Improving and Evaluating Differential Fault Analysis on LED with Algebraic Techniques

Xinjie Zhao, Shize Guo, Fan Zhang, Zhijie Shi, Chujiao Ma and Tao Wang

Ordnance Engineering College, Shijiazhuang China
The Institute of North Electronic Equipment, Beijing, China
University of Connecticut, Storrs, USA

Outline

- **Why?** Algebraic Fault Analysis
- **How?** ADFA on LED
- **Current?** Our results and comparison
- **Future?** Conclusion and open problem

Background

- FA first proposed by Boneh et al in 1996.
  - Received faulty output, guess the fault, find the secret.
- DFA proposed by Biham and Shamir in 1997.
  - Used to break public-key ciphers (ECC), block ciphers (AES, ARIA, Camellia and CLEFIA) and stream ciphers (RC4, Trivium).
- AFA proposed by Courtois in 2010.
  - Used to break DES, Piccolo and Trivium.

Why AFA?

- Algebraic Fault Attack
  - Algebraic analysis are generic and automatic
  - Solvers (automatic) allow easier and simpler analysis
  - Fault information allows optimization

- Assumptions
  - Possible to inject fault
  - Knows the location of the fault, but not the value
  - Known ciphertext

Overview of LED

- *A lightweight block cipher introduced in CHES 2011. Consists of LED-64 and LED-128.*
  - ***64-bit block cipher, AES-like design***
  - ***SPN structure***
  - ***64 bit keys (LED-64)***
  - ***32 rounds***
  - ***no key scheduling***
  - ***Lightweight, 966 Gate Equivalent (GE)***

LED

- 64-bit block size (4x4 nibbles).
- AFA performed during the SC operation of last couple rounds.

