Analysis of Trivium using Compressed Right Hand Side Equations

Thorsten Schilling, Håvard Raddum

Selmer Center, University of Bergen

November 9, 2011

Motivation

- Algebraic cryptanalysis
  - Cipher constraints as equation system
  - Known information → constants
  - Secret information → variables
- Solve equation system → obtain secret information
- NP-hard problem
- “There are no shortcuts.” (M. Albrecht)

Motivation

- Algebraic cryptanalysis
  - Cipher constraints as equation system
  - Known information → constants
  - Secret information → variables
- Solve equation system → obtain secret information
- NP-hard problem
- "There are no shortcuts." (M. Albrecht)

Techniques

By consumed resource (time-out vs. space-out)

- Time
  - Guessing
  - SAT-solving (DPLL)
  - Guessing-/Agreeing (Raddum/Semaev)
  - Contraint programming etc.
- Space
  - Gröbner basis algorithms
  - Gluing-/Agreeing (Raddum/Semaev)
  - Multiple Right Hand Side Equations (Raddum)
  - *Compressed Right Hand Side Equations*

MRHS Equations

- Alternative, space saving representation
- Separate linearity from non-linearity

$$x_1 \cdot x_2 + x_3 + x_4 + x_5 \quad = \quad x_6$$

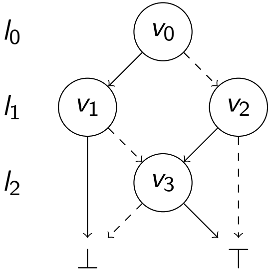
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Join equations: *Gluing*
- Problem: space-out by right hand side

Binary Decision Diagrams (BDDs)

- Graphstructure representing large sets of binary vectors
- Canonical
- Binary operations on sets in *compressed form*
- Mostly used in design/verification systems
- Use in cryptanalysis
  - LFSR results (Krause, 2002)
  - Grain results (Stegemann, 2007)

BDD by example

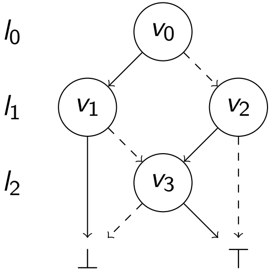


$$l_0 l_1 + l_0 l_2 + l_1 l_2 + l_0 + l_1 = 0$$

- Directed acyclic graph
- 0/1-edges
- $\top/\perp$ nodes
- Fixed variable order l_0, l_1, l_2
- Vectors as *accepting paths*

l_0	l_1	l_2
1	0	1
0	1	1
0	0	1
0	0	0

BDD by example

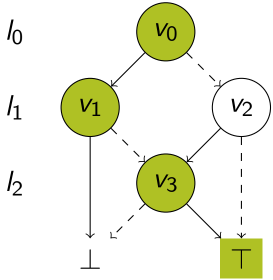


$l_0 l_1 + l_0 l_2 + l_1 l_2 + l_0 + l_1 = 0$

- Directed acyclic graph
- 0/1-edges
- $\top/\bot$ nodes
- Fixed variable order l_0, l_1, l_2
- Vectors as *accepting paths*

l_0	l_1	l_2
1	0	1
0	1	1
0	0	1
0	0	0

BDD by example

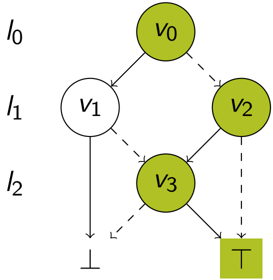


$$l_0 l_1 + l_0 l_2 + l_1 l_2 + l_0 + l_1 = 0$$

- Directed acyclic graph
- 0/1-edges
- $\top/\bot$ nodes
- Fixed variable order l_0, l_1, l_2
- Vectors as *accepting paths*

l_0	l_1	l_2
1	0	1
0	1	1
0	0	1
0	0	0

BDD by example

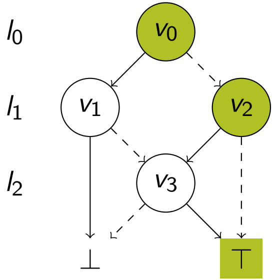


$l_0 l_1 + l_0 l_2 + l_1 l_2 + l_0 + l_1 = 0$

- Directed acyclic graph
- 0/1-edges
- $\top/\bot$ nodes
- Fixed variable order l_0, l_1, l_2
- Vectors as *accepting paths*

l_0	l_1	l_2
1	0	1
0	1	1
0	0	1
0	0	0

BDD by example

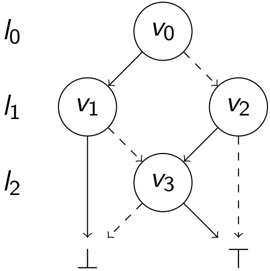


$l_0 l_1 + l_0 l_2 + l_1 l_2 + l_0 + l_1 = 0$

- Directed acyclic graph
- 0/1-edges
- $\top/\bot$ nodes
- Fixed variable order l_0, l_1, l_2
- Vectors as *accepting paths*

l_0	l_1	l_2
1	0	1
0	1	1
0	0	1
0	0	0

BDD by example



$$l_0 l_1 + l_0 l_2 + l_1 l_2 + l_0 + l_1 = 0$$

- Directed acyclic graph
- 0/1-edges
- $\top/\bot$ nodes
- Fixed variable order l_0, l_1, l_2
- Vectors as *accepting paths*

l_0	l_1	l_2
1	0	1
0	1	1
0	0	1
0	0	0

BDD properties

- Potentially space efficient representation
- Depending on variable ordering
- Finding optimal ordering NP-hard
- Very easy to count number of satisfying paths in BDD
- Boolean operations on BDDs
- Size of a BDD A: $\mathcal{B}(A)$, dominated by number of vertices
- Upper bound $\mathcal{B}(A \wedge B) \leq \mathcal{B}(A)\mathcal{B}(B)$

