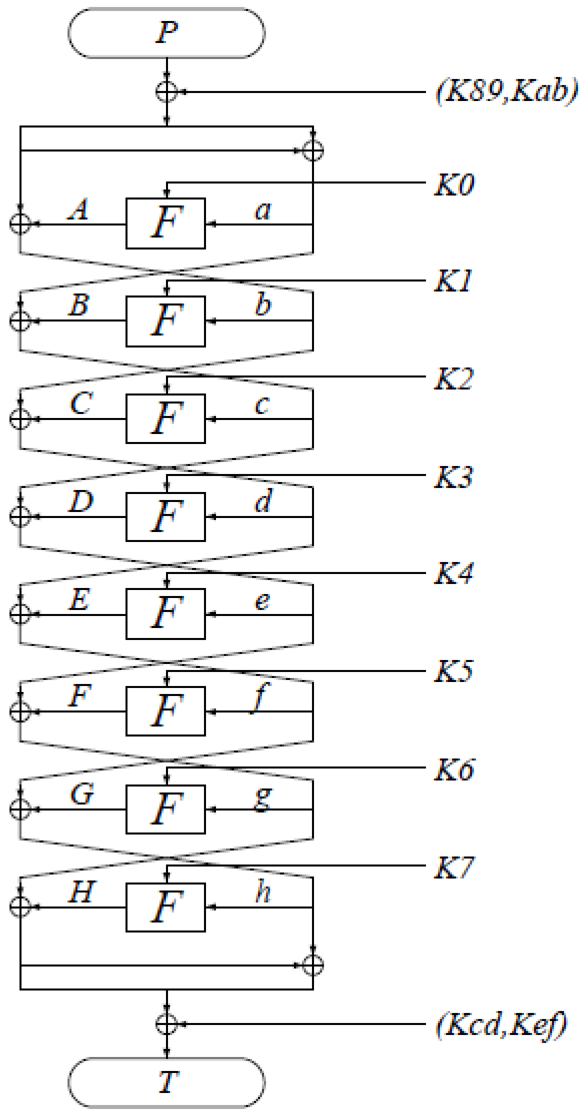
Last Round Attack

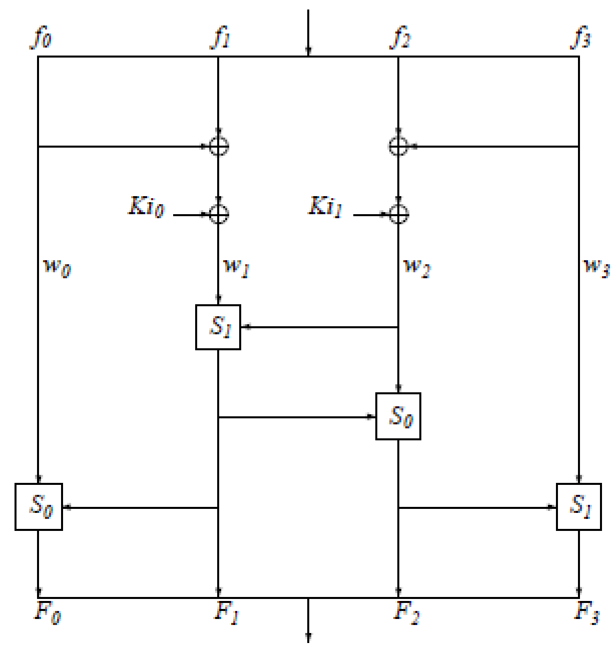
- Cipher of N rounds
- Distinguisher for first N-1 rounds
- Guess the subkey of the last round, decrypt the messages and use the distinguisher to check the guess