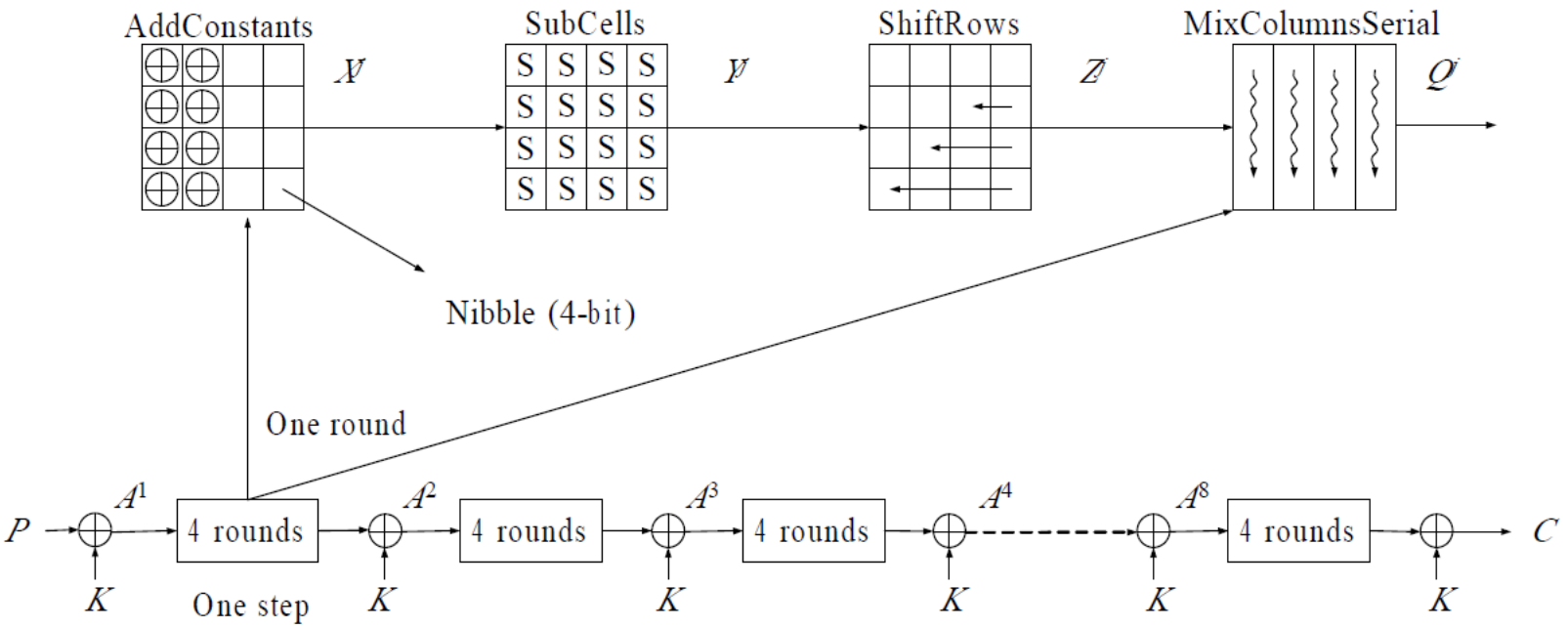


Figure 1. The encryption procedure of LED

Previous attacks

• LED

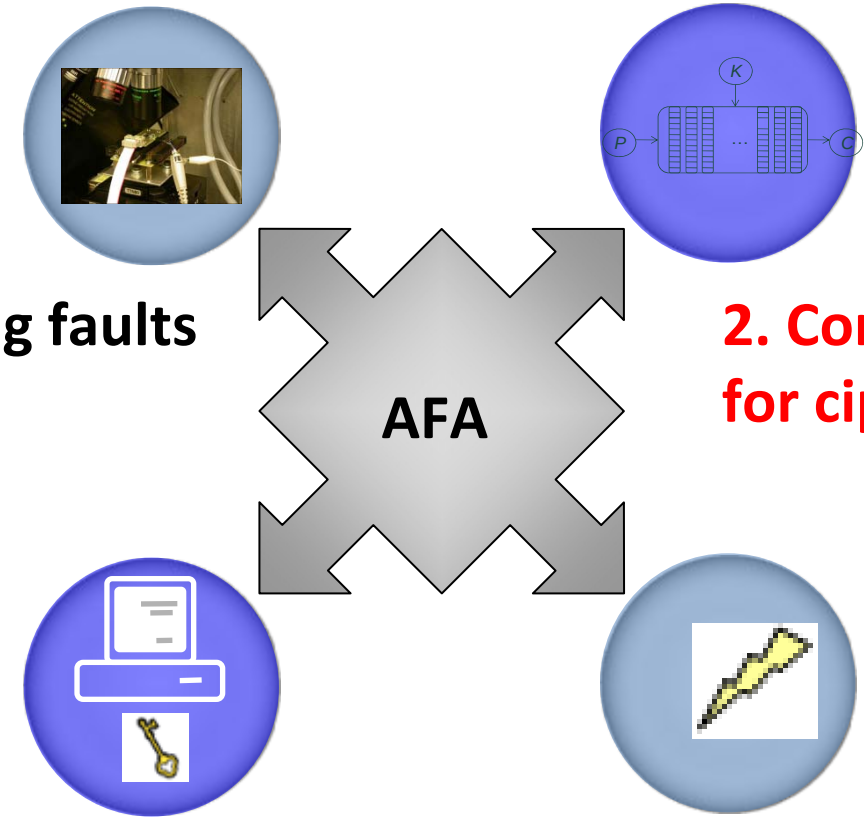
- On nibble-based model in the 30th round, one [2,3,4] or two [1] fault injections.
- Costly attack setup (workstation with 48GB RAM [2] or 50 GB RAM[3].
- Reduced key search space is 2^4 theoretically [4], but 2^{19}-2^{25} from experiments [2].
- AFA (14.67 hours in [3]) work seems to have no advantages over DFA (45 seconds in [2]).

Goal

- Improve efficiency and decrease cost.
- Accurately estimate the reduced key search space of DFA on LED.
- ADFA on LED using other fault models where traditional DFAs are difficult to work with.

