CONSTRAINT PROGRAMMING AND CRYPTANALYSIS

Improving scalability and reusability of differential cryptanalysis models using constraint programming

Virginie Lallemand[1], Marine Minier[1], **Loïc Rouquette**[1,2], Christine Solnon[2]

Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

CITI, INRIA, INSA Lyon, Villeurbanne, France

December, 15th 2022

Slides: 62

Table of contents

Context

Cryptography

From the Ancient Greek:

κρυπτός (kruptós, "hidden, secret") and

γράφειν (graphein, "to write")

Purposes

- data integrity,
- data authenticity,
- data confidentiality,
- non-repudiation.

The fields

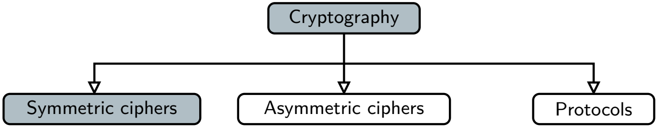


Figure 1 Overview of the field of cryptology [PP10]

The fields

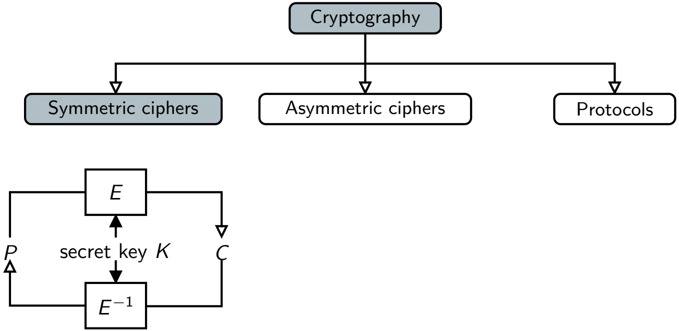


Figure 1 Overview of the field of cryptology [PP10]

The fields

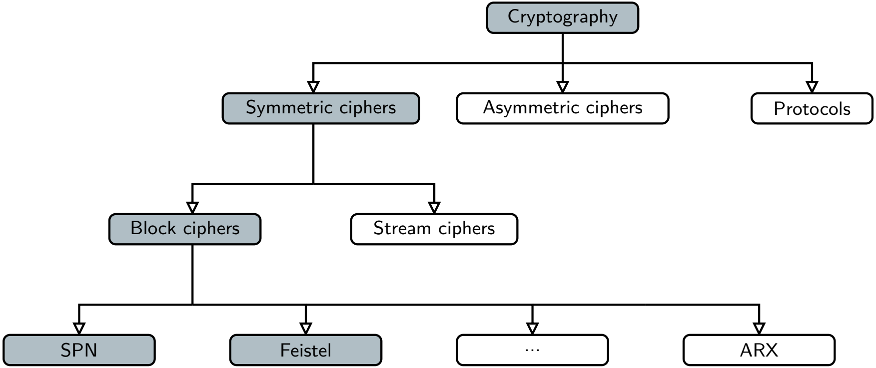


Figure 1 Overview of the field of cryptology [PP10]

Cryptanalysis

From the Greek:

$\kappa\rho\upsilon\pi\tau$ós (kruptós, "hidden, secret") and

$\alpha\nu\alpha\lambda\upsilon\epsilon\iota\nu$ (analýein, "to analyze")

Purposes

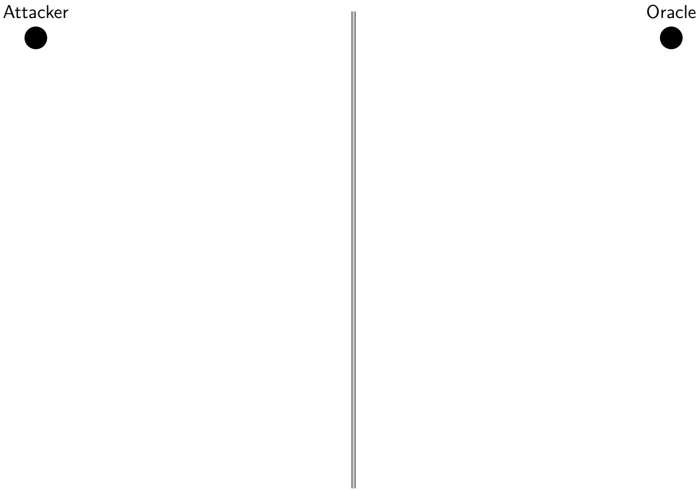
- Analyze ciphers in order to detect and exploit weaknesses to mount attacks

Is a cipher weak?

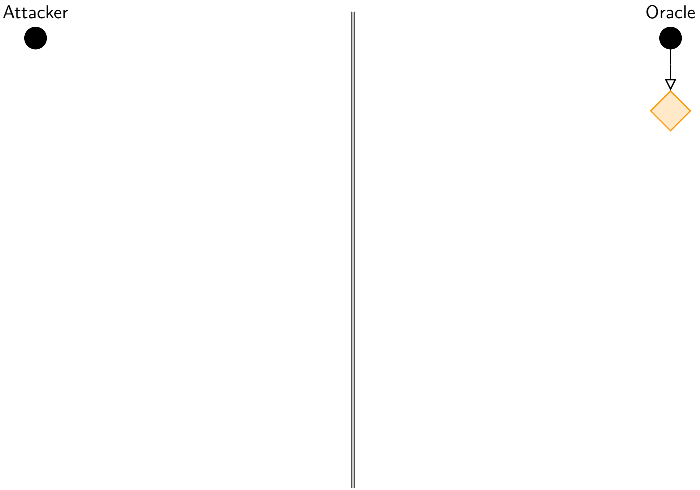
A cipher is weak if it is possible to distinguish it from a random permutation.

under attack conditions

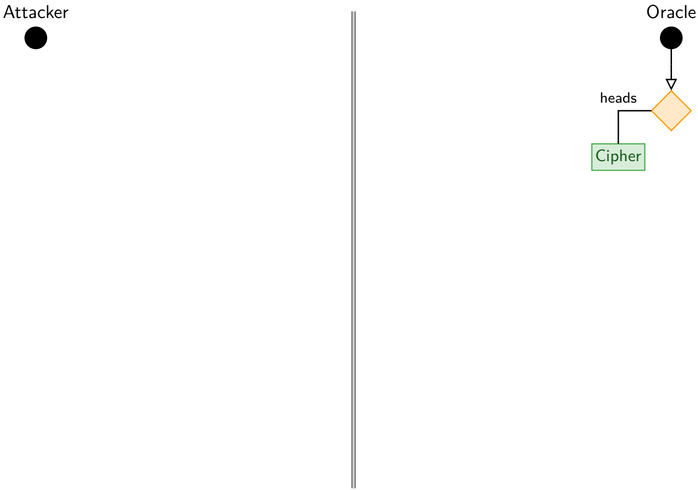
Distinguishability?



Distinguishability?



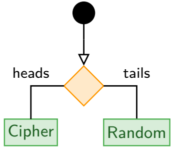
Distinguishability?



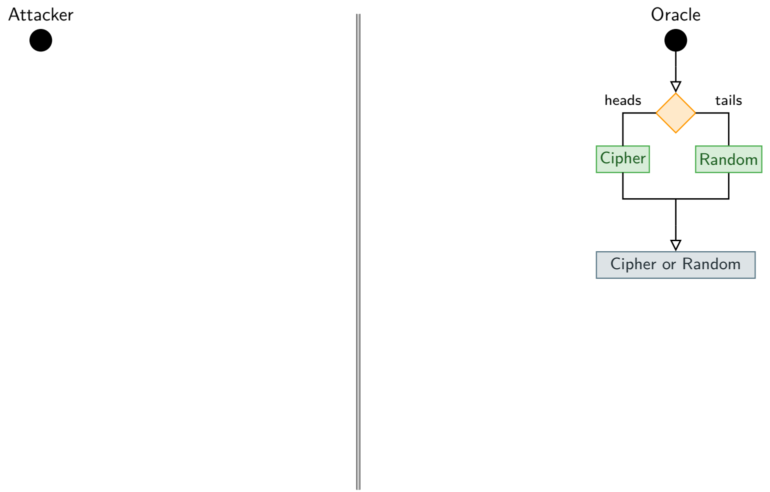
Distinguishability?

Attacker

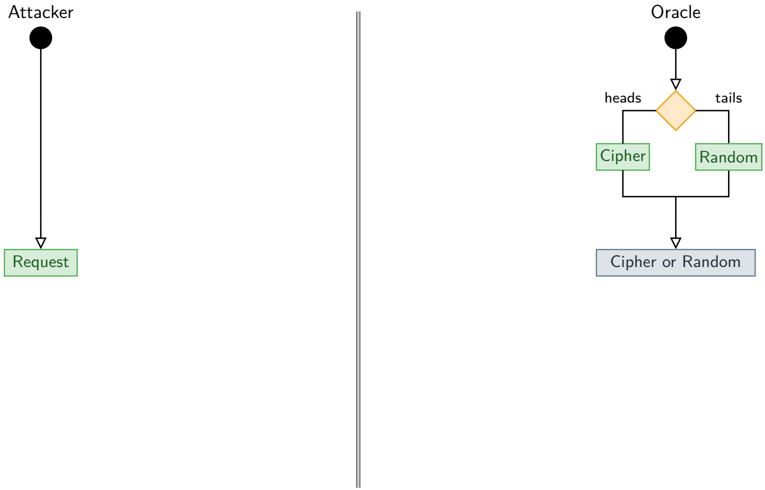
Oracle



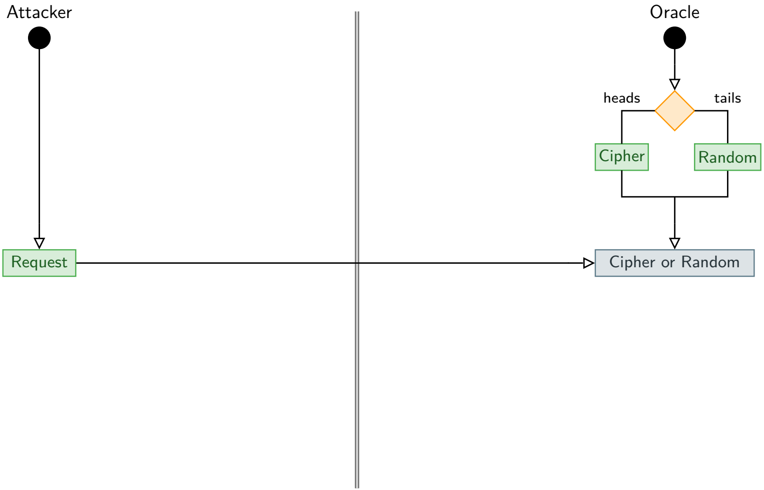
Distinguishability?



