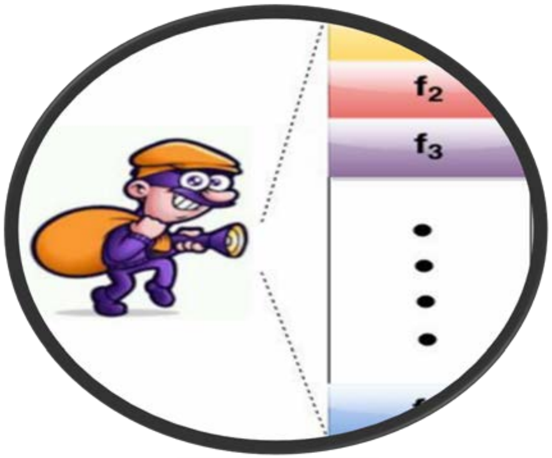
Amsterdam, The Netherlands, 9-12th September, 2018

ExpFault: Automated Framework for Exploitable Fault Characterization in Block Ciphers

Sayandeep Saha, Debdeep Mukhopadhyay, and Pallab Dasgupta

Motivation

Testing Block Ciphers for Fault Attacks



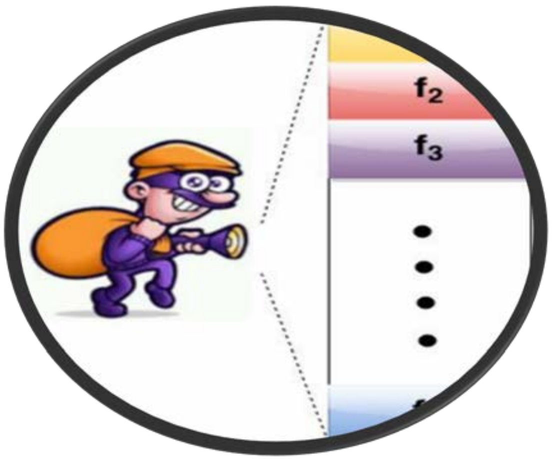
Securing cryptographic devices against fault attacks

- Several countermeasures exist :
  - Either extremely resource hungry
  - Or not robust against all possible faults
- Malicious faults are highly repeatable
- Require precise idea of the *fault model*
  - ***Which faults do we need to prevent***?

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Motivation

Testing Block Ciphers for Fault Attacks



Securing cryptographic devices against fault attacks

- Several countermeasures exist :
  - Either extremely resource hungry
  - Or not robust against all possible faults
- Malicious faults are highly repeatable
- Require precise idea of the *fault model*
  - ***Which faults do we need to prevent***?

"Exploits" of Exploitable Faults

- Designing precise countermeasures
- Testing countermeasures
  - On "non-random" exploitable fault space.
- Cipher evaluation

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Motivation

Challenges

Not all faults are malicious/exploitable

- Example of malicious fault: 8^{th}/ 9^{th} round byte faults in AES
- Example of non-malicious fault: 5^{th} round byte fault in AES

Fault space for a block cipher:

- Prohibitively large !!!
- Manual fault analysis methodologies -- Impractical !!!
  - **It took 8 years to reach the optimal attack for AES**

Too many ciphers !!!

- Trend of designing application-specific lightweight ciphers
- **Recent NIST call for lightweight block ciphers.**

NIST call for lightweight cryptography (http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf)

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Motivation

Challenges

Not all faults are malicious/exploitable
- F … S
- …

Fault …
- …
- …

An automation:
- **Generic**
- **Fast**
- **Scalable.**

Too many ciphe…
- Trend of designing application-specific lightweight ciphers
- **Recent NIST call for lightweight block ciphers.**

NIST call for lightweight cryptography (http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf)

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Fault Attack Automation

State-of-the-art

Algebraic Fault Attack (AFA)

- Generic representation.
- Use of SAT solvers.
- Not so fast !!!
- Lack of interpretability.

F. Zhang et.al. , “A Framework for the Analysis and Evaluation of Algebraic Fault Attacks on Lightweight Block Ciphers”, *IEEE Transactions on Information Forensics and Security*, *11*(5), 1039-1054., 2016

Synthesis of Fault attacks

- Program synthesis based
- Demonstrated on Public key systems

Gilles Barthe, et al. "Synthesis of fault attacks on cryptographic implementations." *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*., 2014.

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

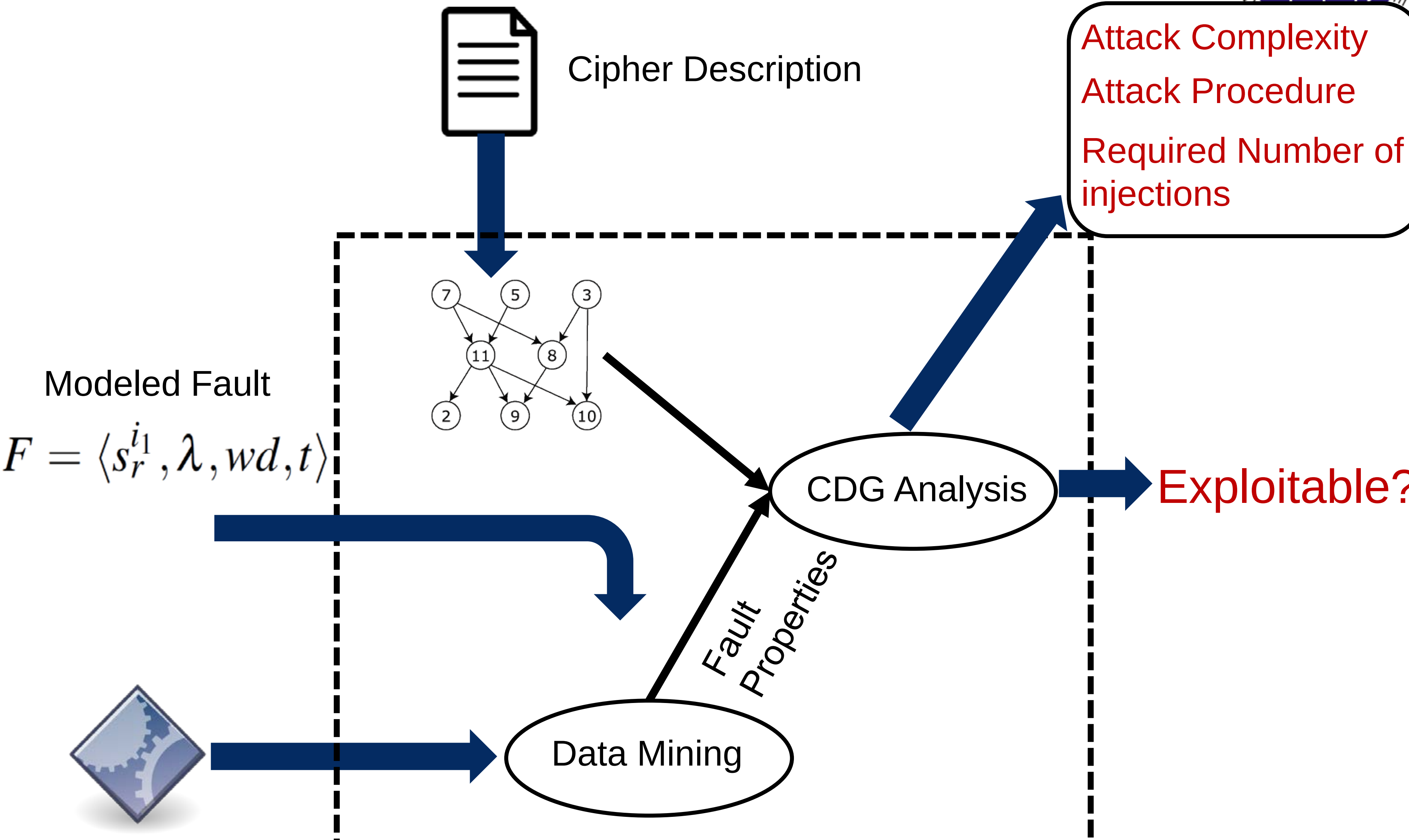
- What DFA does?
  - Reduces the key search space with faults
  - Exhaustive search within practical limits

- What we suggest …
  - Do not perform the exhaustive search
  - Automatically compute the search complexity

- Advantage
  - Fast characterization of each individual faults
  - Challenge: Generic algorithm.

