Keymill: Side-Channel Resilient Key Generator

A New Concept for SCA-Security by Design

Mostafa Taha, Arash Reyhani-Masoleh, and Patrick Schaumont

Post-Doctoral Fellow at Western University

Selected Areas in Cryptography (SAC'16)

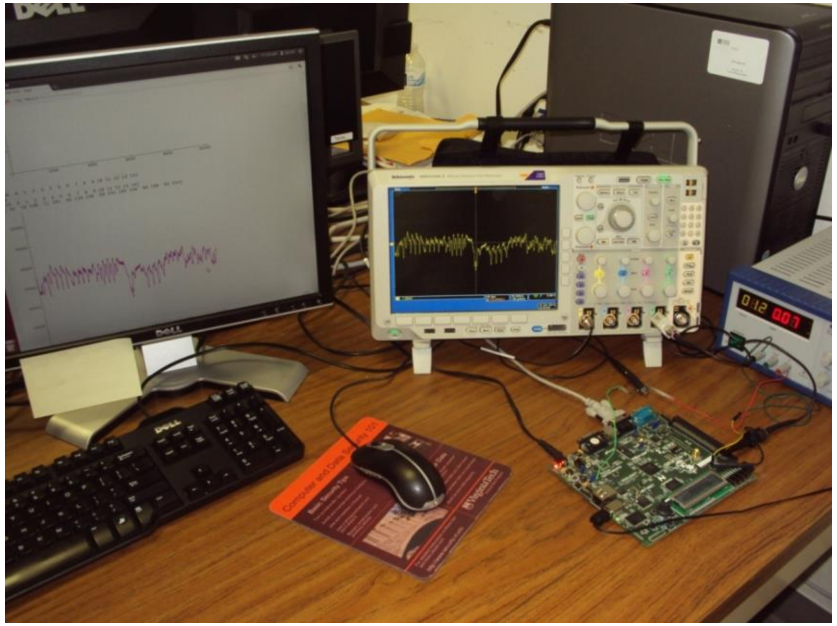
12 August, 2016

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under the Discovery and Discovery Accelerate Supplement (DAS) Grants awarded to A. Reyhani-Masoleh.

Outline

- We propose:
  - A new concept for protection against Side-Channel Analysis, Promoting for new cryptographic designs.
  - A new definition for SCA-security.
  - A new validation test for SCA-security.
  - A practical, lightweight realization, where
    SCA-security = Mathematical-security
- Outline:
  - Introduction to Side-Channel Analysis.
  - The current methods of protection.
  - The new concept for protection.
  - Realization.

Side-Channel Analysis



Side-Channel Analysis

Inputs →

Measured Power
2
3
5
4
:

Subkey Hypothesis

K_0		K_1		K_2	
Sensitive Var.	Modeled Power	Sensitive Var.	Modeled Power	Sensitive Var.	Modeled Power
0x0F	4	0x82	2	0xF1	5
0xAA	4	0x51	3	0x4E	4
0xD3	5	0xA3	4	0x0B	3
0x31	3	0xC7	5	0x92	3
:	:	:	:	:	:

Correlation

Conditions for Side-Channel Analysis

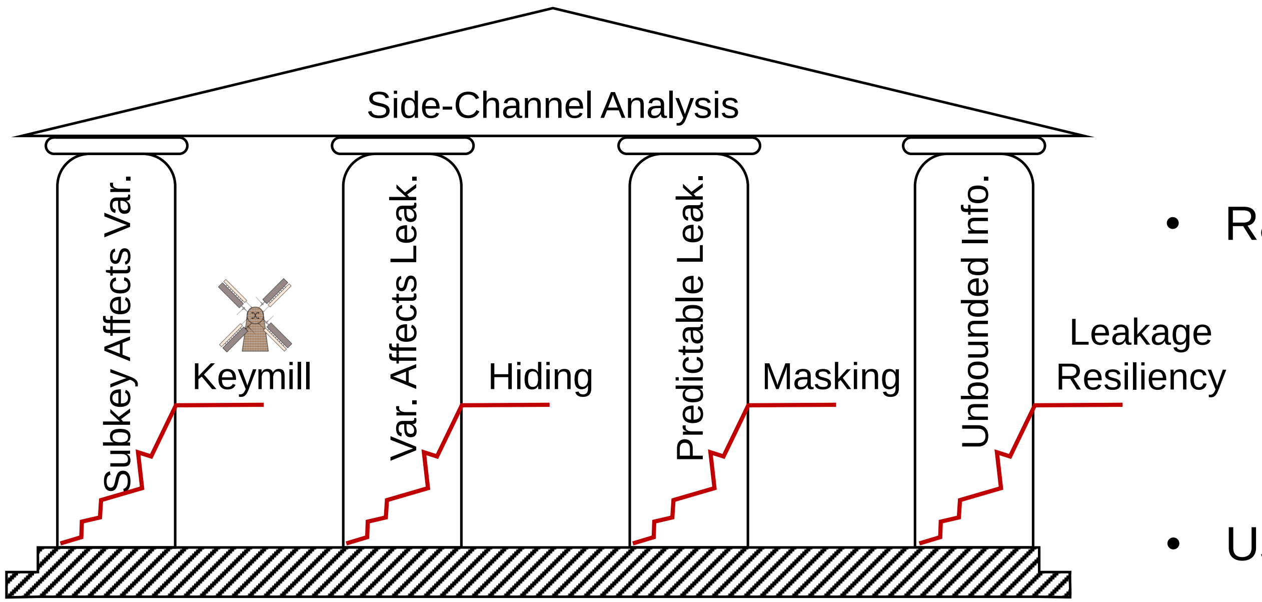
1. A small part of the key affects an intermediate variable.

