SAT-based cryptanalysis: from parallel computing to volunteer computing

Oleg Zaikin

Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russia

RSD'2019. September 24, 2019.

SAT

Boolean satisfiability problem (*SAT*) - for an arbitrary Conjunctive Normal Form (CNF) to determine if there exists such assignment of Boolean variables from this CNF that makes it true.

An example of CNF with 3 clauses over 5 variables:

$$C = (x_1 \vee \overline{x_2}) \cdot (x_2 \vee x_3 \vee \overline{x_4}) \cdot (\overline{x_3} \vee x_4 \vee \overline{x_5})$$

This CNF is satisfiable, e.g., on (11001).

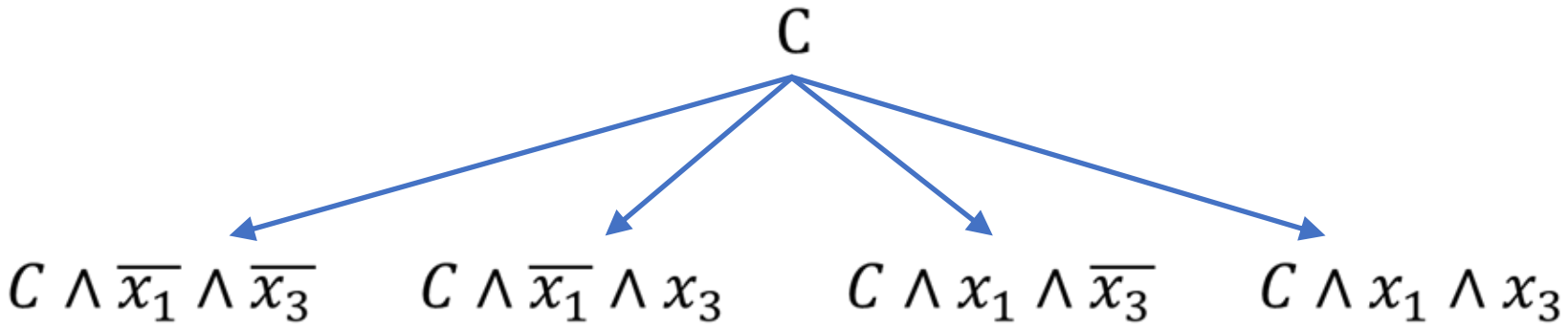
Applications: verification; bioinformatics; cryptanalysis.

Divide-and-Conquer SAT solving

Hard SAT instances are usually solved in parallel.

