

# USAGE OF MIXED INTEGER LINEAR PROGRAMMING IN CRYPTANALYSIS OF BLOCK CIPHERS

Halil İbrahim Kaplan

Istanbul Technical University  
Informatics Institute

2025

# Table of Contents

- 1 Preliminaries
- 2 MILP
- 3 Differential Cryptanalysis
- 4 ITUbee Block Cipher
- 5 MILP representation of block ciphers
- 6 References

# Table of Contents

- 1 Preliminaries
- 2 MILP
- 3 Differential Cryptanalysis
- 4 ITUbee Block Cipher
- 5 MILP representation of block ciphers
- 6 References

# Block Ciphers

Block ciphers designed to securely encrypt and decrypt fixed-size blocks of data by transforming plaintext into ciphertext using a symmetric key.

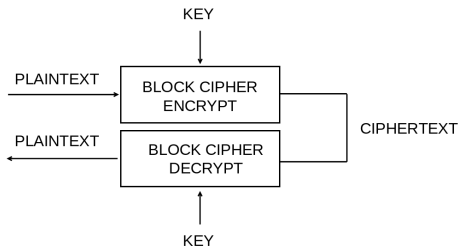


Figure 1: Block Cipher Encryption and Decryption

The main purpose of cryptanalysis:

- Prove that an algorithm is secure against known attacks in the open literature
- Find better complexities for those attacks and contribute to the security of the algorithm
- Break it ! (breaking an algorithm means obtaining the key with a lower complexity than the algorithm claims.)

# Table of Contents

- 1 Preliminaries
- 2 MILP**
- 3 Differential Cryptanalysis
- 4 ITUbee Block Cipher
- 5 MILP representation of block ciphers
- 6 References

# Mixed-Integer Linear Programming (MILP)

**Goal:** Optimize (maximize or minimize) a linear function.

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

**Constraints:**

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1$$

$$\vdots$$

$$a_{m1}x_1 + \cdots + a_{mn}x_n \leq b_m$$

$$x_1, x_2, \dots, x_p \in \mathbb{Z}, \quad x_{p+1}, \dots, x_n \geq 0$$

- $x_i$ : decision variables
- $c_i$ : objective function weights
- $a_{ij}$ : constraint coefficients
- $b_j$ : limits of constraints

# MILP Example

A company produces two products: **A** and **B**. The goal: **maximize profit** under production limits.

$$Z = 40x + 30y$$

- Profit per unit of A = 40
- Profit per unit of B = 30



# MILP Example: Constraints

## Constraints:

- **Labor:**  $2x + y \leq 100$  (100 hours available)
- **Machine:**  $x + 3y \leq 90$  (90 hours available)
- **Demand:**  $x \geq 20$  (min. 20 units of A)
- **Non-negativity:**  $x, y \geq 0$

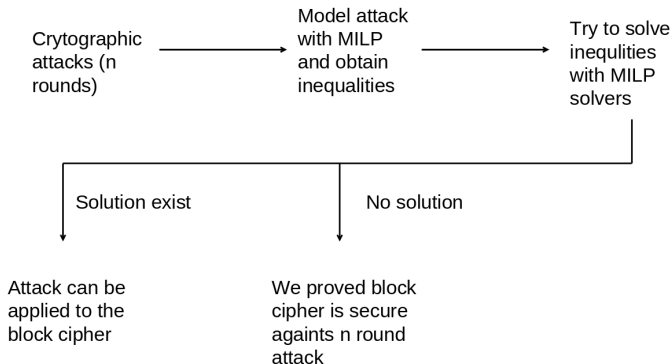
# MILP example

We will use Sagemath and GLPK solver for modeling.

```
1 p.<x,y> = MixedIntegerLinearProgram(maximization=True,
    solver="GLPK")
2 p.set_objective(40*x[0] + 30*y[0])
3 p.add_constraint(2*x[0] + y[0] == 100)
4 p.add_constraint(x[0] + 3*y[0] == 90)
5 p.add_constraint(x[0] >= 20)
6 p.add_constraint(y[0] >= 0)
7 print(p.solve())
8 print(p.get_values(x, y))
```

