Protecting Last Four Rounds of CLEFIA is Not Enough Against Differential Fault Analysis

Dr. Sk. Subidh Ali

CRISSPAD
New York University Abu Dhabi

Prof. Debdeep Mukhopadhyay

Indian Institute of Technology Kharagpur
Outline

- Introduction
- CLEFIA
- Recent contribution
- Basic DFA
- DFA on CLEFIA-128
- DFA on CLEFIA-192 and CLEFIA-256
- Conclusion
CLEFIA Block Cipher

- 128 bit block cipher
- Support three key length: 128, 192, 256 bits
- 4-way Feistel structure $\text{GFN}_4,r$
- Two F-function with two S-box
- Number of rounds: 18, 22, 26
Block Diagram of CLEFIA

Encryption

Function $F_0$

Function $F_1$
Research on DFA on the CLEFIA

- **Chen et al.**
  - Byte level fault
  - 18 faulty ciphertexts
  - Fault at 17th, 16th, and 15th round

- **Fukunaga et al.**
  - Byte level fault
  - 2 faulty ciphertexts
  - Fault at 17th round
Illustration of a DFA

PLAIN TEXT

ENCRIPTION ALGORITHM

FAULT FREE CIPHER TEXT

KEY

Key Schedule

PLAIN TEXT

ENCRIPTION ALGORITHM

FAULTY CIPHER TEXT

KEY

Key Schedule

ANALYSIS

FAULT INDUCTION

10/11/2013  FDTC2013
Fault Model

- Single byte fault model
- Fault induced before the MDS operation
Chen’s Attack

- Repeatedly induce faults in \((r-1)^{\text{th}}\) round F-function.
- Get the input-output difference of \(r^{\text{th}}\) round.
- Get the \(r^{\text{th}}\) round key.
- Do one round decryption and repeat the above steps.
Flow of Faults

Use the input-output difference and get the round key
Flow of Faults

Use the input-output difference and get the round key
Protection Against Chen’s Attack

If the last two rounds are protected, the attack will fail.
Fukunaga’s Attack

- Induce two faults in 15th round F-functions.
  1. Get the input-output difference of 18th round
  2. Get the 18th round key

- Repeat step 1-2 for other round keys.
Flow of Faults
Protection Against Chen’s Attack

If the last four rounds are protected, the attack will fail.
Proposed New Attack

- A single byte fault spread to 4 bytes at MDS

\[ P => \{ p, 2p, 4p, 6p \} \]

\[ p' => \{ p', 8p', 2p', ap' \} \]
Fault Analysis

For each value of \((p, p')\) we know the input output difference of \(F_0\) and \(F_1\).
For each value of \((p, p')\) we,

- Retrieve the values of \(RK_{34}\) from the input-output difference of \(F_0\)
For each value of \((p, p')\) we,

- Retrieve the values of \(R_{K_{34}}\) from the input-output difference of \(F_0\)
- Retrieve the values of \(R_{K_{35}}\) from the input-output difference of \(F_1\)
For each value of \((p,p')\) we,
- Retrieve the values of \(\text{RK}_{34}\) from the input-output difference of \(F_0\)
- Retrieve the values of \(\text{RK}_{35}\) from the input-output difference of \(F_1\)
- Decrypt one round using \(\text{RK}_{34}\) and \(\text{RK}_{35}\)
- Retrieve the value of \(\text{RK}_{32} \oplus \text{WK}_3\) from \(F_0\)
Fault Analysis

- For each value of \((p, p')\) we,
  - Retrieve the values of \(RK_{34}\) from the input-output difference of \(F_0\)
  - Retrieve the values of \(RK_{35}\) from the input-output difference of \(F_1\)
  - Decrypt one round using \(RK_{34}\) and \(RK_{35}\)
  - Retrieve the value of \(RK_{32} \oplus WK_3\) from \(F_0\)
  - Retrieve the value of \(RK_{33} \oplus WK_2\) from \(F_1\)
Fault Analysis

- For each value of \((p, p')\) we,
  - Retrieve the values of \(RK_{34}\) from the input-output difference of \(F_0\)
  - Retrieve the values of \(RK_{35}\) from the input-output difference of \(F_1\)
  - Decrypt one round using \(RK_{34}\) and \(RK_{35}\)
  - Retrieve the value of \(RK_{32} \oplus WK_3\) from \(F_0\)
  - Retrieve the value of \(RK_{33} \oplus WK_2\) from \(F_1\)
  - Decrypt one more round and retrieve \(RK_{30}\)
For each value of \((p, p')\) we,