Algebraic Fault Analysis

- ADFA



1. Inducing faults

2. Constructing equations for cipher

3. Constructing equations for faults

4. Solving equations

Step 1: LED Equation Sets

- Represent AK and AK^{-1}
- Representing AC and AC^{-1}

$$z_i = x_i + y_i \quad x_i = z_i + y_i$$

- Representing SC and SC^{-1}

$$y_0 = 1 + x_0 + x_2 + x_3 + x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_1x_2x_3$$
$$\dots$$
$$y_3 = x_0 + x_1 + x_3 + x_1x_2$$

- Representing SR and SR^{-1}

$$y_i = x_{(4R[i/4])+i\%4} \quad x_i = y_{(4R^{-1}[i/4])+i\%4}$$

- Representing MC and MC^{-1}

$$Y_0 = 4\,.X_0 + X_4 + 2.X_8 + 2.\,X_{12}$$
$$Y_1 = 4.\,X_1 + X_5 + 2.X_9 + 2.X_{13}$$
$$\dots$$
$$Y_{15} = 2.X_3 + 2.X_7 + \mathrm{F}.X_{11} + \mathrm{B}.X_{15}$$

Fault Model

• ADFA

- Assumption: a single random nibble fault into Y^{30}.

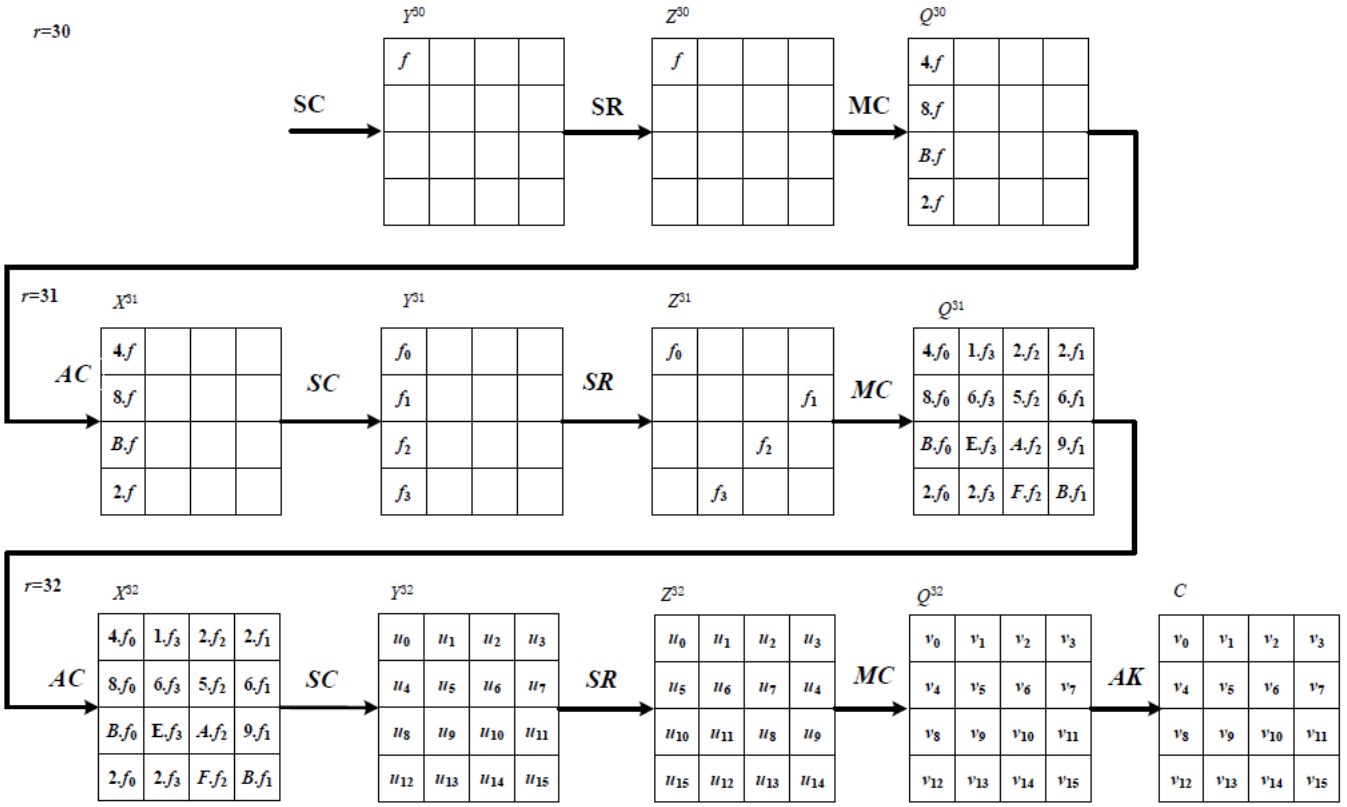


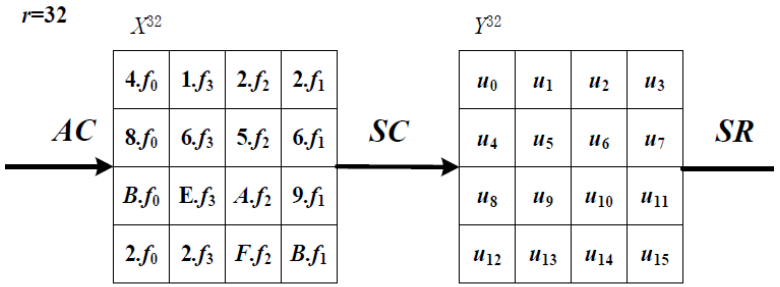
Figure 2. Fault model of ADFA on LED

Step 2: DFA

FDTC 2013

- ADFA

- Deduce ΔX^{32}



  - Calculate ΔY^{32} from ΔC:
  - Deduce the value of four nibbles f_0, f_1, f_2, f_3 to calculate ΔX^{32}. The differential S-Boxes for f_0 would be:

$$S_0[f_0\text{-}1][a] = S[a] + S[a+4*f_0]$$
$$S_4[f_0\text{-}1][a] = S[a] + S[a+8*f_0]$$
$$S_8[f_0\text{-}1][a] = S[a] + S[a+B*f_0]$$
$$S_{12}[f_0\text{-}1][a] = S[a] + S[a+2*f_0]$$

