# Selected Areas in Cryptology
# Cryptanalysis
# Week 2

Marc Stevens

stevens@cwi.nl

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

# Block cipher design

Attacks against the internal structure of a blockcipher

$$E_K: \{0,1\}^n \rightarrow \{0,1\}^n, \quad K \in \{0,1\}^k$$

Blockcipher consists of $R$ rounds of a small keyed round function $E_K^r$

- Small: few operations
- Keyed: involves key material
- 'Confusion': complex operations ⇒ very complex final relations
- 'Diffusion': mix state ⇒ each in-/output bit depends on each out-/input bit

# Block cipher design: SPN framework

Focus on SPN: Substitution Permutation Network

- Substitution: complex permutation "S-BOX" on e.g. 8 bits applied on all 8-bit parts
- Permutation: mixing of entire state ($\mathbb{F}_2$ - linear)
- Keyed: add round key ($\mathbb{F}_2$ - linear) (derived from main key)

AES: state $n = 128$ bits, key $k = 128,192,256$ bits, S-box: 8 bits

Toy-Cipher to demonstrate structural attack techniques

- State $n = 16$ bits, 4 rounds
- 5 round keys $K_1, \ldots, K_5$ of 16 bits
- Small enough to do attacks in practice (if you wanted)

# Toy-Cipher

Key-addition:

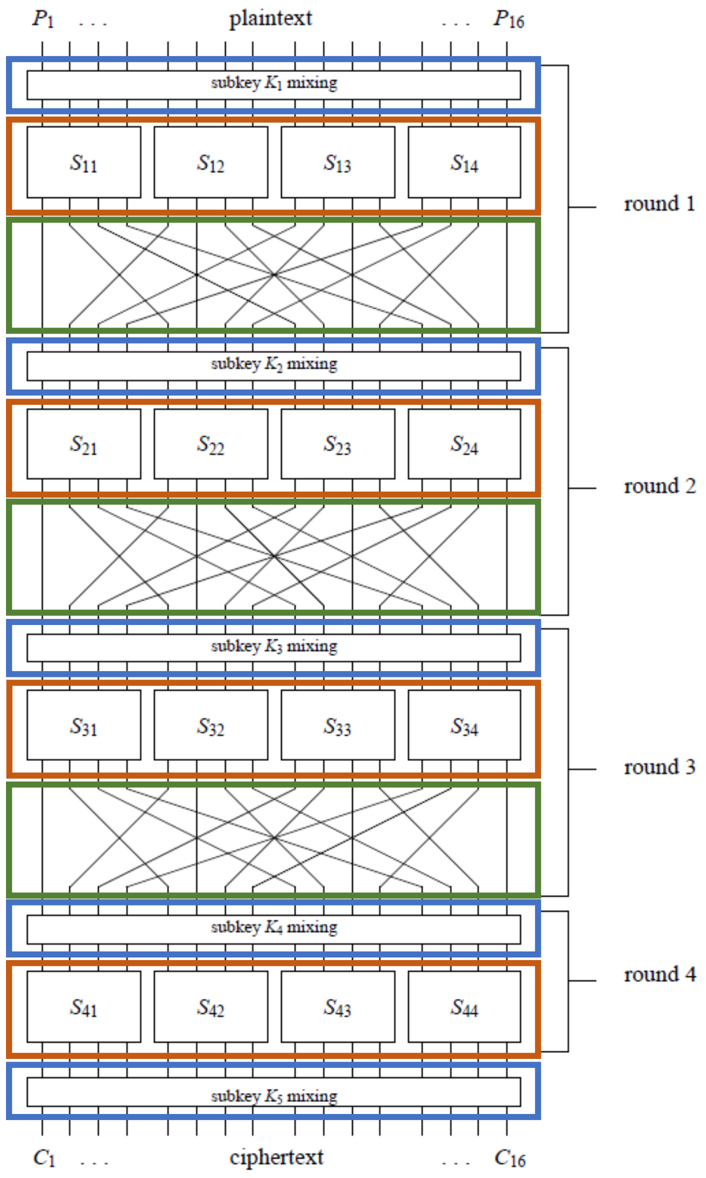
- XOR round key $K_i$
- Final key-addition at end with $K_5$

Substitution: 4-bit S-box

- $\pi_S : \{0,1\}^4 \to \{0,1\}^4$ (see lecture notes)
- called 4 times per round to alter all 16 bits

