Cryptanalysis of the A5/1 Using Power Analysis

Martin Jureček, Jiří Buček and Róbert Lórencz

{jurecmar,bucekj,lorencz}@fit.cvut.cz

Department of Information Security
Faculty of Information Technology
Czech Technical University in Prague

June 25, 2019

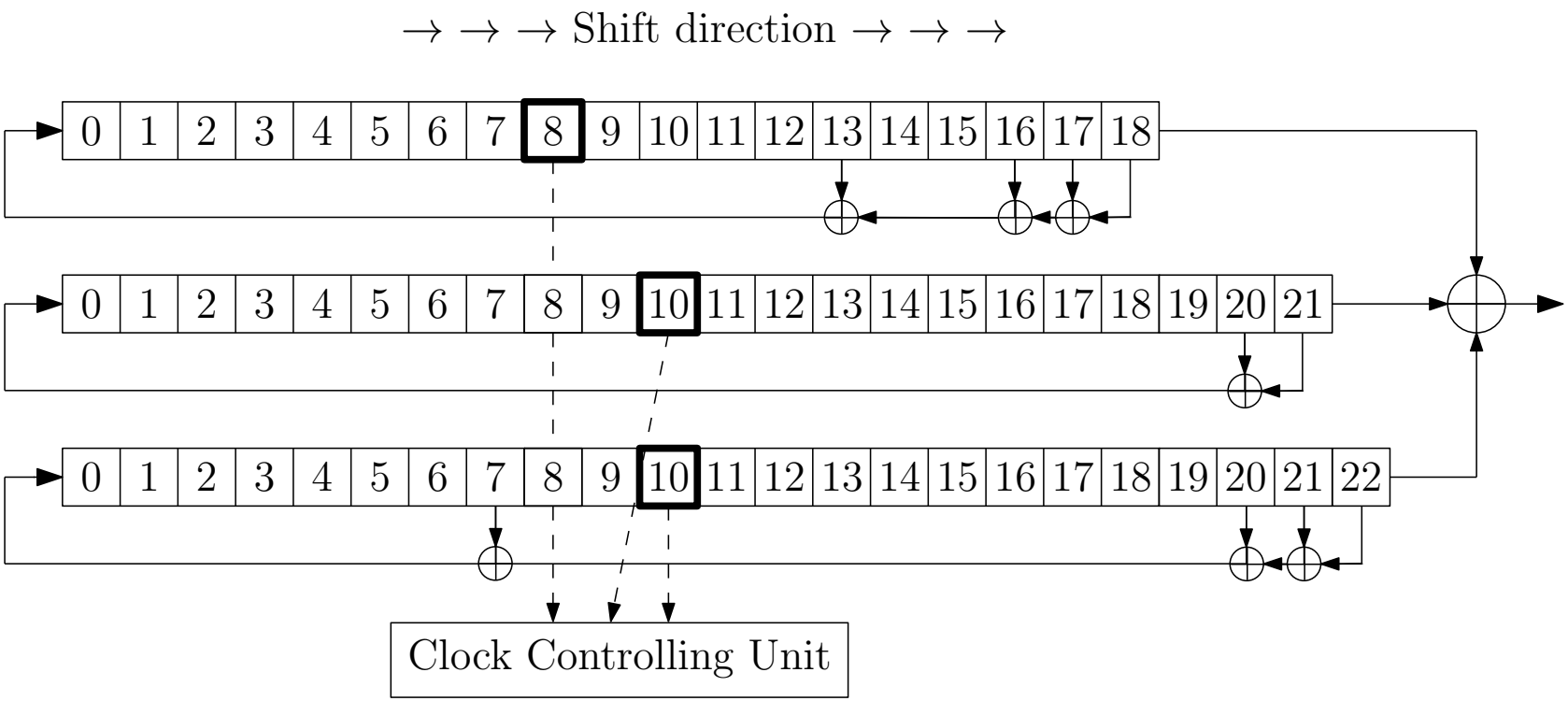
Outline

1. Introduction
   - our contribution
   - description of A5/1
   - previous attacks
2. Power analysis attacks (1st step of the attack)
   - proof-of-concept SPA attack
   - existing DPA attack
3. Recovering the secret key (2nd step of the attack)
   - create and solve a system of equations which describes the information on clocking bits
   - attack complexity derivation
   - example of key recovery