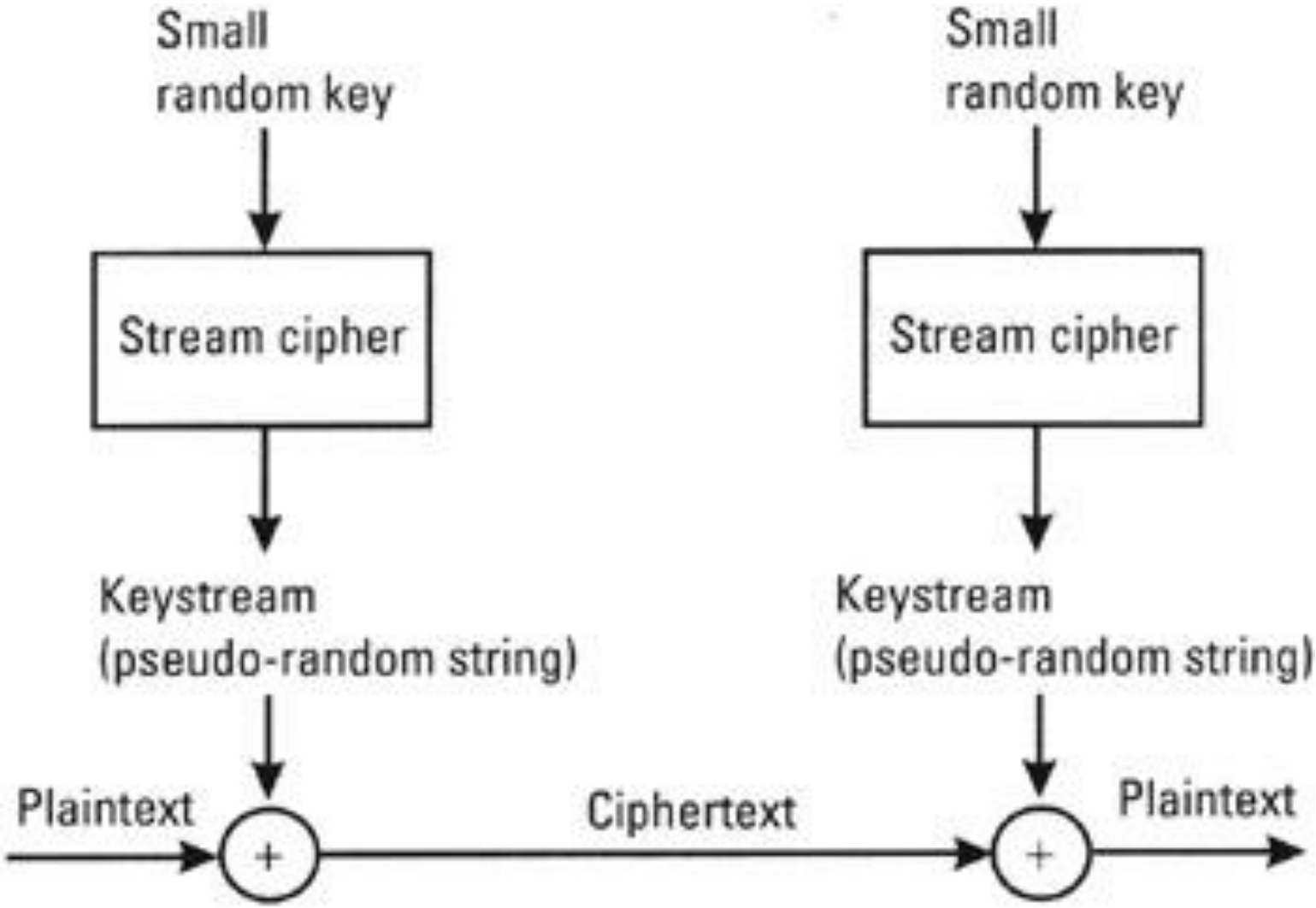
One of the possible approaches: Divide-and-Conquer.

CNF C is divided into a family of CNFs by varying all possible values of variables from S. Example: $S = \{x_1, x_3\}$.



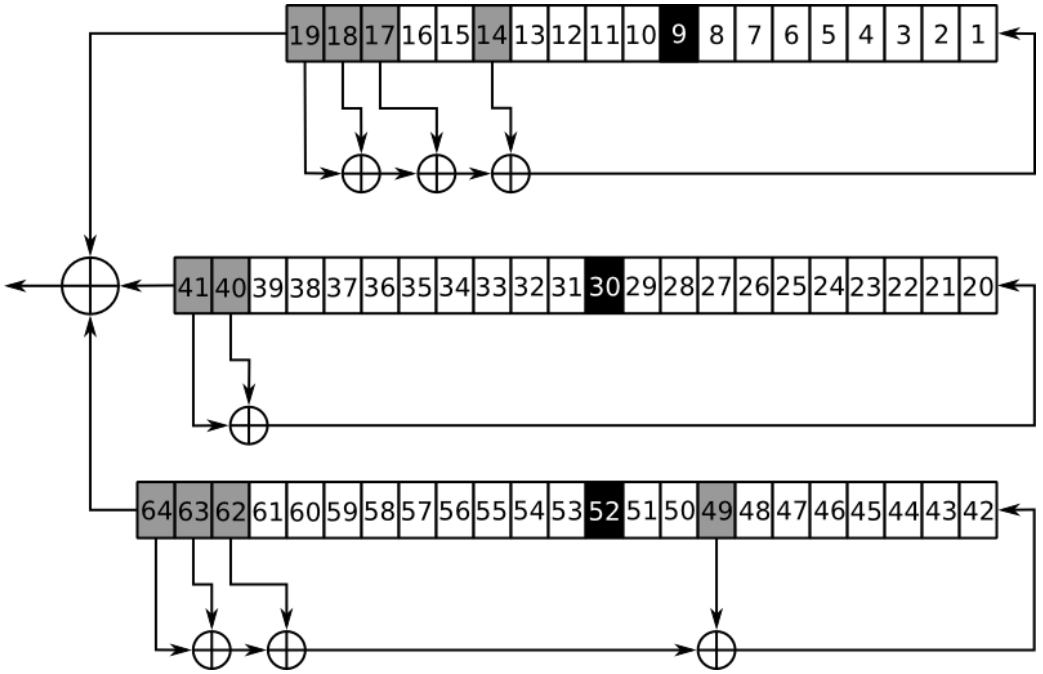
- A nontrivial problem – ***which variables to choose?***
- Usually considered as a black-box optimization problem.