2. The intermediate variable affects observable leakage.

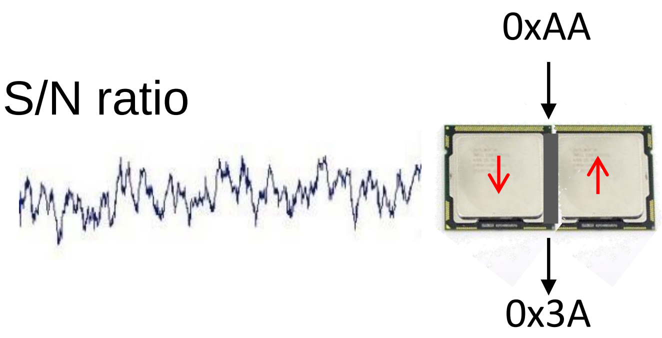
3. The observable leakage can be predicted at a known input.

4. The adversary can collect unbounded information against the small part of the key.

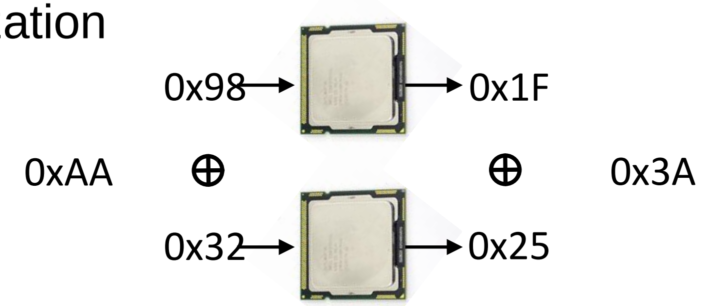
Current Countermeasures



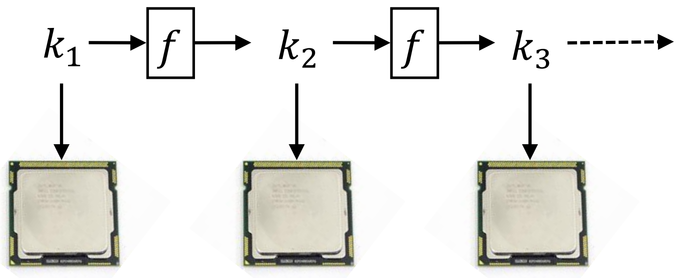
- Reduce S/N ratio



- Randomization

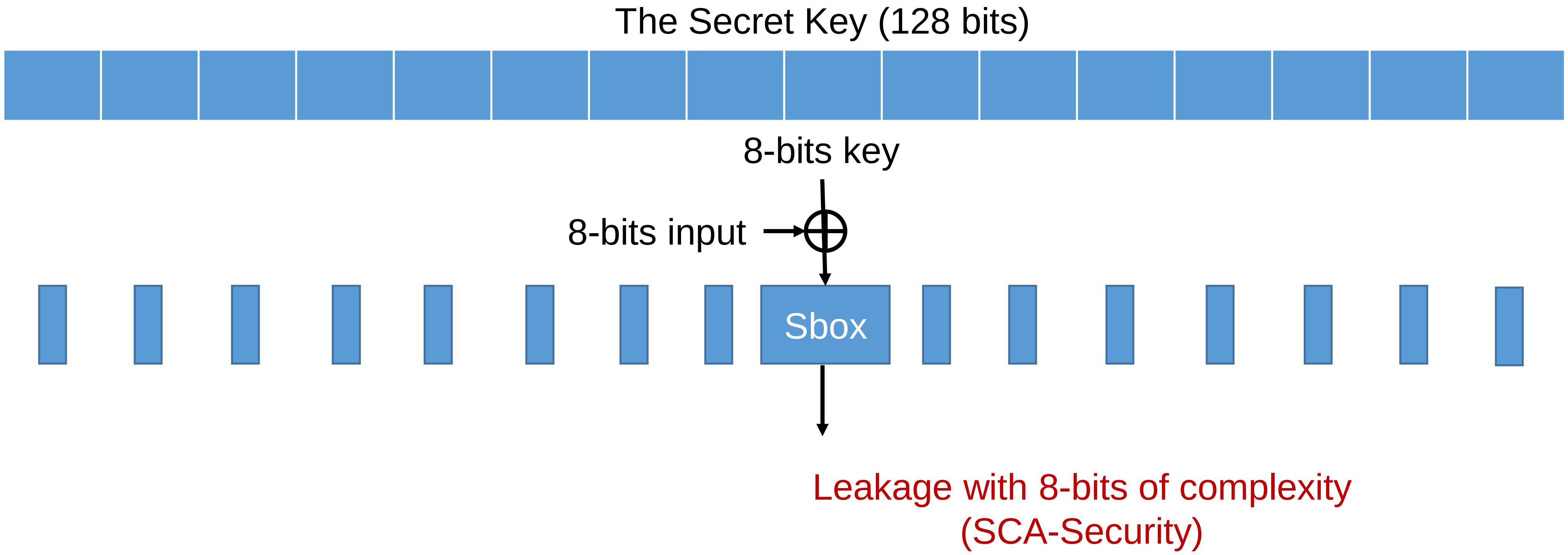


- Use key only one



SCA-Security

- AES



SCA-Security

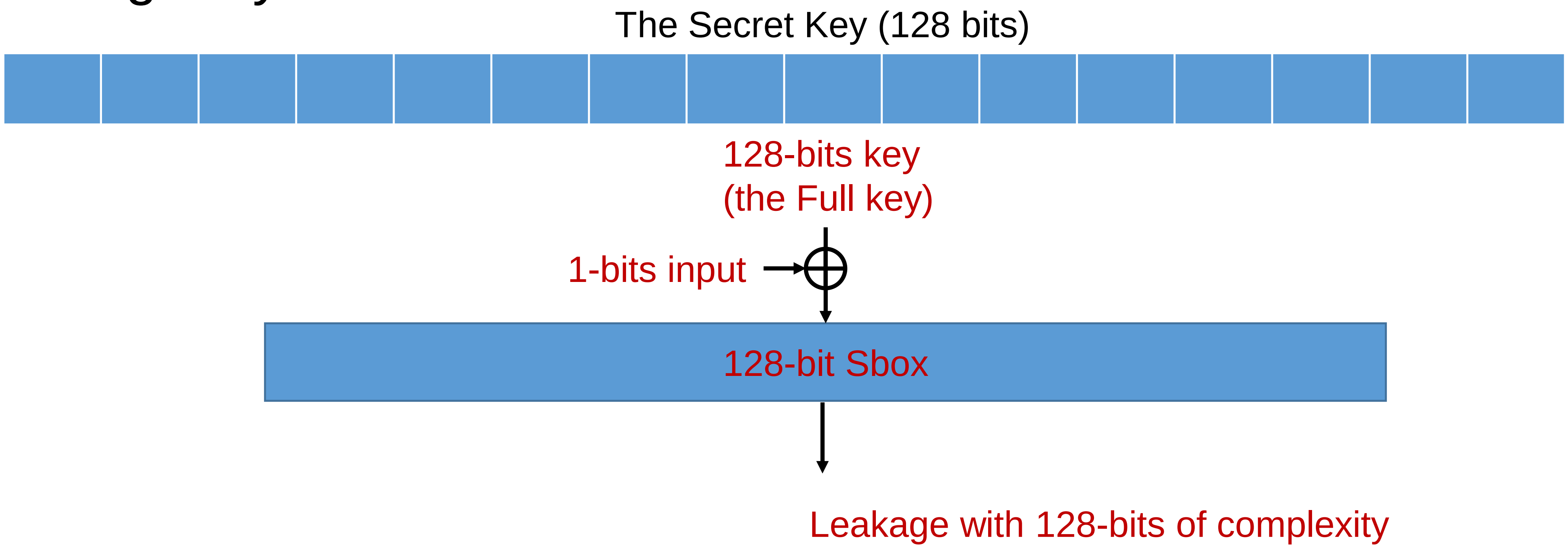
- SCA-Security

