Susceptibility of eSTREAM Candidates towards Side Channel Analysis

Benedikt Gierlichs¹, Lejla Batina¹, Christophe Clavier²
Thomas Eisenbarth³, Aline Gouget², Helena Handschuh⁴
Timo Kasper³, Kerstin Lemke-Rust⁵, Stefan Mangard⁶
Amir Moradi³,⁷, Elisabeth Oswald⁸

K.U. Leuven COSIC, Gemalto Security Labs, Horst Görtz Institute for IT-Security,
Spansion, T-Systems GEI GmbH, Infineon Technologies AG, Sharif University of
technology Tehran, University Bristol

SASC ’08, Lausanne

Outline

1 Introduction
   • Motivation
   • Contribution
   • Evaluation criteria

2 Summary software candidates

3 Summary hardware candidates

4 Conclusion

Cipher candidates in phase 3 of eSTREAM

<table>
<thead>
<tr>
<th>Profile 1 (SW)</th>
<th>Profile 2 (HW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CryptMT (CryptMT Version 3)</td>
<td>DECIM (DECIM v2, DECIM-T28)</td>
</tr>
<tr>
<td>Dragon</td>
<td>Edon80</td>
</tr>
<tr>
<td>HC (HC-128, HC-256)</td>
<td>F-FCSR (F-FCSR-H v2, F-FCSR-16)</td>
</tr>
<tr>
<td>LEX (LEX-128, LEX-192, LEX-256)</td>
<td>Grain (Grain v1, Grain-128)</td>
</tr>
<tr>
<td>NLS (NLSv2, encryption only)</td>
<td>MICKEY (MICKEY 2.0, MICKEY-128 2.0)</td>
</tr>
<tr>
<td>Rabbit</td>
<td>Moustique</td>
</tr>
<tr>
<td>Salsa20</td>
<td>Pomaranch (Pomaranch Version 3)</td>
</tr>
<tr>
<td>SOSEMANUK</td>
<td>Trivium</td>
</tr>
</tbody>
</table>

Which ones are good? Which ones are better?

- Criteria include: security, throughput, codesize / area, memory requirements, power requirements, etc.
- So far: security ~ resistance to classical cryptanalysis
- What about Side Channel Attacks!?
Implementations of a cipher are exposed to a wider range of adversaries than the algorithm. Often: proximity, physical access, no time constraints (think off smartcards, secure flash memory, etc.) → Side Channel and Fault Attacks.

Implementation of a secure algorithm ≠ a secure implementation! But need good code for real-world applications soon.

We assess susceptibility towards SCA in a theoretic approach, based on cipher specification and well-established implementation techniques, side channel leakage models, side channel attacks (Timing, Power). Though, SCAs are implementation attacks (cipher, implementation, platform, logic, ...).

Evaluation points out potential vulnerabilities. Useful at time of implementation. Intuition about cost of countermeasures. Similar work was done for AES and NESSIE candidates.
We assess susceptibility towards SCA in a theoretic approach, based on cipher specification and well established:

- implementation techniques
- side channel leakage models
- side channel attacks (Timing, Power)

Though, SCAs are implementation attacks (cipher, implementation, platform, logic, ...)

Evaluation points out potential vulnerabilities

Useful at time of implementation

Intuition about cost of countermeasures

Similar work was done for AES and NESSIE candidates.

Adversarial model:

- knows IV and keystream, optionally chosen IVs
- able to reset the cipher, *i.e.* re-invoke IV / key setup
- aims at key recovery

Timing Analysis (global)

- exploits a dependency between the execution time of an algorithm and the course of its secret internal state
- conditional branches depending on the internal state?
- look-up tables? Cache timing attacks!

Power Analysis (internal)

- Leakage Model: adversary observes perfect Hamming weight/distance leakage, external noise neglected

Adversarial model:

- knows IV and keystream, optionally chosen IVs
- able to reset the cipher, *i.e.* re-invoke IV / key setup
- aims at key recovery

Timing Analysis (global)

- exploits a dependency between the execution time of an algorithm and the course of its secret internal state
- conditional branches depending on the internal state?
- look-up tables? Cache timing attacks!

