ANGRIFF AUF BIVIUM MITTELS SAT SOLVER

Tobias Eibach, Enrico Pilz

Fakultät für Informatik
Universität Ulm

9.11.2007
OUTLINE

1. ATTACK DESCRIPTION
2. CURRENT RESULTS
3. COMPARING TO OTHER ATTACKS
4. OUTLOOK
5. REFERENCES
Algorithm 1 Bivium Pseudocode

FOR i from 1 to N do
    \[ t_1 \leftarrow s_{66} + s_{93} \]
    \[ t_2 \leftarrow s_{162} + s_{177} \]
    \[ z_i \leftarrow t_1 + t_2 \]
    \[ t_1 \leftarrow t_1 + s_{91} \times s_{92} + s_{171} \]
    \[ t_2 \leftarrow t_2 + s_{175} \times s_{176} + s_{69} \]
    \[ (s_1, s_2, ..., s_{93}) \leftarrow (t_2, s_1, ..., s_{92}) \]
    \[ (s_{94}, s_{95}, ..., s_{177}) \leftarrow (t_1, s_{94}, ..., s_{176}) \]
\[ s_{66} + s_{93} + s_{162} + s_{177} + z_1 = 0 \]
\[ s_{65} + s_{92} + s_{161} + s_{176} + z_2 = 0 \]

\[ s_1 + s_{28} + s_{97} + s_{112} + z_{66} = 0 \]
\[ s_{27} + s_{69} + s_{96} + s_{111} + s_{162} + s_{175} * s_{176} + s_{177} + z_{67} = 0 \]
\[ s_{26} + s_{68} + s_{95} + s_{110} + s_{161} + s_{174} * s_{175} + s_{176} + z_{68} = 0 \]
\[ s_{25} + s_{67} + s_{94} + s_{109} + s_{160} + s_{173} * s_{174} + s_{175} + z_{69} = 0 \]

\[ s_4 + s_{14} * s_{15} + s_{29} * s_{30} + s_{31} + s_{55} + s_{80} * s_{81} + s_{82} + s_{94} + s_{95} * s_{96} + s_{97} + s_{122} * s_{123} + s_{124} + s_{160} + z_{147} = 0 \]

\[ \ldots \]
Some lines of a CNF file:

66 -93 -162 -177 0
-66 93 -162 -177 0
-66 -93 162 -177 0
-66 -93 -162 177 0
-178 66 93 171 91 92 0
-178 66 93 171 -91 92 0
-178 66 93 171 91 -92 0
178 -66 93 171 91 92 0
178 -66 93 171 -91 92 0
178 -66 93 171 91 -92 0

Bivium instances have about 445 variables and 9000 clauses.
Many Variations/Strategies

- How to split the 2 phases? (create CNF - solve CNF)
- When and how to split equations? More variables or higher degree?
- How many equations?
- Using Gaussian elimination?
- Also the following results imply certain strategies.
We studied several questions that come up when implementing the attack:

1. Which SAT solver to use?
2. Which variables to guess?
3. How many variables to guess?
4. What about the Hamming weight?
5. More ... but not in this talk.
# Comparing SAT solvers

<table>
<thead>
<tr>
<th></th>
<th>guess 40</th>
<th>guess 45</th>
<th>guess 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>satelite</td>
<td>46.10</td>
<td>3.32</td>
<td>0.26</td>
</tr>
<tr>
<td>minisat</td>
<td>67.32</td>
<td>5.06</td>
<td>0.36</td>
</tr>
<tr>
<td>picosat</td>
<td>103.96</td>
<td>5.78</td>
<td>0.42</td>
</tr>
<tr>
<td>rsat</td>
<td>229.09</td>
<td>11.49</td>
<td>0.79</td>
</tr>
<tr>
<td>zchaff</td>
<td>735.08</td>
<td>17.36</td>
<td>0.78</td>
</tr>
</tbody>
</table>

**Table:** Comparing SAT solvers

(time for one instance, 100 random instances averaged, guess: Ending)
**Where to guess**

<table>
<thead>
<tr>
<th>method</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beginning</td>
<td>204</td>
</tr>
<tr>
<td><strong>Ending</strong></td>
<td>9</td>
</tr>
<tr>
<td>Ending2</td>
<td>1070</td>
</tr>
<tr>
<td><strong>DreiVier</strong></td>
<td>60</td>
</tr>
<tr>
<td>Zufall1</td>
<td>791</td>
</tr>
<tr>
<td>Zufall2</td>
<td>263</td>
</tr>
<tr>
<td>Zufall3</td>
<td>2540</td>
</tr>
</tbody>
</table>

**Table:** Comparing different guessing strategies

(Time to solve 100 random instances, guessing 48 variables.)
**Time vs Guess Number**

\[(\text{guess: Ending, 48 - 32 variables, time / } 10^{10})\]
Figure: Influence of the Hamming weight

(guess: Ending - 36 variables, averaged over 100 instances)
COMPARING TO OTHER ATTACKS

Just to give a rough idea: (in seconds)

- Raddum: $\approx 2^{56} \rightarrow 7205759 \text{ E10}$
- Maximov: $\approx 2^{52.3} \rightarrow 554458 \text{ E10}$
- McDonald: guess 34 -> 440 -> total: 756 E10.
- Our current attack: guess 37 (Ending) -> 43.85 -> total: 603 E10.
- OBDDs ... ?
- Groebner basis / F4 / F5 ... ?
Besides optimising this attack and producing more experimental results, the following should also be interesting:

- "Explaining" the experimental results
- Extending the results to Trivium
- Extending the approach to other stream ciphers
- Comparing the approach to other generic attacks
Cannière and Preneel.  
*TRIVIUM - a stream cipher construction inspired by block cipher design principles*, 2005.

Bard and Courtois and Jefferson.  

Cameron McDonald and Chris Charnes and Josef Pieprzyk.  
*Attacking Bivium with MiniSat*, 2007.
The End

Thank you!

Questions?