Fast MILP Models for Division Property

Patrick Derbez[1], Baptiste Lambin[2]

[1] Univ Rennes, CNRS, IRISA
[2] University of Luxembourg

Division Trails

Target: $f = f_n \circ f_{n-1} \circ \ldots \circ f_1 \circ f_0$

- Search for division trails through f
- Decompose f into smaller layers (e.g. small Sboxes, linear layer, AND, XOR, Copy, ...)
- Valid transitions known for each layer

Accuracy

- $f_0(x, y, z, t) = (xt \oplus y, xt \oplus z)$, $f_1(u, v) = (u \oplus v)$
- Division trail $(1, 0, 0, 1) \rightarrow (1)$ (i.e. $f_1 \circ f_0$ may depend on xt)
- XOR might lead to accuracy issue

Searching for Division Trails

Increase accuracy $\rightarrow$ handle larger layers

- **[Xia+16]:** Convex Hull (CH) to describe transitions through Sboxes (practical up to 6 bits)
- **[ZR19]:** exact modelization for any linear layer, practical for binary matrices on a field extension
- **[HWW20]:** quadratic constraints to modelize any linear layer $\rightarrow$ SMT solver
- **[DF20]:** propagation table of SuperSboxes (16 bits) $\rightarrow$ ad-hoc algorithm
- **[Udo21]:** modelize propagation table of SuperSboxes with thousands of logical constraints $\rightarrow$ SAT solver

All recent works abandoned MILP solver for either ad-hoc algorithm or SAT/SMT solvers!

MILP Models

Université de Rennes

- A **mixed-integer program** (MIP) is an optimization problem of the form:

$$\begin{array}{ll} \textit{Minimize} & c^T x \\ \textit{Subject to} & Ax = b \\ & l \leq x \leq u \\ \text{some or all } x_j & \text{integer} \end{array}$$

MILP Models

- A **mixed-integer program** (MIP) is an optimization problem of the form:

How to improve MILP models?

- Reduce the number of variables
- Reduce the number of inequalities
- Add dedicated branching strategy
- Solve easier problems
- ...

$$
\begin{aligned}
&\textit{Minimize} && c^T x \\
&\textit{Subject to} && Ax = b \\
& && l \leq x \leq u \\
&\text{some or all } x_j \text{ integer}
\end{aligned}
$$

An Important Property

Université de Rennes

2-subset bit-based division property

A transition $u \xrightarrow{f} v$ through a function f is valid if and only if x^u divides at least one monomial of $f(x)^v$.

A direct consequence for the search of division trails through a cipher is that for all $u' \prec u$ and $v' \succ v$, the transition $u' \xrightarrow{f} v'$ can be safely **added to or removed from** the model.

- Originally used to keep minimal transitions only
- But actually, **adding such "false/unnecessary" transitions does simplify the constraints**

Modelisation of AND and ADDMOD

Operation	*AND*	*ADDMOD*
Trail	$(a_1, a_2, \ldots, a_m) \to b$	$(a_1, \ldots, b_1, \ldots) \to (y_1, \ldots, c_1, \ldots)$
Constraints	$a_1 + \ldots + a_m \geq b$ $a_1 + \ldots + a_m \leq mb$	$-a_i - b_i - c_i + 2c_{i+1} + y_i \geq 0$ $a_i + b_i + c_i - 2c_{i+1} - 2y_i \geq -1$

Modelisation of AND and ADDMOD

Operation	*AND*	*ADDMOD*
Trail	$(a_1, a_2, \ldots, a_m) \to b$	$(a_1, \ldots, b_1, \ldots) \to (y_1, \ldots, c_1, \ldots)$
Constraints	$a_1 + \ldots + a_m \geq b$ $a_1 + \ldots + a_m \leq mb$	$-a_i - b_i - c_i + 2c_{i+1} + y_i \geq 0$ $a_i + b_i + c_i - 2c_{i+1} - 2y_i \geq -1$

- Inequalities in red can be safely removed from models

Classical Problem

Université de Rennes

- Let assume the following values for (x, y, z, t) are impossible

$$(0,0,1,1) \quad (0,1,1,1) \quad (1,0,1,1) \quad (0,0,0,1) \quad (0,0,1,0) \quad (0,0,0,0)$$

