Integral Resistance and Degree Bounds for Complex Linear Layers

Application to PRINCE and Lower-Latency Alternatives

Simon Gerhalter Maria Eichlseder

SAC 2026 - Ottawa

› isec.tugraz.at

Background

- Integral cryptanalysis one of the main cryptanalysis techniques

Background

- Integral cryptanalysis one of the main cryptanalysis techniques
- Difficult to give guarantees against integral distinguisher

Background

- Integral cryptanalysis one of the main cryptanalysis techniques
- Difficult to give guarantees against integral distinguisher
  - Integral resistance provides strong arguments against integral distinguisher

Background

- Integral cryptanalysis one of the main cryptanalysis techniques
- Difficult to give guarantees against integral distinguisher
  - Integral resistance provides strong arguments against integral distinguisher
- Hard to show integral resistance for ciphers with large linear layers

Background

- Integral cryptanalysis one of the main cryptanalysis techniques
- Difficult to give guarantees against integral distinguisher
  - Integral resistance provides strong arguments against integral distinguisher
- Hard to show integral resistance for ciphers with large linear layers
  - Zeng and Tian [ZT25] introduce minimal transition method

Background

- Integral cryptanalysis one of the main cryptanalysis techniques
- Difficult to give guarantees against integral distinguisher
  - Integral resistance provides strong arguments against integral distinguisher
- Hard to show integral resistance for ciphers with large linear layers
  - Zeng and Tian [ZT25] introduce minimal transition method
- We generalize this idea and provide some theoretical foundations

Integral Cryptanalysis Details

- Given an input set, sum over the corresponding outputs

Integral Cryptanalysis Details

- Given an input set, sum over the corresponding outputs
- This sum is constant regardless of used keys

Integral Cryptanalysis Details

- Given an input set, sum over the corresponding outputs
- This sum is constant regardless of used keys
- Integral cryptanalysis linked to the algebraic representation of a cipher

Integral Cryptanalysis Details

- Given an input set, sum over the corresponding outputs
- This sum is constant regardless of used keys
- Integral cryptanalysis linked to the algebraic representation of a cipher
- Algebraic normal form (ANF) of output coordinate *i*

$$E_k^{(i)}(\boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}^{(i)}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}} = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n, \boldsymbol{v} \in \mathbb{F}_2^m} a_{\boldsymbol{u},\boldsymbol{v}}^{(i)} \cdot \boldsymbol{k}^{\boldsymbol{v}} \boldsymbol{x}^{\boldsymbol{u}}$$

Integral Cryptanalysis Details

- Given an input set, sum over the corresponding outputs
- This sum is constant regardless of used keys
- Integral cryptanalysis linked to the algebraic representation of a cipher
- Algebraic normal form (ANF) of output coordinate *i*

$$E_k^{(i)}(\boldsymbol{x}) = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}^{(i)}(\boldsymbol{k}) \cdot \boldsymbol{x}^{\boldsymbol{u}} = \sum_{\boldsymbol{u} \in \mathbb{F}_2^n, \boldsymbol{v} \in \mathbb{F}_2^m} a_{\boldsymbol{u},\boldsymbol{v}}^{(i)} \cdot \boldsymbol{k}^{\boldsymbol{v}} \boldsymbol{x}^{\boldsymbol{u}}$$

- Example: Xor

$$E(\boldsymbol{x}) = 0 \cdot x^{00} + \mathbf{1} \cdot x^{01} + \mathbf{1} \cdot x^{10} + 0 \cdot x^{11} = x_1 + x_2$$

Integral Cryptanalysis Details

- Binary möbius transform allows us to calculate coefficient $a_{\boldsymbol{u}}$

$$a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \preceq \boldsymbol{u}} f(\boldsymbol{x})$$

Integral Cryptanalysis Details

- Binary möbius transform allows us to calculate coefficient $a_{\boldsymbol{u}}$

$$a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \preceq \boldsymbol{u}} f(\boldsymbol{x})$$

Example: Xor

x_1	x_2	$f(x)$
0	0	0
0	1	1
1	0	1
1	1	0

Integral Cryptanalysis Details

- Binary möbius transform allows us to calculate coefficient $a_{\boldsymbol{u}}$

$$a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \preceq \boldsymbol{u}} f(\boldsymbol{x})$$

Example: Xor

x_1	x_2	$f(x)$
0	0	0
0	1	1
1	0	1
1	1	0

$$a_{01} = f(00) + f(01) = 0 + 1 = 1$$

Integral Cryptanalysis Details

- Binary möbius transform allows us to calculate coefficient $a_{\boldsymbol{u}}$

$$a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \preceq \boldsymbol{u}} f(\boldsymbol{x})$$