- Retrieve the values of RK34 from the input-output difference of \(F_0\)
- Retrieve the values of RK35 from the input-output difference of \(F_1\)
- Decrypt one round using RK34 and RK35
- Retrieve the value of RK32 \(\oplus\) WK3 from \(F_0\)
- Retrieve the value of RK33 \(\oplus\) WK2 from \(F_1\)
- Decrypt one more round and retrieve RK30
- Retrieve RK31
Fault Analysis

- For each value of \((p,p')\) we,
  - Retrieve the values of \(RK_{34}\) from the input-output difference of \(F_0\)
  - Retrieve the values of \(RK_{35}\) from the input-output difference of \(F_1\)
  - Decrypt one round using \(RK_{34}\) and \(RK_{35}\)
  - Retrieve the value of \(RK_{32} \oplus WK_3\) from \(F_0\)
  - Retrieve the value of \(RK_{33} \oplus WK_2\) from \(F_1\)
  - Decrypt one more round and retrieve \(RK_{30}\)
  - Retrieve \(RK_{31}\)

\[ RK_{34}, RK_{35} = \sum^{8} (L_2|L_3) \oplus (CON_{58}^{128}|CON_{59}^{128}) \quad \ldots (1) \]
Fault Analysis

- For each value of \((p,p')\) we,
  - Retrieve the values of \(RK_{34}\) from the input-output difference of \(F_0\)
  - Retrieve the values of \(RK_{35}\) from the input-output difference of \(F_1\)
  - Decrypt one round using \(RK_{34}\) and \(RK_{35}\)
  - Retrieve the value of \(RK_{32} \oplus WK_3\) from \(F_0\)
  - Retrieve the value of \(RK_{33} \oplus WK_2\) from \(F_1\)
  - Decrypt one more round and retrieve \(RK_{30}\)
  - Retrieve \(RK_{31}\)
  - Retrieve 57 bits of \((K_2|K_3)\) from \((RK_{34}|RK_{35})\) using inverse of \(\Sigma^8(L)\)

### Double Swap function

\[
RK_{34} | RK_{35} = \Sigma^8 (L_2 | L_3) \oplus (CON_{58}^{128} | CON_{59}^{128}) \quad \ldots (1)
\]

### Inverse Double Swap function

\[
RK_{30} | RK_{31} = \Sigma^7 (L_2 | L_3) \oplus (K_2 | K_3) \oplus (CON_{54}^{128} | CON_{55}^{128}) \quad \ldots (2)
\]
Fault Analysis

For each value of \((p, p')\) we,

- Retrieve the values of \(RK_{34}\) from the input-output difference of \(F_0\)
- Retrieve the values of \(RK_{35}\) from the input-output difference of \(F_1\)
- Decrypt one round using \(RK_{34}\) and \(RK_{35}\)
- Retrieve the value of \(RK_{32} \oplus WK_3\) from \(F_0\)
- Retrieve the value of \(RK_{33} \oplus WK_2\) from \(F_1\)
- Decrypt one more round and retrieve \(RK_{30}\)
- Retrieve \(RK_{31}\)
- Retrieve 57 bits of \((K_2|K_3)\) from \((RK_{34}|RK_{35})\) using inverse of \(\Sigma^8(L)\)
- Get \((WK_2|WK_3)\) and \(L\)
Key Recovery

- For each value of \((p,p')\) we get the value of \((K_2|K_3)\) and L
- We do the inverse GFN\(_{4,12}\) and get the value of K from L
- If the value of \((K_2|K_3)\) matches with the derived value of K (right half) we accept the key
- Only one value of K satisfy above condition
Proposed New Attack

- Induce two faults in 14\textsuperscript{th} round F-functions.
- For each value of (p,p’) do,
  1. Get the input-output difference of 18\textsuperscript{th} round
  2. Get the 18\textsuperscript{th} round key
  3. Repeat step 1-2 for other round keys.
- Get the master key and L from possible round keys.
- From L get the master key:
  - If both the master key matches accept else discard the key.
Attack Results

- The attack will work even if last four rounds are protected.
- Time complexity of the attack $2^{24}$
- Uniquely determines the master key
- Required number of faulty ciphertexts is 2
**Attack on CLEFIA-192 and CLEFIA-256**

- In case of CLEFIA-192 and CLEFIA-256 the attack is same.
- Unlike CLEFIA-128 in this case four faults are induced in (r-4)-th round in order to uniquely determine the last four round key.
- Four more faults are induced in two F-functions of (r-8)-th round to recover four more round keys.
- Last eight round keys are sufficient to retrieve the master key.
Conclusions

- We propose a attack on CLEFIA by inducing faults one round earlier.
- The attack retrieves the secret key in negligible time.
- The attack emphasize the need for protecting last five round of CLEFIA for non-iterative implementation.
Thank You