Selected Areas in Cryptology Cryptanalysis Week 3

Marc Stevens

stevens@cwi.nl

https://homepages.cwi.nl/~stevens/mastermath/

Linear cryptanalysis

$P_5 \oplus P_7 \oplus P_8 \oplus X_{42,2} \oplus X_{42,4} \oplus X_{44,2} \oplus X_{44,4}$

$\oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$

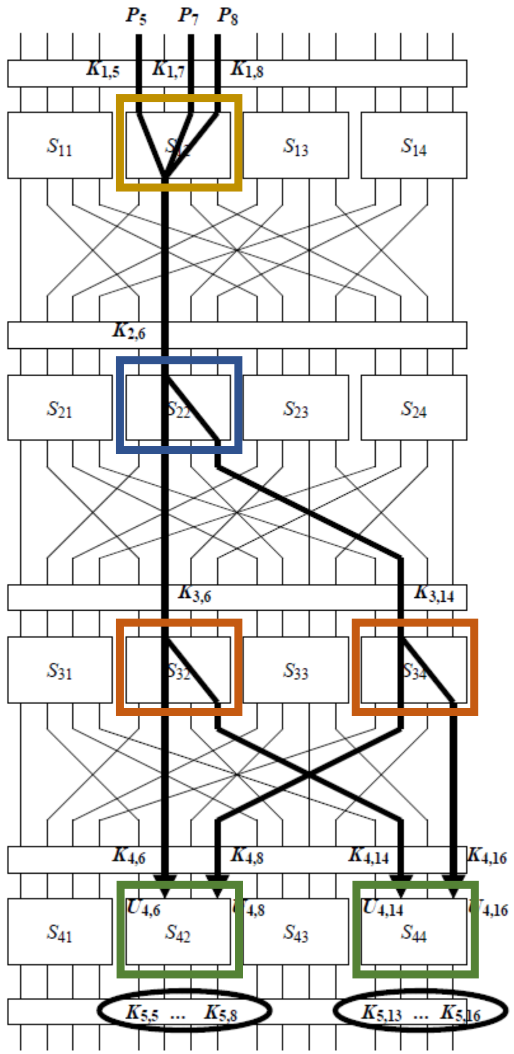
With bias: $2^3 \left(\frac{1}{4}\right)\left(-\frac{1}{4}\right)^3 = -\frac{1}{32}$

Build distinguisher for 3 rounds (w/ 4 key additions)

- Over many plaintext-ciphertext pairs measure probability of relation
- Is $\approx 0.5 \pm \frac{1}{32} \Rightarrow$ is blockcipher oracle with 3 rounds
- Is $\approx 0.5 \Rightarrow$ random oracle

Key-recovery attack idea:

1. Obtain many plaintext-ciphertext pairs
2. Guess last round key => decrypt last round
   - Note how we only need to guess 8 key bits of K_5
3. Do distinguishing check
   - Outputs blockcipher oracle
     $\Rightarrow$ right key guess, stop
   - Outputs random oracle
     $\Rightarrow$ wrong key guess, try again with another guess



Extending the Key-Recovery Attack

Break all round keys:

1. Break the entire last round key
   - Use other linear relations with high bias to learn more bits of last round key

2. Strip last round of all ciphertexts using last round key

3. Repeat attack for r−1 rounds
   using linear approximations over $r - 2$ rounds

Space of linear relations

- We've looked at 1 linear relation with high bias
  - Input & output bits: $P_5 \oplus P_7 \oplus P_8 \oplus X_{42,2} \oplus X_{42,4} \oplus X_{44,2} \oplus X_{44,4}$
  - Key bits: $K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$
  - Note that the involved key bits uniquely determine the trail
  - Bias **estimated** based on 1 trail

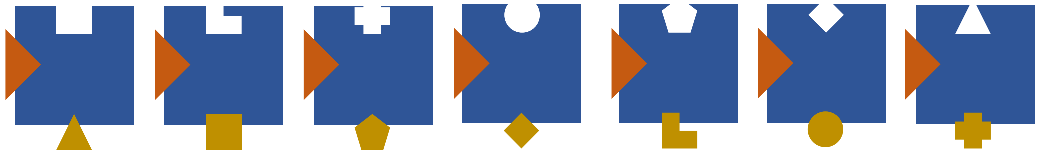
What about other linear relations and trails?

- Relations with same plaintext and round 4 bits:
  - Problematic as total bias on plaintext and round 4 bits depend on all such trails
  - If single high bias then this is a good first approximation
  - If multiple high biases then these can cancel/interfere into low bias or add/strengthen to even higher bias
- Relations with same round 4 active S-Boxes ⇒ Need fewer P-C pairs
  - Independent distinguishers can be used together to get higher confidence on correct key guess
- Relations with other round 4 active S-Boxes ⇒ Learn other key bits