Example: Xor

x_1	x_2	$f(x)$
0	0	0
0	1	1
1	0	1
1	1	0

$$a_{01} = f(00) + f(01) = 0 + 1 = 1$$
$$a_{11} = f(00) + f(01) + f(10) + f(11)$$
$$= 0 + 1 + 1 + 0 = 0$$

Integral Cryptanalysis Details

- Binary möbius transform allows us to calculate coefficient $a_{\boldsymbol{u}}$

$$a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \preceq \boldsymbol{u}} f(\boldsymbol{x})$$

- When $a_{\boldsymbol{u}} = 0$, saturating the input space $\boldsymbol{u}$ leads to an integral property

Example: Xor

x_1	x_2	$f(x)$
0	0	0
0	1	1
1	0	1
1	1	0

$$a_{01} = f(00) + f(01) = 0 + 1 = 1$$
$$a_{11} = f(00) + f(01) + f(10) + f(11)$$
$$= 0 + 1 + 1 + 0 = 0$$

Integral Cryptanalysis Details

- Binary möbius transform allows us to calculate coefficient $a_{\boldsymbol{u}}$

$$a_{\boldsymbol{u}} = \sum_{\boldsymbol{x} \preceq \boldsymbol{u}} f(\boldsymbol{x})$$

- When $a_{\boldsymbol{u}} = 0$, saturating the input space $\boldsymbol{u}$ leads to an integral property
- Degree d of function
  - $d = \max_{\boldsymbol{u}} hw(\boldsymbol{u})$ where $a_{\boldsymbol{u}} = 1$

Example: Xor

x_1	x_2	$f(x)$
0	0	0
0	1	1
1	0	1
1	1	0

$$a_{01} = f(00) + f(01) = 0 + 1 = 1$$
$$a_{11} = f(00) + f(01) + f(10) + f(11)$$
$$= 0 + 1 + 1 + 0 = 0$$

Integral Resistance Property

- Notion introduced by Hebborn et al. [Heb+21]

Integral Resistance Property

- Notion introduced by Hebborn et al. [Heb+21]
- With the $n-1$ degree input monomials ($u = \bar{\boldsymbol{e}}_1 = 011\ldots1$) we construct

$$\mathcal{I}(E) = \begin{pmatrix} a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_1} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_1} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_1} \\ a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_2} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_2} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_s} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_s} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_s} \end{pmatrix}$$

Integral Resistance Property

- Notion introduced by Hebborn et al. [Heb+21]
- With the $n-1$ degree input monomials ($u = \bar{\boldsymbol{e}}_1 = 011\ldots1$) we construct

$$\mathcal{I}(E) = \begin{pmatrix} a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_1} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_1} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_1} \\ a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_2} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_2} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_s} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_s} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_s} \end{pmatrix}$$

- If $\mathcal{I}(E)$ is full rank
  - No linear combination of input/output can lead to integral distinguisher

Integral Resistance Property

- Notion introduced by Hebborn et al. [Heb+21]
- With the $n-1$ degree input monomials ($u = \bar{\boldsymbol{e}}_1 = 011\ldots1$) we construct

$$\mathcal{I}(E) = \begin{pmatrix} a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_1} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_1} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_1} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_1} \\ a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_2} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_2} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_2} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a^{(1)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & a^{(2)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_1,\boldsymbol{v}_s} & a^{(1)}_{\bar{\boldsymbol{e}}_2,\boldsymbol{v}_s} & \cdots & a^{(j)}_{\bar{\boldsymbol{e}}_i,\boldsymbol{v}_s} & \cdots & a^{(n)}_{\bar{\boldsymbol{e}}_n,\boldsymbol{v}_s} \end{pmatrix}$$

- If $\mathcal{I}(E)$ is full rank
  - No linear combination of input/output can lead to integral distinguisher
  - With key whitening statement holds for any input set

Too Many Trails

- How do we determine the coefficients?

