Automation of Differential Cryptanalysis through CP and ML

Rocco Brunelli

Supervisors:

Prof. Marco Pedicini (Università Roma Tre)

Prof. Loïc Rouquette (EPITA, Lyon)

Università Roma Tre

18 June 2026

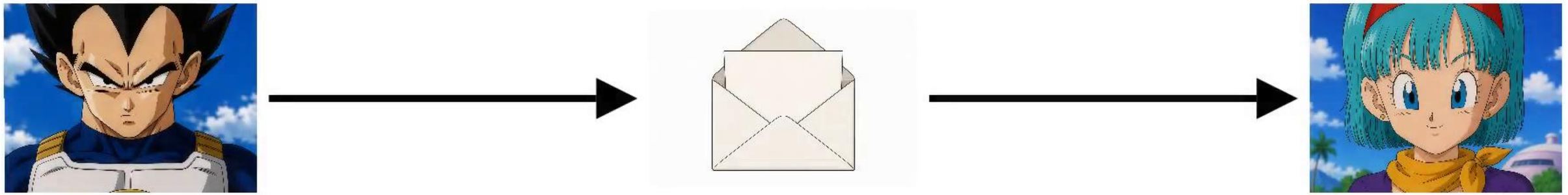
Roma Tre DME

Presentation Outline

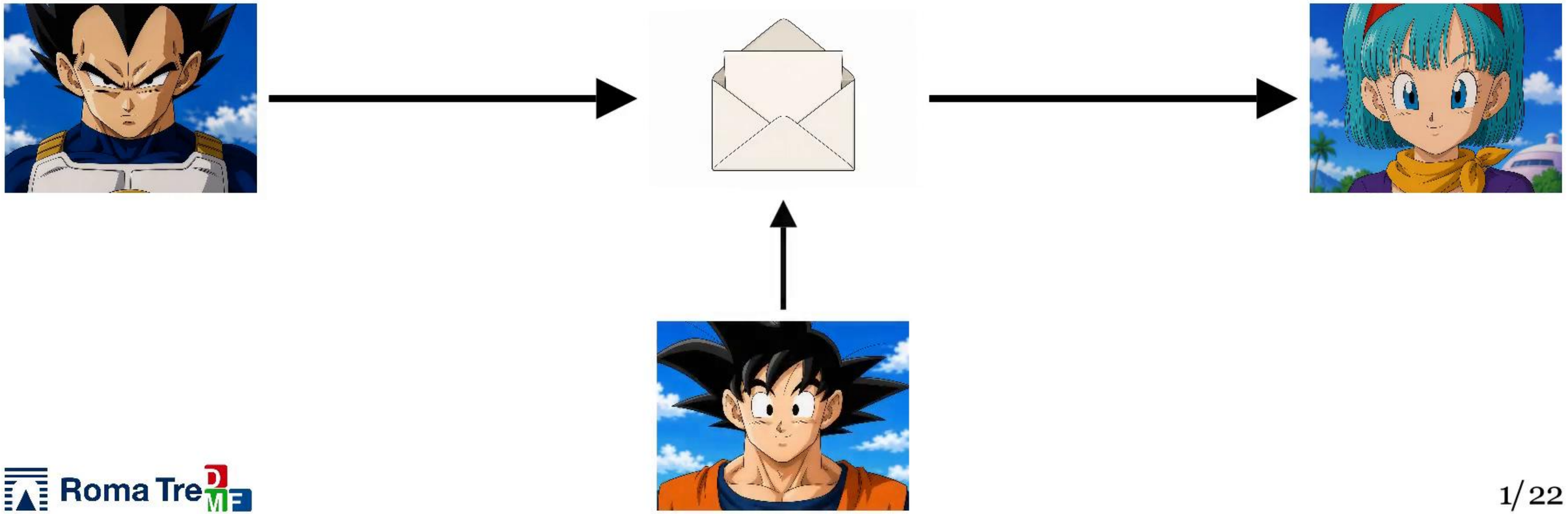
1. Introduction
2. TAGADA
3. Boomerang Attack
4. Boomerang Attack for TAGADA
5. Conclusion and Open Question

Introduction - Symmetric Cryptography

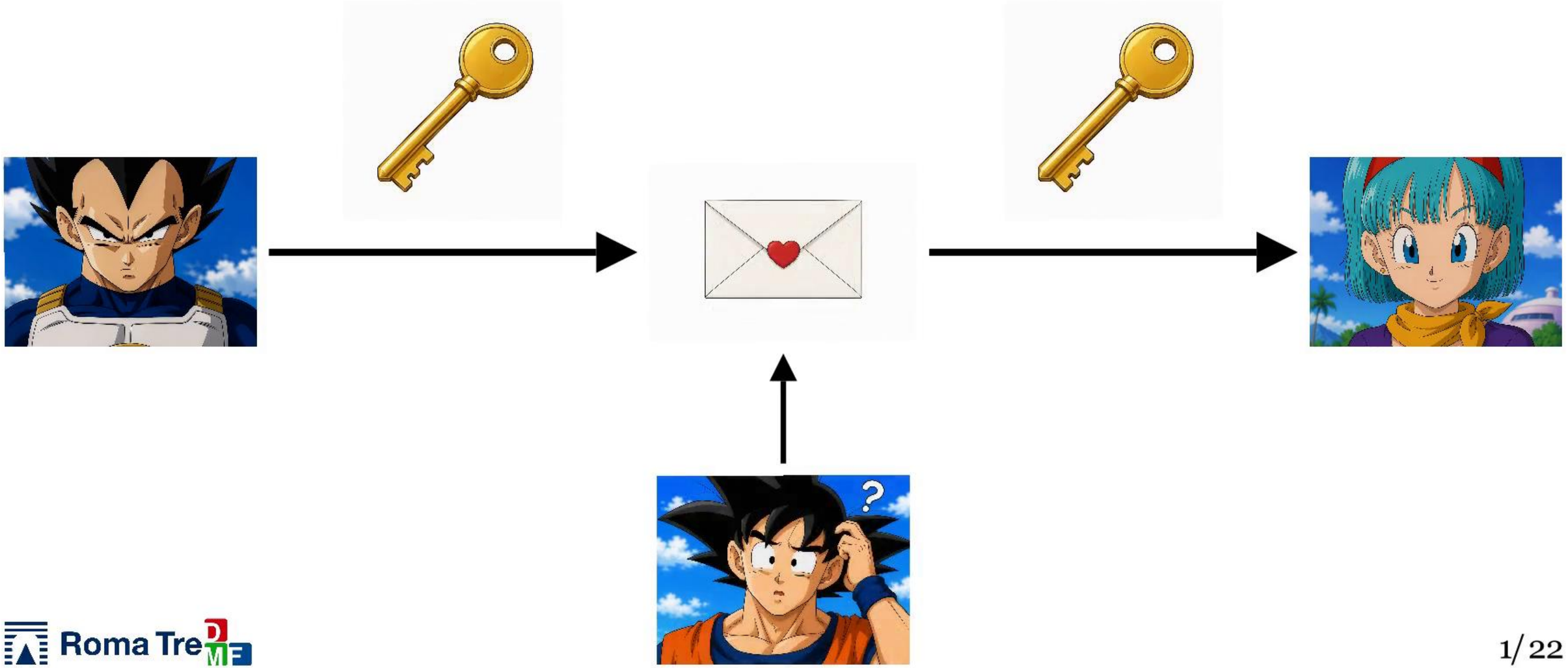
Introduction - Symmetric Cryptography



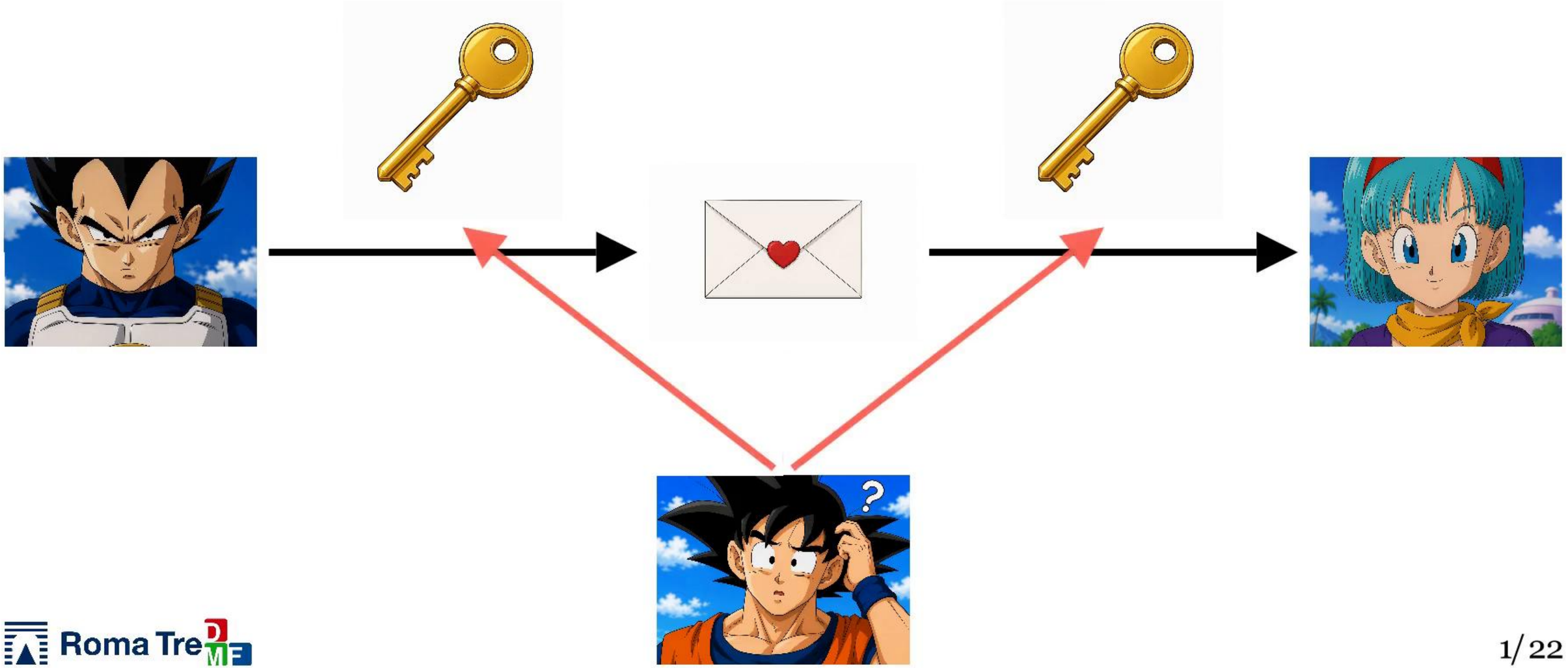
Introduction - Symmetric Cryptography



Introduction - Symmetric Cryptography



Introduction - Symmetric Cryptography



Introduction - Iterated Ciphers

Definition. Let $F : \mathbb{F}_2^n \times \mathbb{F}_2^k \to \mathbb{F}_2^n$, $F_k(\cdot) = F(\cdot, k)$ be called a **round function**. An **iterated encryption function** is

$$E : \mathbb{F}_2^n \times \mathbb{F}_2^{\tilde{k}} \to \mathbb{F}_2^n, \quad E_K(x) = F_{k_r} \circ F_{k_{r-1}} \circ \cdots \circ F_{k_1}(x)$$

where K is the **master key**, k_i are the **round keys** derived from K and r is the number of **rounds**.

