Séminaire

Implémentation d'algorithmes dédiés dans TAGADA

Alexis Mialon
Encadré par Loïc Rouquette
EPITA - Laboratoire de Recherche de l'EPITA (LRE)
17 juillet 2026

Séminaire © 2026 Alexis Mialon
Encadré par Loïc Rouquette is licensed under CC BY 4.0.
To view a copy of this license, visit https ://creativecommons.org/licenses/by/4.0/.

Table des matières

licensed under CC BY 4.0

01

Introduction

Les motivations

- Sécurisation d'un chiffrement dès sa conception.
- Utilisation de solveurs génériques.
- Introduction de bibliothèques pour automatiser ces procédures.

licensed under CC BY 4.0

Qu'est ce que TAGADA (Delobel et al., 2023) ?

- 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.

licensed under CC BY 4.0

02

Etat de l’art

Etude comparative menée dans l'article Delaune et al., 2020

#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. Delaune et al., 2020

L'algorithme utilisé en single key

Algorithm 1 Calculate the minimal number of active S-boxes after R round (SK)

```
Require : R ≥ 1
  for all state s do
    N[s] ← the list of the next states reachable from s through one round
    C[0][s] ← number of actives S-boxes of s
  end for
  for 1 ≤ r < R do
    for all state s do
      C[r][s] ← inf
    end for
    for all state s do
      if C[r − 1][s] ≠ inf then
        for all state s' in N[s] do
          c ← C[0][s'] + C[r − 1][s]
          if c < C[r][s'] then
            C[r][s'] ← c
          end if
        end for
      end if
    end for
  end for
  return min(C[R − 1])
```


licensed under CC BY 4.0

L'algorithme utilisé en related key (1/3)

Algorithm 2 Calculate the minimal number of active S-boxes after R round (TK)

Require : $R \geq 1$
  for all state s, key k **do**
    $N[k][s] \leftarrow$ the list of the next states and key reachable from s and k through one round
  end for
  for all states **do**
    $C[s] \leftarrow$ get the cost of the state s
  end for
  $key \leftarrow construct\_not\_fixed\_key$
  return $guess\_the\_best\_key(0, R, N, key, \inf, nb\_cell)$

licensed under CC BY 4.0

L'algorithme utilisé en related key (2/3)

Algorithm 3 Guess the best key

```
Require : cell ≥ 0
Require : R ≥ 1
Require : key
Require : min
Require : nb_cell
  bound = compute_minimal_s_box(R, key, N)
  if bound ≥ min then
      return min
  else if cell < nb_cell then
      cell ← cell + 1
      key[cell − 1] ← fixed to inactive
      min1 ← guess_the_best_key(cell, R, N, key, min, nb_cell)
      key[cell − 1] ← fixed to active
      min2 ← guess_the_best_key(cell, R, N, key, min, nb_cell)
      return minimum(min1, min2)
  else
      return bound
  end if
```


licensed under CC BY 4.0

L'algorithme utilisé en related key (3/3)

Algorithm 4 Calculate the minimal number of active S-boxes after R round for a specific key

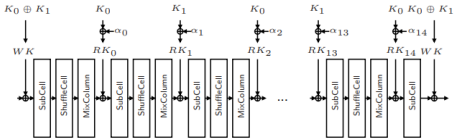
Require : $R \geq 1$
Require : key
Require : N
  for all state s **do**
    $C[0][s] \leftarrow$ {number of actives S-boxes of s, key}
  end for
  for $1 \leq r < R$ **do**
    for all state s **do**
      for all (cost c, key k) in $C[r-1][s]$ **do**
        for all (state s', key k') in $N[k][s]$ **do**
          $C[r][s] \leftarrow C[R][s] \cup \{c +$ number of actives S-boxes of $s', k'\}$
        end for
      end for
    end for
    $C[r-1] \leftarrow get\_best\_combinaison(C[r-1])$
  end for
  return $min\_cost(C[R-1])$

licensed under CC BY 4.0

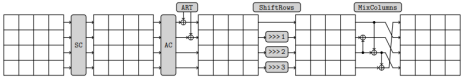
03

Ce qui a été réalisé au cours du semestre

Implémentation de chiffrements



(a) Chiffrement Midori (Banik et al., 2015).



(b) Chiffrement Skinny (Beierle et al., 2016).

Figure 1 – Fonctions de chiffrement utilisées.

licensed under CC BY 4.0

Implémentation de l'algorithme en single-key

N_r	Skinny	Midori
4	0.02	0.06
5	0.03	0.05
6	0.03	0.05
7	0.03	0.05
8	0.03	0.06
9	0.04	0.06
10	0.04	0.07
11	0.04	0.07
12	0.04	0.08
13	0.05	0.08
14	0.05	0.08
15	0.05	0.09
16	0.05	0.09

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

licensed under CC BY 4.0

Implémentation de l'algorithme avec key schedule

N_r	TK1	TK2	TK3
4	1.90	0.02	0.02
5	2.20	4.30	0.03
6	3.00	1.50	0.03
7	4.10	1.90	4.20
8	5.30	1.80	1.60
9	6.90	2.50	1.70
10	9.40	5.10	2.20
11	23.00	11.00	4.00
12	42.00	23.00	5.70
13	37.00	28.00	19.00
14	44.00	45.00	27.00
15	48.00	54.00	41.00
16	59.00	106.00	74.00