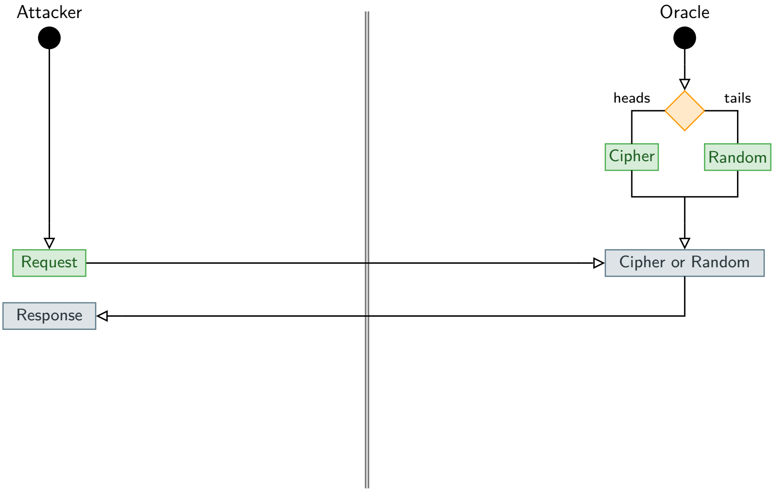
Distinguishability?



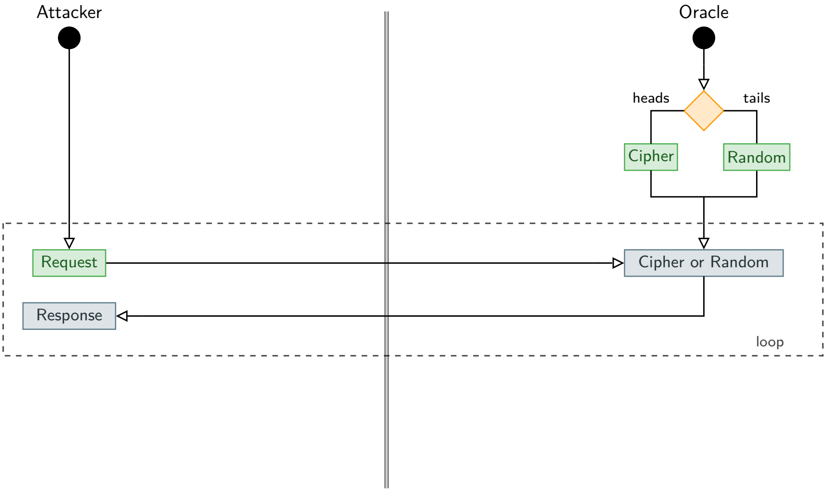
Distinguishability?



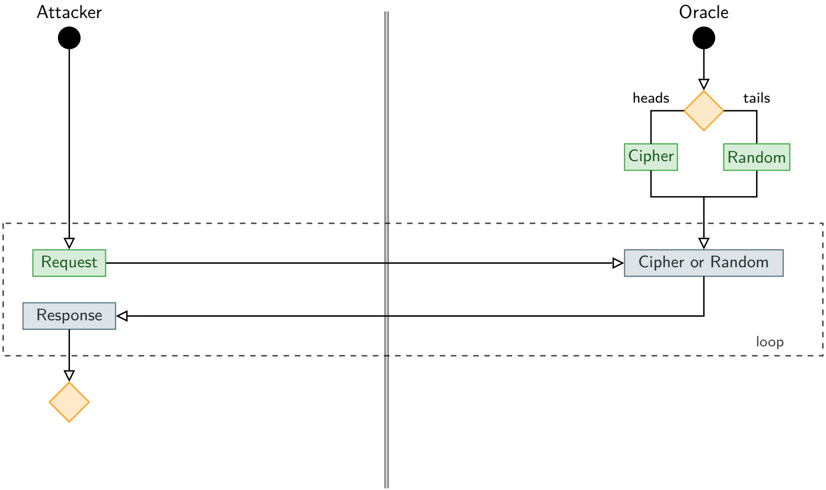
Distinguishability?



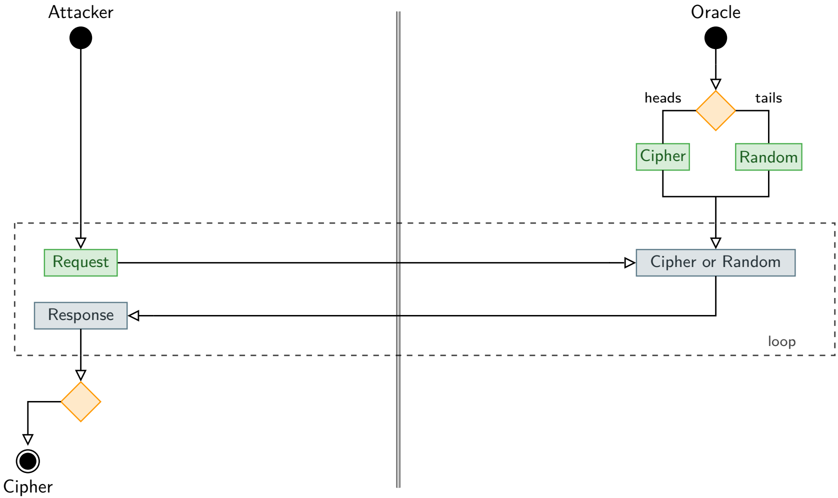
Distinguishability?



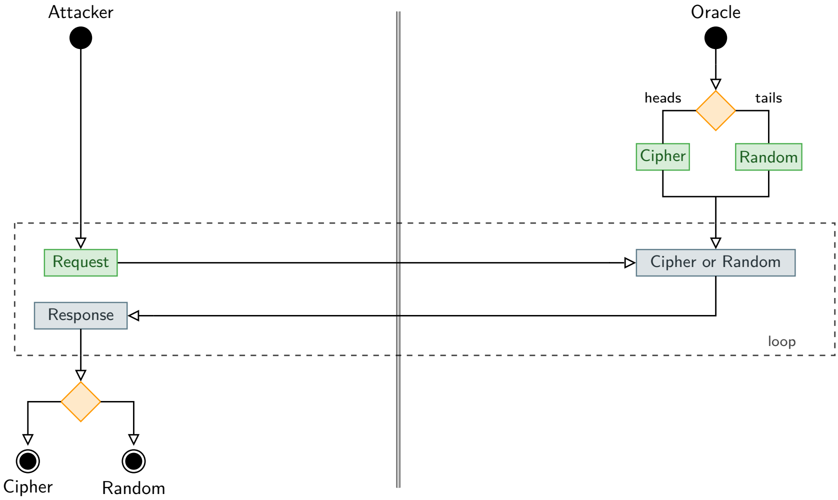
Distinguishability?



Distinguishability?



Distinguishability?



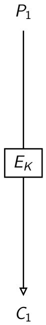
Differential cryptanalysis [Biham and Shamir 1991]

- Based on differential distinguishers

P_1

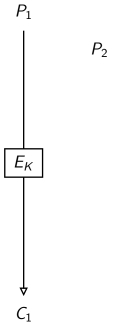
Differential cryptanalysis [Biham and Shamir 1991]

- Based on differential distinguishers



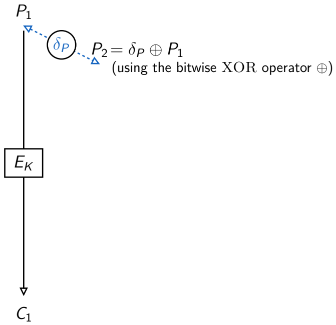
Differential cryptanalysis [Biham and Shamir 1991]

- Based on differential distinguishers



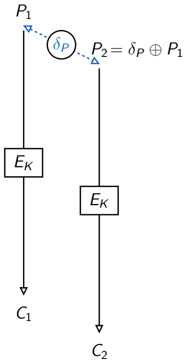
Differential cryptanalysis [Biham and Shamir 1991]

- Based on differential distinguishers



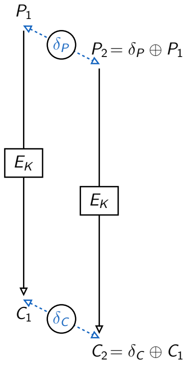
Differential cryptanalysis [Biham and Shamir 1991]

- Based on differential distinguishers



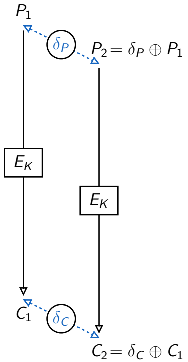
Differential cryptanalysis [Biham and Shamir 1991]

- Based on differential distinguishers



Differential cryptanalysis [Biham and Shamir 1991]

- Based on differential distinguishers



$$Pr[\delta_P \rightsquigarrow \delta_C] = \ ?$$

Compute the differential distinguisher probability?

- Empirically
  - Generate random pairs of messages P_1, P_2 with $P_2 = P_1 \oplus \delta_P$
  - Cipher both messages
  - Count the number of pairs with $E_K(P_1) \oplus E_K(P_2) = \delta_C$ against the number of tried pairs.
- Using approximations

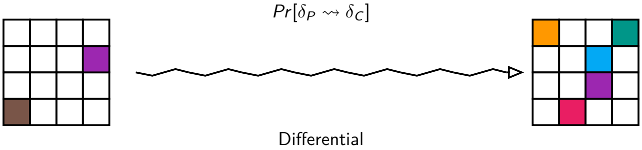
Compute the differential distinguisher probability?

- Empirically Too slow
  - Generate random pairs of messages P_1, P_2 with $P_2 = P_1 \oplus \delta_P$
  - Cipher both messages
  - Count the number of pairs with $E_K(P_1) \oplus E_K(P_2) = \delta_C$ against the number of tried pairs.
- Using approximations

Compute the differential distinguisher probability?

