Susceptibility of eSTREAM Candidates towards Side Channel Analysis

Benedikt Gierlichs[1], Lejla Batina[1], Christophe Clavier[2]
Thomas Eisenbarth[3], Aline Gouget[2], Helena Handschuh[4]
Timo Kasper[3], Kerstin Lemke-Rust[5], Stefan Mangard[6]
Amir Moradi[3,7], Elisabeth Oswald[8]

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

- Cipher candidates in phase 3 of eSTREAM

Profile 1 (SW)	Profile 2 (HW)
CryptMT (CryptMT Version 3)	DECIM (DECIM v2, DECIM-128)
Dragon	Edon80
HC (HC-128, HC-256)	F-FCSR (F-FCSR-H v2, F-FCSR-16)
LEX (LEX-128, LEX-192, LEX-256)	Grain (Grain v1, Grain-128)
NLS (NLSv2, encryption only)	MICKEY (MICKEY 2.0, MICKEY-128 2.0)
Rabbit	Moustique
Salsa20	Pomaranch (Pomaranch Version 3)
SOSEMANUK	Trivium

- Which ones are good? Which ones are better?
- Criteria include: security, throughput, codesize / area, memory requirements, power requirements, etc.
- So far: security $\sim$ resistance to classical cryptanalysis
- What about Side Channel Attacks!?

Introduction Summary profile 1 Summary profile 2 Conclusio Motivation Contribution Evaluation criteria

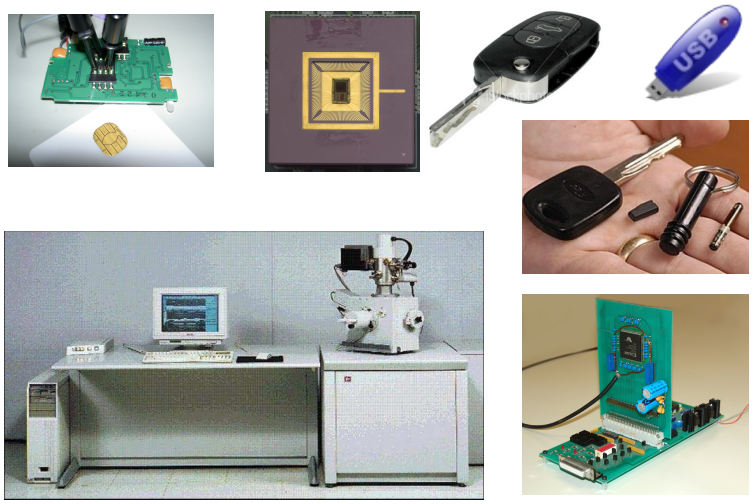
- Cipher candidates in phase 3 of eSTREAM

Profile 1 (SW)	Profile 2 (HW)
CryptMT (CryptMT Version 3)	DECIM (DECIM v2, DECIM-128)
Dragon	Edon80
HC (HC-128, HC-256)	F-FCSR (F-FCSR-H v2, F-FCSR-16)
LEX (LEX-128, LEX-192, LEX-256)	Grain (Grain v1, Grain-128)
NLS (NLSv2, encryption only)	MICKEY (MICKEY 2.0, MICKEY-128 2.0)
Rabbit	Moustique
Salsa20	Pomaranch (Pomaranch Version 3)
SOSEMANUK	Trivium

- Which ones are good? Which ones are better?
- Criteria include: security, throughput, codesize / area, memory requirements, power requirements, etc.
- So far: security $\sim$ resistance to classical cryptanalysis
- What about Side Channel Attacks!?

B. Gierlichs SCA of eSTREAM Candidates SASC '08, Lausanne 2/13

- 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



- Implementations of a cipher are exposed to a wider range of adversaries than the algorithm
- Often: proximity, physical access, no time constraints → Side Channel and Fault Attacks
- Implementation of a secure algorithm $\neq$ a secure implementation!
- But need good code for real-world applications soon
  - implementation of a secure and efficient cipher (usually: + throughput, - codesize/area, - memory, - power)
  - secure against physical attacks (usually: - throughput, + codesize/area, + memory, + power)
  - (ideally for 50% of the original cost...)

- We asses 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 asses 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

Introduction Summary profile 1 Summary profile 2 Conclusio Motivation Contribution Evaluation criteria

- We asses 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

B. Gierlichs SCA of eSTREAM Candidates SASC '08, Lausanne 6/13

- 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

Introduction Summary profile 1 Summary profile 2 Conclusio Motivation Contribution Evaluation criteria

- 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

B. Gierlichs SCA of eSTREAM Candidates SASC '08, Lausanne 7/13

- 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

Introduction Summary profile 1 Summary profile 2 Conclusio Motivation Contribution Evaluation criteria

- 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

B. Gierlichs SCA of eSTREAM Candidates SASC '08, Lausanne 8/13

Table: Summary of (theoretical) side channel susceptibility, profile 1 candidates

Cipher	Exploitable (cache) timing vulnerability	Exploitable conditional branches	Exploitable HW leakage of data	Exploitable DPA vulnerability	DPA attack complexity	Masking effort
CryptMT:	no	no	yes	yes	medium	high
Dragon:	**yes**	no	yes	yes	low	high
HC:	maybe	no	yes	yes	low	high
LEX:	**yes**	maybe	yes	yes	low	medium
NLS:	**yes**	no	yes	yes	high	high
Rabbit:	no	no	yes	yes	medium	high
Salsa20:	no	no	yes	yes	low	high
SOSEMANUK:	maybe	no	yes	yes	low	high

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

Table: Updated summary of (theoretical) side channel susceptibility, profile 2 candidates

Cipher	Exploitable timing vulnerability	Exploitable conditional branches	Exploitable HW leakage of data	Exploitable DPA vulnerability	DPA attack complexity
Decim:	maybe	**yes**	yes	yes	high
Edon-80:	no	no	yes	yes	medium
F-FCSR:	no	no	yes	yes	medium
Grain:	no	no	yes	yes	medium
MICKEY:	**yes**	**yes**	yes	yes	medium
Moustique:	no	no	yes	yes	medium
Trivium:	no	no	yes	yes	medium

- criteria "exploitable" timing vulnerability and conditional branches
  - Decim and MICKEY 2.0 contain conditional branches

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 ~ 3 to 5 and power dissipation by a factor ~ 2 to 3

Introduction Summary profile 1 Summary profile 2 Conclusio

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 ~ 3 to 5 and power dissipation by a factor ~ 2 to 3

B. Gierlichs SCA of eSTREAM Candidates SASC '08, Lausanne 11/13

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

Introduction Summary profile 1 Summary profile 2 Conclusio

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

B. Gierlichs SCA of eSTREAM Candidates SASC '08, Lausanne 11/13

Introduction Summary profile 1 Summary profile 2 Conclusio

- We presented a high-level analysis of SCA susceptibility for eSTREAM candidates
- For software candidates table lookups are crucial
- For hardware candidates conditional branches are crucial

- A real-world cipher implementation has to resist more than cryptanalysis
- Implementation of a secure algorithm $\neq$ a secure implementation
- Side channel and fault attacks are real threats and applied every day!

Introduction Summary profile 1 Summary profile 2 Conclusio

Thank you for your attention.

Questions?