# Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers

**Zejun Xiang** Wentao Zhang Zhenzhen Bao Dongdai Lin

Institute of Information Engineering, CAS, Beijing, China

December 7, 2016. Hanoi

# Overview

# Preliminary

**Definition (Bit-Product Function [Todo, EUROCRYPT 2015])**

For any fixed $\mathbf{u} \in (\mathbb{F}_2^{n_0} \times \mathbb{F}_2^{n_1} \times \cdots \times \mathbb{F}_2^{n_{m-1}})$,

$$\begin{aligned}\pi_{\mathbf{u}}(\mathbf{x}) : (\mathbb{F}_2^{n_0} \times \mathbb{F}_2^{n_1} \times \cdots \times \mathbb{F}_2^{n_{m-1}}) & \longrightarrow \mathbb{F}_2 \\ (x_0, x_1, \cdots, x_{m-1}) & \longmapsto \prod_{i=0}^{m-1} ( \prod_{j=0}^{n_i-1} x_i[j]^{u_i[j]} )\end{aligned}$$

# Preliminary

## Definition (Bit-Product Function [Todo, EUROCRYPT 2015])

For any fixed $\mathbf{u} \in (\mathbb{F}_2^{n_0} \times \mathbb{F}_2^{n_1} \times \cdots \times \mathbb{F}_2^{n_{m-1}})$,

$$\pi_{\mathbf{u}}(\mathbf{x}) : (\mathbb{F}_2^{n_0} \times \mathbb{F}_2^{n_1} \times \cdots \times \mathbb{F}_2^{n_{m-1}}) \longrightarrow \mathbb{F}_2$$

$$(x_0, x_1, \cdots, x_{m-1}) \longmapsto \prod_{i=0}^{m-1} \Big( \prod_{j=0}^{n_i-1} x_i[j]^{u_i[j]} \Big)$$

## Example: $n = 4, m = 2$

$$\mathbf{u} = (u_0^0 || u_0^1 || u_0^2 || u_0^3, u_1^0 || u_1^1 || u_1^2 || u_1^3) = (0||1||1||0, 1||0||1||1)$$

$$\mathbf{x} = (x_0^0 || x_0^1 || x_0^2 || x_0^3, x_1^0 || x_1^1 || x_1^2 || x_1^3) = (0||1||1||1, 1||1||0||1)$$

$$\pi_u(x) = (0^0 1^1 1^1 1^0)(1^1 1^0 0^1 1^1) = 0$$

# Preliminary

### Definition (Bit-Product Function [Todo, EUROCRYPT 2015])

For any fixed $\mathbf{u} \in (\mathbb{F}_2^{n_0} \times \mathbb{F}_2^{n_1} \times \cdots \times \mathbb{F}_2^{n_{m-1}})$,

$$\pi_{\mathbf{u}}(\mathbf{x}) : (\mathbb{F}_2^{n_0} \times \mathbb{F}_2^{n_1} \times \cdots \times \mathbb{F}_2^{n_{m-1}}) \longrightarrow \mathbb{F}_2$$

$$(x_0, x_1, \cdots, x_{m-1}) \longmapsto \prod_{i=0}^{m-1} \Big( \prod_{j=0}^{n_i-1} x_i[j]^{u_i[j]} \Big)$$

### Example: $n = 4, m = 2$

$$\mathbf{u} = (u_0^0 || u_0^1 || u_0^2 || u_0^3, u_1^0 || u_1^1 || u_1^2 || u_1^3) = (0||1||1||0, 1||0||1||1)$$

$$\mathbf{x} = (x_0^0 || x_0^1 || x_0^2 || x_0^3, x_1^0 || x_1^1 || x_1^2 || x_1^3) = (0||1||1||1, 1||1||0||1)$$

$$\pi_u(x) = (0^0 1^1 1^1 1^0)(1^1 1^0 0^1 1^1) = 0$$

### Definition ([Todo, EUROCRYPT 2015])

Define $\mathbf{k} \succeq \mathbf{k}^*$ if $k_i \geq k_i^*$ holds for all $i = 0, 1, \cdots, m-1$. Otherwise we denote $\mathbf{k} \not\succeq \mathbf{k}^*$.

# Definition

Division Property is introduced by Todo at EUROCRYPT 2015, it's a generalized integral property.

# Definition

Division Property is introduced by Todo at EUROCRYPT 2015, it's a generalized integral property.

