Recent works on Fault and Cache Attacks

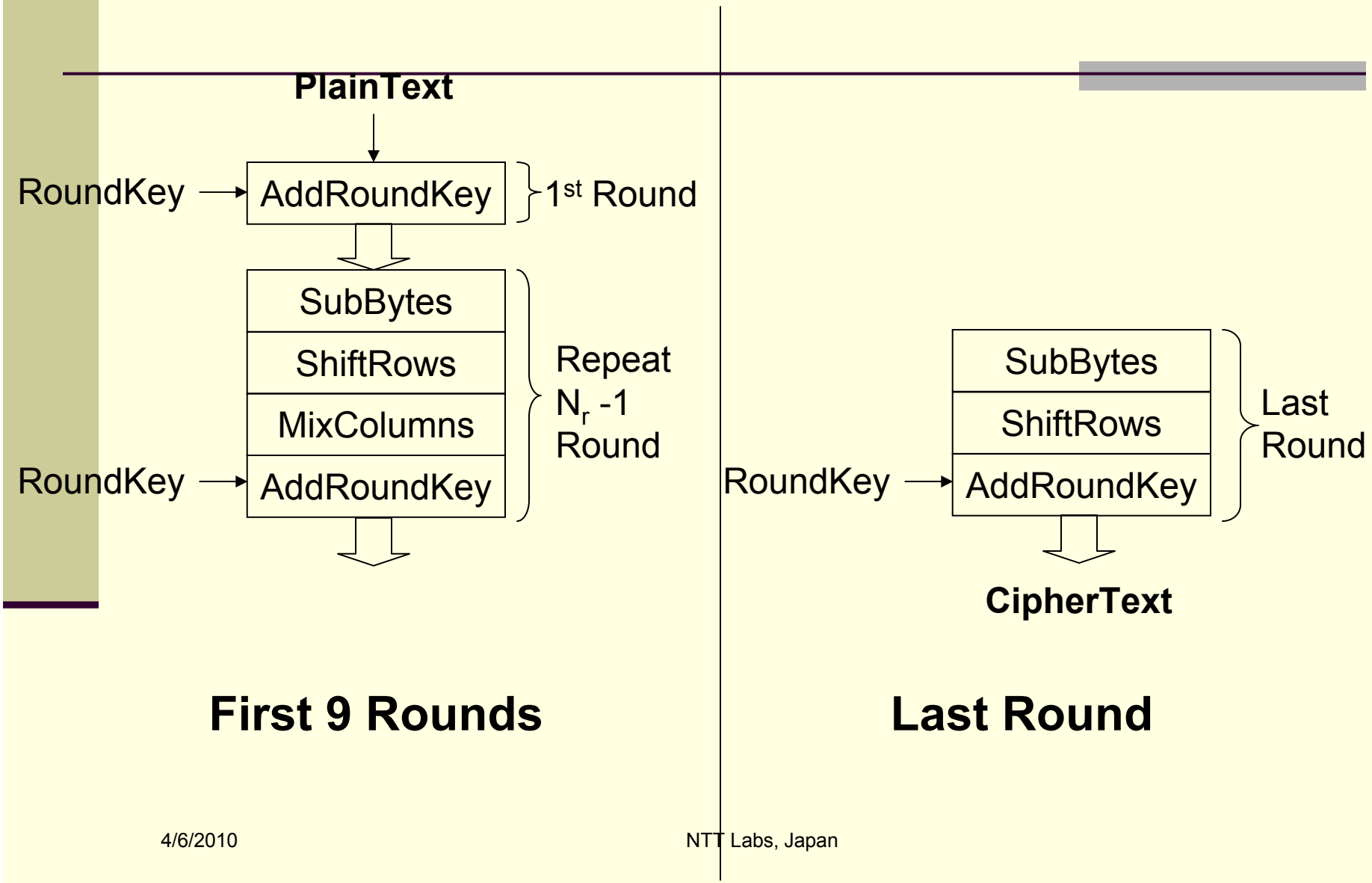
Debdeep Mukhopadhyay
Dept of CSE, IIT Kharagpur

Topics

- Multiple Byte Fault Attacks on AES
- Differential Cache Trace Attacks on CLEFIA

Multi-byte Fault Attacks on AES

AES Algorithm



Fault Model Used

- Multi Byte Faults (more practical)
  - Attacker induces fault at the input of the 8^{th} round in some bytes
  - Fault value should be non-zero but can be arbitrary
- Improves the fault coverage.

