Side Channel Attack On Stream Ciphers: A Three-Step Approach To State/Key Recovery

Satyam Kumar [1] Vishnu Asutosh Dasu [2] Anubhab Baksi [3]
Santanu Sarkar [1] **Dirmanto Jap** [3] Jakub Breier [4]
Shivam Bhasin [3]

[1] IIT Madras, India
[2] TCS Research and Innovation, Bangalore, India
[3] Nanyang Technological University, Singapore
[4] Silicon Austria Labs, Graz, Austria

Cryptographic Hardware and Embedded Systems, CHES-2022

Outline

Introduction I

Problem: Can we design a generic framework that can recover the secret information of a stream cipher with noisy side-channel traces?

- Most of the works that address the above problem are carried out in the Initialisation phase or use multiple IVs for noisy traces.
- However, a proper framework for SCA in the Pseudo-random phase of a stream cipher using a single IV on noisy traces is still missing.

Introduction II

Our Contribution: We have designed a generic framework that works as a state bit recovery/key recovery tool for NLFSR based stream cipher or cipher with a similar structure.

- Combined multiple tools (ML, MILP, SMT) in a single framework.
- Works in both Initialisation and Pseudo-random phase, even in the presence of noise.
- It can be carried out in a single (Key, IV) environment.
- Tested on `TRIVIUM` cipher implemented on 32-bit ARM Cortex-M3.

Comparison of Our Result with Previous Works

Table: A comparative study of our work with other stream cipher SCAs

Research Work	Year	Target Cipher(s)	Attack Phase	Noisy?	# IV?
Fischer et al. [3]	2007	GRAIN, TRIVIUM	Initialisation	Noisy	Multiple
Gierlichs et al. [4]	2008	eSTREAM Candidates	Initialisation	Noisy	–
Strobel [10]	2009	GRAIN-v1, TRIVIUM	Initialisation	Noisy	Multiple
Qu et al. [8]	2013	CRYPTO-I	Initialisation	Noiseless	Multiple
Chakraborty et al. [1]	2015	GRAIN Family	Initialisation	Noisy	Multiple
Tena-Sánchez et al. [12, 11]	2015	TRIVIUM	Initialisation	Noisy	Multiple
Kazmi et al. [7]	2017	CRYPTO-I	Pseudo-random	Noiseless	Single
		TRIVIUM, BIVIUM-B, GRAIN	Initialisation	Noiseless	Single
Jurecek et al. [6]	2019	A5/1	Initialisation	Noiseless	–
Sim et al. [9]	2020	LR-KEYMILL, TRIVIUM	Initialisation	Noiseless	Multiple
Our Paper	2022	TRIVIUM	Initialisation, Pseudo-random	Noisy	Single

Framework Construction

Two main steps:

- Predict the HW/HD from the side-channel traces.
- Fit the information to the tool to retrieve the secret
    - SMT instance is used to return a solution for unknown state/key in a reasonable time.

Brief Overview

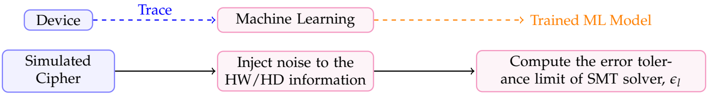


Figure: Offline Phase

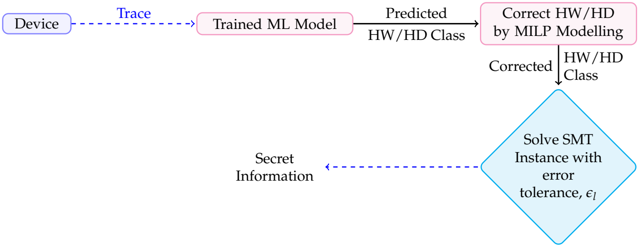


Figure: Online Phase-Brief Description

Error Tolerance

Let the array HW_{org} contains original Hamming weight sequence and HW_{pre} contains the given/predicted/available Hamming weight sequence.

Error tolerance (ϵ): We say that the Hamming weight $HW_{pre}[i]$ is obtained with an error tolerance ϵ, if

$$HW_{org}[i] - \epsilon \leq HW_{pre}[i] \leq HW_{org}[i] + \epsilon \quad (1)$$

MILP

A Mixed Integer Linear Programming (MILP) refers to the problem of the following form:

Objective: $Min/Max : c^T x \;\; (= c_1x_1 + c_2x_2 + \ldots + c_nx_n)$

Subject to constraints:

$Ax = b$ [Linear Constraints]

$lb \leq x \leq ub$ [Bound on variables]

$x_j \in \mathbb{Z}$ for some or all $j \in \{1, 2, \ldots, n\}$

Some of the MILP Solvers are Gurobi, CPLEX, CBC etc.