- Discarding those 6 values from a MILP model can be done with the 6 inequalities:

$$\begin{aligned}
& x + y + (1-z) + (1-t) \geq 1 \\
& x + (1-y) + (1-z) + (1-t) \geq 1 \\
& (1-x) + y + (1-z) + (1-t) \geq 1 \\
& x + y + z + (1-t) \geq 1 \\
& x + y + (1-z) + t \geq 1 \\
& x + y + z + t \geq 1
\end{aligned}$$

Use Quine-McCluskey algorithm to reduce the number of inequalities

Quine-McCluskey Algorithm

- Search for cosets of bit-aligned vector spaces of impossible values
- Example: assume the following values for (x, y, z, t) are impossible

$$(0,0,1,1) \quad (0,1,1,1) \quad (1,0,1,1) \quad (0,0,0,1) \quad (0,0,1,0) \quad (0,0,0,0)$$

- The QM algorithm aims at identifying pairs of impossible values that differ in only one bit

$$(0,0,\star,\star) \quad (\star,0,1,1) \quad (0,\star,1,1)$$

- Number of inequalities reduced to 3:

$$\begin{aligned} &x + y \geq 1 \\ &y + (1-z) + (1-t) \geq 1 \\ &x + (1-z) + (1-t) \geq 1 \end{aligned}$$

Going Further

- Number of inequalities reduced to 3:

$$
\begin{aligned}
&x + y \geq 1 \\
&y + (1 - z) + (1 - t) \geq 1 \\
&x + (1 - z) + (1 - t) \geq 1
\end{aligned}
$$

- Quine-McCluskey algorithm solves a NP-complete problem (complexity: $O(3^n/\sqrt{n})$)
- Adding non-minimal transitions removes the *saturation* step of QM algorithm $\rightarrow$ only need to find a minimal cover

Going Further

Université de Rennes

- Number of inequalities reduced to 3:

$$x + y \geq 1$$
$$y + (1 - z) + (1 - t) \geq 1$$
$$x + (1 - z) + (1 - t) \geq 1$$

- Quine-McCluskey algorithm solves a NP-complete problem (complexity: $O(3^n/\sqrt{n})$)
- Adding non-minimal transitions removes the *saturation* step of QM algorithm $\rightarrow$ only need to find a minimal cover
- Merge the two last inequalities:

$$x + y \geq 1$$
$$x + y + 2((1 - z) + (1 - t)) \geq 2$$

Modelisation Techniques

Université de Rennes

- Sbox:
  - COPY-AND-XOR (Lossy)
  - Convex Hull (Exact)
  - QM (Exact)
- Linear layer:
  - COPY-XOR (Lossy)
  - ZR (Exact)

Modelisation Techniques

- Sbox:
  - COPY-AND-XOR (Lossy)
  - Convex Hull (Exact)
  - QM (Exact)
  - Piecewise modelisation (Lossy)
- Linear layer:
  - COPY-XOR (Lossy)
  - ZR (Exact)

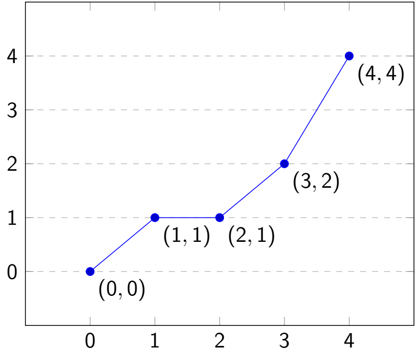

Figure: Piecewise linear function

Modelisation Techniques

- Sbox:
  - COPY-AND-XOR (Lossy)
  - Convex Hull (Exact)
  - QM (Exact)
  - Piecewise modelisation (Lossy)
- Linear layer:
  - COPY-XOR (Lossy)
  - ZR (Exact)

- For most of Sboxes used in practice, the following constraints are quite accurate to describe valid transitions $u \xrightarrow{S} v$:

$$\mathrm{hw}(v) = \begin{cases} 0 & \text{if } \mathrm{hw}(u) = 0 \\ n & \text{if } \mathrm{hw}(u) = n \text{ (for a } n\text{-bit Sbox)} \\ 1 & \text{otherwise} \end{cases}$$

Modelisation Techniques

- Sbox:
  - COPY-AND-XOR (Lossy)
  - Convex Hull (Exact)
  - QM (Exact)
  - Piecewise modelisation (Lossy)
- Linear layer:
  - COPY-XOR (Lossy)
  - ZR (Exact)
  - Weight equality (Lossy)