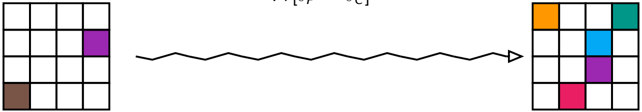
- Empirically Too slow
  - Generate random pairs of messages P_1, P_2 with $P_2 = P_1 \oplus \delta_P$
  - Cipher both messages
  - Count the number of pairs with $E_K(P_1) \oplus E_K(P_2) = \delta_C$ against the number of tried pairs.
- **Using approximations**

From differential distinguisher to differential characteristic

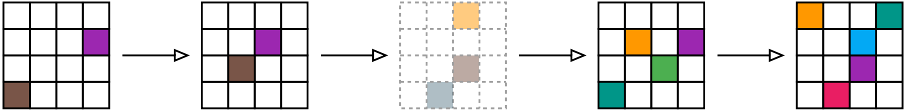


From differential distinguisher to differential characteristic

$Pr[\delta_P \rightsquigarrow \delta_C]$



Differential



$Pr[\delta_P \to \delta_{R_0}] \quad \times \quad Pr[\delta_{R_0} \to \delta_{R_1}] \quad \times \quad \cdots \quad \times \quad Pr[\delta_{R_{n-1}} \to \delta_C]$

Differential characteristic

Computing round probabilities

Linear functions

$Pr[\delta_{in} \rightarrow \delta_{out}] = 1^a$ because $\delta_{out} = f(x) \oplus f(x \oplus \delta_{in}) = f(\delta_{in})$

[a] or 0 when the transition is wrong.

Non-Linear functions (S-Boxes)

For a function $f : \{0,1\}^n \rightarrow \{0,1\}^n$,

$$Pr[\delta_{in} \rightarrow \delta_{out}] = DDT(\delta_{in}, \delta_{out}) = \frac{\#\{x \in \{0,1\}^n \mid f(x) \oplus f(x \oplus \delta_{in}) = \delta_{out}\}}{2^n}$$

Special case: $Pr[\delta_{in} = 0 \rightarrow \delta_{out} = 0] = 1$
because $f(x) \oplus f(x \oplus 0) = 0$

$Pr[\delta_P \rightsquigarrow \delta_C] \approx$ Product of the round probabilities $=$ Product of active S-Box transition probs.

Truncated differential characteristic [Knudsen 1995]

Abstract the differential characteristic

Each differential n-bit word δ_X is abstracted by a differential Boolean Δ_X with:

$$\Delta_X = 0 \iff \delta_X = 0$$
$$\Delta_X = 1 \iff \delta_X \in [1; 2^n[$$

Active S-Boxes

An S-Box with a non-null input difference

Two Step solving process

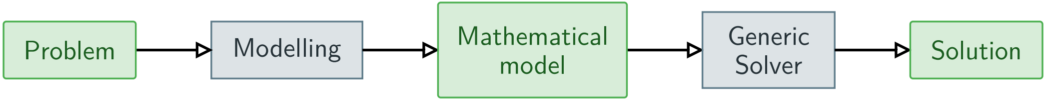
1. Step-1
   1.1 Step-1 Opt: Minimize the number of active S-Boxes (*obj*) in a truncated differential;
   1.2 Step-1 Enum: Enumerate every truncated differential with *obj* active S-Boxes.
2. Step-2 Opt: Search for the corresponding differential characteristic with the highest probability p_{max};

   - if p_{max} may be improved, increment *obj* and go to 1.2.
3. Clustering: Try to improve the distinguisher probability by aggregating differential characteristics.
4. Step-3: Compute the attack complexity using the optimal differential characteristic found with Step-2 Opt.

Find the differential distinguisher with the highest probability?

- Using dedicated algorithms [FJP13; BKN09]
  - Hard to write
  - Hard to adapt
- Using generic solvers

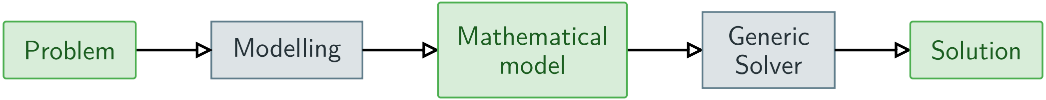
Constraint Programming



Modelling

A Constraint Satisfaction Problem (CSP) is defined by a triplet (X, D, C) with:

- X The set of variables,
- D The domain of each variables noted $D(x)$ with $x \in X$,
- C The set of constraints on the variables



Modelling

A Constrained Optimization Problem (COP) is defined by a quadruplet (X, D, C, f) with:

- X The set of variables,
- D The domain of each variables noted $D(x)$ with $x \in X$,
- C The set of constraints on the variables,
- f The objective function which is to optimize

The variants

Boolean SATisfiability

Restricted to Boolean variables and Boolean formulae

Integer Linear Programming

Restricted to Integer variables and linear inequations

Constraint Programming

Restricted to solver implementations

The variants

$$\Delta_A + \Delta_B + \Delta_C \neq 1 \text{ with } D(\Delta_A) = D(\Delta_B) = D(\Delta_C) = \{0, 1\}$$

$$\Delta_A + \Delta_B + \Delta_C \neq 1 \text{ with } D(\Delta_A) = D(\Delta_B) = D(\Delta_C) = \{0, 1\}$$

Boolean SATisfiability

$$\overline{\Delta_A \Delta_B \Delta_C} \vee \overline{\Delta_A} \Delta_B \Delta_C \vee \overline{\Delta_A} \Delta_B \overline{\Delta_C} \vee$$
$$\Delta_A \Delta_B \overline{\Delta_C} \vee \Delta_A \Delta_B \Delta_C$$

Integer Linear Programming

$$D(tmp) = \{0, 1\}$$
$$\Delta_A + \Delta_B + \Delta_C + 3 \times tmp \leq 3$$
$$-\Delta_A - \Delta_B - \Delta_C - 2 \times tmp \leq 2$$

$$\Delta_A + \Delta_B + \Delta_C \neq 1 \text{ with } D(\Delta_A) = D(\Delta_B) = D(\Delta_C) = \{0, 1\}$$

Constraint Programming

$$sum(\{\Delta_A, \Delta_B, \Delta_C\}) \neq 1$$

or

$$(\Delta_A, \Delta_B, \Delta_C) \in T_{\sum \neq 1} \text{ with } T_{\sum \neq 1} = \{(0,0,0), (0,1,1), (1,0,1), (1,1,0), (1,1,1)\}$$

Same but different

- Each model of one paradigm can be translated into a model of another paradigm
- Different ways of modelling
- Different solving techniques
- Different strengths and weaknesses

Contributions

Differential cryptanalysis of Rijndael

Loïc Rouquette, David Gerault, Marine Minier, and Christine Solnon. "And Rijndael? Automatic Related-key Differential Analysis of Rijndael". In: *AfricaCrypt 2022 - 13th International Conference on Cryptology AfricaCrypt*. Fes, Morocco, July 2022

Global Constraint Abstract XOR

Loïc Rouquette and Christine Solnon. "abstractXOR: A global constraint dedicated to differential cryptanalysis". en. In: *Principles and Practice of Constraint Programming*. Ed. by Helmut Simonis. Vol. 12333. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 566–584. (Visited on 04/30/2021)

Boomerang cryptanalysis of Rijndael

Not yet published.

Automatic Search of Rectangle Attacks on WARP

Virginie Lallemand, Marine Minier, and Loïc Rouquette. "Automatic Search of Rectangle Attacks on Feistel Ciphers: Application to WARP". In: *IACR Trans. Symmetric Cryptol.* 2022.2 (2022), pp. 113–140

Global Constraint Abstract XOR

Computing the differential characteristic of Midori [Ban+15]

- Created as an alternative to AES [01] for ligthweight components
- Two variants with 64 and 128-bit text
- 128-bit key

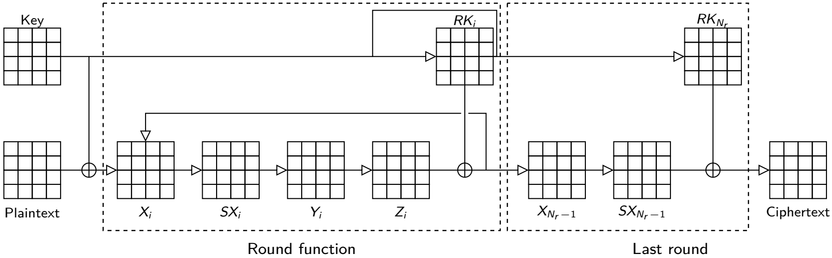


FIGURE 2 Schema of Midori

All δ variables are integer variables in $[0; 255]$.

$$Maximize \sum_{i=0}^{r-1} \sum_{k=0}^{15} P[i, k]$$

$$\forall i \in [0, r[, \forall k \in [0; 15[,$$

$$(\delta_X[i, k], \delta_{SX}[i, k], P[i, k]) \in T_{SB_k}$$

$$\delta_Y[i, \pi(k)] = \delta_{SX}[i, k] \text{ with } \pi \text{ a given permutation}$$

$$\delta_Z[i, k] \oplus \delta_Y[i, (k+4)\%16] \oplus \delta_Y[i, (k+8)\%16] \oplus \delta_Y[i, (k+12)\%16] = 0$$

$$\delta_Z[i, k] \oplus \delta_K[k] \oplus \delta_X[i+1, k] = 0$$

The truncated model

All Δ variables are Boolean variables in $[0;1]$.

$$Minimize \sum_{i=0}^{r-1} \sum_{k=0}^{15} \Delta_X[i,k]$$

$$\forall i \in [0,r[, \forall k \in [0;15[,$$

$$\Delta_X[i,k] = \Delta_{SX}[i,k]$$

$$\Delta_Y[i,\pi(k)] = \Delta_{SX}[i,k]$$

$$\Delta_Z[i,k] \odot \Delta_Y[i,(k+4)\%16] \odot \Delta_Y[i,(k+8)\%16] \odot \Delta_Y[i,(k+12)\%16] = 0$$

$$\Delta_Z[i,k] \odot \Delta_K[k] \odot \Delta_X[i+1,k] = 0$$

Implement the $\odot$ operator

	δ_A		δ_B		δ_C	Δ_A		Δ_B		Δ_C
	0	$\oplus$	0	=	0	0	$\odot$	0	=	0
$\forall\ \alpha > 0,$	α	$\oplus$	0	=	α	1	$\odot$	0	=	1
$\forall\ \alpha > 0,$	0	$\oplus$	α	=	α	0	$\odot$	1	=	1
$\forall\ \alpha,\ \beta > 0$ and $, \alpha \neq \beta$	α	$\oplus$	β	=	γ	1	$\odot$	1	=	1
$\forall\ \alpha > 0,$	α	$\oplus$	α	=	0	1	$\odot$	1	=	0

