# On Fault Injections in Generalized Feistel Networks

Hélène Le Bouder[1], **Gaël Thomas**[2], Yanis Linge[3], Assia Tria[4]

[1]École Nationale Supérieure des Mines de Saint-Étienne
[2]XLIM Université de Limoges
[3]STMicroelectronics
[4]CEA-Tech

Fault Diagnosis and Tolerance in Cryptography 2014

# Introduction

- Security is a key component for information technologies and communication
- Even securely designed algorithm may be vulnerable to physical attacks
- **Fault injection attacks** (FIA): disrupt and exploit the circuit behaviour
- But FIA can damage the circuit

⇒ The number of fault injections is a critical aspect of FIA

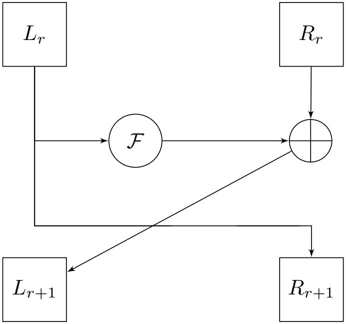
# This Paper

- FIA on **Generalized Feistel Networks**
- Single-bit fault model
- Find the most critical locations for FIA
- Assess the number of faults needed
- **Generic** Approach

# Plan

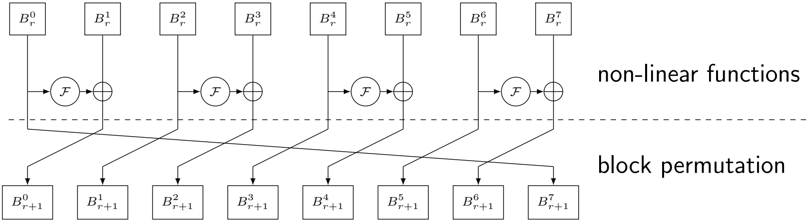
# The Original Feistel Structure

- Designed by Horst Feistel at IBM in the 1970's
- Used in DES, Camellia, MIBS, Simon,...
- Build 2*n*-bit permutation from *n*-bit to *n*-bit (Feistel) functions
- Similar encryption and decryption up to subkeys order



# Generalized Feistel Networks

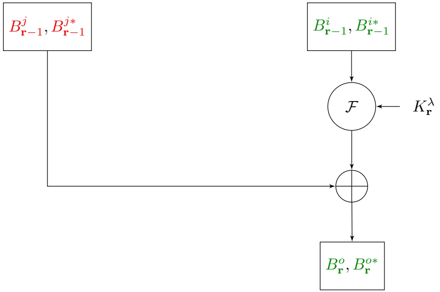
- Introduced by Zheng, Matsumoto, and Imai at CRYPTO '89
- Splits the message into **b** $\geq 2$ *n*-bit-long blocks



# Differential Fault Analysis (DFA) on GFNs

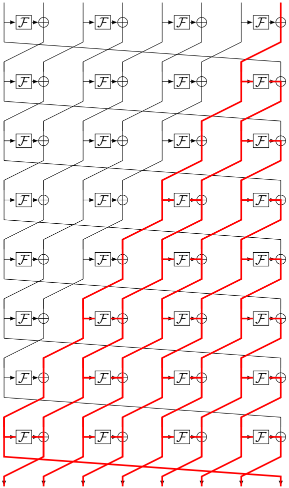
DFA is a powerful cryptanalytic technique that exploits differences between the correct ciphertext and erroneous results due to fault injections.

$$\Delta_{B_{\mathbf{r}}^{o}} = \mathcal{F}\left(B_{\mathbf{r}-1}^{i}, K_{\mathbf{r}}^{\lambda}\right) \oplus \mathcal{F}\left(B_{\mathbf{r}-1}^{i*}, K_{\mathbf{r}}^{\lambda}\right) \oplus \Delta_{B_{\mathbf{r}-1}^{j}}$$



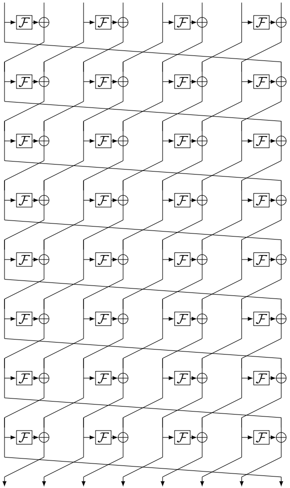
# Diffusion

- Full Diffusion Delay: minimum number of rounds $d$ for every inputs to influence every outputs



# Diffusion

- Full Diffusion Delay: minimum number of rounds $d$ for every inputs to influence every outputs
- A matrix $\mathcal{M}$ to represent the diffusion in the network:

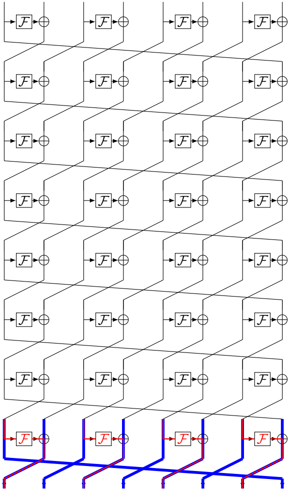


# Diffusion

- Full Diffusion Delay: minimum number of rounds $d$ for every inputs to influence every outputs

- A matrix $\mathcal{M}$ to represent the diffusion in the network:
  - 0: $B^i_{r+1}$ is influenced by $B^j_r$ directly
  - 1: $B^i_{r+1}$ is influenced by $B^j_r$ via the Feistel function $\mathcal{F}$
  - $-\infty$: not influenced (noted '.')

$$\mathcal{M} = \begin{pmatrix} 1 & 0 & . & . & . & . & . & . \\ . & . & 0 & . & . & . & . & . \\ . & . & 1 & 0 & . & . & . & . \\ . & . & . & . & 0 & . & . & . \\ . & . & . & . & 1 & 0 & . & . \\ . & . & . & . & . & . & 0 & . \\ . & . & . & . & . & . & 1 & 0 \\ 0 & . & . & . & . & . & . & . \end{pmatrix}$$



## Feistel function

- Xor with the subkey
- S-boxes: non linear
- Layers of linear functions

## Diffusion in the Feistel function

- A divide-and-conquer approach at the S-box level
- Influence of the fault on $\Delta_{B^j_{\mathbf{r}-1}}$

$$\Delta_{B^o_{\mathbf{r}}} = \mathcal{F}\left(B^i_{\mathbf{r}-1}, K^\lambda_{\mathbf{r}}\right) \oplus \mathcal{F}\left(B^{i*}_{\mathbf{r}-1}, K^\lambda_{\mathbf{r}}\right) \oplus \Delta_{B^j_{\mathbf{r}-1}}$$

- Goals:
  1. Number of pieces of subkey attacked
  2. Number of possible differences
  3. $\Rightarrow$ Number of faults required

For each block and each round where the fault can be injected:

For each block and each round where the fault can be injected:

1. Use $\mathcal{M}$ to compute:

For each block and each round where the fault can be injected:

1. Use $\mathcal{M}$ to compute:
   - $V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the penultimate round

For each block and each round where the fault can be injected:

1. Use $\mathcal{M}$ to compute:
    - $V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the penultimate round
    - $W_{\mathcal{F}} = \mathcal{M} \cdot V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the last round

For each block and each round where the fault can be injected:

1. Use $\mathcal{M}$ to compute:
   - $V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the penultimate round
   - $W_{\mathcal{F}} = \mathcal{M} \cdot V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the last round

2. Deduce the number $n_\lambda$ of blocks $K_r^\lambda$ that can be attacked

For each block and each round where the fault can be injected:

1. Use $\mathcal{M}$ to compute:
   - $V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the penultimate round
   - $W_{\mathcal{F}} = \mathcal{M} \cdot V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the last round
2. Deduce the number $n_\lambda$ of blocks $K_r^\lambda$ that can be attacked
3. Use $\mathcal{F}$ to find the number of possible differential $\Delta_{B^j_{\mathbf{r}-1}}$

For each block and each round where the fault can be injected:

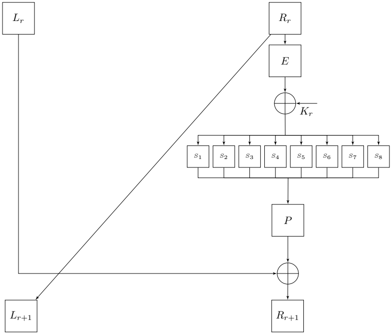
1. Use $\mathcal{M}$ to compute:
   - $V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the penultimate round
   - $W_{\mathcal{F}} = \mathcal{M} \cdot V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the last round
2. Deduce the number $n_\lambda$ of blocks $K_r^\lambda$ that can be attacked
3. Use $\mathcal{F}$ to find the number of possible differential $\Delta_{B^j_{\mathbf{r}-1}}$
4. Deduce the number $n_l$ of pieces $K_r^{\lambda,l}$ attacked

For each block and each round where the fault can be injected:

1. Use $\mathcal{M}$ to compute:
   - $V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the penultimate round
   - $W_{\mathcal{F}} = \mathcal{M} \cdot V_{\mathcal{F}}$ vector of the number of passages of the fault in the Feistel function on the last round
2. Deduce the number $n_\lambda$ of blocks $K_r^\lambda$ that can be attacked
3. Use $\mathcal{F}$ to find the number of possible differential $\Delta_{B^j_{\mathbf{r}-1}}$
4. Deduce the number $n_l$ of pieces $K_r^{\lambda,l}$ attacked
5. Estimate the number $n_J$ of faults required to attack that subkey block

