Towards an Efficient CNF Encoding of Block Ciphers (work in progress)

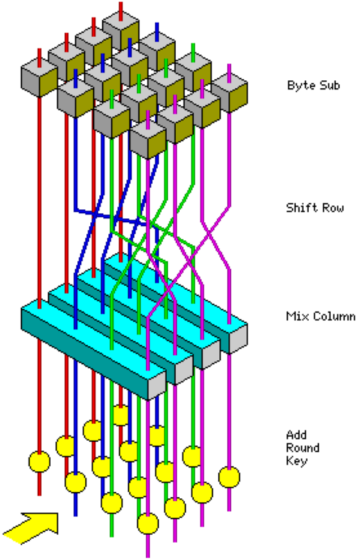
Konstanty Junosza-Szaniawski[1] Daniel Waszkiewicz[1,2]

Warsaw University of Technology, Warsaw, Poland

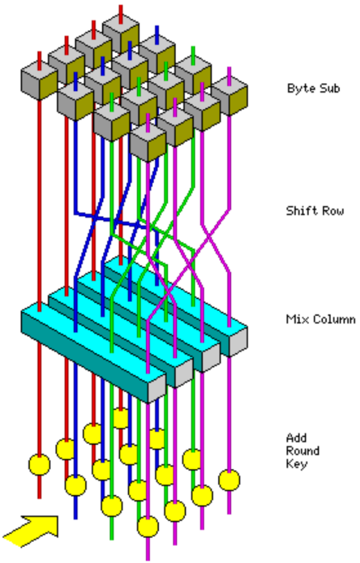
National Institute of Telecommunications, Warsaw, Poland

Pragmatics of SAT, 2022

SAT and Block Ciphers

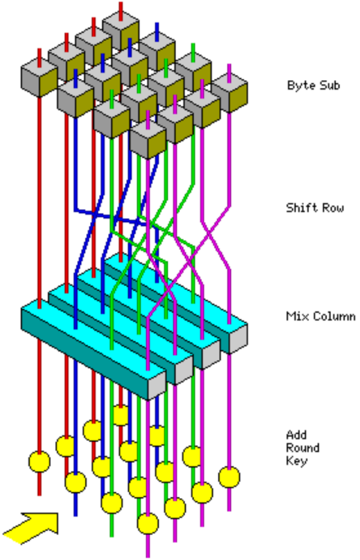


SAT and Block Ciphers



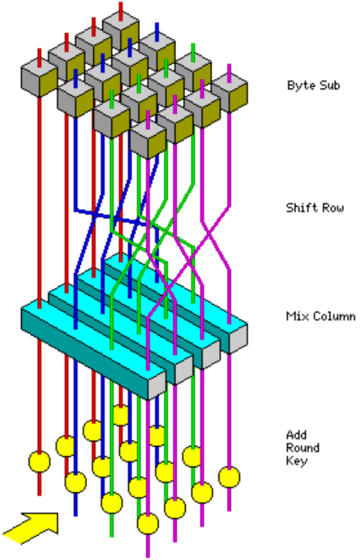
- Minimal differential path, number of *active* s-boxes.
- Counting keys with fix points, $\#k, \exists p \ Enc(p,k) = p$.
- Algebraic/logical cryptanalysis, Given p, c, find k that $Enc(p,k) = c$.

SAT and Block Ciphers



- Minimal differential path, number of *active* s-boxes.
  Only one formula for each problem and block cipher.
- Counting keys with fix points, $\#k, \exists p \; Enc(p, k) = p$.
  Only one formula for each problem and block cipher.
- Algebraic/logical cryptanalysis, Given p, c, find k that $Enc(p, k) = c$.
  One formula for each block cipher's private key.

Small Scale AES



- We consider 3 rounds of Small Scale AES with 4 bit s-box, 4 rows and 4 columns.
- 64 bits of private key.
- The nonlinear function, s-box, is given as a look-up table:

  $[6, B, 5, 4, 2, E, 7, A, 9, D, F, C, 3, 1, 0, 8]$

- The linear mapping is an 4×4 MDS matrix in $GF(2^4)$, which we represent as 16×16 matrix over GF(2).

Linear mapping

$$B = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

$$\begin{aligned} x_0 \oplus x_2 \oplus x_3 &= y_0, \\ x_0 \oplus x_1 \oplus x_2 &= y_1, \\ x_0 \oplus x_1 \oplus x_2 \oplus x_3 &= y_2, \\ x_1 \oplus x_2 \oplus x_3 &= y_3. \end{aligned}$$

With cutting number equal to 3:

$$\begin{aligned} x_0 \oplus x_2 = e_0,\ e_0 \oplus x_3 &= y_0, \\ x_0 \oplus x_1 = e_1,\ e_1 \oplus x_2 &= y_1, \\ e_1 \oplus x_2 = e_2,\ e_2 \oplus x_3 &= y_2, \\ x_1 \oplus x_2 = e_3,\ e_3 \oplus x_3 &= y_3. \end{aligned}$$

Observation: resulting system is linear straight-line program.