Too Many Trails

- How do we determine the coefficients?
  - Use monomial prediction

Too Many Trails

- How do we determine the coefficients?
  - Use monomial prediction
    - Pre-calculate monomial transitions of cipher components (MPT)

Table: Monomial Prediction Table (MPT) of an XOR

u \ v	0	1
0	×	·
1	·	×
2	·	×
3	·	·

Too Many Trails

- How do we determine the coefficients?
  - Use monomial prediction
    - Pre-calculate monomial transitions of cipher components (MPT)
    - Use theory of trails to connect cipher components

Table: Monomial Prediction Table (MPT) of an XOR

u \ v	0	1
0	×	·
1	·	×
2	·	×
3	·	·

Too Many Trails

- How do we determine the coefficients?
  - Use monomial prediction
    - Pre-calculate monomial transitions of cipher components (MPT)
    - Use theory of trails to connect cipher components
- Challenges
  - Calculation of MPT for large linear layers infeasible

Table: Monomial Prediction Table (MPT) of an XOR

u \ v	0	1
0	×	·
1	·	×
2	·	×
3	·	·

Too Many Trails

- How do we determine the coefficients?
  - Use monomial prediction
    - Pre-calculate monomial transitions of cipher components (MPT)
    - Use theory of trails to connect cipher components
- Challenges
  - Calculation of MPT for large linear layers infeasible
  - Since we are in $\mathbb{F}_2$, an even number of trails cancels out
    - Enumerate trails

Table: Monomial Prediction Table (MPT) of an XOR

u \ v	0	1
0	×	·
1	·	×
2	·	×
3	·	·

Too Many Trails

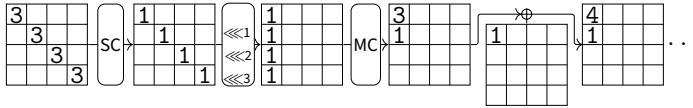
- How do we determine the coefficients?
  - Use monomial prediction
    - Pre-calculate monomial transitions of cipher components (MPT)
    - Use theory of trails to connect cipher components
- Challenges
  - Calculation of MPT for large linear layers infeasible
  - Since we are in $\mathbb{F}_2$, an even number of trails cancels out
    - Enumerate trails
    - Large number of intermediate monomials leading to spurious trails

Table: Monomial Prediction Table (MPT) of an XOR

u \ v	0	1
0	×	·
1	·	×
2	·	×
3	·	·

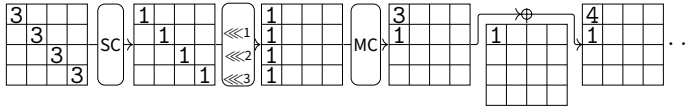
Minimize Transitions

- Zeng and Tian [ZT25] chose key variables in monomial deliberately



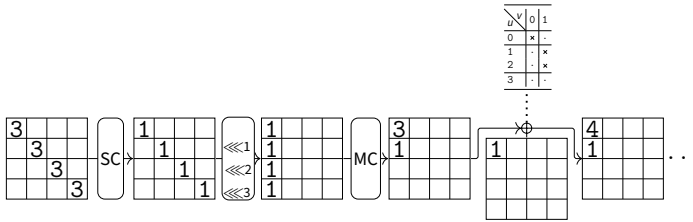
Minimize Transitions

- Zeng and Tian [ZT25] chose key variables in monomial deliberately
  - Each intermediate monomial can only have a certain degree



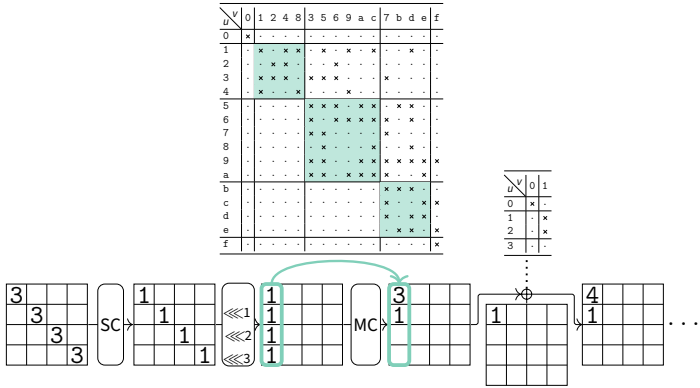
Minimize Transitions

- Zeng and Tian [ZT25] chose key variables in monomial deliberately
  - Each intermediate monomial can only have a certain degree



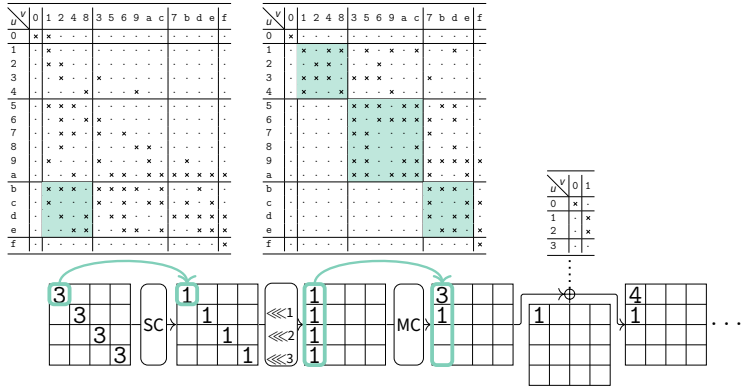
Minimize Transitions

- Zeng and Tian [ZT25] chose key variables in monomial deliberately
  - Each intermediate monomial can only have a certain degree