SMT

Satisfiability Modulo Theories (SMT)

- A generalisation of Boolean Satisfiability Problem (SAT) to a larger domain which involves real numbers, integer, bit vectors etc.
- Example: $(x_1 \leq 2) \wedge (x_1 + x_2 \leq x_3) \wedge \ldots$
- SMT Solvers: Z3, Simple Theorem Prover (STP), Boolector etc.

Trivium: An Overview

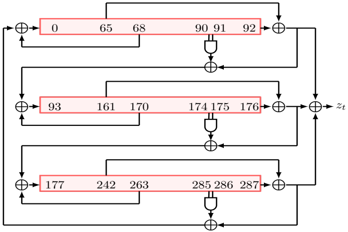


Figure: Trivium Design

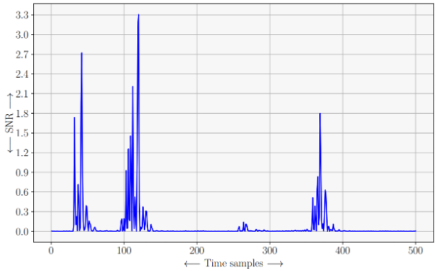
- Secret Key and IV are both 80-bit.
- State update function is invertible, i.e., state bits recovery can lead to secret key recovery.
- Comprises three NLFSRs of size: 93-bit, 84-bit and 111-bit respectively, with Internal state of size: 288-bit.

Experiments: Trace Collection and ML Training

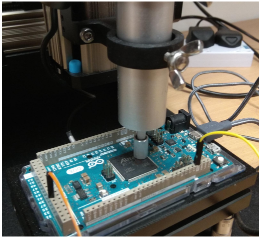
For the measurements, we used the following setup:

- The target is implemented in assembly on ARM Cortex-M3.
- Riscure high precision EM probe is used.
- $2^{21.17}$ traces used in total for training, validation and testing.

We consider a supervised classification problem with 33 classes, based on the HW value and trained MLP classifier.



(a) Best SNR from grid search



(b) EM SCA setup

ML Accuracy and Error Tolerance

Table: Trade-off between ML tolerance and SMT solution time (sec.) for `TRIVIUM`

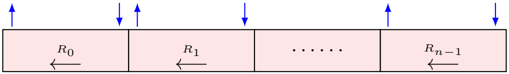
	Tolerance (ϵ)				
	0	1	2	3	4
ML Accuracy	0.39330	0.86678	0.98262	0.99784	0.99967
SMT Solving Time	5.42 (110)	254.36 (110)	1819.21 (130)	28755.36 (170)	76797.41 (130)

$(\cdot)$: Number of `TRIVIUM` pseudo-random rounds considered

- We use MLP with 2 Hidden Layers: (128, 128).
- $\epsilon = 3$ provides a good speed/accuracy trade-off.
- However, 0.216% classes are still predicted incorrectly.
- We use MILP to correct these predictions.

MILP model

- Using MILP model, we tried to find a sequence of HWs that follows the same pattern as that of the original HWs sequence and is near to the predicted HWs sequence, HW_{pre}.
- We constructed 3 types of constraints for the MILP model based upon the incoming/outgoing bits relation from the stream cipher construction.



(a) General structure of a stream cipher based on FSRs

(b) Consecutive blocks of internal states

Figure: Internal state and its block representation

MILP Results

- Implemented MILP model on a simulated TRIVIUM cipher.
- Took probability distribution of ML output into account.
- Ran 1000 experiments for each set of parameters in the Table below.

Success Rate: Success rate at 110 rounds ($\epsilon = 3$) is 97.6%, taking 3.36 seconds on average, and if we repeat the experiment twice with different sequence of HW, the success rate is 99.94%.

