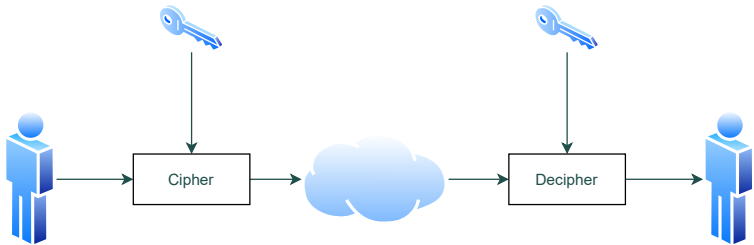
Automatic Generation of Models for Differential Cryptanalysis

Luc Libralesso, François Delobel, Pascal Lafourcade, Christine Solnon
<luc.libralesso@uca.fr>

CP 2021 October 2020

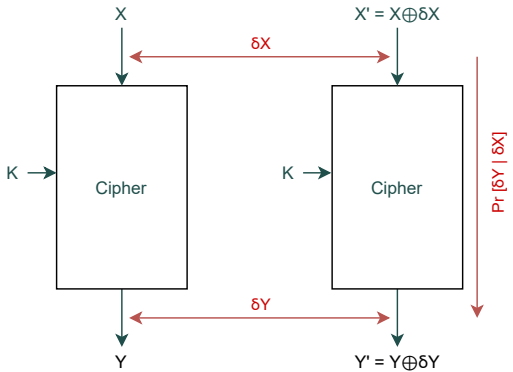
Université Clermont Auvergne

Symmetric cryptography (AES, DES, ...)



How to assess the security of the cipher?

Related-key differential cryptanalysis



Combinatorial optimization problem:

Single key: Find the differences in the text that maximize $Pr[\delta Y|\delta X]$

Related key: Differences may also be injected in the key

2-step solving process [Knu94]

Similar to *abstract interpretation*

Step 1

- Group bits in k-bit sequences
- Search for difference positions
- $\delta X = 1 \iff$ δX contains a difference
- Upper bounds on optimal probabilities

Step 2

Given a Step 1 solution:

- Integer variable $\delta X \in \{0 \dots 255\}$
- Maximizes the probabilities

What is challenging?

Step 2 is straightforward (thanks to table constraints), but **Step 1** is challenging:

- Many **skills** required
- **Takes time** to find accurate and efficient models
- May contain **bugs**
- Many **redundancies**

Can we automatize this process? (AI style)

- Describe the cipher into a unified language
- **Push a button**
- Obtain a MiniZinc model for solving the Step 1

Can we design such a button?

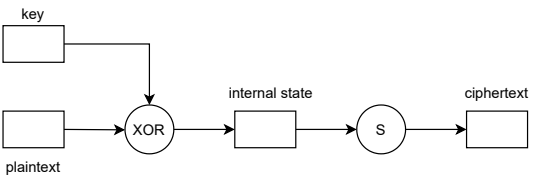
YES

Contribution 1 — A language to rule them all

A language to define ciphers (DAG):

Parameter: value taken by a variable

Operator: Parameter$^n \rightarrow$ Parameter^m (black-box function)



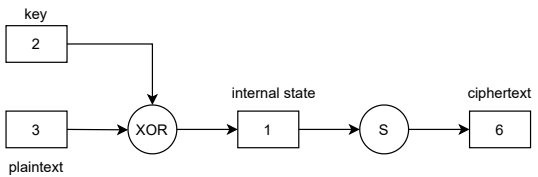
We test correctness of input/output pairs with a reference implementation.

Contribution 1 — A language to rule them all

A language to define ciphers (DAG):

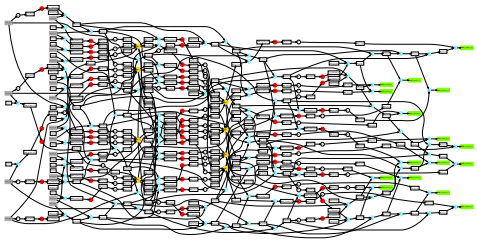
Parameter: value taken by a variable

Operator: Parameter$^n \rightarrow$ Parameter^m (black-box function)

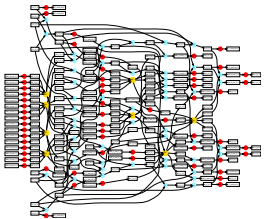


We test correctness of input/output pairs with a reference implementation.