  - For each possible f_0 candidate, if u_0, u_4, u_8, u_{12} are all in the joint set of $S[f_0\text{-}1]$, then the candidate is kept for f_0.

Step 3: Equation of Faults

FDTC 2013

• ADFA

- Convert deduction of ΔX^{32} into algebraic equations.
  - Let $d=d^0, d^1, d^2, d^3$ denote the correct deduction of f. Let $D=d_1, d_2 \ldots d_n$ denote possible deduction set on d.
  - Representing relations of d and d_i (There is only one correct deduction in D)

$$e_i^j = \neg(d_i^j + d^j), \quad c_i = \prod_{j=1}^{b} e_i^j$$

$$c_1 \vee c_2 \vee \ldots \vee c_{s_p} = 1,$$
$$\neg c_i \vee \neg c_j = 1, \quad 1 \leq i < j \leq s_p$$

  - Representing Y^{30}
    - First nibble (with fault): $(1+\Delta y_0^{30}) \wedge (1+\Delta y_1^{30}) \wedge (1+\Delta y_2^{30}) \wedge (1+\Delta y_3^{30}) = 0$
    - Other 15 nibbles: $\Delta y_i^{30} = 0 \ (4 \leq i \leq 63).$

Step 4: Solver

- Combine the equation set of LED with injected fault and use CryptoMiniSat solver to recover the secret key.

- Most solver stops when one solution is found, but the first solution may not be the correct solution. CryptoMiniSat outputs all possible solutions. We modified it to count and output all possible solutions of the secret key.

Results

- Simulated fault injection.
  - 10,000 runs

- ADFA using nibble-based model.
  - Generate P, K, and get C
  - Inject fault at Y^{30} to get C*
  - Representing LED and faults
  - Deduce K