SCA-security is the minimum size of key hypothesis (in bits) such that the leakage-model using the correct key correlates to the measured leakage significantly higher than the leakage-model using any other key.

- In other words, the minimum size of key hypothesis that is required to accurately estimate the leakage.
  - SCA-security of AES → 8 bits
  - SCA-security of Present → 4 bits
  - SCA-security of square-and-multiply RSA → 1 bits

How Our Protection Works

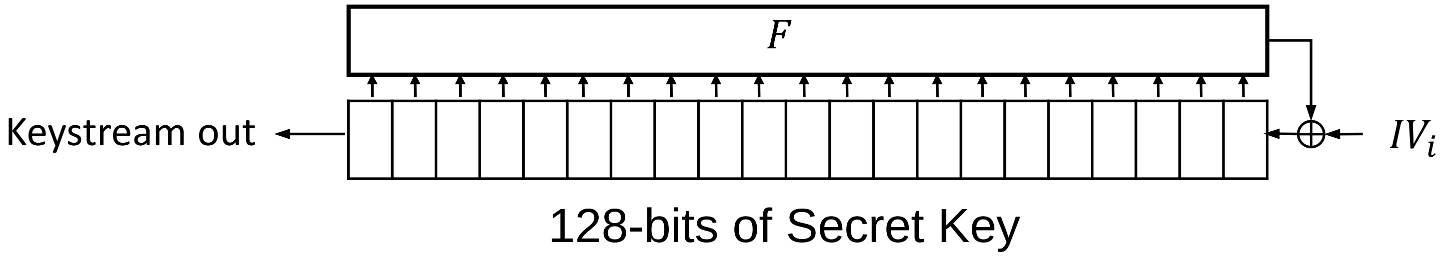
- Imaginary SBox



The goal is to find an implementable non-linear function
that takes 128 bits of key and one bit of input

Non-Linear Feedback Shift Register

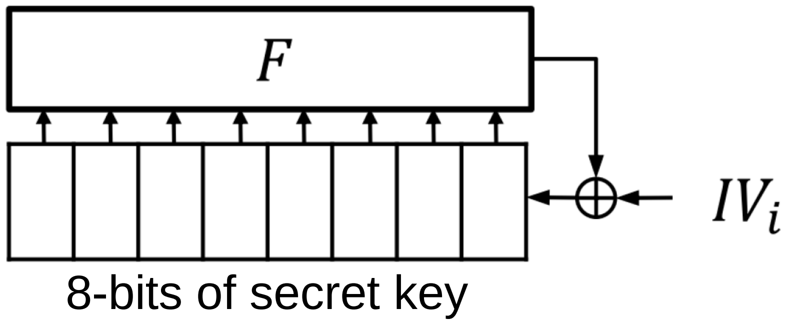
- 128-bit register, with 128 taps.