Contribution 2 — Shaving



AES-128 3 rounds (before)



AES-128 3 rounds (after)

Iteratively apply rules:

1. Merge equal parameters
2. Suppress constant parameters
3. Suppress free parameters

Contribution 3 — constraint generation

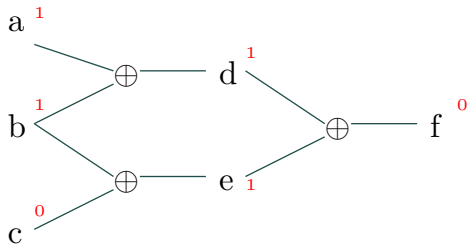
- Relation between input and output difference positions
- Automatic generation of a Boolean table from executable functions

XOR semantic				Constraint
[0,255] × [0,255] → [0,255]				$\{0,1\} \times \{0,1\} \times \{0,1\}$
a	b	a ⊕ b	→	abstraction (a,b,XOR(a,b))
(0	0	0)	→	(0,0,0)
…	…	…	→	…
(255	255	0)	→	(1,1,0)

(0,0,0), (0,1,1), (1,0,1), (1,1,0), (1,1,1), (1,0,0), (0,1,0), (0,0,1)

Same semantics as handcrafted constraint: $a + b + \mathrm{XOR}(a, b) \neq 1$

Contribution 4 — Additional constraints



XOR constraints:

✓ $a \oplus b \oplus d$
✓ $b \oplus c \oplus e$
✓ $d \oplus e \oplus f$
✗ $a \oplus c \oplus f$ (combination)

Huge impact ([RS20, GL16, GLMS20])
Time / abstraction trade-off

RESULTS

Benchmark instances

Considered ciphers:

- Midori
- AES
- Craft
- Skinny

Considered attacks:

- Single-key
- Related-key

Considered problems:

- Step1-opt
- Step1-enum

Total: 16 benchmarks, 254 instances[1]

[1] Evaluate your favorite solver: `https://gitlab.limos.fr/iia_lulibral/tagada`

Performance measures

Quality: Model tightness

- **Measure:** Number of “false alarms” due to the abstraction
- **Conclusion:** Same quality as state-of-the-art models!

Efficiency:

- **Measure:** CPU time of 3 solvers (Picat SAT, Chuffed, Gurobi)
- **Conclusion:** Competitive with state-of-the-art models!

Conclusions

- Automatic generation of state-of-the-art MiniZinc models
- Evaluation on 4 ciphers, 2 attacks, and 2 problems (16 new benchmarks and 254 instances)

Further work:

- More ciphers
- More attacks (new challenging problems to solve)
- Integration of the Step 2
- Study the interest of using dynamic programming

source code: `https://gitlab.limos.fr/iia_lulibral/tagada`

Automatic Generation of Models for Differential Cryptanalysis

Luc Libralesso, François Delobel, Pascal Lafourcade, Christine Solnon
<luc.libralesso@uca.fr>

CP 2021 October 2020

UNIVERSITÉ Clermont Auvergne

Stéphanie Delaune, Patrick Derbez, Paul Huynh, Marine Minier, Victor Mollimard, and Charles Prud'homme.
SKINNY with scalpel - comparing tools for differential analysis.
IACR Cryptol. ePrint Arch., 2020:1402, 2020.

