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

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

Cryptographic Hardware and Embedded Systems, CHES-2022
Outline

1. Introduction
2. Related Works
3. Brief Overview
4. Framework Description
   - Preliminaries
   - Experiments
5. Conclusion
Introduction

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.
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

<table>
<thead>
<tr>
<th>Research Work</th>
<th>Year</th>
<th>Target Cipher(s)</th>
<th>Attack Phase</th>
<th>Noisy?</th>
<th># IV?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fischer et al. [3]</td>
<td>2007</td>
<td>GRAIN, TRIVIUM</td>
<td>Initialisation</td>
<td>Noisy</td>
<td>Multiple</td>
</tr>
<tr>
<td>Strobel [10]</td>
<td>2009</td>
<td>GRAIN-v1, TRIVIUM</td>
<td>Initialisation</td>
<td>Noisy</td>
<td>Multiple</td>
</tr>
<tr>
<td>Qu et al. [8]</td>
<td>2013</td>
<td>CRYPTO-I</td>
<td>Initialisation</td>
<td>Noiseless</td>
<td>Multiple</td>
</tr>
<tr>
<td>Chakraborty et al. [1]</td>
<td>2015</td>
<td>GRAIN Family</td>
<td>Initialisation</td>
<td>Noisy</td>
<td>Multiple</td>
</tr>
<tr>
<td>Tena-Sánchez et al. [12, 11]</td>
<td>2015</td>
<td>TRIVIUM</td>
<td>Initialisation</td>
<td>Noisy</td>
<td>Multiple</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TRIVIUM, BIVIUM-B, GRAIN</td>
<td>Initialisation</td>
<td>Noiseless</td>
<td>Single</td>
</tr>
<tr>
<td>Jurecek et al. [6]</td>
<td>2019</td>
<td>A5/1</td>
<td>Initialisation</td>
<td>Noiseless</td>
<td>–</td>
</tr>
<tr>
<td>Sim et al. [9]</td>
<td>2020</td>
<td>LR-KEYMILL, TRIVIUM</td>
<td>Initialisation</td>
<td>Noiseless</td>
<td>Multiple</td>
</tr>
<tr>
<td><strong>Our Paper</strong></td>
<td>2022</td>
<td>TRIVIUM</td>
<td>Initialisation, Pseudo-random</td>
<td>Noisy</td>
<td>Single</td>
</tr>
</tbody>
</table>
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** ($\epsilon$): We say that the Hamming weight $HW_{pre}[i]$ is obtained with an error tolerance $\epsilon$, if

$$HW_{org}[i] - \epsilon \leq HW_{pre}[i] \leq HW_{org}[i] + \epsilon$$

(1)
A Mixed Integer Linear Programming (MILP) refers to the problem of the following form:

**Objective:** $\text{Min/Max} : c^T x \ (= c_1 x_1 + c_2 x_2 + \ldots + c_n x_n)$

**Subject to constraints:**

- $Ax = b \ [\text{Linear Constraints}]
- lb \leq x \leq ub \ [\text{Bound on variables}]
- x_j \in \mathbb{Z} \ \text{for some or all} \ j \in \{1, 2, \ldots, n\}$

Some of the MILP Solvers are **Gurobi**, **CPLEX**, **CBC** etc.
Satisfiability Modulo Theories (SMT)

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

- 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.

**Figure:** Trivium Design
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.
ML Accuracy and Error Tolerance

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

<table>
<thead>
<tr>
<th>Tolerance ($\epsilon$)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML Accuracy</td>
<td>0.39330</td>
<td>0.86678</td>
<td>0.98262</td>
<td>0.99784</td>
<td>0.99967</td>
</tr>
<tr>
<td>SMT Solving Time</td>
<td>5.42 (110)</td>
<td>254.36 (110)</td>
<td>1819.21 (130)</td>
<td>28755.36 (170)</td>
<td>76797.41 (130)</td>
</tr>
</tbody>
</table>

(•) : 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.
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

<table>
<thead>
<tr>
<th>Constraint Type</th>
<th>#Rounds</th>
<th>Success Rate</th>
<th>Solution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean (sec.)</td>
<td>S.D. (sec.)</td>
</tr>
<tr>
<td>I + II + III</td>
<td>110</td>
<td>97.6</td>
<td>3.36</td>
</tr>
<tr>
<td></td>
<td>130</td>
<td>95.9</td>
<td>4.84</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>96.8</td>
<td>6.38</td>
</tr>
<tr>
<td></td>
<td>170</td>
<td>94.6</td>
<td>7.86</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>94.3</td>
<td>8.61</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>93.3</td>
<td>10.53</td>
</tr>
</tbody>
</table>

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 $\epsilon \ (\leq 3)$, 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 $\epsilon$, and recovery procedure might be needed.
### Table: Results on TRIVIUM in pseudo-random phase (unique solution)

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

(a) Without key-stream information

(b) With key-stream information
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).
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

ANY QUESTIONS
References


References V