  - Multiple deductions on ΔX_i^{32}

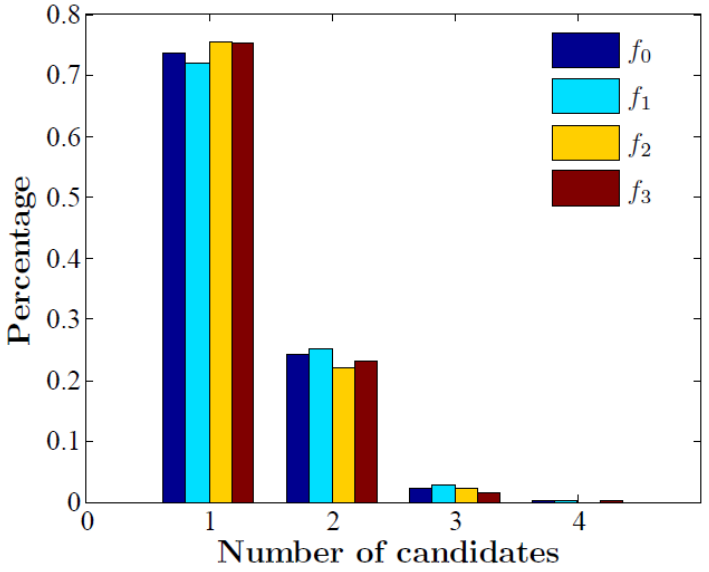


Figure 3. Distributions of the number of candidates for f_0, f_1, f_2, f_3

Equation solving time

• Results

- Algebraic equations for the reverse operations.
  - Includes P and full set of LED eq.
  - Solving time exp dist.
  - Correct key have success rate of 99.5% within 1hr.

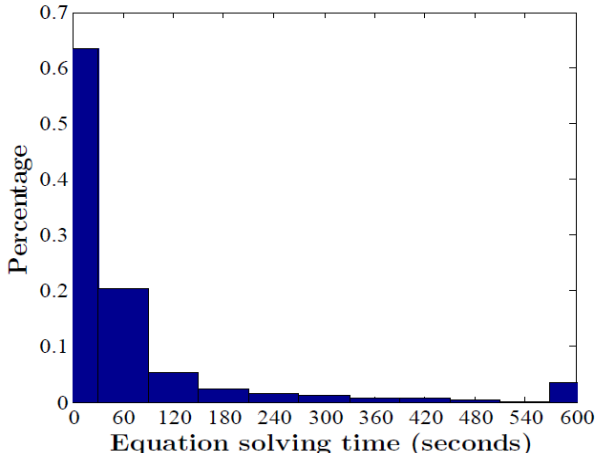


Figure 4. The distribution of time on solving equations in ADFA on LED

Attack	Model	Technique	Fault	Setup	Time
[1]	L=4, r=30	DFA	2	PC, 2GB RAM	-
[2]	L=4, r=30	DFA	1	Workstation, 48 GB RAM	45s
[3]	L=4, r=30	AFA	1	Workstation, 50 GB RAM	14.67 hours
This paper	L=4, r=30	ADFA	1	PC, 4 GB RAM	1-3 minutes
This paper	L=8, r=30	ADFA	1	PC, 4 GB RAM	1 hour

Reduced Key Search Space

• Results

- Based on incomplete avalanche effect of last round.

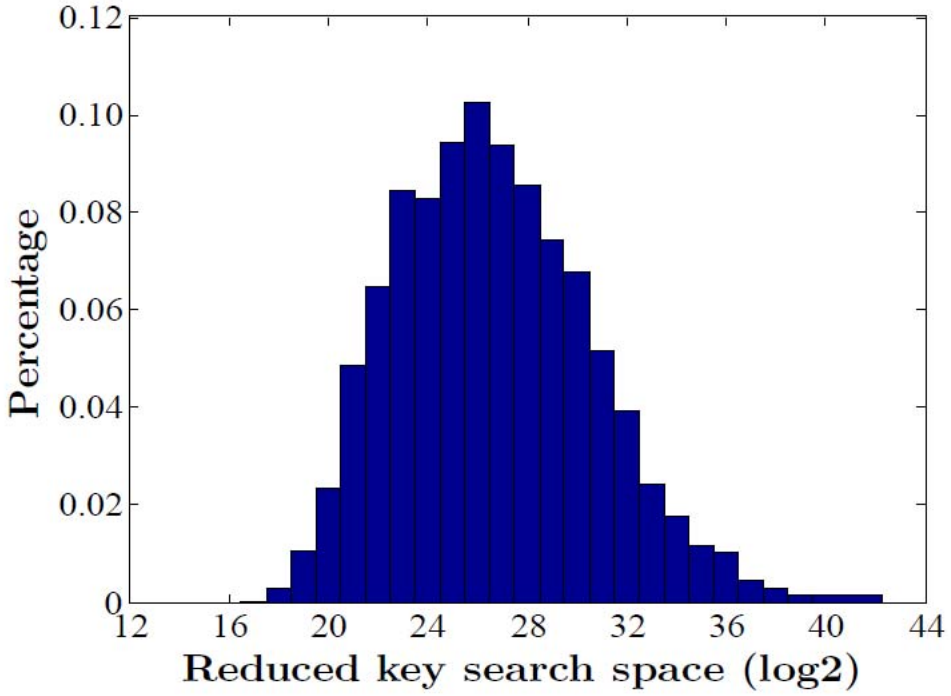


Figure 5. Reduced key search space of DFA on LED by analyzing the last round using Algorithm 1