- SCA-security = 128 bits ☺. Awesome, but
  - There is no structure like this in the literature. ☹
  - Very difficult to be implemented. ☹

Toy Model I:

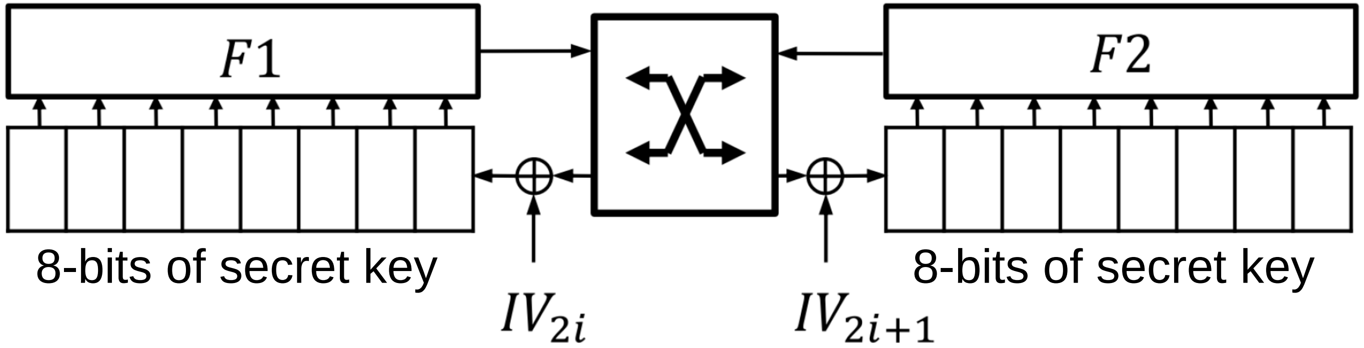
- 8-bit register, with 8 taps.



- SCA-security = 8 bits.
- Can only be broken on clock cycle number 8.

Toy Model II:

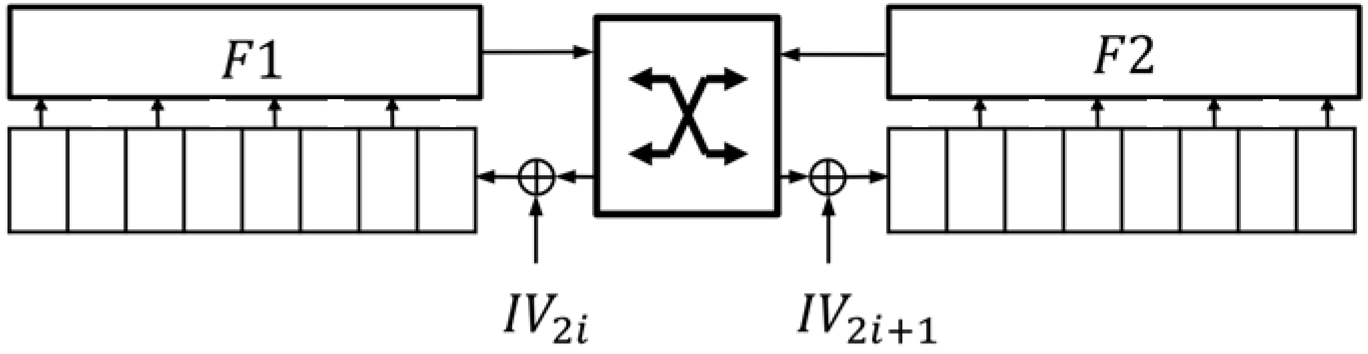
- Two 8-bit, 8-tap NLFSRs with Rotating Cross-Connect.



- SCA-security = 16 bits.
  - The other R is feeding data-dependent noise, and cannot be isolated.
  - The first structure ever to combine two non-linear functions while being immune against the divide-and-conquer principle of SCA. ☺
  - Similar number of taps is still a limitation ☹