Abstract XOR

Implement the $\odot$ operator

	δ_A		δ_B		δ_C	Δ_A		Δ_B		Δ_C	$\sum_{\Delta_i}$
	0	$\oplus$	0	$=$	0	0	$\odot$	0	$=$	0	
$\forall\ \alpha > 0,$	α	$\oplus$	0	$=$	α	1	$\odot$	0	$=$	1	
$\forall\ \alpha > 0,$	0	$\oplus$	α	$=$	α	0	$\odot$	1	$=$	1	
$\forall\ \alpha,\ \beta > 0$ and $, \alpha \neq \beta$	α	$\oplus$	β	$=$	γ	1	$\odot$	1	$=$	1	
$\forall\ \alpha > 0,$	α	$\oplus$	α	$=$	0	1	$\odot$	1	$=$	0	

Implement the $\odot$ operator

	δ_A		δ_B		δ_C	Δ_A		Δ_B		Δ_C	$\sum_{\Delta_i}$
	0	$\oplus$	0	$=$	0	0	$\odot$	0	$=$	0	0
$\forall\ \alpha > 0$,	α	$\oplus$	0	$=$	α	1	$\odot$	0	$=$	1	2
$\forall\ \alpha > 0$,	0	$\oplus$	α	$=$	α	0	$\odot$	1	$=$	1	2
$\forall\ \alpha,\ \beta > 0$ and $, \alpha \neq \beta$	α	$\oplus$	β	$=$	γ	1	$\odot$	1	$=$	1	3
$\forall\ \alpha > 0$,	α	$\oplus$	α	$=$	0	1	$\odot$	1	$=$	0	2

Implement the $\odot$ operator

	δ_A		δ_B		δ_C	Δ_A		Δ_B		Δ_C	$\sum_{\Delta_i}$
	0	$\oplus$	0	$=$	0	0	$\odot$	0	$=$	0	0
$\forall\ \alpha > 0,$	α	$\oplus$	0	$=$	α	1	$\odot$	0	$=$	1	2
$\forall\ \alpha > 0,$	0	$\oplus$	α	$=$	α	0	$\odot$	1	$=$	1	2
$\forall\ \alpha,\ \beta > 0$ and $, \alpha \neq \beta$	α	$\oplus$	β	$=$	γ	1	$\odot$	1	$=$	1	3
$\forall\ \alpha > 0,$	α	$\oplus$	α	$=$	0	1	$\odot$	1	$=$	0	2

$$\sum_{\Delta_i} \neq 1$$

All Δ variables are Boolean variables in $[0;1]$.

$$Minimize \sum_{i=0}^{r-1} \sum_{k=0}^{15} \Delta_X[i,k]$$

$$\forall i \in [0,r[, \forall k \in [0;15[,$$

$$\Delta_X[i,k] = \Delta_{SX}[i,k]$$

$$\Delta_Y[i,\pi(k)] = \Delta_{SX}[i,k]$$

$$\Delta_Z[i,k] + \Delta_Y[i,(k+4)\%16] + \Delta_Y[i,(k+8)\%16] + \Delta_Y[i,(k+12)\%16] \neq 1$$

$$\Delta_Z[i,k] + \Delta_K[k] + \Delta_X[i+1,k] \neq 1$$

False positive Step-1 solutions

Exemple

$$
\begin{aligned}
& & \delta_A \oplus \delta_D \oplus \delta_E &= 0 \\
\delta_B \oplus \delta_C \oplus \delta_D \oplus \delta_E &= 0
\end{aligned}
$$

False positive Step-1 solutions

Abstraction

$$
\begin{array}{ccccccccc}
\Delta_A & & & + & \Delta_D & + & \Delta_E & \neq & 1 \\
& \Delta_B & + & \Delta_C & + & \Delta_D & + & \Delta_E & \neq & 1
\end{array}
$$

False positive Step-1 solutions

Abstraction

$$\Delta_A = 1 \qquad \Delta_B \in \{0,1\} \qquad \Delta_C = 0 \qquad \Delta_D = 1 \qquad \Delta_E = 1$$

$$\begin{array}{ccccccccccc} \Delta_A & & & & & + & \Delta_D & + & \Delta_E & \neq & 1 \\ & & \Delta_B & + & \Delta_C & + & \Delta_D & + & \Delta_E & \neq & 1 \end{array}$$

False positive Step-1 solutions

Abstraction

$$\begin{array}{ccccccccccc}
\Delta_A = 1 & & \Delta_B \in \{0,1\} & & \Delta_C = 0 & & \Delta_D = 1 & & \Delta_E = 1 & & \\
1 & & & & & + & 1 & + & 1 & = & 3 \\
 & & ? & + & 0 & + & 1 & + & 1 & \geq & 2
\end{array}$$

False positive Step-1 solutions

Abstraction

$$\Delta_A = 1 \quad \Delta_B \in \{0,1\} \quad \Delta_C = 0 \quad \Delta_D = 1 \quad \Delta_E = 1$$

$$\begin{array}{ccccccccc} 1 & & & & + & 1 & + & 1 & = & 3 \\ & ? & + & 0 & + & 1 & + & 1 & \geq & 2 \end{array}$$

May $\Delta_B = 0$ and 1 ?

False positive Step-1 solutions

Abstraction

$$\Delta_A = 1 \qquad \Delta_B \in \{0,1\} \qquad \Delta_C = 0 \qquad \Delta_D = 1 \qquad \Delta_E = 1$$

$$\begin{array}{ccccccccccc} 1 & & & & & + & 1 & + & 1 & = & 3 \\ & & ? & + & 0 & + & 1 & + & 1 & \geq & 2 \end{array}$$

May $\Delta_B = 0$ and 1 ?

The abstraction says **yes**

False positive Step-1 solutions

Abstraction

$$\Delta_A = 1 \qquad \Delta_B \in \{0,1\} \qquad \Delta_C = 0 \qquad \Delta_D = 1 \qquad \Delta_E = 1$$

$$\begin{array}{ccccccccc} 1 & & & & + & 1 & + & 1 & = & 3 \\ & ? & + & 0 & + & 1 & + & 1 & \geq & 2 \end{array}$$

May $\Delta_B = 0$ and 1 ?

The abstraction says **yes**

$$\begin{array}{ccccccccccc} & & \delta_A & \oplus & \delta_D & \oplus & \delta_E & = & 0 \\ \delta_B & \oplus & 0 & \oplus & \delta_D & \oplus & \delta_E & = & 0 \end{array} \iff \delta_A = \delta_D \oplus \delta_E \land \delta_A = \delta_B$$

Abstraction

$$\begin{array}{ccccccccccc} \Delta_A = 1 & & \Delta_B \in \{0,1\} & & \Delta_C = 0 & & \Delta_D = 1 & & \Delta_E = 1 & & \\ 1 & & & & & + & 1 & + & 1 & = & 3 \\ & & ? & + & 0 & + & 1 & + & 1 & \geq & 2 \end{array}$$

May $\Delta_B = 0$ and 1 ?

The abstraction says **yes**

$$\begin{array}{ccccccccc} & & \delta_A & \oplus & \delta_D & \oplus & \delta_E & = & 0 \\ \delta_B & \oplus & 0 & \oplus & \delta_D & \oplus & \delta_E & = & 0 \end{array} \iff \delta_A = \delta_D \oplus \delta_E \wedge \delta_A = \delta_B$$

$$\implies \Delta_A = \Delta_B \text{ and } \Delta_B = 1$$

False positive Step-1 solutions

Abstraction

$$\begin{array}{ccccccccccc} \Delta_A = 1 & & \Delta_B \in \{0,1\} & & \Delta_C = 0 & & \Delta_D = 1 & & \Delta_E = 1 & & \\ 1 & & & & & + & 1 & + & 1 & = & 3 \\ & & ? & + & 0 & + & 1 & + & 1 & \geq & 2 \end{array}$$

May $\Delta_B = 0$ and 1 ?

The abstraction says **yes**

$$\begin{array}{ccccccccccc} & & \delta_A & \oplus & \delta_D & \oplus & \delta_E & = & 0 \\ \delta_B & \oplus & 0 & \oplus & \delta_D & \oplus & \delta_E & = & 0 \end{array} \iff \delta_A = \delta_D \oplus \delta_E \wedge \delta_A = \delta_B$$

$$\implies \Delta_A = \Delta_B \text{ and } \Delta_B = 1$$

The initial equations say **no**

Advanced model [Gérault and Lafourcade 2016]

All Δ variables are Boolean variables in [0; 1].

$$Minimize \sum_{i=0}^{r-1} \sum_{k=0}^{15} \Delta_X[i,k]$$

$\forall i \in [0, r[, \forall k \in [0; 15[,$

$$\Delta_X[i,k] = \Delta_{SX}[i,k]$$
$$\Delta_Y[i,\pi(k)] = \Delta_{SX}[i,k]$$
$$\Delta_Z[i,k] + \Delta_Y[i,(k+4)\%16] + \Delta_Y[i,(k+8)\%16] + \Delta_Y[i,(k+12)\%16] \neq 1$$
$$\Delta_Z[i,k] + \Delta_K[k] + \Delta_X[i+1,k] \neq 1$$

$$\forall i \in [0, r-1[, \forall k \in [0;3], \sum_{j=0}^{3} \Delta_Y[i, j \times 4 + k] + \Delta_Z[i, j \times 4 + k] \in \{0,5,6,7,8\}$$

$$\forall D \in \{D_{K_j}, D_{Y_j}, D_{Z_j} : j \in [0;3]\}, \forall \{\delta_{B_1}, \delta_{B_2}\} \in D, diff_{\delta_{B_1},\delta_{B_2}} = diff_{\delta_{B_2},\delta_{B_1}}$$

$$\forall D \in \{D_{K_j}, D_{Y_j}, D_{Z_j} : j \in [0;3]\}, \forall \{\delta_{B_1}, \delta_{B_2}, \delta_{B_3}\} \in D, diff_{\delta_{B_1},\delta_{B_2}} + diff_{\delta_{B_2},\delta_{B_3}} + diff_{\delta_{B_1},\delta_{B_3}} \neq 1$$

$$\forall D \in \{D_{K_j}, D_{Y_j}, D_{Z_j} : j \in [0;3]\}, \forall \{\delta_{B_1}, \delta_{B_2}\} \in D, diff_{\delta_{B_1},\delta_{B_2}} + \Delta_{B_1} + \Delta_{B_2} \neq 1$$

$$\forall i_1, i_2 \in [0; r-1[^2, \forall k_1, k_2 \in [0;3]^2 : \sum_{j=0}^{3} (diff_{\delta_{Y_{i_1}},\delta_{Y_{i_2}}} \neq 0) + \sum_{j=0}^{3} (diff_{\delta_{Z_{i_1}},\delta_{Z_{i_2}}} \neq 0) \in \{0,5,6,7,8\}$$