- Maximum profit: 2160
- A : 42
- B : 16

# Cryptanalysis and MILP



# Table of Contents

- 1 Preliminaries
- 2 MILP
- 3 Differential Cryptanalysis**
- 4 ITUbee Block Cipher
- 5 MILP representation of block ciphers
- 6 References

**Idea:** Study how input differences between two plaintexts affect output differences in ciphertexts.

## Definition

For two plaintexts  $P$  and  $P'$ :

$$\Delta P = P \oplus P'$$

# Differential Cryptanalysis: Probability

- Input difference:  $\Delta X$
- Output difference:  $\Delta Y$

**In an ideal cipher:** - For  $n$ -bit input,

$$\Pr[\Delta X \rightarrow \Delta Y] = \frac{1}{2^n}$$

**In practice:** - Some input differences lead to output differences with much higher probability.

# Differential Characteristics

Differential		Probability	
Differential characteristic	$(\Delta X, \Delta Y_1)$	$P_1$	Round 1
	$(\Delta Y_1, \Delta Y_2)$	$P_2$	Round 2
	$(\Delta Y_2, \Delta Y)$	$P_3$	Round 3
$(\Delta X, \Delta Y)$		$P_1 \times P_2 \times P_3$	

Figure 2: Concept of constructing differential characteristic

# Differential Cryptanalysis

$\Delta X / \Delta Y$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0
2	0	4	0	4	0	4	4	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
4	0	0	4	0	0	0	2	2	0	0	0	4	2	2	0	0
5	0	0	4	0	0	0	2	2	0	0	4	0	2	2	0	0
6	0	2	0	2	2	0	0	2	2	0	2	0	0	2	2	0
7	0	2	0	2	2	0	0	2	0	2	0	2	2	0	0	2
8	0	0	0	0	4	4	0	0	0	0	0	0	2	2	2	2
9	0	0	0	0	4	4	0	0	0	0	0	0	2	2	2	2
A	0	0	0	0	0	4	4	0	2	2	2	2	0	0	0	0
B	0	4	0	4	0	0	0	0	0	0	0	0	2	2	2	2
C	0	0	4	0	0	0	2	2	4	0	0	0	0	0	2	2
D	0	0	4	0	0	0	2	2	0	4	0	0	0	0	2	2
E	0	2	0	2	2	0	0	2	0	2	0	2	0	2	2	0
F	0	2	0	2	2	0	0	2	2	0	2	0	2	0	0	2

Table 1: Differential Distribution Table (DDT) of PRINCE algorithm



# Steps of Differential Cryptanalysis

- 1 **Select Differentials:** Use DDT to find  $(\Delta X, \Delta Y)$  pairs with high probability.
- 2 **Collect Pairs:** Generate many plaintext pairs  $P, P'$  with  $P \oplus P' = \Delta X$ .
- 3 **Encrypt:** Process these pairs through the cipher.
- 4 **Analyze Outputs:** Check if observed ciphertext differences match expected  $\Delta Y$ .
- 5 **Key Recovery:** Use biases to guess round subkeys (esp. in the last rounds).

# Differential Cryptanalysis

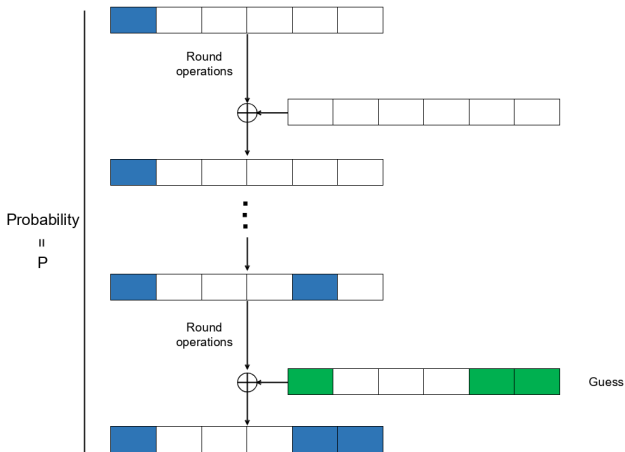


Figure 3: Differential Cryptanalysis

# Related-key Differential Cryptanalysis

Adds differentials also to key schedule.