Permutation of 16 bits:

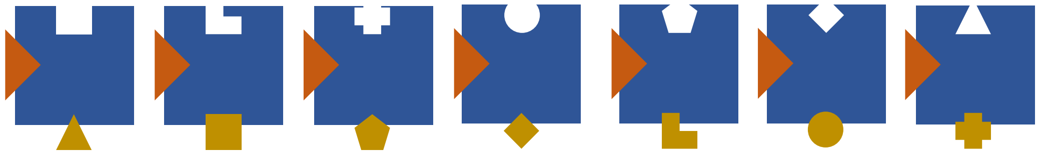
- $\pi_P : \{1, \ldots, 16\} \to \{1, \ldots, 16\}$ (see lecture notes)
- Skipped in last round, as it can be removed anyway (swap Perm and AddKey with $K_5'[i] = K_5[\pi_P(i)]$)



# Structural attacks: linear & differential cryptanalysis

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

2. Obtain a family of round attack building blocks

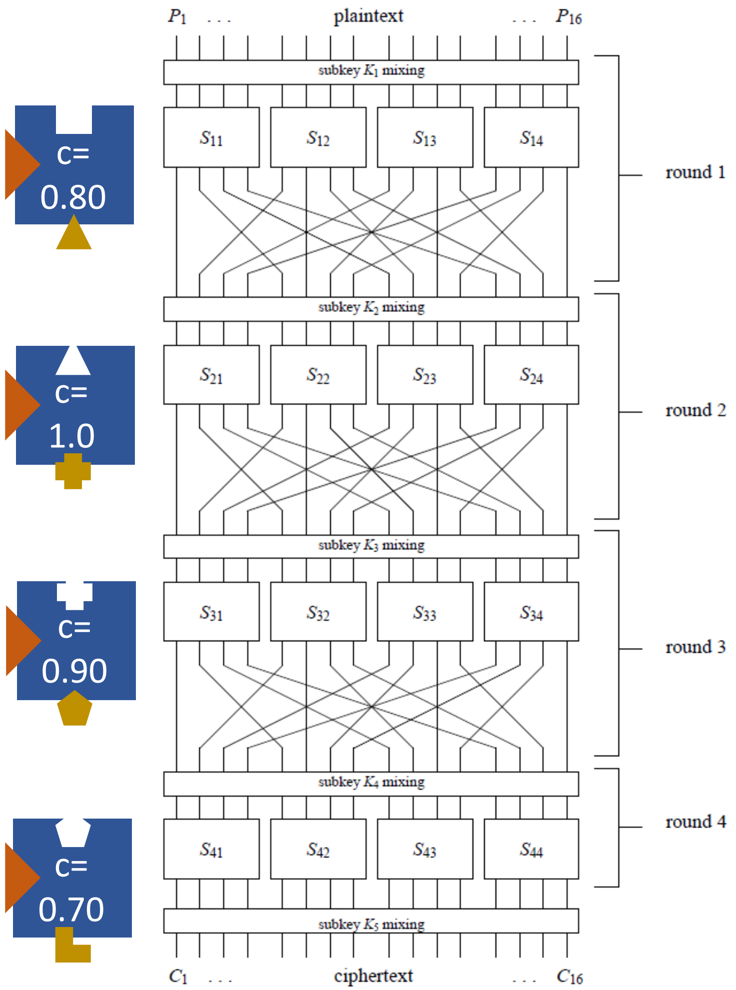


3. Combine to attack on full blockcipher

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



# Linear Cryptanalysis

Linear approximate each round:

- probabilistic $\mathbb{F}_2$- linear input-output relation for $C = E_K^r(P)$

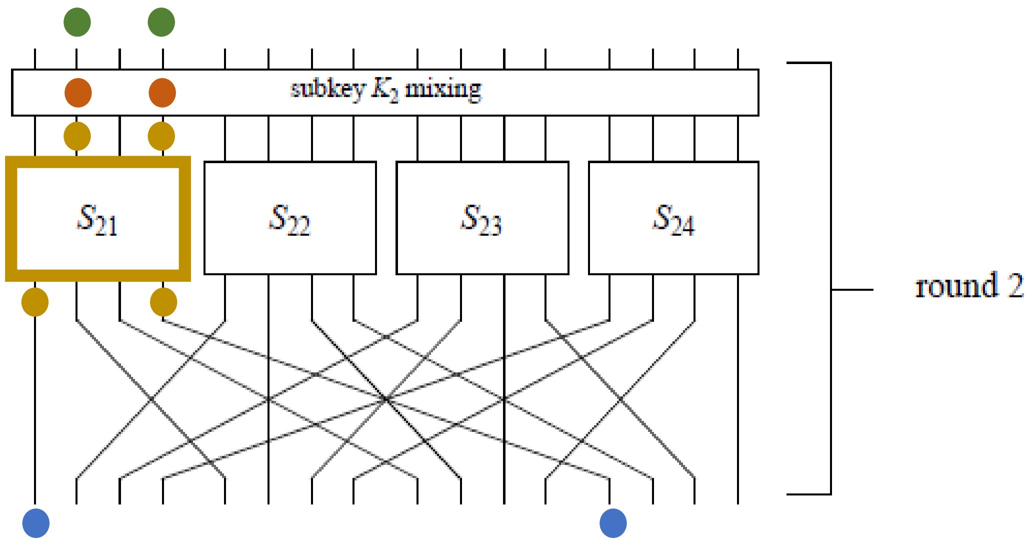
$$\sum_{i \in I} P[i] \quad \oplus \quad \sum_{j \in J} C[j] \quad \oplus \sum_{l \in \mathrm{L}} K[l] = c$$

  - Involves selected input bits $P[i]$, output bits $C[j]$, key bits $K[l]$, and a constant $c$
- E.g.: $P[2] \oplus P[4] \oplus C[1] \oplus C[7] \oplus K_1[2] \oplus K_1[4] \cdots \oplus K_5[7] = 1$

- $\mathbb{F}_2$: constant either $c = 0$ or $c = 1$
- Probabilistic: $P, K, C$ are seen as random variables where $C = E_K^r(P)$:
  - Ideal secure situation: $p = 0.5$ exactly for any such relation
  - $\Rightarrow$ approximation doesn't give any information on key bits given $P, C$
  - Actual case $p = 0.5 + \epsilon$, where $\epsilon \in [-.5, +.5]$ is the bias
  - $\Rightarrow$ larger bias means larger probability of correct prediction
- Search for round relations with large (absolute) bias!

# Linear Cryptanalysis

- Notation round variables:
  - Round input: $P[1], \dots, P[16]$
  - Round key: $K[1], \dots, K[16]$
  - S-Box input: $X[1], \dots, X[16]$
  - S-Box output: $Y[1], \dots, Y[16]$
  - Round output: $C[1], \dots, C[16]$



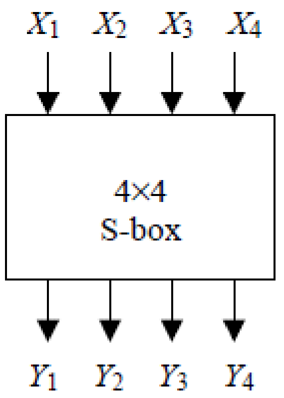

- Choose input bits: $P[2], P[4]$
- Key addition: involves key bits $K[2], K[4]$
- Substitution:
  - $S_{22}, S_{23}, S_{24}$: Inactive S-Boxes: no input bits selected
  - 1 active S-Box: $S_{21}$
    - Inputs: $X[2] = P[2] \oplus K[2]$ and $X[4] = P[4] \oplus K[4]$
    - Choose outputs: $Y[1], Y[4]$
- Permutation: $C[1] = Y[1], C[13] = Y[4]$

Note:
Only Substitution is NOT $\mathbb{F}_2$-linear
and has probabilistic relation!
S-Box in/out also defines round in/out!

- Relation: $Rel\colon\ P[2] \oplus P[4] \oplus C[1] \oplus C[13] = 0 \oplus K[2] \oplus K[4]$
- Probability: $\Pr_{P}[Rel] = \Pr_{X}[X[2] \oplus X[4] \oplus Y[1] \oplus Y[4] = 0 \mid Y = \pi_S(X)]$

# LAT: Linear Approximation Table

- S-Box relation directly defines round relation and probability!
- Thus analyze all linear relations for S-Box $\pi_S$ of the form:
  - $\Pr_X[X[2] \oplus X[4] \oplus Y[1] \oplus Y[4] = 0 \mid Y = \pi_S(X)]$




