Lightning Talk #1

Implémentation d'algorithmes dédiées dans TAGADA

Alexis Mialon
EPITA - Laboratoire de Recherche de l'EPITA (LRE)
12 mars 2026

Table des matières

01

Introduction

Qu'est ce que TAGADA ?

- Outil de cryptanalyse différentielle.
- Recherche à maximiser $\mathbb{P}(\delta_{out} \mid \delta_{in})$
- Technique actuellement utilisée dans l'outil : programmation par contrainte.
- But : utiliser des techniques de programmation dynamique pour résoudre ce problème.

02

Etat de l’art

Etude comparative menée dans l'article [3]

#Rounds	*Step1-enum* Obj_{Step1} (Nb sols)				MILP				MiniZinc/SAT				Ad-Hoc				Choco			
	SK	**TK1**	**TK2**	**TK3**	**SK**	**TK1**	**TK2**	**TK3**	**SK**	**TK1**	**TK2**	**TK3**	**SK**	**TK1**	**TK2**	**TK3**	**SK**	**TK1**	**TK2**	**TK3**
3	5 (4)	1 (12)	0 (1)	0 (1)	1s	1s	-	-	1s	1s	-	-	1s	21s	69s	69s	7s	1s	-	-
4	8 (3)	2 (9)	0 (1)	0 (1)	1s	1s	-	-	4s	1s	-	-	1s	22s	25s	76s	7s	1s	-	-
5	12 (2)	3 (2)	1 (12)	0 (1)	1s	1s	1s	-	4s	1s	1s	-	1s	21s	22s	103s	7s	1s	1s	-
6	16 (1)	6 (2)	2 (10)	0 (1)	1s	1s	1s	-	6s	1s	1s	-	1s	22s	22s	25s	7s	3s	1s	-
7	26 (4)	10 (2)	3 (2)	1 (12)	1s	1s	1s	1s	17s	8s	1s	1s	1s	21s	22s	22s	7s	7s	1s	1s
8	36 (17)	13 (1)	6 (2)	2 (11)	1s	1s	1s	1s	140s	7s	4s	2s	1s	22s	31s	23s	7s	8s	3s	1s
9	41 (2)	16 (1)	9 (1)	3 (3)	2s	2s	2s	2s	57s	11s	7s	1s	1s	22s	24s	26s	8s	9s	7s	1s
10	46 (2)	23 (1)	12 (2)	6 (3)	7s	5s	3s	2s	97s	46s	15s	10s	1s	22s	24s	27s	9s	60s	55s	2s
11	51 (2)	32 (2)	16 (1)	10 (3)	8s	11s	4s	3s	312s	29m	22s	24s	1s	23s	25s	32s	23s	188m	86s	34s
12	55 (2)	38 (7)	21 (1)	13 (2)	13s	35s	7s	3s	468s	> 24h	113s	35s	1s	24s	27s	25s	75s	> 24h	43m	288s
13	58 (6)	41 (2)	25 (2)	16 (2)	9s	53s	17s	6s	14m		14m	104s	1s	24s	30s	27s	249s		> 24h	56m
14	61 (2)	45 (3)	31 (1)	19 (1)	23s	93s	27s	8s	491s		72m	148s	1s	24s	39s	28s	10m			> 24h
15	66 (2)	49 (1)	35 (1)	24 (4)	69s	245s	75s	21s	27m		> 24h	157m	1s	25s	46s	34s	85m			
16	75 (8)	54 (1)	40 (2)	27 (1)	12m	423s	148s	39s	128m			251m	1s	25s	57s	38s	> 24h			
17	82 (4)	59 (5)	43 (1)	31 (2)	46m	22m	213s	53s	106m			> 24h	1s	27s	59s	48s				
18	88 (4)	62 (1)	47 (1)	35 (1)	178m	31m	535s	64s	403m				1s	27s	76s	73s				
19	92 (4)	66 (1)	52 (1)	43 (14)	529m	56m	29m	218s	436m				1s	28s	110s	283s				
20	96 (2)	70 (2)	57 (2)	45 (2)	16h	87m	33m	340s	174m				1s	28s	193s	326s				