All Δ variables are Boolean variables in $[0;1]$.

$$Minimize \sum_{i=0}^{r-1}\sum_{k=0}^{15}\Delta_X[i,k]$$

$\forall i \in [0,r[, \forall k \in [0;15[,$

$$\Delta_X[i,k] = \Delta_{SX}[i,k]$$

$$\Delta_Y[i,\pi(k)] = \Delta_{SX}[i,k]$$

$$\text{abstractXOR}(\{\ C,\ 255,\ X\ \}) \text{ with}$$

$$C = \left\{ \begin{matrix} \delta_Z[i,k] \oplus \delta_Y[i,(k+4)\%16] \oplus \delta_Y[i,(k+8)\%16] \oplus \delta_Y[i,(k+12)\%16] = 0 \\ \delta_Z[i,k] \oplus \delta_K[k] \oplus \delta_X[i+1,k] = 0 \end{matrix}, \forall i \in [0,r[, \forall k \in [0;15[\right\}$$

$$X = \left\{ \Delta_Z[i,k], \Delta_Y[i,(k+4)\%16], \Delta_Y[i,(k+8)\%16], \Delta_Y[i,(k+12)\%16], \Delta_K[k], \Delta_X[i+1,k] = 0, \forall i \in [0,r[, \forall k \in [0;15[\right\}$$

What is required to define a new constraint?

- Semantic and Syntax
- An algorithm to check the satisfiability of the constraint
- An algorithm to propagate

Let be:

- C A set of concrete XOR equations
- n An integer
- X A set of Boolean variables

$\text{AbstractXOR}_{C,k}(X)$ is satisfied **iff** there is a realization of X on the domain $[0; n]$ which satisfies C.

Example

$$C = \begin{cases} \delta_A = \delta_C \oplus \delta_D \\ \delta_B = \delta_C \oplus \delta_D \oplus \delta_E \end{cases}, \quad n = 4, \quad X = \{\Delta_A, \Delta_B, \Delta_C, \Delta_D, \Delta_E\}$$

$\Delta_A =$ true, $\Delta_B =$ true, $\Delta_C =$ false, $\Delta_D =$ true and $\Delta_E =$ true is a solution with : $\delta_A = 3$, $\delta_B = 3$, $\delta_C = 0$, $\delta_D = 2$ and $\delta_E = 1$ as concrete values.

Resolution by adapting the Gauss Jordan method

The system:

$$\begin{array}{ccccccccccc} \delta_A & & & & & \oplus & \delta_D & \oplus & \delta_E & = & 0 \\ & & \delta_B & \oplus & \delta_C & \oplus & \delta_D & \oplus & \delta_E & = & 0 \end{array}$$

is represented by:

Δ_A	Δ_B	Δ_C	Δ_D	Δ_E	$=$	0
1			1	1	=	0
	1	1	1	1	=	0

Solving process

1. Maintains the matrix in RRE (Row Reduced Echelon) form $\begin{bmatrix} \blacksquare & 0 & \star & \star & \star \\ 0 & \blacksquare & \star & \star & \star \end{bmatrix}$
2. Inference of the new values according to the selected consistency (Feas or Gac)

How to check if a complete assignment is satisfiable?

The abstract values:

$$\Delta_A = \text{ true}, \Delta_B = \text{ true}, \Delta_C = \text{ false}, \Delta_D = \text{ true and } \Delta_E = \text{ true}$$

The concrete system:

$$\begin{array}{ccccccccc} \delta_A & & & & \oplus & \delta_D & \oplus & \delta_E & = & 0 \\ & \delta_B & \oplus & \delta_C & \oplus & \delta_D & \oplus & \delta_E & = & 0 \end{array}$$

How to check if a complete assignment is satisfiable?

The abstract values:

$$\Delta_A = \text{ true}, \Delta_B = \text{ true}, \Delta_C = \text{ false}, \Delta_D = \text{ true and } \Delta_E = \text{ true}$$

The concrete system:

$$\begin{array}{ccccccccc} \delta_A & & & \oplus & \delta_D & \oplus & \delta_E & = & 0 \\ & & \delta_B & \oplus & \delta_D & \oplus & \delta_E & = & 0 \end{array}$$

How to check if a complete assignment is satisfiable?

The abstract values:

$$\Delta_A = \text{ true}, \Delta_B = \text{ true}, \Delta_C = \text{ false}, \Delta_D = \text{ true and } \Delta_E = \text{ true}$$

The concrete system:

$$\begin{array}{ccccccccc} \delta_A & & & \oplus & \delta_D & \oplus & 1 & = & 0 \\ & & \delta_B & \oplus & \delta_D & \oplus & 1 & = & 0 \end{array}$$

How to check if a complete assignment is satisfiable?

The abstract values:

$$\Delta_A = \text{ true}, \Delta_B = \text{ true}, \Delta_C = \text{ false}, \Delta_D = \text{ true and } \Delta_E = \text{ true}$$

The concrete system:

$$\begin{array}{ccccccc} \delta_A & & \oplus & 2 & \oplus & 1 & = & 0 \\ & \delta_B & \oplus & 2 & \oplus & 1 & = & 0 \end{array}$$

How to check if a complete assignment is satisfiable?

The abstract values:

$$\Delta_A = \text{ true}, \Delta_B = \text{ true}, \Delta_C = \text{ false}, \Delta_D = \text{ true and } \Delta_E = \text{ true}$$

The concrete system:

$$\begin{array}{ccccccccc} 3 & & \oplus & 2 & \oplus & 1 & = & 0 \\ & 3 & \oplus & 2 & \oplus & 1 & = & 0 \end{array}$$

How to check if a complete assignment is satisfiable?

The abstract values:

$$\Delta_A = \text{ true}, \Delta_B = \text{ true}, \Delta_C = \text{ false}, \Delta_D = \text{ true and } \Delta_E = \text{ true}$$

The concrete system:

$$\begin{array}{ccccccccc} 3 & & \oplus & 2 & \oplus & 1 & = & 0 \\ & 3 & \oplus & 2 & \oplus & 1 & = & 0 \end{array}$$

- Valid with $n = 4$ $(\forall \delta_x,\ \delta_x \in [1;4])$
- Invalid with $n = 2$ $(\forall \delta_x,\ \delta_x \in [1;2])$

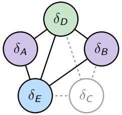
How to check if a complete assignment is satisfiable?

The abstract values:

$$\Delta_A = \text{true}, \Delta_B = \text{true}, \Delta_C = \text{false}, \Delta_D = \text{true and } \Delta_E = \text{true}$$

The concrete system:

$$\begin{array}{ccccccccc} 3 & & & \oplus & 2 & \oplus & 1 & = & 0 \\ & & 3 & \oplus & 2 & \oplus & 1 & = & 0 \end{array} \longrightarrow$$



- Valid with $n = 4$ $(\forall \delta_x,\ \delta_x \in [1;4])$
- Invalid with $n = 2$ $(\forall \delta_x,\ \delta_x \in [1;2])$

NP-Complete problem when the values are bounded, otherwise polynomial.

Let be:

- C A set of concrete XOR equations
- n An integer
- X A set of Boolean variables

$\text{AbstractXOR}_{C,k}(X)$ is satisfied **iff** there is a realization of X on the domain $[0; n]$ which satisfies C.

Let be:

- C A set of concrete XOR equations
- n An integer
- X A set of Boolean variables

AbstractXOR$_{C,k}(X)$ is satisfied **iff** there is a realization of X on the domain $[0; n]$ which satisfies C.

Let be:

- C A set of concrete XOR equations
- X A set of Boolean variables

$\mathrm{AbstractXOR}_{C,k}(X)$ is satisfied **iff** there is a realization of X which satisfies C.

Notations

$\Delta_A = 1$	$\Delta_B \in ?$	$\Delta_C = 0$	$\Delta_D = 1$	$\Delta_E = 1$		
1			1	1	=	0
	1	1	1	1	=	0

Notations

$$\Delta_A = 1 \qquad \Delta_B \in ? \qquad \Delta_C = 0 \qquad \Delta_D = 1 \qquad \Delta_E = 1$$

$$\underbrace{1}_{\text{pivot}} \qquad\qquad\qquad\qquad 1 \quad 1 \quad = 0$$

$$\underbrace{1}_{\text{pivot}} \quad \underbrace{1 \quad 1 \quad 1}_{\text{non pivot}} \quad = 0$$

Notations

$\Delta_A = 1$	$\Delta_B \in ?$	$\Delta_C = 0$	$\Delta_D = 1$	$\Delta_E = 1$		
1			1	1	=	0
	1	1	1	1	=	0

null variables are removed from the system
since $x \oplus 0 = x$

Propagation rule to check feasibility

Case 1

$$\forall j \in [0; n[, \quad eq_j = \{var_k\} \implies var_k = 0$$

$$\begin{bmatrix} \Delta_k & \in \{0,1\} & \\ & 1 & = 0 \end{bmatrix} \implies \Delta_k = 0$$

Proof: $\delta_k \oplus 0 = 0 \iff \delta_k = 0$

Propagation rule to ensure GAC 1/2

Case 2

$$\forall j \in [0; n[, \quad eq_j = \{var_k, var_l\} \wedge var_k = 1 \implies var_l = 1$$

$$\left[\begin{array}{cccc} \Delta_k = 1 & \Delta_l \in \{0,1\} & \\ 1 & 1 & = 0 \end{array}\right] \implies \Delta_l \neq 0$$

Proof: $\delta_k \oplus \delta_l = 0 \wedge \delta_k \neq 0 \implies \delta_l \neq 0$

Propagation rule to ensure GAC 2/2

Case 3

$$\forall j, j' \in [0; n[, \quad \left.\begin{array}{r} pivot(eq_j) = 1 \\ nonpivot(eq_j) = nonpivot(eq_{j'}) \end{array}\right\} \implies pivot(eq_{j'}) = 1$$

$$\begin{bmatrix} \Delta_k = 1 & \Delta_l \in \{0,1\} & \Delta_m \in \{0,1\} & \dots & \Delta_z \in \{0,1\} & \\ 1 & & 1 & \dots & 1 & = 0 \\ & 1 & 1 & \dots & 1 & = 0 \end{bmatrix} \implies \Delta_l \neq 0$$

Proof: $(\delta_k \bigoplus S = 0 \land \delta_l \bigoplus S = 0 \land \delta_j \neq 0) \implies \delta_l \neq 0$

Abstract XOR

$\Delta_A = 1$	$\Delta_B \in \{0, 1\}$	$\Delta_C = 0$	$\Delta_D = 1$	$\Delta_E = 1$		
1			1	1	=	0
	?	0	1	1	=	0

Abstract XOR

$\Delta_A = 1$	$\Delta_B \in \{0, 1\}$	$\Delta_C = 0$	$\Delta_D = 1$	$\Delta_E = 1$		
1			1	1	=	0
	?	0	1	1	=	0

May $\Delta_B = 0$ and 1 ?