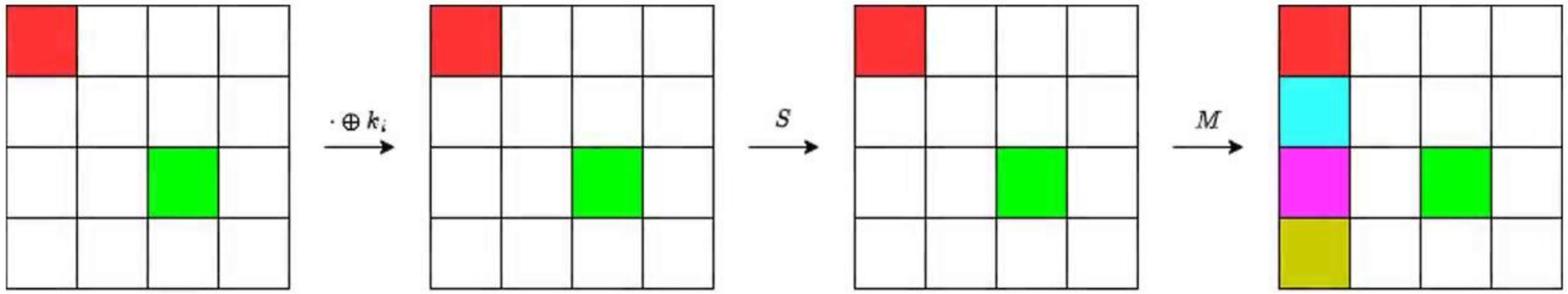
Introduction - Iterated Ciphers

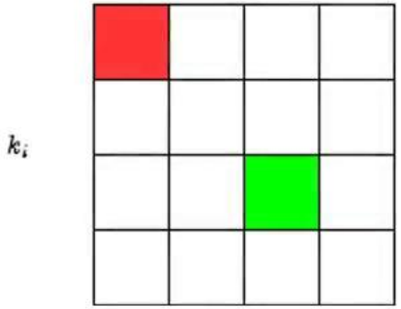
Definition. Let $F : \mathbb{F}_2^n \times \mathbb{F}_2^k \to \mathbb{F}_2^n$, $F_k(\cdot) = F(\cdot, k)$ be called a **round function**. An **iterated encryption function** is

$$E : \mathbb{F}_2^n \times \mathbb{F}_2^{\tilde{k}} \to \mathbb{F}_2^n, \quad E_K(x) = F_{k_r} \circ F_{k_{r-1}} \circ \cdots \circ F_{k_1}(x)$$

where K is the **master key**, k_i are the **round keys** derived from K and r is the number of **rounds**.

Definition. A **block cipher** is an iterated encryption function where the block size m is fixed and we apply the round functions on each block separately.





Introduction - Differential Cryptanalysis [1]

Introduction - Differential Cryptanalysis [1]

$$x \xrightarrow{F_{k_1}} y(1) \xrightarrow{F_{k_2}} y(2) \longrightarrow \cdots \longrightarrow y(r-1) \xrightarrow{F_{k_r}} y$$

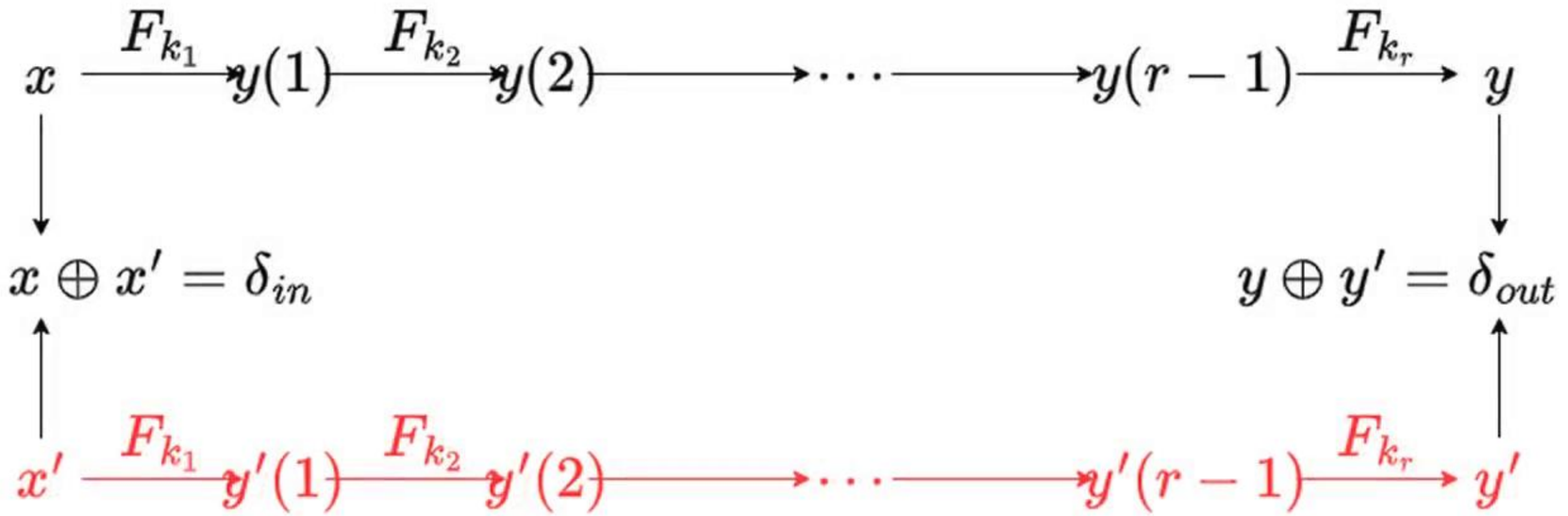
Introduction - Differential Cryptanalysis [1]

$$x \xrightarrow{F_{k_1}} y(1) \xrightarrow{F_{k_2}} y(2) \longrightarrow \cdots \longrightarrow y(r-1) \xrightarrow{F_{k_r}} y$$

$$x \xrightarrow{F_{k_1}} y(1) \xrightarrow{F_{k_2}} y(2) \longrightarrow \cdots \longrightarrow y(r-1) \xrightarrow{F_{k_r}} y$$

Introduction - Differential Cryptanalysis [1]

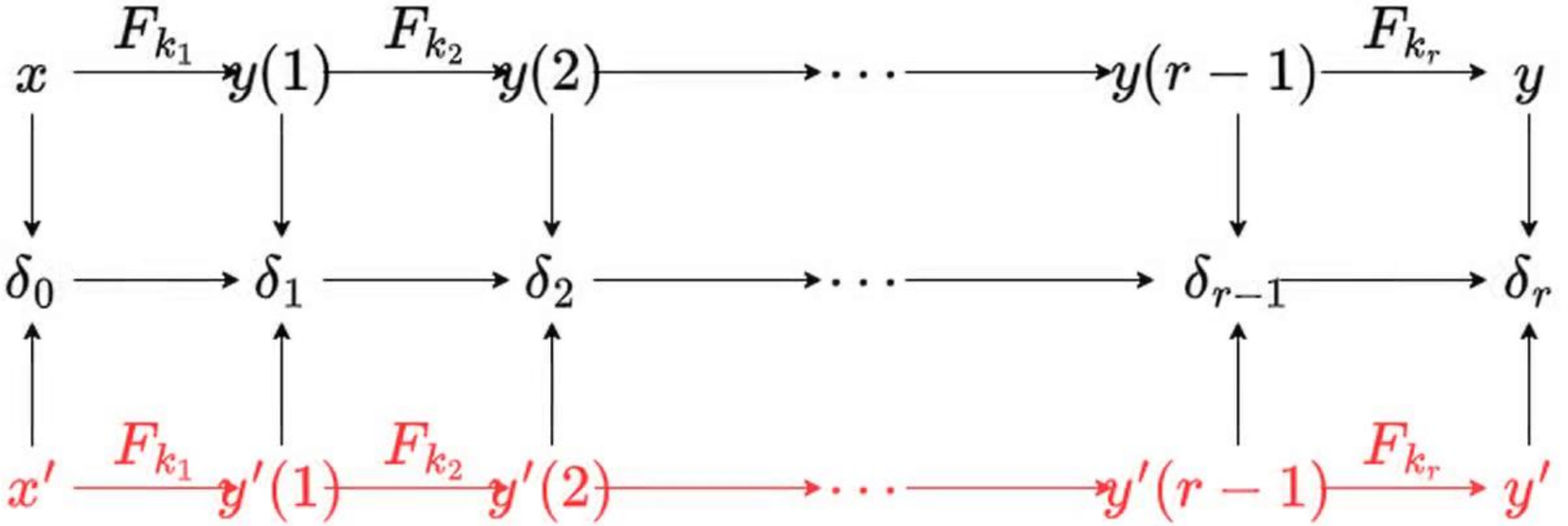
$$\mathbb{P}(y \oplus y' = \delta_{out} \mid x \oplus x' = \delta_{in}) = ?$$



Introduction - Differential Cryptanalysis [1]

$$\mathbb{P}(y \oplus y' = \delta_{out} \mid x \oplus x' = \delta_{in}) = \prod_{i=1}^{r} \mathbb{P}(y(i) \oplus y'(i) = \delta_i \mid y(i-1) \oplus y'(i-1) = \delta_{i-1})$$

↑ Markov Hypothesis



Introduction - Differential Cryptanalysis [1]

Definition. A **differential trail** is a sequence of differences through r round of the cipher:

$$\delta_0 \xrightarrow{E_K^r} \delta_r$$

with differential probability:

$$p = \prod_{i=1}^{r} \mathbb{P}(y(i) \oplus y'(i) = \delta_i \mid y(i-1) \oplus y'(i-1) = \delta_{i-1}) = \prod_{i=1}^{r} \mathcal{D}_S(\delta(i-1), \delta(i))$$

Introduction - Differential Cryptanalysis [1]

Definition. A **differential trail** is a sequence of differences through r round of the cipher:

$$\delta_0 \xrightarrow{E_K^r} \delta_r$$

with differential probability:

$$p = \prod_{i=1}^{r} \mathbb{P}(y(i) \oplus y'(i) = \delta_i \mid y(i-1) \oplus y'(i-1) = \delta_{i-1}) = \prod_{i=1}^{r} \mathcal{D}_S(\delta(i-1), \delta(i))$$

Definition. The **Differential Distribution Table** (DDT) is:

$$\mathcal{D}_S(\delta, \Delta) = |\{x \in \{0,1\}^m \mid S(x) \oplus S(x \oplus \delta) = \Delta\}|$$

Introduction - Classical Differential Attack

a. Find Differential Trail for a reduced number of rounds $r' < r$: $\delta_0 \to \delta_{r'}$ with high p

a.

Introduction - Classical Differential Attack

a. Find Differential Trail for a reduced number of rounds $r' < r$: $\delta_0 \to \delta_{r'}$ with high p

b. Build a differential distinguisher: $D_{\delta_0, \delta_{r'}}$

a.

b.

Introduction - Classical Differential Attack

a. Find Differential Trail for a reduced number of rounds $r' < r$: $\delta_0 \to \delta_{r'}$ with high p

b. Build a differential distinguisher: $D_{\delta_0, \delta_{r'}}$

c. Key Recovery Attack for r'' rounds $(r' < r'' \leq r)$

a.

b.

c.

Introduction - Classical Differential Attack

a. Find Differential Trail for a reduced number of rounds $r' < r$: $\delta_0 \to \delta_{r'}$ with high p

b. Build a differential distinguisher: $D_{\delta_0, \delta_{r'}}$

c. Key Recovery Attack for r'' rounds $(r' < r'' \leq r)$

Q. Where do we apply an automatic approach ?

a.

b.

c.