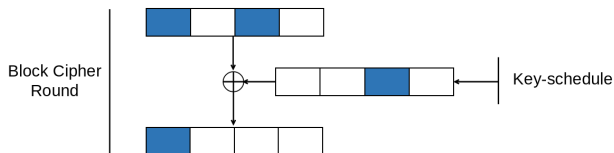


Figure 4: Related-key Differential Cryptanalysis

# Table of Contents

- 1 Preliminaries
- 2 MILP
- 3 Differential Cryptanalysis
- 4 ITUbee Block Cipher**
- 5 MILP representation of block ciphers
- 6 References

# ITUbee Block Cipher

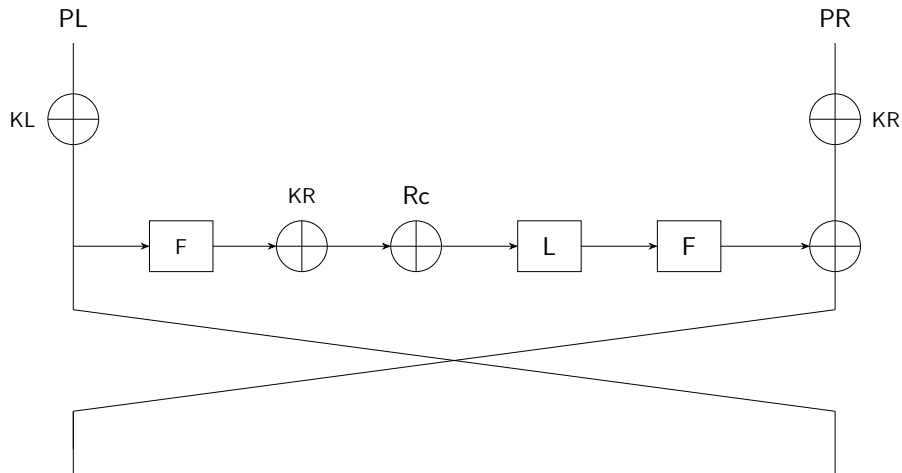


Figure 5: ITUbee round structure

## L Function

$$L(a, b, c, d, e) = (e \oplus a \oplus b, a \oplus b \oplus c, b \oplus c \oplus d, c \oplus d \oplus e, d \oplus e \oplus a)$$

## F function

$$F(X) = S(L(S(X)))$$

$$S(a \parallel b \parallel c \parallel d \parallel e) = s[a] \parallel s[b] \parallel s[c] \parallel s[d] \parallel s[e]$$

## Key Schedule

- 80-bit key is split into two 40-bit halves:

$$K = K_L \parallel K_R$$

- Odd rounds use  $K_R$
- Even rounds use  $K_L$
- No dedicated key schedule

# Table of Contents

- 1 Preliminaries
- 2 MILP
- 3 Differential Cryptanalysis
- 4 ITUbee Block Cipher
- 5 MILP representation of block ciphers**
- 6 References



# MILP Modelling of XOR Operation

Following [2],

$x_{in1}$ ,  $x_{in2}$  : input differences of XOR

$x_{out}$  : output difference of XOR.

The branch number is 2.

We introduce a binary variable  $d$ :

- $d = 0$  if  $x_{in1} = x_{in2} = x_{out} = 0$
- $d = 1$  otherwise

The XOR is modelled with:

$$x_{in1} + x_{in2} + x_{out} \geq 2d$$

$$d \geq x_{in1}$$

$$d \geq x_{in2}$$

$$d \geq x_{out}$$

# MILP Modelling of Linear Transformation

Similarly, for a linear transformation with:

- Inputs:  $x_{in1}, x_{in2}, \dots, x_{inM}$
- Outputs:  $x_{out1}, x_{out2}, \dots, x_{outM}$

Let  $B$  be the branch number and  $d$  the dummy variable (as in XOR modelling).

The model is:

$$x_{in1} + \dots + x_{inM} + x_{out1} + \dots + x_{outM} \geq B \cdot d$$

and

$$d \geq x_{in_i}, \quad d \geq x_{out_j} \quad \forall i, j$$

# MILP Representation of S-boxes

Goal: minimize the number of **active S-boxes** to find the best differential trails.

## Key idea:

- Active input ( $\neq 0$ )  $\Rightarrow$  active output
- Passive input ( $= 0$ )  $\Rightarrow$  passive output

We only track **activity** with a binary variable — no internal structure is modelled.