FEAL AND THE EQUIVALENT DESCRIPTIONS

FEAL-8X



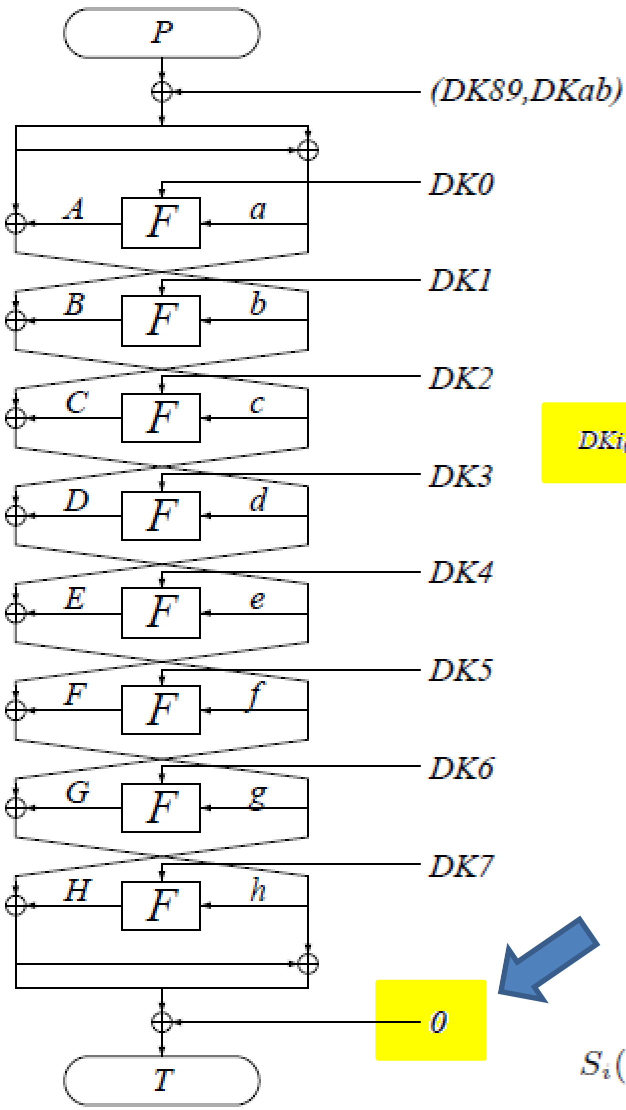


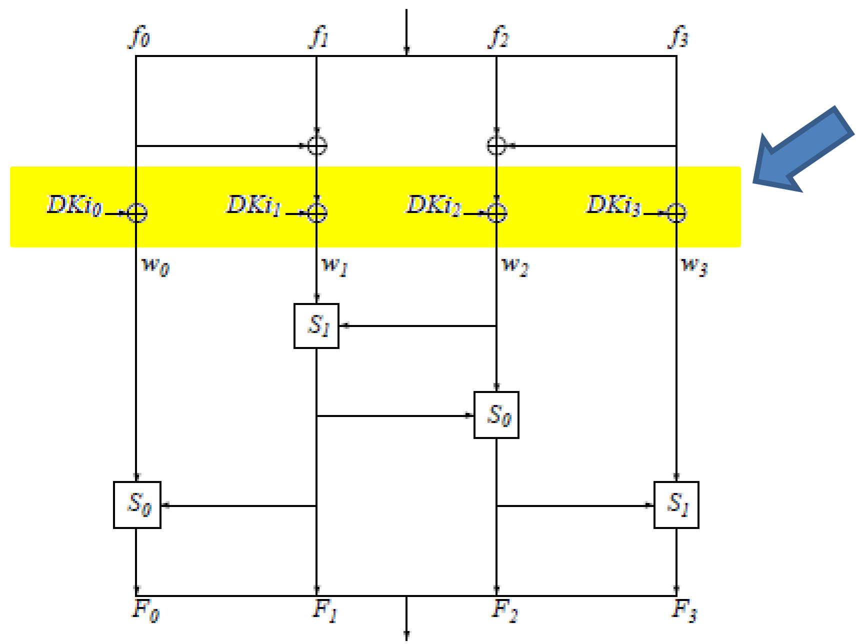
$$S_i(x,y) = \mathrm{ROL2}(x+y+i \pmod{256})$$

FEAL-8X Equivalent Descriptions

- Eliminate the whitening keys on the plaintext side.
  - Using an equivalent description with 32-bit subkeys, EK0–EK7.
  - Useful when analyzing the first round.
- Similarly, we can eliminate the whitening keys on the ciphertext side.
  - Using DK0–DK7.
  - Useful when analyzing the last round.