Introduction - Automatic Differential Attack

Q. Where do we apply an automatic approach ?

Introduction - Automatic Differential Attack

Q. Where do we apply an automatic approach ?

Constraint Programming

Machine Learning

Introduction - Automatic Differential Attack

Q. Where do we apply an automatic approach ?

Constraint Programming

Step 1. Automatic search of differentials with: MILP, CP, SAT

Machine Learning

Step 1. Automatic search of input difference with: Meta Algorithm

Introduction - Automatic Differential Attack

Q. Where do we apply an automatic approach ?

Constraint Programming

Step 1. Automatic search of differentials with: MILP, CP, SAT

Step 2. Create a Classical distinguisher

Machine Learning

Step 1. Automatic search of input difference with: Meta Algorithm

Step 2. Create a Neural distinguisher

$\mathcal{NN}(C, C') :=$

$\mathbb{P}(C = E_K^{r'}(P), C' = E_K^{r'}(P') | P \oplus P' = \delta_0)$

Introduction - Automatic Differential Attack

Q. Where do we apply an automatic approach ?

Constraint Programming

Step 1. Automatic search of differentials with: MILP, CP, SAT

Step 2. Create a Classical distinguisher

Step 3. Key Recovery Attack

Machine Learning

Step 1. Automatic search of input difference with: Meta Algorithm

Step 2. Create a Neural distinguisher

$$\mathcal{NN}(C, C') :=$$

$$\mathbb{P}(C = E_K^{r'}(P), C' = E_K^{r'}(P') | P \oplus P' = \delta_0)$$

Step 3. Key Recovery Attack

Overview Contributions

Q. Where do we apply an automatic approach ?

Constraint Programming

Step 1. Automatic search of differentials with: MILP, CP, SAT

Step 2. Create a Classical distinguisher

Step 3. Key Recovery Attack

- Modelize Boomerang in TAGADA [2]

Machine Learning

Step 1. Automatic search of input difference with: Meta Algorithm

Step 2. Create a Neural distinguisher

$$\mathcal{NN}(C, C') :=$$
$$\mathbb{P}(C = E_K^{r'}(P), C' = E_K^{r'}(P') | P \oplus P' = \delta_0)$$

Step 3. Key Recovery Attack

- Generic Partial Decryption [3]

TAGADA - General Idea

TAGADA - General Idea

TAGADA (Tools for Automatic Generation of Abstraction-based Differential Attack) is divided in two steps:

Step 1. Find Truncated Differential Trail

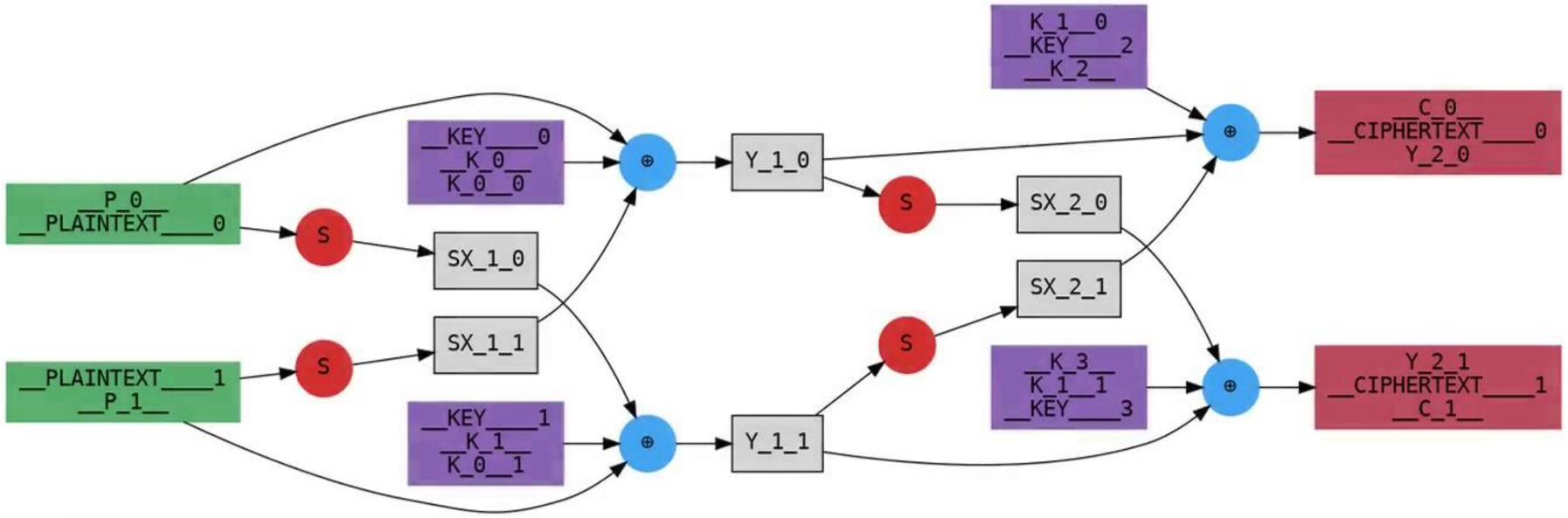
Step 2. Instatiate the Truncated in Differential

TAGADA - Representation Cipher

TAGADA (Tools for Automatic Generation of Abstraction-based Differential Attack) is divided in two steps:

Step 1. Find Truncated Differential Trail

Step 2. Instatiate the Truncated in Differential

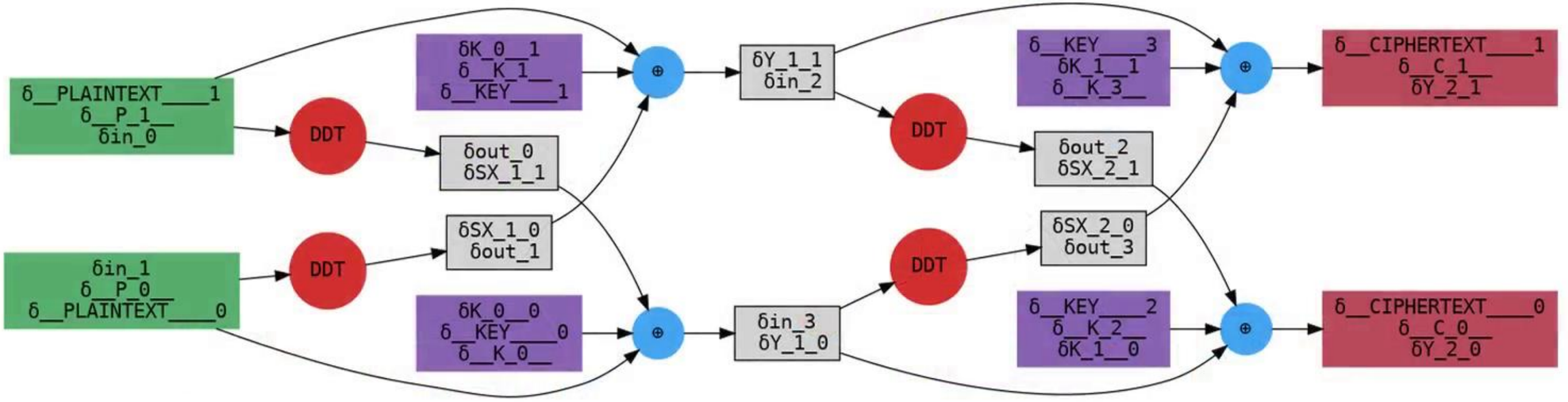


TAGADA - Representation Cipher

TAGADA (Tools for Automatic Generation of Abstraction-based Differential Attack) is divided in two steps:

Step 1. Find Truncated Differential Trail

Step 2. Instatiate the Truncated in Differential

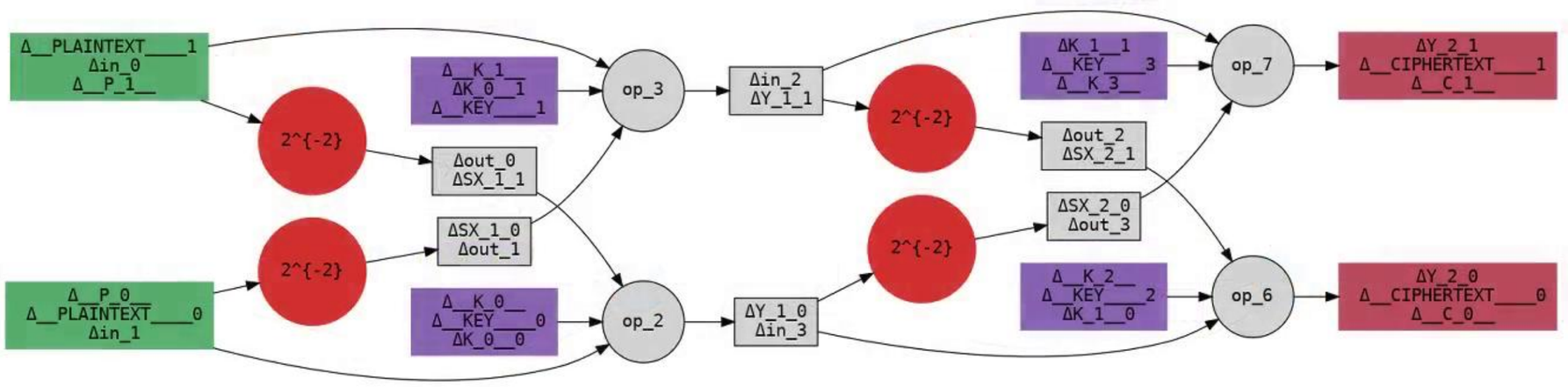


TAGADA - Representation Cipher

TAGADA (Tools for Automatic Generation of Abstraction-based Differential Attack) is divided in two steps:

Step 1. Find Truncated Differential Trail

Step 2. Instatiate the Truncated in Differential



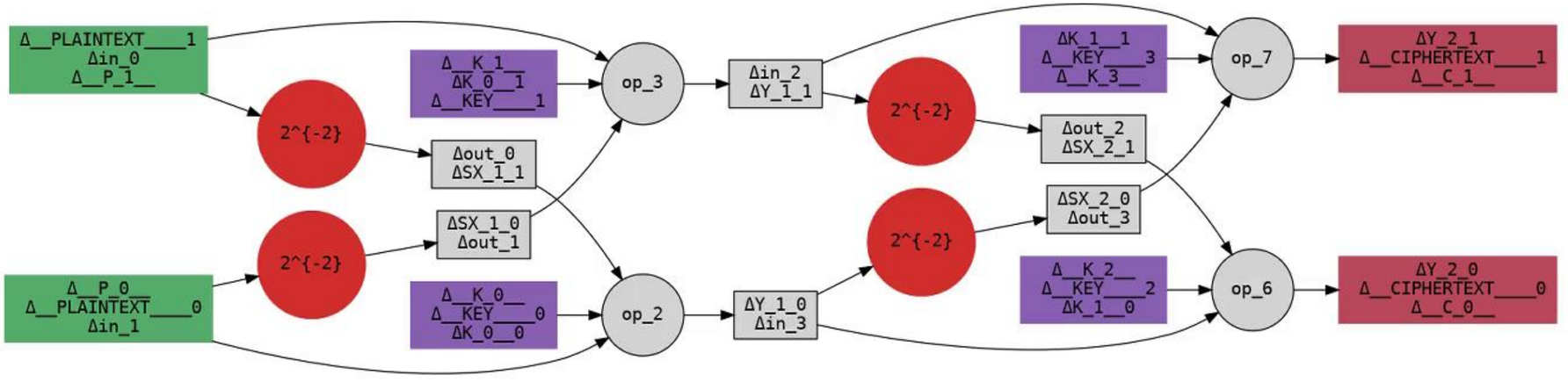
TAGADA - Truncated Cryptanalysis [5]

Abstraction. Let $x^j = x^j_{<1>} \,||\, \cdots \,||\, x^j_{<m>}$ be the state after j round, with $x^j_{<i>} \in \{0,1\}^t$. We denote:

$$\delta^j_{<i>} = x^j_{<i>} \oplus (x')^j_{<i>}$$

the difference at round j for the i-th word. The **Abstraction** of a difference is:

$$\Delta(\delta^j_{<i>}) = \Delta^j_{<i>} = \begin{cases} 1 & \text{if } \delta^j_{<i>} \in (0, 2^t - 1) \\ 0 & \text{if } \delta^j_{<i>} = 0 \end{cases}$$



TAGADA - Truncated Cryptanalysis [5]