Stream ciphers



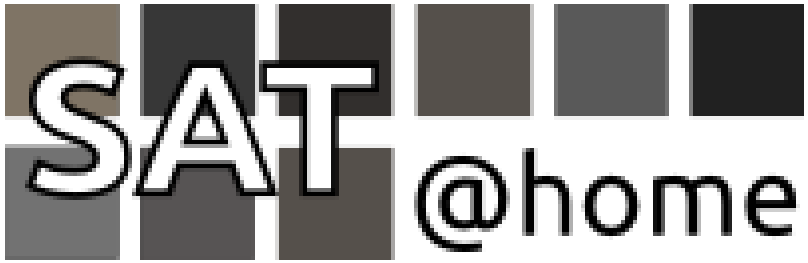
A5/1 generator

- A5/1 is used in GSM
- Secret key is 64 bits
- Cryptanalysis problem: given a keystream of size 114 bits to find a secret key.
- This cryptanalysis problem can be easily reduced to SAT.



Volunteer computing project SAT@home

- Started in September 2011.
- Based on BOINC.
- Aimed at solving hard large-scale problems that can be effectively reduced to SAT.
- Solved problems:cryptanalysis, combinatorics.
- Inactive since autumn 2016.
- **SAT@home 2.0 is planned**.



Preliminary stage of SAT-based cryptanalysis in SAT@home

In this report:

1. It is described how cryptanalysis problems are chosen for SAT@home.
2. It is shown how the hardness of the chosen problems is studied on a computing cluster.
3. It is discussed how decompositions of the chosen problems are constructed for further study in SAT@home.

Considered cryptanalysis problems

SAT-based cryptanalysis of four keystream generators is considered.

Generator	Secret key size	Analyzed keystream size
Grain_v1	160	200
Mickey	200	250
SHRINKING-SC64	64	96
SHRINKING-SC72	72	108
SELF-SHRINKING-SC64	64	96
SELF-SHRINKING-SC72	72	108

Considered cryptanalysis problems

While Grain_v1 and Mickey are very resistant to any known type of cryptanalysis, two weakened cryptanalysis problems are considered for both of them.

For Grain_v1, in the first (second) variant only the first 64 (72) bits of the secret key are unknown, while all remaining 96 (88) bits out of 160 are known (i.e. correct values of these variables are assigned in a CNF).

As for Mickey, only the first 64 (72) bits out of 200 are unknown. The corresponding crypt analysis problems are denoted as *grain-v1-sc-first64*, *grain-v1-sc-first72*, *mickey-sc-first64*, and *mickey-sc-first72* respectively.

Considered cryptanalysis problems

Thus, 8 SAT-based cryptanalysis problems are considered:

1. *shrinking-64*
2. *shrinking-72*
3. *self-shrinking-64*
4. *self-shrinking-72*
5. *grain-v1-sc-first64*
6. *grain-v1-sc-first72*
7. *mickey-sc-first64*
8. *mickey-sc-first72*

For each of these problems 3 instances were constructed by randomly generating secret keys. Thus, 24 SAT instances were constructed in total.