Contribution

- proof-of-concept SPA attack
- recovering the secret key based on results of SPA or DPA
- **no bit of a keystream is required**
- assumption of PA attacks: power consumption while clocking 3 LFSRs is different than when clocking 2 LFSRs
- good scalability for parallel processing in key recovery

Description of A5/1



A5/1 algorithm

Algorithm 1 A5/1 - generating keystream for one frame

1: all bits of the registers are set to zero
2: **for** $i = 1$ to 64 **do**
3: $R_j[0] := R_j[0] \oplus K_i, j = 1, 2, 3$ and clock all registers
4: **end for**
5: **for** $i = 1$ to 64 **do**
6: $R_j[0] := R_j[0] \oplus f_i, j = 1, 2, 3$ and clock all registers
7: **end for**
8: **for** $t = 1$ to 100 **do**
9: *clock the cipher by majority function and discard output bits*
10: **end for**
11: **for** $t = 101$ to 328 **do**
12: clock the cipher by majority function and produce the 228 bits of keystream
13: **end for**

Previous attacks: Guess-and-determine, TMTO

- Guess-and-determine attacks introduced in [1, 2]
  - Guessing part of the internal state and determine the remaining bits, 64 bits of keystream needed [1]
- Time-Memory Tradeoff attacks presented in [1, 3]
  - Precomputation phase and an attack phase

[1] Golić, J. D. Cryptanalysis of alleged A5 stream cipher. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 239-255, Springer, Berlin, Heidelberg, 1997.

[2] Biham, E., Dunkelman, O. Cryptanalysis of the A5/1 GSM stream cipher. In *International Conference on Cryptology in India*, pp. 43-51, Springer, Berlin, Heidelberg, 2000.

[3] Biryukov, A., Shamir, A., Wagner, D. Real Time Cryptanalysis of A5/1 on a PC. In *International Workshop on Fast Software Encryption*, pp. 1-18, Springer, Berlin, Heidelberg, 2000.

Previous attacks: Correlation, HW based

- Attacks based on correlation introduced in [4, 5]
  - Almost independent of the LFSR length
- Hardware based attacks presented in [6, 7]
  - Special architectures, parallel processing in FPGAs

[4] Ekdahl, P., Johansson, T. Another attack on A5/1. *IEEE transactions on information theory*, vol. 49, no. 1, pp. 284-289, 2003.

[5] Maximov, A., Johansson, T., Babbage, S. An improved correlation attack on A5/1. In *International Workshop on Selected Areas in Cryptography*, pp. 1-18, Springer, Berlin, Heidelberg, 2004.

[6] Pornin, T., Stern, J. Software-hardware trade-offs: Application to A5/1 cryptanalysis. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 318-327, Springer, Berlin, Heidelberg, 2000.

[7] Gendrullis, T., Novotný, M., Rupp, A. A real-world attack breaking A5/1 within hours. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 266-282, Springer, Berlin, Heidelberg, 2008.