**Definition (Division Property [Todo, EUROCRYPT 2015])**

Let $\mathbb{X} \subset (\mathbb{F}_2^n)^m$, and $\mathbf{k^{(i)}} \in \{0, 1, \cdots, n\}^m$. $\mathbb{X}$ has the division property $\mathcal{D}^{n,m}_{\mathbf{k^{(0)}}, \mathbf{k^{(1)}}, \cdots, \mathbf{k^{(q-1)}}}$, if $\sum_{\mathbf{x} \in \mathbb{X}} \pi_{\mathbf{u}}(\mathbf{x}) = 0$ for any

$$\mathbf{u} \in \left\{ (u_0, u_1, \cdots, u_{m-1}) \in (\mathbb{F}_2^n)^m | W(\mathbf{u}) \nsucceq \mathbf{k^{(0)}}, \cdots, W(\mathbf{u}) \nsucceq \mathbf{k^{(q-1)}} \right\},$$

among which, $W(\mathbf{u}) = (\mathrm{wt}(u_0), \mathrm{wt}(u_1), \cdots, \mathrm{wt}(u_{m-1}))$ .

# Using Division Property

# Using Division Property

1. Construct an input set with division property $\mathcal{D}^{n,m}_{\mathbb{K}_0}$.

# Using Division Property

1. Construct an input set with division property $\mathcal{D}^{n,m}_{\mathbb{K}_0}$.
2. Propagate the initial division property $r$ rounds to get the dividion property of r-round output $\mathcal{D}^{n,m}_{\mathbb{K}_r}$.

# Using Division Property

1. Construct an input set with division property $\mathcal{D}^{n,m}_{\mathbb{K}_0}$.
2. Propagate the initial division property $r$ rounds to get the dividion property of r-round output $\mathcal{D}^{n,m}_{\mathbb{K}_r}$.
3. Extract some useful integral property from $\mathcal{D}^{n,m}_{\mathbb{K}_r}$.

# Propagations of Division Property

# Propagations of Division Property

**Copy**
[Todo, CRYPTO 2015]

| | | |
|---|---|---|
| $\mathbb{F}_2^n$ | $\longrightarrow$ | $\mathbb{F}_2^n \times \mathbb{F}_2^n$ |
| $x$ | $\longmapsto$ | $(x, x)$ |
| $\mathbb{X}$ | $\longmapsto$ | $\mathrm{Copy}(\mathbb{X})$ |
| $\mathcal{D}_k^n$ | $\longmapsto$ | $\mathcal{D}^{2,n}_{(0,k),(1,k-1),\cdots,(k,0)}$ |

**Xor**
[Todo, CRYPTO 2015]

| | | |
|---|---|---|
| $\mathbb{F}_2^n \times \mathbb{F}_2^n$ | $\longrightarrow$ | $\mathbb{F}_2^n$ |
| $(x_0, x_1)$ | $\longmapsto$ | $x_0 \oplus x_1$ |
| $\mathbb{X}$ | $\longmapsto$ | $\mathrm{Xor}(\mathbb{X})$ |
| $\mathcal{D}^n_{(k_0,k_1)}$ | $\longmapsto$ | $\mathcal{D}^n_{k_0+k_1}$ |

**And**
[Xiang, IWSEC 2016]

| | | |
|---|---|---|
| $\mathbb{F}_2^n \times \mathbb{F}_2^n$ | $\longrightarrow$ | $\mathbb{F}_2^n$ |
| $(x_0, x_1)$ | $\longmapsto$ | $x_0 \& x_1$ |
| $\mathbb{X}$ | $\longmapsto$ | $\mathrm{And}(\mathbb{X})$ |
| $\mathcal{D}^n_{(k_0,k_1)}$ | $\longmapsto$ | $\mathcal{D}^n_{\max(k_0,k_1)}$ |

# Bit-based Division Property

- The division property is defined and computed on $(\mathbb{F}_2^n)^m$. If $n = 1$, this is the bit-based division property [Todo, FSE 2016].

# Bit-based Division Property

- The division property is defined and computed on $(\mathbb{F}_2^n)^m$. If $n = 1$, this is the bit-based division property [Todo, FSE 2016].

**Advantages**

Detailed division property

Longer distinguishers

Better results.

# Bit-based Division Property

- The division property is defined and computed on $(\mathbb{F}_2^n)^m$. If $n = 1$, this is the bit-based division property [Todo, FSE 2016].

**Advantages**

Detailed division property

Longer distinguishers

Better results.

**Disadvantages**

More computation

Upper bounded by $O(2^n)$

Only small size cipher.

# Bit-based Division Property

- The division property is defined and computed on $(\mathbb{F}_2^n)^m$. If $n = 1$, this is the bit-based division property [Todo, FSE 2016].

**Advantages**

Detailed division property

Longer distinguishers

Better results.

**Disadvantages**

More computation

Upper bounded by $O(2^n)$

Only small size cipher.

How to compute bit-based division property efficiently?

# Basic Strategy

We will use Mixed Integer Linear Programming (MILP) method to characterize the division property propagations.

# Basic Strategy

We will use Mixed Integer Linear Programming (MILP) method to characterize the division property propagations.

