A Framework for Accelerating Cryptographic SAT Solver with CUDA

Taeill Yoo[1], Yongjin Yeom[1], Daewan Han[2]
[1]Department of Financial Information Security, Kookmin University, South Korea
[2]The Attached Institute of ETRI, South Korea

Abstract

Making use of massively parallel computing with CUDA, we propose a new framework which provides a method for improving cryptographic SAT solver. With known plaintexts/ciphertexts, cryptographic SAT solvers have been trying to find the encryption key by solving the corresponding SAT problem. Since most ciphers can be expressed as SAT problems with more than 10,000 variables, it is still infeasible to find a solution even with the reduced-round ciphers in general.

In our framework, key variables are managed as pivot variables. During the solving procedure, if the number of undetermined key variables is small enough, the kernel function on GPU is invoked to perform the exhaustive search for the rest of the keys. Even when the kernel cannot reach the solution, it feedbacks a learned clause with useful information accelerating SAT solver in the host PC.

Cryptographic SAT solver

SAT Problem:
- Is a problem, for a given Boolean expression in CNF(Conjunctive Normal Form), finding a solution satisfying the CNF (to be TRUE) or proving no solutions exist.

SAT Solver:
- Is an algorithm which answers a solution to the SAT problem.
- If there exist no solutions, the solver has to prove it.

(Example) CNF with 6 variables and 4 clauses

Attached learned clause

$CNF = (\overline{x_1} \vee x_2) \wedge (\overline{x_3} \vee x_4) \wedge (\overline{x_5} \vee \overline{x_6}) \wedge (x_6 \vee \overline{x_5} \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3} \vee \overline{x_5})$

Literal, Clause

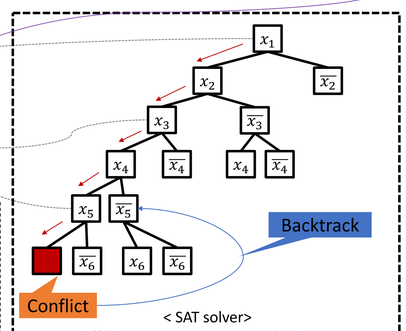
learning clause scheme

$(\overline{x_1} \vee \overline{x_3} \vee \overline{x_5})$

Learned Clause

(Cryptographic SAT solver)

SAT solver has been considered as a tool for cryptanalysis. For a given cipher with unknown encryption key, SAT solver tries to find the key by performing known plaintext attack as following steps:



< SAT solver>
Efficient exhaustive search algorithm
with backtracking & learning by conflicting

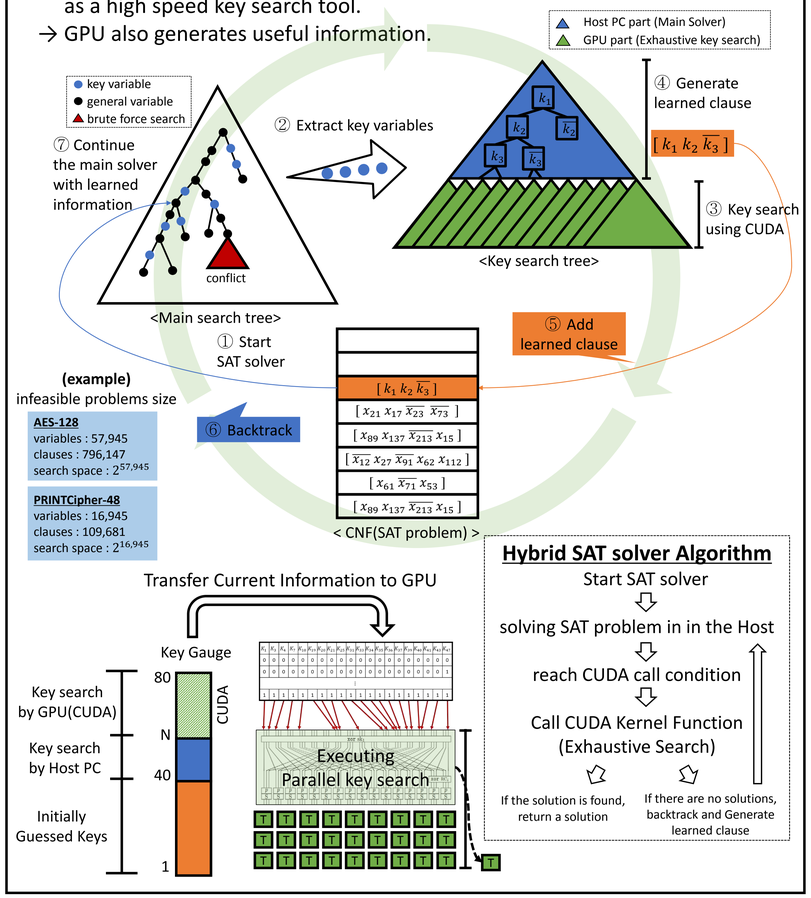
Target Block Cipher
⇩
Algebraic Equation
⇩
CNF
⇩
SAT solver (miniSAT)

- Step 1 : Express the target cipher as a system of algebraic equations
- Step 2 : Transform this into CNF(Conjunctive Normal Form)
- Step 3 : Solving SAT problem with the CNF using SAT solver

※ AES-128 : 57,945 variables, 796,147 clauses.
※ PRINTCipher-48 : 16,945 variables, 109,681 clauses.