Structural attacks: linear & differential cryptanalysis

1. Analyze individual rounds with probabilistic input/output-relation

2. Obtain a family of round attack building blocks

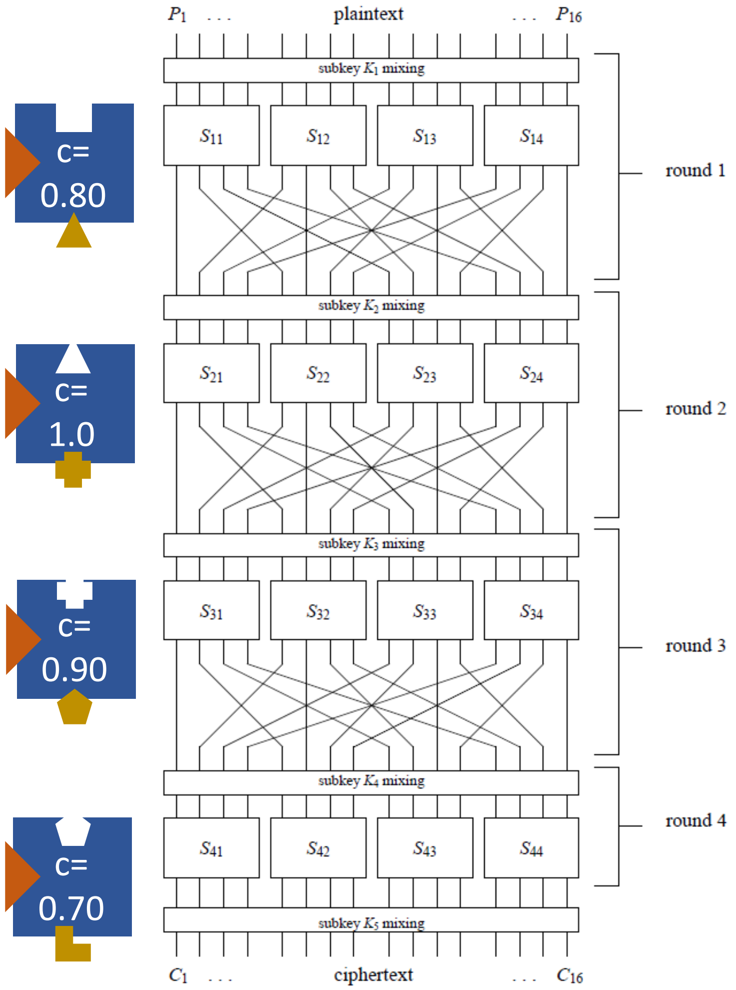


3. Combine to attack on full blockcipher

4. Approximate complexity
   by combining individual round costs

$$C = c(r) \cdot 0.8 \cdot 1.0 \cdot 0.9 \cdot 0.7$$

5. Find optimal attack



Differential Cryptanalysis

Consider two related encryptions:

1. $C = Enc_K(P)$ (with internal variables $X, Y, \ldots$)
2. $C' = Enc_K(P')$ (with internal variables $X', Y', \ldots$)

- Define difference variable: $\Delta X = X \oplus X'$

- Study relations between input difference ΔP and output difference ΔC:
  - $p_{\Delta P, \Delta C} := \Pr[\, \Delta C \mid \Delta P \,] = \Pr_P[Enc_K(P) \oplus Enc_K(P \oplus \Delta P) = \Delta C]$
  - Ideal secure situation:
    for every ΔP every ΔC is equally likely: $p_{\Delta P, \Delta C} \approx 2^{-n}$

Differential Cryptanalysis

- Differences are not affected by:
  - Key-addition:

$$Y = X \oplus K, \qquad Y' = X' \oplus K$$

$$\Rightarrow \qquad \Delta Y = X \oplus K \oplus X' \oplus K = \Delta X$$

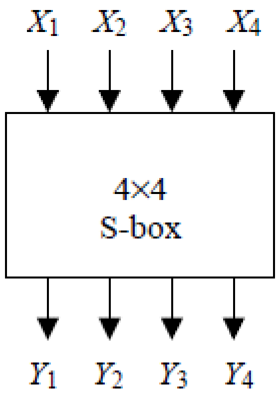
  - State permutation:

$$Y[i] = X[\,\pi_P(i)\,], \qquad Y'[i] = X'[\,\pi_P(i)\,]$$

$$\Rightarrow \qquad \Delta Y[i] = \Delta X[\,\pi_P(i)\,]$$

DDT: Difference Distribution Table

- Analyze all differential relations for S-Box π_S of the form:
  - $p_{\Delta X,\Delta Y} = \Pr_X[\Delta Y = \pi_S(X) \oplus \pi_S(X \oplus \Delta X)]$
- S-Box is permutation on $\{0000_b, \dots, 1111_b\}$
  - 16 possible input differences $\Delta X \in \{0000_b, \dots, 1111_b\}$
  - 16 possible output differences $\Delta Y \in \{0000_b, \dots, 1111_b\}$
  - Represent $\Delta X, \Delta Y$ as integer value: $1000_b = 8, \quad 0011_b = 3$



- Difference Distribution Table (DDT):
  - 16 x 16 table
  - Row $I \in \{0, \dots, 15\}$, Column $J \in \{0, \dots, 15\}$ contains:
  - $DDT(I, J) := \#\{\, X \in \{0,1\}^4 \mid J = \pi_S(X) \oplus \pi_S(X \oplus I) \,\}$
  - Probability $p_{I,J} = \Pr_X[J = \pi_S(X) \oplus \pi_S(X \oplus I)] = DDT(I, J)/16$
  - Important tool!
    - Easily precomputed, independent of keys
    - Convenient look-up for large probabilities

DDT: Difference Distribution Table

- Compute entry given ΔX
  1. Write all values for X with corresponding Y-values
  2. Compute $X' = \mathrm{X} \oplus \Delta X$
  3. Compute Y' and $\Delta Y = Y \oplus Y'$
  4. Count occurrences of each ΔY

- $\Delta X = 1000_b$: occurrences
  - $1101_b: 4 \Rightarrow DDT(8{,}13) = 4$
  - $1110_b: 2 \Rightarrow DDT(8{,}14) = 2$
  - $1011_b: 4 \Rightarrow DDT(8{,}11) = 4$
  - $0111_b: 2 \Rightarrow DDT(8{,}7) = 2$
  - $0110_b: 2 \Rightarrow DDT(8{,}6) = 2$
  - $1111_b: 2 \Rightarrow DDT(8{,}15) = 2$
- Note: all counts are even:
  X' and X can swap values, while ΔX and ΔY remain the same