Abstract XOR

$$\begin{array}{ccccccc} \Delta_A = 1 & \Delta_B \in \{0,1\} & \Delta_C = 0 & \Delta_D = 1 & \Delta_E = 1 & & \\ 1 & & & 1 & 1 & = & 0 \\ & ? & 0 & 1 & 1 & = & 0 \end{array}$$

May $\Delta_B = 0$ and 1 ?

We apply rule n°3. $\Delta_B = 1$

Abstract XOR

$\Delta_A = 1$	$\Delta_B \in \{0,1\}$	$\Delta_C = 0$	$\Delta_D = 1$	$\Delta_E = 1$		
1			1	1	=	0
	?	0	1	1	=	0

May $\Delta_B = \cancel{0}$ and **1** ?

We apply rule n°3. $\Delta_B = 1$

Abstract XOR

$$\begin{array}{ccccccc} \Delta_A = 1 & \Delta_B = \mathbf{1} & \Delta_C = 0 & \Delta_D = 1 & \Delta_E = 1 & & \\ 1 & & & 1 & 1 & = & 0 \\ & \mathbf{1} & 0 & 1 & 1 & = & 0 \end{array}$$

May $\Delta_B = \cancel{0}$ and $\mathbf{1}$?

We apply rule n°3. $\Delta_B = 1$

Abstract XOR says **no**

Number of Step-1 solutions on Midori

r	Basic Model	AbstractXOR Model	Advanced Model
3	64	28	38
4	30	16	16
5	26	16	16
6	122	16	16
7	74	16	16
8	32	16	16
9	282	16	16
10	218	16	16

TABLE 1 The number of different Step-1 solutions on Midori for Basic, AbstractXOR, Advanced [GL16]

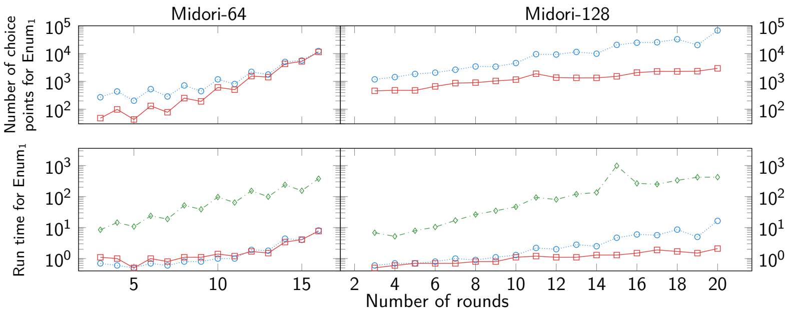


FIGURE 3 Comparison of AbstractXOR$_{Feas}$ (⋯⊙⋯) AbstractXOR$_{GAC}$ (—⊡—) Advanced[1] (-·◇·-) [GL16] on Midori

[1] Solved with SAT (Lingiling solver) for Step 1

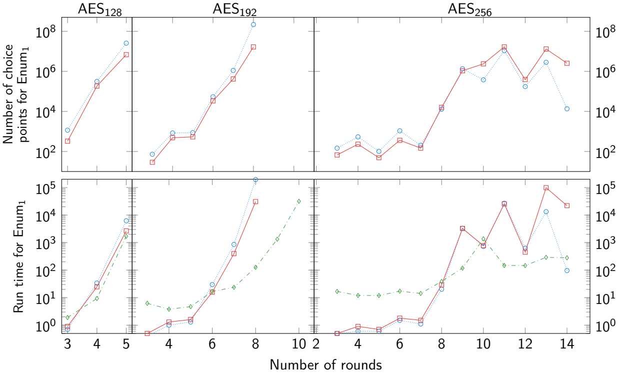


Figure 4 Comparison of AbstractXOR$_{Feas}$ (···○···), AbstractXOR$_{GAC}$ (—□—), Advanced[2] (-·◇·-) [Gér+20] on AES

[2]Solved with SAT (Lingiling solver) for Step 1

Cons

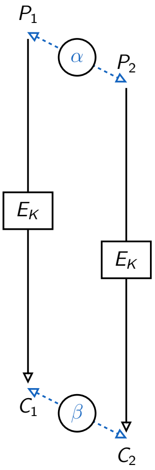
- Loss of performances when the cipher contains other functions

Pros

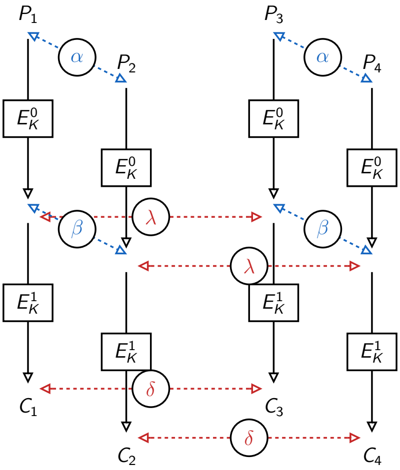
- Simplify truncated differential modelling
- Improve CP solver performances near SAT solver preformances

Automatic Search of Rectangle Attacks on WARP

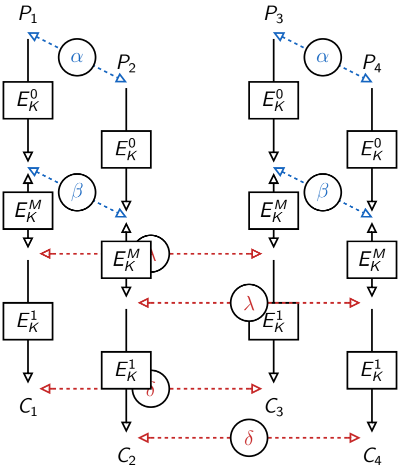
Computing Boomerang [Wag99] distinguisher probabilities



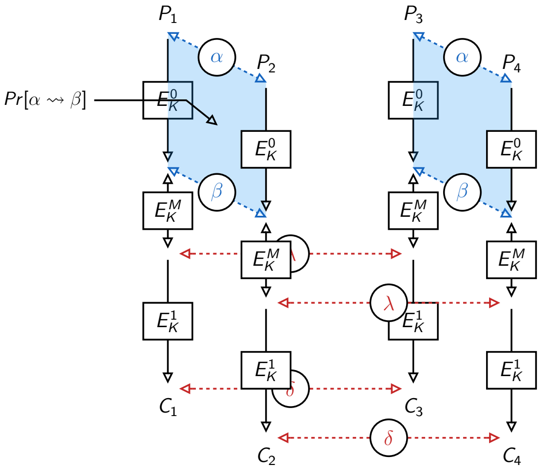
Computing Boomerang [Wag99] distinguisher probabilities



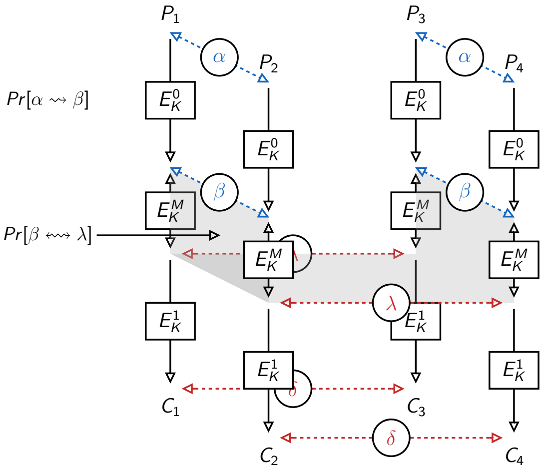
Computing Boomerang [Wag99] distinguisher probabilities using Sandwich [DKS10]



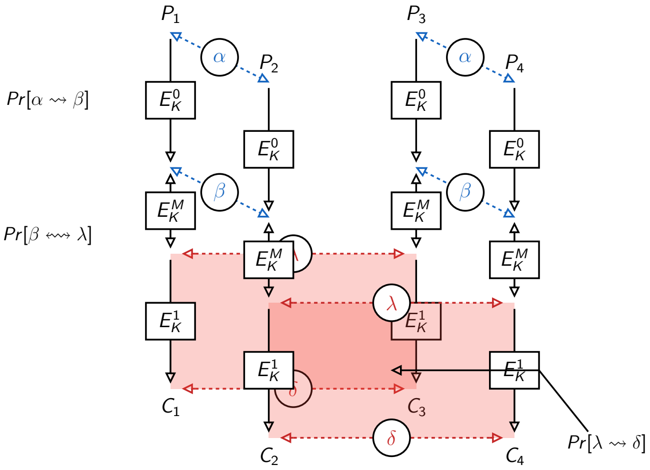
Computing Boomerang [Wag99] distinguisher probabilities using Sandwich [DKS10]



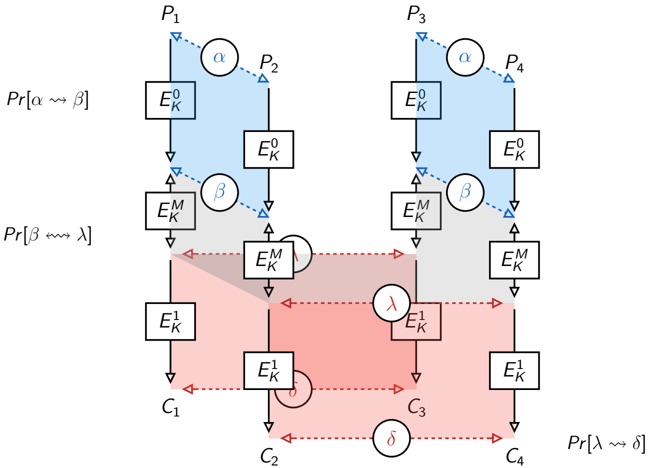
Computing Boomerang [Wag99] distinguisher probabilities using Sandwich [DKS10]