Diagonal of AES State Matrix

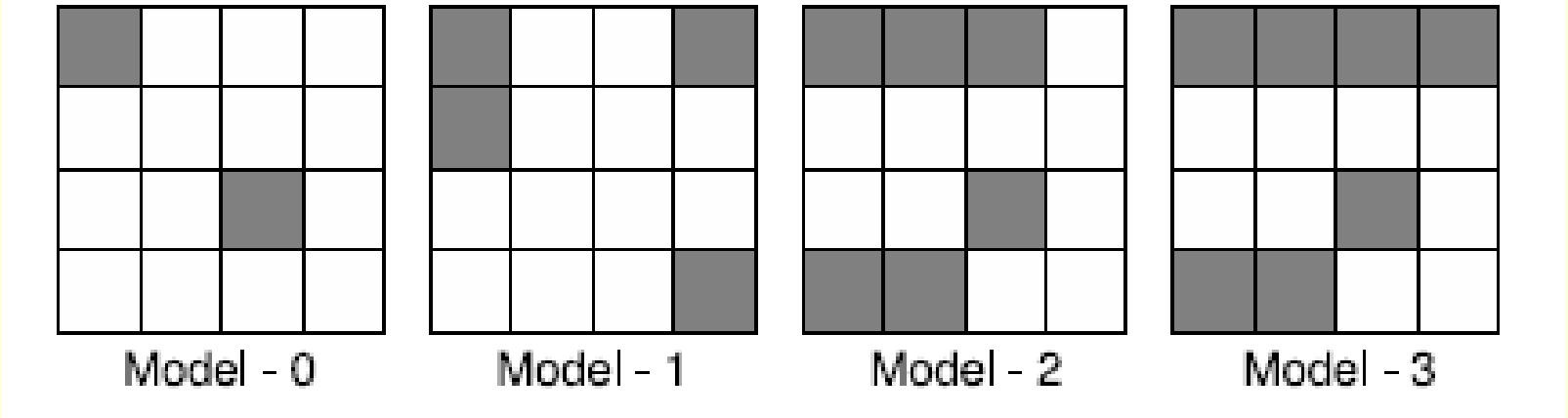
Definition 1. *Diagonal: A diagonal is a set of four bytes of the state matrix, where the i^{th} diagonal is defined as follows:*

$$D_i = \{b_{j,(j+i)mod4} \quad ; \quad 0 \leq j < 4\} \tag{1}$$

According to the above definition and with reference to the state matrix of AES (refer figure 2*) we obtain the following four diagonals.*

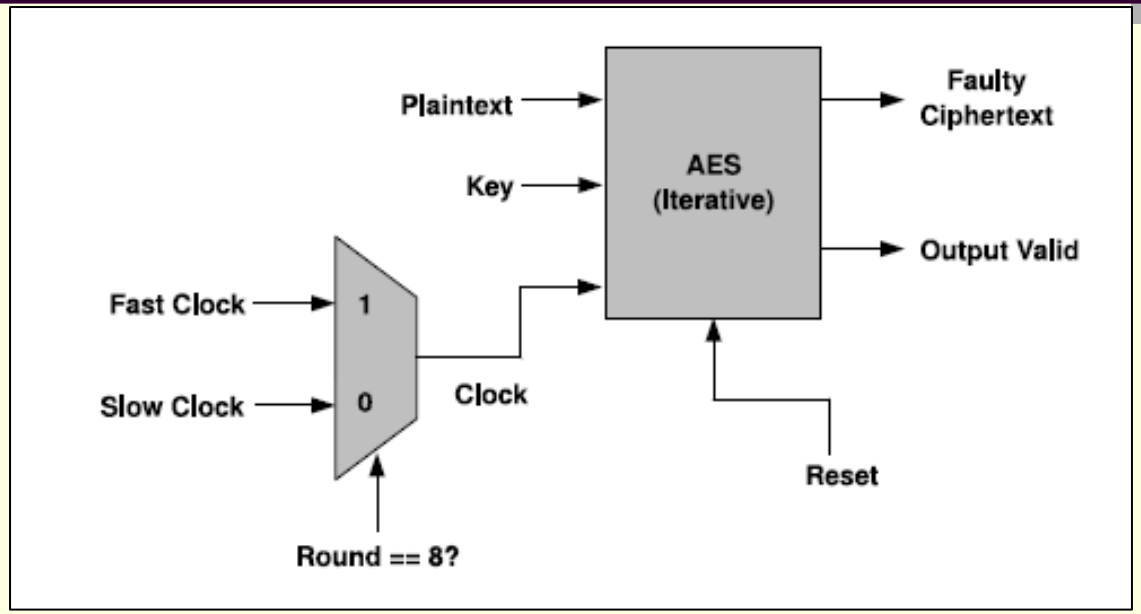
$$D_0 = (b_{00}, b_{11}, b_{22}, b_{33})$$
$$D_1 = (b_{01}, b_{12}, b_{23}, b_{30})$$
$$D_2 = (b_{02}, b_{13}, b_{20}, b_{31})$$
$$D_3 = (b_{03}, b_{10}, b_{21}, b_{32})$$