Minimize Transitions

- Zeng and Tian [ZT25] chose key variables in monomial deliberately
  - Each intermediate monomial can only have a certain degree



S-Box Requirements

Complete S-Box [ZT25]

If the monomial $\boldsymbol{x}^{\boldsymbol{u}}$ appears in the ANF of some output bit y_i for every vector $\boldsymbol{u}$ with $hw(\boldsymbol{u}) = n - 1$, then S is called a complete S-box.

u \ v	0	1	2	4	8	3	5	6	9	a	c	7	b	d	e	f
0	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1	·	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·
2	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·
3	·	·	×	·	·	×	·	·	·	·	·	·	·	·	·	·
4	·	·	·	·	×	·	·	·	×	·	·	·	·	·	·	·
5	·	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·
6	·	·	×	·	×	×	·	·	·	·	·	·	·	·	·	·
7	·	·	×	×	·	×	·	×	·	·	·	·	·	·	·	·
8	·	·	×	·	·	·	·	·	×	×	·	·	·	·	·	·
9	·	×	·	·	·	×	·	·	·	×	·	·	×	·	·	·
a	·	·	·	×	·	·	×	×	·	×	×	×	×	×	×	×
b	·	×	×	×	·	×	×	×	×	·	×	·	·	×	·	·
c	·	×	·	·	·	×	·	×	·	×	×	·	×	·	×	·
d	·	·	×	·	×	·	×	×	·	·	·	×	×	×	×	×
e	·	·	·	×	×	·	·	×	×	·	×	·	·	×	×	×
f	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×

S-Box Requirements

Complete S-Box [ZT25]

If the monomial $\boldsymbol{x}^{\boldsymbol{u}}$ appears in the ANF of some output bit y_i for every vector $\boldsymbol{u}$ with $hw(\boldsymbol{u}) = n - 1$, then S is called a complete S-box.

- This is not a sufficient criterion

u \ v	0	1	2	4	8	3	5	6	9	a	c	7	b	d	e	f
0	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1	·	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·
2	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·
3	·	·	×	·	·	×	·	·	·	·	·	·	·	·	·	·
4	·	·	·	·	×	·	·	·	×	·	·	·	·	·	·	·
5	·	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·
6	·	·	×	·	×	×	·	·	·	·	·	·	·	·	·	·
7	·	·	×	×	·	×	·	×	·	·	·	·	·	·	·	·
8	·	·	×	·	·	·	·	·	×	×	·	·	·	·	·	·
9	·	×	·	·	·	×	·	·	·	×	·	·	×	·	·	·
a	·	·	·	×	·	·	×	×	·	×	×	×	×	×	×	×
b	·	×	×	×	·	×	×	×	×	·	×	·	·	×	·	·
c	·	×	·	·	·	×	·	×	·	×	×	·	×	·	×	·
d	·	·	×	·	×	·	×	×	·	·	·	×	×	×	×	×
e	·	·	·	×	×	·	·	×	×	·	×	·	·	×	×	×
f	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×