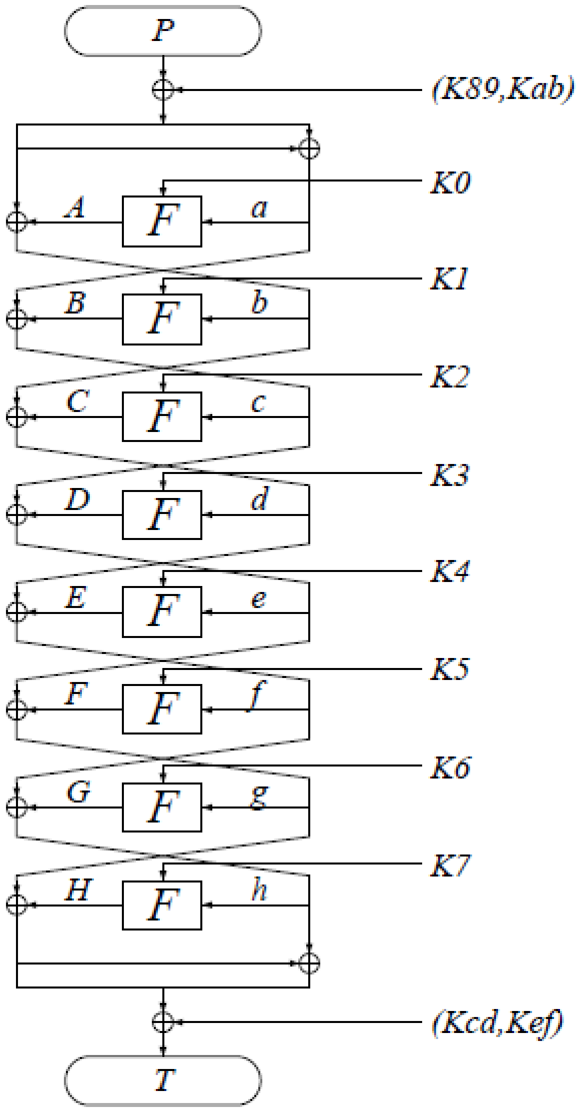
FEAL-8X Equivalent Description

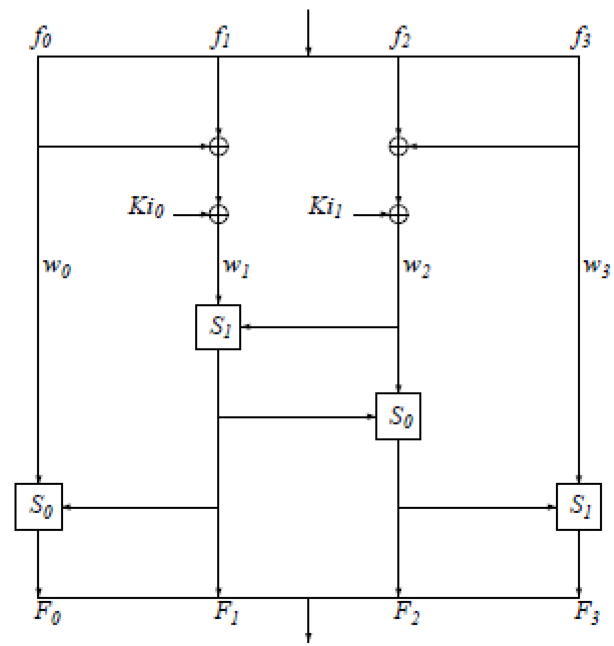




$$S_i(x, y) = \mathrm{ROL2}(x + y + i \pmod{256})$$

FEAL-8X





$$S_i(x, y) = \mathrm{ROL2}(x + y + i \pmod{256})$$

The Subkeys of the Equivalent Descriptions

Subkeys of FEAL-8X	Equivalent description without whitening at the beginning	Equivalent description without whitening at the end
$K89ab$	0	$(K89 \oplus Kcd \oplus Kef, Kab \oplus Kef)$
$K0$	$EK0 = mw(K0, K89 \oplus Kab)$	$DK0 = mw(K0, Kcd)$
$K1$	$EK1 = mw(K1, K89)$	$DK1 = mw(K1, Kcd \oplus Kef)$
$K2$	$EK2 = mw(K2, K89 \oplus Kab)$	$DK2 = mw(K2, Kcd)$
$K3$	$EK3 = mw(K3, K89)$	$DK3 = mw(K3, Kcd \oplus Kef)$
$K4$	$EK4 = mw(K4, K89 \oplus Kab)$	$DK4 = mw(K4, Kcd)$
$K5$	$EK5 = mw(K5, K89)$	$DK5 = mw(K5, Kcd \oplus Kef)$
$K6$	$EK6 = mw(K6, K89 \oplus Kab)$	$DK6 = mw(K6, Kcd)$
$K7$	$EK7 = mw(K7, K89)$	$DK7 = mw(K7, Kcd \oplus Kef)$
$Kcdef$	$(K89 \oplus Kab \oplus Kcd, Kab \oplus Kef)$	0