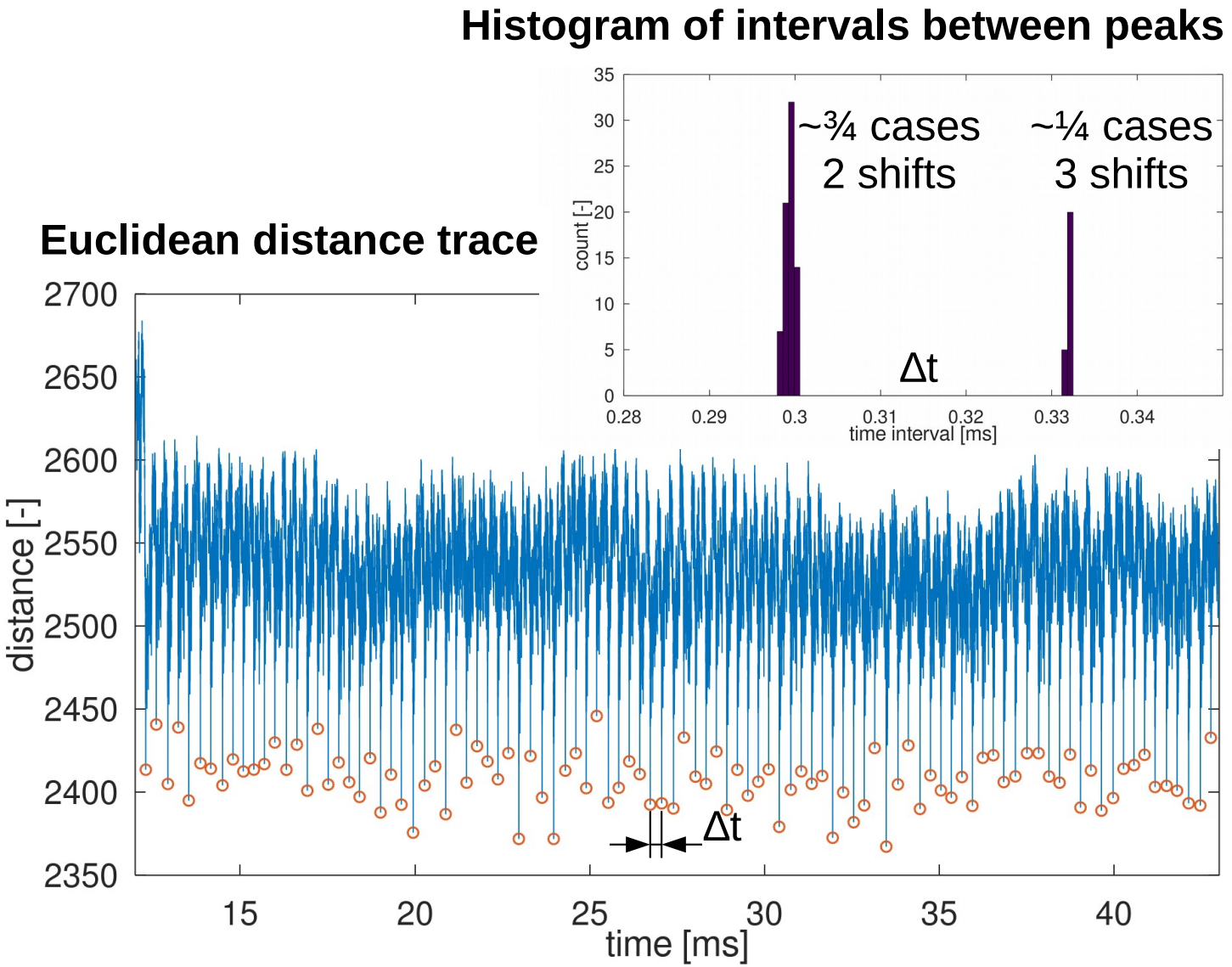
SPA demonstration

- Goal: proof-of-concept attack
- Simple 8-bit MCU based prototype
- Not realistic scenario, for demonstration only
- Measure power traces
- Extract the number of LFSRs shifted in each of 100 clocks

SPA trace analysis

- Select a pattern – part of power trace
  - Sequence of instructions that occurs in each of 100 clocks
  - Sequential implementation
  - Time interval between patterns $\rightarrow$ how many LFSRs shifted
- Find the pattern using Euclidean distance metric
  - Square root of sum of squared differences
  - Minimum distance $\rightarrow$ pattern match