Reduced Key Search Space

• Results

- Extend fault analysis to 30^{th} and 31^{st} round and uses a solver.
- 2^6-2^{17} (on average $2^{12.9}$).

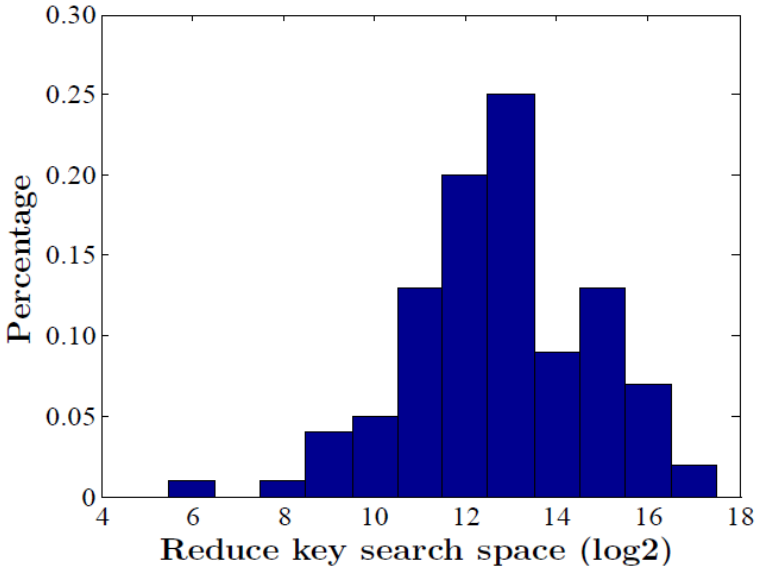


Figure 6. The reduced key search space of ADFA on LED under nibble-based fault model

Key Space Comparison

- Compared with the experimental results of other attacks, ours results in a smaller key search space.

Attack	Model	Technique	Key Space
[1]	L=4, r=30	DFA	2^{25}
[4]	L=4, r=30	DFA	2^{4}
[3]	L=4, r=30	AFA	$2^{19} \sim 2^{25}$
This paper	L=4, r=30	ADFA	$2^{12.9}$
This paper	L=8, r=30	ADFA	$2^{10.1}$

- Key search space is important for testing the resistance of ciphers against fault attacks.

Other Fault Models

• Results

- Byte-based fault model
- Diagonal-based fault model (1 diagonal fault)
- For both: 30th round, 100 runs, break in ~1hr.

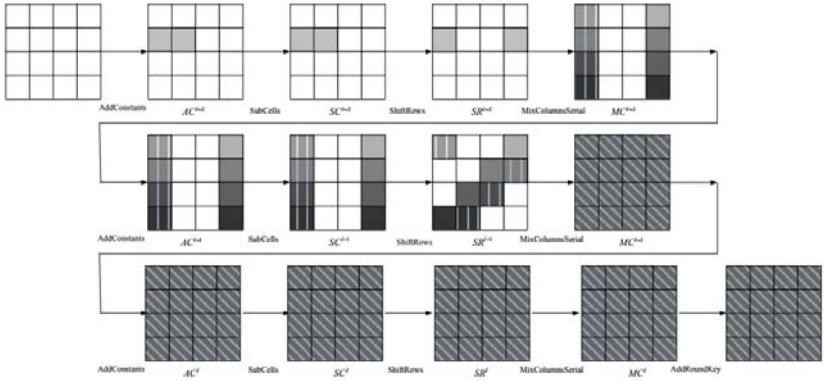
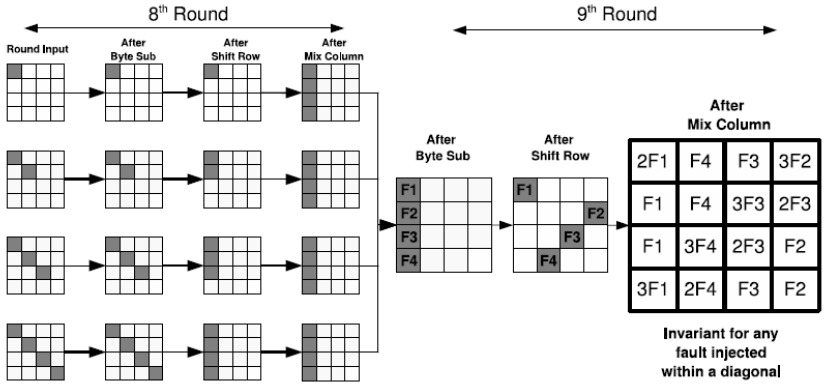