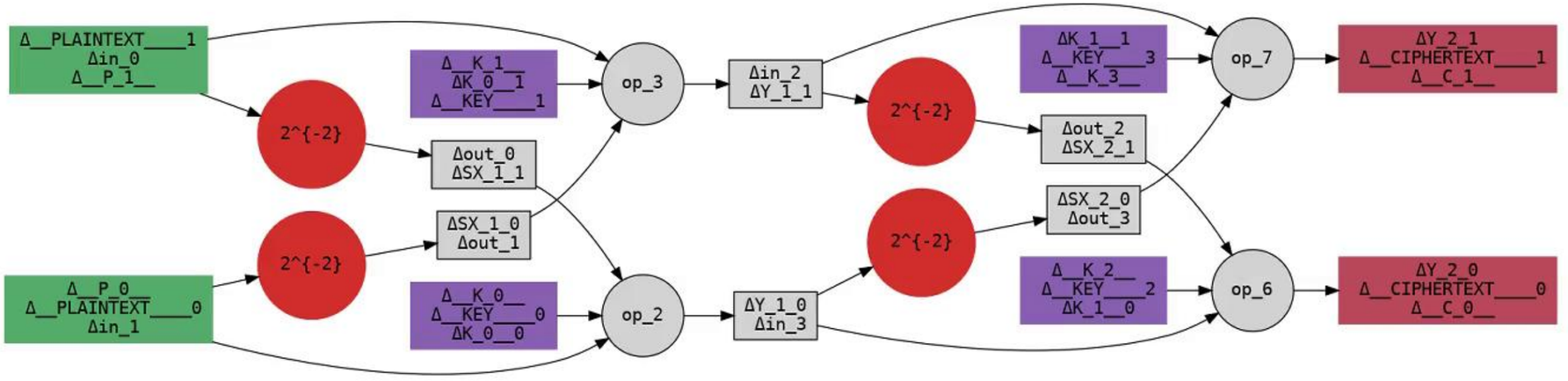
Abstraction. Let $x^j = x^j_{<1>} \,||\, \cdots \,||\, x^j_{<m>}$ be the state after j round, with $x^j_{<i>} \in \{0,1\}^t$. We denote:

$$\delta^j_{<i>} = x^j_{<i>} \oplus (x')^j_{<i>}$$

the difference at round j for the i-th word. The **Abstraction** of a difference is:

$$\Delta(\delta^j_{<i>}) = \Delta^j_{<i>} = \begin{cases} 1 & \text{if } \delta^j_{<i>} \in (0, 2^t - 1) \\ 0 & \text{if } \delta^j_{<i>} = 0 \end{cases}$$

Probability: $p_{trunc} \approx (\max_{\alpha,\beta} \mathcal{D}_S(\alpha, \beta))^{\#NL}$



TAGADA - Truncated Cryptanalysis [5]

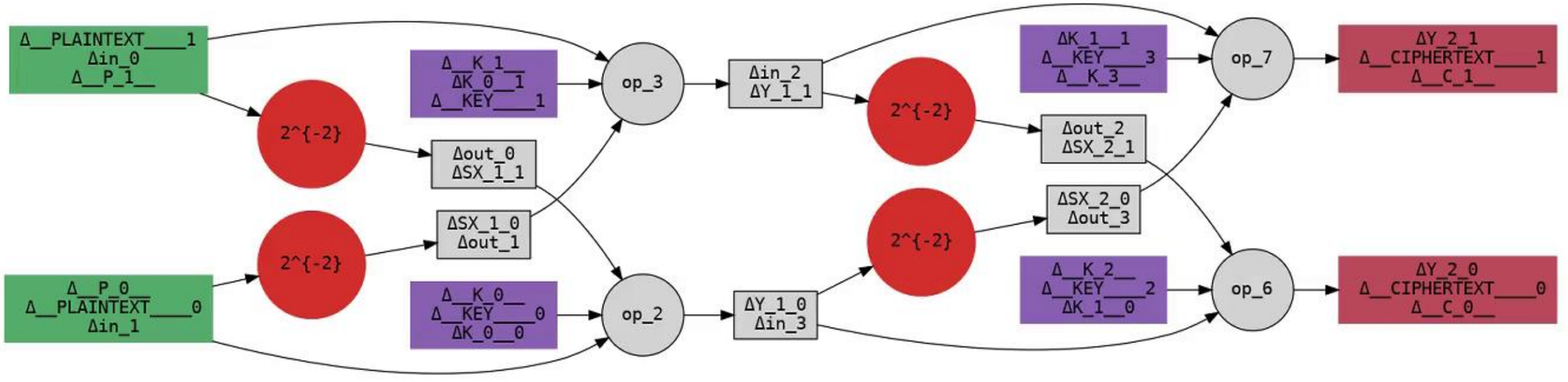
Abstraction. Let $x^j = x^j_{<1>} \,||\, \cdots \,||\, x^j_{<m>}$ be the state after j round, with $x^j_{<i>} \in \{0,1\}^t$. We denote:

$$\delta^j_{<i>} = x^j_{<i>} \oplus (x')^j_{<i>}$$

the difference at round j for the i-th word. The **Abstraction** of a difference is:

$$\Delta(\delta^j_{<i>}) = \Delta^j_{<i>} = \begin{cases} 1 & \text{if } \delta^j_{<i>} \in (0, 2^t - 1) \\ 0 & \text{if } \delta^j_{<i>} = 0 \end{cases}$$

Probability: $p_{trunc} \approx (\max_{\alpha,\beta} \mathcal{D}_S(\alpha,\beta))^{\#NL} \geq \mathbf{p_{diff}}$



TAGADA - Logic Modelization XOR

TAGADA - Logic Modelization XOR

Let $x_1, x_2, x_3 \in \{0, 1\}$ such that:

$$x_1 \oplus x_2 = x_3$$

The possible combinations are:

$$(x_1, x_2, x_3) \in \{(1, 0, 1), (0, 1, 1), (0, 0, 0), (1, 1, 0)\}$$

TAGADA - Logic Modelization XOR

Let $x_1, x_2, x_3 \in \{0, 1\}$ such that:

$$x_1 \oplus x_2 = x_3$$

The possible combinations are:

$$(x_1, x_2, x_3) \in \{(1, 0, 1), (0, 1, 1), (0, 0, 0), (1, 1, 0)\}$$

Considering the abstracted value:

$$\Delta(x_1) \oplus \Delta(x_2) = \Delta(x_3)$$

The possible tuples are:

$$(\Delta(\delta x_1), \Delta(\delta x_2), \Delta(\delta x_3)) \in \{(1, 0, 1), (0, 1, 1), (0, 0, 0), (1, 1, 0), (\mathbf{1}, \mathbf{1}, \mathbf{1})\}$$

TAGADA - Logic Modelization XOR

Let $x_1, x_2, x_3 \in \{0, 1\}$ such that:

$$x_1 \oplus x_2 = x_3$$

The possible combinations are:

$$(x_1, x_2, x_3) \in \{(1, 0, 1), (0, 1, 1), (0, 0, 0), (1, 1, 0)\}$$

Considering the abstracted value:

$$\Delta(x_1) \oplus \Delta(x_2) = \Delta(x_3)$$

The possible tuples are:

$$(\Delta(\delta x_1), \Delta(\delta x_2), \Delta(\delta x_3)) \in \{(1, 0, 1), (0, 1, 1), (0, 0, 0), (1, 1, 0), (\mathbf{1}, \mathbf{1}, \mathbf{1})\}$$

TAGADA uses the **DNF**:

$$(\neg\Delta(x_1) \wedge \neg\Delta(x_2) \wedge \neg\Delta(x_3)) \vee$$
$$(\Delta(x_1) \wedge \Delta(x_3)) \vee (\Delta(x_1) \wedge \Delta(x_2)) \vee (\Delta(x_2) \wedge \Delta(x_3))$$

TAGADA - Logic Modelization XOR

Let $x_1, x_2, x_3 \in \{0, 1\}$ such that:

$$x_1 \oplus x_2 = x_3$$

The possible combinations are:

$$(x_1, x_2, x_3) \in \{(1, 0, 1), (0, 1, 1), (0, 0, 0), (1, 1, 0)\}$$

Considering the abstracted value:

$$\Delta(x_1) \oplus \Delta(x_2) = \Delta(x_3)$$

The possible tuples are:

$$(\Delta(\delta x_1), \Delta(\delta x_2), \Delta(\delta x_3)) \in \{(1, 0, 1), (0, 1, 1), (0, 0, 0), (1, 1, 0), (\mathbf{1}, \mathbf{1}, \mathbf{1})\}$$

TAGADA uses the **DNF**:

$$(\neg\Delta(x_1) \wedge \neg\Delta(x_2) \wedge \neg\Delta(x_3)) \vee$$
$$(\Delta(x_1) \wedge \Delta(x_3)) \vee (\Delta(x_1) \wedge \Delta(x_2)) \vee (\Delta(x_2) \wedge \Delta(x_3))$$

Remark. TAGADA has implemented the n-variable XOR with **abstracted tables**.

TAGADA - Step 1

TAGADA - Step 1

The C(O)P problem (X, D, C, f_w) defined to solve Step 1 has:

TAGADA - Step 1

The C(O)P problem (X, D, C, f_w) defined to solve Step 1 has:

- **Set of Variables** X

$$X = \{\Delta(x^j_{<i>}) \mid x^j_{<i>} \text{ is a variable}\}_{i,j}$$

TAGADA - Step 1

The C(O)P problem (X, D, C, f_w) defined to solve Step 1 has:

- **Set of Variables** X

$$X = \{\Delta(x^j_{<i>}) \mid x^j_{<i>} \text{ is a variable}\}_{i,j}$$

- **Domain** of the variables D

$$D = \{0, 1\}$$

TAGADA - Step 1

The C(O)P problem (X, D, C, f_w) defined to solve Step 1 has:

- **Set of Variables** X

$$X = \{\Delta(x^j_{<i>}) \mid x^j_{<i>} \text{ is a variable}\}_{i,j}$$

- **Domain** of the variables D

$$D = \{0, 1\}$$

- **Constraint set** C

$$C = \{\text{Logic Modelization Operators}\}$$

TAGADA - Step 1

The C(O)P problem (X, D, C, f_w) defined to solve Step 1 has:

- **Set of Variables** X

$$X = \{\Delta(x^j_{<i>}) \mid x^j_{<i>} \text{ is a variable}\}_{i,j}$$

- **Domain** of the variables D

$$D = \{0, 1\}$$

- **Constraint set** C

$$C = \{\text{Logic Modelization Operators}\}$$

- **Objective function**

$$\min \sum_{i,j} w^j_{<i>} \Delta(x^j_{<i>})$$

TAGADA - Step 1

The C(O)P problem (X, D, C, f_w) defined to solve Step 1 has:

- **Set of Variables** X

$$X = \{\Delta(x^j_{<i>}) \mid x^j_{<i>} \text{ is a variable}\}_{i,j}$$

- **Domain** of the variables D

$$D = \{0, 1\}$$

- **Constraint set** C

$$C = \{\text{Logic Modelization Operators}\}$$

- **Objective function**

$$\min \sum_{i,j} w^j_{<i>} \Delta(x^j_{<i>})$$

The CP problem is written in MiniZinc [6] and it can be resolved by different solvers:

- PICAT [7]
- GUROBI [8]
- OR-TOOLS [9]

TAGADA - Step 2

The C(O)P problem (X, D, C, f_w) defined to solve Step 2 has:

- **Set of Variables** X

$$X = \{\delta(x^j_{<i>}) \mid x^j_{<i>} \text{ is a variable}\}_{i,j}$$

- **Domain** of the variables D

$$D = [0, 2^s - 1]$$

- **Constraint set** C

$$C = \{\text{Propagators modelized with CHOCO [10]}\}$$

- **Objective function**

$$\min \sum_{i,j} -\log_2 \left(\mathcal{D}_S \left(\delta(x^j_{<i>})\right)\right) \Delta(x^j_{<i>})$$