X	Y	X'	Y'	ΔY
0000	1110	1000	0011	1101
0001	0100	1001	1010	1110
0010	1101	1010	0110	1011
0011	0001	1011	1100	1101
0100	0010	1100	0101	0111
0101	1111	1101	1001	0110
0110	1011	1110	0000	1011
0111	1000	1111	0111	1111
1000	0011	0000	1110	1101
1001	1010	0001	0100	1110
1010	0110	0010	1101	1011
1011	1100	0011	0001	1101
1100	0101	0100	0010	0111
1101	1001	0101	1111	0110
1110	0000	0110	1011	1011
1111	0111	0111	1000	1111

DDT: Difference Distribution Table

DDT properties:

- Compute with sage (see lecture notes)
- $DDT(0,0) = 16, \;\; LAT(x,0) = 0, \;\; LAT(0,x) = 0, \; x > 0$

- Every entry is even
- Every entry is non-negative
- Sum of every row/columns = 16

Input sum \ Output sum	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
10	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
11	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
12	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
13	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
14	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
15	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Piling-Up Lemma

How to combine two differential relations ?

- Let X, Y, Z be the internal state after 1, 2 and 3 rounds

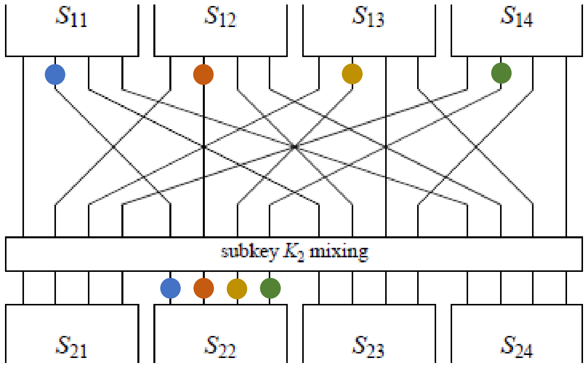
- Parallel combination:
  - Let $\Delta X[1,2,3,4] \Rightarrow \Delta Y[1,5,9,13]$ with probability p_3
  - Let $\Delta X[5,6,7,8] \Rightarrow \Delta Y[2,6,10,14]$ with probability p_4
  - Then $\Delta X[1,2,3,4,5,6,7,8] \Rightarrow \Delta Y[1,2,5,6,9,10,13,14]$ with probability $p_3 p_4$

- Sequential combination:
  - Let $\Delta X \Rightarrow \Delta Y$ with probability p_1
  - Let $\Delta Y \Rightarrow \Delta Z$ with probability p_2
  - Then $\Delta X \Rightarrow \Delta Z$ with probability $\geq p_1 p_2$
  - Why $\geq p_1 p_2$?
    - $\Pr[\Delta X \Rightarrow \Delta Z] = \sum_{\Delta Y} \Pr[\Delta X \Rightarrow \Delta Y \wedge \Delta Y \Rightarrow \Delta Z]$

Bringing everything together

- DDT to find those high probability S-Box relations
- Inactive S-Boxes don't affect probability, as:
  - $DDT(0,0) = 16 \Rightarrow p_1 = \frac{16}{16} = 1$
  - Piling-Up Lemma: $p_{1,2} = p_1 p_2 = p_2$
- Only active S-Boxes matter ⇒ minimize active S-boxes

- Make use of π_P properties
  - i-th output bit active difference of S-Box S_{1j} ⇒ S-Box S_{2i} active in next round
  - It is its own inverse, so also vice-versa:
  - i-th input bit active difference of S-Box S_{2j} ⇒ S-Box S_{1i} active in previous round



- If multiple active S-boxes in one round then try to have active input bits on same S-box bit position (and same for output bits)