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach: ExpFault



© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

What we have obtained so far…

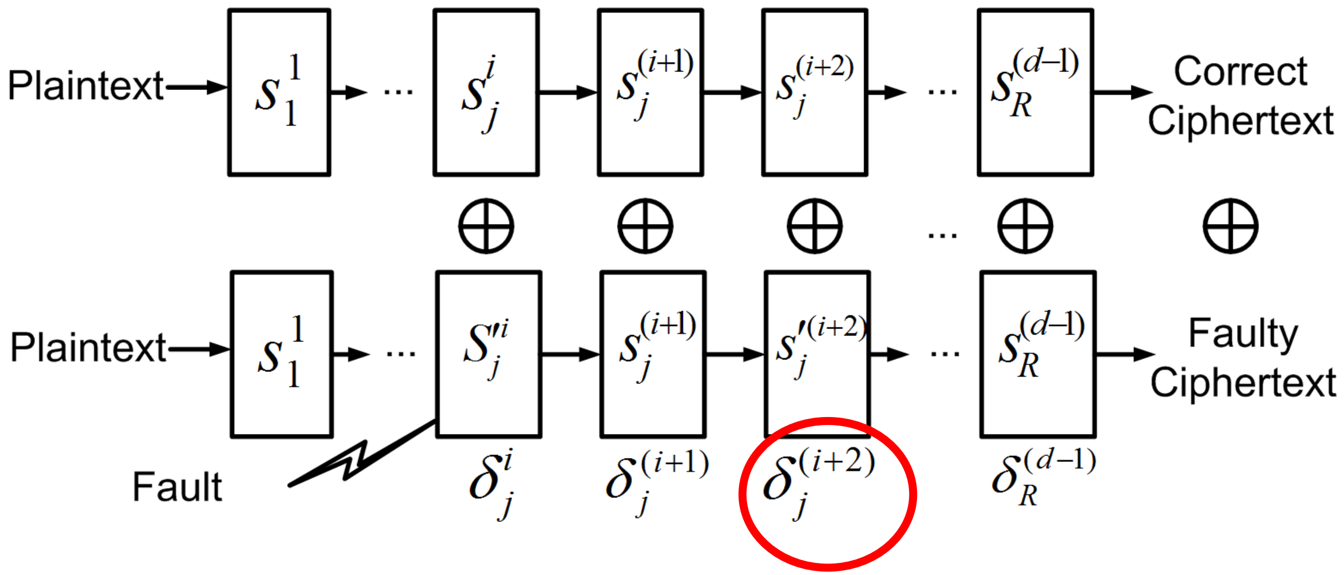
Block Cipher	Fault Location	Fault Model	Remain key space complexity	Number of Injections Required	Optimal?
AES	8^{th} round SBox	byte	2^8	1	Yes
AES	7^{th} round SBox	byte	1	2043	_
PRESENT	28^{th} round SBox	Multi-byte	1	2	Yes
GIFT	25^{th} and 23^{rd} round SBox	nibble	$2^{7.06}$	2	Yes

The attacks on GIFT were not previously known!!!!

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

How to Represent a Cipher



We analyze the dataset for each "state-differential" δ_j^i

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

How to Represent a Fault

Width of the fault (fault model)

State of injection

$$F = \langle s_r^{i_1}, \lambda, wd, t \rangle$$

Width of the state

Location of fault in the state

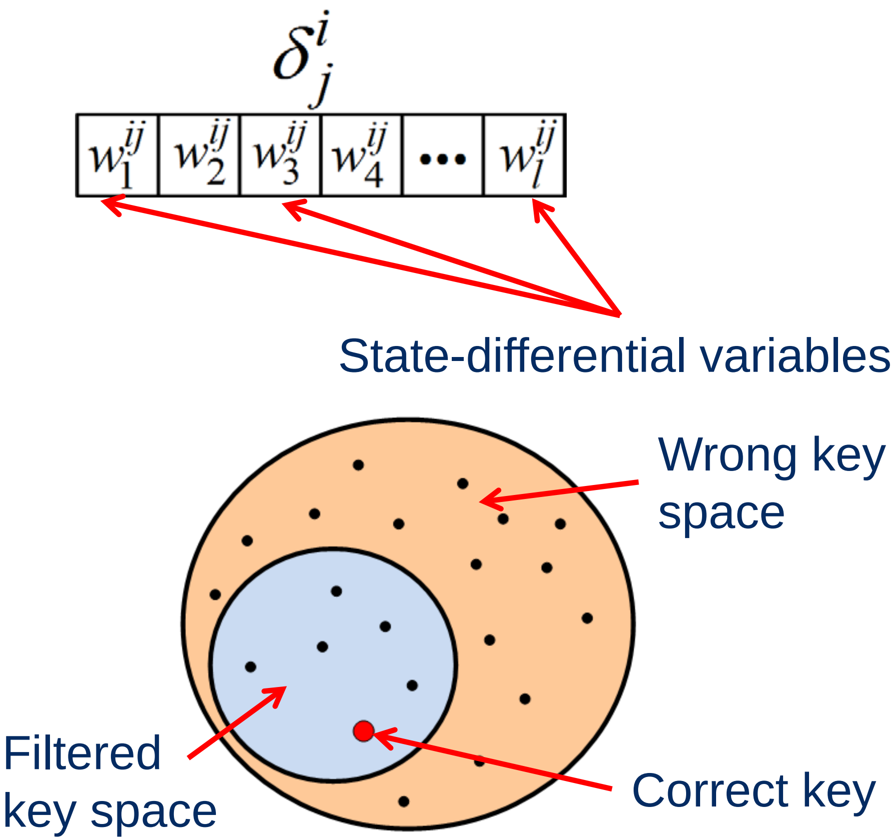
Injection round

Fault and plaintext values are abstracted out

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

What is a "Distinguisher"?