TAGADA - TwoStep

TAGADA - TwoStep

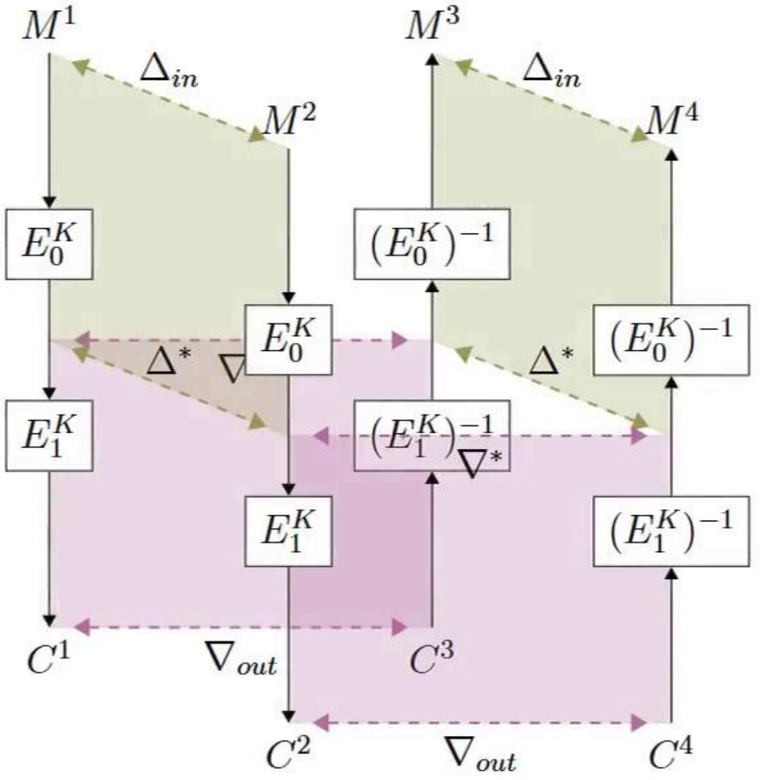
Algorithm:[2]TWOSTEP(G_Δ, G_δ)

Input:
G_Δ: step1 model
G_δ: step2 model

```
1  LB ← 0
2  UB ← 1
3  best ← null
4  sol_1 ← STEP1-OPT(G_Δ, seen, UB)
5  seen ← {}
6  UB ← P(sol_1)
7  while LB < UB do
8      seen ← seen ∪ {sol_1}
9      sol_2 ← STEP2(G_δ, sol_1, LB)
10     LB ← P(sol_2)
11     if LB < UB then
12         sol_1 ← STEP1-NEXT(G_Δ, seen, UB)
13         if sol_1 is null then
14             UB ← STEP1-NEXT-POSSIBLE-UB(G_Δ, seen, UB)
15             if LB ≥ UB then
16                 break
17             sol_1 ← STEP1-OPT(G_Δ, seen, UB)
18         UB ← P(sol_1)
19 return best
```

Boomerang Attack [11]

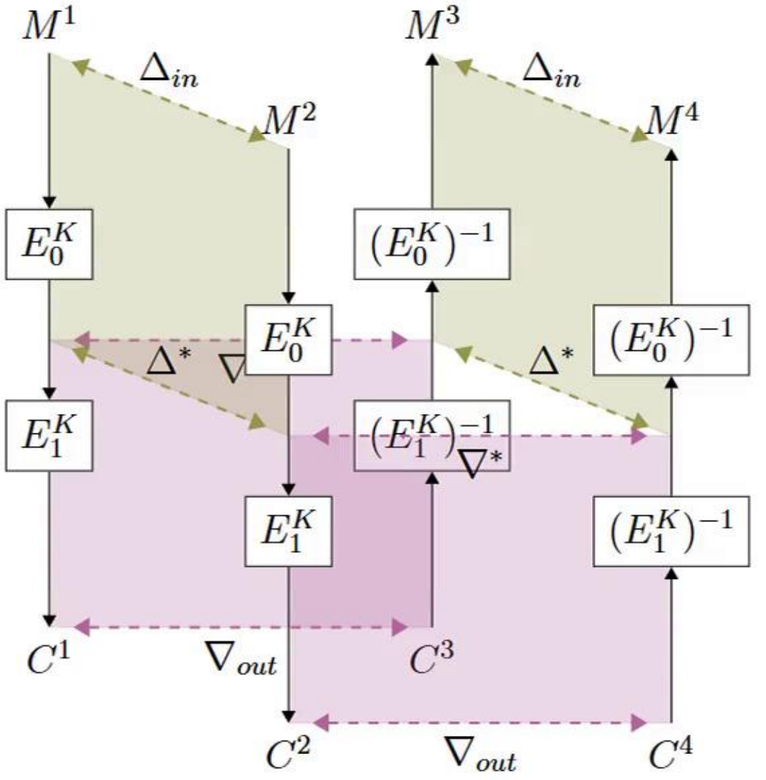
Boomerang Attack [11]



Boomerang Attack [11]

- **Boomerang Condition:**

$$E^{-1}(E(x) \oplus \nabla_{out}) \oplus E^{-1}(E(x \oplus \Delta_{in}) \oplus \nabla_{out}) = \Delta_{in}$$



Boomerang Attack [11]

- **Boomerang Condition:**

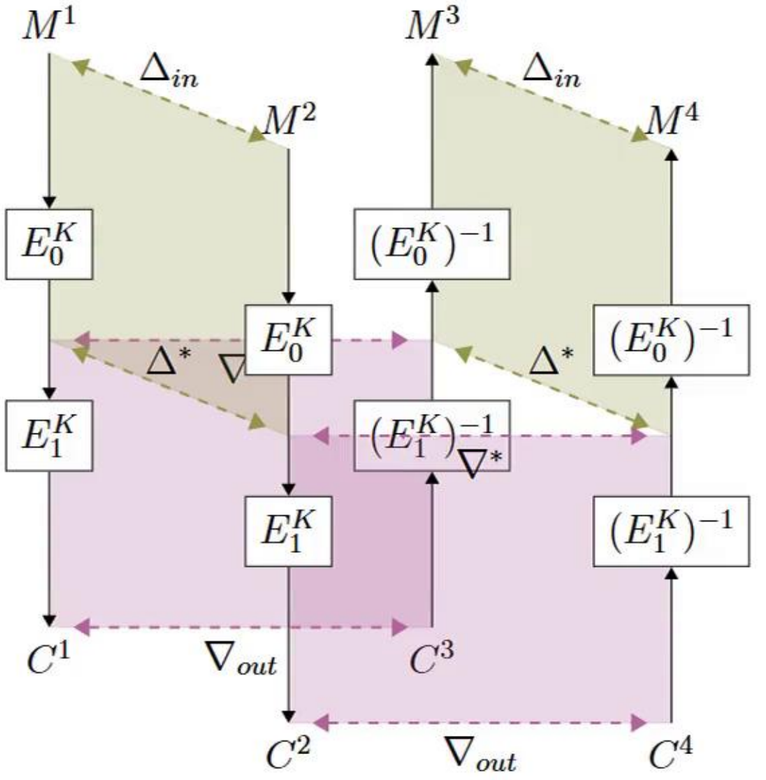
$$E^{-1}(E(x) \oplus \nabla_{out}) \oplus E^{-1}(E(x \oplus \Delta_{in}) \oplus \nabla_{out}) = \Delta_{in}$$

- Let:

$$E = E_1 \circ E_0$$

$$\mathbb{P}(\Delta_{in} \xrightarrow{E_0} \Delta^*) = p$$

$$\mathbb{P}(\nabla_{out} \xrightarrow{E_1^{-1}} \nabla^*) = q$$



Boomerang Attack [11]

- **Boomerang Condition:**

$$E^{-1}(E(x) \oplus \nabla_{out}) \oplus E^{-1}(E(x \oplus \Delta_{in}) \oplus \nabla_{out}) = \Delta_{in}$$

- Let:

$$E = E_1 \circ E_0$$

$$\mathbb{P}(\Delta_{in} \xrightarrow{E_0} \Delta^*) = p$$

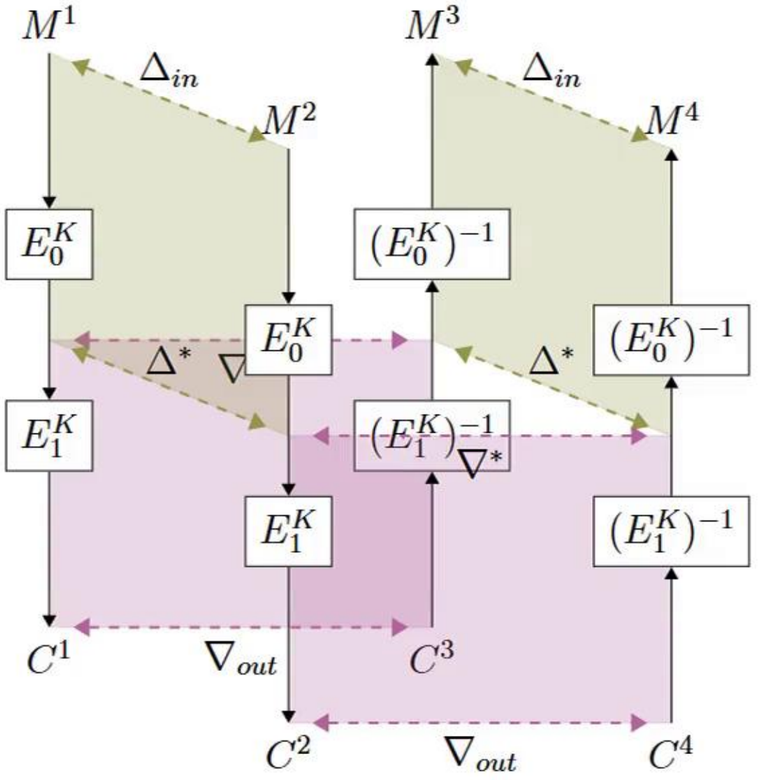
$$\mathbb{P}(\nabla_{out} \xrightarrow{E_1^{-1}} \nabla^*) = q$$

- **Boomerang Switch**:

$$S^{-1}(S(x) \oplus \nabla^*) \oplus S^{-1}(S(x \oplus \Delta^*) \oplus \nabla^*) = \Delta^*$$

- **Boomerang Probability**:

$$\mathbb{P}(\mathrm{BC}) \approx p^2 q^2$$



Sandwich Attack [12]

- **Sandwich Condition:**

$$E^{-1}(E(x) \oplus \nabla_{out}) \oplus E^{-1}(E(x \oplus \Delta_{in}) \oplus \nabla_{out}) = \Delta_{in}$$

- Let:

$$E = E_1 \circ E_m \circ E_0$$

$$\mathbb{P}(\Delta_{in} \xrightarrow{E_0} \Delta^*) = p$$

$$\mathbb{P}(\nabla_{out} \xrightarrow{E_1^{-1}} \nabla^*) = q$$

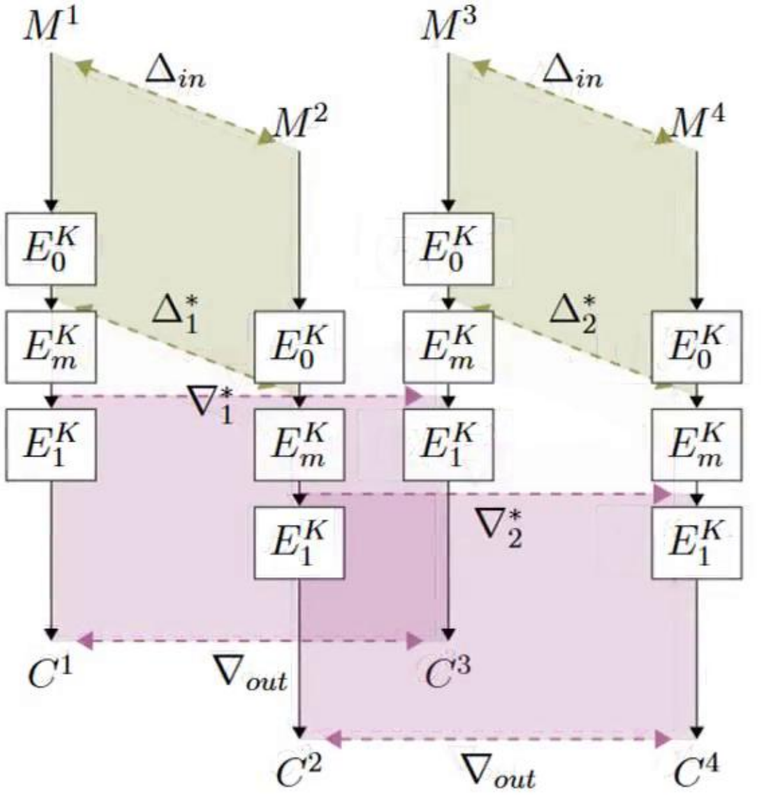
- **Sandwich Switch**:

$$E_m^{-1}(E_m(x) \oplus \nabla^*) \oplus E_m^{-1}(E_m(x \oplus \Delta^*) \oplus \nabla^*) = \Delta^*$$