Computing Boomerang [Wag99] distinguisher probabilities using Sandwich [DKS10]

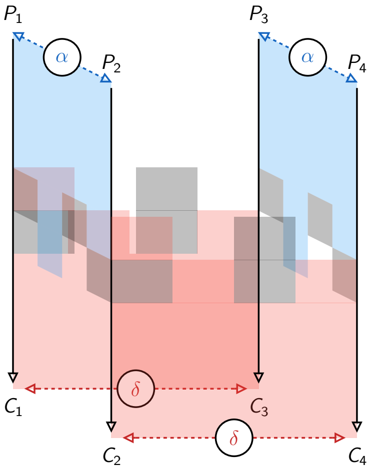


Computing Boomerang [Wag99] distinguisher probabilities using Sandwich [DKS10]



$$Pr[\alpha \leftrightsquigarrow \delta] = Pr[\alpha \rightsquigarrow \beta]^2 \times Pr[\beta \leftrightsquigarrow \lambda] \times Pr[\lambda \rightsquigarrow \delta]^2$$

Computing Boomerang [Wag99] distinguisher probabilities using the model of Delaune et al. [DDV20]



Model of Delaune et al. - Step 1

Each S-Box is abstracted by 3 Boolean variables:

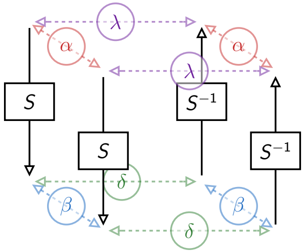
- Δ which indicates wether δ_{in} and δ_{out} are active or not,
- *free* which indicates wether the **input** difference is free of condition or not,
- $free_S$ which indicates wether the **output** difference is free of condition or not.

The other states are only represented by 2 Boolean variables:

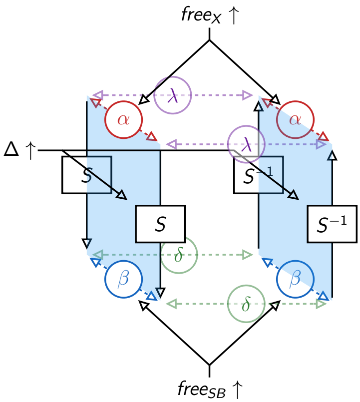
- Δ which indicates wether δ is active or not,
- *free* which indicates wether the state is free of condition or not,

The Step-1 defines the transitions to use.

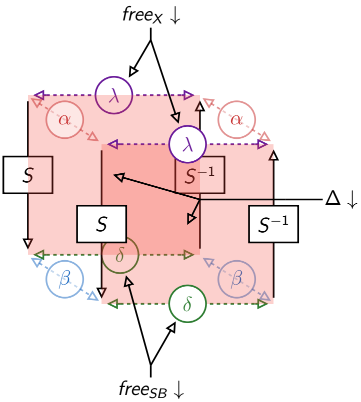
S-Box representation in the model of Delaune et al.



S-Box representation in the model of Delaune et al.

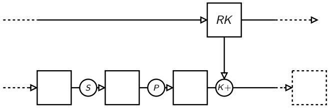


S-Box representation in the model of Delaune et al.

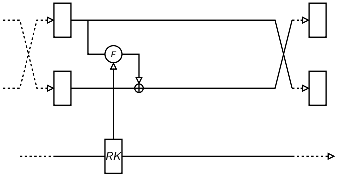


Adapt the model of Delaune et al. to Feistel Networks

SPN (such as Skinny [Bei+16])

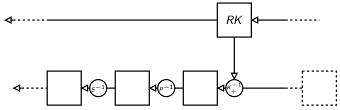


Feistel (such as WARP [Ban+20])

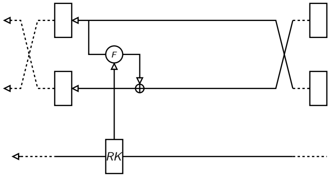


Adapt the model of Delaune et al. to Feistel Networks

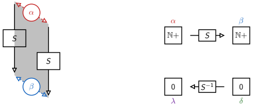
SPN (such as Skinny [Bei+16])



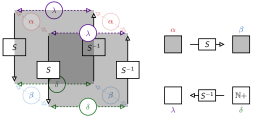
Feistel (such as WARP [Ban+20])



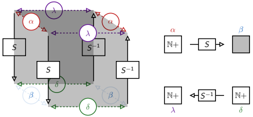
Boomerang transitions on SPN [Cid+18; WP19; DDV20]



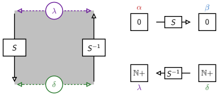
$DDT\uparrow$



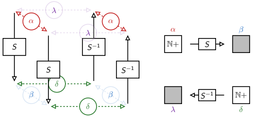
$DDT^2_\downarrow$



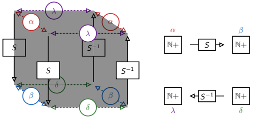
$LBCT$



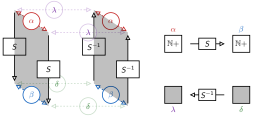
$DDT\downarrow$



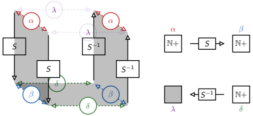
BCT



$EBCT$



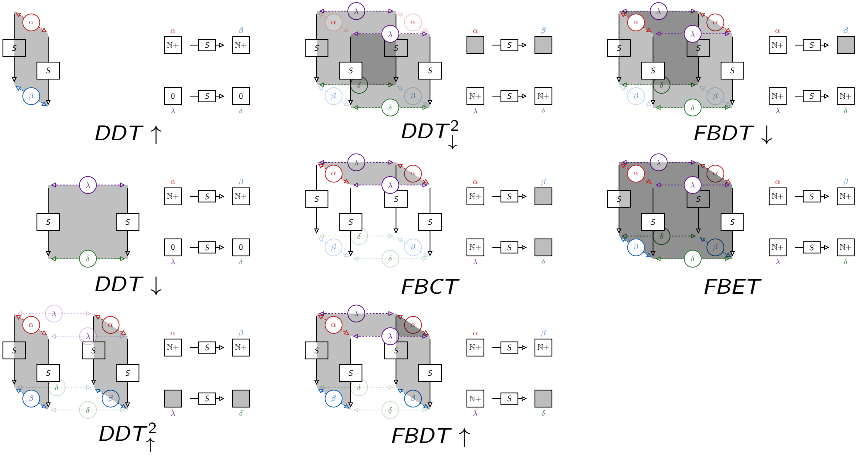
$DDT^2_\uparrow$



$UBCT$

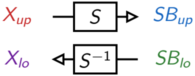
Automatic Search of Rectangle Attacks on WARP

Boomerang transitions on Feistel [BS91; Bou+20]



S-Box rules

Delaune et al.



- Rule 1

$$free_{Xup} \implies free_{SBup}$$
$$free_{SBlo} \implies free_{Xlo}$$

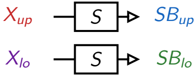
- Rule 2

$$free_{SBup} \implies \Delta_{Xup}$$
$$free_{Xlo} \implies \Delta_{Xlo}$$

- Rule 3

$$\neg free_{Xup} \lor \neg free_{Xlo}$$
$$\neg free_{SBup} \lor \neg free_{SBlo}$$

Feistel Adaptation



- Rule 1

$$free_{Xup} \implies free_{SBup}$$
$$free_{Xlo} \implies free_{SBlo}$$

- Rule 2

$$free_{SBup} \implies \Delta_{Xup}$$
$$free_{SBlo} \implies \Delta_{Xlo}$$

- Rule 3

$$\neg free_{Xup} \lor \neg free_{SBlo}$$
$$\neg free_{Xlo} \lor \neg free_{SBup}$$

WARP

- Designed to be a faster concurrent of AES
- Only one variant with 128-bit key and text

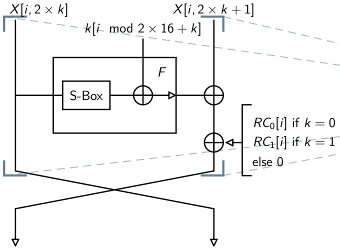


FIGURE 5 One round of WARP for two branches

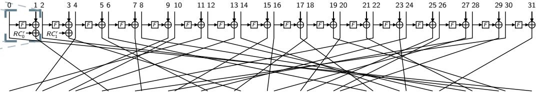


FIGURE 6 One round of WARP

What is similar to the Delaune et al.'s model?

- The boomerang representation
- The search steps

What is different compared to the Delaune et al.'s model?

- Specific optimizations dedicated to WARP
- The S-Box representation
  - S-Box rules
  - Transition tables
- **Integration of the attack complexity in the optimisation process**

Results on WARP

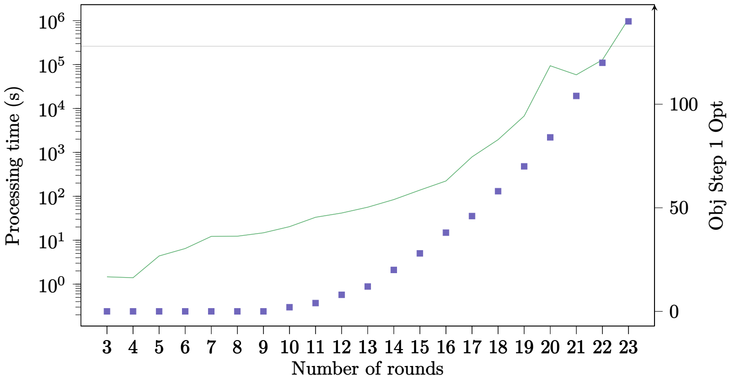


Figure 7 Execution time for Step-1 and Step-2 (—). Best probability found with Step-1 Opt (■). The black line corresponds to the probability 2^{-128}.