- Linear translation between the subkeys of the three descriptions.
  - Note that mw() is linear.

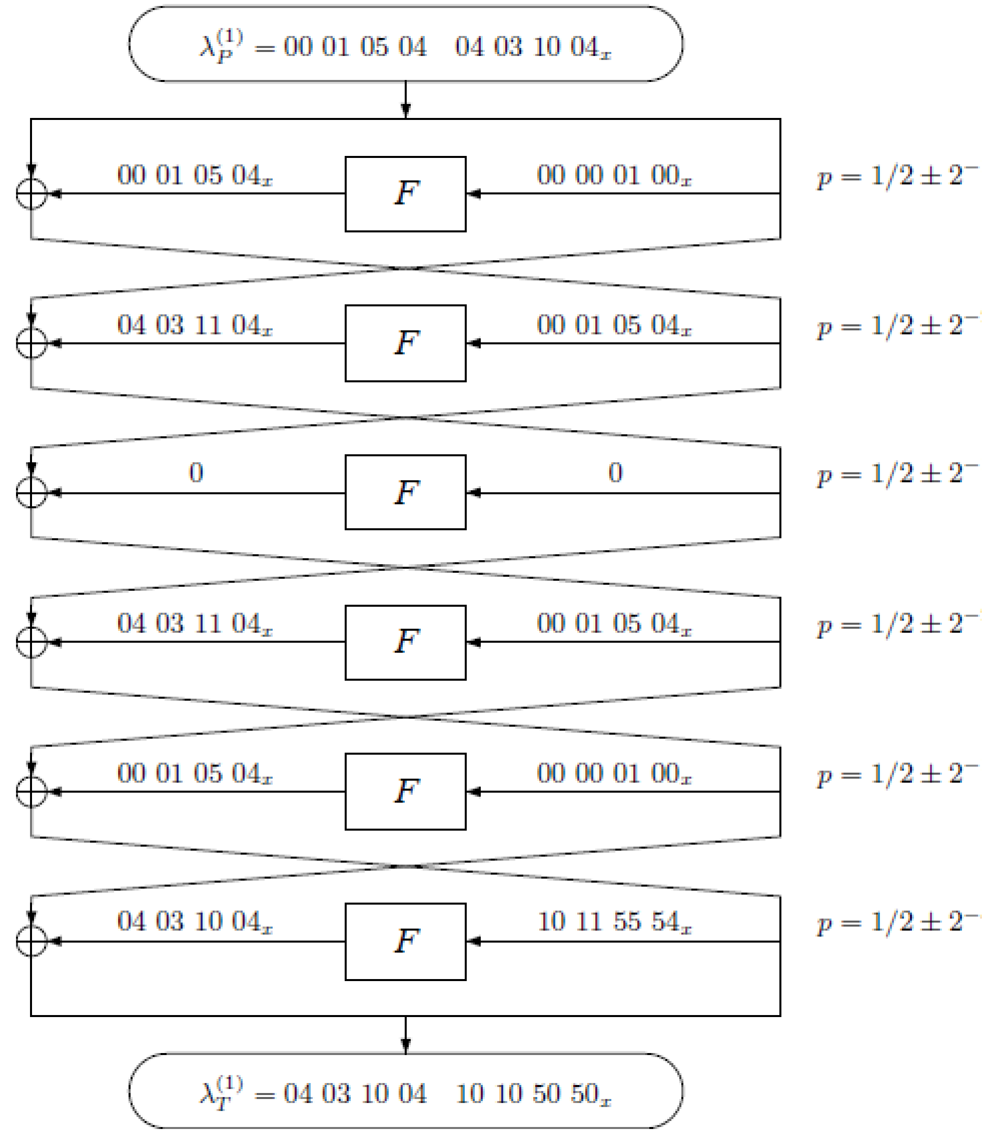
A BASIC LINEAR ATTACK ON FEAL-8X

2^{15} Known Plaintexts, 26 hours of computation