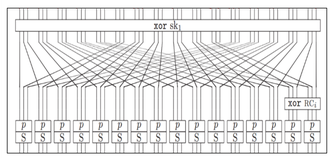
Main Idea

The main strategy is to extract a key search tree from the main tree.
- The key search tree contains only key variables
→ the solver handles key variables more importantly than others.
→ GPU, with parallel computing power, participates as a high speed key search tool.
→ GPU also generates useful information.



Hybrid SAT solver Algorithm
Start SAT solver
⇩
solving SAT problem in in the Host
⇩
reach CUDA call condition
⇩
Call CUDA Kernel Function (Exhaustive Search)
⇩
If the solution is found, return a solution
If there are no solutions, backtrack and Generate learned clause

Target Algorithm(PRINTCipher-48)



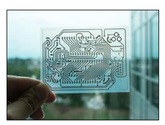
<Round Function of PRINTCipher-48>

PRINTCipher-48 is an encryption algorithm to protect important data in restricted computing environment such as RFID cards. It has been designed to assure integrity and authentication with minimum resources.

Specification
block size : 48
key size : 80
rounds : 48
- it uses identical key in each rounds without key schedule.

Structure of Round Function
Add RoundKey (sk1 : 48 bits)
Linear Diffusion
Combine Round Constant
Key-dependent Permutation
(sk2 : 32bits)
S-Box Layer


<Chip in the RFID Card>


<IC-Printed Circuit>

http://www.news.gatech.edu/2013/11/05/georgia-tech-develops-inkjet-based-circuits-fraction-time-and-cost
http://www.smartapps.tn/rfid2.htm

Experiments for PRINTCipher-48

In order to measure performance of this framework, we choose PRINTCipher-48 as an encryption algorithm because PRINTCipher-48 can be expressed simply in the aspect of algebra.

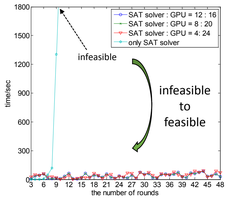
There has been several analysis for PRINTCipher-48 using SAT solver. Most successful results are 8 round and 6 round analysis for 15 bit and 35 bit key guessing, respectively.

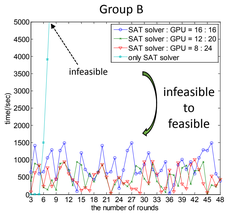
Test Group

In the key search tree, we measured influence of CUDA in the search performance by dividing portion of SAT solver with CUDA. We experimented that unknown keys are 28 bits and 32 bits, and also divided each cases with three groups. In the experiments, CUDA interplays with miniSAT for 16, 20, 24 bits exhaustive search.

Search Key Bits	SAT solver Key Bits	CUDA Key Bits
Group A (28)	12	16
	8	20
	4	24
Group B (32)	16	16
	12	20
	8	24

<Experimental Data Set>





In the experiments, we obtained following results for PRINTCipher-48.
- We improved time performance from infeasible to feasible.
- Hybrid SAT solver carried out efficient analysis for full(48) round.

※ Experimental Setup : Intel i7-4770K 3.5Ghz, 16GB RAM, GTX 780

Conclusion

- We propose a hybrid SAT solver based on miniSAT with CUDA.
- In our framework, GPU takes part in the solving procedure not only by executing brute-force key search but also by providing learned information to the main solver in the host PC.
- We improved time performance of SAT solver from infeasible range to feasible range for PRINTCipher-48.
- Further work will be considered including optimization of the performance, application to other ciphers, combining with other cryptanalytic attacks, etc.

Acknowledgements

1. This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (No. NRF-2014M3C4A7030648)
2. This research was partially supported by BK21PLUS through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (Grant No. 31Z20130012918)

References

[1] L. Knudsen, G. Leander, A. Poschmann, M.J.B.Robshaw, "PRINTCIPHER : A Block Cipher for IC-Printing", LNCS, Vol 6225, pp.16-32, CHES 2010
[2] S. Bulygin, J. Buchmann, "Algebraic Cryptanalysis of the Round-Reduced and Side Channel Analysis of the Full PRINTCipher-48", LNCS Vol. 7092, pp.54-75, 2011