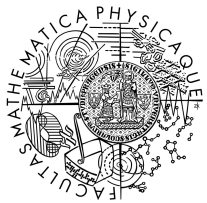
# Examples of Boolean Functions in Ciphers

Jakub Skalický

Department of Algebra
Charles University in Prague

Spring School of Algebra
28 March 2010



# Outline

- Data Encryption Standard:
  - S-Boxes as Boolean functions and their properties
  - differential attack against 3-round DES
- Advanced Encryption Standard
- Impossible differential attack

# Outline

- Data Encryption Standard:
  - S-Boxes as Boolean functions and their properties
  - differential attack against 3-round DES
- Advanced Encryption Standard
- Impossible differential attack

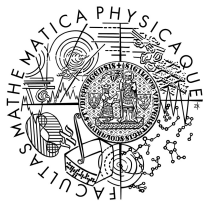
# Outline

- Data Encryption Standard:
  - S-Boxes as Boolean functions and their properties
  - differential attack against 3-round DES
- Advanced Encryption Standard
- Impossible differential attack

# Criteria studied

During our study of S-Boxes, we will look at:

- nonlinear order
- distance from the set of affine functions
- balancedness
- propagation criterion



# Criteria studied

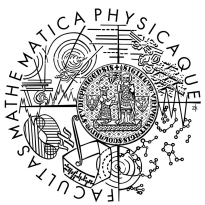
During our study of S-Boxes, we will look at:

- nonlinear order
- distance from the set of affine functions
- balancedness
- propagation criterion

# Criteria studied

During our study of S-Boxes, we will look at:

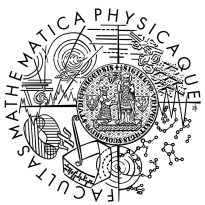
- nonlinear order
- distance from the set of affine functions
- balancedness
- propagation criterion



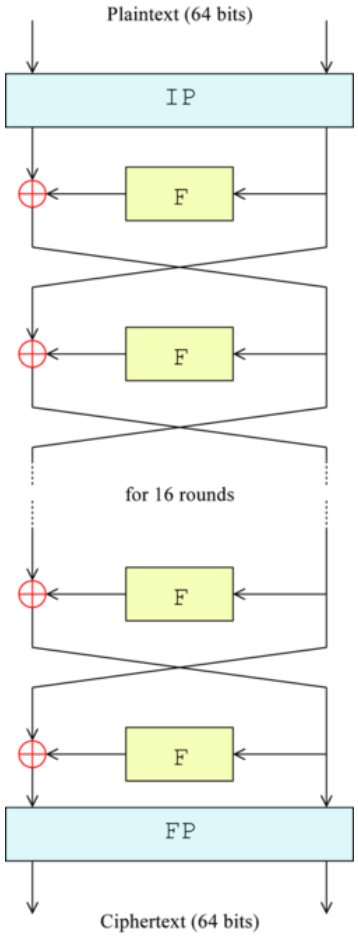
# Criteria studied

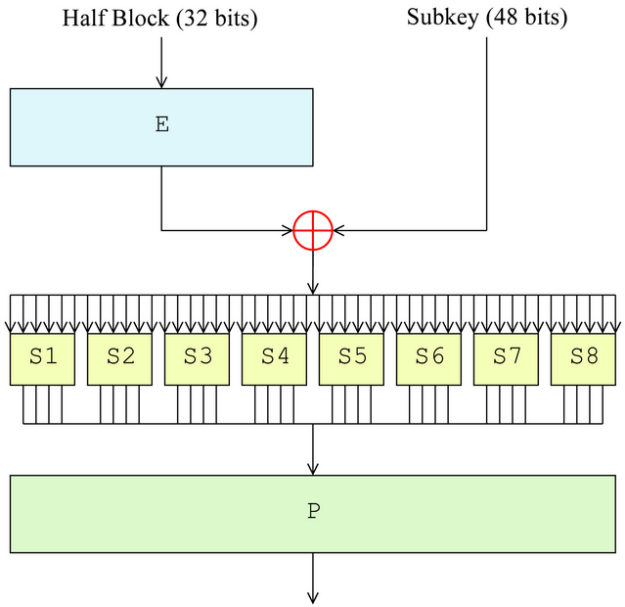
During our study of S-Boxes, we will look at:

- nonlinear order
- distance from the set of affine functions
- balancedness
- propagation criterion



# DES

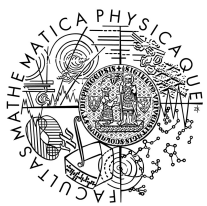

Plaintext (64 bits)
IP
F
F
for 16 rounds
F
F
FP
Ciphertext (64 bits)


Half Block (32 bits)
Subkey (48 bits)
E
S1
S2
S3
S4
S5
S6
S7
S8
P

# DES S-Box

- DES S-Box: random boolean function $\mathbb{Z}_2^6 \to \mathbb{Z}_2^4$ represented by a look-up table
- DES comprises eight different S-Boxes with different properties
- S-Boxes provide non-linearity in DES

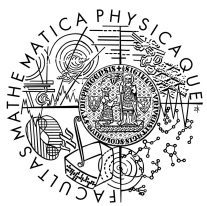
| $S_5$ | | Middle 4 bits of input | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| **Outer bits** | **00** | 0010 | 1100 | 0100 | 0001 | 0111 | 1010 | 1011 | 0110 | 1000 | 0101 | 0011 | 1111 | 1101 | 0000 | 1110 | 1001 |
| | **01** | 1110 | 1011 | 0010 | 1100 | 0100 | 0111 | 1101 | 0001 | 0101 | 0000 | 1111 | 1010 | 0011 | 1001 | 1000 | 0110 |
| | **10** | 0100 | 0010 | 0001 | 1011 | 1010 | 1101 | 0111 | 1000 | 1111 | 1001 | 1100 | 0101 | 0110 | 0011 | 0000 | 1110 |
| | **11** | 1011 | 1000 | 1100 | 0111 | 0001 | 1110 | 0010 | 1101 | 0110 | 1111 | 0000 | 1001 | 1010 | 0100 | 0101 | 0011 |



# DES S-Box

- DES S-Box: random boolean function $\mathbb{Z}_2^6 \to \mathbb{Z}_2^4$ represented by a look-up table
- DES comprises eight different S-Boxes with different properties
- S-Boxes provide non-linearity in DES

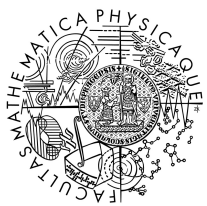
| $S_5$ | | Middle 4 bits of input | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| **Outer bits** | **00** | 0010 | 1100 | 0100 | 0001 | 0111 | 1010 | 1011 | 0110 | 1000 | 0101 | 0011 | 1111 | 1101 | 0000 | 1110 | 1001 |
| | **01** | 1110 | 1011 | 0010 | 1100 | 0100 | 0111 | 1101 | 0001 | 0101 | 0000 | 1111 | 1010 | 0011 | 1001 | 1000 | 0110 |
| | **10** | 0100 | 0010 | 0001 | 1011 | 1010 | 1101 | 0111 | 1000 | 1111 | 1001 | 1100 | 0101 | 0110 | 0011 | 0000 | 1110 |
| | **11** | 1011 | 1000 | 1100 | 0111 | 0001 | 1110 | 0010 | 1101 | 0110 | 1111 | 0000 | 1001 | 1010 | 0100 | 0101 | 0011 |



# DES S-Box

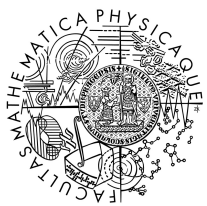
- DES S-Box: random boolean function $\mathbb{Z}_2^6 \to \mathbb{Z}_2^4$ represented by a look-up table
- DES comprises eight different S-Boxes with different properties
- S-Boxes provide non-linearity in DES

| $S_5$ | | Middle 4 bits of input | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| **Outer bits** | **00** | 0010 | 1100 | 0100 | 0001 | 0111 | 1010 | 1011 | 0110 | 1000 | 0101 | 0011 | 1111 | 1101 | 0000 | 1110 | 1001 |
| | **01** | 1110 | 1011 | 0010 | 1100 | 0100 | 0111 | 1101 | 0001 | 0101 | 0000 | 1111 | 1010 | 0011 | 1001 | 1000 | 0110 |
| | **10** | 0100 | 0010 | 0001 | 1011 | 1010 | 1101 | 0111 | 1000 | 1111 | 1001 | 1100 | 0101 | 0110 | 0011 | 0000 | 1110 |
| | **11** | 1011 | 1000 | 1100 | 0111 | 0001 | 1110 | 0010 | 1101 | 0110 | 1111 | 0000 | 1001 | 1010 | 0100 | 0101 | 0011 |