The Approximation

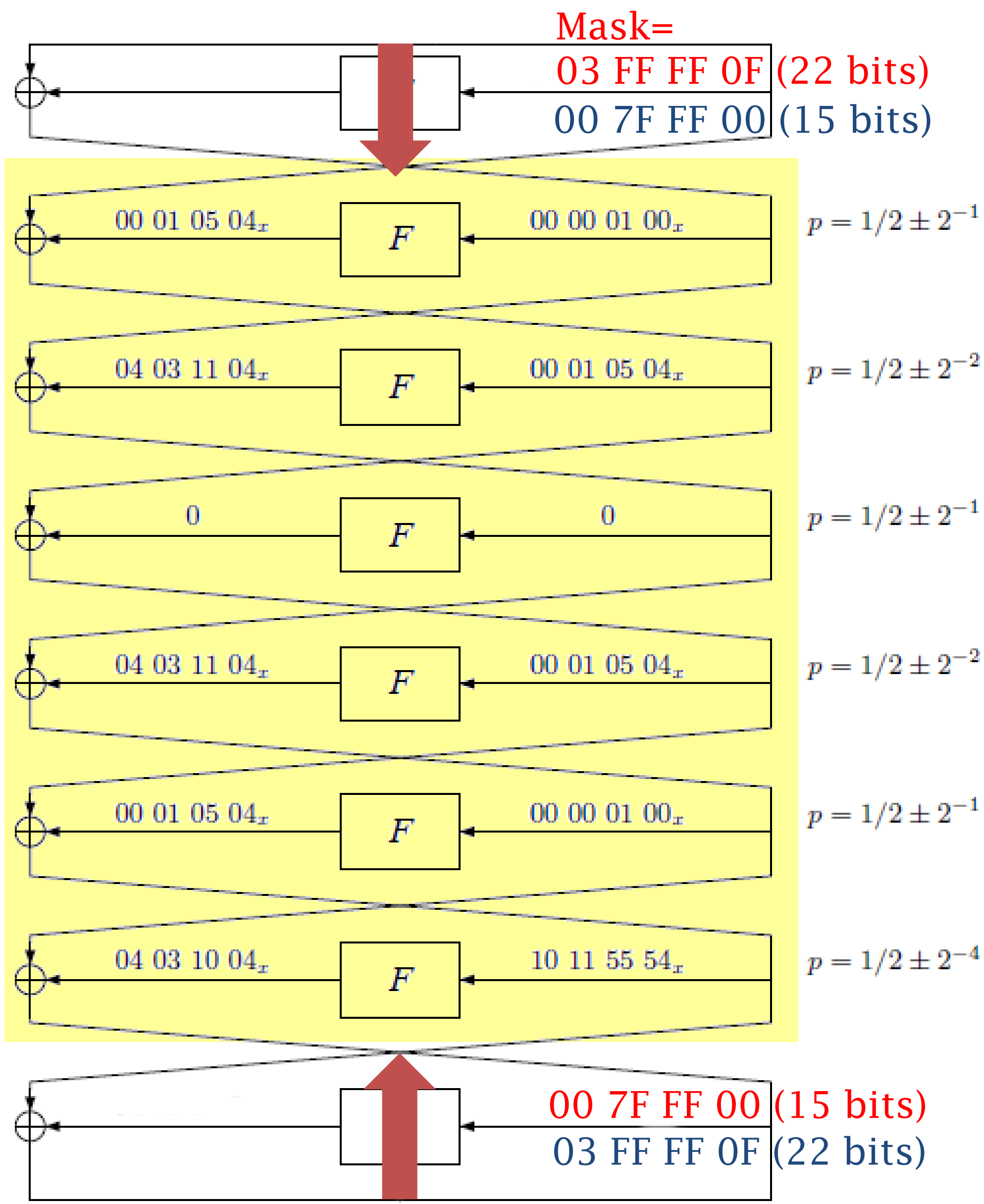
- 6-round approximation by Aoki, Moriai, Matsui et al.
  - Bias of 2^{-6}

- We can also use the reverse approximation.