Table 1 – Comparison of the times of the different Step 1 tools for solving *Step 1 – enum* (`SKINNY`), *i.e.* to enumerate all solutions for the optimal Obj_{Step1} bound given in the first column in each scenario : **SK**, **TK1**, **TK2** and **TK 3**. We report the `real` time on our server. [3]

Les algorithmes utilisés

Algorithm 1: Search for the best truncated representation (**SK**).

```
foreach state s do
    M[s] ⟵ list of states s' reachable from s through one round
end
foreach state s do
    C[0][s] ⟵ number of active cells of s
end
for 1 ≤ r < R do
    foreach state s do C[r][s] ⟵ ∞
    foreach state s do
        foreach state s' in M[s] do
            c ⟵ C[r − 1][s] + number of active cells of s'
            if c < C[r][s'] then C[r][s'] ⟵ c
        end
    end
end
return C
```

(a) Version *clé simple*.

Algorithm 2: Search for the best truncated representation (**TK**).

```
foreach state s, round key k do
    M[k][s] ⟵ list of states s' reachable from s and k through one round
end
foreach state s do
    C[0][s] ⟵ {(number of active cells of s, 0)}
end
for 1 ≤ r < R do
    foreach state s do C[r][s] ⟵ ∅
    foreach state s, round key k do
        foreach state s' in M[k][s] do
            foreach (cost, cancelled) ∈ C[r − 1][s] do
                if cancelled compatible with k then
                    c ⟵ cost + number of active cells of s'
                    C[r][s'] ⟵ C[r][s'] ∪ {(c, update(cancelled, k))}
                end
            end
        end
    end
    foreach state s do keepOptimals(C[r][s])
end
return C
```

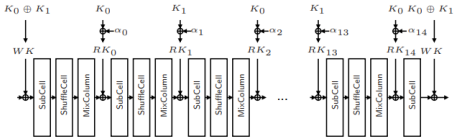
(b) Version *clés apparentées*.

Figure 1 – Algorithmes utilisés pour la résolution.

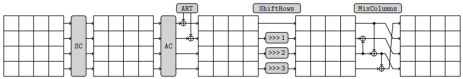
03

L’avancée du projet

Implémentation de chiffrements



(a) Chiffrement Midori ([1]).



(b) Chiffrement Skinny ([2]).

Figure 2 – Fonctions de chiffrement utilisées.

Implémentation de l'algorithme de minimisation de boites-S actives

N_r	Skinny	Midori
4	0.32	0.47
5	0.27	0.42
6	0.32	0.47
7	0.35	0.43
8	0.45	0.43
9	0.44	0.44
10	0.76	0.46
11	0.51	0.58
12	0.70	0.58
13	0.59	0.62
14	0.67	0.62
15	0.66	0.69
16	0.79	0.69

Table 2 – Temps de calcul (s) pour les chiffrements Skinny et Midori en fonction du nombre de tours N_r.

Des Questions ?

References i

[1] Subhadeep Banik et al. « Midori : A Block Cipher for Low Energy ». In : *Advances in Cryptology – ASIACRYPT 2015*. Sous la dir. de Tetsu Iwata et Jung Hee Cheon. Berlin, Heidelberg : Springer Berlin Heidelberg, 2015, p. 411-436. isbn : 978-3-662-48800-3.

[2] Christof Beierle et al. « The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS ». In : *Advances in Cryptology – CRYPTO 2016*. Sous la dir. de Matthew Robshaw et Jonathan Katz. Berlin, Heidelberg : Springer Berlin Heidelberg, 2016, p. 123-153. isbn : 978-3-662-53008-5.

[3] Stéphanie Delaune et al. *SKINNY with Scalpel - Comparing Tools for Differential Analysis*. Cryptology ePrint Archive, Paper 2020/1402. 2020. url : `https://eprint.iacr.org/2020/1402`.