- S-Box is permutation on $\{0,1\}^4$
  - 16 possible selections of sums $\sum_{i\in \mathrm{I}} X[i],\ I \subseteq \{1,2,3,4\}$
  - 16 possible selections of sums $\sum_{j\in \mathrm{I}} Y[j],\ J \subseteq \{1,2,3,4\}$
  - Represent $I//J$ as 4-bit mask / integer value:
    $\{1\} \to 1000_b = 8, \qquad \{3,4\} \to 0011_b = 3$

- Linear Approximation Table (LAT):
  - 16 x 16 table
  - Row $I \in \{0, \dots, 15\}$, Column $J \in \{0, \dots, 15\}$ contains:
  - $LAT(I,J) \coloneqq \#\{X \in \{0,1\}^4, Y = \pi_S(X) \mid \sum X[i] \oplus \sum Y[j] = 0\} - 8$
  - Bias $\epsilon_{I,J} = \Pr[\sum X[i] \oplus \sum Y[j] = 0] - 0.5 = LAT(I,J)/16$
  - Important tool!
    - Easily precomputed, independent of keys
    - Convenient look-up for large biases to construct large bias relations

# LAT: Linear Approximation Table

- Compute entry
  1. Write all values for $X$ with corresponding $Y$-values
  2. Compute $X$-sum
  3. Compute $Y$-sum
  4. Count total matching values ($A \oplus B = 0 \;\Leftrightarrow\; A = B$)
  5. Subtract 8

- $X[2] \oplus \mathrm{X}[3] \oplus Y[1] \oplus Y[3] \oplus Y[4]$
  - 12 matching
  - $\Pr[\sum = 0] = \frac{12}{16}, \quad \epsilon = \frac{12}{16} - \frac{1}{2} = \frac{12-8}{16} = \frac{4}{16}$
  - $x = 0110_b = 6$
  - $y = 1011_b = 11$
  - $\Rightarrow LAT(6{,}11) = 12 - 8 = 4$
    $= 16\,\epsilon$

| $X_1X_2X_3X_4$ | $Y_1Y_2Y_3Y_4$ | $X_2 + X_3$ | $Y_1 + Y_3 + Y_4$ |
|---|---|---|---|
| 0000 | 1110 | 0 | 0 |
| 0001 | 0100 | 0 | 0 |
| 0010 | 1101 | 1 | 0 |
| 0011 | 0001 | 1 | 1 |
| 0100 | 0010 | 1 | 1 |
| 0101 | 1111 | 1 | 1 |
| 0110 | 1011 | 0 | 1 |
| 0111 | 1000 | 0 | 1 |
| 1000 | 0011 | 0 | 0 |
| 1001 | 1010 | 0 | 0 |
| 1010 | 0110 | 1 | 1 |
| 1011 | 1100 | 1 | 1 |
| 1100 | 0101 | 1 | 1 |
| 1101 | 1001 | 1 | 0 |
| 1110 | 0000 | 0 | 0 |
| 1111 | 0111 | 0 | 0 |

# LAT: Linear Approximation Table

LAT properties:

- Compute with sage (see lecture notes)
- $LAT(0,0) = 16 - 8 = 8$
- $LAT(x, 0) = LAT(0, x) = 8 - 8 = 0,\ x > 0$

- Every entry is even
- Sum of every row/column = $\pm 8$

Toy-Cipher LAT

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

# Piling-Up Lemma

How to combine two linear relations ?

- Let $X_1, X_2$ be two independent binary random variables
  (think of them as the output of the sum of $X$ & $Y$ bits)
- Let $p_1 \coloneqq \Pr[X_1 = 0]\,, p_2 \coloneqq \Pr[X_2 = 0]$
- Then: $\Pr[X_1 \oplus X_2 = 0] = \Pr[X_1 = 0 \wedge X_2 = 0] + \Pr[X_1 = 1 \wedge X_2 = 1]$
  $= p_1 \cdot p_2 + (1 - p_1) \cdot (1 - p_2)$
- Now consider the biases:
  $\epsilon_1 \coloneqq p_1 - 0.5, \quad \epsilon_2 \coloneqq p_2 - 0.5, \quad \epsilon_{1,2} \coloneqq \Pr[X_1 \oplus X_2 = 0] - 0.5$
- Then: $\epsilon_{1,2} = (0.5 + \epsilon_1)(0.5 + \epsilon_2) + (0.5 - \epsilon_1)(0.5 - \epsilon_2) - 0.5$
  $= (0.25 + 0.5(\epsilon_1 + \epsilon_2) + \epsilon_1\epsilon_2) + (0.25 - 0.5(\epsilon_1 + \epsilon_2) + \epsilon_1\epsilon_2) - 0.5$
  $= 2\epsilon_1\epsilon_2$