Table: Success rate of MILP model for TRIVIUM

Constraint Type	#Rounds	Success Rate	Solution Time	
			Mean (sec.)	S.D. (sec.)
I + II + III	110	97.6	3.36	1.37
	130	95.9	4.84	0.98
	150	96.8	6.38	0.77
	170	94.6	7.86	1.18
	180	94.3	8.61	1.01
	200	93.3	10.53	0.83

Tolerance = 3; Number of Threads used = 1

SMT to Solve for Unknown State/Key I

- We convert the whole system of equations/constraints into a system of modular equations/constraints, which is well supported in Z3.
- We feed all the constraints to the SMT solver, and if all of the HW classes predictions are within tolerance ϵ (≤ 3), the solver returns a solution in a feasible time, which can be verified.
    1. If the solution cannot be verified, run SMT again by increasing the number of rounds and predicted sequence length.
    2. If it returns inconsistent, that means at least one predicted HW class fall outside the tolerance ϵ, and recovery procedure might be needed.

SMT to Solve for Unknown State/Key II

Table: Results on TRIVIUM in pseudo-random phase (unique solution)

Leakage Model	Tolerance	# Rounds	Trials	Mean (sec.)	S.D. (sec.)
HW/8	1	110	20	1.16	0.05
	2	110	20	1.37	0.07
	3	110	20	1.58	0.13
HW/16	1	110	20	3.91	0.94
	2	110	20	7.38	2.18
	3	110	20	15.41	6.35
	4	110	20	91.40	187.50
		130	20	40.01	22.85
		150	20	39.89	18.62
HW/32	1	110	20	239.74	192.64
	2	110	20	7975.88	9277.67
		130	20	4764.87	4489.14
	3	130	6	122582.24	65397.23
		150	6	49975.49	31924.09
		180	5	36288.8	27153.63
	4	130	1	475778.30	–
		150	3	49445.14	38190.05
		170	2	226005.79	71432.18

(a) Without key-stream information

Leakage Model	Tolerance	# Rounds	Trials	Mean (sec.)	S.D. (sec.)
HW/32	0	110	20	5.42	1.255
	1	70	20	1309.19	1144.36
		90	20	821.76	1444.34
		110	20	254.36	195.43
	2	70	1	3408.72	–
		90	8	17404.32	27533.08
		110	16	10628.26	13962.97
		130	20	1819.21	1558.38
	3	110	1	140523.60	–
		130	3	44911.09	31619.74
		170	1	28755.36	–
	4	130	1	76797.41	–
		170	1	12582.60	–

(b) With key-stream information

SMT to Solve for Unknown State/Key III

- For HW Model:
  - We can recover the state bit of TRIVIUM uniquely in 28763.22 seconds for 170 rounds (with key-stream equations) with probability 0.946, and in 36297.41 seconds for 180 rounds (without key-stream equation) with probability 0.943.
  - For HW/32 in initialization phase, HW classes can be predicted with accuracy 100% for $\epsilon = 7$. Thus, MILP is not used, and best solution can achieved for 79.49 seconds (170 rounds).

SMT to Solve for Unknown State/Key IV

- For HD Model:
  - For initialization phase, up to tolerance 1, we can recover state bits without any guess. However, for tolerance 2 and 3, it needs at least 10 and 20 guessed bits respectively.
  - For pseudo-random phase, with guess of 140 state bits, we can get the results, but it exceeds the exhaustive search complexity on key bits (80 bits).

Summary of the Proposed Approach

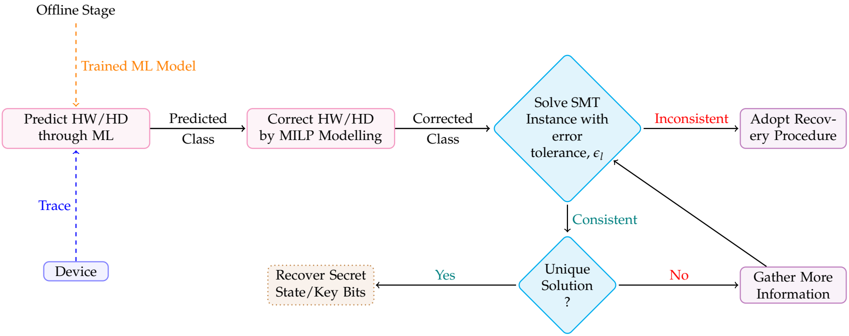


Figure: Final Framework for SCA

Summary

- We developed a generic framework that recovers the state/key from stream ciphers and related constructions from the side-channel information (power or EM).
- Our framework is able to attack the initialisation phase (i.e., before the cipher reaches its pseudo-random phase) and, more importantly even after the cipher reaches its pseudo-random phase to produce key-stream.
- We have tested with different variation of SNR and we observed that beyond SNR of 1.12124, the success probability drops to 0, and as such, SNR 1.12124 is the threshold for our framework with the current experiment setting.