Analysis via multithreaded SAT solvers

Top-3 multithreaded SAT solvers were chosen to study the considered 24 SAT instances: plingeling2018, painless-mcomsps, abcdsat-p18.

The experiments were conducted on the “Academician V.M. Matrosov” computing cluster. Each computing node of this cluster is equipped with 2 18-core Intel Xeon E5-2695 CPUs and 128 Gb RAM.

Each solver was launched on one node, so 36 CPU cores were used. The time limit was 1 day.

It turned out, that all 64 bit instances are quite simple, while all 72 bit instances are hard for these solvers.

Considered cryptanalysis problems

1. *shrinking-64*
2. *shrinking-72*
3. *self-shrinking-64*
4. *self-shrinking-72*
5. *grain-v1-sc-first64*
6. *grain-v1-sc-first72*
7. *mickey-sc-first64*
8. *mickey-sc-first72*

Divide-and-conquer solving

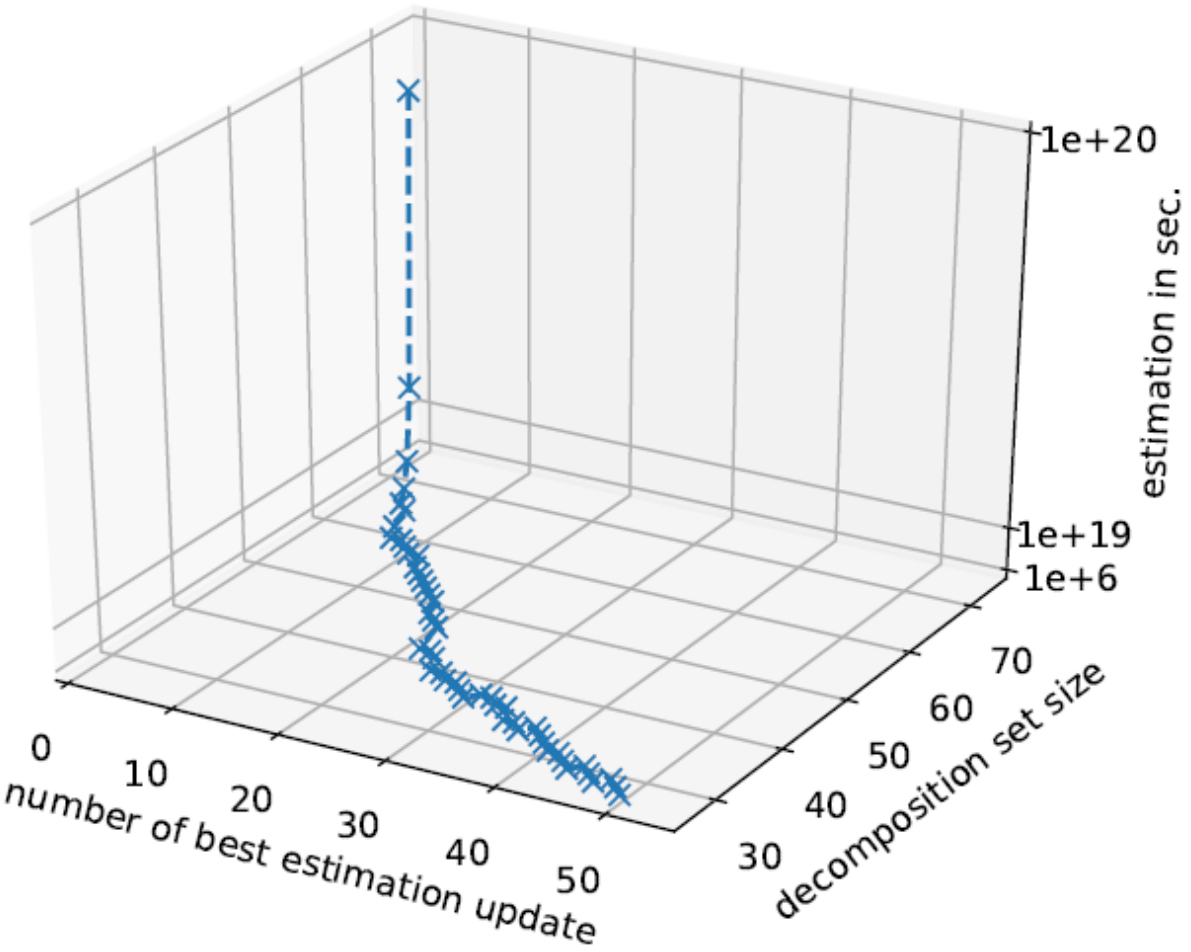
The ALIAS tool was used to find a decomposition set (set of variables for divide-and-conquer) for each 72-bit cryptanalysis problem.