Power Analysis (internal)

- Leakage Model: adversary observes perfect Hamming weight/distance leakage, external noise neglected
**Adversarial model:**
- knows IV and keystream, optionally chosen IVs
- able to reset the cipher, i.e. re-invoke IV / key setup
- aims at key recovery

**Timing Analysis (global)**
- exploits a dependency between the execution time of an algorithm and the course of its secret internal state
- conditional branches depending on the internal state?
- look-up tables? Cache timing attacks!

**Power Analysis (internal)**
- Leakage Model: adversary observes perfect Hamming weight/distance leakage, external noise neglected

---

**Simple Power Analysis**
- exploits a dependency between the power dissipation of a device and **instructions** and/or (secret) data being processed
- conditional branches depending on the internal state?

**Differential Power Analysis**
- exploits a dependency between the power dissipation of a device and (secret) data being processed
- in particular: differences in power dissipation due to processing of different data values
- what are the fundamental operations of the stream cipher?

**Intuition for cost of countermeasures, assuming that**
- masking of boolean operations is easy
- masking of small tables (up to 256 bytes) is medium
- masking of larger tables and combinations of boolean and arithmetic operations is expensive
- assumptions based on experience with publicly available countermeasures for block ciphers
Table: Summary of (theoretical) side channel susceptibility, profile 1 candidates

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Exploitable (cache) timing vulnerability</th>
<th>Exploitable conditional branches</th>
<th>Exploitable HW leakage of data</th>
<th>Exploitable DPA vulnerability</th>
<th>Masking effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>CryptMT</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>medium</td>
</tr>
<tr>
<td>Dragon:</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>high</td>
</tr>
<tr>
<td>HC:</td>
<td>yes</td>
<td>maybe</td>
<td>yes</td>
<td>yes</td>
<td>high</td>
</tr>
<tr>
<td>LEX:</td>
<td>yes</td>
<td>maybe</td>
<td>yes</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>NLS:</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>Rabbit:</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>Salsa20:</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>SOSEMANUK:</td>
<td>maybe</td>
<td>no</td>
<td>yes</td>
<td>high</td>
<td>high</td>
</tr>
</tbody>
</table>

criterion “exploitable” (cache) timing vulnerability
- CryptMT, Rabbit, Salsa20 appear to be immune
- Dragon, LEX, HC, NLS, SOSEMANUK contain look-up tables

Something you should know...
- Defining (a part of) the IV as a counter is a good thing!
- It can turn a simple DPA attack infeasible. Why?
  - Basic requirement for DPA: a sensitive intermediate result $x := f(IV, k)$, $IV$ known and varying, $k$ unknown constant
  - Preferably $x$ depends on a small part of $k$
  - Now think of, say, a 32-bit device and Lex in SW:
    - In order to recover the 8 MSBs of a 32-bit chunk of the key, an adversary needs to either obtain $> 2^{24}$ measurements or attack an $f'$ deeper in the algorithm, higher complexity
  - For you: (maybe) small number; for us: wait for a year!
  - Interesting option, also for hardware ciphers: implementation in a DPA-resistant logic style (e.g. SABL, WDDL) would increase area by a factor $\sim 3$ to 5 and power dissipation by a factor $\sim 2$ to 3
Defining (a part of) the IV as a counter is a good thing!

It can turn a simple DPA attack infeasible. Why?

Basic requirement for DPA: a sensitive intermediate result

\[ x := f(IV, k) \]

\( IV \) known and varying, \( k \) unknown constant

Preferably \( x \) depends on a small part of \( k \)

Now think of, say, a 32-bit device and \( \text{Lex} \) in SW:

In order to recover the 8 MSBs of a 32-bit chunk of the key, an adversary needs to either obtain \( > 2^{24} \) measurements or attack an \( f' \) deeper in the algorithm, higher complexity

For you: (maybe) small number; for us: wait for a year!

Interesting option, also for hardware ciphers; implementation in a DPA-resistant logic style (e.g. SABL, WDDL) would increase area by a factor \( \sim 3 \) to 5 and power dissipation by a factor \( \sim 2 \) to 3