- **Sandwich Probability**:

$$\mathbb{P}(\mathrm{SC}) \approx p^2 q^2 r(\Delta^*, \nabla^*)$$

$$r(\Delta^*, \nabla^*) = \mathbb{P}(SS)$$



Boomerang for TAGADA - Step 1

Boomerang for TAGADA - Step 1

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the best truncated trial for $E_\uparrow$ and one for $E_\downarrow$

Boomerang for TAGADA - Step 1

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the best truncated trial for $E_\uparrow$ and one for $E_\downarrow$

a. Automatically split the truncated DAG into $E_\uparrow$,$E_\downarrow$ and E_m

Boomerang for TAGADA - Step 1

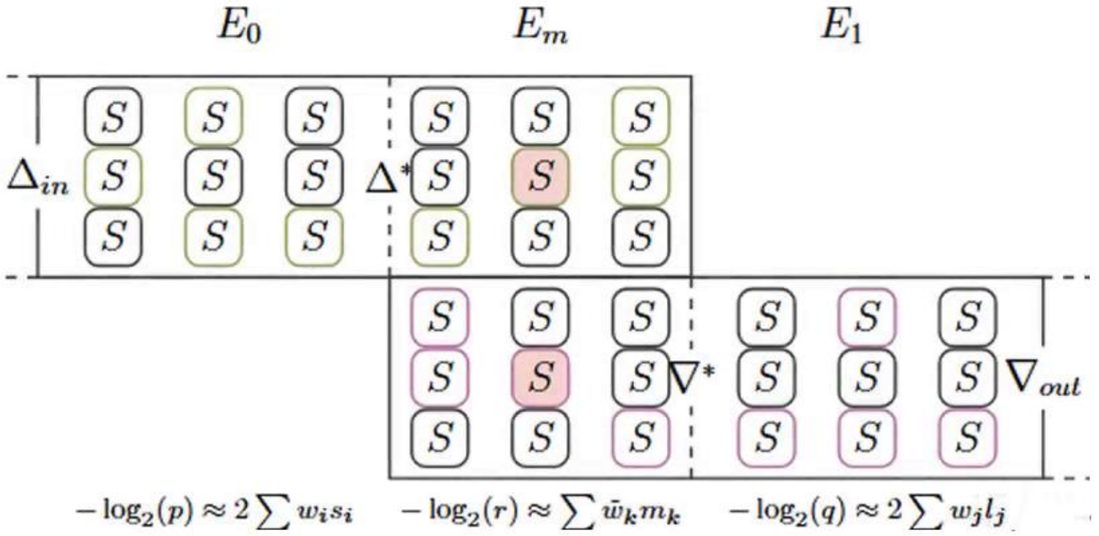
Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the best truncated trial for $E_\uparrow$ and one for $E_\downarrow$

a. Automatically split the truncated DAG into $E_\uparrow$,$E_\downarrow$ and E_m

$$E_\uparrow = \texttt{extract\_boomerang\_part}_\leftarrow(E, X_{k_1})$$
$$E_\downarrow = \texttt{extract\_boomerang\_part}_\rightarrow(E, X_{r_0})$$
$$E_m = E_\uparrow \cap E_\downarrow$$

Image from [13]



Roma Tre

Boomerang for TAGADA - Step 1

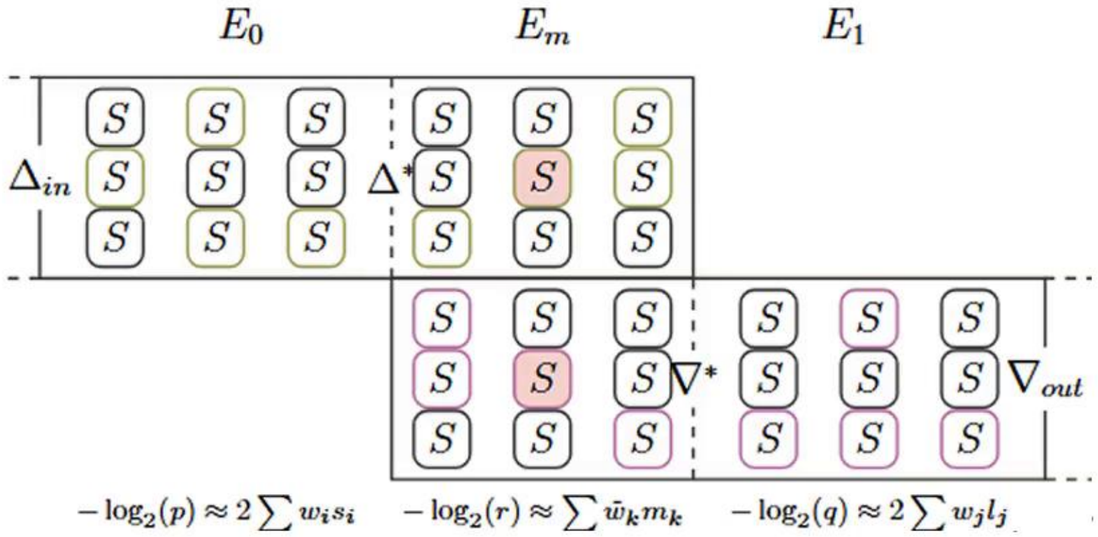
Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the best truncated trial for $E_\uparrow$ and one for $E_\downarrow$

a. Automatically split the truncated DAG into $E_\uparrow$,$E_\downarrow$ and E_m

b. Define the COP to solve

Image from [13]



Roma Tre

Boomerang for TAGADA - Step 1

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the best truncated trial for $E_\uparrow$ and one for $E_\downarrow$

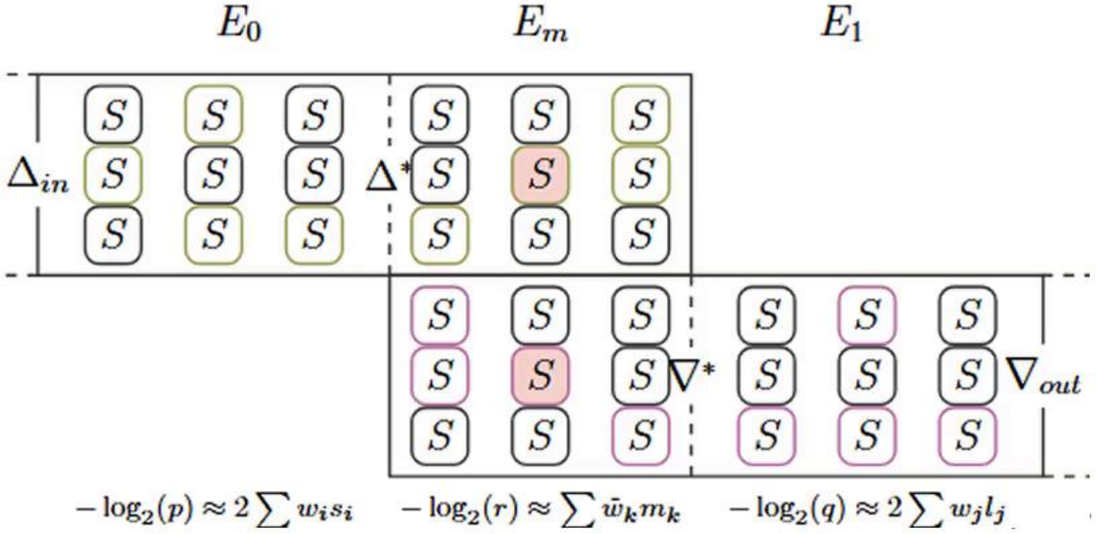
a. Automatically split the truncated DAG into $E_\uparrow$,$E_\downarrow$ and E_m

b. Define the COP to solve

- **Set of Variables** $X = \{\Delta(x^j_{<i>}) \mid x^j_{<i>}$ is a variable$\}_{i,j}$
- **Domain** of the variables $D = \{0, 1\}$
- **Constraint set** $C = *$
- **Objective function**

$$2\sum_i w_i s_i + \sum_j w_j m_j + 2\sum_k w_k l_k$$

Image from [13]

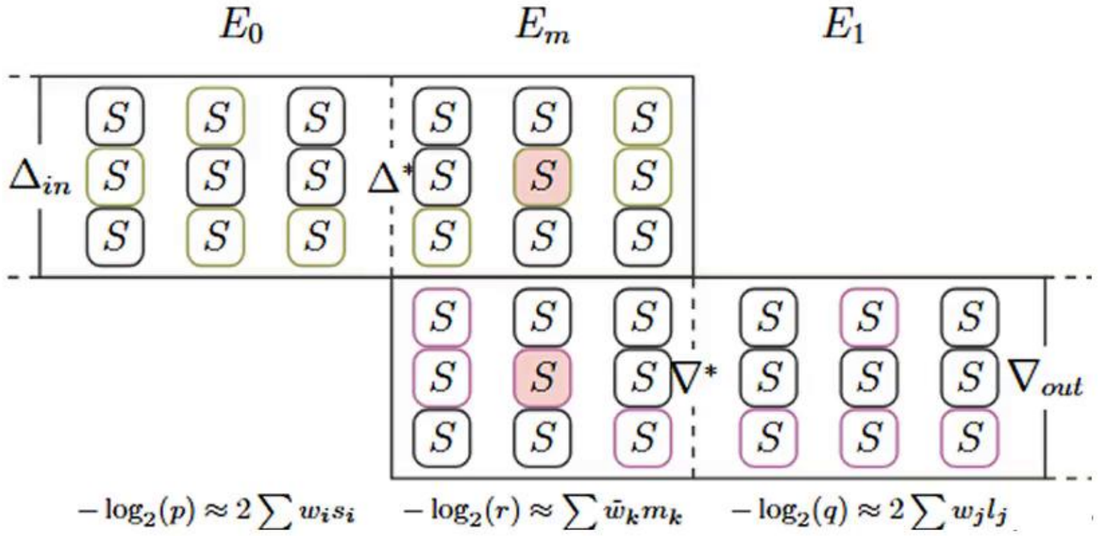


Roma Tre

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Image from [13]