Table 3 – Temps de calcul (s) pour le chiffrement Skinny en fonction du nombre de tours N_r.

licensed under CC BY 4.0

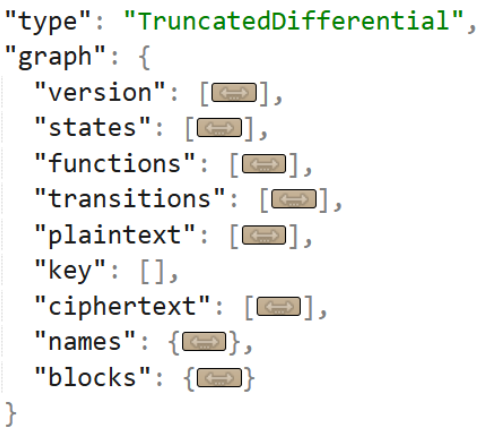
Généralisations des algorithmes à TAGADA

Algorithm 5 Compute table of transition

```
graph ← load_graph
round_transition ← extract_round(graph)
for all state s, key k do
    N[k][s] ← round_transition(k, s)
end for
for all states do
    C[s] ← get the cost of the state s
end for
for all keyk do
    K[k] ← get the reachable key from k
end for
return N
```


licensed under CC BY 4.0

Généralisations des algorithmes à TAGADA



(a) Exemple de graphe généré par TAGADA sous format json.

licensed under CC BY 4.0

Généralisation de l'algorithme en clé simple à TAGADA

N_r	Skinny	Midori	Twine
4	0.08	0.13	0.04
5	0.08	0.15	0.04
6	0.09	0.15	0.04
7	0.10	0.15	0.05
8	0.10	0.16	0.05
9	0.11	0.16	0.05
10	0.12	0.16	0.05
11	0.13	0.17	0.06
12	0.13	0.17	0.05
13	0.14	0.18	0.06
14	0.14	0.18	0.08
15	0.15	0.19	0.08
16	0.16	0.20	0.07

Table 4 – Temps de calcul (s) pour les chiffrements Skinny, Midori et Twine en fonction du nombre de tours N_r.

licensed under CC BY 4.0

Généralisation de l'algorithme avec key-schedule à TAGADA

N_r	TK1	TK2	TK3
4	16.00	17.00	17.00
5	16.00	18.00	18.00
6	17.00	19.00	18.00
7	18.00	19.00	21.00
8	21.00	22.00	25.00
9	31.00	26.00	28.00
10	36.00	30.00	31.00
11	54.00	59.00	41.00
12	59.00	91.00	51.00
13	76.00	110.00	123.00
14	78.00	154.00	126.00
15	100.00	261.00	332.00
16	119.00	294.00	434.00

Table 5 – Temps de calcul (s) pour le chiffrement Skinny en fonction du nombre de tours N_r.

licensed under CC BY 4.0

Comparaison de l'algorithme avec d'autres modèles

N_r	AR	Step1Opt_Picat	Step1Opt_Ortools	Step1Opt_Our_Work	Step1Opt_Tagada
3	7	1.94	0.64	5.69	10.38
4	16	3.28	4.15	6.63	23.28
5	23	11.78	9.53	5.92	34.64
6	30	> 2m	76.11	6.77	46.5
7	35	> 2m	> 2m	9.14	> 2m
8	38	> 2m	> 2m	6.64	> 2m
9	41	> 2m	> 2m	6.24	> 2m
10	50	> 2m	> 2m	7.02	> 2m
11	57	> 2m	> 2m	6.23	> 2m
12	62	> 2m	> 2m	6.36	> 2m
13	67	> 2m	> 2m	6.41	> 2m
14	72	> 2m	> 2m	6.25	> 2m
15	75	> 2m	> 2m	9.54	> 2m
16	84	> 2m	> 2m	9.29	> 2m

Table 6 – Temps de calculs (s) pour le chiffrement Midori (single-key) en fonction du nombre de tours N_r.

licensed under CC BY 4.0

04

Améliorations futures

Ce qui peut être amélioré

- Détection lors de la récupération d'états à transitions similaires.
- Gestion de la mémoire dans l'implémentation.

licensed under CC BY 4.0

Des Questions ?

References i

- Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., & Regazzoni, F. (2015)
  Midori : A Block Cipher for Low Energy. In T. Iwata & J. H. Cheon (Éd.), *Advances in Cryptology – ASIACRYPT 2015*. Springer Berlin Heidelberg.

- Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., & Sim, S. M. (2016)
  The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In M. Robshaw & J. Katz (Éd.), *Advances in Cryptology – CRYPTO 2016*. Springer Berlin Heidelberg.

- Delaune, S., Derbez, P., Huynh, P., Minier, M., Mollimard, V., & Prud'homme, C. (2020)
  SKINNY with Scalpel - Comparing Tools for Differential Analysis.

- Delobel, F., Derbez, P., Gontier, A., Rouquette, L., & Solnon, C. (2023)
  A CP-Based Automatic Tool for Instantiating Truncated Differential Characteristics.

licensed under CC BY 4.0