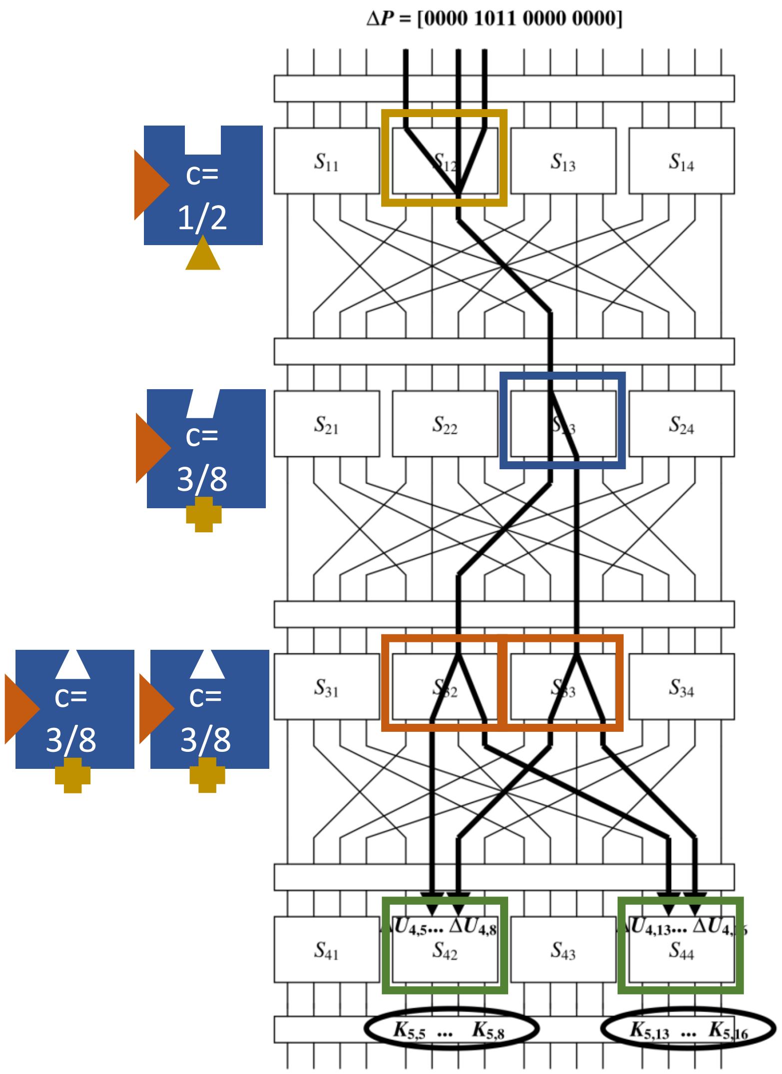
Bringing everything together

Goal is to build a differential relation over three rounds

- First find S-Box relation for middle round with high probability and minimal active wires
- E.g.: $\mathrm{DDT}(0100_b, 0110_b) = \mathrm{DDT}(4{,}6) = 6$
- If we use it at S-Box 3 (00**1**0) then next round:
  - Has 2 active S-Boxes (0110_b: 2 active output wires)
  - Both have active input wire 3 ⇒ 0010_b

- Round 3:
  - E.g. $\mathrm{DDT}(0010_b, 0101_b) = \mathrm{DDT}(2{,}5) = 6$
  - ⇒ rounds 2 and 3 done
  - Round 4 has 2 active S-Boxes

- First round:
  - Active S-Box 2 with output mask 0010_b
  - Find highest probability
  - Input mask is not important: no S-Boxes before
  - E.g. $\mathrm{DDT}(1011_b, 0010_b) = \mathrm{DDT}(11{,}2) = 8$



Bringing everything together

First round:

- $\Delta I_1 = \Delta P = [0000\ 1011\ 0000\ 0000]$
- S-Box: $\Delta X_{12} = [1011] \Rightarrow \Delta Y_{12}[0010]$ with probability $p_{12} = 1/2$
- π_P: $\Delta Y_1 = [0000\ 0010\ 0000\ 0000] \Rightarrow \Delta O_1 = [0000\ 0000\ 0100\ 0000]$

Second round:

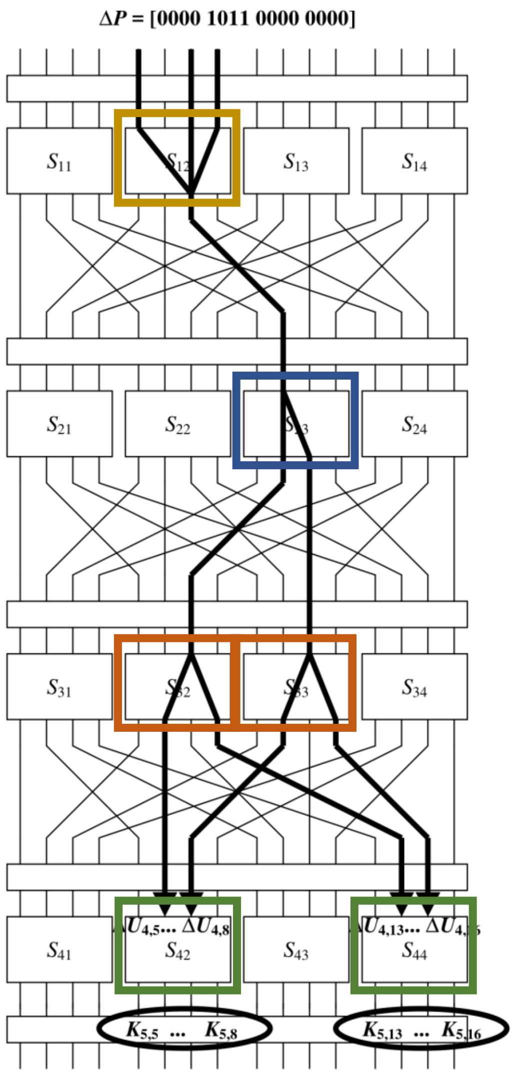
- $\Delta I_2 = \Delta O_1 = [0000\ 0000\ 0100\ 0000]$
- S-Box: $\Delta X_{23} = [0100] \Rightarrow \Delta Y_{23} = [0110]$ with probability $p_{23} = 3/8$
- π_P: $\Delta Y_2 = [0000\ 0000\ 0110\ 0000] \Rightarrow \Delta O_2 = [0000\ 0010\ 0010\ 0000]$

Third round:

- $\Delta I_3 = \Delta O_3 = [0000\ 0010\ 0010\ 0000]$
- S-Box: $\Delta X_{32} = [0010] \Rightarrow \Delta Y_{32} = [0101]$ with probability $p_{32} = 3/8$
- S-Box: $\Delta X_{33} = [0010] \Rightarrow \Delta Y_{33} = [0101]$ with probability $p_{33} = 3/8$
- π_P: $\Delta Y_3 = [0000\ 0101\ 0101\ 0000] \Rightarrow \Delta O_3 = [0000\ 0110\ 0110\ 0000]$

Connect all relations above:

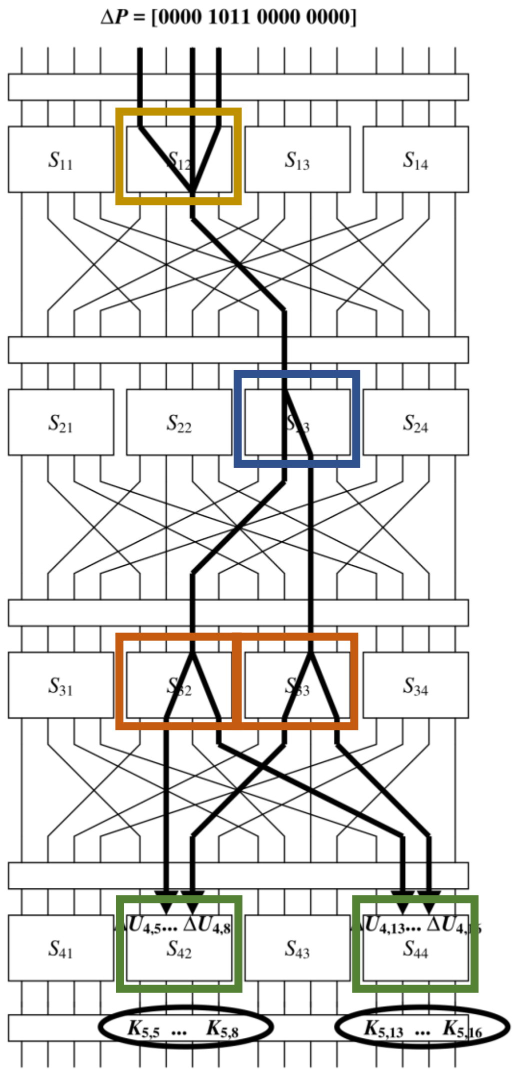
- Output difference of round i must match input difference of round $i + 1$
- $(\Delta P, \Delta O_3) = ([0000\ 1011\ 0000\ 0000], [0000\ 0110\ 0110\ 0000])$
- Probability (Piling-Up Lemma): $\geq \frac{1}{2}\left(\frac{3}{8}\right)^3 = \frac{27}{1024} \approx 0.026$



Key-recovery attack

$(\Delta P, \Delta O_3) = ([0000\ 1011\ 0000\ 0000], [0000\ 0110\ 0110\ 0000])$

Probability: $\mathrm{p_{diff}} \geq \frac{1}{2}\left(\frac{3}{8}\right)^3 = \frac{27}{1024} \approx 0.026$



Build distinguisher for 3 rounds (w/ 4 key additions)

- Over many (P_1, P_2, C_1, C_2)-tuples with $\mathrm{P}_1 \oplus P_2 = \Delta P$ measure probability of $C_1 \oplus C_2 = \Delta O_3$
- Is $\approx \frac{27}{1024} \approx 0.026 \Rightarrow$ is blockcipher oracle with 3 rounds
- Is $\approx 2^{-16} \approx 0.000015 \Rightarrow$ random oracle

Key-recovery attack idea:

1. Obtain many PPCC-tuples
2. Guess last round key => decrypt last round
   - Note how we only need to guess 8 key bits of K_5
3. Do distinguishing check
   - Outputs blockcipher oracle $\Rightarrow$ right key guess, stop
   - Outputs random oracle $\Rightarrow$ wrong key guess, try again with another guess

Key-recovery attack analysis

Count PPCC tuples that match relation: C

Case correct key-guess:

- Binomial distribution with n samples and $p = p_{diff}$
- $E[C] = n \cdot p_{diff}$

Case wrong key-guess:

- Decrypt, observe & compare only 8 bits of ΔO_3:
- Binomial distribution with n samples and $p = 2^{-8}$
- $E[C] = n/2^8$

However, there are $\approx 2^8$ wrong key-guesses

- Does the correct key-guess stand out among all of them?
- Approximate with Normal distribution N: mean $n/2^8$ and SD $\sqrt{n/2^8}$
- Then $\Pr[|N - mean| > x \cdot SD] \leq e^{-x^2/2}$ (see lecture notes)
- For $x = 4$, this probability is $\ll 2^{-8}$ $\Rightarrow$ expect all samples bounded by $4 \cdot SD$

How many samples do we need to have the correct key-guess stand out?

- $n \cdot p_{diff} > n/2^8 + 4 \cdot \sqrt{n/2^8}$
- For e.g. $n = 6/p_{diff}$: $\quad n \cdot p_{diff} = 6 > 4.67 \approx n/2^8 + 4\sqrt{n/2^8}$

Summary

- Linear cryptanalysis:
  - Break all round keys
  - Search for single high-bias linear relation
- Differential cryptanalysis
  - Input/output- difference relations with high probability
  - DDT: Difference Distribution Table for S-Box
  - Build differential relation for block cipher
    by combining internal differential relations with piling-up lemma
- Differential distinguisher
  - Blockcipher oracle vs Random oracle
  - Distinguish by measuring low probability 1/N vs high probability
- Key-recovery attack
  - Use distinguisher on R-1 rounds
  - Guess last key and distinguish: random oracle ⇒ wrong key guess
  - Number of P-C pairs: $O(1/p_{diff})$

Space of differential relations

- We've looked at 1 differential relation with high probability
  - Starts with plaintext difference ΔP
  - End with round 3 difference ΔO_3
  - Probability computed based on 1 trail $\Delta P \Rightarrow \Delta O_1 \Rightarrow \Delta O_2 \Rightarrow \Delta O_3$

What about other differential relations and trails?