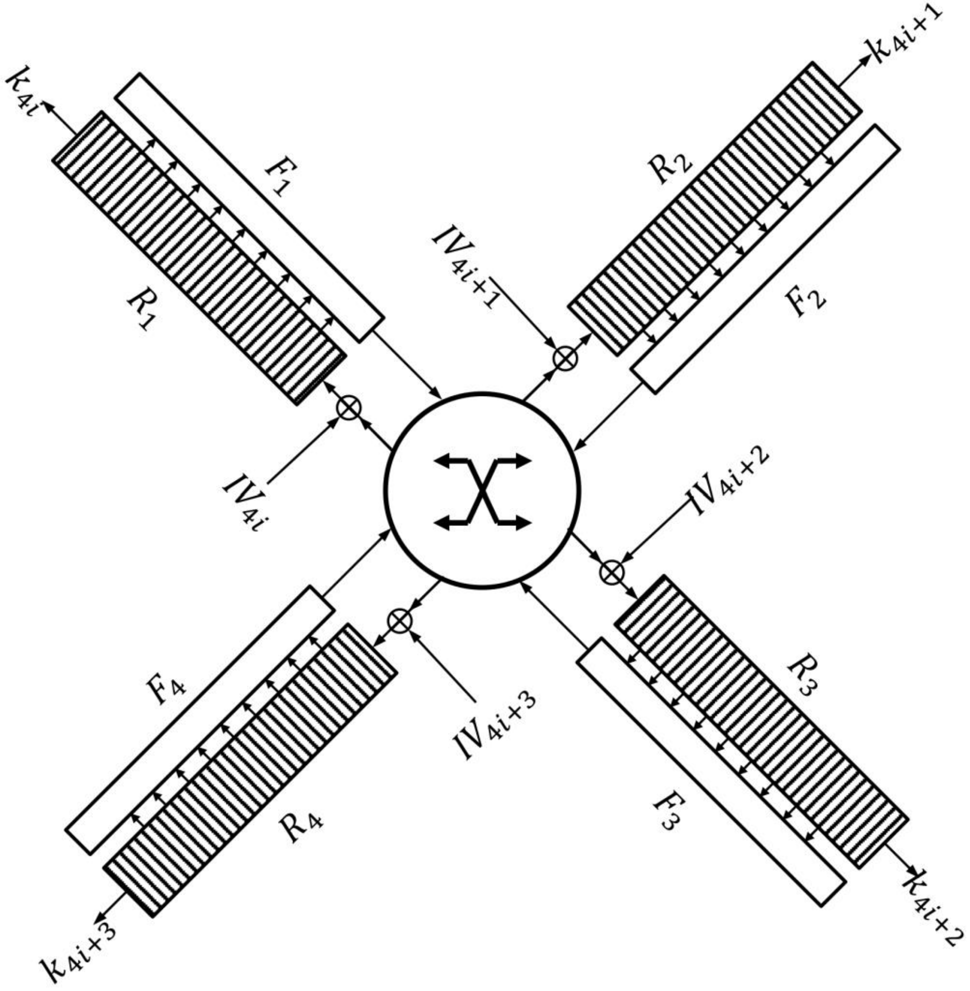
Toy Model III:

- Two 8-bit registers with 4-bit feedback function



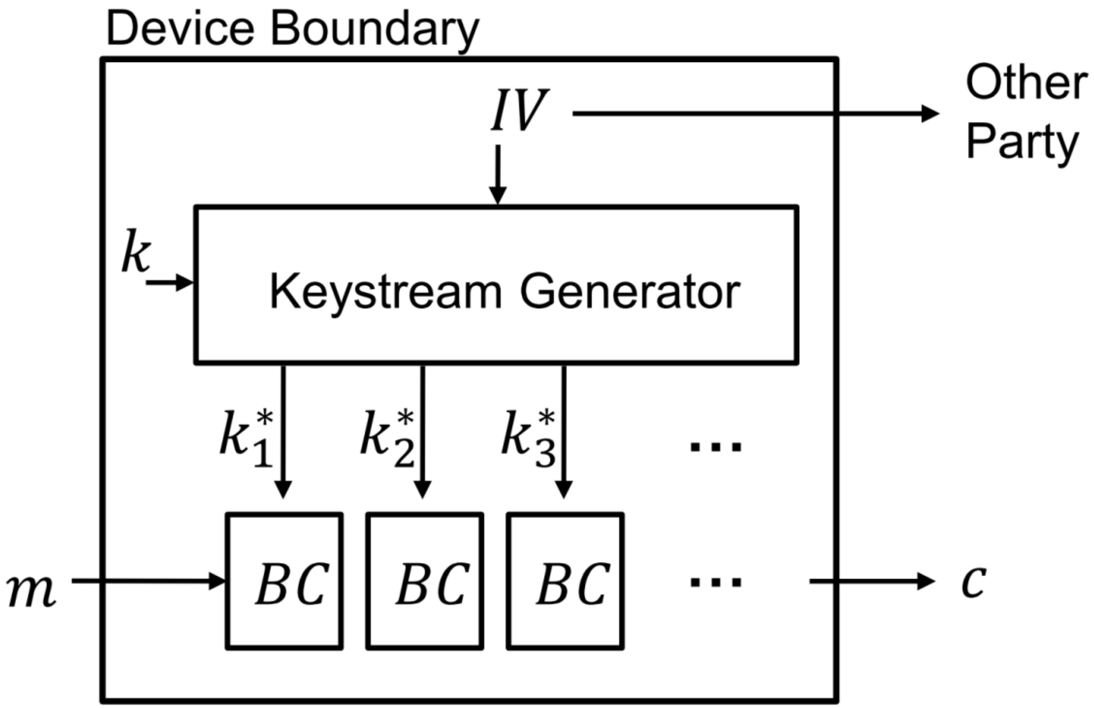
- If the taps are distributed over the odd (or even) bits, then SCA-security = 16 bits.
  - The other secret bits are feeding data-dependent noise, and cannot be isolated.