$u \xrightarrow{L} v$ valid iif the minor is invertible

$$\implies \mathrm{hw}(u) = \mathrm{hw}(v)$$

Modelisation Techniques

Université de Rennes

- Sbox:
  - COPY-AND-XOR (Lossy)
  - Convex Hull (Exact)
  - QM (Exact)
  - Piecewise modelisation (Lossy)

- Linear layer:
  - COPY-XOR (Lossy)
  - ZR (Exact)
  - Weight equality (Lossy)
  - Local ZR (Exact)

if the minor is not invertible

1. find a linear combination of rows equals to 0
2. compute the same linear combination of rows on the **full** matrix
3. look at columns with a non-zero coefficient
4. add a constraint to ensure that if those lines are selected, at least one of the columns is selected as well

Use **Callbacks** to remove false-positive trails

Running Times

Université de Rennes

Cipher	Rounds	Type of Result	Word Size	Our Time	Previous Time
AES	5	No Ext. Dist.	8-bit	13min	31min [EY21]†
ARIA	5	No Ext. Dist.	8-bit	5h	$\geq$ 24h [EY21]†
CRAFT	13	Conv. Dist.	-	3.6s	-
	14	No Ext. Dist.	16-bit	11min	-
HIGHT	20	Ext. Dist.	16-bit	12min	13 days [DF20]
	21	No Ext. Dist.	16-bit	14min	-
LED	8	No Ext. Dist.	16-bit	3h*	16h [Udo21]
Skinny	11	Ext. Dist.	16-bit	9min	22min [DF20]
	12	No Ext. Dist.	16-bit	80s	4min [DF20]
Camellia	7	Conv. Dist.	-	30s	99min [HWW20]
CLEFIA	10	Conv. Dist.	-	23min	82min [HWW20]
LEA	8	Conv. Dist.	-	20s	30min [SWW17]

Best Strategies

Université de Rennes

Cipher	Rounds	Modeling	LC-S-box	LC-Lin	LC-SSB
AES	4	PWL + WE	0	0	-
	5	QM + WE	-	60	-
ARIA	4	PWL + WE	0	0	-
	5	QM + CX	-	91	-
CRAFT	13	QM/CH + QM/CH	-	-	0
	14	QM/CH + CX	-	0	314
HIGHT	20	CX	-	6	0
	21	CX	-	21	0
LED	8	QM/CH + WE	-	107	54
Skinny	11	QM/CH + QM/CH	-	-	7
	12	QM/CH + QM/CH	-	-	93
Camellia	7	PWL + WE	0	0	-
	8	QM + CX	-	31	-
CLEFIA	10	PWL + WE	0	0	-
	11	QM + CX	-	9	-

About Weight Equality

Université de Rennes

What is the probability for a minor of an invertible matrix to be invertible?

- A random binary matrix is invertible with probability between 30 and 50%
- But matrices used in block ciphers are (most often) not random!
- Percentage of invertible minors for AES MixColumns matrix:

$1: 15.9\%$	$5: 1.4\%$	$9: 1.9\%$	$13: 3.5\%$	$17: 5.1\%$	$21: 12.3\%$	$25: 18.6\%$	$29: 26.3\%$
$2: 6.4\%$	$6: 0.9\%$	$10: 1.3\%$	$14: 3.6\%$	$18: 6.0\%$	$22: 13.0\%$	$26: 18.7\%$	$30: 31.5\%$
$3: 2.9\%$	$7: 0.6\%$	$11: 1.6\%$	$15: 3.4\%$	$19: 7.5\%$	$23: 15.3\%$	$27: 20.5\%$	$31: 44.5\%$
$4: 1.9\%$	$8: 1.8\%$	$12: 1.9\%$	$16: 3.8\%$	$20: 11.4\%$	$24: 17.5\%$	$28: 22.5\%$	$32: 100\%$

Conclusion

Université de Rennes

- New modelisation techniques for 2-subset division property
- Much better to not add all constraints into the model $\rightarrow$ use callbacks!
- Considering SuperSboxes not as useful as expected
- Optimizing models is important
- Code: `https://github.com/FastMILPDivisionProperty/FastMILPDivision`

Conclusion

Université de Rennes

- New modelisation techniques for 2-subset division property
- Much better to not add all constraints into the model → use callbacks!
- Considering SuperSboxes not as useful as expected
- Optimizing models is important
- Code: `https://github.com/FastMILPDivisionProperty/FastMILPDivision`

Thank you for your attention!