# Nonlinear Order

- nonlinear order is best established from algebraic normal form
- e.g. $S5_1(x_1, x_2, x_3, x_4, x_5, x_6) = x_2 + x_5 + x_6 + x_1x_3 + x_1x_5 + x_2x_4 + x_3x_4 + x_3x_6 + x_4x_5 + x_4x_6 + x_5x_6 + x_1x_2x_4 + x_1x_3x_6 + x_1x_4x_6 + x_1x_5x_6 + x_2x_3x_6 + x_2x_4x_5 + x_2x_4x_6 + x_3x_4x_5 + x_3x_4x_6 + x_1x_2x_3x_4 + x_1x_2x_4x_5 + x_1x_2x_3x_6 + x_1x_2x_4x_6 + x_1x_2x_5x_6 + x_1x_3x_4x_5 + x_1x_3x_5x_6 + x_2x_3x_5x_6 + x_3x_4x_5x_6 + x_1x_2x_4x_5x_6$, hence nonlinear order is 5
- all of the 32 S-Boxes' components have nonlinear order 5
- rather informative property: e.g. $a(x_1, \ldots, x_n) + x_1x_2 \cdots x_n$, where $a(x_1, \ldots, x_n)$ is an affine function, differs from an affine function only in one point



# Nonlinear Order

- nonlinear order is best established from algebraic normal form
- e.g. $S5_1(x_1, x_2, x_3, x_4, x_5, x_6) = x_2 + x_5 + x_6 + x_1x_3 + x_1x_5 + x_2x_4 + x_3x_4 + x_3x_6 + x_4x_5 + x_4x_6 + x_5x_6 + x_1x_2x_4 + x_1x_3x_6 + x_1x_4x_6 + x_1x_5x_6 + x_2x_3x_6 + x_2x_4x_5 + x_2x_4x_6 + x_3x_4x_5 + x_3x_4x_6 + x_1x_2x_3x_4 + x_1x_2x_4x_5 + x_1x_2x_3x_6 + x_1x_2x_4x_6 + x_1x_2x_5x_6 + x_1x_3x_4x_5 + x_1x_3x_5x_6 + x_2x_3x_5x_6 + x_3x_4x_5x_6 + x_1x_2x_4x_5x_6$, hence nonlinear order is 5
- all of the 32 S-Boxes' components have nonlinear order 5
- rather informative property: e.g. $a(x_1, \ldots, x_n) + x_1x_2 \cdots x_n$, where $a(x_1, \ldots, x_n)$ is an affine function, differs from an affine function only in one point

# Nonlinear Order

- nonlinear order is best established from algebraic normal form
- e.g. $S5_1(x_1, x_2, x_3, x_4, x_5, x_6) = x_2 + x_5 + x_6 + x_1x_3 + x_1x_5 + x_2x_4 + x_3x_4 + x_3x_6 + x_4x_5 + x_4x_6 + x_5x_6 + x_1x_2x_4 + x_1x_3x_6 + x_1x_4x_6 + x_1x_5x_6 + x_2x_3x_6 + x_2x_4x_5 + x_2x_4x_6 + x_3x_4x_5 + x_3x_4x_6 + x_1x_2x_3x_4 + x_1x_2x_4x_5 + x_1x_2x_3x_6 + x_1x_2x_4x_6 + x_1x_2x_5x_6 + x_1x_3x_4x_5 + x_1x_3x_5x_6 + x_2x_3x_5x_6 + x_3x_4x_5x_6 + x_1x_2x_4x_5x_6$, hence nonlinear order is 5
- all of the 32 S-Boxes' components have nonlinear order 5
- rather informative property: e.g. $a(x_1, \ldots, x_n) + x_1x_2 \cdots x_n$, where $a(x_1, \ldots, x_n)$ is an affine function, differs from an affine function only in one point

# Nonlinear Order

- nonlinear order is best established from algebraic normal form
- e.g. $S5_1(x_1, x_2, x_3, x_4, x_5, x_6) = x_2 + x_5 + x_6 + x_1x_3 + x_1x_5 + x_2x_4 + x_3x_4 + x_3x_6 + x_4x_5 + x_4x_6 + x_5x_6 + x_1x_2x_4 + x_1x_3x_6 + x_1x_4x_6 + x_1x_5x_6 + x_2x_3x_6 + x_2x_4x_5 + x_2x_4x_6 + x_3x_4x_5 + x_3x_4x_6 + x_1x_2x_3x_4 + x_1x_2x_4x_5 + x_1x_2x_3x_6 + x_1x_2x_4x_6 + x_1x_2x_5x_6 + x_1x_3x_4x_5 + x_1x_3x_5x_6 + x_2x_3x_5x_6 + x_3x_4x_5x_6 + x_1x_2x_4x_5x_6$, hence nonlinear order is 5
- all of the 32 S-Boxes' components have nonlinear order 5
- rather informative property: e.g. $a(x_1, \ldots, x_n) + x_1x_2 \cdots x_n$, where $a(x_1, \ldots, x_n)$ is an affine function, differs from an affine function only in one point

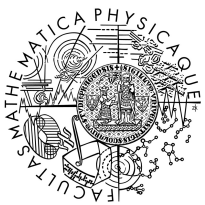
# Nonlinear Order – cont.

Why do the functions have nonlinear order only 5?

**Proposition**

*Let $f$ be a Boolean function of $n$ variables with $n > 2$. If $\mathrm{ord}(f) = n$, the autocorrelation function $\hat{r}_f$ can have no zeroes.*

However, zeros emerge in the function values of $\hat{r}_f$ rather scarcely.



# Distance from affine functions

## Definition

Let $f$ be a Boolean function of $n$ variables. The distance to the set of affine functions of $f$ is defined as

$$dist_{aff}(f) = \min_a d_H(f, a),$$

where $a$ is an arbitrary affine function in $n$ variables.

The $dist_{aff}(f)$ can be connected with the function's Walsh-Hadamard spectrum by

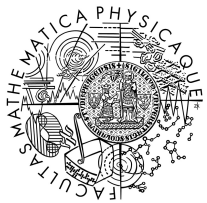
## Lemma

*Let $f$ be a Boolean function of $n$ variables. Then $dist_{aff}(f) = 2^{n-1} - 2^{n-1}\max_w |\mathcal{W}(\hat{f})|$, where $\mathcal{W}(\hat{f})$ is the Walsh-Hadamard spectrum of $f$.*

# Distance from affine functions – cont.

- it can be shown that $dist_{aff}(f)$ is upper-bounded by $2^{n-1} - 2^{\lfloor \frac{n}{2} \rfloor - 1}$, in our case $(n = 6)$ by 28
- what are the actual values of distance in DES S-Boxes?

| $dist_{aff}(f)$ | total |
|---|---|
| 14 | 1 ($S7_2$) |
| 18 | 8 |
| 20 | 12 |
| 22 | 11 |



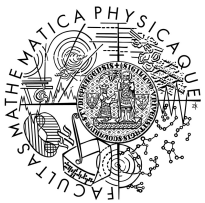
# Distance from affine functions – cont.