# Description

- NIST standard from 1977 to 2001
- A 16-round Feistel cipher
- Starts by a 64-bit permutation $IP$ and finishes by its inverse $IP^{-1}$
- Feistel function consists in 4 steps:
  - Expansion $E$ which maps 32 bits in 48 bits by duplicating half of the bits
  - Xor with the 48 bits of subkey $K_r$
  - 8 S-boxes $6 \times 4$
  - Bit permutation $P$ of 32 bits



# Results

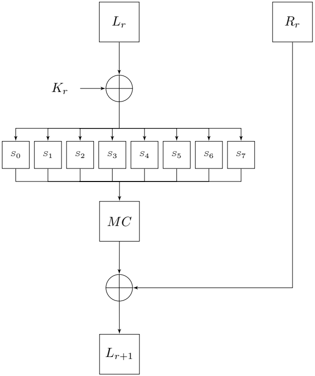
- Full diffusion delay: $d = 2$
- $\mathcal{M} = \begin{pmatrix} . & 0 \\ 0 & 1 \end{pmatrix}$
- Number of subkey blocks: $\Lambda = 1$
- Number of pieces in subkey blocks: $L = 8$
- Number of faults required to retrieve a piece of subkey: $n = 3$

Table: Results of our analysis on the DES

| Blocks $B$ | $V_{\mathcal{F}}$ | $W_{\mathcal{F}}$ | $n_l$ | $\Delta$ |
|---|---|---|---|---|
| $R_{15}$ | $(.,0)$ | $(1,0)$ | $1 \leq n_l \leq 2$ | 1 |
| $R_{14}$ | $(0,1)$ | $(2,1)$ | $2 \leq n_l \leq 8$ | 2 |
| $R_{13}$ | $(1,2)$ | $(3,2)$ | $2 \leq n_l \leq 8$ | $32 * 247$ |

# Description

- Presented at CANS'09
- A 32-round Feistel cipher
- Key of 64 or 80 bits
- Feistel function operates in 3 steps:
    - Xor with the subkey
    - 8 S-boxes $4 \times 4$
    - A linear mixing layer *MC* acting at nibble level



# Results

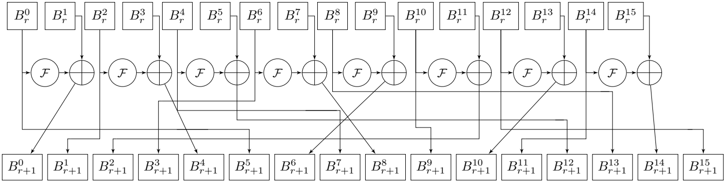
- Full diffusion delay: $d = 2$
- $\mathcal{M} = \begin{pmatrix} 1 & 0 \\ 0 & . \end{pmatrix}$
- Number of subkey blocks: $\Lambda = 1$
- Number of pieces in subkey blocks: $L = 8$
- Number of faults required to retrieve a piece of subkey: $n = 2$

Table: Results of our analysis on MIBS

| Blocks $B$ | $V_{\mathcal{F}}$ | $W_{\mathcal{F}}$ | $n_I$ | $\Delta$ |
|---|---|---|---|---|
| $L_{31}$ | $(0, .)$ | $(0, 1)$ | 1 | 1 |
| $L_{30}$ | $(1, 0)$ | $(1, 2)$ | $5 \leq n_I \leq 6$ | 4 |
| $L_{29}$ | $(2, 1)$ | $(2, 3)$ | 8 | 112 |

# Description

- 64-bit block cipher presented at SAC ’12
- GFN with 16 blocks, 4 bits each, and with 80 or 128-bit keys
- 36 rounds for both key lengths
- Feistel function used 8 times per round and consecutively made of:
  - 4-bit Xor with a subkey block
  - A single S-box $4 \times 4$



# Results

- Full diffusion delay: $d = 8$
- $$\mathcal{M} = \begin{pmatrix}
1 & 0 & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & 0 & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & 1 & 0 & . & . & . & . \\
. & . & . & . & . & . & 0 & . & . & . & . & . & . & . & . & . \\
. & . & 1 & 0 & . & . & . & . & . & . & . & . & . & . & . & . \\
0 & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & 1 & 0 & . & . & . & . & . & . \\
. & . & . & . & 0 & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & 1 & 0 & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & 0 & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & 1 & 0 & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & 0 & . \\
. & . & . & . & 1 & 0 & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & 0 & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & 1 & 0 \\
. & . & . & . & . & . & . & . & . & . & . & . & 0 & . & . & .
\end{pmatrix}$$
- Number of subkey blocks: $\Lambda = 8$
- Number of pieces in subkey blocks: $L = 1$
- Number of faults required to retrieve a piece of subkey: $n = 2$