Euclidean distance



Existing DPA attack on A5/1

- Introduced in [8]
- For each triplet of bits (k_1, k_2, k_3) and for all traces compute clocking bits:
  $c_j^{(i)} := k_j \oplus f_j^{(i)}, \text{ for } j = 1, 2, 3, \text{ and } i = 1, \ldots, 1000$
- if $c_1^{(i)} = c_2^{(i)} = c_3^{(i)}$ then append the corresponding power trace to the set S_3, otherwise to the set S_2.
- for each triplet (k_1, k_2, k_3) compute the differential trace $\Delta = mean(S_3) - mean(S_2)$

[8] Lano, J., Mentens, N., Preneel, B., Verbauwhede, I. Power analysis of synchronous stream ciphers with resynchronization mechanism. In *ECRYPT Workshop, SASC-The State of the Art of Stream Ciphers*, pp. 327-333, 2004.

Recovering the secret key based on SPA results - main idea

- output from SPA is a sequence s of numbers of LFSRs shifted for $t = 1, \ldots, 100$ in the initialization phase
- create a system of equations which describes the information on clocking bits
- perform Gaussian elimination of each new equation relatively to the equations already appended
- recovering the *initial state* from the internal state corresponding to time t is done via reversion [1]
- then we can recover the secret key by solving the system of linear equations in variables of bits of the secret key

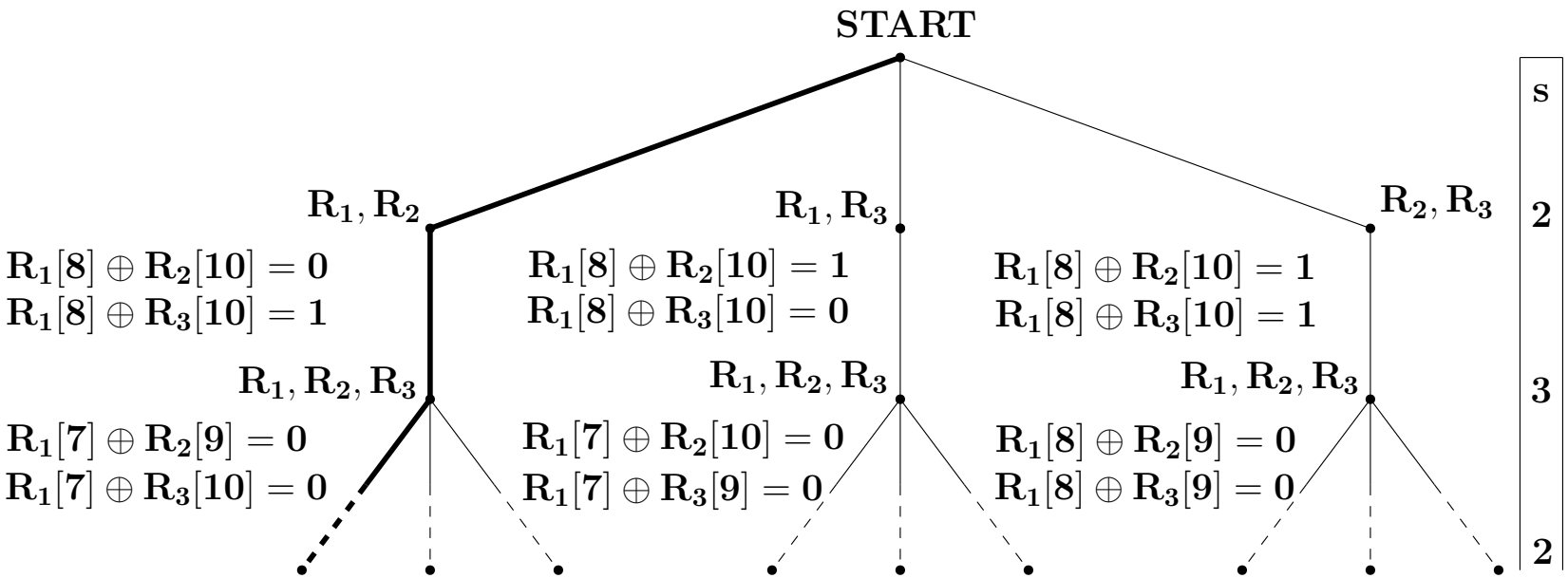
Recovering the internal state based on SPA results