Piling-Up Lemma:

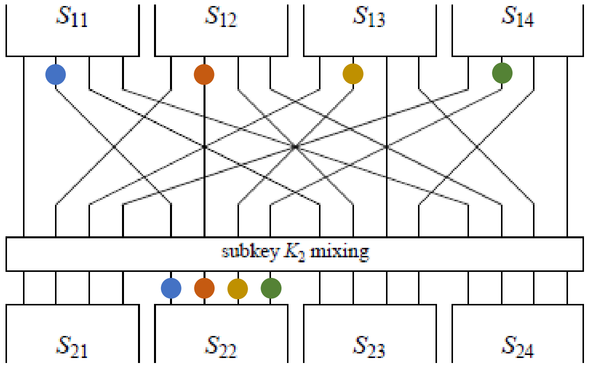
For $X_1, \ldots, X_N$ independent binary variables with biases $\epsilon_i$:

Their sum $X_{1,\ldots,N} = X_1 \oplus \cdots \oplus X_N$ has bias: $\epsilon_{1,\ldots,N} = 2^{N-1} \prod_{i=1}^{N} \epsilon_i$

# Bringing everything together

- LAT to find those high bias S-Box relations
- Inactive S-Boxes don't affect bias, as:
  - $LAT(0,0) = 8 \Rightarrow \epsilon_1 = \frac{8}{16} = \frac{1}{2}$
  - Piling-Up Lemma: $\epsilon_{1,2} = 2\epsilon_1\epsilon_2 = \epsilon_2$
- Only active S-Boxes matter ⇒ minimize active S-boxes

- Make use of $\pi_P$ properties
  - $i$-th output bit active 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 of S-Box $S_{2j}$ ⇒ S-Box $S_{1i}$ active in previous round

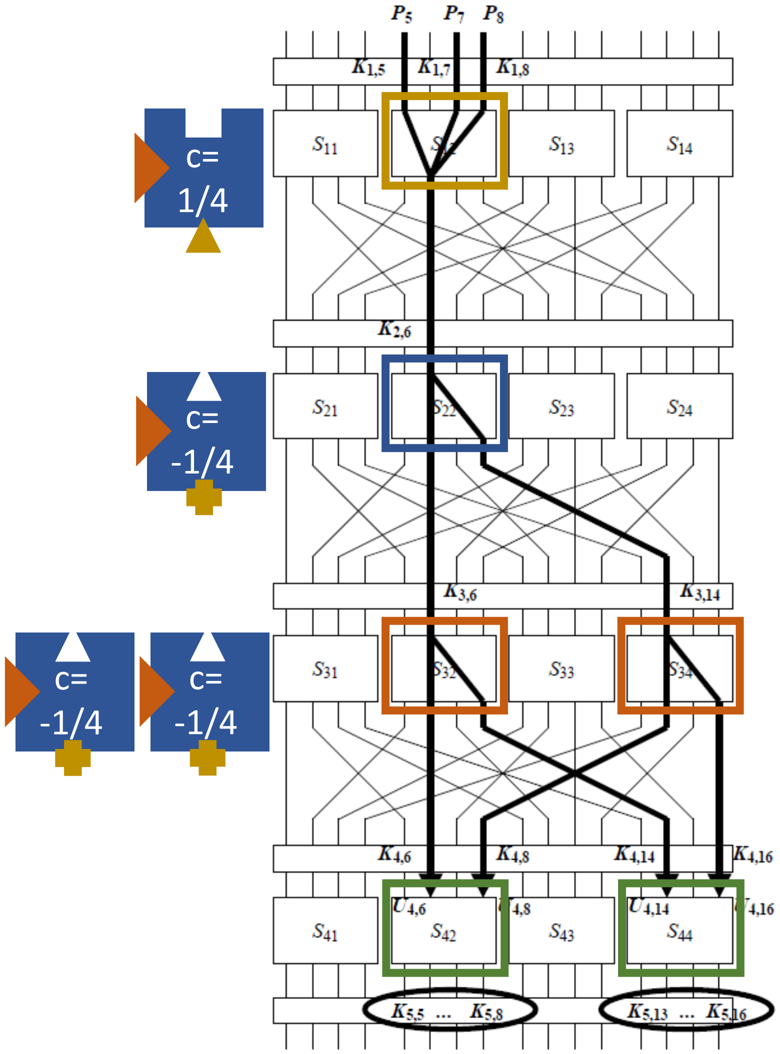



- If multiple active S-boxes in one round then try to have active input bits on same S-box bit position (and same for output bits)