Roma Tre

Boomerang for TAGADA - Step 1 Constraint

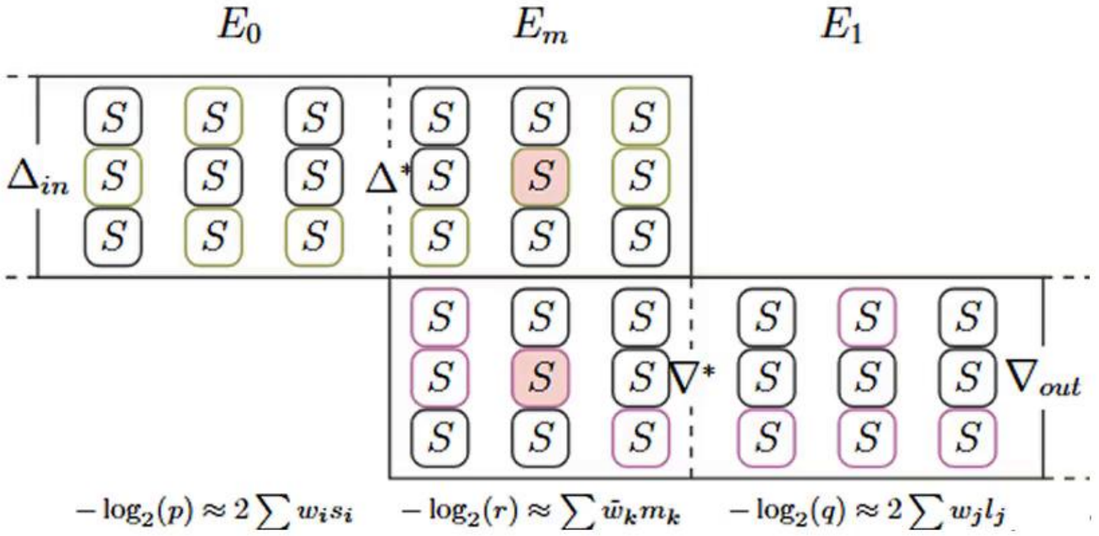
Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Non-Linear Operator

$E_\uparrow : \Delta_\uparrow^{in} = \Delta_\uparrow^{out}$

$E_\downarrow : \Delta_\downarrow^{in} = \Delta_\downarrow^{out}$

Image from [13]



Roma Tre

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

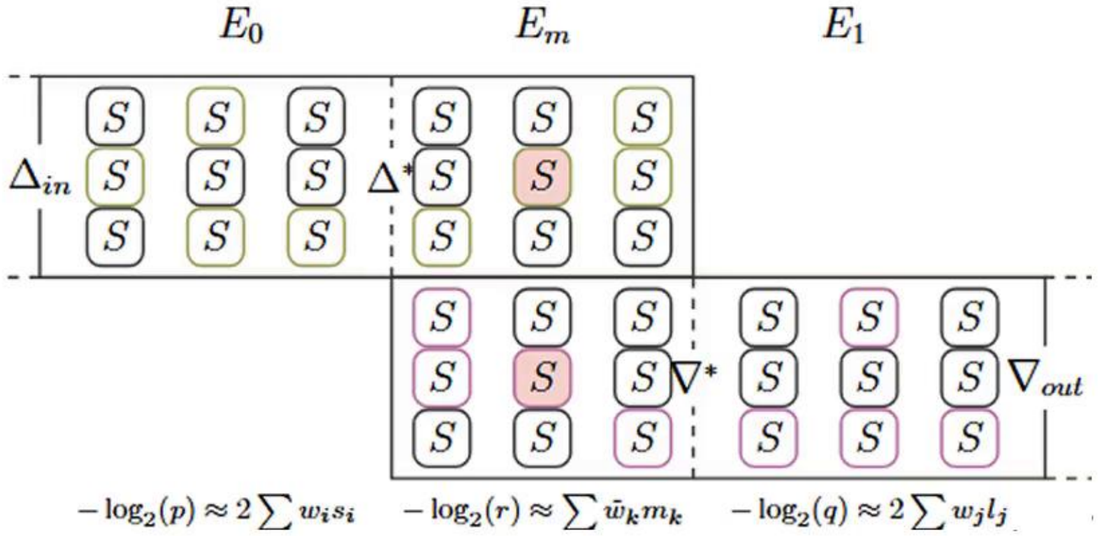
Non-Linear Operator

$E_\uparrow : \Delta_\uparrow^{in} = \Delta_\uparrow^{out}$

$E_\downarrow : \Delta_\downarrow^{in} = \Delta_\downarrow^{out}$

$E_m : \Delta_\uparrow^{in} \wedge \Delta_\downarrow^{in}$

Image from [13]



Roma Tre

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

$$\mathbf{E_\uparrow \setminus E_0} : \Delta_\uparrow^{out} = \bigvee_{\Delta_\uparrow^{in} \in t_I} \Delta_\uparrow^{in}$$

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

$\mathbf{E_\uparrow \setminus E_0} : \Delta_\uparrow^{out} = \bigvee_{\Delta_\uparrow^{in} \in t_I} \Delta_\uparrow^{in}$

$\mathbf{E_\downarrow \setminus E_1}$:

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

$\mathbf{E_\uparrow \setminus E_0} : \Delta_\uparrow^{out} = \bigvee_{\Delta_\uparrow^{in} \in t_I} \Delta_\uparrow^{in}$

$\mathbf{E_\downarrow \setminus E_1}$:

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

$\mathbf{E_\uparrow \setminus E_0} : \Delta_\uparrow^{out} = \bigvee_{\Delta_\uparrow^{in} \in t_I} \Delta_\uparrow^{in}$

$\mathbf{E_\downarrow \setminus E_1}$:  TAGADA doesn't invert the graph

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

$\mathbf{E_\uparrow \setminus E_0} : \Delta_\uparrow^{out} = \bigvee_{\Delta_\uparrow^{in} \in t_I} \Delta_\uparrow^{in}$

$\mathbf{E_\downarrow \setminus E_1}$:  TAGADA doesn't invert the graph

1. **Key-Schedule Operator**: As forward propagation

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

$\mathbf{E_\uparrow \setminus E_0}$: $\Delta_\uparrow^{out} = \bigvee_{\Delta_\uparrow^{in} \in t_I} \Delta_\uparrow^{in}$

$\mathbf{E_\downarrow \setminus E_1}$:  TAGADA doesn't invert the graph

1. **Key-Schedule Operator**: As forward propagation
2. **Invertible Operator**: Let f be the operator

$$\bigoplus_{x_w \in f_{X_w}} x_w \oplus \bigoplus_{x_u \in f_{X_u}} x_u = f_Y$$

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

$\mathbf{E_\uparrow \setminus E_0} : \Delta_\uparrow^{out} = \bigvee_{\Delta_\uparrow^{in} \in t_I} \Delta_\uparrow^{in}$

$\mathbf{E_\downarrow \setminus E_1}$:  TAGADA doesn't invert the graph

1. **Key-Schedule Operator**: As forward propagation
2. **Invertible Operator**: Let f be the operator

$$\bigoplus_{x_w \in f_{X_w}} x_w \oplus \bigoplus_{x_u \in f_{X_u}} x_u = f_Y$$

`if can_be_evaluated`(f) :

`then` $x_u = \bigoplus_{x_w \in f_{X_w}} x_w \oplus \bigoplus_{y \in f_Y} y$

Boomerang for TAGADA - Step 1 Constraint

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Linear Operator [13]:

$\mathbf{E_\uparrow \setminus E_0}$: $\Delta_\uparrow^{out} = \bigvee_{\Delta_\uparrow^{in} \in t_I} \Delta_\uparrow^{in}$

$\mathbf{E_\downarrow \setminus E_1}$:  TAGADA doesn't invert the graph

1. **Key-Schedule Operator**: As forward propagation
2. **Invertible Operator**: Let f be the operator

$$\bigoplus_{x_w \in f_{X_w}} x_w \oplus \bigoplus_{x_u \in f_{X_u}} x_u = f_Y$$

$$\left.\begin{array}{l} \texttt{if can\_be\_evaluated}(f): \\ \texttt{then } x_u = \bigoplus_{x_w \in f_{X_w}} x_w \oplus \bigoplus_{y \in f_Y} y \end{array}\right\} \texttt{inversion algorithm}$$

Boomerang for TAGADA - Step 1 Output

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Output.

Boomerang for TAGADA - Step 1 Output

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Output. A JSON file with:

$E_\uparrow$ with $-\log_2(p)$ computed by the active S-box in E_0

$E_\downarrow$ with $-\log_2(q)$ computed by the active S-box in E_1

$-\log_2(r)$, i.e. **penalty probability**, and computed by the common active S-box in $E_\uparrow$ and $E_\downarrow$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize p^2q^2r

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize p^2q^2r

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate E_0 with original Step 2:

- Instatiate E_1 with original Step 2:

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate E_0 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (E_0, -\log(p))$$

- Instatiate E_1 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (E_1, -\log(q))$$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate E_0 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (E_0, -\log(p))$$

- Instatiate E_1 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (E_1, -\log(q))$$

- Compute r:

1. Extract the output of E_0: $\boldsymbol{\Delta}_\uparrow^{\mathbf{in}}$
2. Extract the input of E_1: $\nabla_\downarrow^{\mathbf{out}}$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate E_0 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (E_0, -\log(p))$$

- Instatiate E_1 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (E_1, -\log(q))$$

- Compute r:

1. Extract the output of E_0: $\boldsymbol{\Delta}_\uparrow^{\mathbf{in}}$

2. Extract the input of E_1: $\nabla_\downarrow^{\mathbf{out}}$

$\Big\}$ $\texttt{compute\_r\_empirically}(\Delta_\uparrow^{in}, \nabla_\downarrow^{out})$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize p^2q^2r

Case 2. $r_0 = 0$ and $r_1 = 0$

- ~~Instatiate E_0 with original Step 2:~~

  ~~$\texttt{step\_2\_tagada}(E_\uparrow, -\log(p_1'))$ $(E_0, -\log(p))$~~

- ~~Instatiate E_1 with original Step 2:~~

  ~~$\texttt{step\_2\_tagada}(E_\downarrow, -\log(q_1'))$ $(E_1, -\log(q))$~~

- Compute r:

  1. Extract the input of $E_\uparrow$: $\boldsymbol{\Delta}_\uparrow^*$

  2. Extract the output of $E_\downarrow$: $\nabla_\downarrow^*$

  } $\texttt{compute\_r\_empirically}(\Delta_\uparrow^*, \nabla_\downarrow^*)$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize p^2q^2r

Case 3. $r_0 > 0$ and $r_1 = 0$

- Instatiate E_0 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (E_0, -\log(p))$$

- Instatiate E_1 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (E_1, -\log(q))$$

- Compute r:

1. Extract the output of E_0: $\boldsymbol{\Delta}_\uparrow^{\mathbf{in}}$
2. Extract the output of $E_\downarrow$: $\nabla_\downarrow^*$

} $\texttt{compute\_r\_empirically}(\Delta_\uparrow^{in}, \nabla_\downarrow^*)$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize p^2q^2r

Case 4. $r_0 = 0$ and $r_1 > 0$

- Instatiate E_0 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (E_0, -\log(p))$$

- Instatiate E_1 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (E_1, -\log(q))$$

- Compute r:

1. Extract the input of $E_\uparrow$: $\boldsymbol{\Delta}_\uparrow^*$
2. Extract the input of E_1: $\nabla_\downarrow^{\mathbf{out}}$

$\Big\}$ $\texttt{compute\_r\_empirically}(\Delta_\uparrow^*, \nabla_\downarrow^{out})$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize p^2q^2r

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate E_0 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (E_0, -\log(p))$$

- Instatiate E_1 with original Step 2:

$$\texttt{step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (E_1, -\log(q))$$

- Compute r:

1. Extract the output of E_0: $\boldsymbol{\Delta}_\uparrow^{\mathbf{in}}$

2. Extract the input of E_1: $\nabla_\downarrow^{\mathbf{out}}$

$\Big\}$ $\texttt{compute\_r\_empirically}(\Delta_\uparrow^{in}, \nabla_\downarrow^{out})$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate $(E_0)_i$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (\{(E_0)_i\}_i, -\log(p))$$

- Instatiate $(E_1)_j$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (\{(E_1)_j\}_j, -\log(q))$$

- Compute r:

1. Extract the output of $(E_0)_i$: $(\boldsymbol{\Delta}_\uparrow^{\mathbf{in}})_\mathbf{i}$

2. Extract the input of $(E_1)_j$: $(\nabla_\downarrow^{\mathbf{out}})_\mathbf{j}$

$\Big\}$ $\texttt{compute\_r\_empirically}(\Delta_\uparrow^{in}, \nabla_\downarrow^{out})$

