Side-Channel Analysis of Keymill

Christoph Dobraunig, Maria Eichlseder, Thomas Korak, Florian Mendel

COSADE 2017

Part I

Introduction

Countermeasures

Power consumption independent of intermediate variables

- Hiding
- Masking

Limit usage of secret key

- Fresh re-keying
- (Sometimes) leakage resilient cryptography

Keymill

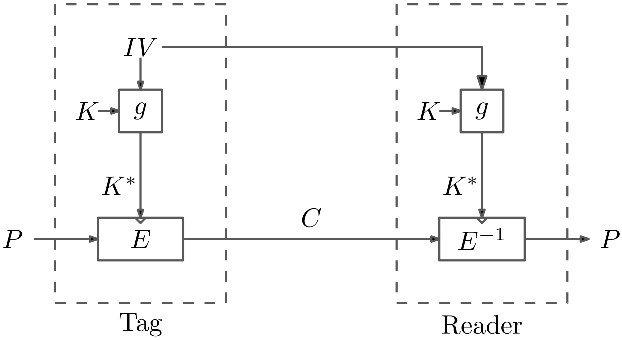
Side-channel resilient key generator [TRS16] (SAC 2016)

Inspired by fresh re-keying [Med+10]

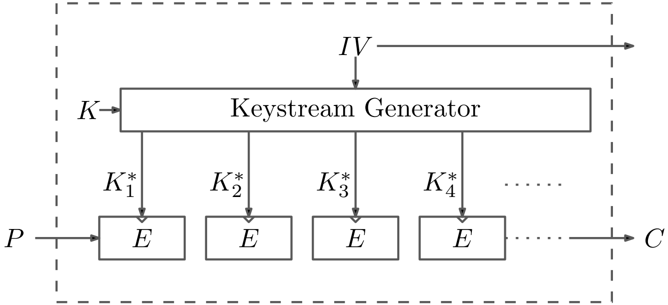
"...secure against SCA attacks inherently by design without requiring any redundant circuit" [TRS16]

We show a side-channel attack

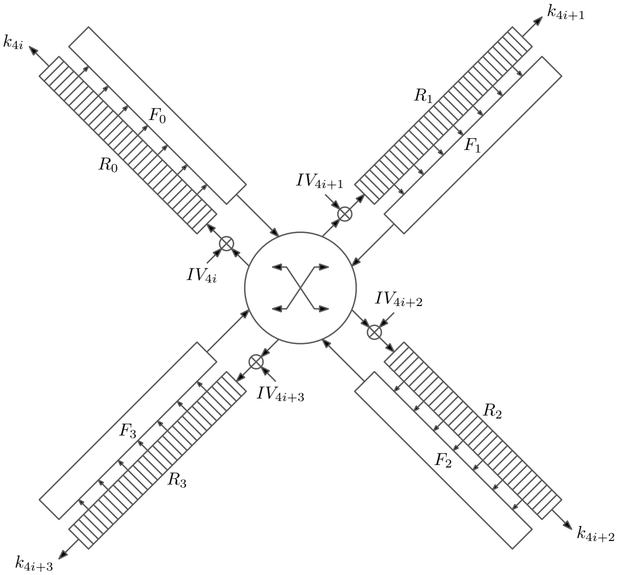
Fresh Re-keying [Med+10]



Keymill [TRS16]



Keymill [TRS16]



Part II

Side-Channel Attack

Attack Rationale

“Essentially, the adversary cannot make an accurate estimate about the data-dependent power changes in the structure unless he makes a correct hypothesis over the entire secret key.” [TRS16]

Idea: recover internal differences instead of values

Require that IV can be controlled

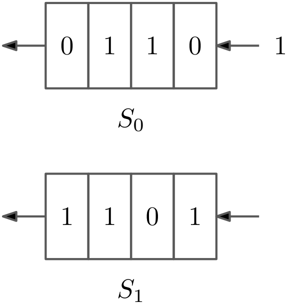
Power Consumption of a Shift-Register [ZH14]

Assume shift register build out of D-flip-flops

For D-flip-flops power consumption higher if state changes

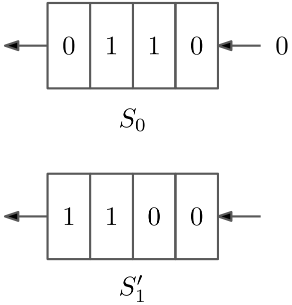
Hence power consumption depends on HD of two states