# Results

- Full diffusion delay: $d = 8$
- Number of subkey blocks: $\Lambda = 8$
- Number of pieces in subkey blocks: $L = 1$
- Number of faults required to retrieve a piece of subkey: $n = 2$

## Summary of Results

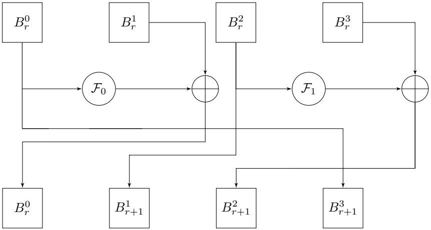
- Best case achievable: inject a fault at round 31

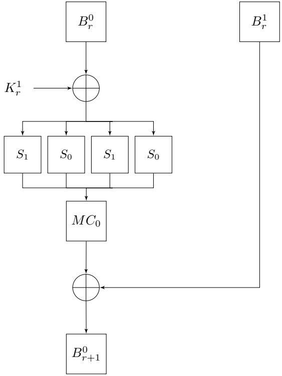
$\Rightarrow$ Attack $n_\lambda = 5$ functions (4 with non faulted $B^j_{\mathbf{r}-1}$ and one with $\#\{B^j_{\mathbf{r}-1}\} = 7$)

- If injected earlier $\Rightarrow$ at most 4 functions
- If injected after $\Rightarrow$ only up to 3 functions

# Description

- 128-bit block cipher presented at FSE '07
- Key sizes: 128, 192 or 256 bits
- Part of standard ISO/IEC 29192-2
- GFN with 4 blocks, 32 bits each
- 2 slightly different Feistel functions:
    - Xor with the subkey
    - 4 S-boxes $8 \times 8$
    - 2 linear diffusion layers, $MC_0$ and $MC_1$



# Results

- Full diffusion delay $d = 4$
- $\mathcal{M} = \begin{pmatrix} 1 & 0 & . & . \\ . & . & 0 & . \\ . & . & 1 & 0 \\ 0 & . & . & . \end{pmatrix}$
- Number of subkey blocks: $\Lambda = 2$
- Number of pieces in subkey blocks: $L = 4$
- Number of faults required to retrieve a piece of subkey: $n = 2$

# Results

- Full diffusion delay $d = 4$
- Number of subkey blocks: $\Lambda = 2$
- Number of pieces in subkey blocks: $L = 4$
- Number of faults required to retrieve a piece of subkey: $n = 2$

Table: Results of our analysis on CLEFIA

| Blocks $B$ | $V_{\mathcal{F}}$ | $W_{\mathcal{F}}$ | $n_\lambda$ | $n_l$ | $\Delta$ |
|---|---|---|---|---|---|
| $B^0_{17}$ | $(0, ., ., .)$ | $(0, 1, ., .)$ | 1 | $(1, 0)$ | $(1, -)$ |
| $B^0_{16}$ | $(1, ., ., 0)$ | $(1, 2, ., 0)$ | 1 | $(4, 0)$ | $(1, -)$ |
| $B^0_{15}$ | $(2, ., 0, 1)$ | $(2, 3, 0, 1)$ | 2 | $(4, 1)$ | $(1, \leq 127)$ |
| $B^0_{14}$ | $(3, 0, 1, 2)$ | $(3, 4, 1, 2)$ | 2 | $(4, 4)$ | $(4, \text{huge})$ |
| $B^0_{13}$ | $(4, 1, 2, 3)$ | $(4, 5, 2, 3)$ | 2 | $(4, 4)$ | $(946, \text{huge})$ |

# Conclusion

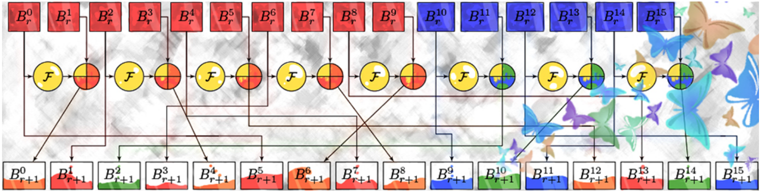
- It has been shown that some blocks are more vulnerable to DFA than others in GFNs
- A method has been proposed to identify these blocks allowing attackers to minimize single-bit fault injections
- The vulnerability evaluation is not optimal but is generic and a method to assess the vulnerabilities automatically is possible
- Further work will include multi-bit faults injection

# Acknowledgements

The authors would like to thank Ronan Lashermes and Thierry P. Berger for their valuable contributions to the development and understanding of the issues discussed in the paper.

This work was partially funded by the French DGCIS (Direction Générale de la Compétitivité de l'Industrie et des Services) through the CALISSON 2 project; and partially supported by the French National Agency of Research: ANR-11-INS-011.

Thank you for your attention




Do you have any questions ?