- it can be shown that $dist_{aff}(f)$ is upper-bounded by $2^{n-1} - 2^{\lfloor \frac{n}{2} \rfloor - 1}$, in our case $(n = 6)$ by 28
- what are the actual values of distance in DES S-Boxes?

| $dist_{aff}(f)$ | total |
|---|---|
| 14 | 1 ($S7_2$) |
| 18 | 8 |
| 20 | 12 |
| 22 | 11 |

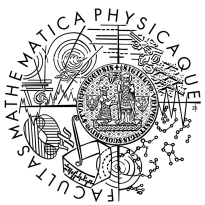
# Balancedness

- each S-Box is a map from $\mathbb{Z}_2^6$ **onto** $\mathbb{Z}_2^4$
- every component of each S-Box is balanced



# Balancedness

- each S-Box is a map from $\mathbb{Z}_2^6$ **onto** $\mathbb{Z}_2^4$
- every component of each S-Box is balanced



# Propagation criterion

## Definition

Let $f$ be a Boolean function of $n$ variables. Then $f$ satisfies the propagation criterion of degree $k$, $PC(k)$ $(1 \leq k \leq n)$, iff $f(x)$ changes with a probability of $1/2$ whenever $i$ $(1 \leq i \leq k)$ bits of $x$ are complemented. Equivalently, $f$ satisfies $PC(k)$, iff $\hat{r}_f(\underline{w}) = 0$ for all $\underline{w} \in \mathbb{Z}_2^n$ such that $d_H(\underline{w}) \leq k$.

- the historic Strict Avalanche Criterion is equivalent to $PC(1)$
- the most desirable result, the *perfect nonlinearity*, is equivalent to $PC(n)$

# Propagation criterion

## Definition

Let $f$ be a Boolean function of $n$ variables. Then $f$ satisfies the propagation criterion of degree $k$, $PC(k)$ $(1 \leq k \leq n)$, iff $f(x)$ changes with a probability of $1/2$ whenever $i$ $(1 \leq i \leq k)$ bits of $x$ are complemented. Equivalently, $f$ satisfies $PC(k)$, iff $\hat{r}_f(\underline{w}) = 0$ for all $\underline{w} \in \mathbb{Z}_2^n$ such that $d_H(\underline{w}) \leq k$.

- the historic Strict Avalanche Criterion is equivalent to $PC(1)$
- the most desirable result, the *perfect nonlinearity*, is equivalent to $PC(n)$

# Propagation criterion – cont.

- important from the differential point of view
- every single component of DES S-Boxes does not satisfy even $PC(1)\ldots$
- However, some autocorrelation function do have some zeros:

| $\underline{w}$ | $\hat{r}_{S1_1}(\underline{w})$ | $\hat{r}_{S8_1}(\underline{w})$ | $\hat{r}_{S7_2}(\underline{w})$ |
|---|---|---|---|
| (0, 0, 0, 0, 0, 1) | 0.000 | 0.375 | 0.125 |
| (0, 0, 0, 0, 1, 0) | 0.500 | 0.250 | 0.250 |
| (0, 0, 0, 1, 0, 0) | 0.000 | 0.500 | 0.125 |
| (0, 0, 1, 0, 0, 0) | 0.000 | 0.250 | -0.125 |
| (0, 1, 0, 0, 0, 0) | 0.375 | 0.250 | 0.750 |
| (1, 0, 0, 0, 0, 0) | 0.500 | 0.000 | 0.250 |

# Propagation criterion – cont.

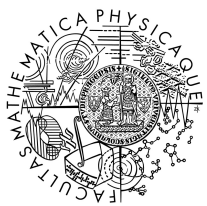
- important from the differential point of view
- every single component of DES S-Boxes does not satisfy even $PC(1)\ldots$
- However, some autocorrelation function do have some zeros:

| $\underline{w}$ | $\hat{r}_{S1_1}(\underline{w})$ | $\hat{r}_{S8_1}(\underline{w})$ | $\hat{r}_{S7_2}(\underline{w})$ |
|---|---|---|---|
| (0, 0, 0, 0, 0, 1) | 0.000 | 0.375 | 0.125 |
| (0, 0, 0, 0, 1, 0) | 0.500 | 0.250 | 0.250 |
| (0, 0, 0, 1, 0, 0) | 0.000 | 0.500 | 0.125 |
| (0, 0, 1, 0, 0, 0) | 0.000 | 0.250 | -0.125 |
| (0, 1, 0, 0, 0, 0) | 0.375 | 0.250 | 0.750 |
| (1, 0, 0, 0, 0, 0) | 0.500 | 0.000 | 0.250 |

# Propagation criterion – cont.

- important from the differential point of view
- every single component of DES S-Boxes does not satisfy even $PC(1)\ldots$
- However, some autocorrelation function do have some zeros:

| $\underline{w}$ | $\hat{r}_{S1_1}(\underline{w})$ | $\hat{r}_{S8_1}(\underline{w})$ | $\hat{r}_{S7_2}(\underline{w})$ |
|---|---|---|---|
| (0, 0, 0, 0, 0, 1) | 0.000 | 0.375 | 0.125 |
| (0, 0, 0, 0, 1, 0) | 0.500 | 0.250 | 0.250 |
| (0, 0, 0, 1, 0, 0) | 0.000 | 0.500 | 0.125 |
| (0, 0, 1, 0, 0, 0) | 0.000 | 0.250 | -0.125 |
| (0, 1, 0, 0, 0, 0) | 0.375 | 0.250 | 0.750 |
| (1, 0, 0, 0, 0, 0) | 0.500 | 0.000 | 0.250 |



# Differential attack on 3-round DES

## Definition

Let $S_j$ be a DES S-Box, $1 \leq j \leq 8$, and $(B_j, B_j^*)$ an ordered pair of bitstrings of length 6. We define the input x-or as $B_j' = B_j \oplus B_j^*$ and the output x-or as $C_j' = S_j(B_j) \oplus S_j(B_j^*)$.

## Definition

For $1 \leq j \leq 8$ and for bitstrings $B_j'$ of length 6 and $C_j'$ of length 4, define $IN_j(B_j', C_j') = \{B_j \in \mathbb{Z}_2^6 : S_j(B_j) \oplus S_j(B_j \oplus B_j') = C_j'\}$ and $N_j(B_j', C_j') = |IN_j(B_j', C_j')|$.

# Differential attack on 3-round DES

## Definition

Let $S_j$ be a DES S-Box, $1 \leq j \leq 8$, and $(B_j, B_j^*)$ an ordered pair of bitstrings of length 6. We define the input x-or as $B_j' = B_j \oplus B_j^*$ and the output x-or as $C_j' = S_j(B_j) \oplus S_j(B_j^*)$.

## Definition

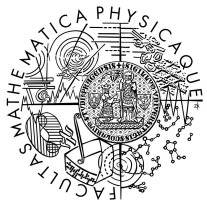
For $1 \leq j \leq 8$ and for bitstrings $B_j'$ of length 6 and $C_j'$ of length 4, define $IN_j(B_j', C_j') = \{B_j \in \mathbb{Z}_2^6 : S_j(B_j) \oplus S_j(B_j \oplus B_j') = C_j'\}$ and $N_j(B_j', C_j') = |IN_j(B_j', C_j')|$.

# Differential attack on 3-round DES – cont.

- input x-or in round $i$ does not depend on key bits: let $B = E \oplus J$, where $E$ is the expansion of input word $(R_{i-1})$ and $J = K_i$ consists of key bits for the $i$-th round. Then $B \oplus B^* = (E \oplus J) \oplus (E^* \oplus J) = E \oplus E^*$.

**Definition**

Let $E_j$ and $E_j^*$ be bitstrings of length 6 and $C_j'$ of length 4. Define $test_j = (E_j, E_j^*, C_j') = \{B_j \oplus E_j : B_j \in IN_j(E_j', C_j')\}$, where $E_j' = E_j \oplus E_j^*$.

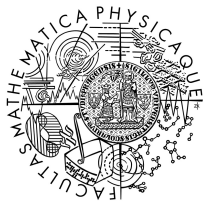


# Differential attack on 3-round DES – cont.

- input x-or in round $i$ does not depend on key bits: let $B = E \oplus J$, where $E$ is the expansion of input word $(R_{i-1})$ and $J = K_i$ consists of key bits for the $i$-th round. Then $B \oplus B^* = (E \oplus J) \oplus (E^* \oplus J) = E \oplus E^*$.

**Definition**

Let $E_j$ and $E_j^*$ be bitstrings of length 6 and $C_j'$ of length 4. Define $test_j = (E_j, E_j^*, C_j') = \{B_j \oplus E_j : B_j \in IN_j(E_j', C_j')\}$, where $E_j' = E_j \oplus E_j^*$.