**Mixed Integer Linear Programming, MILP**

$$\text{Minimize or (Maximize)} : a^T \cdot x$$

$$\text{Subject To} : Mx >= 0$$

part of or all the variables in $x$ are restricted in integers.

# Basic Strategy

Two issues need to be addressed:

# Basic Strategy

Two issues need to be addressed:

1. Describe the division propagations by linear (in)equalities.

## Basic Strategy

Two issues need to be addressed:

1. Describe the division propagations by linear (in)equalities.
2. Convert search problem to estimate the minimal value of the objective function.

# Division Trail

**Definition (Division Trail)**

Assume the input set to the block cipher has initial division property $\mathcal{D}_{\mathbf{k}}^{n,m}$, and denote the division property after $i$-round encryption by $\mathcal{D}_{\mathbb{K}_i}^{n,m}$. Thus, we have the following chain of division property propagations:

$$\{\mathbf{k}\} \stackrel{def}{=} \mathbb{K}_0 \xrightarrow{f_r} \mathbb{K}_1 \xrightarrow{f_r} \mathbb{K}_2 \xrightarrow{f_r} \cdots$$

For $(\mathbf{k}_0, \mathbf{k}_1, \cdots, \mathbf{k}_r) \in \mathbb{K}_0 \times \mathbb{K}_1 \times \cdots \times \mathbb{K}_r$, if $\mathbf{k}_{i-1}$ can propagate to $\mathbf{k}_i$ for all $i \in \{1, 2, \cdots, r\}$ by propagation rules, we call $(\mathbf{k}_0, \mathbf{k}_1, \cdots, \mathbf{k}_r)$ an $r$-round **division trail**.

# Division Trail

**Definition (Division Trail)**

Assume the input set to the block cipher has initial division property $\mathcal{D}_{\mathbf{k}}^{n,m}$, and denote the division property after $i$-round encryption by $\mathcal{D}_{\mathbb{K}_i}^{n,m}$. Thus, we have the following chain of division property propagations:

$$\{\mathbf{k}\} \stackrel{def}{=} \mathbb{K}_0 \xrightarrow{f_r} \mathbb{K}_1 \xrightarrow{f_r} \mathbb{K}_2 \xrightarrow{f_r} \cdots$$

For $(\mathbf{k}_0, \mathbf{k}_1, \cdots, \mathbf{k}_r) \in \mathbb{K}_0 \times \mathbb{K}_1 \times \cdots \times \mathbb{K}_r$, if $\mathbf{k}_{i-1}$ can propagate to $\mathbf{k}_i$ for all $i \in \{1, 2, \cdots, r\}$ by propagation rules, we call $(\mathbf{k}_0, \mathbf{k}_1, \cdots, \mathbf{k}_r)$ an $r$-round **division trail**.

**Proposition**

*The set of the last vectors of all $r$-round division trails which start with $\mathbf{k}$ is equal to $\mathbb{K}_r$.*

# Set without Integral Property

**Proposition (Set without Integral Property)**

*Assume $\mathbb{X}$ is a set with division property $\mathcal{D}_{\mathbb{K}}^{1,n}$, then $\mathbb{X}$ does not have integral property if and only if $\mathbb{K}$ contains all the $n$ unit vectors.*

# Set without Integral Property

**Proposition (Set without Integral Property)**

*Assume $\mathbb{X}$ is a set with division property $\mathcal{D}_{\mathbb{K}}^{1,n}$, then $\mathbb{X}$ does not have integral property if and only if $\mathbb{K}$ contains all the $n$ unit vectors.*

Given initial division property $\mathcal{D}_{\mathbf{k}}^{n,m}$ and round number $r$, there doesn't exist $r$-round distinguisher if and only if there exists $n$ division trails which start with the initial division property and ends up with the $n$ different unit vectors.

# Basic Strategy

Two issues need to be addressed:

1. Describe the division propagations by linear (in)equalities.
2. Convert search problem to estimate the minimal value of the objective function.

# Modeling Copy

# Modeling Copy

General Rule: $\mathcal{D}_k^n \longmapsto \mathcal{D}^{2,n}_{(0,k),(1,k-1),\cdots,(k,0)}$.

# Modeling Copy

General Rule: $\mathcal{D}_k^n \longmapsto \mathcal{D}^{2,n}_{(0,k),(1,k-1),\cdots,(k,0)}$.

Bit Based: $\mathcal{D}_k^1 \longmapsto \mathcal{D}^{2,1}_{(0,k),(1,k-1),\cdots,(k,0)}, (k \in \{0,1\})$.

# Modeling Copy

General Rule: $\mathcal{D}_k^n \longmapsto \mathcal{D}^{2,n}_{(0,k),(1,k-1),\cdots,(k,0)}$.