Keymill


k_{4i}
k_{4i+1}
k_{4i+2}
k_{4i+3}
F_1
F_2
F_3
F_4
R_1
R_2
R_3
R_4
IV_{4i}
IV_{4i+1}
IV_{4i+2}
IV_{4i+3}

Keymill

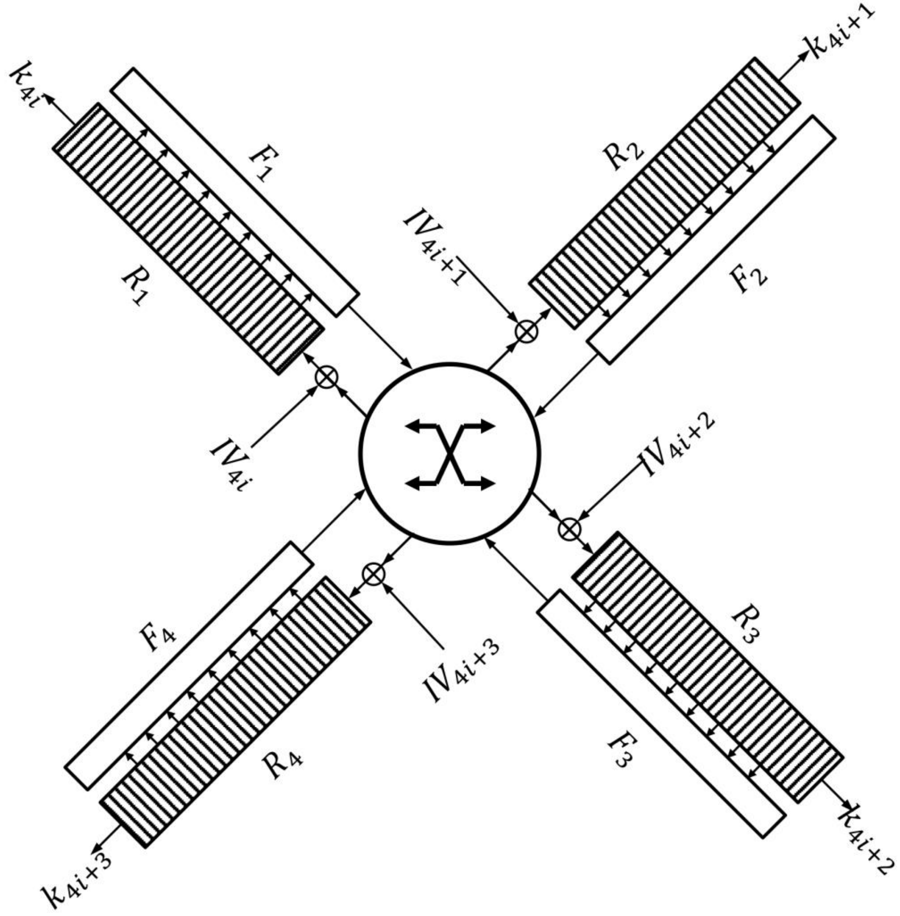
- Keystream generator



- Why? Less emphasis on cryptographic properties, and focus on SCA-properties. Keymill as a stream cipher is coming …

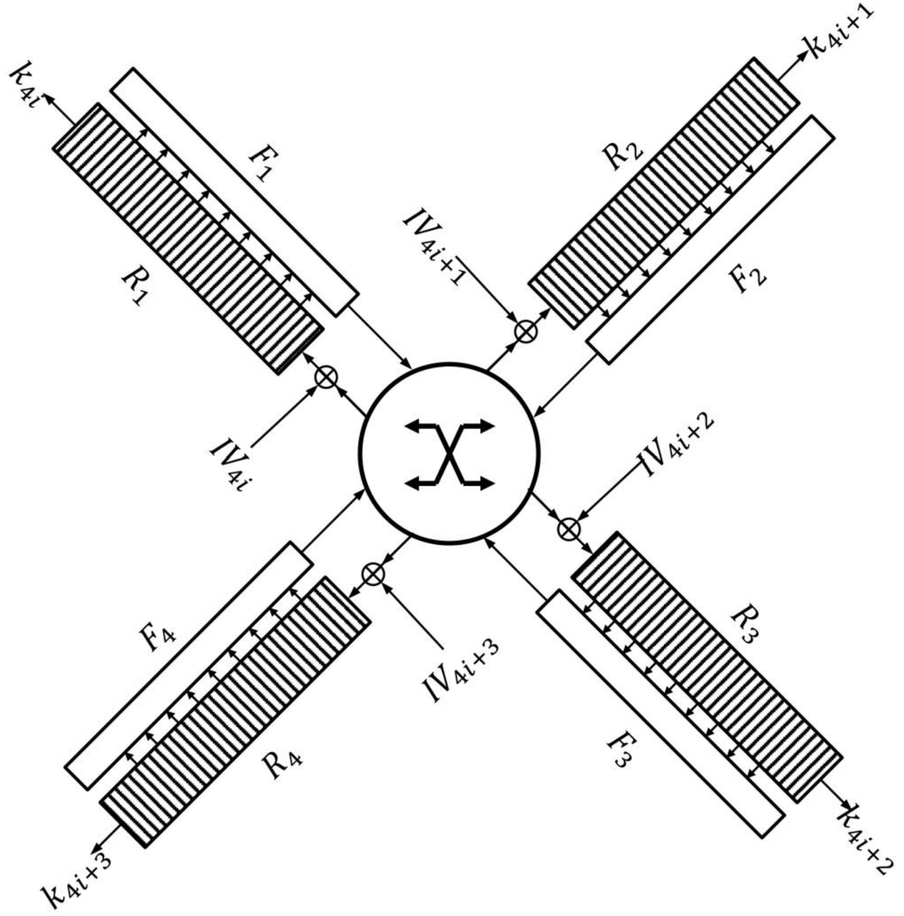
Keymill

- SCA-security = 128 bits.
- 4 Registers
  - 31, 32, 32 and 33 bits, with 17, 17, 17 and 18 taps.
- Cumulative length is 128-bits to maintain entropy of the secret key.
- The taps are distributed nicely over the registers to keep SCA-security.
- Selected from the Achterbahn stream cipher.