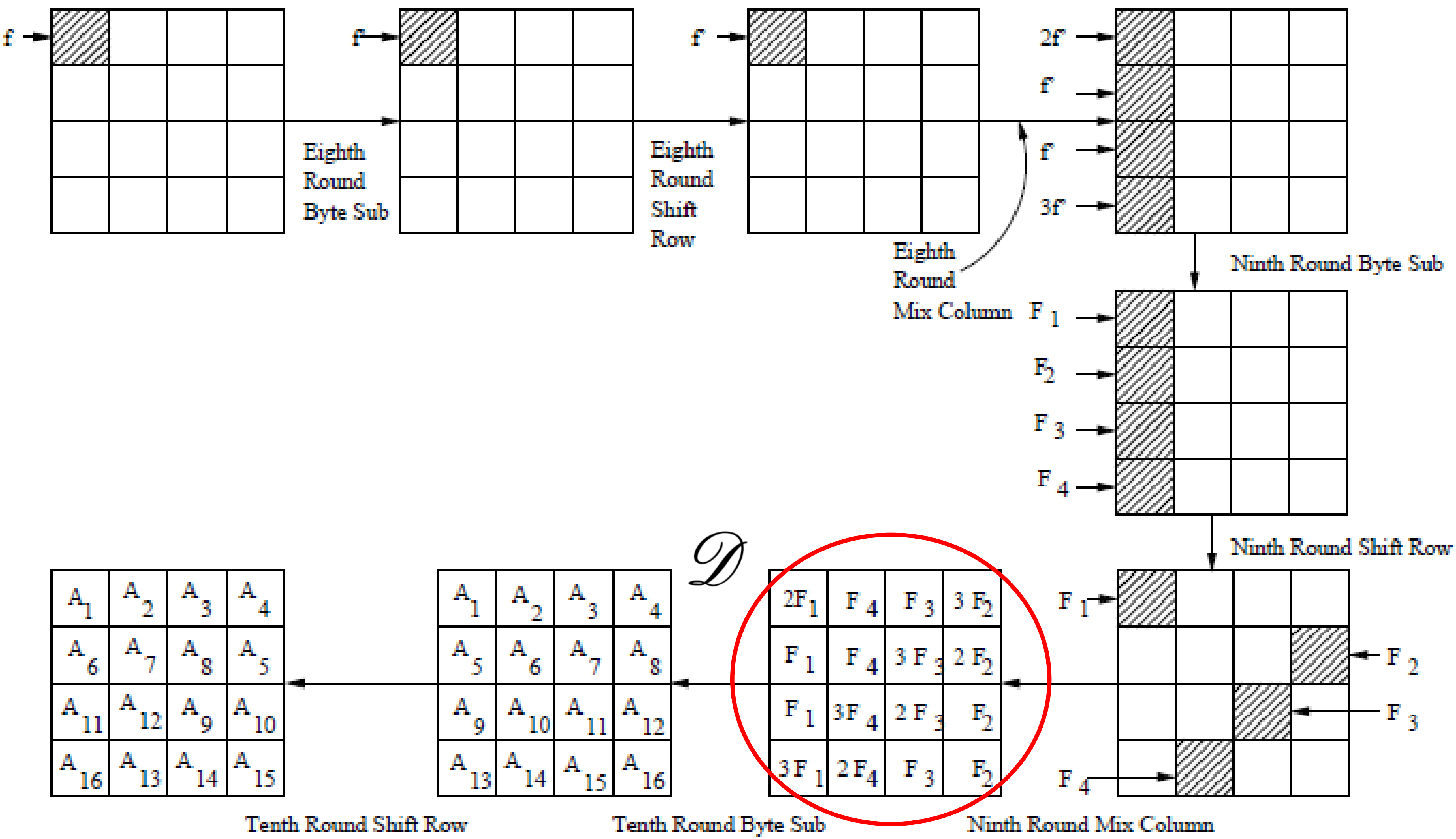


- Constraints over state-differential variables
- Satisfiable for the correct key and few others
- Results in non-uniform nature of the state-differential distribution for a "small" subset of keys
- Works as "filter" for wrong keys

Some of the δ_j^i s are "Distinguishers", But not all of them

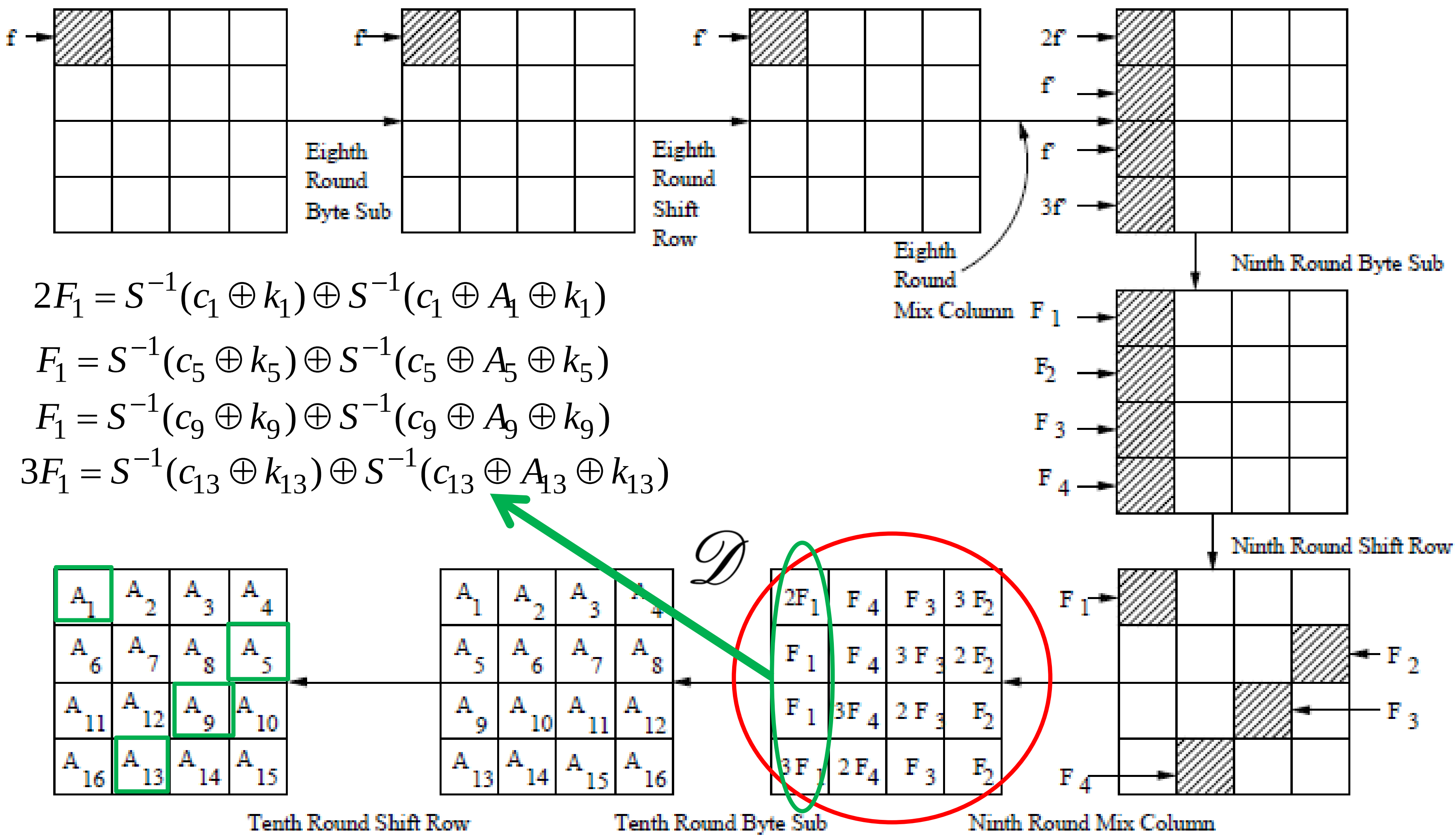
© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach



© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach



$$2F_1 = S^{-1}(c_1 \oplus k_1) \oplus S^{-1}(c_1 \oplus A_1 \oplus k_1)$$

$$F_1 = S^{-1}(c_5 \oplus k_5) \oplus S^{-1}(c_5 \oplus A_5 \oplus k_5)$$

$$F_1 = S^{-1}(c_9 \oplus k_9) \oplus S^{-1}(c_9 \oplus A_9 \oplus k_9)$$

$$3F_1 = S^{-1}(c_{13} \oplus k_{13}) \oplus S^{-1}(c_{13} \oplus A_{13} \oplus k_{13})$$

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

- Formalization of DFA:
  - A DFA algorithm can be represented as follows:

$$\mathscr{A} = \langle \{\mathscr{D}_j^i\}, \mathscr{T}, \mathscr{R} \rangle$$

$\{\mathscr{D}_j^i\}$ A set of fault distinguishers, Constructed over the XOR differentials of a cipher state

$\mathscr{T}$ An algorithm to evaluate the distinguisher over key guesses

$\mathscr{R}$ Remaining key space, filtered with the distinguisher

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

Phase 1: Distinguisher from Fault Simulation Data?

Distinguishing Criteria: A state differential is a distinguisher iff state differential entropy is less than its maximum possible value.

Two Possible Cases

Case 1

Each w_z^{ij} is statistically independent

- Find variable ranges
- Reduced range → reduced entropy

Case 2

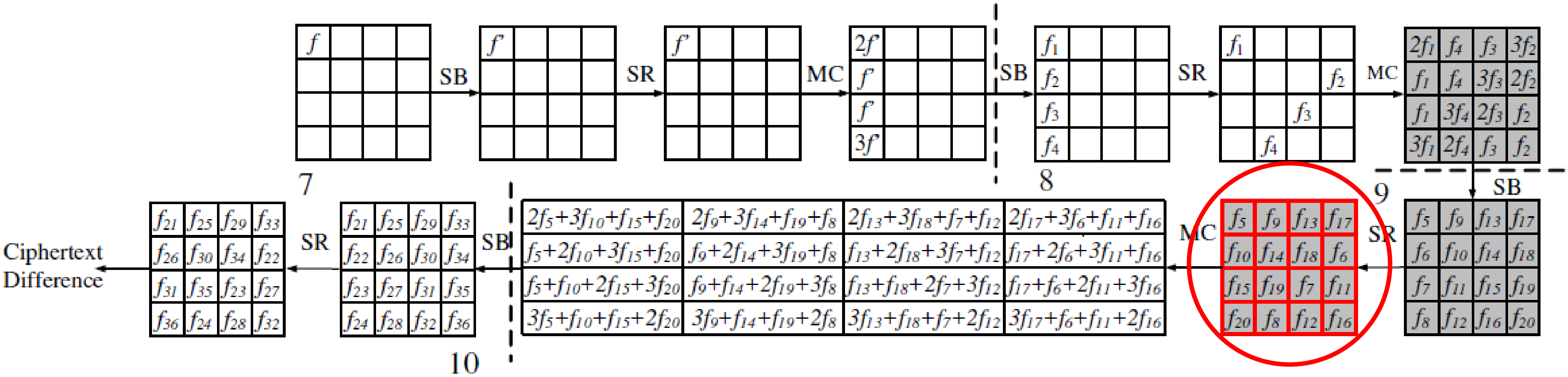
Some of the w_z^{ij} s are statistically dependent

- **Mine** variable dependencies
- Dependency → reduced entropy

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

Example 1: Impossible Differential Distinguisher

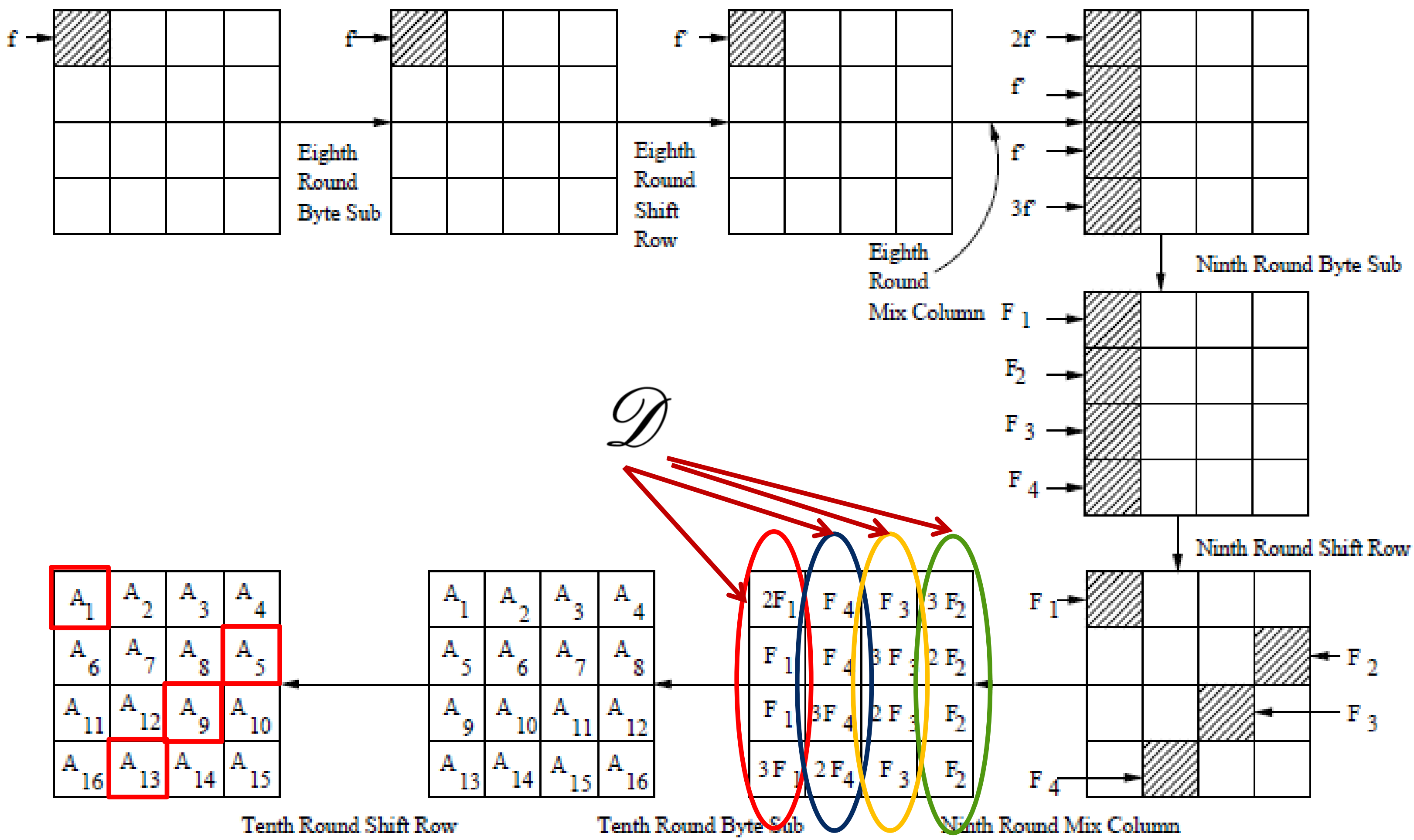


$$H_{Ind}(\delta_9^4) = 127.90$$

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

Example 2: Distinguisher with Correlated Variables



© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

Phase 2: Calculate Distinguisher Evaluation Complexity

- Target: Evaluate a distinguisher
  - Identify the key bits to guess:
    - To evaluate each w_z^{ij}
    - To evaluate each variable set if exists.