Bit Based: $\mathcal{D}_k^1 \longmapsto \mathcal{D}^{2,1}_{(0,k),(1,k-1),\cdots,(k,0)}, (k \in \{0,1\})$.

Division Trail: $(0) \longrightarrow (0,0)$, $(1) \longrightarrow (0,1)$, $(1) \longrightarrow (1,0)$.

# Modeling Copy

General Rule: $\mathcal{D}_k^n \longmapsto \mathcal{D}^{2,n}_{(0,k),(1,k-1),\cdots,(k,0)}$.

Bit Based: $\mathcal{D}_k^1 \longmapsto \mathcal{D}^{2,1}_{(0,k),(1,k-1),\cdots,(k,0)}, (k \in \{0,1\})$.

Division Trail: $(0) \longrightarrow (0,0)$, $(1) \longrightarrow (0,1)$, $(1) \longrightarrow (1,0)$.

**Linear Inequality Description**

Denote $(a) \longrightarrow (b_0, b_1)$ a division trail of Copy operation, the following (in)equalities are sufficient to describe the division property propagations:

$$\begin{cases} a - b_0 - b_1 = 0 \\ a, b_0, b_1 \in \{0,1\} \end{cases}$$

# Modeling Xor

# Modeling Xor

General Rule: $\mathcal{D}^{n,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{n}_{k_0+k_1}$.

# Modeling Xor

General Rule: $\mathcal{D}^{n,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{n}_{k_0+k_1}$.

Bit Based: $\mathcal{D}^{1,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{1}_{k_0+k_1}, (k \in \{0,1\})$.

# Modeling Xor

General Rule: $\mathcal{D}^{n,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{n}_{k_0+k_1}$.

Bit Based: $\mathcal{D}^{1,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{1}_{k_0+k_1}, (k \in \{0,1\})$.

Division Trail: $(0,0) \longrightarrow (0)$, $(0,1) \longrightarrow (1)$, $(1,0) \longrightarrow (1)$, $(1,1) \xrightarrow{abort} (2)$.

# Modeling Xor

General Rule: $\mathcal{D}^{n,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{n}_{k_0+k_1}$.

Bit Based: $\mathcal{D}^{1,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{1}_{k_0+k_1}, (k \in \{0,1\})$.

Division Trail: $(0,0) \longrightarrow (0)$, $(0,1) \longrightarrow (1)$, $(1,0) \longrightarrow (1)$, $(1,1) \xrightarrow{abort} (2)$.

## Linear Inequality Description

Denote $(a_0, a_1) \longrightarrow (b)$ a division trail of Xor operation, the following (in)equalities are sufficient to describe the division property propagations:

$$\begin{cases} a_0 + a_1 - b = 0 \\ a_0, a_1, b \in \{0,1\} \end{cases}$$

# Modeling And

# Modeling And

General Rule: $\mathcal{D}^{n,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{n}_{\max(k_0,k_1)}$.

# Modeling And

General Rule: $\mathcal{D}^{n,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{n}_{\max(k_0,k_1)}$.

Bit Based: $\mathcal{D}^{1,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{1}_{\max(k_0,k_1)}$.

# Modeling And

General Rule: $\mathcal{D}^{n,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{n}_{\max(k_0,k_1)}$.

Bit Based: $\mathcal{D}^{1,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{1}_{\max(k_0,k_1)}$.

Division Trail: $(0,0) \longrightarrow (0)$, $(0,1) \longrightarrow (1)$, $(1,0) \longrightarrow (1)$, $(1,1) \longrightarrow (1)$.

# Modeling And

General Rule: $\mathcal{D}^{n,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{n}_{\max(k_0,k_1)}$.

Bit Based: $\mathcal{D}^{1,2}_{(k_0,k_1)} \longmapsto \mathcal{D}^{1}_{\max(k_0,k_1)}$.

Division Trail: $(0,0) \longrightarrow (0)$, $(0,1) \longrightarrow (1)$, $(1,0) \longrightarrow (1)$, $(1,1) \longrightarrow (1)$.

## Linear Inequality Description

Denote $(a_0, a_1) \longrightarrow (b)$ a division trail of And operation, the following inequalities are sufficient to describe the division property propagations:

$$\begin{cases} b - a_0 \geq 0 \\ b - a_1 \geq 0 \\ b - a_0 - a_1 \leq 0 \\ a_0, a_1, b \in \{0,1\} \end{cases}$$

# Modeling Sbox — PRESENT Sbox

$\mathcal{D}^{1,4}_{(0,1,1,1)}$

# Modeling Sbox — PRESENT Sbox

$\mathcal{D}^{1,4}_{(0,1,1,1)}$

## PRESENT Sbox

ANF of PRESENT Sbox

$$\begin{cases} y_3 = 1 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_1x_2 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_2 = 1 \oplus x_2 \oplus x_3 \oplus x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_1 = x_1 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_0 = x_0 \oplus x_2 \oplus x_3 \oplus x_1x_2 \end{cases}$$