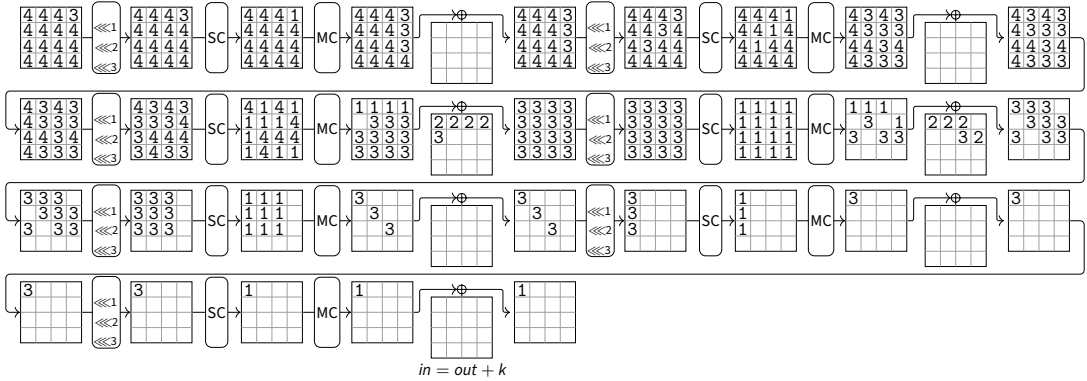
S-Box Requirements

Max-degree full-rank S-Box

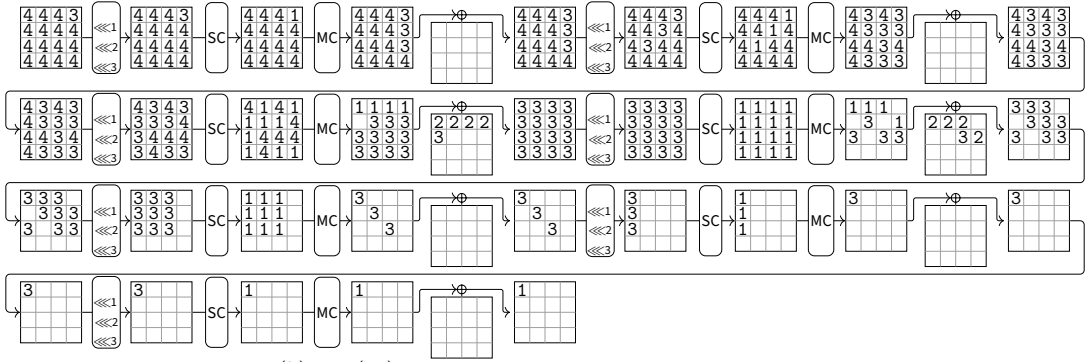
A S-Box is max-degree full-rank iff the submatrix of the MPT encompassing all transitions $\boldsymbol{x}^{\boldsymbol{u}}$ to $\boldsymbol{y}^{\boldsymbol{w}}$, with $hw(\boldsymbol{u}) = n - 1$ and $hw(\boldsymbol{w}) = 1$, is full rank.

u \ v	0	1	2	4	8	3	5	6	9	a	c	7	b	d	e	f
0	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1	·	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·
2	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·
3	·	·	×	·	·	×	·	·	·	·	·	·	·	·	·	·
4	·	·	·	·	×	·	·	·	×	·	·	·	·	·	·	·
5	·	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·
6	·	·	×	·	×	×	·	·	·	·	·	·	·	·	·	·
7	·	·	×	×	·	×	·	×	·	·	·	·	·	·	·	·
8	·	·	×	·	·	·	·	·	×	×	·	·	·	·	·	·
9	·	×	·	·	·	×	·	·	·	×	·	·	×	·	·	·
a	·	·	·	×	·	·	×	×	·	×	×	×	×	×	×	×
b	·	×	×	×	·	×	×	×	×	·	×	·	·	×	·	·
c	·	×	·	·	·	×	·	×	·	×	×	·	×	·	×	·
d	·	·	×	·	×	·	×	×	·	·	·	×	×	×	×	×
e	·	·	·	×	×	·	·	×	×	·	×	·	·	×	×	×
f	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×