- Relations with same plaintext difference and round 3 output difference:
  - Disjoint events: thus probabilities sum up!
  - A single high probability can be a good first approximation
- Relations with same plaintext difference and round 4 active S-Boxes:
  - Independent distinguishers can be used together to get higher confidence on correct key guess
  - $\Rightarrow$ need fewer PPCC tuples
- Relations with other plaintext differences
  - Cannot directly reuse tuples $\Rightarrow$ new samples, or recombine into new tuples
- Relations with other round 4 active S-Boxes:
  - Learn other key bits

Other differential attacks

Key-recovery: any efficient distinguisher works

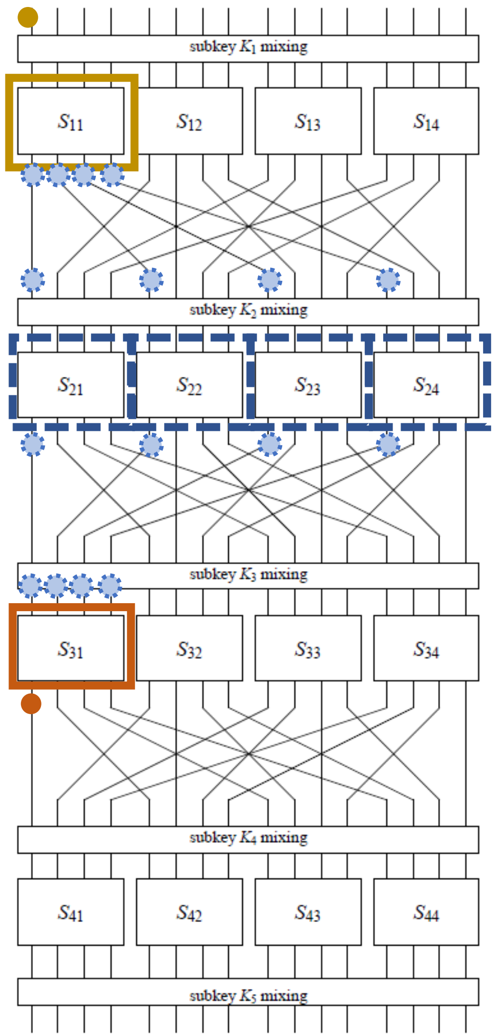
So any high probability relation that is easily checkable works

Three variant attacks based on differential cryptanalysis

1. Truncated differential cryptanalysis
   - Allow sets of differences for internal variables instead of one chosen difference
   - Potentially higher probabilities
2. Impossible differential cryptanalysis
   - Use a differential relation with probability 0
   - Have to prove no trail exists
3. Boomerang distinguishers
   - 2nd order differential: P_a, P'_a, P_b, P'_b with $\Delta P_a = \Delta P_b$
   - Analyze difference $\Delta X_a \oplus \Delta X_b$ between differences ΔX_a and ΔX_b

Impossible differential cryptanalysis

- Idea: find differential relations with probability 0 in total
- Since differential 'trails' probability add up for a relation, one needs to prove no differential trail exists with p>0
- E.g.:
  - $(\Delta P, \Delta O_3) = (1000\ 0 \ldots \ldots \ldots 0, 1000\ 0 \ldots \ldots \ldots 0)$
  - Note that ΔY_{11} & ΔX_{31} are unknown so unknown which round 2 S-Boxes are active
  - However, any active round 2 S-Box must use $DDT(1000_b, 1000_b) = DDT(8,8) = 0$
  - Hence, no $p > 0$ differential trail exists
- Similar for any $\Delta P = (**** \ 0000\ 0000\ 0000)$
- Similar for any ΔO_3 based on $\Delta Y_{31} = (****)$ and $\Delta Y_{32} = \Delta Y_{33} = \Delta Y_{34} = 0$



Impossible differential cryptanalysis

- Set of $(\Delta P, \Delta O_3)$:
  - $\Delta P = (1000\ 0000\ 0000\ 0000)$
  - $\Delta O_3 \in \mathcal{O} \coloneqq \{(a000\ b000\ c000\ d000)\}$
- Distinguisher:
  - Set of n $PPCC$-tuples with given ΔP
  - For each possible guess K_5:
    - Decrypt last round of C, C' of each tuple
    - If any $\Delta O_3 \in \mathcal{O}$ is observed $\Rightarrow$ wrong key guess
- Analysis:
  - Correct key guess: $\Delta O_3 \in \mathcal{O}$ never occurs
  - Wrong key guess:

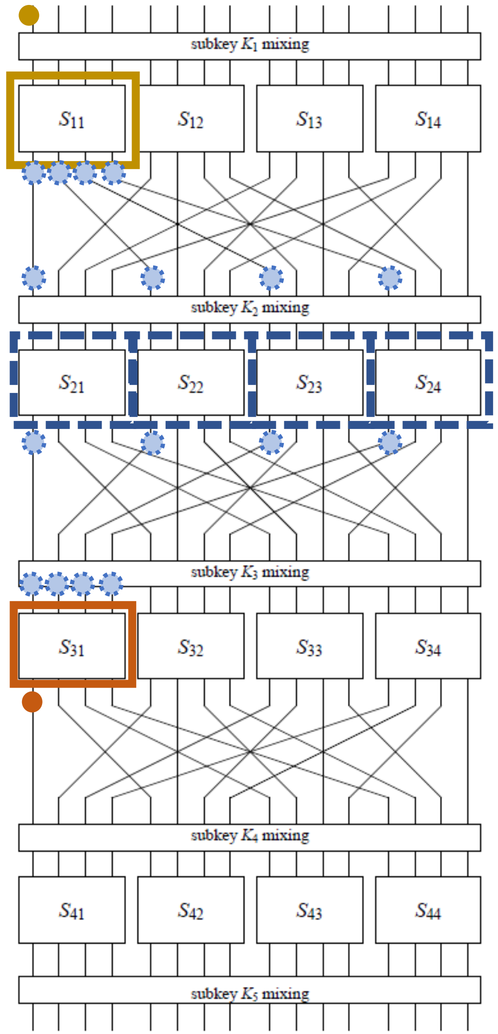
    Assume each $\Delta O_3 \in \mathcal{O}$ occurs with $p \approx n \cdot 2^{-16}$