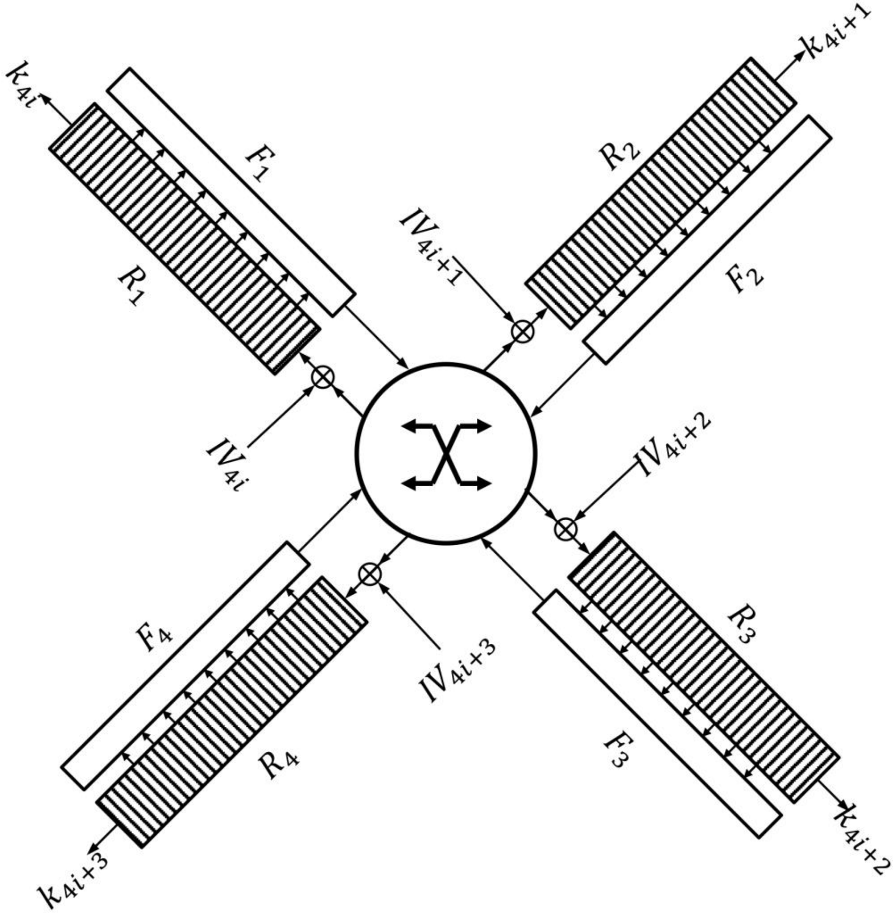
Keymill

- The key is loaded into the registers.
- The structure accepts Initialization Vector (IV) of any length, 1-bit per clock for each register.
- Runs for 33 clock cycles without output (warm-up).
- Then, generates keystream bits.



Keymill

- Cautionary Notes:
  - Not a stream cipher yet!
    only a keystream generator.
  - The two sides of communication need to apply the same countermeasure.
  - Entropy of the keystream depends on entropy of the input key.



Validation

- Hiding

  is validated by comparing the success rate of a practical attack.

- Masking

  is validated by the ability to distinguish leakage of a fixed input versus random inputs.

- Leakage Resiliency

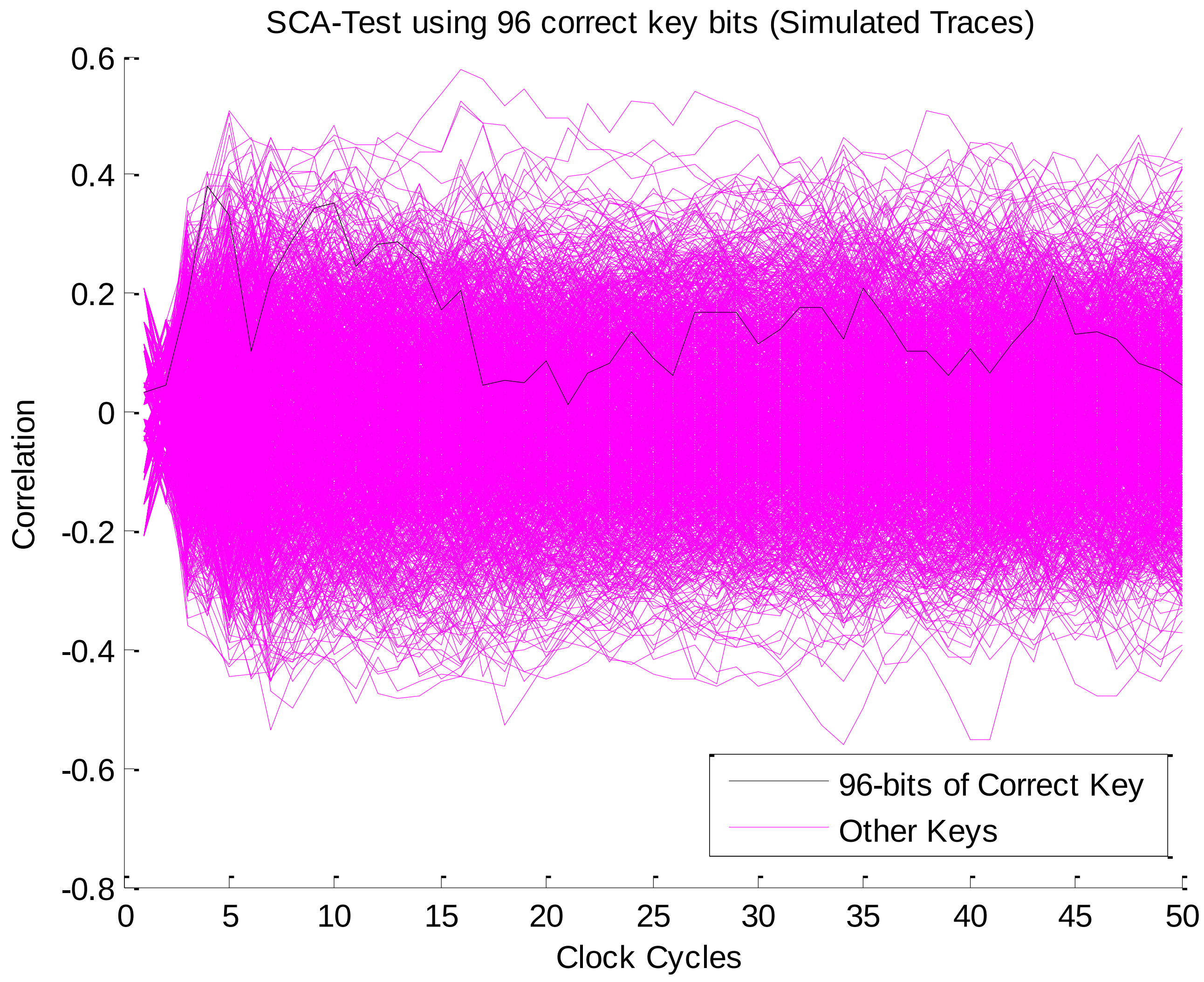
  is validated by mathematical proofs.