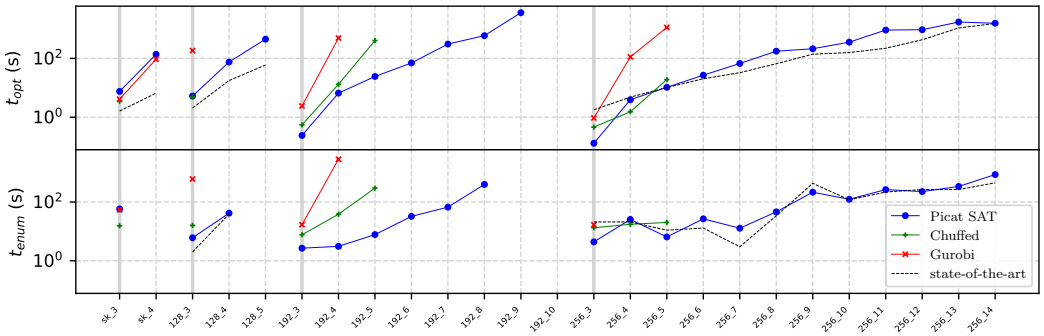
D. Gérault and P. Lafourcade.
Related-key cryptanalysis of midori.
In *INDOCRYPT*, volume 10095 of *LNCS*, pages 287–304, 2016.

D. Gerault, P. Lafourcade, M. Minier, and C. Solnon.
Computing AES related-key differential characteristics with constraint programming.
Artif. Intell., 278, 2020.

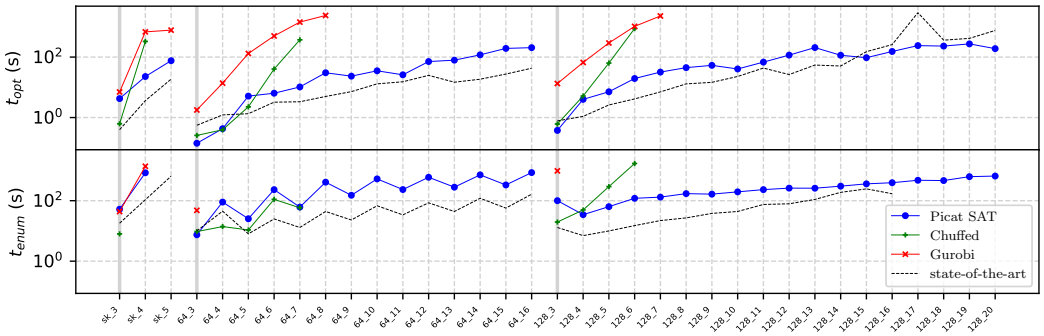
Lars R Knudsen.
Truncated and higher order differentials.
In *International Workshop on Fast Software Encryption*, pages 196–211. Springer, 1994.

L. Rouquette and C. Solnon.
abstractXOR: A global constraint dedicated to differential cryptanalysis.
In *26th International Conference on Principles and Practice of Constraint Programming*, volume 12333 of *LNCS*, pages 566–584, Louvain-la-Neuve, Belgium, September 2020. Springer.

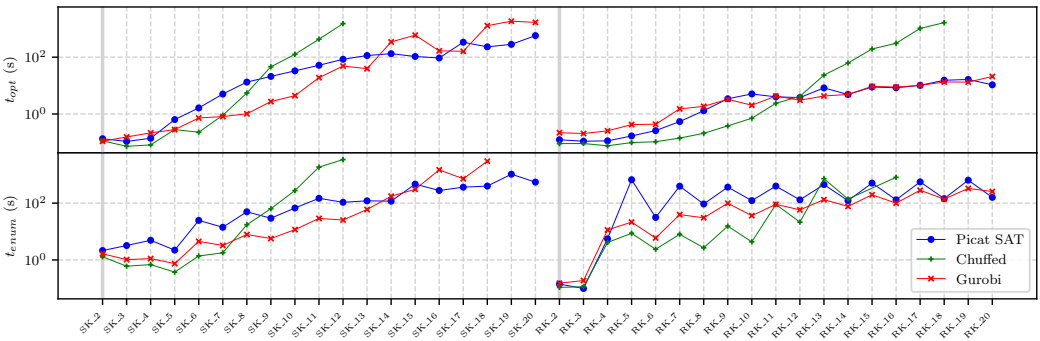
Solving time (AES [GLMS20])



Solving time (Midori [GL16])



Solving time (Craft)



Solving time (SKINNY [DDH+20])