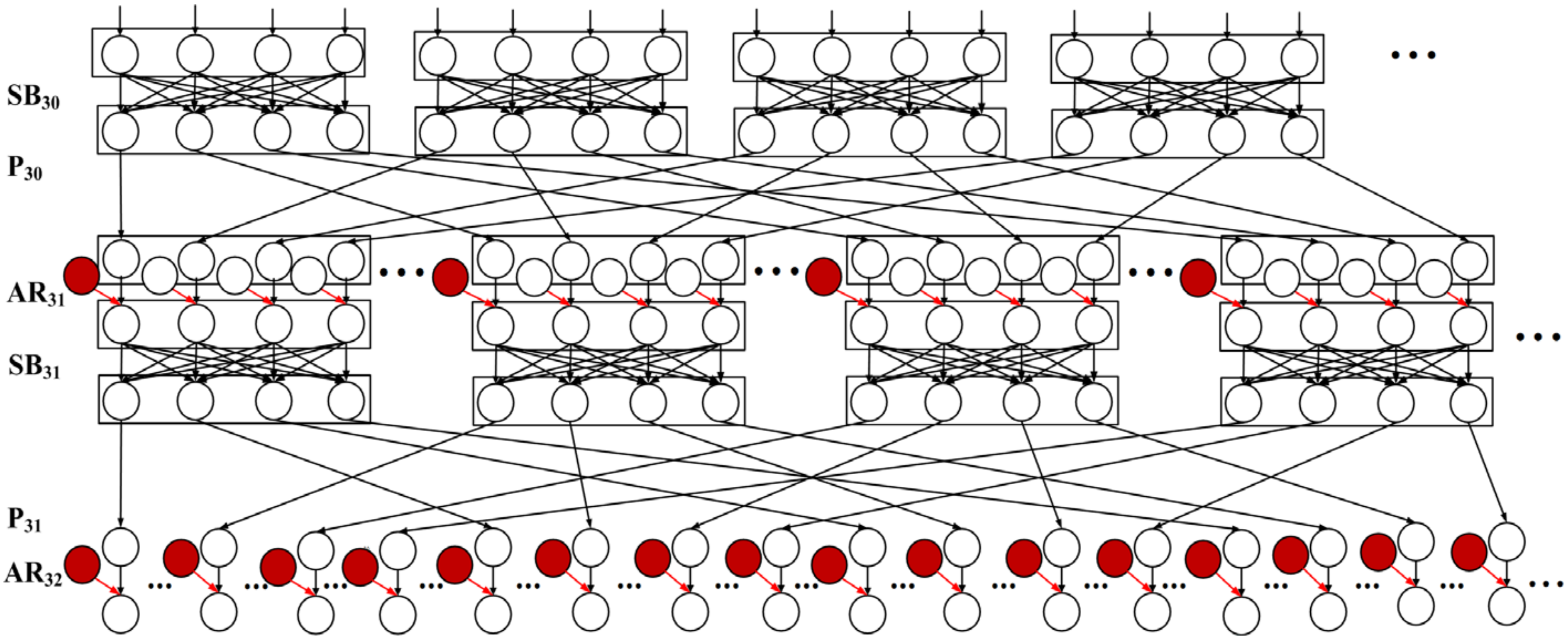
A Graph Based Abstraction of the Cipher

- Cipher Dependency Graph (CDG)
  - Each node is a bit from any cipher state.
  - Directed links: Causal dependencies among bits

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

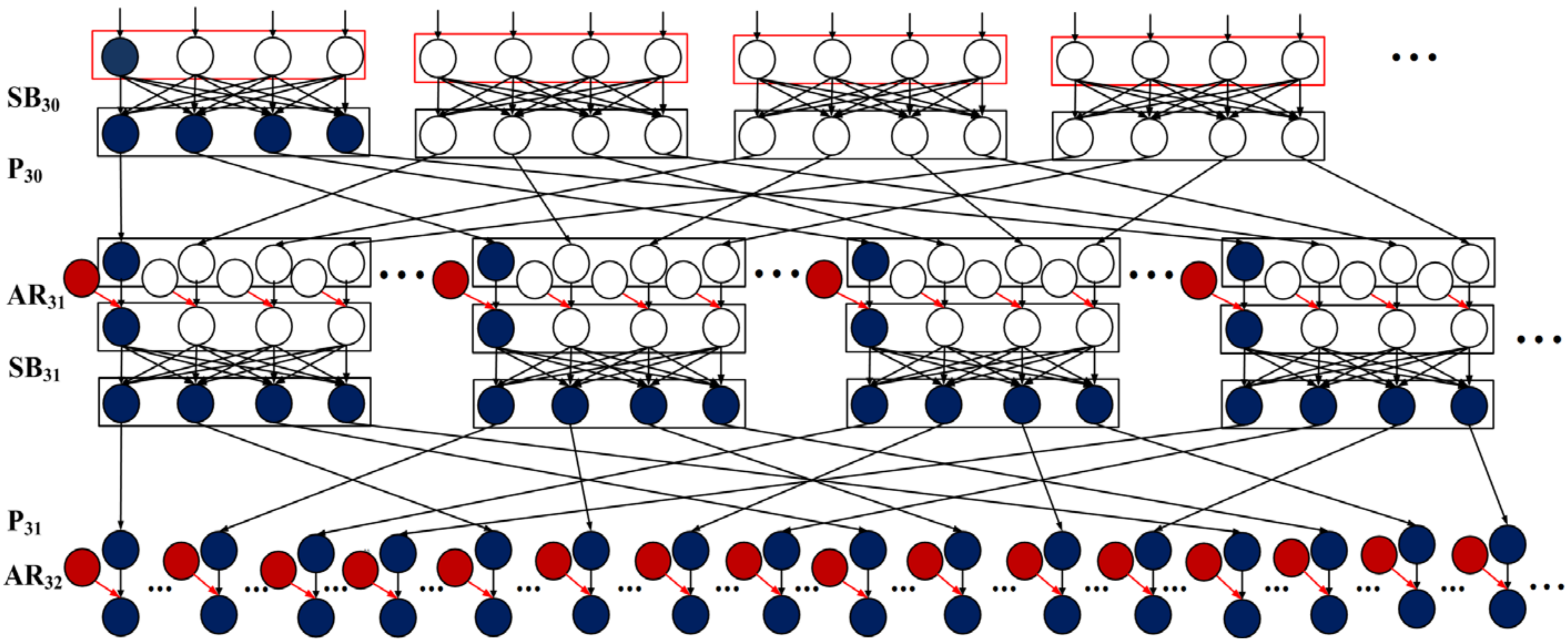
Proposed Approach

CDG: An Example



Proposed Approach

CDG: How Does it Work?



© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

CDG: Maximum Independent Key Set and Variable Group

- Each (MKS_h, VG_h) pair represents an independent subpart of the distinguisher evaluation.
- Each Subpart can be evaluated in parallel.

$$\mathbb{T}(h) = 2^{|MKS_h|}$$

$$max_h(\mathbb{T}(1), \mathbb{T}(2), ..., \mathbb{T}(M))$$

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Proposed Approach

Phase 3: Size of the Remaining Key Space

- Calculate the "probability of occurrence of the distinguishing property" with each VG_h

$$\mathscr{D}_j^i := \langle \{w_z^{ij}\}_{z=1}^{l}, \{Rng_{w_z^{ij}}\}_{z=1}^{l}, VS_{\delta_j^i}, \{IS_{\delta_j^i}^{v}\}_{v=1}^{|VS_{\delta_j^i}|} \rangle$$

for each VG_h

$$k_{size} := \texttt{BitCount}(MKS_h)$$

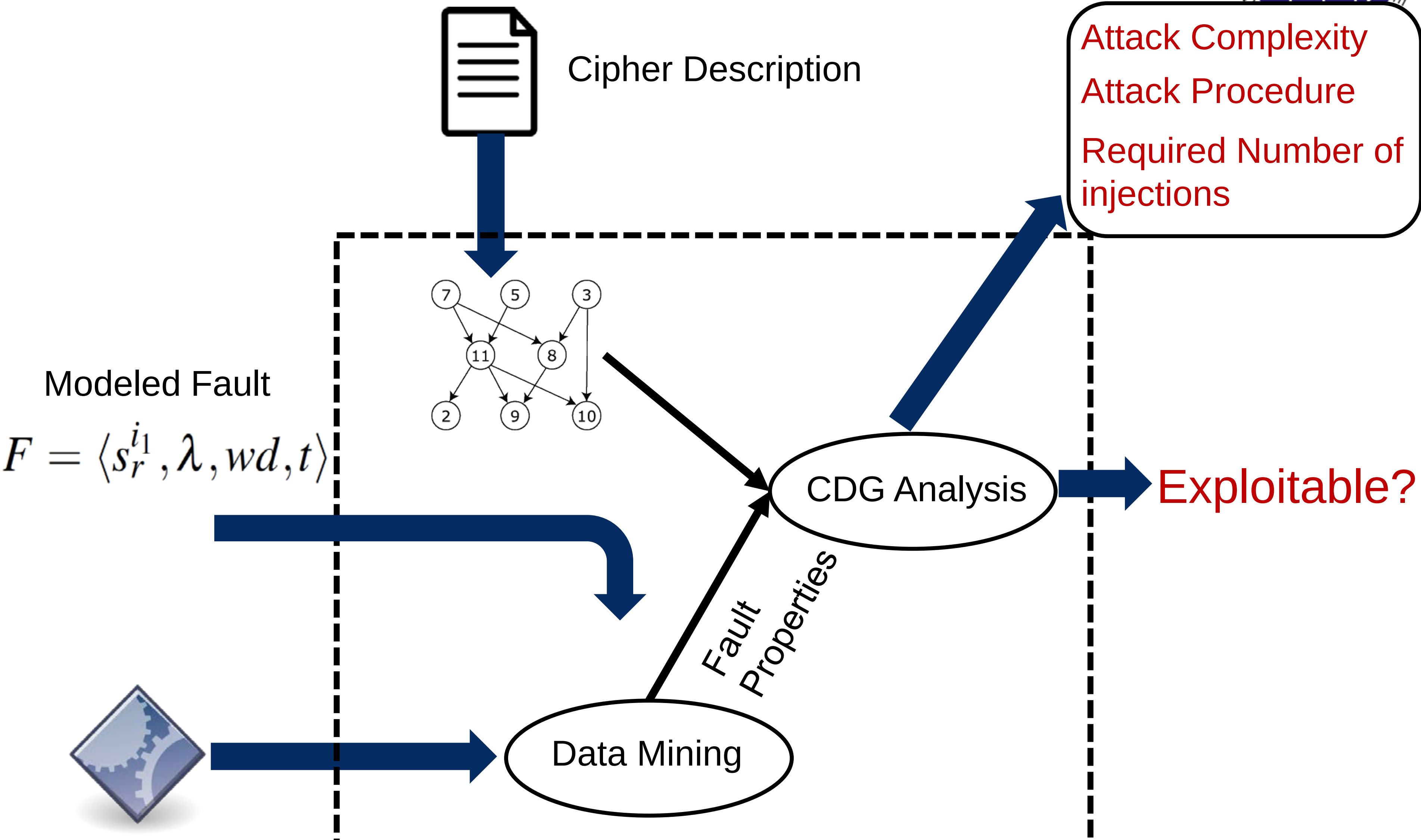
$$|\mathscr{R}|_{VG_h} := 2^{k_{size}} \times \mathbb{P}[VG_h]$$

$$|\mathscr{R}| := |\mathscr{R}| \times |\mathscr{R}|_{VG_h}$$

Calculated using these information

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

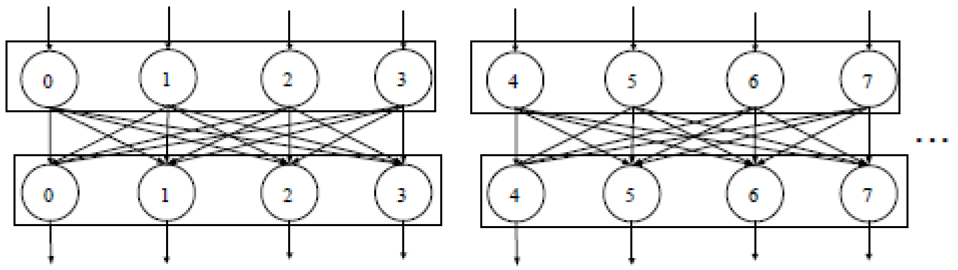
Proposed Approach: ExpFault



© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Implementation Details

```
BEGINBLOCK SLAYER
OPTYPE NONLINEAR
OPINPUT 64
OPOUTPUT 64
% 0 = 0,1,2,3
% 1 = 0,1,2,3
% 2 = 0,1,2,3
% 3 = 0,1,2,3

% 4 = 4,5,6,7
% 5 = 4,5,6,7
% 6 = 4,5,6,7
% 7 = 4,5,6,7
:
:
:
:
ENDBLOCK SLAYER
```