Boomerang for TAGADA - Step 2

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate $(E_0)_i$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (\{(E_0)_i\}_i, -\log(p))$$

- Instatiate $(E_1)_j$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (\{(E_1)_j\}_j, -\log(q))$$

- Compute r:

1. Extract the output of $(E_0)_i$: $(\boldsymbol{\Delta}_\uparrow^{\mathbf{in}})_\mathbf{i}$

2. Extract the input of $(E_1)_j$: $(\nabla_\downarrow^{\mathbf{out}})_\mathbf{j}$

$\Big\}$ $\forall i, j$ $\texttt{compute\_r\_empirically}((\Delta_\uparrow^{in})_i, (\nabla_\downarrow^{out})_j)$

Boomerang for TAGADA - Optimization

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate $(E_0)_i$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (\{(E_0)_i\}_i, -\log(p))$$

- Instatiate $(E_1)_j$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (\{(E_1)_j\}_j, -\log(q))$$

- Compute r:

1. Extract the output of $(E_0)_i$: $(\boldsymbol{\Delta}_\uparrow^{\mathbf{in}})_\mathbf{i}$

2. Extract the input of $(E_1)_j$: $(\nabla_\downarrow^{\mathbf{out}})_\mathbf{j}$

$\Big\}$ $\forall i, j$ $\texttt{compute\_r\_empirically}((\Delta_\uparrow^{in})_i, (\nabla_\downarrow^{out})_j)$

Boomerang for TAGADA - Optimization

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate $(E_0)_i$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (\{(E_0)_i\}_i, -\log(p))$$

- Instatiate $(E_1)_j$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (\{(E_1)_j\}_j, -\log(q))$$

- **Filter Step:**

$$\texttt{filter}(\{(E_0)_i\}_i) = \{(\tilde{E}_0)_i\}_i \mid \ (\Delta_\uparrow^{in})_i \neq (\Delta_\uparrow^{in})_k \ \forall i \neq k\}$$

$$\texttt{filter}(\{(E_1)_j\}_j) = \{(\tilde{E}_1)_j\}_i \mid \ (\nabla_\downarrow^{out})_j \neq (\nabla_\downarrow^{out})_h \ \forall j \neq h\}$$

- Compute r:

1. Extract the output of $(\tilde{E}_0)_i$: $(\mathbf{\Delta_\uparrow^{in}})_\mathbf{i}$

2. Extract the input of $(\tilde{E}_1)_j$: $(\mathbf{\nabla_\downarrow^{out}})_\mathbf{j}$

$\Big\}$ $\forall i, j$ $\texttt{compute\_r\_empirically}((\Delta_\uparrow^{in})_i, (\nabla_\downarrow^{out})_j)$

Boomerang for TAGADA - Optimization

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize p^2q^2r

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate $(E_0)_i$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (\{(E_0)_i\}_i, -\log(p))$$

- Instatiate $(E_1)_j$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (\{(E_1)_j\}_j, -\log(q))$$

- **Heuristic Filter Step:**

$$\texttt{score}(\{(E_0)_i, (E_1)_j\}_{i,j}) = \{((E_0)_i, (E_1)_j) \mid f((\Delta_\uparrow^{in})_i, (\nabla_\downarrow^{out})_j) \geq \alpha\}$$

$$\text{where } f((\Delta_\uparrow^{in})_i, (\nabla_\downarrow^{out})_j) = \mathbf{(F)BCT}((\Delta_\uparrow^{in})_i, (\nabla_\downarrow^{out})_j)$$

- Compute r:

1. Extract the output of $(\tilde{E}_0)_i$: $(\mathbf{\Delta_\uparrow^{in}})_\mathbf{i}$

2. Extract the input of $(\tilde{E}_1)_j$: $(\mathbf{\nabla_\downarrow^{out}})_\mathbf{j}$

$\Big\}$ $\forall i, j$ $\texttt{compute\_r\_empirically}((\Delta_\uparrow^{in})_i, (\nabla_\downarrow^{out})_j)$

Boomerang for TAGADA - Optimization

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize $p^2 q^2 r$

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate $(E_0)_i$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (\{(E_0)_i\}_i, -\log(p))$$

- Instatiate $(E_1)_j$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (\{(E_1)_j\}_j, -\log(q))$$

- Compute r:

1. Extract the output of $(\tilde{E}_0)_i$: $(\boldsymbol{\Delta}^{\mathbf{in}}_\uparrow)_\mathbf{i}$

2. Extract the input of $(\tilde{E}_1)_j$: $(\nabla^{\mathbf{out}}_\downarrow)_\mathbf{j}$

$\Big\}$ $\forall i, j$ $\texttt{compute\_r\_empirically}((\Delta^{in}_\uparrow)_i, (\nabla^{out}_\downarrow)_j)$

Boomerang for TAGADA - Optimization

Let $E^K = E_1^K \circ E_m^K \circ E_0^K$, $E_\uparrow = E_m \circ E_0$ and $E_\downarrow = E_1 \circ E_m$ for the rounds $r_0 + r_m = k_1$ and $r_1 + r_m$ respectively

Goal. Find the differentials for E_0 and E_1 to maximize p^2q^2r

Case 1. $r_0 > 0$ and $r_1 > 0$

- Instatiate $(E_0)_i$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\uparrow, -\log(p_T)) = (\{(E_0)_i\}_i, -\log(p))$$

- Instatiate $(E_1)_j$ with Step 2:

$$\texttt{list\_step\_2\_tagada}(E_\downarrow, -\log(q_T)) = (\{(E_1)_j\}_j, -\log(q))$$

- Compute r:

1. Extract the output of $(\tilde{E}_0)_i$: $(\mathbf{\Delta_\uparrow^{in}})_\mathbf{i}$

2. Extract the input of $(\tilde{E}_1)_j$: $(\nabla_\downarrow^{\mathbf{out}})_\mathbf{j}$

$\Big\}$ $\forall i, j$ $\texttt{compute\_r\_empirically}((\Delta_\uparrow^{in})_i, (\nabla_\downarrow^{out})_j)$

- **Stop Criteria**:

$$p_{step2}^2 q_{step2}^2 r_{emp} < p_{trunc}^2 q_{trunc}^2 r_{trunc}$$

Boomerang for TAGADA - Results

Boomerang for TAGADA - Results

Cipher	Rnds	p	q	r	p^2q^2r	Reference	Time (in minutes)
SKINNY-64	11				$2^{-80.0}$ ($2^{-59.23}$)	[14]	161m †
	11	$2^{-8.00}$	$2^{-8.00}$	$2^{-24.00}$	$2^{-56.00}$	**Our Work**	62m
SKINNY-128	13				2^{-122} ($2^{-112.53}$)	[14]	700m †
	13	$2^{-22.00}$	$2^{-22.00}$	$2^{-24.00}$	$2^{-112.00}$	**Our Work**	713m
	14				2^{-158} ($2^{-128.52}$)	[14]	547m †
	14	$2^{-22.00}$	$2^{-42.00}$	$2^{-24.00}$	$2^{-152.00}$	**Our Work**	3600m
TWINE	15	2^{-8}	2^{-8}	$2^{-19.03}$	$2^{-51.03}$	[13]	1m
	15	$2^{-4.00}$	$2^{-8.00}$	$2^{-24.00}$	$2^{-48.00}$	**Our Work**	271m
	16	2^{-8}	2^{-8}	$2^{-26.04}$	$2^{-58.04}$	[13]	305m
	16	$2^{-8.00}$	$2^{-8.00}$	$2^{-24.00}$	$(2^{-56.00})^b$	**Our Work**	371m
	16	$2^{-8.00}$	$2^{-8.00}$	$2^{-23.00}$	$(2^{-55.00})^a$	**Our Work**	3180m
WARP	21	2^{-10}	2^{-19}	$2^{-26.55}$	$2^{-84.55}$	[15]	1m
					2^{-104}	[16]	1270m
	21	$2^{-14.00}$	$2^{-20.00}$	$2^{-23.00}$	$2^{-91.00}$	**Our Work**	105m
	22	2^{-16}	2^{-19}	$2^{-26.55}$	$2^{-96.55}$	[15]	180m
					2^{-120}	[16]	3014m
	22	$2^{-16.00}$	$2^{-20.00}$	$2^{-24.00}$	$2^{-96.00}$	**Our Work**	361m
	23	$2^{-24.00}$	$2^{-20.00}$	$2^{-27.59}$	$2^{-115.59}$	[15]	563m
	23	$2^{-16.00}$	$2^{-22.00}$	$2^{-24.00}$	$2^{-100.00}$	**Our Work**	2400m
MIDORI64	8	$2^{-14.00}$	$2^{-8.00}$	$2^{-16.08}$	$2^{-60.08}$	**Our Work**	1094m
MIDORI128	9	$2^{-42.00}$	$2^{-14.00}$	$2^{-4.77}$	$2^{-116.77}$		1980m

$(\cdot)^a$ means $\alpha = 0.1$,$(\cdot)^b$ means $\alpha = 0.5$

Exectuted on **Grid5000** gros cluster, with node equipped with four **Intel Xeon Gold 6240L** CPUs (18 cores per CPU, 2.60 GHz), used a single CPU (18 cores)

Roma Tre

Conclusion and Open Question

Conclusion and Open Question

Conclusion:

1. We showed that boomerang cryptanalysis can be significantly automated through CP techniques.
2. The proposed framework efficiently models and searches for boomerang trails on a wide range of primitives.
3. Results demonstrate that automation can reduce the manual effort required for cryptanalytic evaluations.

Conclusion and Open Question

Conclusion:

1. We showed that boomerang cryptanalysis can be significantly automated through CP techniques.
2. The proposed framework efficiently models and searches for boomerang trails on a wide range of primitives.
3. Results demonstrate that automation can reduce the manual effort required for cryptanalytic evaluations.

Open Question:

1. How can related-key boomerang distinguishers be automatically modeled and evaluated within the framework?
2. Can the framework be extended beyond distinguishers to support fully automated key-recovery attacks?
3. How can automated tools provide tight and reliable security bounds by unifying differential, boomerang analyses?

'The core of automated cryptanalysis lies in the ability to translate the linear and non-linear properties of a cipher into a mathematical model that can be interpreted by a solver', ChatGPT

Merci de votre attention

Grazie per l'attenzione

Thank you for the attention

Bibliography

[1] E. Biham and A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, 1991.

[2] R. Brunelli, M. Minier, and L. Rouquette, *Integrating boomerang into TAGADA*, 2026.

[3] R. Brunelli, et al. *Generic Partial Decryption as Feature Engineering for Neural Distinguishers*, 2025

[4] F. Delobel, et al. *A CP-Based Automatic Tool for Instantiating Trun- cated Differential Characteristics*, 2024

[5] L. R. Knudsen, *Truncated and higher order differentials*, 1995

[6] N. Nethercote, et al. *Minizinc: Towards a standard cp modelling language*, 2007

[7] Y. Zhou and H. Kjellerstrand, *The picat-sat compiler*, 2016

[8] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2026

[9] T. Cuvelier, et al. *OR-Tools Vehicle Routing Solver: a Generic Constraint-Programming Solver...*, 2023

[10] C. Prud'homme, J. Fages, and X. Lorca, *Choco Solver Documentation*, 2016

[11] D. Wagner, *The boomerang attack*, 1999

[12] O. Dunkelman, et al. *A practical-time related-key attack on the kasumi cryptosystem used in gsm...*, 2014

Bibliography

[13] H. Hadipour, et al. *Throwing boomerangs into feistel structures: Application to clefia, warp...*, 2022

[14] S. Delaune, P. Derbez, and M. Vavrille, *Catching the Fastest Boomerangs: Application to SKINNY*, 2020

[15] H. Hadipour, N. Bagheri, and L. Song, *Improved Rectangle Attacks on SKINNY and CRAFT* 2021

[16] V. Lallemand, M. Minier, and L. Rouquette, *Automatic Search of Rectangle Attacks on Feistel Ciphers...*, 2022