# Differential attack on 3-round DES – cont.

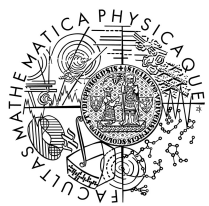
- Let $L_0R_0$ and $L_3R_3$ be respectively input and output in a 3-round DES. Then $R_3 = L_2 \oplus f(R_2, K_3) = R_1 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3)$.
- When having two such pairs of plaintext and ciphertext, then $R'_3 = L'_0 \oplus f(R_0, K_1) \oplus f(R^*_0, K_1) \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- Suppose that $R_0 = R^*_0$, hence $f(R_0, K_1) = f(R^*_0, K_1)$ and $R'_3 = L'_0 \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- $R'_3$ as well as $L'_0$ are known (they can be computed from ciphertexts and plaintexts)
- therefore holds that $f(R_2, K_3) \oplus f(R^*_2, K_3) = R'_3 \oplus L'_0$
- $f(R_2, K_3) = P(C)$, where $C$ is the output of S-Box and $P$ publicly known fixed permutation, hence $P(C) \oplus P(C^*) = R'_3 \oplus L'_0$
- finally, $C' = C \oplus C^* = P^{-1}(R'_3 \oplus L'_0)$ is the output x-or of S-Box in round 3.

# Differential attack on 3-round DES – cont.

- Let $L_0R_0$ and $L_3R_3$ be respectively input and output in a 3-round DES. Then $R_3 = L_2 \oplus f(R_2, K_3) = R_1 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3)$.
- When having two such pairs of plaintext and ciphertext, then $R'_3 = L'_0 \oplus f(R_0, K_1) \oplus f(R^*_0, K_1) \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- Suppose that $R_0 = R^*_0$, hence $f(R_0, K_1) = f(R^*_0, K_1)$ and $R'_3 = L'_0 \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- $R'_3$ as well as $L'_0$ are known (they can be computed from ciphertexts and plaintexts)
- therefore holds that $f(R_2, K_3) \oplus f(R^*_2, K_3) = R'_3 \oplus L'_0$
- $f(R_2, K_3) = P(C)$, where $C$ is the output of S-Box and $P$ publicly known fixed permutation, hence $P(C) \oplus P(C^*) = R'_3 \oplus L'_0$
- finally, $C' = C \oplus C^* = P^{-1}(R'_3 \oplus L'_0)$ is the output x-or of S-Box in round 3.

# Differential attack on 3-round DES – cont.

- Let $L_0R_0$ and $L_3R_3$ be respectively input and output in a 3-round DES. Then $R_3 = L_2 \oplus f(R_2, K_3) = R_1 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3)$.
- When having two such pairs of plaintext and ciphertext, then $R'_3 = L'_0 \oplus f(R_0, K_1) \oplus f(R^*_0, K_1) \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- Suppose that $R_0 = R^*_0$, hence $f(R_0, K_1) = f(R^*_0, K_1)$ and $R'_3 = L'_0 \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- $R'_3$ as well as $L'_0$ are known (they can be computed from ciphertexts and plaintexts)
- therefore holds that $f(R_2, K_3) \oplus f(R^*_2, K_3) = R'_3 \oplus L'_0$
- $f(R_2, K_3) = P(C)$, where $C$ is the output of S-Box and $P$ publicly known fixed permutation, hence $P(C) \oplus P(C^*) = R'_3 \oplus L'_0$
- finally, $C' = C \oplus C^* = P^{-1}(R'_3 \oplus L'_0)$ is the output x-or of S-Box in round 3.

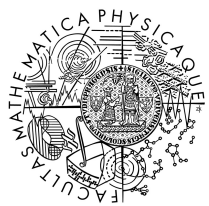


# Differential attack on 3-round DES – cont.

- Let $L_0R_0$ and $L_3R_3$ be respectively input and output in a 3-round DES. Then $R_3 = L_2 \oplus f(R_2, K_3) = R_1 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3)$.
- When having two such pairs of plaintext and ciphertext, then $R'_3 = L'_0 \oplus f(R_0, K_1) \oplus f(R^*_0, K_1) \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- Suppose that $R_0 = R^*_0$, hence $f(R_0, K_1) = f(R^*_0, K_1)$ and $R'_3 = L'_0 \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- $R'_3$ as well as $L'_0$ are known (they can be computed from ciphertexts and plaintexts)
- therefore holds that $f(R_2, K_3) \oplus f(R^*_2, K_3) = R'_3 \oplus L'_0$
- $f(R_2, K_3) = P(C)$, where $C$ is the output of S-Box and $P$ publicly known fixed permutation, hence $P(C) \oplus P(C^*) = R'_3 \oplus L'_0$
- finally, $C' = C \oplus C^* = P^{-1}(R'_3 \oplus L'_0)$ is the output x-or of S-Box in round 3.

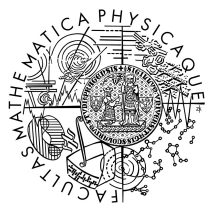
# Differential attack on 3-round DES – cont.

- Let $L_0R_0$ and $L_3R_3$ be respectively input and output in a 3-round DES. Then $R_3 = L_2 \oplus f(R_2, K_3) = R_1 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3)$.
- When having two such pairs of plaintext and ciphertext, then $R'_3 = L'_0 \oplus f(R_0, K_1) \oplus f(R^*_0, K_1) \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- Suppose that $R_0 = R^*_0$, hence $f(R_0, K_1) = f(R^*_0, K_1)$ and $R'_3 = L'_0 \oplus f(R_2, K_3) \oplus f(R^*_2, K_3)$.
- $R'_3$ as well as $L'_0$ are known (they can be computed from ciphertexts and plaintexts)
- therefore holds that $f(R_2, K_3) \oplus f(R^*_2, K_3) = R'_3 \oplus L'_0$
- $f(R_2, K_3) = P(C)$, where $C$ is the output of S-Box and $P$ publicly known fixed permutation, hence $P(C) \oplus P(C^*) = R'_3 \oplus L'_0$
- finally, $C' = C \oplus C^* = P^{-1}(R'_3 \oplus L'_0)$ is the output x-or of S-Box in round 3.



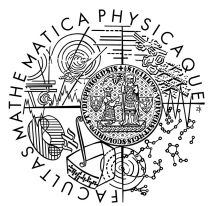
# Differential attack on 3-round DES – cont.

- Let $L_0R_0$ and $L_3R_3$ be respectively input and output in a 3-round DES. Then $R_3 = L_2 \oplus f(R_2, K_3) = R_1 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3)$.
- When having two such pairs of plaintext and ciphertext, then $R_3' = L_0' \oplus f(R_0, K_1) \oplus f(R_0^*, K_1) \oplus f(R_2, K_3) \oplus f(R_2^*, K_3)$.
- Suppose that $R_0 = R_0^*$, hence $f(R_0, K_1) = f(R_0^*, K_1)$ and $R_3' = L_0' \oplus f(R_2, K_3) \oplus f(R_2^*, K_3)$.
- $R_3'$ as well as $L_0'$ are known (they can be computed from ciphertexts and plaintexts)
- therefore holds that $f(R_2, K_3) \oplus f(R_2^*, K_3) = R_3' \oplus L_0'$
- $f(R_2, K_3) = P(C)$, where $C$ is the output of S-Box and $P$ publicly known fixed permutation, hence $P(C) \oplus P(C^*) = R_3' \oplus L_0'$
- finally, $C' = C \oplus C^* = P^{-1}(R_3' \oplus L_0')$ is the output x-or of S-Box in round 3.



# Differential attack on 3-round DES – cont.

- Let $L_0R_0$ and $L_3R_3$ be respectively input and output in a 3-round DES. Then $R_3 = L_2 \oplus f(R_2, K_3) = R_1 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3)$.
- When having two such pairs of plaintext and ciphertext, then $R_3' = L_0' \oplus f(R_0, K_1) \oplus f(R_0^*, K_1) \oplus f(R_2, K_3) \oplus f(R_2^*, K_3)$.
- Suppose that $R_0 = R_0^*$, hence $f(R_0, K_1) = f(R_0^*, K_1)$ and $R_3' = L_0' \oplus f(R_2, K_3) \oplus f(R_2^*, K_3)$.
- $R_3'$ as well as $L_0'$ are known (they can be computed from ciphertexts and plaintexts)
- therefore holds that $f(R_2, K_3) \oplus f(R_2^*, K_3) = R_3' \oplus L_0'$
- $f(R_2, K_3) = P(C)$, where $C$ is the output of S-Box and $P$ publicly known fixed permutation, hence $P(C) \oplus P(C^*) = R_3' \oplus L_0'$
- finally, $C' = C \oplus C^* = P^{-1}(R_3' \oplus L_0')$ is the output x-or of S-Box in round 3.