Algorithm 2 Attack on internal state at time t

Input: *clocking sequence* $\{s\}_{i=1}^{100}$
Output: initial state of the three registers

1: **if** s_t LFSRs will be clocked **then**
2: **for all** options for s_t LFSRs **do**
3: **Tree_of_clocking_bits(**s**, option)**
4: **end for**
5: **end if**
6: **return** internal state at time t

Tree of clocking bits



Relation between a shape of the tree and equations I

# shifting registers	equations
2	$R_1, R_2 : \quad c_1 \oplus c_2 = 0,\ c_1 \oplus c_3 = 1$
2	$R_1, R_3 : \quad c_1 \oplus c_2 = 1,\ c_1 \oplus c_3 = 0$
2	$R_2, R_3 : \quad c_1 \oplus c_2 = 1,\ c_1 \oplus c_3 = 1$
3	$R_1, R_2, R_3 : c_1 \oplus c_2 = 0,\ c_1 \oplus c_3 = 0$

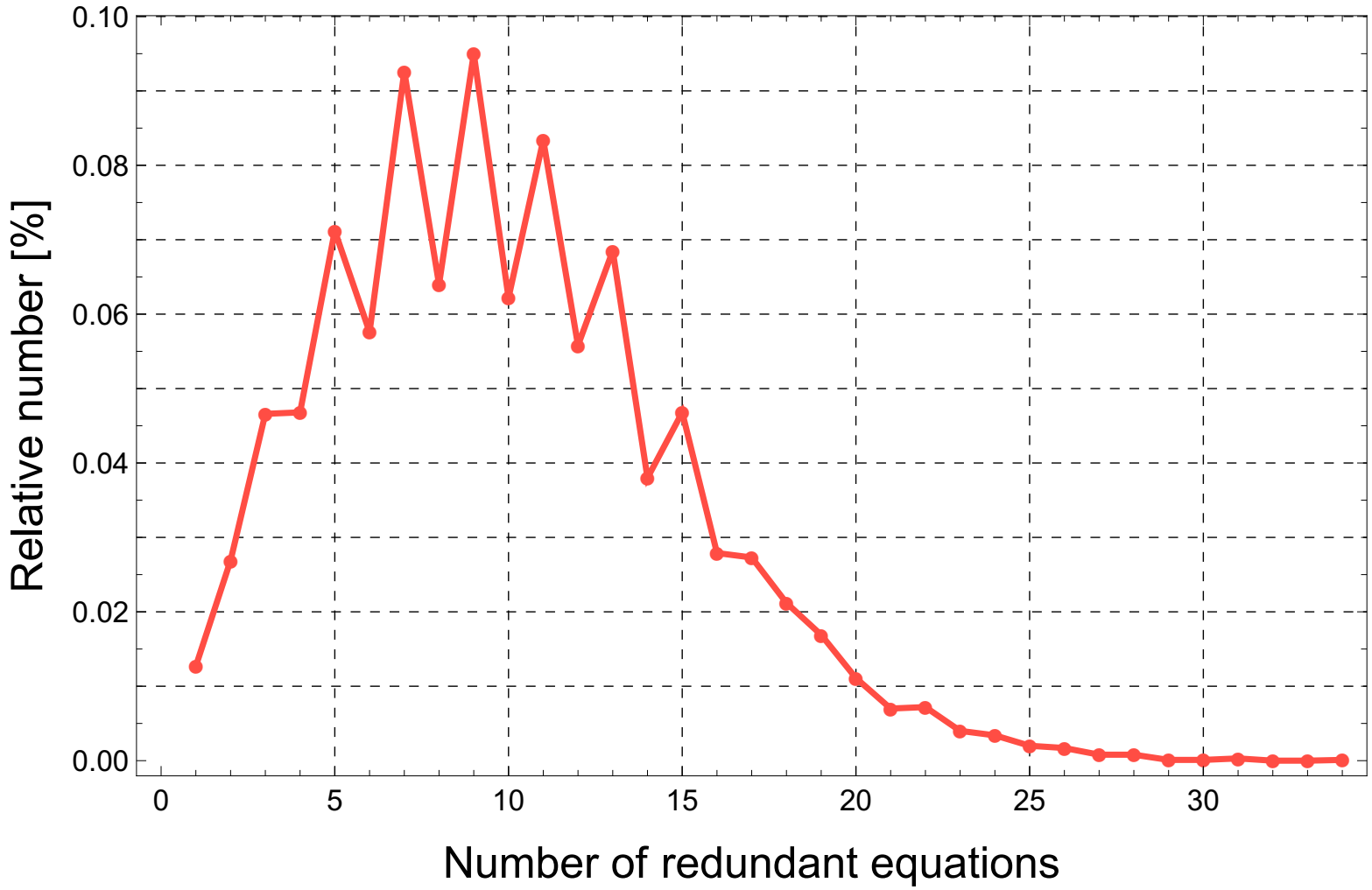
- advantages: smaller tree, less redundant equations
- disadvantages: later detection of irrelevant subtrees