Results on WARP

Technique	Rounds	Probability	Time	Data	Mem.	Ref.
DC distinguisher	18	2^{-122}	-	-	-	[KY21]
DC distinguisher	20	$2^{-122.71}$	-	-	-	[TB21]
ID distinguisher	21	1	-	-	-	[Ban+20]
Boomerang distinguisher	21	$2^{-121.11}$	-	-	-	[TB21]
Boomerang distinguisher	23	2^{-124}	-	-	-	[LMR22]
Boomerang distinguisher	23	$2^{-115.59}$	-	-	-	[HNE22]
Differential attack	21	-	2^{113}	2^{113}	2^{72}	[KY21]
Differential attack	23	-	$2^{106.68}$	$2^{106.62}$	$2^{106.62}$	[TB21]
Rectangle attack	24	-	$2^{125.18}$	$2^{126.06}$	$2^{127.06}$	[TB21]
Rectangle attack	26	-	$2^{115.9}$	$2^{120.6}$	$2^{120.6}$	[LMR22]

Results on TWINE and LBlock-s

Cipher	Distinguishers	Rounds	Probability	Ref.
TWINE	Boomerang distinguisher	15	$2^{-58.92}$	[TB22]
TWINE	Boomerang Distinguisher + Clustering	15	$2^{-47.7}$	[LMR22]
TWINE	Boomerang Distinguisher	15	$2^{-51.03}$	[HNE22]
TWINE	Boomerang distinguisher	16	$2^{-61.62}$	[TB22]
TWINE	Boomerang Distinguisher + Clustering	16	$2^{-59.8}$	[LMR22]
TWINE	Boomerang Distinguisher	16	$2^{-58.04}$	[HNE22]
LBlock-s	Boomerang distinguisher	15	$2^{-58.64}$	[TB22]
LBlock-s	Boomerang Distinguisher + Clustering	16	$2^{-56.14}$	[Bou+20]
LBlock-s	Boomerang Distinguisher + Clustering	16	$2^{-54.8}$	[LMR22]
LBlock-s	Boomerang Distinguisher	16	$2^{-53.59}$	[HNE22]

Outlooks and Conclusion

Summary

- CP brings the ability to reuse and improve cryptanalysis models
- Find new attacks

Further search

- Integration in Tagada [Lib+21]

Contributions Overview

Differential cryptanalysis of Rijndael [Rou+22]

- Improving the overall process resolution time
- Compute all, except one, differential characteristics
- Find 2 new differential attacks

Global Constraint Abstract XOR [RS20]

- Better Step-1 abstraction
- Make the performance of a CP solver closer to a SAT solver's one

Boomerang cryptanalysis of Rijndael

- Extend the model of Delaune et al. to non-linear key schedules
- Find 1 new weak key boomerang attack

Automatic Search of Rectangle Attacks on WARP [LMR22]

- Adaptation of the model of Delaune *et. al.* to Feistel ciphers
- Results on WARP
  - 1 new state of the art distinguisher
  - 1 new state of the art rectangle attack
- Results Twine
  - 2 new state of the art distinguishers
- Results LBlock-s
  - 1 new state of the art distinguisher

[01] *Advanced Encryption Standard (AES)*. National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce. Nov. 2001.

[Ban+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. "Midori: A Block Cipher for Low Energy". In: *ASIACRYPT 2015, Part II*. Ed. by Tetsu Iwata and Jung Hee Cheon. Vol. 9453. LNCS. Springer, Heidelberg, Nov. 2015, pp. 411–436. DOI: 10.1007/978-3-662-48800-3_17.

[Ban+20] Subhadeep Banik, Zhenzhen Bao, Takanori Isobe, Hiroyasu Kubo, Fukang Liu, Kazuhiko Minematsu, Kosei Sakamoto, Nao Shibata, and Maki Shigeri. "WARP : Revisiting GFN for Lightweight 128-Bit Block Cipher". In: *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*. Ed. by Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn. Vol. 12804. Lecture Notes in Computer Science. Springer, 2020, pp. 535–564. DOI: 10.1007/978-3-030-81652-0_21. URL: https://doi.org/10.1007/978-3-030-81652-0%5C_21.

[Bei+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS". In: *CRYPTO 2016, Part II*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9815. LNCS. Springer, Heidelberg, Aug. 2016, pp. 123–153. DOI: 10.1007/978-3-662-53008-5_5.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. "Distinguisher and Related-Key Attack on the Full AES-256". In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Heidelberg, Aug. 2009, pp. 231–249. DOI: 10.1007/978-3-642-03356-8_14.

[Bou+20] Hamid Boukerrou, Paul Huynh, Virginie Lallemand, Bimal Mandal, and Marine Minier. "On the Feistel Counterpart of the Boomerang Connectivity Table (Long Paper)". In: *IACR Trans. Symm. Cryptol.* 2020.1 (2020), pp. 331–362. ISSN: 2519-173X. DOI: 10.13154/tosc.v2020.i1.331-362.

[BS91] Eli Biham and Adi Shamir. "Differential Cryptanalysis of DES-like Cryptosystems". In: *CRYPTO'90*. Ed. by Alfred J. Menezes and Scott A. Vanstone. Vol. 537. LNCS. Springer, Heidelberg, Aug. 1991, pp. 2–21. DOI: 10.1007/3-540-38424-3_1.

[Cid+18] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. "Boomerang Connectivity Table: A New Cryptanalysis Tool". In: *EUROCRYPT 2018, Part II*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10821. LNCS. Springer, Heidelberg, Apr. 2018, pp. 683–714. DOI: 10.1007/978-3-319-78375-8_22.

[DDV20] Stéphanie Delaune, Patrick Derbez, and Mathieu Vavrille. "Catching the Fastest Boomerangs Application to SKINNY". In: *IACR Trans. Symm. Cryptol.* 2020.4 (2020), pp. 104–129. ISSN: 2519-173X. DOI: 10.46586/tosc.v2020.i4.104-129.

[DKS10] Orr Dunkelman, Nathan Keller, and Adi Shamir. "A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony". In: *CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. LNCS. Springer, Heidelberg, Aug. 2010, pp. 393–410. DOI: 10.1007/978-3-642-14623-7_21.

[FJP13] Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin. "Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128". In: *CRYPTO 2013, Part I*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Heidelberg, Aug. 2013, pp. 183–203. DOI: 10.1007/978-3-642-40041-4_11.

[Gér+20] David Gérault, Pascal Lafourcade, Marine Minier, and Christine Solnon. "Computing AES related-key differential characteristics with constraint programming". In: *Artif. Intell.* 278 (2020). DOI: 10.1016/j.artint.2019.103183. URL: https://doi.org/10.1016/j.artint.2019.103183.

[GL16] David Gérault and Pascal Lafourcade. "Related-Key Cryptanalysis of Midori". In: *INDOCRYPT 2016*. Ed. by Orr Dunkelman and Somitra Kumar Sanadhya. Vol. 10095. LNCS. Springer, Heidelberg, Dec. 2016, pp. 287–304. DOI: 10.1007/978-3-319-49890-4_16.

[HNE22] Hosein Hadipour, Marcel Nageler, and Maria Eichlseder. "Throwing Boomerangs into Feistel Structures: Application to CLEFIA, WARP, LBlock, LBlock-s and TWINE". In: *IACR Transactions on Symmetric Cryptology* 2022.3 (Sept. 2022), pp. 271–302. DOI: 10.46586/tosc.v2022.i3.271-302. URL: https://tosc.iacr.org/index.php/ToSC/article/view/9858.

[Knu95] Lars R. Knudsen. "Truncated and Higher Order Differentials". In: *FSE'94*. Ed. by Bart Preneel. Vol. 1008. LNCS. Springer, Heidelberg, Dec. 1995, pp. 196–211. DOI: 10.1007/3-540-60590-8_16.

[KY21] Manoj Kumar and Tarun Yadav. "MILP Based Differential Attack on Round Reduced WARP". In: *Security, Privacy, and Applied Cryptography Engineering - 11th International Conference, SPACE 2021, Kolkata, India, December 10-13, 2021, Proceedings*. Ed. by Lejla Batina, Stjepan Picek, and Mainack Mondal. Vol. 13162. Lecture Notes in Computer Science. Springer, 2021, pp. 42–59. DOI: 10.1007/978-3-030-95085-9_3. URL: https://doi.org/10.1007/978-3-030-95085-9%5C_3.

[Lib+21] Luc Libralesso, François Delobel, Pascal Lafourcade, and Christine Solnon. "Automatic Generation of Declarative Models For Differential Cryptanalysis". In: *CP*. Vol. 210. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 40:1–40:18.

[LMR22] Virginie Lallemand, Marine Minier, and Loïc Rouquette. "Automatic Search of Rectangle Attacks on Feistel Ciphers: Application to WARP". In: *IACR Trans. Symmetric Cryptol.* 2022.2 (2022), pp. 113–140. DOI: 10.46586/tosc.v2022.i2.113-140. URL: https://doi.org/10.46586/tosc.v2022.i2.113-140.

[PP10] Christof Paar and Jan Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. en. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-44649-8 978-3-642-04101-3. DOI: 10.1007/978-3-642-04101-3. URL: http://link.springer.com/10.1007/978-3-642-04101-3 (visited on 03/04/2022).

[Rou+22] Loïc Rouquette, David Gerault, Marine Minier, and Christine Solnon. "And Rijndael? Automatic Related-key Differential Analysis of Rijndael". In: *AfricaCrypt 2022 - 13th International Conference on Cryptology AfricaCrypt*. Fes, Morocco, July 2022. URL: https://hal.archives-ouvertes.fr/hal-03671013.

[RS20] Loïc Rouquette and Christine Solnon. "abstractXOR: A global constraint dedicated to differential cryptanalysis". en. In: *Principles and Practice of Constraint Programming*. Ed. by Helmut Simonis. Vol. 12333. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 566–584. ISBN: 978-3-030-58474-0 978-3-030-58475-7. DOI: 10.1007/978-3-030-58475-7_33. URL: https://link.springer.com/10.1007/978-3-030-58475-7_33 (visited on 04/30/2021).

[TB21] Je Sen Teh and Alex Biryukov. “Differential Cryptanalysis of WARP”. In: *IACR Cryptol. ePrint Arch.* (2021), p. 1641. URL: https://eprint.iacr.org/2021/1641.

[TB22] Je Sen Teh and Alex Biryukov. “Differential cryptanalysis of WARP”. In: *Journal of Information Security and Applications* 70 (2022), p. 103316. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2022.103316. URL: https://www.sciencedirect.com/science/article/pii/S2214212622001648.

[Wag99] David Wagner. “The Boomerang Attack”. In: *FSE'99*. Ed. by Lars R. Knudsen. Vol. 1636. LNCS. Springer, Heidelberg, Mar. 1999, pp. 156–170. DOI: 10.1007/3-540-48519-8_12.

[WP19] Haoyang Wang and Thomas Peyrin. “Boomerang Switch in Multiple Rounds”. In: *IACR Trans. Symm. Cryptol.* 2019.1 (2019), pp. 142–169. ISSN: 2519-173X. DOI: 10.13154/tosc.v2019.i1.142-169.