    Observing any $\Delta O_3 \in \mathcal{O}$ occurs with $p \approx n \cdot 2^{-12}$

    $\Rightarrow n = O(2^{12})$ needed to filter wrong guesses

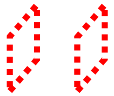
  - Improve using many $\Delta P \in (efgh\ 0000\ 0000\ 0000)$

    $\Rightarrow n = O(2^{8})$ needed

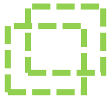


Boomerang distinguishers

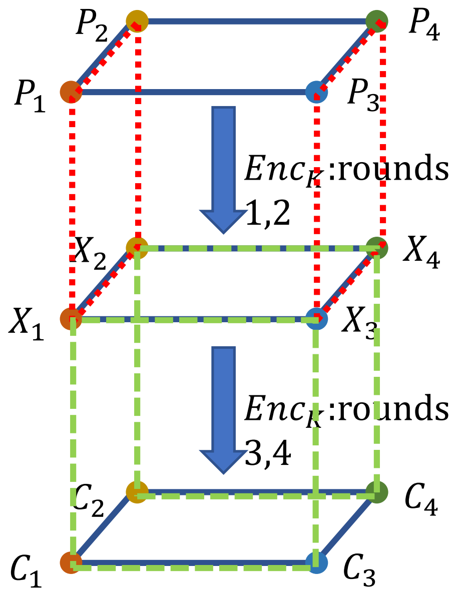
- Boomerang distinguishers are based on 2nd order differential cryptanalysis
- Involves 4 PC-pairs: $(P_1, C_1), (P_2, C_2), (P_3, C_3), (P_4, C_4)$
- These are studied in 2 combinations:
  - $(P_1, C_1)\&(P_2, C_2)$ and $(P_3, C_3)\&(P_4, C_4)$ with $P_1 \oplus P_2 = \Delta P$ and $P_3 \oplus P_4 = \Delta P$ for rounds 1 & 2



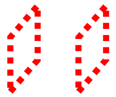
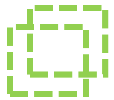
  - $(P_1, C_1)\&(P_3, C_3)$ and $(P_2, C_2)\&(P_4, C_4)$ with $C_1 \oplus C_3 = \Delta C$ and $C_2 \oplus C_4 = \Delta C$ for rounds 3 & 4

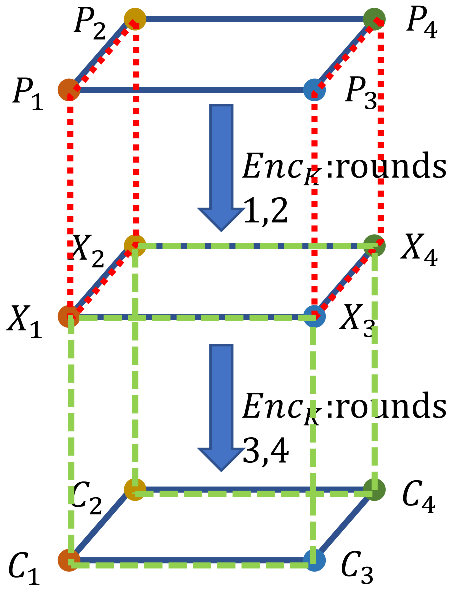


  - Note how ΔC is used orthogonal to ΔP



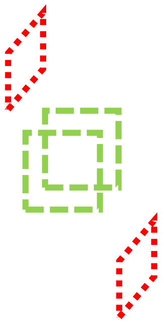
Boomerang distinguishers

- Find two high probability differential relations
  - Rounds 1&2: $\Delta P \rightarrow \Delta O_2$ with probability $p_1 \coloneqq p_{(\Delta P, \Delta O_2)}$
  - Rounds 3&4: $\Delta I_3 \rightarrow \Delta C$ with probability $p_2 \coloneqq p_{(\Delta I_3, \Delta C)}$
  - (X, O_2, I_3 describe the same variable, but different names are used to keep the 2 relations apart)
- Two combinations:
  - $(P_1, C_1)\&(P_2, C_2)$ and $(P_3, C_3)\&(P_4, C_4)$ 
    - with $P_1 \oplus P_2 = \Delta P$ and $P_3 \oplus P_4 = \Delta P$
    - then $X_1 \oplus X_2 = \Delta O_2$ with probability p_1
    - and $X_3 \oplus X_4 = \Delta O_2$ with probability p_1
  - $(P_1, C_1)\&(P_3, C_3)$ and $(P_2, C_2)\&(P_4, C_4)$ 
    - with $C_1 \oplus C_3 = \Delta C$ and $C_2 \oplus C_4 = \Delta C$
    - then $X_1 \oplus X_3 = \Delta I_3$ with probability p_2
    - and $X_2 \oplus X_4 = \Delta I_3$ with probability p_2