Straight-line program

$$x_0 \oplus x_2 = e_0,\; e_0 \oplus x_3 = y_0,$$
$$x_0 \oplus x_1 = e_1,\; e_1 \oplus x_2 = y_1,$$
$$e_1 \oplus x_2 = e_2,\; e_2 \oplus x_3 = y_2,$$
$$x_1 \oplus x_2 = e_3,\; e_3 \oplus x_3 = y_3.$$

Linear straight-line program (SLP) computing $A \cdot [x_1, \ldots, x_n]^T = [y_1, \ldots, y_m]^T$ are *lines*:

- each line of shape $v = \delta u \oplus \lambda w$ and $\delta, \lambda \in GF(2)$, u, v, w variables
- $x_1, \ldots, x_n$ input variables
- $y_1, \ldots, y_m$ output variables

Length of SLP is number of its lines.
Finding minimal SLP for given matrix A is NP-Hard.

Paar, 1997

- Record the frequency for all possible pairs of form $x_i \oplus x_j$ occurring in the matrix.
- The pair with the highest frequency is replaced by a new variable.
- Procedure continues until all remaining pairs occur at most once in the matrix. Then gates are computed naively.

$$B = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

$$\begin{aligned} x_0 \oplus x_2 = a_0,\ a_0 \oplus x_3 &= y_0, \\ a_0 \oplus x_1 &= y_1, \\ x_3 \oplus y_1 &= y_2, \\ x_1 \oplus x_3 = a_1,\ a_1 \oplus x_2 &= y_3. \end{aligned}$$

Boyar-Peralta, 2010

$$B = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

$$\begin{aligned} x_0 \oplus x_1 = a_0,\ a_0 \oplus x_2 &= y_1, \\ x_3 \oplus y_1 &= y_2, \\ x_1 \oplus y_2 &= y_0, \\ x_0 \oplus y_2 &= y_3. \end{aligned}$$

The BP algorithm:

- AES matrix from 152 naively to 97 XOR operations (the best 92, improved tie heuristics).
- Khazad matrix from 1232 naively to 507 XOR operations.
- Small Scale AES matrix from 72 naively to 47 XOR operations.

We consider SLP resulting from Paar's algorithm for $[A|I_n]$ input matrix.
The Naive approach is greedy algorithm with cutting number equal to 4.

Nonlinear function, ANF

- $S : \{0,1\}^4 \rightarrow \{0,1\}^4$
- *Forward* (FW):
  - one boolean function for each output in ANF,
  - $y_i = S_i(x_0, x_1, x_2, x_3)$.
- *Multivariate Quadratic* (MQ):
  - 21 polynomial equations of degree 2 in ANF,
  - minimal degree annihilators of characteristic function, $\chi(x_0, x_1, x_2, x_3, S(x_0), S(x_1), S(x_2), S(x_3))$
  - *theoretically good* for Groebner basis, XL, ElimLin algorithms.
  - Similar encodings: Sparse MQ, Groebner bases, Bosphorus (BS)*

CNF encoding	FW	MQ					BS
nr of variables	29	84					8
nr of clauses	150	596					35

Nonlinear function, symbolic execution

- $S : \{0,1\}^4 \to \{0,1\}^4$
- *Cryptol* (CRY):
  - Cryptol impl as look-up table + SAW (Software Analysis Workbench) to And-Inverter Graph,
  - *easy-to-use*, 5 lines Cryptol+SAW,
  - smaller than AIGs from Yosys and Quartus.
- *functionally reduced AIG* (FRAIG):
  - reduce AIGs from Cryptol with ABC.
- Similar encodings: Yosys, Quartus, CBMC

CNF encoding	FW	MQ	CRY	FRAIG			BS
nr of variables	29	84	42	40			8
nr of clauses	150	596	114	108			35

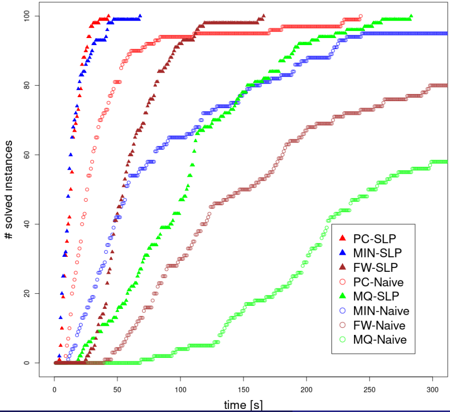
Nonlinear function, propagation complete

- $S : \{0,1\}^4 \to \{0,1\}^4$
- *Propagation complete* (PC):
  - using Optic for minimal propagation complete encoding,
  - FW equations as input,
  - *theoretically good* for CDCL algorithm.
- *Minimal* (MIN):
  - using Optic for arbitrary minimal,
  - FW equations as input,
- Similar encodings: genpce

CNF encoding	FW	MQ	CRY	FRAIG	MIN	PC	BS
nr of variables	29	84	42	40	8	8	8
nr of clauses	150	596	114	108	22	66	35