BDD properties

- Potentially space efficient representation
- Depending on variable ordering
- Finding optimal ordering NP-hard
- Very easy to count number of satisfying paths in BDD
- Boolean operations on BDDs
- Size of a BDD A: $\mathcal{B}(A)$, dominated by number of vertices
- Upper bound $\mathcal{B}(A \wedge B) \leq \mathcal{B}(A)\mathcal{B}(B)$

Compressed Right Hand Side Equation

Typical Trivium equation:

$$x_1 \cdot x_2 + x_3 + x_4 + x_5 = x_6$$

Compressed Right Hand Side Equation

Typical Trivium equation:

$$x_1 \cdot x_2 + x_3 + x_4 + x_5 = x_6$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Compressed Right Hand Side Equation

Typical Trivium equation:

$$x_1 \cdot x_2 + x_3 + x_4 + x_5 = x_6$$

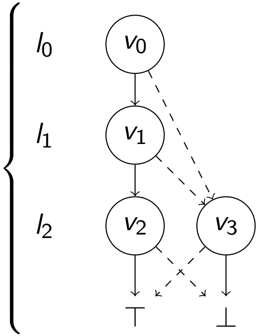
$$\begin{pmatrix} x_1 & = l_0 \\ x_2 & = l_1 \\ x_3 + x_4 + x_5 + x_6 & = l_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Compressed Right Hand Side Equation

Typical Trivium equation:

$$x_1 \cdot x_2 + x_3 + x_4 + x_5 = x_6$$

$$\begin{pmatrix} x_1 & = l_0 \\ x_2 & = l_1 \\ x_3 + x_4 + x_5 + x_6 & = l_2 \end{pmatrix} =$$



CRHS properties

- Keep separation between linearity and non-linearity
- Conjunction analogous to MRHS-gluing without final reduction step

$$\{[C_1]x = \mathcal{D}_1\} \circ \{[C_2]x = \mathcal{D}_2\} =$$
$$\left\{ \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x = \mathcal{D}_1 \wedge \mathcal{D}_2 \right\}$$

- Operations in compressed form
- Problem after some gluings: "BDD oblivious to linear dependencies in left hand side"

Trivium, Trivium-N

- One of final eStream Candidates
- Still incredibly simple in design, still not broken
- Reduced version Bivium not *convincing* $\rightarrow$ different structure in equation system (too easy?)
- Trivium-N
  - Trivium with N-bit state
  - Feedback very close to original Trivium
  - Equation system very similar to Trivium-288 (full)

Trivium, Trivium-N

- One of final eStream Candidates
- Still incredibly simple in design, still not broken
- Reduced version Bivium not *convincing* $\rightarrow$ different structure in equation system (too easy?)
- Trivium-N
    - Trivium with N-bit state
    - Feedback very close to original Trivium
    - Equation system very similar to Trivium-288 (full)

Results Trivium-N

N	n	k	$\mathcal{B}$	lc	Sol.	Mem.
35	85	173	$2^{18.86}$	88	$2^{85.67}$	87
40	94	191	$2^{20.57}$	97	$2^{93.77}$	182
45	106	215	$2^{21.68}$	109	$2^{106.60}$	358
50	115	233	$2^{21.15}$	118	$2^{115.60}$	258
55	127	257	$2^{21.55}$	130	$2^{127.60}$	329
60	138	282	$2^{22.34}$	144	$2^{140.35}$	560
65	148	299	$2^{22.66}$	151	$2^{148.60}$	687
70	160	323	$2^{22.42}$	163	$2^{160.49}$	588
75	171	349	$2^{22.78}$	178	$2^{173.83}$	742

Results Full Trivium

- Managed to glue all 666 equations into two CRHSs C_1, C_2:
  - $\mathcal{B}(C_1) = 2^{22.9}$
  - $\mathcal{B}(C_2) = 2^{24.8}$
- By upper bound $\Rightarrow \mathcal{B}(C_1 \wedge C_2) \leq 2^{47.7} << 2^{80}$

Open Questions

- How do we find efficiently path in BDD which satisfies all linear dependencies in left hand side.
- Problem not necessarily hard:
  - CRHS-Gluing still exponential problem but **possible to do**
  - There should be just one path which satisfies all constraints
  - Do we have to expect another exponential problem?

Summary

- New technique for solving equation systems in algebraic cryptanalysis
- Processing of equation system possible than with earlier approaches
- Technique able to handle huge number of solutions and count them
- Graphtheoretic problem: Find only path in DAG which satisfies some linear constraints