Basic Attack

- Standard linear attack.
  - Guess subkeys of first and last rounds (EK0 and DK7)
    - 22 bits of EK0
    - 15 bits of DK7
  - Repeat with the reverse approx.
    - 30 bits overlap



The Attack

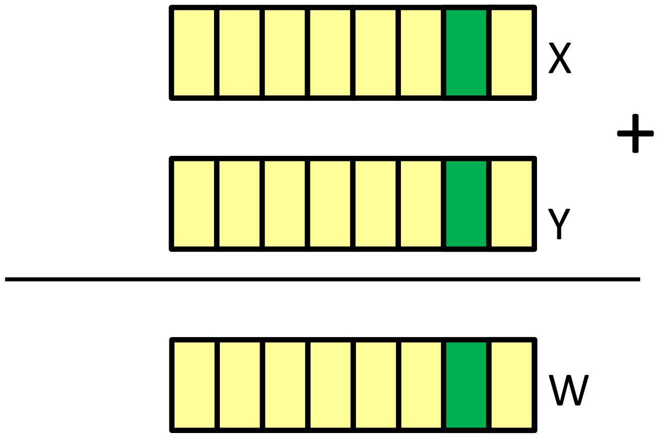
- Try all 2^{37} choices of the 22+15=37 bits of the first and last actual subkeys (DK7 and EK0)
  - A few of those bits have only a small effect on the results
- The bias of the approximation is 2^{-6}
  - About 2^{15} known plaintexts are required to recover the 37 bits
  - In practice, the result is not unique
- So we apply twice
  - Once for each approximation
  - Take the result that matches in the 30 common bits

Recovering the rest of the subkeys

- It is hard to complete the first and last subkeys at this time
- So, using the known bits, we recover bits of DK6
- And then complete further bits of DK7
- Then Complete DK5, DK4, DK3, DK2, DK1, EK0, EK1, EK2, EK3
- Finally, recover the FEAL-8X key from these subkeys.