Fig. 2 One single-byte fault progagation path in the last three rounds of the LED cipher

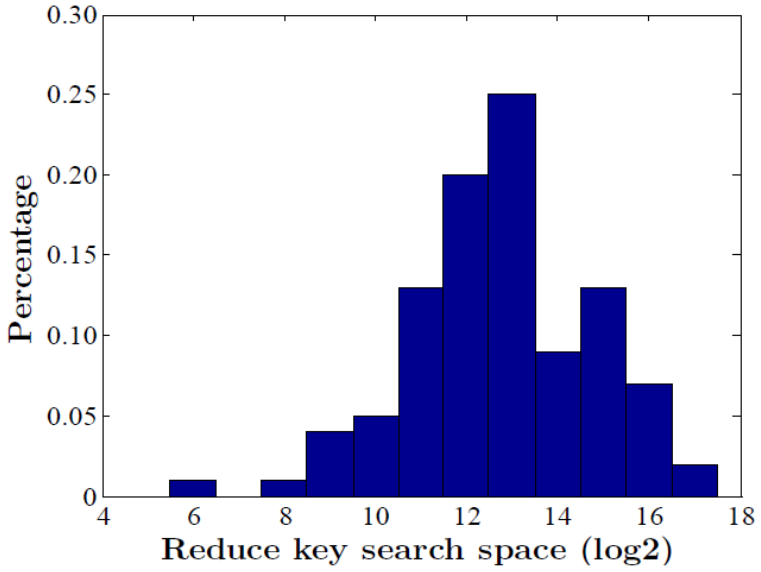


Byte Model: http://www.tandfonline.com/doi/pdf/10.1080/18756891.2012.733223
Diagonal Model: https://www.researchgate.net/publication/220336591_A_Diagonal_Fault_Attack_on_the_Advanced_Encryption_Sta

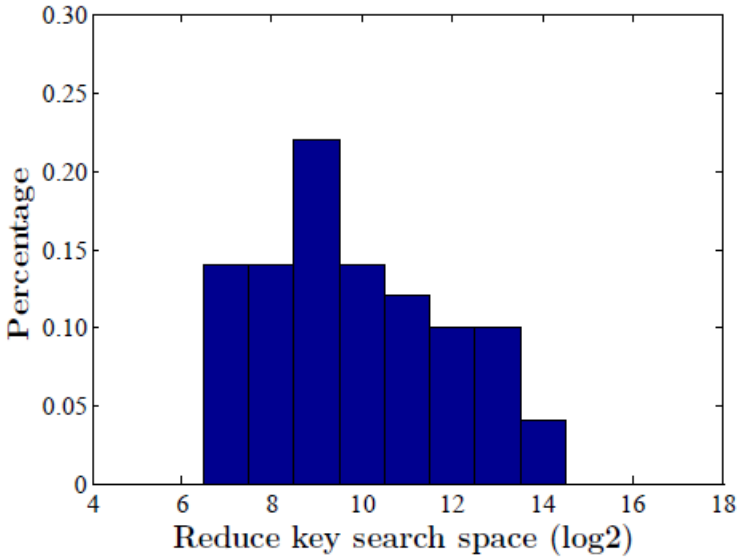
Fault Model Analysis

• Results

- Automated key search space algorithm used (more generic).
  - Only used the LED equations for the last 3 rounds (simpler).
  - Results shown are for first nibble. Does location affect key search space?