# Modeling Sbox — PRESENT Sbox

$\mathcal{D}^{1,4}_{(0,1,1,1)}$

## PRESENT Sbox

ANF of PRESENT Sbox

$$\begin{cases} y_3 = 1 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_1x_2 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_2 = 1 \oplus x_2 \oplus x_3 \oplus x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_1 = x_1 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_0 = x_0 \oplus x_2 \oplus x_3 \oplus x_1x_2 \end{cases}$$

- $\mathcal{D}^{1,4}_{(0,1,1,1)} \Longrightarrow$ only $\sum_x x_2x_1x_0$ and $\sum_x x_3x_2x_1x_0$ are unknow

# Modeling Sbox — PRESENT Sbox

$\mathcal{D}^{1,4}_{(0,1,1,1)}$

## PRESENT Sbox

ANF of PRESENT Sbox

$$\begin{cases} y_3 = 1 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_1x_2 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_2 = 1 \oplus x_2 \oplus x_3 \oplus x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_1 = x_1 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_0 = x_0 \oplus x_2 \oplus x_3 \oplus x_1x_2 \end{cases}$$

- $\mathcal{D}^{1,4}_{(0,1,1,1)} \Longrightarrow$ only $\sum_x x_2x_1x_0$ and $\sum_x x_3x_2x_1x_0$ are unknow

# Modeling Sbox — PRESENT Sbox

$\mathcal{D}^{1,4}_{(0,1,1,1)}$

## PRESENT Sbox

ANF of PRESENT Sbox

$$\begin{cases} y_3 = 1 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_1x_2 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_2 = 1 \oplus x_2 \oplus x_3 \oplus x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_1 = x_1 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_0 = x_0 \oplus x_2 \oplus x_3 \oplus x_1x_2 \end{cases}$$

- $\mathcal{D}^{1,4}_{(0,1,1,1)} \Longrightarrow$ only $\sum_x x_2x_1x_0$ and $\sum_x x_3x_2x_1x_0$ are unknow $\Longrightarrow \sum_x y_0, \sum_x y_2$ are zero

# Modeling Sbox — PRESENT Sbox

$\mathcal{D}^{1,4}_{(0,1,1,1)}$

## PRESENT Sbox

ANF of PRESENT Sbox

$$\begin{cases} y_3 = 1 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_1x_2 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_2 = 1 \oplus x_2 \oplus x_3 \oplus x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_1 = x_1 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_0 = x_0 \oplus x_2 \oplus x_3 \oplus x_1x_2 \end{cases}$$

- $\mathcal{D}^{1,4}_{(0,1,1,1)} \Longrightarrow$ only $\sum_x x_2x_1x_0$ and $\sum_x x_3x_2x_1x_0$ are unknow $\Longrightarrow \sum_x y_0, \sum_x y_2$ are zero
- Moreover, $y_0y_2$ does not contain $x_2x_1x_0$ or $x_3x_2x_1x_0$ $\Longrightarrow \sum_x y_0y_2$ is zero

# Modeling Sbox — PRESENT Sbox

$$\mathcal{D}^{1,4}_{(0,1,1,1)} \xrightarrow{S} \mathcal{D}^{1,4}_{(0,0,1,0),(1,0,0,0)}$$

## PRESENT Sbox

ANF of PRESENT Sbox

$$\begin{cases} y_3 = 1 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_1x_2 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_2 = 1 \oplus x_2 \oplus x_3 \oplus x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_1 = x_1 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3 \\ y_0 = x_0 \oplus x_2 \oplus x_3 \oplus x_1x_2 \end{cases}$$

- $\mathcal{D}^{1,4}_{(0,1,1,1)} \Longrightarrow$ only $\sum_x x_2x_1x_0$ and $\sum_x x_3x_2x_1x_0$ are unknow $\Longrightarrow \sum_x y_0, \sum_x y_2$ are zero
- Moreover, $y_0y_2$ does not contain $x_2x_1x_0$ or $x_3x_2x_1x_0$ $\Longrightarrow \sum_x y_0y_2$ is zero

# Modeling Sbox

**Algorithm 1:** Calculating Division Trails of Sbox

**Input :** Input division property $\mathcal{D}_{\mathbf{k}}^{1,n}$ of $n$-bit Sbox, with $\mathbf{k} = (k_{n-1}, \cdots, k_0)$