Example: Power Consumption of a Shift-Register



3 registers change state

Example: Power Consumption of a Shift-Register



2 registers change state

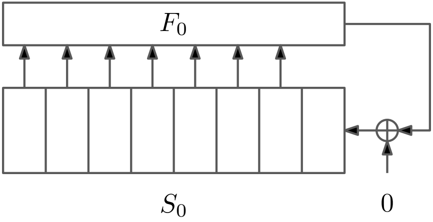
Power Consumption of a Shift-Register

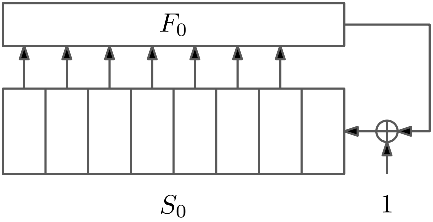
Power consumption reaching state S_1 higher than S_1'

- First two bits in S_1' equal
- First two bits in S_1 different

Learn information even if S_0 unknown

Feedback Shift Register





Feedback Shift Register

In both cases same input to F_0

So we expect a similar power consumption

Only difference in first bit of shift register

Learn in which case the first two bits are different

State Recovery for FSRs

Idea: recover internal differences instead of values

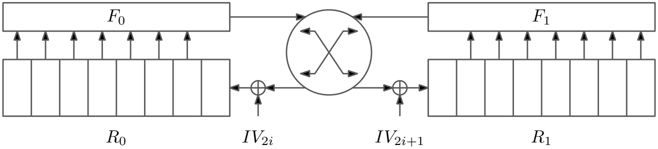
Record power trace for IV 0000000, 1000000, 0100000, ...

Compare power consumption of zero IV with others when 1 is absorbed

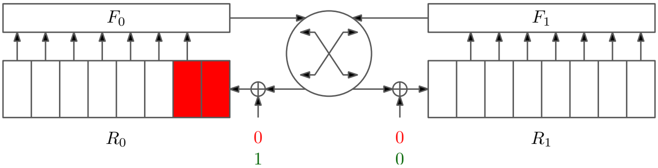
Learn internal difference of all zero IV at each position

Guess of 1 bit reveals all other bits

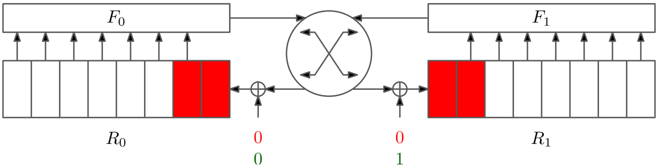
Toy Model II



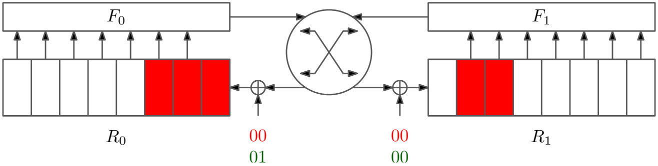
Toy Model II



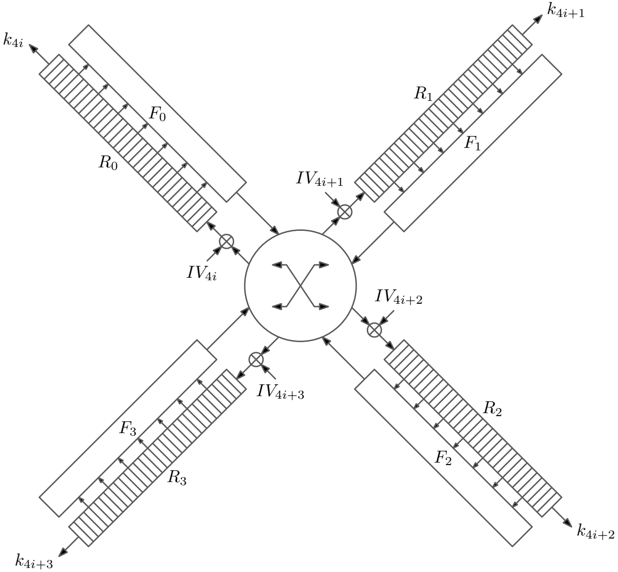
Toy Model II



Toy Model II



Keymill


k_{4i}
k_{4i+1}
F_0
R_0
R_1
F_1
IV_{4i+1}
IV_{4i}
IV_{4i+2}
IV_{4i+3}
R_2
F_3
F_2
R_3
k_{4i+2}
k_{4i+3}