The (1+1) evolutionary algorithm was used for this purpose: the search space of size $\{0,1\}^{72}$ was used in all cases.

ALIAS was launched for 1 day on the rst instance of each of 4 problems on 1 cluster's node. In all cases as a start decomposition set all variables which encode a secret key were used.

Divide-and-conquer solving

Optimization process for mickey-sc-first72.



Divide-and-conquer solving

Details on found decomposition sets for the considered generators with 72-bit secret keys. Estimations are presented for 1 CPU core.

Generator	Decomposition set size (out of 72)	Estimation (seconds)
MICKEY-SC-FIRST72	26	6.21e+06
GRAIN-V1-SC-FIRST72	28	5.02e+07
SHRINKING-SC72	30	6.66e+08
SELF-SHRINKING-SC72	67	1.05e+20

Self-shrinking-sc72 does not suit for the employed type of the Divide-and-Conquer approach: the found estimation is too high, and the size of the found decomposition set (67) is too close to the secret key size (72).

Therefore, self-shrinking-sc72 is not considered further.

Considered cryptanalysis problems

1. *shrinking-64*
2. *shrinking-72*
3. *self-shrinking-64*
4. *self-shrinking-72*
5. *grain-v1-sc-first64*
6. *grain-v1-sc-first72*
7. *mickey-sc-first64*
8. *mickey-sc-first72*

Divide-and-conquer solving

It is assumed, that SAT@home's performance will quickly return on its high level after restart, and as a result a performance comparable to 500 CPU cores will be achieved.

In the table, estimations on solving one instance of each considered problem in SAT@home are presented.

Generator	Decomposition set size	Estimation, SAT@home
MICKEY-SC-FIRST72	26	3 h 27 min
GRAIN-V1-SC-FIRST72	28	1 d 4 h
SHRINKING-SC72	30	15 d 10 h

Divide-and-conquer solving

It turned out, that mickey-sc-first72 is way too simple for in SAT@home, so it is better so solve the corresponding instances on a computing cluster.

All 3 SAT instances of mickey-sc-first72 were solved on the computing cluster via the parallel SAT solver PDSAT.

Considered cryptanalysis problems

1. *shrinking-64*
2. *shrinking-72*
3. *self-shrinking-64*
4. *self-shrinking-72*
5. *grain-v1-sc-first64*
6. *grain-v1-sc-first72*
7. *mickey-sc-first64*
8. *mickey-sc-first72*

Future SAT@home 2.0 experiment

grain-v1-sc-first72 and shrinking-sc72 suit well for SAT@home.

For grain-v1-sc-first72, using the found decomposition set (28 variables), each subinstance is solved within 0.19 seconds on average.

It is planned to vary the first 14 variables (out of 28) from the decomposition set on the project’s server.

The remaining 14 variables will be varied by a client application. For one instance of grain-v1-sc-first72, 16384 workunits will be created. To calculate one task, it would take about 52 minutes on a volunteer's host.

Future SAT@home 2.0 experiment

For shrinking-sc72, using the found decomposition set, each subinstance is solved in 0.62 seconds on average.

It is planned to vary the first 18 variables (out of 30) from the decomposition set on the project's server.

The remaining 12 variables will be varied by a client application. As a result, for one instance of shrinking-sc72, 262144 workunits will be created.

To calculate a task of one such workunit, it would take about 42 minutes on a volunteer’s host.
According to the estimations, It should take about 25 days to solve all six SAT instances in SAT@home.

Thank you for your attention!