Degree Abstraction Model

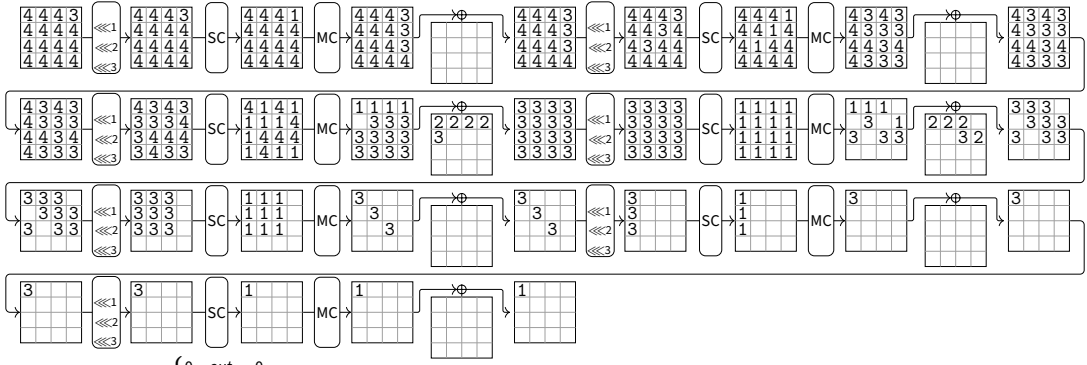


Degree Abstraction Model



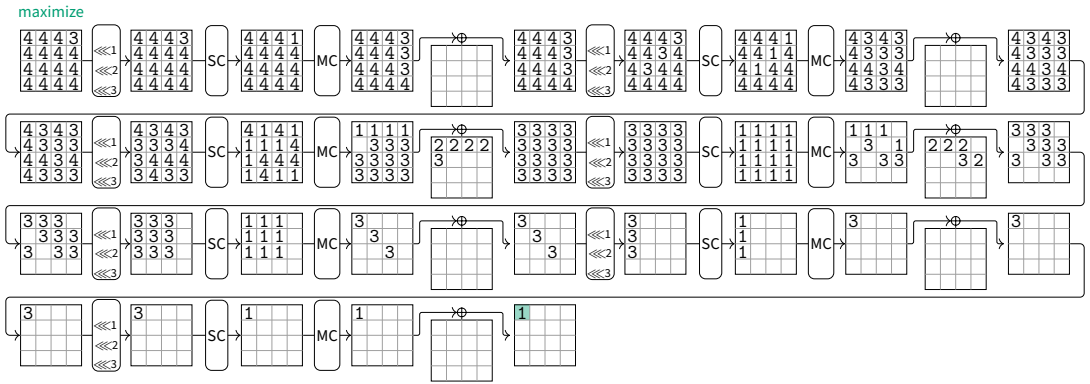
$sum(in) \leq sum(out)$

Degree Abstraction Model

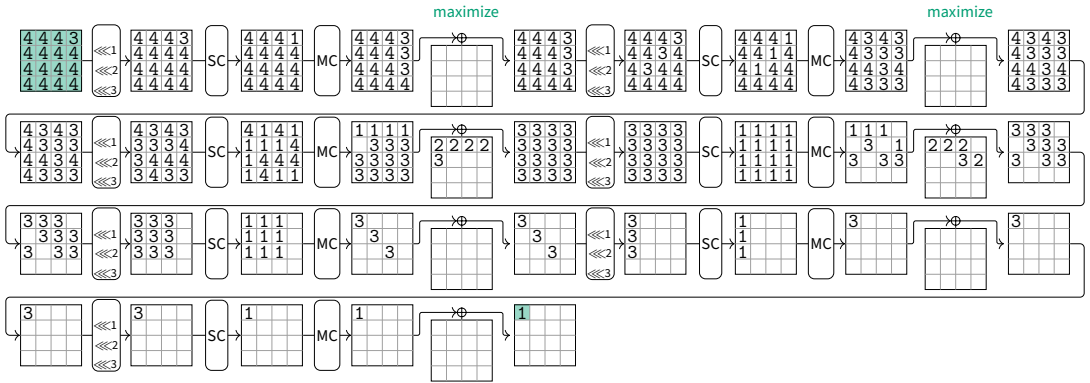


$$in = \begin{cases} 0 & out = 0 \\ 4 & out = 4 \\ 3 & out = 1 \end{cases}$$

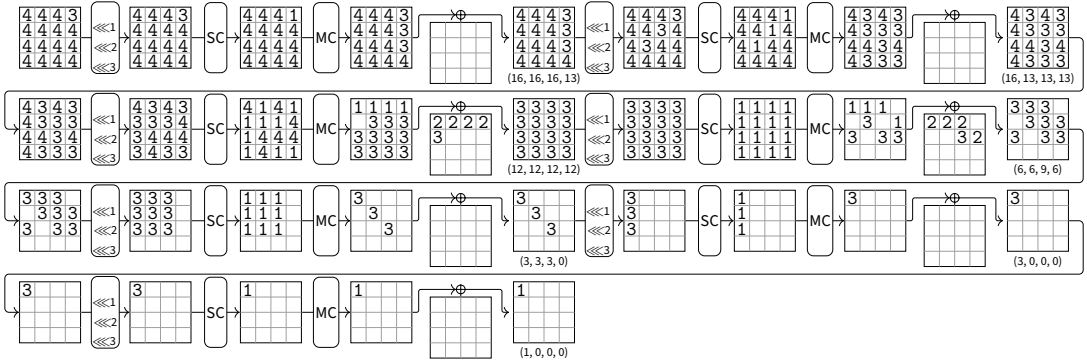
Degree Abstraction Model



Degree Abstraction Model



Degree Abstraction Model



Degree Abstraction Results

Cipher	Model	1	2	3	4	5	6	7
PRINCE like	Cell-Based	3	9	27	49	57	61	63
PRINCE with 64-bit MC	Cell-Based	3	9	27	51	59	62	**63**
Rijndael-256	Bound [BC13]	7	28	113	235	250	254	255
	Cell-Based	7	28	**112**	**232**	**249**	254	255

Degree Abstraction Results

Cipher	Model	1	2	3	4	5	6	7
PRINCE like	Cell-Based	3	9	27	49	57	61	63
PRINCE with 64-bit MC	Cell-Based	3	9	27	51	59	62	**63**
Rijndael-256	Bound [BC13]	7	28	113	235	250	254	255
	Cell-Based	7	28	**112**	**232**	**249**	254	255