# Differential attack on 3-round DES – cont.

- input x-or can be computed as well, since $R_2 = L_3$ and $R_2^* = L_3^*$ are known
- we thus know $E$, $E^*$ and $C'$ for the third round
- let us denote $E = E_1E_2E_3E_4E_5E_6E_7E_8$ ($E_j$ being a string of 8 bits) and similarly $E^*$, $C'$ and $J$ (the key bits for round 3)

**Proposition**

*Suppose $E_j$ and $E_j^*$ are two inputs to the S-Box $S_j$ and the output x-or for $S_j$ is $C_j'$. Denote $E_j' = E_j \oplus E_j^*$. Then the key bits $J_j$ occur in the set test$_j(E_j, E_j^*, C_j')$.*

# Differential attack on 3-round DES – cont.

- input x-or can be computed as well, since $R_2 = L_3$ and $R_2^* = L_3^*$ are known
- we thus know $E$, $E^*$ and $C'$ for the third round
- let us denote $E = E_1E_2E_3E_4E_5E_6E_7E_8$ ($E_j$ being a string of 8 bits) and similarly $E^*$, $C'$ and $J$ (the key bits for round 3)

**Proposition**

*Suppose $E_j$ and $E_j^*$ are two inputs to the S-Box $S_j$ and the output x-or for $S_j$ is $C_j'$. Denote $E_j' = E_j \oplus E_j^*$. Then the key bits $J_j$ occur in the set $test_j(E_j, E_j^*, C_j')$.*

# Differential attack on 3-round DES – cont.

- input x-or can be computed as well, since $R_2 = L_3$ and $R_2^* = L_3^*$ are known
- we thus know $E$, $E^*$ and $C'$ for the third round
- let us denote $E = E_1E_2E_3E_4E_5E_6E_7E_8$ ($E_j$ being a string of 8 bits) and similarly $E^*$, $C'$ and $J$ (the key bits for round 3)

**Proposition**

*Suppose $E_j$ and $E_j^*$ are two inputs to the S-Box $S_j$ and the output x-or for $S_j$ is $C_j'$. Denote $E_j' = E_j \oplus E_j^*$. Then the key bits $J_j$ occur in the set test$_j(E_j, E_j^*, C_j')$.*

# Differential attack on 3-round DES – cont.

- input x-or can be computed as well, since $R_2 = L_3$ and $R_2^* = L_3^*$ are known
- we thus know $E$, $E^*$ and $C'$ for the third round
- let us denote $E = E_1E_2E_3E_4E_5E_6E_7E_8$ ($E_j$ being a string of 8 bits) and similarly $E^*$, $C'$ and $J$ (the key bits for round 3)

**Proposition**

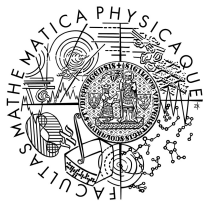
*Suppose $E_j$ and $E_j^*$ are two inputs to the S-Box $S_j$ and the output x-or for $S_j$ is $C_j'$. Denote $E_j' = E_j \oplus E_j^*$. Then the key bits $J_j$ occur in the set test$_j(E_j, E_j^*, C_j')$.*

# Differential attack on 3-round DES – cont.

- according to the proposition, we construct the sets $test_j$, $1 \leq j \leq 8$
- this we perform for more plaintext-ciphertext pairs, enciphered by the same unknown key
- we hope that after intersecting all the sets, only one element remains: the right key bits for round 3
- we get only 48 out of 56 key bits, but the remaining bits can be found by an exhaustive search of the $2^8 = 256$ possibilities

# Differential attack on 3-round DES – cont.

- according to the proposition, we construct the sets $test_j$, $1 \leq j \leq 8$
- this we perform for more plaintext-ciphertext pairs, enciphered by the same unknown key
- we hope that after intersecting all the sets, only one element remains: the right key bits for round 3
- we get only 48 out of 56 key bits, but the remaining bits can be found by an exhaustive search of the $2^8 = 256$ possibilities



# Differential attack on 3-round DES – cont.

- according to the proposition, we construct the sets $test_j$, $1 \leq j \leq 8$
- this we perform for more plaintext-ciphertext pairs, enciphered by the same unknown key
- we hope that after intersecting all the sets, only one element remains: the right key bits for round 3
- we get only 48 out of 56 key bits, but the remaining bits can be found by an exhaustive search of the $2^8 = 256$ possibilities

# Differential attack on 3-round DES – cont.

- according to the proposition, we construct the sets $test_j$, $1 \leq j \leq 8$
- this we perform for more plaintext-ciphertext pairs, enciphered by the same unknown key
- we hope that after intersecting all the sets, only one element remains: the right key bits for round 3
- we get only 48 out of 56 key bits, but the remaining bits can be found by an exhaustive search of the $2^8 = 256$ possibilities

# AES S-Box

- AES uses Rijndael's Finite Field $GF(2^8)$ with the irreducible polynomial being $x^8 + x^4 + x^3 + x + 1$
- each byte represents coefficients of a polynomial in Rijndael's Finite Field
- byte operations are thus performed as polynomial addition/multiplication modulo $x^8 + x^4 + x^3 + x + 1$

# AES S-Box

- AES uses Rijndael's Finite Field $GF(2^8)$ with the irreducible polynomial being $x^8 + x^4 + x^3 + x + 1$
- each byte represents coefficients of a polynomial in Rijndael's Finite Field
- byte operations are thus performed as polynomial addition/multiplication modulo $x^8 + x^4 + x^3 + x + 1$

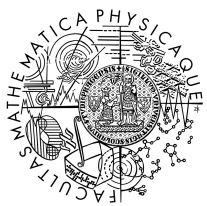
# AES S-Box

- AES uses Rijndael's Finite Field $GF(2^8)$ with the irreducible polynomial being $x^8 + x^4 + x^3 + x + 1$
- each byte represents coefficients of a polynomial in Rijndael's Finite Field
- byte operations are thus performed as polynomial addition/multiplication modulo $x^8 + x^4 + x^3 + x + 1$

# AES S-Box – cont.

- since we work in Rijndael's Finite Field, there exists a multiplicative inverse for all nonzero elements
- for completness, we formally define zero's inverse as zero itself
- inverse operation is followed by an affine transformation:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$



# AES S-Box – cont.

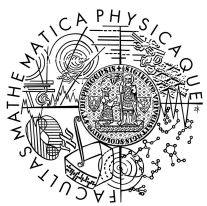
- since we work in Rijndael's Finite Field, there exists a multiplicative inverse for all nonzero elements
- for completness, we formally define zero's inverse as zero itself
- inverse operation is followed by an affine transformation:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

# AES S-Box – cont.

- since we work in Rijndael's Finite Field, there exists a multiplicative inverse for all nonzero elements
- for completness, we formally define zero's inverse as zero itself
- inverse operation is followed by an affine transformation:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$



# AES S-Box – cont.

- AES S-Box is precisely defined and its properties are well researched, not a random table like DES
- invertion in Rijndael's Finite Field is highly nonlinear, the affine transformation breaks down the field structure
- S-Box is either represented by a look-up table or, in some 8-bit hardware applications, implemented as circuits and computed on the fly

# AES S-Box – cont.

- AES S-Box is precisely defined and its properties are well researched, not a random table like DES
- invertion in Rijndael's Finite Field is highly nonlinear, the affine transformation breaks down the field structure
- S-Box is either represented by a look-up table or, in some 8-bit hardware applications, implemented as circuits and computed on the fly

# AES S-Box – cont.

- AES S-Box is precisely defined and its properties are well researched, not a random table like DES
- invertion in Rijndael's Finite Field is highly nonlinear, the affine transformation breaks down the field structure
- S-Box is either represented by a look-up table or, in some 8-bit hardware applications, implemented as circuits and computed on the fly

# Description of Skipjack

- Skipjack is an iterated block cipher with block size 64 bits, key size 80 bits and 32 rounds of two types (Rule A and B)
- rounds can be described as a linear feedback shift register with additional keyed permutation $G$ and a counter, which starts at 1 and increases in each round by one
- first 16 rounds of Skipjack can be depicted as follows (the other 16 are similar)
- we consider 31-round version of Ski (minus the last one)