**Output:** $\mathbb{K} \subset \{0,1\}^n$, such that the output division property is $\mathcal{D}_{\mathbb{K}}^{1,n}$

```
begin
    S̄ = {k̄ | k̄ ⪰ k}
    F(X) = {π_k̄(x) | k̄ ∈ S̄} // all unknown monomials
    K̄ = ∅
    for u ∈ (F_2)^n do
        if π_u(y) contains any monomial of F(X) then
            K̄ = K̄ ∪ {u}
        end
    end
    K = SizeReduce(K̄)
    return K
end
```

# Modeling Sbox — our new way

## PRESENT Sbox

Table: Division Trails of PRESENT Sbox

| Input $\mathcal{D}_{\mathbf{k}}^{1,4}$ | Output $\mathcal{D}_{\mathbb{K}}^{1,4}$ |
|---|---|
| (0,0,0,0) | (0,0,0,0) |
| (0,0,0,1) | (0,0,0,1) (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,0,1,0) | (0,0,0,1) (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,0,1,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,1,0,0) | (0,0,0,1) (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,1,0,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,1,1,0) | (0,0,0,1) (0,0,1,0) (1,0,0,0) |
| (0,1,1,1) | (0,0,1,0) (1,0,0,0) |
| (1,0,0,0) | (0,0,0,1) (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,0,0,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,0,1,0) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,0,1,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,1,0,0) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,1,0,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,1,1,0) | (0,1,0,1) (1,0,1,1) (1,1,1,0) |
| (1,1,1,1) | (1,1,1,1) |

The tables show 47 division trails of PRESENT Sbox.

# Modeling Sbox — our new way

## PRESENT Sbox

Table: Division Trails of PRESENT Sbox

| Input $\mathcal{D}_{\mathbf{k}}^{1,4}$ | Output $\mathcal{D}_{\mathbb{K}}^{1,4}$ |
|---|---|
| (0,0,0,0) | (0,0,0,0) |
| (0,0,0,1) | (0,0,0,1) (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,0,1,0) | (0,0,0,1) (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,0,1,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,1,0,0) | (0,0,0,1) (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,1,0,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (0,1,1,0) | (0,0,0,1) (0,0,1,0) (1,0,0,0) |
| (0,1,1,1) | (0,0,1,0) (1,0,0,0) |
| (1,0,0,0) | (0,0,0,1) (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,0,0,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,0,1,0) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,0,1,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,1,0,0) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,1,0,1) | (0,0,1,0) (0,1,0,0) (1,0,0,0) |
| (1,1,1,0) | (0,1,0,1) (1,0,1,1) (1,1,1,0) |
| (1,1,1,1) | (1,1,1,1) |

The tables show 47 division trails of PRESENT Sbox.

# Modeling Sbox — continued

- For any $n$-bit Sbox, compute all division trails.

# Modeling Sbox — continued

- For any $n$-bit Sbox, compute all division trails.
- Treat the division trails as $2n$-dimensional vectors.

# Modeling Sbox — continued

- For any $n$-bit Sbox, compute all division trails.
- Treat the division trails as $2n$-dimensional vectors.
- According to [Sun, eprint 2014], a set of linear inequalities can be computed with the help of Sage software whose feasible solutions are all the division trails.

# Modeling Sbox — continued

## Linear Inequalities Description of PRESENT Sbox

$$
\mathcal{L} = \begin{cases}
a_3 + a_2 + a_1 + a_0 - b_3 - b_2 - b_1 - b_0 \geq 0 \\
-a_2 - a_1 - 2a_0 + b_3 + b_1 - b_0 + 3 \geq 0 \\
-a_2 - a_1 - 2a_0 + 4b_3 + 3b_2 + 4b_1 + 2b_0 \geq 0 \\
-2a_3 - a_2 - a_1 + 2b_3 + 2b_2 + 2b_1 + b_0 + 1 \geq 0 \\
-2a_3 - a_2 - a_1 + 3b_3 + 3b_2 + 3b_1 + 2b_0 \geq 0 \\
-b_3 + b_2 - b_1 + b_0 + 1 \geq 0 \\
-2a_3 - 2a_2 - 2a_1 - 4a_0 + b_3 + 4b_2 + b_1 - 3b_0 + 7 \geq 0 \\
a_3 + a_2 + a_1 + a_0 - 2b_3 - 2b_2 + b_1 - 2b_0 + 1 \geq 0 \\
-4a_2 - 4a_1 - 2a_0 + b_3 - 3b_2 + b_1 + 2b_0 + 9 \geq 0 \\
-2a_0 - b_3 - b_2 - b_1 + 2b_0 + 3 \geq 0 \\
a_0 + b_3 - b_2 - 2b_1 - b_0 + 2 \geq 0 \\
a_3, a_2, a_1, a_0, b_3, b_2, b_1, b_0 \in \{0, 1\}
\end{cases}
$$