# Bringing everything together

Goal is to build a linear approximation over three rounds

- First find S-Box relation for middle round with high bias and minimal active wires
  - The number of active wires equals the number of active S-Boxes in round 1 and 3 together
- E.g.: $\mathrm{LAT}(0100_b, 0101_b) = \mathrm{LAT}(4,5) = -4$
- If we use it at S-Box 2 (0**1**00) then next round:
  - Has 2 active S-Boxes ($0101_b$: 2 active output wires)
  - Both have active input wire 2 $\Rightarrow 0100_b$
- So can use same high bias relation again
  - $\Rightarrow$ rounds 2 and 3 done
  - Round 4 has 2 active S-Boxes
- First round:
  - Active S-Box 2 with output mask $0100_b$
  - Find highest bias
  - Input mask is not important: no S-Boxes before
  - E.g. $\mathrm{LAT}(1011_b, 0100_b) = \mathrm{LAT}(11,4) = 4$



# Bringing everything together

First round:

- $X_{12,1} \oplus X_{12,3} \oplus X_{12,4} = P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8}$
- $X_{12,1} \oplus X_{12,3} \oplus X_{12,4} \oplus Y_{12,2} = 0$ with bias $\epsilon_{12} = 4/16$

Second round:

- $X_{22,2} = Y_{12,2} \oplus K_{2,6}$
- $X_{22,2} \oplus Y_{22,2} \oplus Y_{22,4} = 0$ with bias $\epsilon_{22} = -4/16$

Third round:

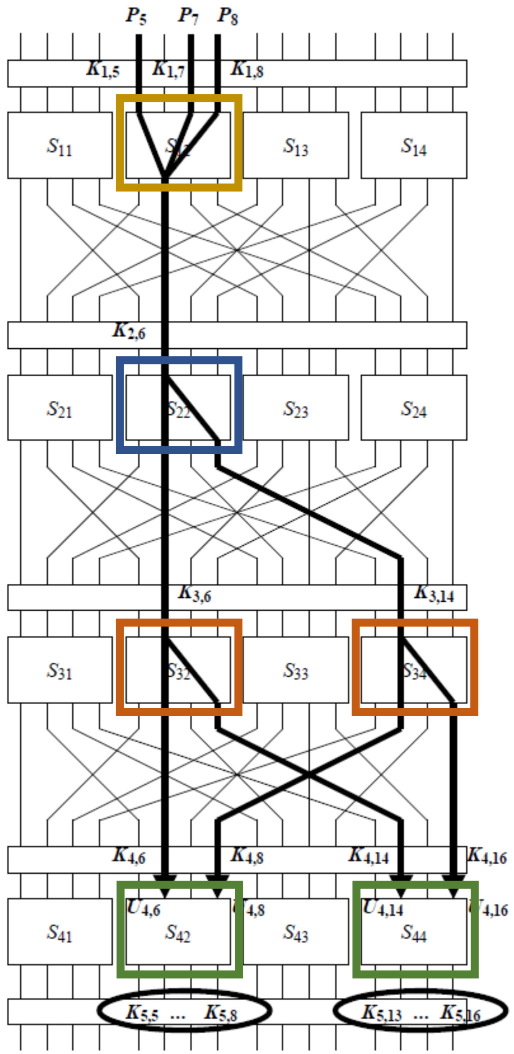
- $X_{32,2} = Y_{22,2} \oplus K_{3,6}$, $X_{34,2} = Y_{22,4} \oplus K_{3,14}$
- $X_{32,2} \oplus Y_{32,2} \oplus Y_{32,4} = 0$ with bias $\epsilon_{32} = -4/16$
- $X_{34,2} \oplus Y_{34,2} \oplus Y_{34,4} = 0$ with bias $\epsilon_{34} = -4/16$

Partial fourth round:

- $X_{42,2} \oplus X_{42,4} = Y_{32,2} \oplus Y_{34,2} \oplus K_{4,6} \oplus K_{4,8}$
- $X_{44,2} \oplus X_{44,4} = Y_{32,4} \oplus Y_{34,4} \oplus K_{4,14} \oplus K_{4,16}$

Sum all relations above (move only key bits on RSH):

- $P_5 \oplus P_7 \oplus P_8 \oplus X_{42,2} \oplus X_{42,4} \oplus X_{44,2} \oplus X_{44,4} =$
  $K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,1}$
- Note how all internal variables occur exactly twice & cancel
- Bias (Piling-Up Lemma): $2^3 \left(\frac{1}{4}\right)\left(-\frac{1}{4}\right)^3 = -\frac{1}{32}$



# Key-recovery attack

$P_5 \oplus P_7 \oplus P_8 \oplus X_{42,2} \oplus X_{42,4} \oplus X_{44,2} \oplus X_{44,4} =$
$K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$

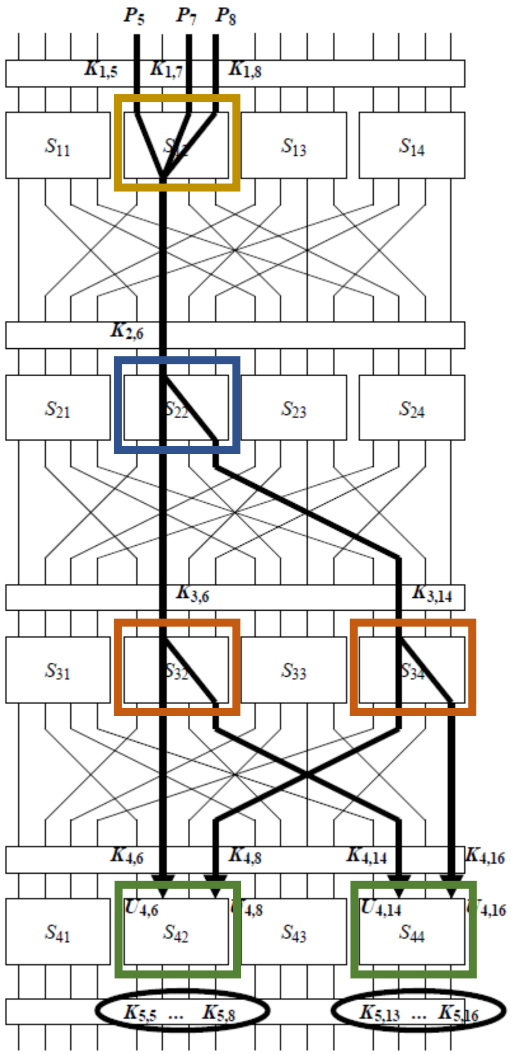
With bias: $2^3 \left(\frac{1}{4}\right)\left(-\frac{1}{4}\right)^3 = -\frac{1}{32}$

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



# Key-recovery attack analysis

Count P-C pairs that match relation: $C$

Case correct key-guess:

- Binomial distribution with $n$ samples and $p = 0.5 + \epsilon$
- $E[C] = n/2 + n \cdot \epsilon$

Case wrong key-guess:

- Binomial distribution with $n$ samples and $p = 0.5$
- $E[C] = n/2$

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 $n/2$ and SD $\sqrt{n/4}$
- 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?

- $n \cdot \epsilon > 4 \cdot SD \quad \Rightarrow \quad n \cdot \epsilon > 4\sqrt{n/4} \quad \Rightarrow \quad n > 4 \cdot \epsilon^{-2}$

# Wrap-up

- Block-cipher design:
  - Substitution: S-Box
  - Permutation: linear
  - Key-addition: linear
- Linear cryptanalysis
  - Input/output- linear relations with probability bias
  - LAT: Linear Approximation Table for S-Box
  - Build linear relation for block cipher
    by combining S-Box linear relations with piling-up lemma
- Linear distinguisher
  - Blockcipher oracle vs Random oracle
  - Distinguish by measuring non-zero bias vs zero bias
  - Based on good linear relation for block cipher
- Key-recovery attack
  - Use linear distinguisher on R-1 rounds
  - Guess last key and distinguish: random oracle $\Rightarrow$ wrong key guess
  - Number of P-C pairs: $O(\epsilon^{-2})$