LINEAR PROPERTIES OF ADDITION

(Bitwise) Linear Properties of Addition

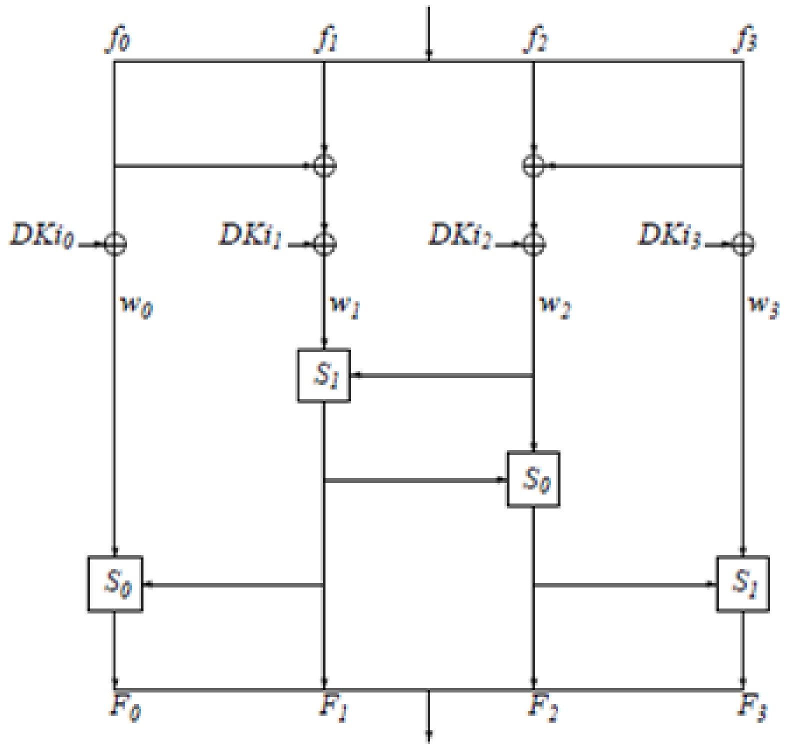


$$p(x_0 \oplus y_0 \oplus z_0 = 0) = \frac{1}{2} + \frac{1}{2}$$

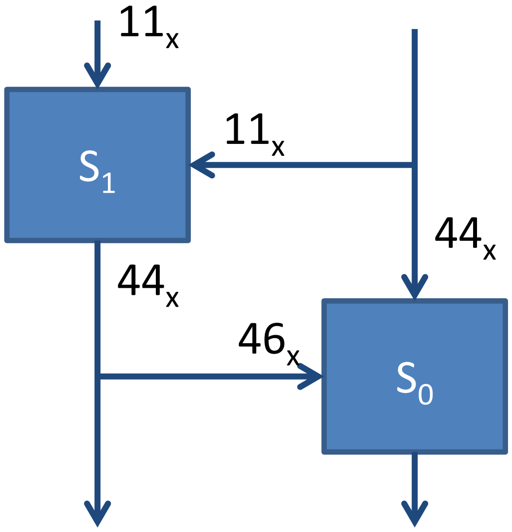
$$p(x_1 \oplus y_1 \oplus z_1 = 0) = \frac{1}{2} + \frac{1}{4}$$

THE PARTITIONING TECHNIQUE

2^{14} Known Plaintexts, 14 hours of computation



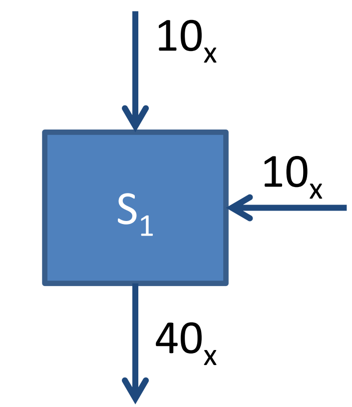
$$S_i(x, y) = \mathrm{ROL2}(x + y + i \pmod{256})$$



The two middle S-boxes of the seventh round

Partition – The Case of S_1

- The approximation of S_1 in the seventh round includes 10 10→40
- Consider Bit 3 of x, Bit 3 of y and the carry to Bit 4 of the sum:



$S_1(x,y)=(x+y+1) \ll 2$

Will there be carry to Bit 4?

Bits 3 of x, y	0	1
0	NEVER	Sometimes
1	Sometimes	ALWAYS

Unpredictable parity

Behaves according to the linear approximation

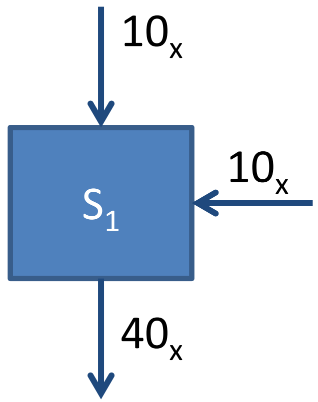
Reverses the parity

Partition – The Case of S_1

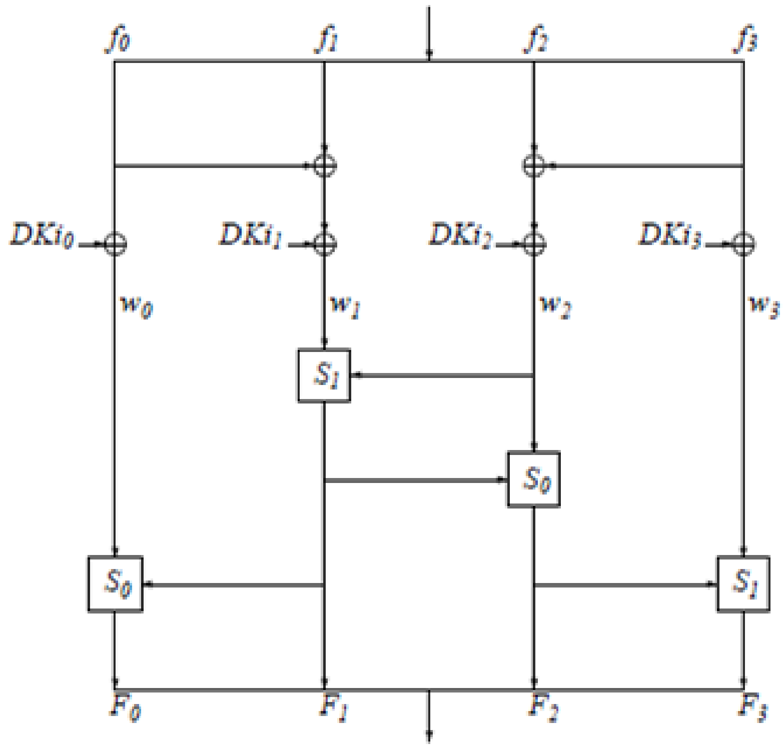
Will there be carry to Bit 4?