Results

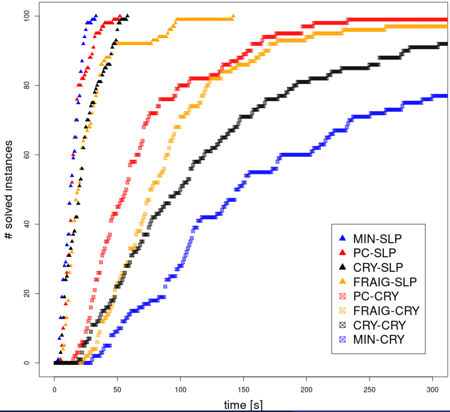
Tests for AES-3444 with 10 pairs plaintext-ciphertext. 100 CNFs for each encoding.
SAT-solver: plingeling with 20 threads. No timeout.



Encoding	# var.	# cl.	μ time[s]
MIN-SLP	8848	32712	14.683
MIN-Naive	7248	43912	85.040
PC-SLP	8848	54360	14.282
PC-Naive	7248	65560	39.065
FW-SLP	19180	95688	61.481
FW-Naive	17580	106888	164.701
MQ-SLP	46240	315120	109.459
MQ-Naive	44640	326320	283.046

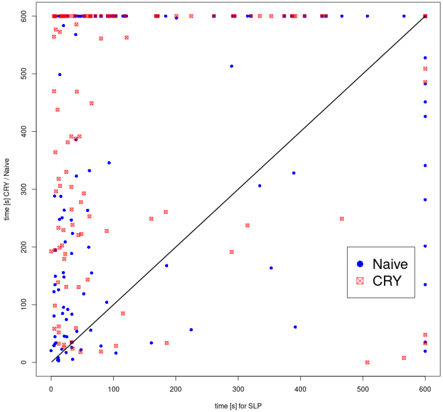
Results

Tests for AES-3444 with 10 pairs plaintext-ciphertext. 100 CNFs for each encoding.
SAT-solver: plingeling with 20 threads. No timeout.



Encoding	# var.	# cl.	μ time[s
MIN-SLP	8848	32712	13.62
MIN-CRY	24848	79672	229.47
PC-SLP	8848	54360	16.09
PC-CRY	24848	101320	73.43
FRAIG-SLP	24592	75024	26.34
FRAIG-CRY	40592	121984	97.83
CRY-SLP	25576	77976	22.23
CRY-CRY	41576	124936	134.01

Results for AES-3424, 30 random MDSs, timeout 600

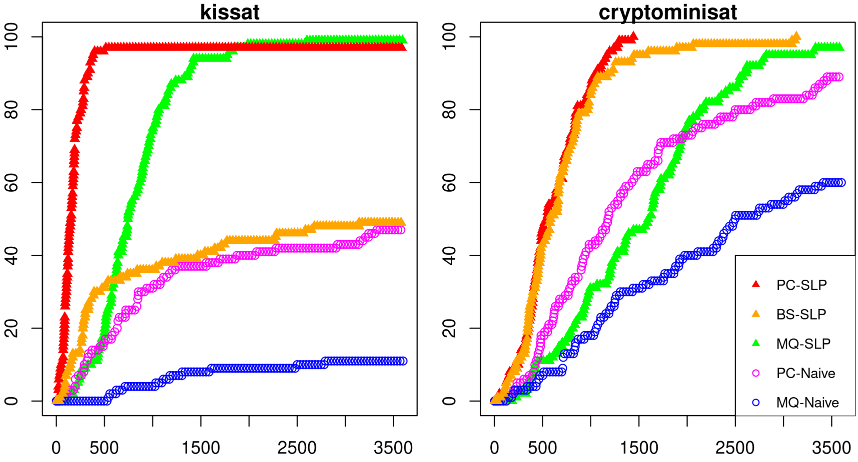


Encoding	nr of solved	total time [s]	PAR-2
SLP	108	36680.5	61880.5
Naive	76	57739.1	102139.1
CRY	67	66210.2	116010.2

Comments from Reviewers

- *XOR Local Search for Boolean Brent Equations*
    - cnf2xnf + xnfsat: no solution for AES-2444 in 3600 seconds.
    - maybe s-boxes are hard for SLS?
    - idea: cnf2xnf $\rightarrow$ SLP on XOR $\rightarrow$ xnf2cnf ?
- Single thread SAT-solverss (next slide)
- Bosphorus encoding (next slide)

Tests for single thread solvers



Conclusions

- S-box: PC with 2x-4x speed-up,
- Linear mapping: SLP with 3x-5x speed up,
- Combination PC-SLP 20x faster than default SageMath MQ-Naive.
- **Ongoing work**:
  - Minimal encoding of linear XOR system. For matrix A, the inverse A^{-1} is not straight-line program, but could be good encoding.
  - Given matrix A over GF(2), and $l \in N$. Find minimal d, that matrices B,C exists and $A|\overbrace{\mathbf{0}\ldots\mathbf{0}}^{d} = B \cdot C$ and $||B||_{max} \leq l$.
  - NP-Hard.