# Description of Skipjack

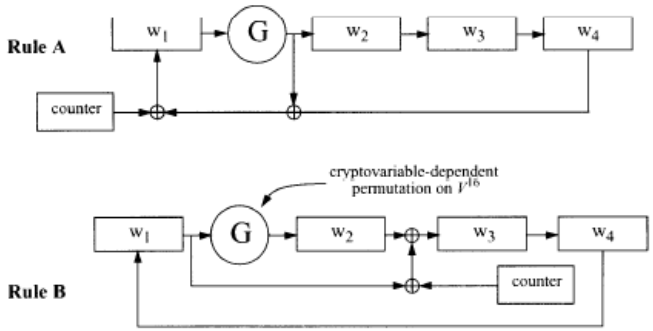
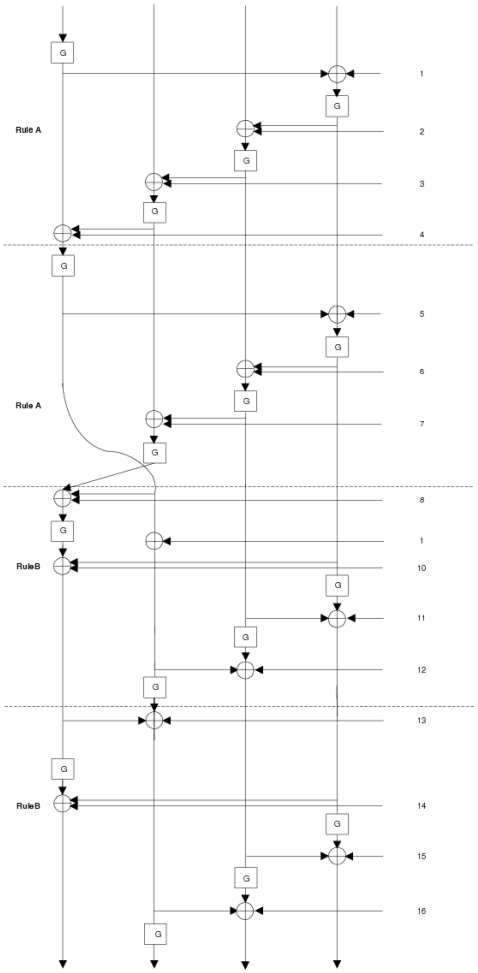



*Figure 5.* "SKIPJACK Stepping Rules"

- Skipjack is an iterated block cipher with block size 64 bits, key size 80 bits and 32 rounds of two types (Rule A and B)
- rounds can be described as a linear feedback shift register with additional keyed permutation $G$ and a counter, which starts at 1 and increases in each round by one
- first 16 rounds of Skipjack can be depicted as follows (the other 16 are similar)
- we consider 31-round version of Ski (minus the last one)

# Description of Skipjack




- Skipjack is an iterated block cipher with block size 64 bits, key size 80 bits and 32 rounds of two types (Rule A and B)
- rounds can be described as a linear feedback shift register with additional keyed permutation $G$ and a counter, which starts at 1 and increases in each round by one
- first 16 rounds of Skipjack can be depicted as follows (the other 16 are similar)
- we consider 31-round version of Skipjack (minus the last one)

# Description of Skipjack

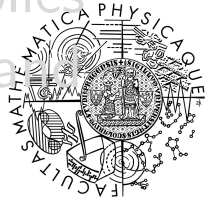
- Skipjack is an iterated block cipher with block size 64 bits, key size 80 bits and 32 rounds of two types (Rule A and B)
- rounds can be described as a linear feedback shift register with additional keyed permutation $G$ and a counter, which starts at 1 and increases in each round by one
- first 16 rounds of Skipjack can be depicted as follows (the other 16 are similar)
- we consider 31-round version of Skipjack (minus the last one)

# Impossible differentials

- uses *impossible differentials* instead of the most probable (desirable) ones
- to find these, we use the miss-in-the-middle technique
- e.g. second input word of round 5 does not influence the fourth word after round 16, hence input x-or $(0, a, 0, 0)$ to the round 5 implies output x-or $(c, d, e, 0)$ for some non-zero $c, d, e$ after round 16
- backwards, first output word of round 28 is not affected by the third word before round 16, therefore output x-or $(b, 0, 0, 0)$ after round 28 implies input x-or $(f, g, 0, h)$ to the round 16
- combining these, having input x-or to the round 5 equal to $(0, a, 0, 0)$ and output x-or after round 28 $(b, 0, 0, 0)$ implies that after round 16, the x-ors are respectively $(c, d, e, 0)$ a $(f, g, 0, h)$. But $e, h \neq 0$, which is contradictory and this differential can never occur.

# Impossible differentials

- uses *impossible differentials* instead of the most probable (desirable) ones
- to find these, we use the miss-in-the-middle technique
- e.g. second input word of round 5 does not influence the fourth word after round 16, hence input x-or $(0, a, 0, 0)$ to the round 5 implies output x-or $(c, d, e, 0)$ for some non-zero $c, d, e$ after round 16
- backwards, first output word of round 28 is not affected by the third word before round 16, therefore output x-or $(b, 0, 0, 0)$ after round 28 implies input x-or $(f, g, 0, h)$ to the round 16
- combining these, having input x-or to the round 5 equal to $(0, a, 0, 0)$ and output x-or after round 28 $(b, 0, 0, 0)$ implies that after round 16, the x-ors are respectively $(c, d, e, 0)$ a $(f, g, 0, h)$. But $e, h \neq 0$, which is contradictory and this differential can never occur.

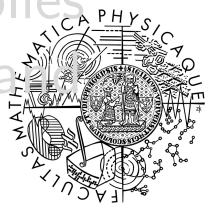


# Impossible differentials

- uses *impossible differentials* instead of the most probable (desirable) ones
- to find these, we use the miss-in-the-middle technique
- e.g. second input word of round 5 does not influence the fourth word after round 16, hence input x-or $(0, a, 0, 0)$ to the round 5 implies output x-or $(c, d, e, 0)$ for some non-zero $c, d, e$ after round 16
- backwards, first output word of round 28 is not affected by the third word before round 16, therefore output x-or $(b, 0, 0, 0)$ after round 28 implies input x-or $(f, g, 0, h)$ to the round 16
- combining these, having input x-or to the round 5 equal to $(0, a, 0, 0)$ and output x-or after round 28 $(b, 0, 0, 0)$ implies that after round 16, the x-ors are respectively $(c, d, e, 0)$ a $(f, g, 0, h)$. But $e, h \neq 0$, which is contradictory and this differential can never occur.

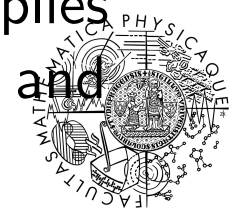
# Impossible differentials

- uses *impossible differentials* instead of the most probable (desirable) ones
- to find these, we use the miss-in-the-middle technique
- e.g. second input word of round 5 does not influence the fourth word after round 16, hence input x-or $(0, a, 0, 0)$ to the round 5 implies output x-or $(c, d, e, 0)$ for some non-zero $c, d, e$ after round 16
- backwards, first output word of round 28 is not affected by the third word before round 16, therefore output x-or $(b, 0, 0, 0)$ after round 28 implies input x-or $(f, g, 0, h)$ to the round 16
- combining these, having input x-or to the round 5 equal to $(0, a, 0, 0)$ and output x-or after round 28 $(b, 0, 0, 0)$ implies that after round 16, the x-ors are respectively $(c, d, e, 0)$ a $(f, g, 0, h)$. But $e, h \neq 0$, which is contradictory and this differential can never occur.



# Impossible differentials

- uses *impossible differentials* instead of the most probable (desirable) ones
- to find these, we use the miss-in-the-middle technique
- e.g. second input word of round 5 does not influence the fourth word after round 16, hence input x-or $(0, a, 0, 0)$ to the round 5 implies output x-or $(c, d, e, 0)$ for some non-zero $c, d, e$ after round 16
- backwards, first output word of round 28 is not affected by the third word before round 16, therefore output x-or $(b, 0, 0, 0)$ after round 28 implies input x-or $(f, g, 0, h)$ to the round 16
- combining these, having input x-or to the round 5 equal to $(0, a, 0, 0)$ and output x-or after round 28 $(b, 0, 0, 0)$ implies that after round 16, the x-ors are respectively $(c, d, e, 0)$ and $(f, g, 0, h)$. But $e, h \neq 0$, which is contradictory and this differential can never occur.