- Keymill

  is validated by testing the SCA-Security.

Validation

- New SCA-Security-Test:
  1. Choose a random key.
  2. Collect power traces.
  3. Generate modeled traces using 96-bits of the key. Find the correlation
  4. Generate modeled traces at random wrong keys. Find the correlations.
  5. Compare.



Implementation

Contribution	Area (GE)	Clock cycles
Modular Mul of [12]	7,300	562
Minimum SP network of [2]	5,302	61
The proposed `Keymill`	775	97

- Only 775 GEs
- 97 Clock cycles (including 33 clocks for warm-up)

12. M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In *Progress in Cryptology, AFRICACRYPT 2010*, page 279:296. Springer, 2010.

2. S. Belaid, F. D. Santis, J. Heyszl, S. Mangard, M. Medwed, J.-M. Schmidt, F.-X. Standaert, and S. Tillich. Towards fresh re-keying with leakage-resilient PRFs: Cipher design principles and analysis. Cryptology ePrint Archive, Report 2013/305, 2013.

Conclusion

- With Keymill, we promote to
  - Measure security against SCA attacks in terms of bits, rather than success rate.
  - Design new cryptographic schemes with inherent SCA-security, with less dependence on the underlying implementation.

  - Design new schemes with
    SCA-security = Mathematical-security.

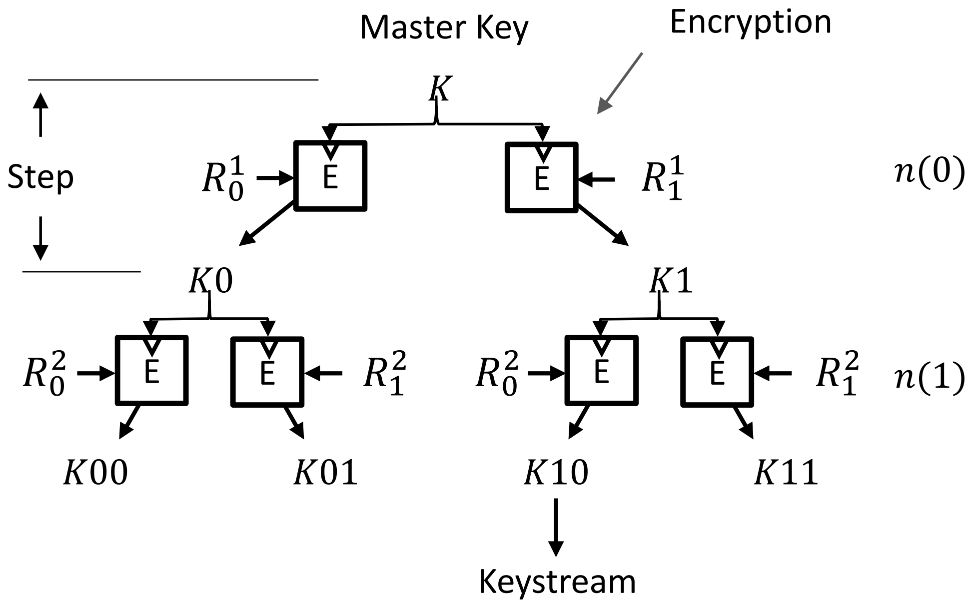
  - Currently, we propose Keymill as a keystream generator, rather than an actual stream cipher (future goal).

Thank You

mtaha9@uwo.ca

Keymill Vs GGM

- GGM is a tree structure that was proposed to realize PRFs from PRGs.
- It was re-introduced to initialize leakage resilient primitives.
- Each step accepts 1-bit of IV, followed by full randomization.
- GGM is an algorithmic countermeasure using a *leaky* function.



[SPY+10]