# Modeling Initial Division Property

$$\{\mathbf{k}\} \stackrel{def}{=} \mathbb{K}_0 \xrightarrow{f_r} \mathbb{K}_1 \xrightarrow{f_r} \mathbb{K}_2 \xrightarrow{f_r} \cdots \xrightarrow{f_r} \mathbb{K}_r$$

# Modeling Initial Division Property

$$\{\mathbf{k}\} \stackrel{def}{=} \mathbb{K}_0 \xrightarrow{f_r} \mathbb{K}_1 \xrightarrow{f_r} \mathbb{K}_2 \xrightarrow{f_r} \cdots \xrightarrow{f_r} \mathbb{K}_r$$

## Modeling Initial Division Property

$$\{\mathbf{k}\} \stackrel{def}{=} \mathbb{K}_0 \xrightarrow{f_r} \mathbb{K}_1 \xrightarrow{f_r} \mathbb{K}_2 \xrightarrow{f_r} \cdots \xrightarrow{f_r} \mathbb{K}_r$$

- Denote $(a_{n-1}^0, \cdots, a_0^0) \to \cdots \to (a_{n-1}^r, \cdots, a_0^r)$ as an $r$-round division trail, let $\mathbf{k} = (k_{n-1}, \cdots, k_0)$, then, add $a_i^0 = k_i$ for all $i = 0, 1, \cdots, n-1$ into the model.

# Modeling Initial Division Property

$$\{\mathbf{k}\} \stackrel{def}{=} \mathbb{K}_0 \xrightarrow{f_r} \mathbb{K}_1 \xrightarrow{f_r} \mathbb{K}_2 \xrightarrow{f_r} \cdots \xrightarrow{f_r} \mathbb{K}_r$$

- Denote $(a_{n-1}^0, \cdots, a_0^0) \rightarrow \cdots \rightarrow (a_{n-1}^r, \cdots, a_0^r)$ as an $r$-round division trail, let $\mathbf{k} = (k_{n-1}, \cdots, k_0)$, then, add $a_i^0 = k_i$ for all $i = 0, 1, \cdots, n-1$ into the model.

## Basic Strategy

Two issues need to be addressed:

1. Describe the division propagations by linear (in)equalities.
2. Convert search problem to estimate the minimal value of the objective function.

# Objective Function

Condition: If $\mathbb{K}_r$ contains all the $n$ unit vectors, $r$-round integral distinguisher doesn't exist.

# Objective Function

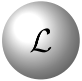
Condition: If $\mathbb{K}_r$ contains all the $n$ unit vectors, $r$-round integral distinguisher doesn't exist.

Objective Function: Denote $(a_{n-1}^0, \cdots, a_0^0) \rightarrow \cdots \rightarrow (a_{n-1}^r, \cdots, a_0^r)$ an $r$-round division trail, set the objective function as:

$$Obj : Min\{a_0^r + a_1^r + \cdots a_{n-1}^r\}$$

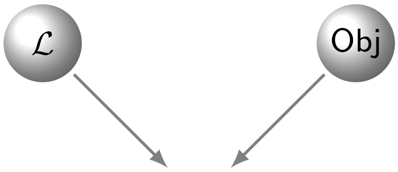
# Search Algorithm — Preparation

# Search Algorithm — Preparation

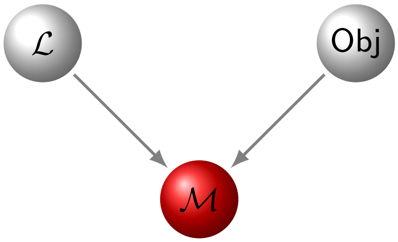


# Search Algorithm — Preparation

# Search Algorithm — Preparation



# Search Algorithm — Preparation



# Search Algorithm

**Algorithm 2:** Return $r$-round Distinguishers

**Input :** $M = M(\mathcal{L}, Obj)$.

**Output:** A set $\mathbb{S}$ indicating balanced bit positions.

**begin**
  $\mathbb{S} = \{a_0^r, \cdots, a_{n-1}^r\}$
  **for** *i in range (0,n)* **do**
    **if** *M has feasible solutions* **then**
      $M.optimize()$
      **if** $M.ObjVal = 1$ **then**
        $p =$ the bit position taking a value 1 in the objective function.
        $\mathbb{S} \backslash \{p\}$
        Remove the unit vector from $\mathcal{M}$
      **else**
        **return** $\mathbb{S}$
      **end**
    **else**
      **return** $\mathbb{S}$
    **end**
  **end**
  **return** $\mathbb{S}$
**end**