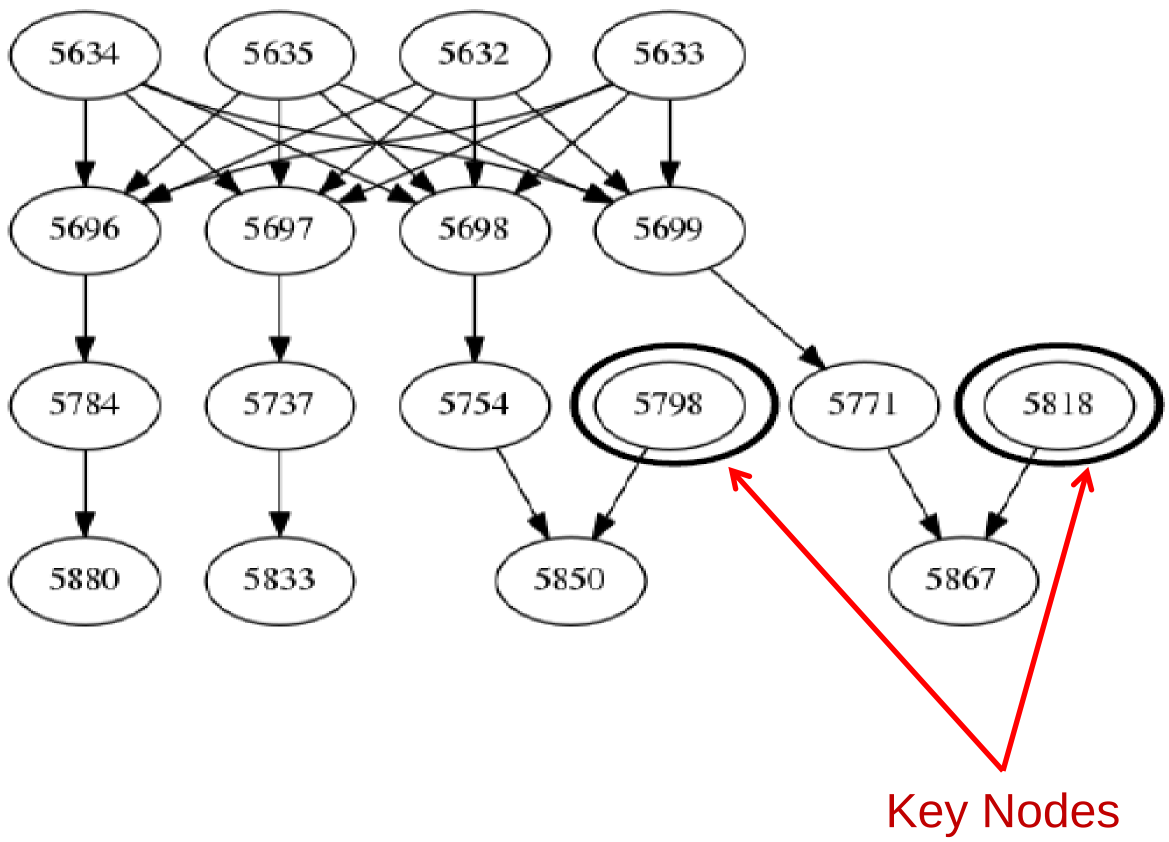
© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Implementation Details

```
-----------------------------------------------------------
Distinguisher Evaluation Complexity (in log scale) 8
Remaining Key Space Complexity (in log scale) 11.536682076433 74

Distinguisher Level 79
Round_no 27
Subop_no 2
Has_associations False
Entropy  43.536682076433735

V2 [0, 3, 5, 7, 9, 13, ]
V0 [0, 3, 5, 7, 9, 13, ]
V12 [0, 3, 7, 11, 15, ]
V6 [0, 5, 6, 9, 10, 13, 14, ]
V8 [0, 5, 6, 8, 9, 10, 11, 12, 15, ]
V14 [0, 3, 7, 11, 15, ]
V3 [0, 3, 5, 7, 9, 13, ]
V1 [0, 3, 5, 7, 9, 13, ]
V5 [0, 5, 6, 9, 10, 13, 14, ]
V10 [0, 5, 6, 8, 9, 10, 11, 12, 15, ]
V9 [0, 5, 6, 8, 9, 10, 11, 12, 15, ]
V13 [0, 3, 7, 11, 15, ]
V7 [0, 5, 6, 9, 10, 13, 14, ]
V4 [0, 5, 6, 9, 10, 13, 14, ]
V11 [0, 5, 6, 8, 9, 10, 11, 12, 15, ]
V15 [0, 3, 7, 11, 15, ]

No Variable sets exist..

----------------------------------
```


© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

How They Really Look Like?



© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Summary

- Characterization of the exploitable fault space for a block cipher is a problem of immense practical value.
- Exploitable fault space characterization demands fast, generic and automated mechanism for the characterization of individual fault instances.
- A fast automation is proposed
  - Need not to do the attack; just calculate the complexity
  - Best case attack complexity.
  - Number of injections
  - Attack description.
- Future works:
  - Further generalization – Key schedule attacks, DFIA attacks
  - Automatic generation of attack equations.
  - Synthesis of optimal countermeasures.
  - Countermeasure Vulnerability analysis.

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Acknowledgements

Demo: ExpFault on AES

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Acknowledgements

This project is funded by Synopsys CAD Labs under the project entitled "*Formal Methods for Physical Security Verification of Cryptographic Designs Against Fault Attacks*"

 © CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

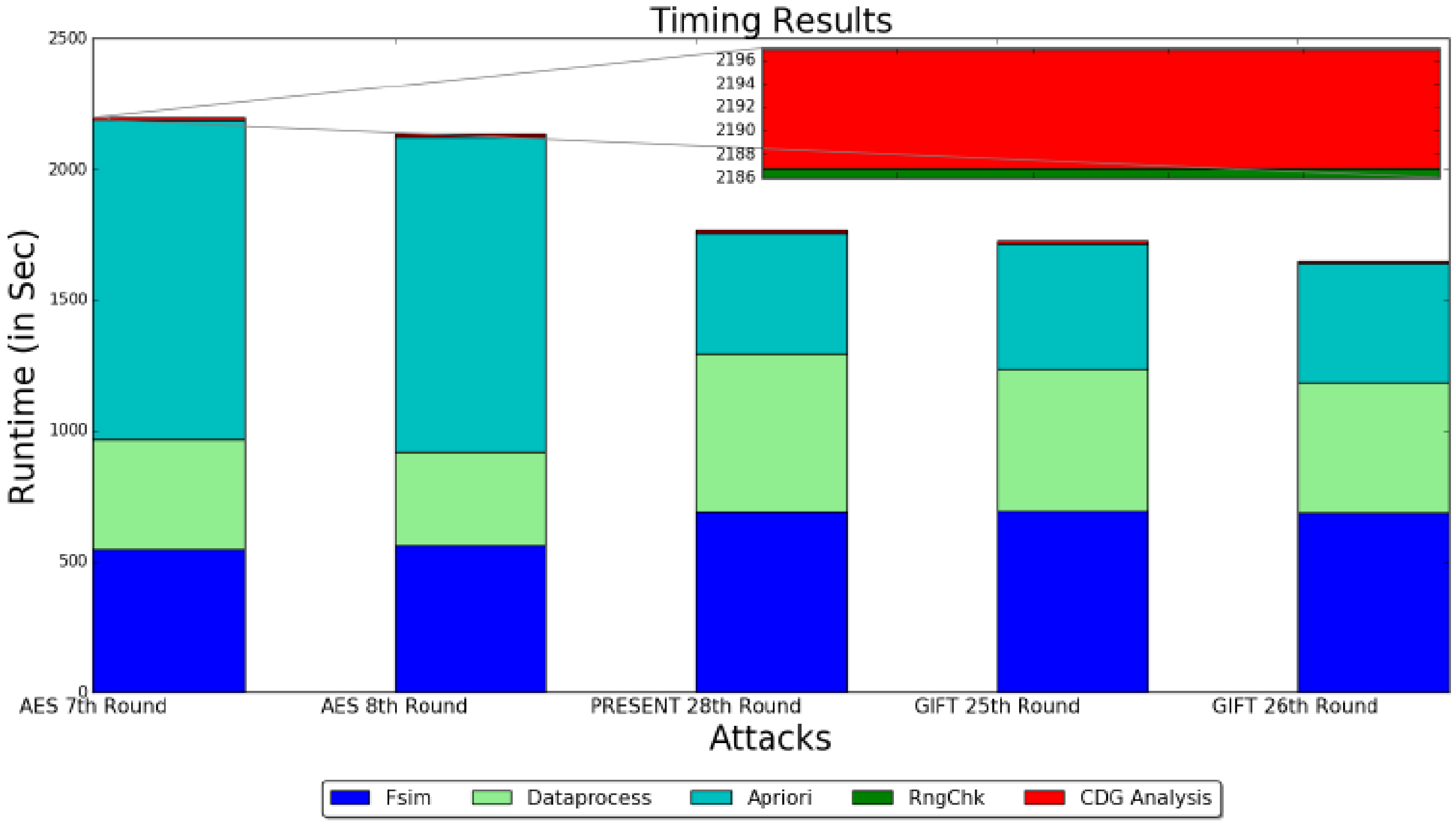
Thank You

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Backup Slides

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

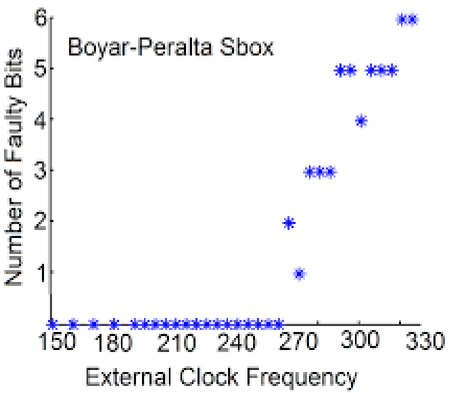
Implementation Details: Runtime



© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Introduction

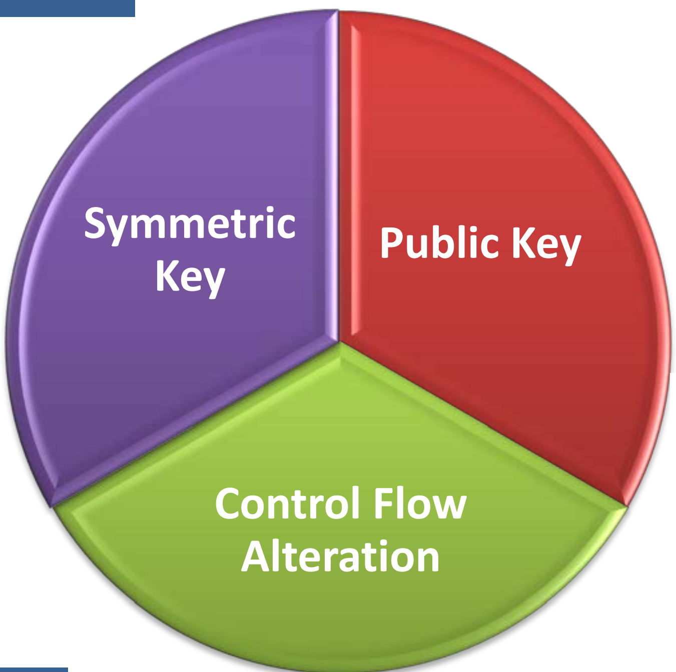
Localized Random Faults



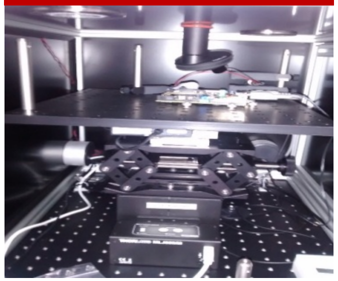
Biased Faults

Instruction Skip/Modify

```
LDD  R24, Y+i    // load subkey
LD   R25, X      // load state
EOR  R24, R25    // ExclusiveOR
STD  Z+i, R24    // store result
```

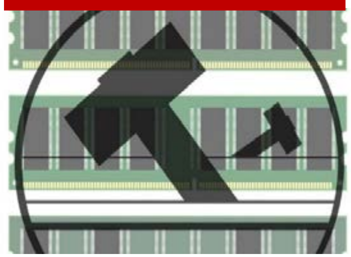


Laser-FI



EM-FI

Row-Hammer



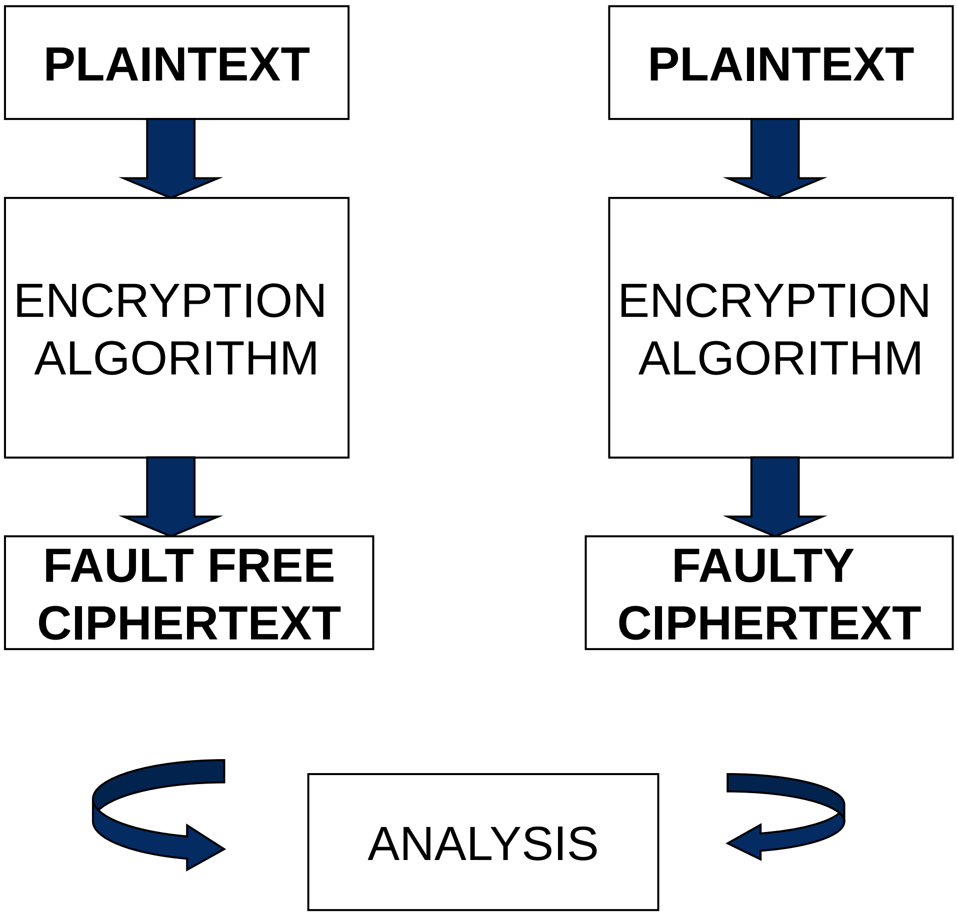
Voltage-Glitch

Clock-Glitch

 © CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Introduction

Differential Fault Analysis (DFA)



- Most widely explored
- Low fault complexity
- Analysis: complex
- Fault Locations
  - Datapath
  - Key-schedule
- Fault models
  - Bit based
  - Nibble based
  - Byte based
  - Multiple byte based

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018

Motivation

Cipher Designer

Engineer

© CONFERENCE ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 2018