Future Works

- **Analytical approach for less than perfect accuracy:** Consider another analytical approach, such as a *Hidden Markov Model* (HMM), that can work (with a high probability) when the accuracy is lower than 100%.
  - This can take some load from the ML module and can potentially remove the MILP module.
- **Form of leakage function:** Extension of our framework for polynomial leakage function [5] and weighted HW/HD leakage function. [2].
- **Improvement of ML:** Experiments with other types of ML models, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), or Long Short-Term Memory (LSTM) to improve the accuracy of ML model.

Thank You

References I

[1] Abhishek Chakraborty, Bodhisatwa Mazumdar, and Debdeep Mukhopadhyay. "Combined Side-Channel and Fault Analysis Attack on Protected Grain Family of Stream Ciphers". In: *IACR Cryptol. ePrint Arch.* 2015 (2015), p. 602. URL: http://eprint.iacr.org/2015/602.

[2] Julien Doget et al. "Univariate side channel attacks and leakage modeling". In: *J. Cryptogr. Eng.* 1.2 (2011), pp. 123–144. DOI: 10.1007/s13389-011-0010-2. URL: https://doi.org/10.1007/s13389-011-0010-2.

References II

[3] Wieland Fischer et al. "Differential Power Analysis of Stream Ciphers". In: *Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings*. Ed. by Masayuki Abe. Vol. 4377. Lecture Notes in Computer Science. Springer, 2007, pp. 257–270. DOI: 10.1007/11967668_17. URL: https://doi.org/10.1007/11967668%5C_17.

[4] Benedikt Gierlichs et al. "Susceptibility of eSTREAM Candidates towards Side Channel Analysis. SASC –The State of the Art of Stream Ciphers". In: *Workshop Record*. 2008, pp. 123–150.

References III

[5] Vincent Grosso, François-Xavier Standaert, and Emmanuel Prouff. "Low Entropy Masking Schemes, Revisited". In: *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*. Ed. by Aurélien Francillon and Pankaj Rohatgi. Vol. 8419. Lecture Notes in Computer Science. Springer, 2013, pp. 33–43. DOI: 10.1007/978-3-319-08302-5_3. URL: https://doi.org/10.1007/978-3-319-08302-5%5C_3.

[6] Martin Jurecek, Jirí Bucek, and Róbert Lórencz. "Side-Channel Attack on the A5/1 Stream Cipher". In: *2019 22nd Euromicro Conference on Digital System Design (DSD)*. IEEE. 2019, pp. 633–638.

References IV

[7] Asif Raza Kazmi et al. "Algebraic side channel attack on trivium and grain ciphers". In: *IEEE Access* 5 (2017), pp. 23958–23968.

[8] Bo Qu et al. "Differential power analysis of stream ciphers with LFSRs". In: *Comput. Math. Appl.* 65.9 (2013), pp. 1291–1299. DOI: 10.1016/j.camwa.2012.02.024. URL: https://doi.org/10.1016/j.camwa.2012.02.024.

[9] Siang Meng Sim, Dirmanto Jap, and Shivam Bhasin. "DAPA: Differential Analysis aided Power Attack on (Non-) Linear Feedback Shift Registers (Extended version).". In: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 1241. URL: https://eprint.iacr.org/2020/1241.pdf.

References V

[10] Daehyun Strobel. "Side channel analysis attacks on stream ciphers". In: *Masterarbeit Ruhr-Universität Bochum, Lehrstuhl Embedded Security* (2009). https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.182.1943&rep=rep1&type=pdf.

[11] Erica Tena-Sánchez and Antonio J Acosta. "Optimized DPA attack on Trivium stream cipher using correlation shape distinguishers". In: *2015 Conference on Design of Circuits and Integrated Systems (DCIS)*. IEEE. 2015, pp. 1–6.

References VI

[12] Erica Tena-Sánchez and Antonio J. Acosta. “DPA vulnerability analysis on Trivium stream cipher using an optimized power model”. In: *2015 IEEE International Symposium on Circuits and Systems, ISCAS 2015, Lisbon, Portugal, May 24-27, 2015*. IEEE, 2015, pp. 1846–1849. DOI: 10.1109/ISCAS.2015.7169016. URL: https://doi.org/10.1109/ISCAS.2015.7169016.