# H-representation example

$L(a, b, c) = (a \oplus b, a \oplus c, b \oplus c)$  where  $a, b, c$  are 2 bit values

Input Differences	Output Differences
0 0 0	0 0 0
0 0 1	0 1 1
0 1 0	1 0 1
0 1 1	1 1 0
0 1 1	1 1 1
1 0 0	1 1 0
1 0 1	1 0 1
1 0 1	1 1 1
1 1 0	0 1 1
...	...
1 1 1	1 0 1
1 1 1	1 1 0
1 1 1	1 1 1

Table 2: Input and Output Differences of  $L$  function

# H-representation example

No.	Inequality
1	$(0, 0, -1, 0, 0, 0)x + 1 \geq 0$
2	$(0, -1, 0, 0, 0, 0)x + 1 \geq 0$
3	$(-1, 0, 0, 0, 0, 0)x + 1 \geq 0$
4	$(0, 0, 0, -1, 0, 0)x + 1 \geq 0$
5	$(0, 0, 0, 0, -1, 0)x + 1 \geq 0$
6	$(0, 0, 0, 0, 0, -1)x + 1 \geq 0$
7	$(0, -1, 1, 0, 0, 1)x + 0 \geq 0$
8	$(0, 0, 0, -1, 1, 1)x + 0 \geq 0$
...	...
18	$(-1, 1, 0, 1, 0, 0)x + 0 \geq 0$
19	$(1, -1, -1, 1, 1, -1)x + 1 \geq 0$
20	$(1, -1, 0, 1, 0, 0)x + 0 \geq 0$
21	$(-1, 1, -1, 1, -1, 1)x + 1 \geq 0$
22	$(-1, -1, 1, -1, 1, 1)x + 1 \geq 0$

Table 3: H-representation of  $L$  function

# Differential cryptanalysis model with MILP

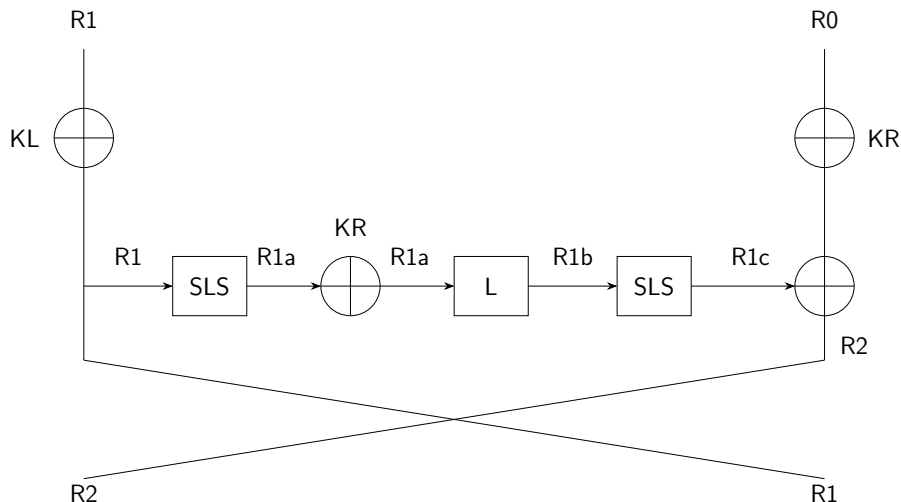


Figure 6: ITUbee MILP differential cryptanalysis sketch

# Differential cryptanalysis model with MILP

- Number of Constrains: 18169
- Number of Variables: 85
- Minimum number of active s-box for 3 round: 16.0
- Best probability for s-box:  $2^{-6}$
- Best probability for 3 round differential characteristic will be  $(2^{-6})^{16} = 2^{-96}$

which is not usable for differential attack.