# Cryptanalysis of Skipjack reduced to 31 rounds

- the whole key (80 bits) is used during first four rounds, therefore candidate keys are of this size
- we use the impossible differential presented above and observe next that, if it holds for some pair of plaintexts-ciphertexts, then the third word of the plaintexts and third and fourth word of the ciphertexts are the same and the others have non-zero difference
- therefore, assuming that the difference holds, we get three 16-bit restrictions in round 1, 4 and 29
- it is reasonable to assume that one $2^{-48}$-th of all keys, i.e. $2^{32}$ candidate keys, encrypt the plaintext pair of chosen structure to input difference of the differential after round 4 and decrypt the ciphertext pair to the output difference of the differential after round 28

# Cryptanalysis of Skipjack reduced to 31 rounds

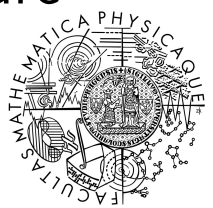
- the whole key (80 bits) is used during first four rounds, therefore candidate keys are of this size
- we use the impossible differential presented above and observe next that, if it holds for some pair of plaintexts-ciphertexts, then the third word of the plaintexts and third and fourth word of the ciphertexts are the same and the others have non-zero difference
- therefore, assuming that the difference holds, we get three 16-bit restrictions in round 1, 4 and 29
- it is reasonable to assume that one $2^{-48}$-th of all keys, i.e. $2^{32}$ candidate keys, encrypt the plaintext pair of chosen structure to input difference of the differential after round 4 and decrypt the ciphertext pair to the output difference of the differential after round 28

# Cryptanalysis of Skipjack reduced to 31 rounds

- the whole key (80 bits) is used during first four rounds, therefore candidate keys are of this size
- we use the impossible differential presented above and observe next that, if it holds for some pair of plaintexts-ciphertexts, then the third word of the plaintexts and third and fourth word of the ciphertexts are the same and the others have non-zero difference
- therefore, assuming that the difference holds, we get three 16-bit restrictions in round 1, 4 and 29
- it is reasonable to assume that one $2^{-48}$-th of all keys, i.e. $2^{32}$ candidate keys, encrypt the plaintext pair of chosen structure to input difference of the differential after round 4 and decrypt the ciphertext pair to the output difference of the differential after round 28

# Cryptanalysis of Skipjack reduced to 31 rounds

- the whole key (80 bits) is used during first four rounds, therefore candidate keys are of this size
- we use the impossible differential presented above and observe next that, if it holds for some pair of plaintexts-ciphertexts, then the third word of the plaintexts and third and fourth word of the ciphertexts are the same and the others have non-zero difference
- therefore, assuming that the difference holds, we get three 16-bit restrictions in round 1, 4 and 29
- it is reasonable to assume that one $2^{-48}$-th of all keys, i.e. $2^{32}$ candidate keys, encrypt the plaintext pair of chosen structure to input difference of the differential after round 4 and decrypt the ciphertext pair to the output difference of the differential after round 28

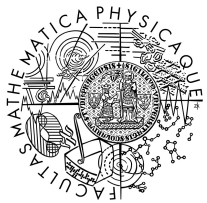


# Cryptanalysis of Skipjack reduced to 31 rounds – cont.

- so, for each pair, we can discard $2^{32}$ candidate keys
- after $2^{48}$ pairs, $2^{80}$ keys are discarded – $\frac{1}{e}$-th of keys however remain due to collisions
- choosing enough pairs to ensure that there remains only the correct key is more complex than exhaustive search
- we use only $2^{49}$ pairs, so about $2^{77}$ keys remain and among these we search exhaustively

# Cryptanalysis of Skipjack reduced to 31 rounds – cont.

- so, for each pair, we can discard $2^{32}$ candidate keys
- after $2^{48}$ pairs, $2^{80}$ keys are discarded – $\frac{1}{e}$-th of keys however remain due to collisions
- choosing enough pairs to ensure that there remains only the correct key is more complex than exhaustive search
- we use only $2^{49}$ pairs, so about $2^{77}$ keys remain and among these we search exhaustively

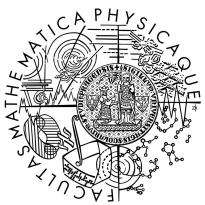


# Cryptanalysis of Skipjack reduced to 31 rounds – cont.

- so, for each pair, we can discard $2^{32}$ candidate keys
- after $2^{48}$ pairs, $2^{80}$ keys are discarded – $\frac{1}{e}$-th of keys however remain due to collisions
- choosing enough pairs to ensure that there remains only the correct key is more complex than exhaustive search
- we use only $2^{49}$ pairs, so about $2^{77}$ keys remain and among these we search exhaustively

# Cryptanalysis of Skipjack reduced to 31 rounds – cont.

- so, for each pair, we can discard $2^{32}$ candidate keys
- after $2^{48}$ pairs, $2^{80}$ keys are discarded – $\frac{1}{e}$-th of keys however remain due to collisions
- choosing enough pairs to ensure that there remains only the correct key is more complex than exhaustive search
- we use only $2^{49}$ pairs, so about $2^{77}$ keys remain and among these we search exhaustively



# Description of the Impossible Differential attack

- choose $2^{41}$ plaintexts with third word equal
- choose those pairs of ciphertexts, which collide at the third and fourth word – about $2^{49}$ are selected
- following analysis is applied to each pair:
- **analysis of round 1:**
  - we know two inputs to the permutation $G$ and its output difference
  - $G$ is keyed by 32 bits, so there are about $2^{16}$ possible subkeys
  - we guess first two bytes of the keys and compute the other two by differential techniques
  - this analysis lasts $2^{16}$ steps
  - since the round key is the same for rounds 1 and 31, we can peel off the last round for each candidate key

# Description of the Impossible Differential attack

- choose $2^{41}$ plaintexts with third word equal
- choose those pairs of ciphertexts, which collide at the third and fourth word – about $2^{49}$ are selected
- following analysis is applied to each pair:
- **analysis of round 1:**
  - we know two inputs to the permutation $G$ and its output difference
  - $G$ is keyed by 32 bits, so there are about $2^{16}$ possible subkeys
  - we guess first two bytes of the keys and compute the other two by differential techniques
  - this analysis lasts $2^{16}$ steps
  - since the round key is the same for rounds 1 and 31, we can peel off the last round for each candidate key

# Description of the Impossible Differential attack

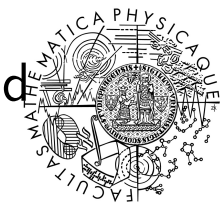
- choose $2^{41}$ plaintexts with third word equal
- choose those pairs of ciphertexts, which collide at the third and fourth word – about $2^{49}$ are selected
- following analysis is applied to each pair:
- **analysis of round 1:**
  - we know two inputs to the permutation $G$ and its output difference
  - $G$ is keyed by 32 bits, so there are about $2^{16}$ possible subkeys
  - we guess first two bytes of the keys and compute the other two by differential techniques
  - this analysis lasts $2^{16}$ steps
  - since the round key is the same for rounds 1 and 31, we can peel off the last round for each candidate key

# Description of the Impossible Differential attack

- choose $2^{41}$ plaintexts with third word equal
- choose those pairs of ciphertexts, which collide at the third and fourth word – about $2^{49}$ are selected
- following analysis is applied to each pair:
- **analysis of round 1:**
  - we know two inputs to the permutation $G$ and its output difference
  - $G$ is keyed by 32 bits, so there are about $2^{16}$ possible subkeys
  - we guess first two bytes of the keys and compute the other two by differential techniques
  - this analysis lasts $2^{16}$ steps
  - since the round key is the same for rounds 1 and 31, we can peel off the last round for each candidate key