Keymill

Registers have 31 bits, 32 bits, 32 bits and 33 bits

Can recover all internal differences

Results in 2 possible values per register

2^4 values in total

Key can be recovered since feedback functions are nonsingular

Feedback Functions

$$
\begin{aligned}
F_0(S) = {} & s_0 + s_2 + s_5 + s_6 + s_{15} + s_{17} + s_{18} + s_{20} + s_{25} + s_8 s_{18} + s_8 s_{20} \\
& + s_{12} s_{21} + s_{14} s_{19} + s_{17} s_{21} + s_{20} s_{22} + s_4 s_{12} s_{22} + s_4 s_{19} s_{22} \\
& + s_7 s_{20} s_{21} + s_8 s_{18} s_{22} + s_8 s_{20} s_{22} + s_{12} s_{19} s_{22} + s_{20} s_{21} s_{22} \\
& + s_4 s_7 s_{12} s_{21} + s_4 s_7 s_{19} s_{21} + s_4 s_{12} s_{21} s_{22} + s_4 s_{19} s_{21} s_{22} \\
& + s_7 s_8 s_{18} s_{21} + s_7 s_8 s_{20} s_{21} + s_7 s_{12} s_{19} s_{21} + s_8 s_{18} s_{21} s_{22} \\
& + s_8 s_{20} s_{21} s_{22} + s_{12} s_{19} s_{21} s_{22} \\
F_1(S) = F_2(S) = {} & s_0 + s_3 + s_{17} + s_{22} + s_{28} + s_2 s_{13} + s_5 s_{19} + s_7 s_{19} \\
& + s_8 s_{12} + s_8 s_{13} + s_{13} s_{15} + s_2 s_{12} s_{13} + s_7 s_8 s_{12} + s_7 s_8 s_{14} \\
& + s_8 s_{12} s_{13} + s_2 s_7 s_{12} s_{13} + s_2 s_7 s_{13} s_{14} + s_4 s_{11} s_{12} s_{24} \\
& + s_7 s_8 s_{12} s_{13} + s_7 s_8 s_{13} s_{14} + s_4 s_7 s_{11} s_{12} s_{24} + s_4 s_7 s_{11} s_{14} s_{24} \\
F_3(S) = {} & s_0 + s_2 + s_7 + s_9 + s_{10} + s_{15} + s_{23} + s_{25} + s_{30} + s_8 s_{15} + s_{12} s_{16} \\
& + s_{13} s_{15} + s_{13} s_{25} + s_1 s_8 s_{14} + s_1 s_8 s_{18} + s_8 s_{12} s_{16} + s_8 s_{14} s_{18} \\
& + s_8 s_{15} s_{16} + s_8 s_{15} s_{17} + s_{15} s_{17} s_{24} + s_1 s_8 s_{14} s_{17} + s_1 s_8 s_{17} s_{18} \\
& + s_1 s_{14} s_{17} s_{24} + s_1 s_{17} s_{18} s_{24} + s_8 s_{12} s_{16} s_{17} + s_8 s_{14} s_{17} s_{18} \\
& + s_8 s_{15} s_{16} s_{17} + s_{12} s_{16} s_{17} s_{24} + s_{14} s_{17} s_{18} s_{24} + s_{15} s_{16} s_{17} s_{24}
\end{aligned}
$$