Relation between a shape of the tree and equations II

# shifting registers	equations
2	$R_1, R_2 : \quad c_1 = c_2 = 0, c_3 = 1$
2	$R_1, R_2 : \quad c_1 = c_2 = 1, c_3 = 0$
2	$R_1, R_3 : \quad c_1 = c_3 = 0, c_2 = 1$
2	$R_1, R_3 : \quad c_1 = c_3 = 1, c_2 = 0$
2	$R_2, R_3 : \quad c_2 = c_3 = 0, c_1 = 1$
2	$R_2, R_3 : \quad c_2 = c_3 = 1, c_1 = 0$
3	$R_1, R_2, R_3 : c_1 = c_2 = c_3 = 1$
3	$R_1, R_2, R_3 : c_1 = c_2 = c_3 = 0$

- advantages: earlier detection of irrelevant subtrees
- disadvantages: larger tree, more redundant equations

Redundant equations expressed as relative frequencies



Complexity of the attack based on results of SPA

- 72.92 equations on average are needed to obtain the regular system with dimension 64
- the length of clocking sequence s should be at least 36.46
- if there are k clocking 3 LFSRs, then the average time complexity of the attack is $1/2 \cdot 3^{(36.46-k)} \approx 2^{42.34}$ for $k = 36.46/4 = 9.115$
- if the *clocking sequence* were determined from the whole initialization phase (i.e. length is 100 bits) then there are 64.54 subsequences of length 36.46
- we can find the subsequence having approximately 14.84 clocking 3 LFSRs and the average time complexity would decrease to $1/2 \cdot 3^{36.46-14.84} \approx 2^{33.27}$

Example of key recovery

Initial inner state of the A5/1:

R_1 0 1 0 0 1 0 1 0 **0** 1 0 0 0 1 0 0 0 0 0

R_2 1 1 0 0 1 0 0 1 1 0 **1** 0 1 0 1 1 0 1 0 0 0 0

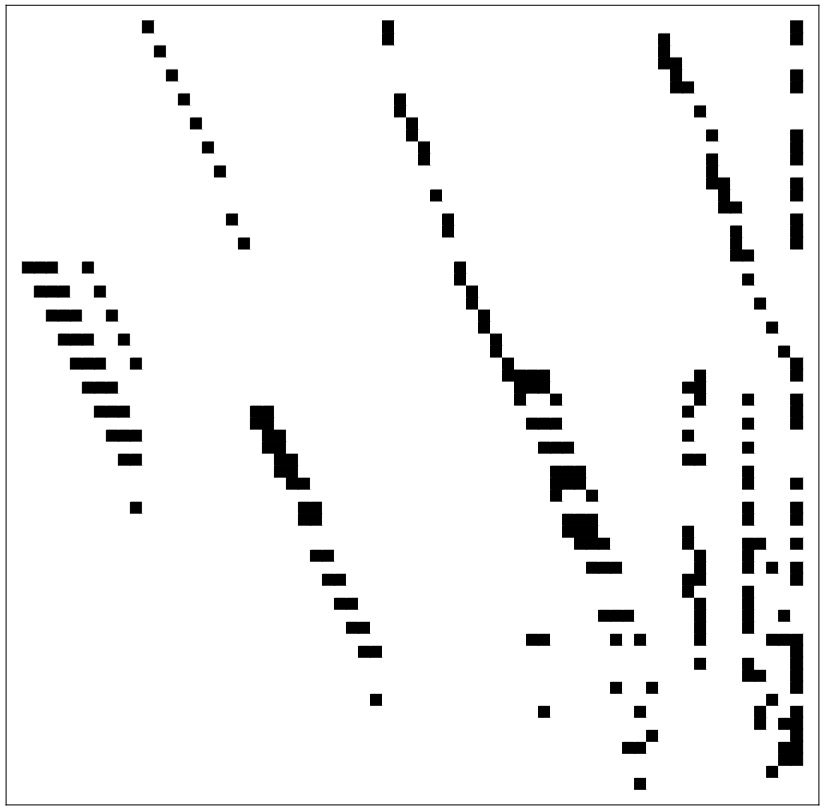
R_3 1 0 0 1 1 1 0 0 1 0 **0** 1 0 0 0 0 0 1 0 0 1 1 0

The sequence of number of shifted registers for $t = 1, ..., 34$:
2 2 3 3 2 2 2 2 2 2 3 3 3 3 2 2 3 3 2 2 2 3 2 2 3 2 2 3 3 3 3 3 2 ...
All 3 LFSRs were shifted at once 16 times and number of redundant equations were 3.

Matrix columns: $R_1[18], \ldots, R_1[0], R_2[21], \ldots, R_2[0], R_3[22], \ldots, R_3[0], b$
Chosen lines of the matrix (just indices of ones):

line	ones	line	ones
1.	10, 30, 64	4.	53, 54
2.	30, 53, 64	21.	0, 1, 2, 5, 36
3.	11, 53	64.	51

Example: Augmented matrix of the linear system



First 3 rows:

$$\left(\begin{array}{ccccccccccccccccc|c}
\overbrace{0 & \cdots & 0}^{0} & \overbrace{1}^{10} & \overbrace{0}^{11} & 0 & \cdots & 0 & \overbrace{1}^{30} & 0 & \cdots & 0 & \overbrace{0}^{53} & 0 & \cdots & \overbrace{0}^{64} & 1 \\
0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 \\
0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0
\end{array}\right)$$

Properties of the SPA based attack

- **100% success rate**
- **no keystream is required**
- minimal storage (a boolean matrix of size 64x65 – a few MBs)
- complexity of SPA based attack is less then $2^{33.27}$
- implementation:
  - JAVA 11, single thread
  - Ubuntu 18 LTS operating system
  - 1.80GHz Intel Core i7-8550U CPU with 16 GB of RAM

Conclusion and future work

- SPA based attack can be run in parallel and executed within seconds
- execution of DPA based attack is even much more faster
- how to decrease execution time of the attack:
  - using parallel computation
  - use more frames and select a frame having subsequences containing the highest number of clocking 3 LFSRs
- we plan to perform experiments to confirm a practical application of the DPA

Thank you for your attention!

The authors acknowledge the support of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

RESEARCH
CENTER FOR
INFORMATICS
rci.cvut.cz