# Differential cryptanalysis model with MILP

Stage	Binary Index	Characteristic
1. round input	R1	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
	R0	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
1. round middle values	R1a	{0: 1.0, 1: 0.0, 2: 0.0, 3: 1.0, 4: 1.0}
	R1b	{0: 0.0, 1: 1.0, 2: 1.0, 3: 0.0, 4: 1.0}
	R1c	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
2. round input	R2	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
	R1	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
2. round middle values	R2a	{0: 1.0, 1: 0.0, 2: 0.0, 3: 1.0, 4: 1.0}
	R2b	{0: 0.0, 1: 1.0, 2: 1.0, 3: 0.0, 4: 1.0}
	R2c	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
3. round input	R3	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
	R2	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
3. round middle values	R3a	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
	R3b	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
	R3c	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
4. round input	R4	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
	R3	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

**Table 4:** Best 3 round differential characteristic of ITUbee algorithm



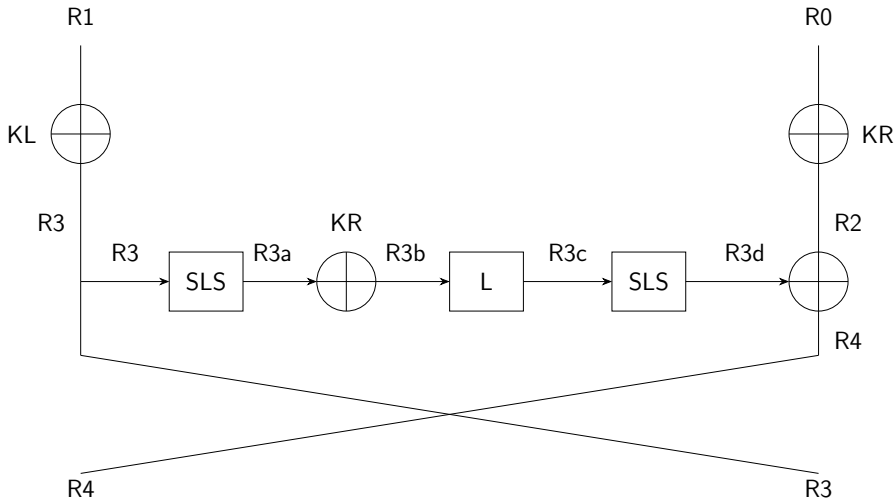


Figure 7: ITUbee MILP Related-Key Differential Cryptanalysis Sketch

# Differential cryptanalysis model with MILP

- Number of Constrains: 48649
- Number of Variables: 320
- Minimum number of active s-box for 8 round: 16.0
- Best probability for s-box:  $2^{-6}$
- Best probability for 8 round differential characteristic will be  $(2^{-6})^{16} = 2^{-96}$

which is not usable for related-key differential attack.

Stage	Binary Index	Characteristic
Key	KL	{0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0}
	KR	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
1. round input	R1	{0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0}
	R0	{0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
1. round middle values	R2	{0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
	R3-3a-3b-3c-3d	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
2. round input	R4	{0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
	R3	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
2. round middle values	R4a	{0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0}
	R4b-4c-4d	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
...	...	...
7. round input	R9	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
	R8	{0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
7. round middle values	R9a-9b-9c-9d	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
8. round input	R10	{0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
	R9	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
8. round middle values	R10a	{0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0}
	R10b-10c-10d	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
9. round input	R11	{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
	R10	{0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}

**Table 5:** Best 8 round related-key differential characteristic of ITUbee algorithm

# Our contribution

Stage	Differential	Linear	Related-key Differential
[1]	3 round	3 round	10 round
Our Result	3 round	3 round	<b>8 round</b>

Table 6: Result Comparison

# Table of Contents

- 1 Preliminaries
- 2 MILP
- 3 Differential Cryptanalysis
- 4 ITUbee Block Cipher
- 5 MILP representation of block ciphers
- 6 References**

- [1] Ferhat Karakoç, Hüseyin Demirci, and A Emre Harmancı. “ITUbee: a software oriented lightweight block cipher”. In: *Lightweight Cryptography for Security and Privacy: Second International Workshop, LightSec 2013, Gebze, Turkey, May 6-7, 2013, Revised Selected Papers 2*. Springer. 2013, pp. 16–27.
- [2] Nicky Mouha et al. “Differential and linear cryptanalysis using mixed-integer linear programming”. In: *Information Security and Cryptology: 7th International Conference, Inscrypt 2011, Beijing, China, November 30–December 3, 2011. Revised Selected Papers 7*. Springer. 2012, pp. 57–76.