# Description of the Impossible Differential attack – cont.

- **analysis of rounds 4 and 29:**
  - we know input and output differences of $G$ in round 4
  - $G$ has complementation properties: let $O = G_K(I)$ and let $d = (d_1, d_2)$ be any 16-bit value; then $O \oplus d = G_{K \oplus (d_1, d_0, d_1, d_0)}(I \oplus d)$
  - therefore we can assume that the input is equal to any arbitrary pair of values and find $2^{16}$ corresponding keys
  - extend to the complete list by means of complementation
  - similar analysis applies for round 29, moreover, rounds 4 and 29 use same round keys
  - we want to join these lists – but due to complementation and effectivity, we can not search directly
  - x-or of two first bytes is independent of complementation, so it can be used to join lists
  - we get about $2^{16}$ tuples of the subkey, input of round 4 and output of round 29
  - this lasts about $2^{16}$ steps



# Description of the Impossible Differential attack – cont.

**third phase of the analysis:**

- we now join the two lists (from the first and second phase) together
- the result would be a list of about $2^{32}$ entries of the form $(cv_0, \ldots, cv_5, X_3, X_{30})$ where $cv_0, \ldots, cv_5$ are the six key bytes from rounds 1, 4 and 29, $X_3$ is the output of $G$ after round 3 and $X_{30}$ is the input to $G$ in round 30.
- we consider second half of round 2 and first half of round 3 as one permutation (say $G'$) with the third plaintext word as a feedback in the middle.
- for each pair, to establish the remaining four bytes of the key (namely $cv_6, \ldots, cv_9$), we have two equalities: from round 3 (we know input and output of round 30) and from $G'$ (we know its two outputs)

# Description of the Impossible Differential attack – cont.

**third phase of the analysis:**

- we now join the two lists (from the first and second phase) together
- the result would be a list of about $2^{32}$ entries of the form $(cv_0, \ldots, cv_5, X_3, X_{30})$ where $cv_0, \ldots, cv_5$ are the six key bytes from rounds 1, 4 and 29, $X_3$ is the output of $G$ after round 3 and $X_{30}$ is the input to $G$ in round 30.
- we consider second half of round 2 and first half of round 3 as one permutation (say $G'$) with the third plaintext word as a feedback in the middle.
- for each pair, to establish the remaining four bytes of the key (namely $cv_6, \ldots, cv_9$), we have two equalities: from round 3 (we know input and output of round 30) and from $G'$ (we know its two outputs)

# Description of the Impossible Differential attack – cont.

**third phase of the analysis:**

- we now join the two lists (from the first and second phase) together
- the result would be a list of about $2^{32}$ entries of the form $(cv_0, \ldots, cv_5, X_3, X_{30})$ where $cv_0, \ldots, cv_5$ are the six key bytes from rounds 1, 4 and 29, $X_3$ is the output of $G$ after round 3 and $X_{30}$ is the input to $G$ in round 30.
- we consider second half of round 2 and first half of round 3 as one permutation (say $G'$) with the third plaintext word as a feedback in the middle.
- for each pair, to establish the remaining four bytes of the key (namely $cv_6, \ldots, cv_9$), we have two equalities: from round 3 (we know input and output of round 30) and from $G'$ (we know its two outputs)

# Description of the Impossible Differential attack – cont.

**third phase of the analysis:**

- we now join the two lists (from the first and second phase) together
- the result would be a list of about $2^{32}$ entries of the form $(cv_0, \ldots, cv_5, X_3, X_{30})$ where $cv_0, \ldots, cv_5$ are the six key bytes from rounds 1, 4 and 29, $X_3$ is the output of $G$ after round 3 and $X_{30}$ is the input to $G$ in round 30.
- we consider second half of round 2 and first half of round 3 as one permutation (say $G'$) with the third plaintext word as a feedback in the middle.
- for each pair, to establish the remaining four bytes of the key (namely $cv_6, \ldots, cv_9$), we have two equalities: from round 30 (we know input and output of round 30) and from $G'$ (we know its two outputs)

# Description of the Impossible Differential attack – cont.

- on average, there is one solution of $cv_6, \ldots, cv_9$ for each pair, so that the key encrypts plaintexts and decrypts ciphertexts to match the impossible differential, hence from each pair we gain one key to discard

**fourth phase of the analysis:**

- during phase three, we discarded about $2^{32}$ keys for each pair, we selected $2^{49}$ pairs from $2^{41}$ chosen plaintexts, but due to collisions, about $\frac{1}{e^2} \approx \frac{1}{8}$ of them is not discarded
- that leaves $2^{77}$ keys to try in phase four, which is done by exhaustive search

# Description of the Impossible Differential attack – cont.

- on average, there is one solution of $cv_6, \ldots, cv_9$ for each pair, so that the key encrypts plaintexts and decrypts ciphertexts to match the impossible differential, hence from each pair we gain one key to discard

**fourth phase of the analysis:**

- during phase three, we discarded about $2^{32}$ keys for each pair, we selected $2^{49}$ pairs from $2^{41}$ chosen plaintexts, but due to collisions, about $\frac{1}{e^2} \approx \frac{1}{8}$ of them is not discarded
- that leaves $2^{77}$ keys to try in phase four, which is done by exhaustive search

# Description of the Impossible Differential attack – cont.

- on average, there is one solution of $cv_6, \ldots, cv_9$ for each pair, so that the key encrypts plaintexts and decrypts ciphertexts to match the impossible differential, hence from each pair we gain one key to discard

**fourth phase of the analysis:**

- during phase three, we discarded about $2^{32}$ keys for each pair, we selected $2^{49}$ pairs from $2^{41}$ chosen plaintexts, but due to collisions, about $\frac{1}{e^2} \approx \frac{1}{8}$ of them is not discarded
- that leaves $2^{77}$ keys to try in phase four, which is done by exhaustive search

# Impossible Differential attack overview

- chosen plaintext attack, which requires a pool of $2^{41}$ plaintexts
- total complexity: $2^{78}$ steps, which is four times faster than brute-force
- can not be extended to cover all 32 rounds: the complexity would be in excess of exhaustive search
- the attacks are independent of the $G$ permutations
- lesson learned: block ciphers designers should not only prove resistance against conventional differential cryptanalysis by a low upper bound on the probability of characteristics, they should provide such proofs for the lower bound as well

# Impossible Differential attack overview

- chosen plaintext attack, which requires a pool of $2^{41}$ plaintexts
- total complexity: $2^{78}$ steps, which is four times faster than brute-force
- can not be extended to cover all 32 rounds: the complexity would be in excess of exhaustive search
- the attacks are independent of the $G$ permutations
- lesson learned: block ciphers designers should not only prove resistance against conventional differential cryptanalysis by a low upper bound on the probability of characteristics, they should provide such proofs for the lower bound as well

# Impossible Differential attack overview

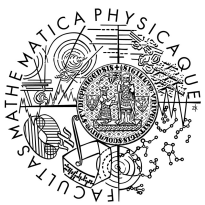
- chosen plaintext attack, which requires a pool of $2^{41}$ plaintexts
- total complexity: $2^{78}$ steps, which is four times faster than brute-force
- can not be extended to cover all 32 rounds: the complexity would be in excess of exhaustive search
- the attacks are independent of the $G$ permutations
- lesson learned: block ciphers designers should not only prove resistance against conventional differential cryptanalysis by a low upper bound on the probability of characteristics, they should provide such proofs for the lower bound as well

# Impossible Differential attack overview

- chosen plaintext attack, which requires a pool of $2^{41}$ plaintexts
- total complexity: $2^{78}$ steps, which is four times faster than brute-force
- can not be extended to cover all 32 rounds: the complexity would be in excess of exhaustive search
- the attacks are independent of the $G$ permutations
- lesson learned: block ciphers designers should not only prove resistance against conventional differential cryptanalysis by a low upper bound on the probability of characteristics, they should provide such proofs for the lower bound as well

# Impossible Differential attack overview

- chosen plaintext attack, which requires a pool of $2^{41}$ plaintexts
- total complexity: $2^{78}$ steps, which is four times faster than brute-force
- can not be extended to cover all 32 rounds: the complexity would be in excess of exhaustive search
- the attacks are independent of the $G$ permutations
- lesson learned: block ciphers designers should not only prove resistance against conventional differential cryptanalysis by a low upper bound on the probability of characteristics, they should provide such proofs for the lower bound as well



# Thanks

Thanks for your attention!
Let's have a lunch :-)