- General property of PRINCE like ciphers without max-degree full-rank S-Boxes

Degree Abstraction Results

Cipher	Model	1	2	3	4	5	6	7
PRINCE like	Cell-Based	3	9	27	49	57	61	63
PRINCE with 64-bit MC	Cell-Based	3	9	27	51	59	62	**63**
Rijndael-256	Bound [BC13]	7	28	113	235	250	254	255
	Cell-Based	7	28	**112**	**232**	**249**	254	255

- General property of PRINCE like ciphers without max-degree full-rank S-Boxes
  - Force degree 3 to 2 transition in first round

Degree Abstraction Results

Cipher	Model	1	2	3	4	5	6	7
PRINCE like	Cell-Based	3	9	27	49	57	61	63
PRINCE with 64-bit MC	Cell-Based	3	9	27	51	59	62	**63**
Rijndael-256	Bound [BC13]	7	28	113	235	250	254	255
	Cell-Based	7	28	**112**	**232**	**249**	254	255

- General property of PRINCE like ciphers without max-degree full-rank S-Boxes
  - Force degree 3 to 2 transition in first round
  - 7 round model becomes unsatisfiable
    - Always 7 round distinguisher for ciphers like LED, uKNIT

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state
   - We introduce degree abstraction model that simplifies this step

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state
   - We introduce degree abstraction model that simplifies this step
2. **Precompute monomial** transitions
   - We use the $\mathcal{ZR}$ [ZR19] method to massively speed up the process

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state
   - We introduce degree abstraction model that simplifies this step
2. **Precompute monomial** transitions
   - We use the $\mathcal{ZR}$ [ZR19] method to massively speed up the process
3. **Propagate monomials** backwards through cipher

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state
   - We introduce degree abstraction model that simplifies this step
2. **Precompute monomial** transitions
   - We use the $\mathcal{ZR}$ [ZR19] method to massively speed up the process
3. **Propagate monomials** backwards through cipher
   - Start with set consisting of single output monomial

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state
   - We introduce degree abstraction model that simplifies this step
2. **Precompute monomial** transitions
   - We use the $\mathcal{ZR}$ [ZR19] method to massively speed up the process
3. **Propagate monomials** backwards through cipher
   - Start with set consisting of single output monomial
   - Each round
     - Apply look-up table to each monomial in set

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state
   - We introduce degree abstraction model that simplifies this step
2. **Precompute monomial** transitions
   - We use the $\mathcal{ZR}$ [ZR19] method to massively speed up the process
3. **Propagate monomials** backwards through cipher
   - Start with set consisting of single output monomial
   - Each round
     - Apply look-up table to each monomial in set
     - Remove parity 0 monomials

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state
   - We introduce degree abstraction model that simplifies this step
2. **Precompute monomial** transitions
   - We use the $\mathcal{ZR}$ [ZR19] method to massively speed up the process
3. **Propagate monomials** backwards through cipher
   - Start with set consisting of single output monomial
   - Each round
     - Apply look-up table to each monomial in set
     - Remove parity 0 monomials
     - Choose key variables such that they follow degree abstraction

Minimal Transitions Method: 3 Step approach

1. Find **unique degree** contained in each state
   - We introduce degree abstraction model that simplifies this step
2. **Precompute monomial** transitions
   - We use the $\mathcal{ZR}$ [ZR19] method to massively speed up the process
3. **Propagate monomials** backwards through cipher
   - Start with set consisting of single output monomial
   - Each round
     - Apply look-up table to each monomial in set
     - Remove parity 0 monomials
     - Choose key variables such that they follow degree abstraction
   - At input we get all $n - 1$ degree monomials contained in the ANF of output

Integral Resistance of PRINCE

- Properties of PRINCE
  - 7 round integral resistance

Integral Resistance of PRINCE

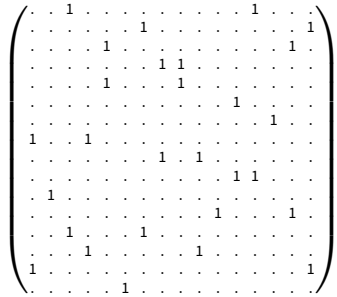
- Properties of PRINCE
  - 7 round integral resistance
  - Reduce runtime by a factor of 4, since some outputs lead to same set of input monomials

Integral Resistance of PRINCE

- Properties of PRINCE
  - 7 round integral resistance
  - Reduce runtime by a factor of 4, since some outputs lead to same set of input monomials
  - Takes a diverse set of key variables to obtain full-rank integral resistance matrix
    - Choose key variables random