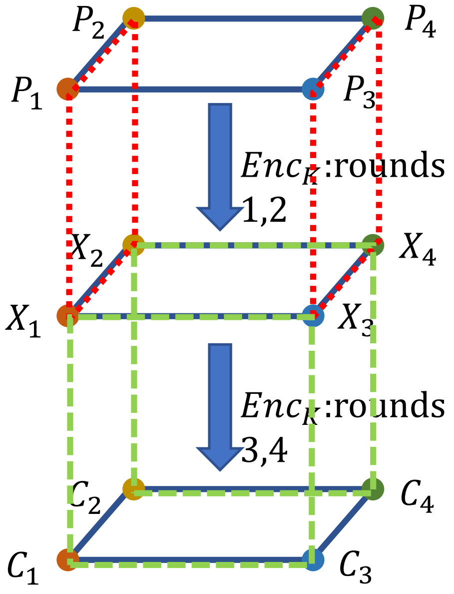


Boomerang distinguishers

- Constructing a boomerang tuple
  1. Pick $P_1 \leftarrow \{0,1\}^{16}$, set $P_2 \coloneqq P_1 \oplus \Delta P$
  2. Ask to encrypt $C_1 \coloneqq Enc(P_1), C_2 \coloneqq Enc(P_2)$
  3. Set $C_3 \coloneqq C_1 \oplus \Delta C, C_4 \coloneqq C_2 \oplus \Delta C$
  4. Ask to decrypt $P_3 \coloneqq Dec(C_3), P_4 \coloneqq Dec(C_4)$
  5. Repeat until $P_3 \oplus P_4 = \Delta P$

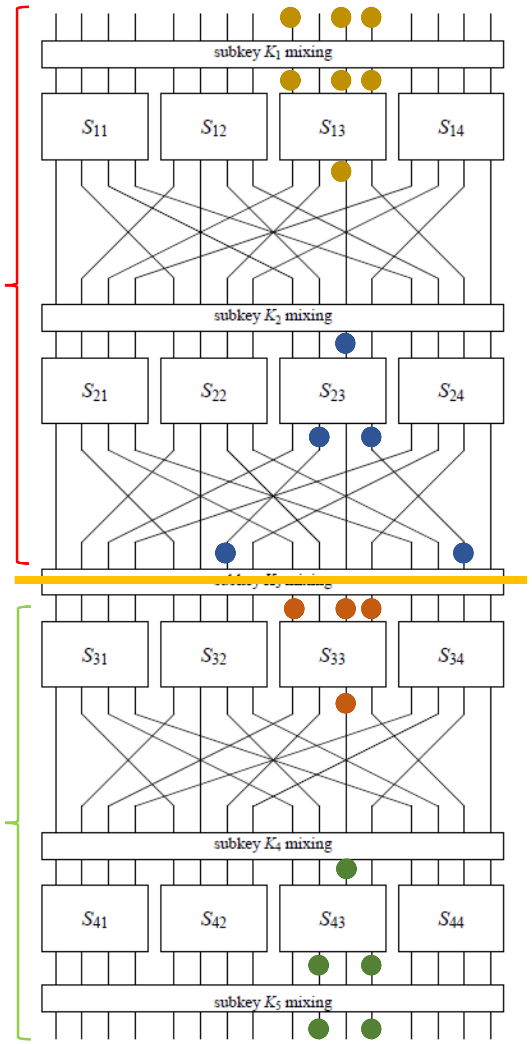


- Success probability (first approximation):
  - $X_1 \oplus X_2 = \Delta O_2$ with probability p_1
  - $X_1 \oplus X_3 = \Delta I_3$ with probability p_2
  - $X_2 \oplus X_4 = \Delta I_3$ with probability p_2
  - $\Rightarrow X_3 \oplus X_4 = \Delta O_2$ with probability 1
  - $\Rightarrow P_3 \oplus P_4 = \Delta P$ with probability p_1
  - Total probability $p_1^2 \cdot p_2^2$
- Similarly any other choice for ΔO_2 & ΔI_3
  - These are all disjoint events $\Rightarrow$ probabilities add up:
  - $p_{success} = \sum_{\Delta O_2} \sum_{\Delta I_3} p^2_{(\Delta P, \Delta O_2)} \cdot p^2_{(\Delta I_3, \Delta C)}$
- Success probability random oracle:
  - $P_3 \oplus P_4$ is random
  - $\Pr[P_3 \oplus P_4 = \Delta P] = 2^{-N}$



Example Boomerang

- Use the same 2-round differential
  - For round 1&2
  - For round 3&4 (but round 4 does not use π_P)
- 2-round differential:
  - $\Delta I_1, \Delta I_3 = (0000\ 0000\ 1011\ 0000)$
  - S-Box S_{13}, S_{33} active: $DDT(1011_b, 0010_b) = DDT(11{,}2) = 8$
    $\Rightarrow$ probability $1/2$
  - $\Delta I_2, \Delta I_4 = (0000\ 0000\ 0010\ 0000)$
  - S-Box S_{23}, S_{43} active: $DDT(0010_b, 0101_b) = DDT(2{,}5) = 6$
    $\Rightarrow$ probability $3/8$
  - $\Delta O_2 = (0000\ 0010\ 0000\ 0010)$ for rounds 1&2
    or $\Delta C = (0000\ 0000\ 0101\ 0000)$ for rounds 3&4
  - Probability: $3/16$
- Boomerang prob $\geq (3/16)^4 \approx 0.001236 \approx 1/809$
- Measured boomerang prob: ≈ 0.01 !!!



Summary

- Differential cryptanalysis variants
  - Any efficient distinguisher is an attack
  - So any easily checkable relation with high probability works
- Truncated differential cryptanalysis
  - Use sets of differences instead of a chosen difference
  - Larger differential probabilities:
    add probabilities of several output differences
- Impossible differential cryptanalysis
  - Use relations that have proven probability 0
  - Distinguisher:
    - When relation is observed ⇒ random oracle / wrong key guess
- Boomerang distinguishers
  - 2^{nd} order differential cryptanalysis: 4 encryptions
  - Find tuple satisfying ΔP for 1-2 & 3-4 and ΔC for 1-3 & 2-4
  - Short & open-ended trails: lots & lots of trails
  - ⇒ very high probability