Bits 3 of x, y	0	1
0	NEVER	Sometimes
1	Sometimes	ALWAYS

- Given the bits we guess in the 8^{th} round, we can partition all plaintexts into four sets.
  - In the "yellow" sets the bias is now two times higher, since we discarded some of the "noise".
  - ➔We need half the data size.
    - Because each set contains a quarter of the original data size.
  - Notice that we do not know which set is which.



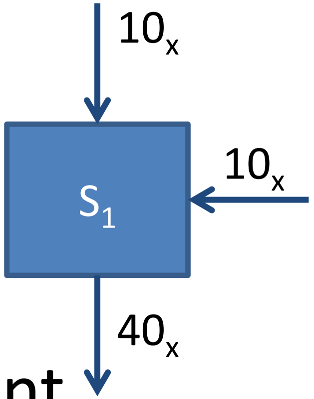
$S_1(x,y)=(x+y+1) <<2$



$$S_i(x, y) = \mathrm{ROL2}(x + y + i \pmod{256})$$

Partition – The Case of S_1

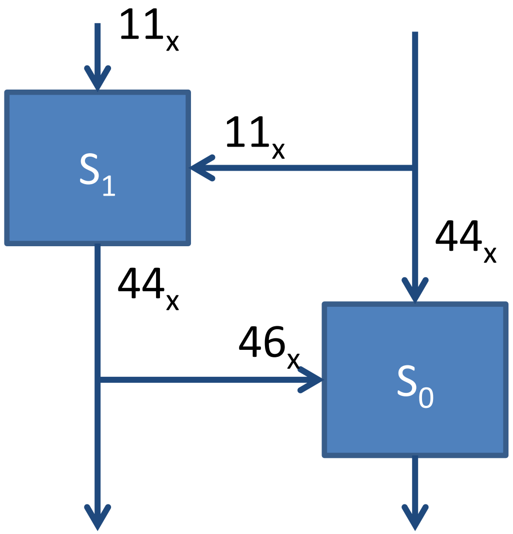
- While this method works for a single S-box, it does not work for the entire F-Function!



- The two middle S-boxes are not independent
  - The second S-box inverts this effect

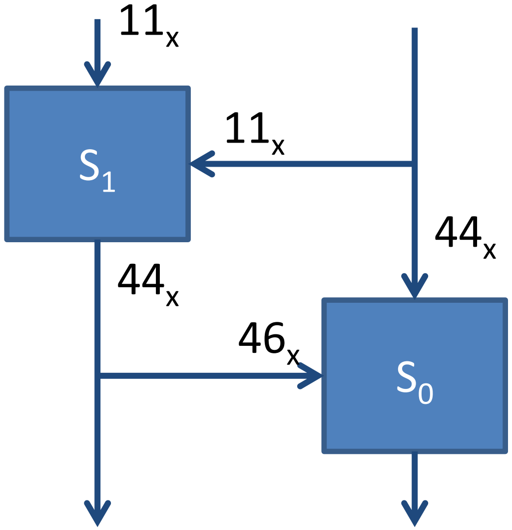
The Partition

- Analyzing the joint distribution of the middle S-boxes as a single big S-box:
  - In the “yellow” sets (0+0 or 1+1), the correlation between S_0 and S_1 causes the overall linear bias to be close to zero.
  - Analyzing the joint distribution of the middle S-boxes shows that the “green” (0+1 or 1+0) sets are those that amplify the bias.



The Partition

Bits 3 of x, y	0	1
0	No bias	Amplified bias
1	Amplified bias	No bias



- We partition the known plaintexts into two sets, according to the XOR of Bit 3 of x and Bit 3 of y.
  - Again, without guessing additional key bits we do not know in advance which of the sets is the green (0+1 and 1+0) one and which is the yellow (0+0 and 1+1).
- We thus compute the bias in each set separately.

QUESTIONS?