# Applications

Table: Results on Some Block Ciphers

| Ciphers | Block Size | Round(Pre.) | Round | Data | Balanced Bits | Time |
|---|---|---|---|---|---|---|
| SIMON32 | 32 | 15(Todo) | 14 | 31 | 16 | 4.1s |
| SIMON48 | 48 | 14(Zhang) | **16** | 47 | 24 | 48.2s |
| SIMON64 | 64 | 17(Zhang) | **18** | 63 | 22 | 6.7m |
| SIMON96 | 96 | 21(Zhang) | **22** | 95 | 5 | 17.4m |
| SIMON128 | 128 | 25(Zhang) | **26** | 127 | 3 | 58.4m |
| SIMECK32 | 32 | 15(Todo) | 15 | 31 | 7 | 6.5s |
| SIMECK48 | 48 | 12(Todo) | **18** | 47 | 5 | 56.6s |
| SIMECK64 | 64 | 12(Todo) | **21** | 63 | 5 | 3.0m |
| PRESENT | 64 | 7(Wu) | **9** | 60 | 1 | 3.4m |
| RECTANGLE | 64 | 7(Zhang) | **9** | 60 | 16 | 4.1m |
| LBlock | 64 | 16(Zhang) | 16 | 63 | 32 | 4.9m |
| TWINE | 64 | 16(Zhang) | 16 | 63 | 32 | 2.6m |

# Applications

Table: Results on Some Block Ciphers

| Ciphers | Block Size | Round(Pre.) | Round | Data | Balanced Bits | Time |
|---|---|---|---|---|---|---|
| SIMON32 | 32 | 15(Todo) | 14 | 31 | 16 | 4.1s |
| SIMON48 | 48 | 14(Zhang) | **16** | 47 | 24 | 48.2s |
| SIMON64 | 64 | 17(Zhang) | **18** | 63 | 22 | 6.7m |
| SIMON96 | 96 | 21(Zhang) | **22** | 95 | 5 | 17.4m |
| SIMON128 | 128 | 25(Zhang) | **26** | 127 | 3 | 58.4m |
| SIMECK32 | 32 | 15(Todo) | 15 | 31 | 7 | 6.5s |
| SIMECK48 | 48 | 12(Todo) | **18** | 47 | 5 | 56.6s |
| SIMECK64 | 64 | 12(Todo) | **21** | 63 | 5 | 3.0m |
| PRESENT | 64 | 7(Wu) | **9** | 60 | 1 | 3.4m |
| RECTANGLE | 64 | 7(Zhang) | **9** | 60 | 16 | 4.1m |
| LBlock | 64 | 16(Zhang) | 16 | 63 | 32 | 4.9m |
| TWINE | 64 | 16(Zhang) | 16 | 63 | 32 | 2.6m |

# Applications

Table: Results on Some Block Ciphers

| Ciphers | Block Size | Round(Pre.) | Round | Data | Balanced Bits | Time |
|---|---|---|---|---|---|---|
| SIMON32 | 32 | 15(Todo) | 14 | 31 | 16 | 4.1s |
| SIMON48 | 48 | 14(Zhang) | **16** | 47 | 24 | 48.2s |
| SIMON64 | 64 | 17(Zhang) | **18** | 63 | 22 | 6.7m |
| SIMON96 | 96 | 21(Zhang) | **22** | 95 | 5 | 17.4m |
| SIMON128 | 128 | 25(Zhang) | **26** | 127 | 3 | 58.4m |
| SIMECK32 | 32 | 15(Todo) | 15 | 31 | 7 | 6.5s |
| SIMECK48 | 48 | 12(Todo) | **18** | 47 | 5 | 56.6s |
| SIMECK64 | 64 | 12(Todo) | **21** | 63 | 5 | 3.0m |
| PRESENT | 64 | 7(Wu) | **9** | 60 | 1 | 3.4m |
| RECTANGLE | 64 | 7(Zhang) | **9** | 60 | 16 | 4.1m |
| LBlock | 64 | 16(Zhang) | 16 | 63 | 32 | 4.9m |
| TWINE | 64 | 16(Zhang) | 16 | 63 | 32 | 2.6m |

# References


Todo Yosuke (2015)
Structural Evaluation by Generalized Integral Property
*Advances in Cryptology–EUROCRYPT 2015* 287–314

Todo Yosuke (2015)
Integral Cryptanalysis on Full MISTY1
*Annual Cryptology Conference–CRYPTO 2015* 413–432

Todo Yosuke *et al.* (2016)
Bit-Based Division Property and Application to SIMON Family
*Fast Software Encryption–FSE 2016* To be appear.

Christina Boura *et al.* (2016)
Another View of Division Property
*Advances in Cryptology–CRYPTO 2016* To be appear.

Siwei Sun *et al.* (2014)
Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlcok, DES(L) and other bit-oriented block ciphers
*eprint–http://eprint.iacr.org/*

Zejun Xiang *et al.* (2016)
On the Division Property of SIMON48 AND SIMON64
*International Workshop on Security 2016*

# Thanks for Listening !

https://eprint.iacr.org/2016/857