Part III

Practical Evaluation

The noise

Distinguish power consumption for input change

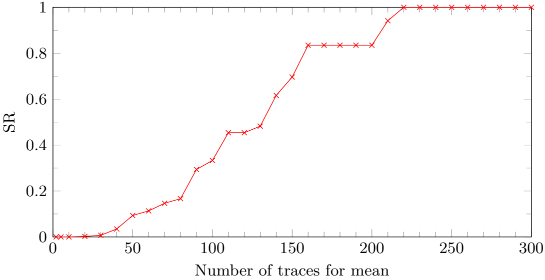
Noise level small enough to distinguish

Average traces and filter the noise

If IV cannot be repeated iterate over last bits of IV

Practical Evaluation

Evaluated FPGA implementation of Keymill



Countermeasures Against the Attack

Use of random IVs

- Recovery of a fraction of the bits still possible

Clock several times between injection of IVs

- Limits number of differences an attacker can learn

Straightforward application of attack not possible

No guarantee that more sophisticated attacks do not work

Part IV

Cryptographic Considerations

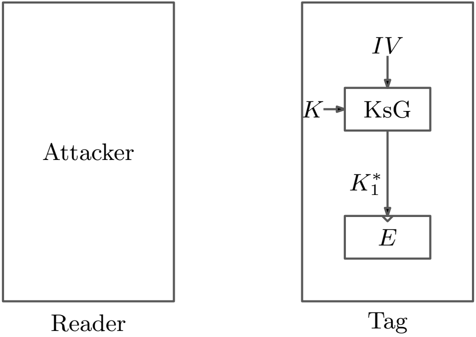
The Small State of Keymill

Internal state 128 bits

Allows for time-memory trade-off attacks

If an attacker can control plaintext

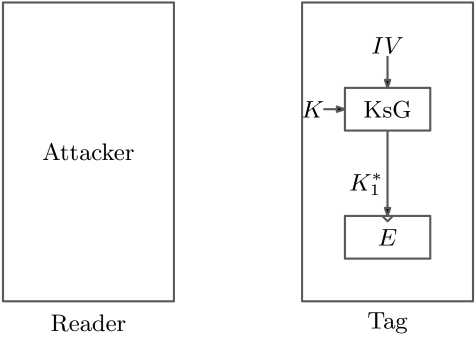
Recovery of Internal State



Recovery of Internal State

Precalculated List

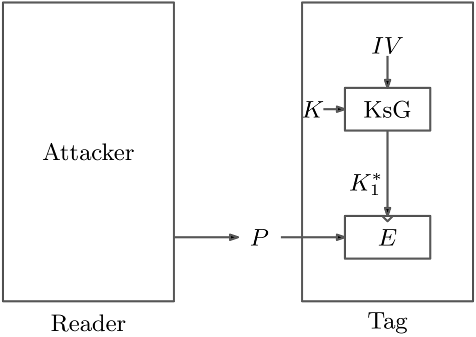
State	C
$ac359f$	$00589f$
$03689c$	$01c341$
887597	$1abd59$
⋮	⋮
$cf8765$	$f1c897$
$fa8633$	$fa9855$
$00c5a9$	$fac359$



Recovery of Internal State

Precalculated List

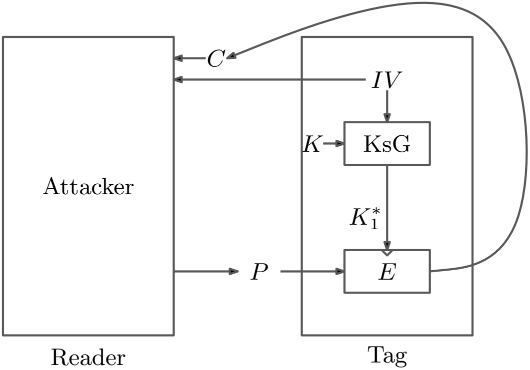
State	C
$ac359f$	$00589f$
$03689c$	$01c341$
887597	$1abd59$
$\vdots$	
$cf8765$	$f1c897$
$fa8633$	$fa9855$
$00c5a9$	$fac359$



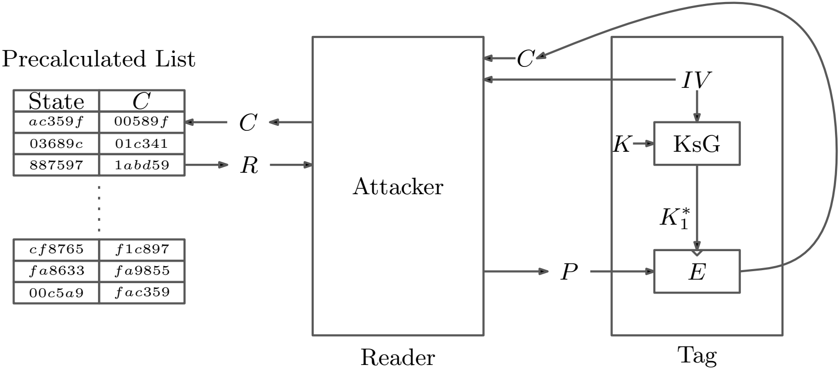
Recovery of Internal State

Precalculated List

State	C
$ac359f$	$00589f$
$03689c$	$01c341$
887597	$1abd59$
⋮	⋮
$cf8765$	$f1c897$
$fa8633$	$fa9855$
$00c5a9$	$fac359$



Recovery of Internal State



Recovery of Internal State

How big is the list?

How many online queries?

Recovery of Internal State

How big is the list?

How many online queries?

list entries	online queries	total complexity
$2^{n/4}$	$2^{3n/4}$	$2^{3n/4}$
$2^{n/3}$	$2^{2n/3}$	$2^{2n/3}$
$\mathbf{2^{n/2}}$	$\mathbf{2^{n/2}}$	$\mathbf{2 \cdot 2^{n/2}}$
$2^{2n/3}$	$2^{n/3}$	$2^{2n/3}$
$2^{3n/4}$	$2^{n/4}$	$2^{3n/4}$

Conclusion

Seems that Keymill is not inherently secure against SCA

Probably larger internal state needed

Can Keymill be tweaked to make it more secure?

Are additional countermeasures always required?

Thank you

References I

[Med+10] M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni
Fresh Re-keying: Security against Side-Channel and Fault Attacks for Low-Cost Devices
AFRICACRYPT 2010

[TRS16] M. Taha, A. Reyhani-Masoleh, and P. Schaumont
Keymill: Side-Channel Resilient Key Generator
SAC 2016

[ZH14] A. A. Zadeh and H. M. Heys
Simple power analysis applied to nonlinear feedback shift registers
IET Information Security 8:3, 2014