ADFA on 4-bit fault model 2^6-2^{17} (~$2^{12.9}$)



ADFA on 8-bit fault model 2^6-2^{14} (~$2^{9.9}$)

Fault Model Analysis

• Results

- Modified ADFA approach to evaluate DFA with solver
  - FA under byte-based model is more effective than that under nibble-based model for all fault locations

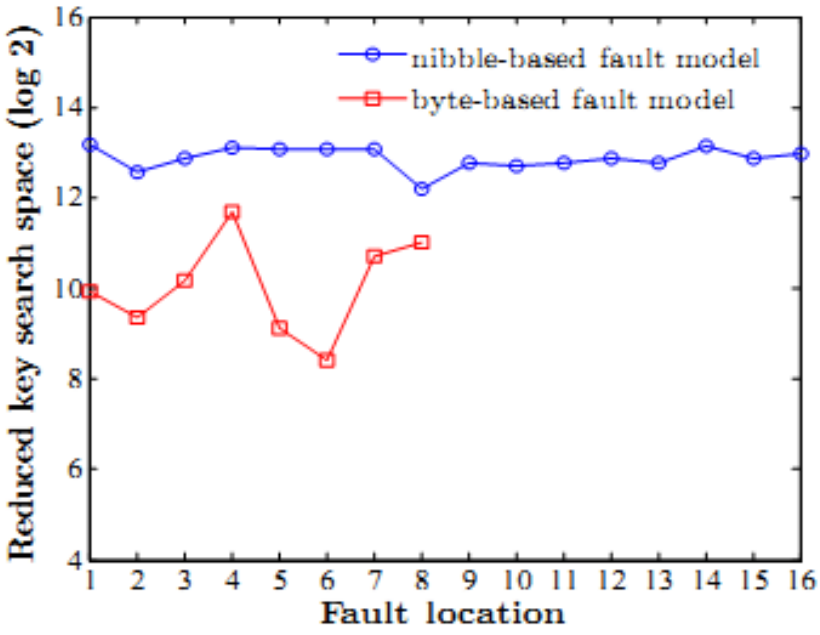


Figure 8. The reduced key search space of ADFA on LED for different fault locations

Conclusion

- **Optimized**: succeed on LED within 3 minutes and with one fault injection.
- **Efficient**: require only 1 fault (time depends on the algebraic structure and the fault models).
- **Automatic**: no need to analyze the fault propagation.
- **Generic**: apply to different ciphers and fault models.

Open Problems

Optimizing AFA

- Optimize the equation set
- Optimize the solving strategy

Analyzing AFA

- What are the dependencies of AFA?
- How to compare AFA, DFA, SFA?
- When to use AFA, DFA, SFA?

Applying AFA

- Apply to more complicated ciphers
- Generate a universal evaluating tool

Defending AFA

- Design AFA resistant nonlinear function

References

- [1] W. Li, D.W. Gu, et al. Security Analysis of the LED Lightweight Cipher in the Internet of Things. Chinese Journal of Computers, 2012, 35(3): 434-445. DFA, two faults, key search space of LED-64 is reduced to 2^{25}, with experiments!
- [2] P. Jovanovic, M. Kreuzer, and I. Polian. A Fault Attack on the LED Block Cipher. In Proceedings of COSADE 2012, LNCS, vol. 7275, pp. 120-134, 2012. DFA, one fault, workstation with 48GB RAM, key search space of LED-64 is reduced to 2^{19}-2^{25}, with experiments!
- [3] P. Jovanovic, M. Kreuzer and I. Polian, An Algebraic Fault Attack on the LED Block Cipher. Cryptology ePrint Archive. Available: http://eprint.iacr.org/2012/400.pdf, 2012. AFA, One fault, workstation with 50GB RAM, 14.67 hours.
- [4] K. Jeong and C. Lee. Differential Fault Analysis on Block Cipher LED-64. Future Information Technology, Application, and Service, LNEE, vol. 164, pp.747-755, 2012. DFA, one fault, key search space of LED-64 is reduced to 2^4, no experiments.