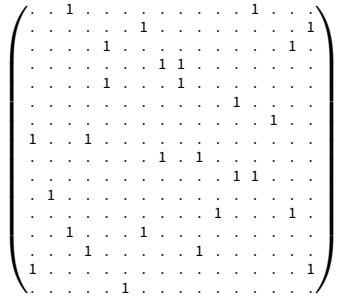
Integral Resistance of PRINCE

- Properties of PRINCE
  - 7 round integral resistance
  - Reduce runtime by a factor of 4, since some outputs lead to same set of input monomials
  - Takes a diverse set of key variables to obtain full-rank integral resistance matrix
    - Choose key variables random
- We use MILP model to find lower-latency MixColumns
  - Retains 7 round integral resistance



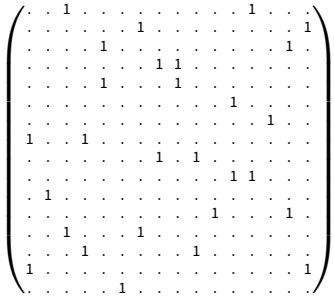
Integral Resistance of PRINCE

- Properties of PRINCE
  - 7 round integral resistance
  - Reduce runtime by a factor of 4, since some outputs lead to same set of input monomials
  - Takes a diverse set of key variables to obtain full-rank integral resistance matrix
    - Choose key variables random
- We use MILP model to find lower-latency MixColumns
  - Retains 7 round integral resistance
  - Useful for ciphers like Orthros



Integral Resistance of PRINCE

- Properties of PRINCE
  - 7 round integral resistance
  - Reduce runtime by a factor of 4, since some outputs lead to same set of input monomials
  - Takes a diverse set of key variables to obtain full-rank integral resistance matrix
    - Choose key variables random
- We use MILP model to find lower-latency MixColumns
  - Retains 7 round integral resistance
  - Useful for ciphers like Orthros
  - Allows us to verify correctness of approach with trail counting



Conclusion

- Integral Resistance provides strong arguments against integral distinguishers

Conclusion

- Integral Resistance provides strong arguments against integral distinguishers
- We are able to show this for ciphers with large linear layers

Conclusion

- Integral Resistance provides strong arguments against integral distinguishers
- We are able to show this for ciphers with large linear layers
  - 7 round integral resistance of PRINCE and 6 rounds of Beanie

Conclusion

- Integral Resistance provides strong arguments against integral distinguishers
- We are able to show this for ciphers with large linear layers
  - 7 round integral resistance of PRINCE and 6 rounds of Beanie
  - Our framework can be easily adapted for new ciphers

Conclusion

- Integral Resistance provides strong arguments against integral distinguishers
- We are able to show this for ciphers with large linear layers
  - 7 round integral resistance of PRINCE and 6 rounds of Beanie
  - Our framework can be easily adapted for new ciphers
- Degree abstraction model leads to new results

Conclusion

- Integral Resistance provides strong arguments against integral distinguishers
- We are able to show this for ciphers with large linear layers
  - 7 round integral resistance of PRINCE and 6 rounds of Beanie
  - Our framework can be easily adapted for new ciphers
- Degree abstraction model leads to new results

simon.gerhalter@tugraz.at

</> `https://extgit.isec.tugraz.at/castle/tool/intres`

Acknowledgments

This research was funded in whole by the Austrian Science Fund (FWF) SFB project SPyCoDe (10.55776/F85).

Bibliography I

[BC13] Christina Boura and Anne Canteaut. **On the Influence of the Algebraic Degree of F^{-1} on the Algebraic Degree of $G \circ F$.** **IEEE Trans. Inf. Theory** 59.1 (2013), pp. 691–702. DOI: 10.1109/TIT.2012.2214203.

[Heb+21] Phil Hebborn et al. **Strong and Tight Security Guarantees Against Integral Distinguishers.** ASIACRYPT 2021. Vol. 13090. LNCS. Springer, 2021, pp. 362–391. DOI: 10.1007/978-3-030-92062-3_13.

[ZR19] Wenying Zhang and Vincent Rijmen. **Division cryptanalysis of block ciphers with a binary diffusion layer.** **IET Inf. Secur.** 13.2 (2019), pp. 87–95. DOI: 10.1049/IET-IFS.2018.5151.

[ZT25] Fanyang Zeng and Tian Tian. **A New Method for Constructing Integral-Resistance Matrix for 5-Round AES.** **IET Inf. Secur.** 2025.1 (2025). DOI: 10.1049/ise2/3447652.