Fault Models



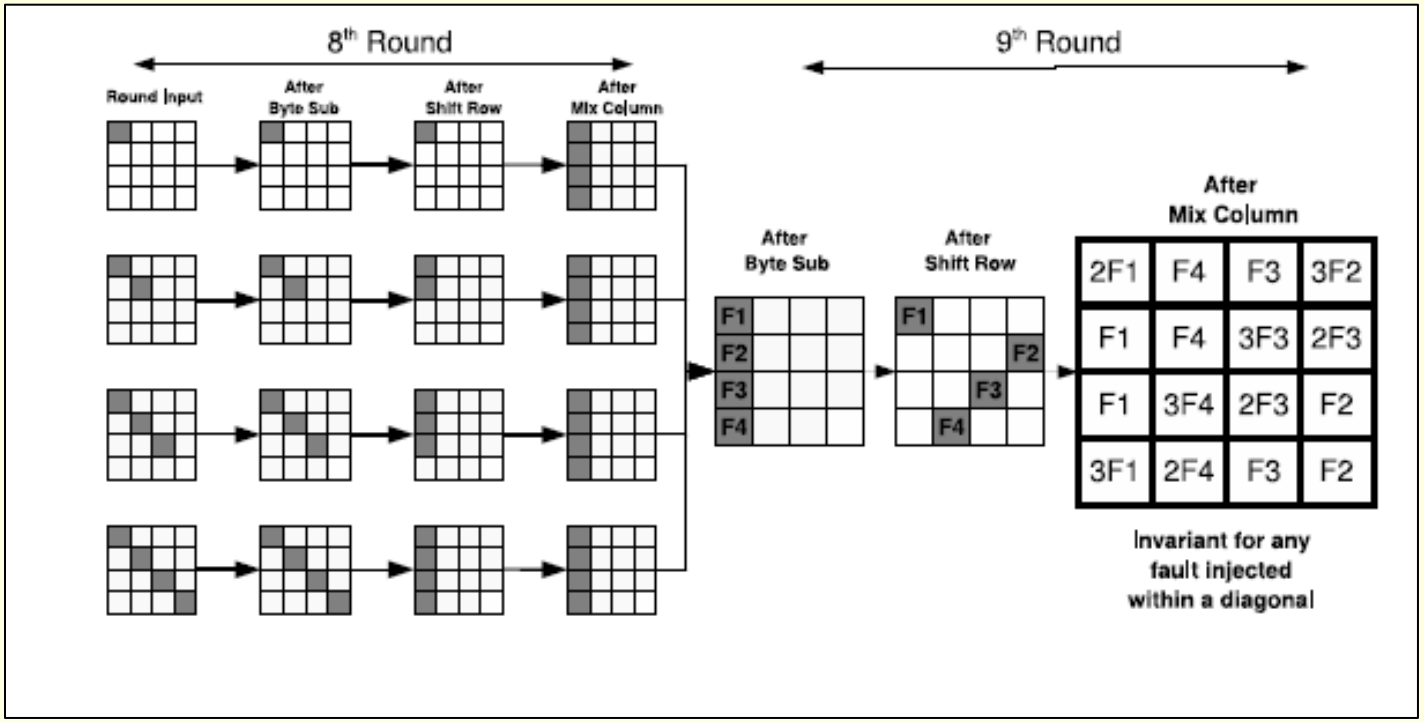
- M0: One Diagonal affected.
- M1: Two Diagonals affected.
- M2: Three Diagonals affected.
- M3: Four Diagonals affected.

Fault Injection Set Up



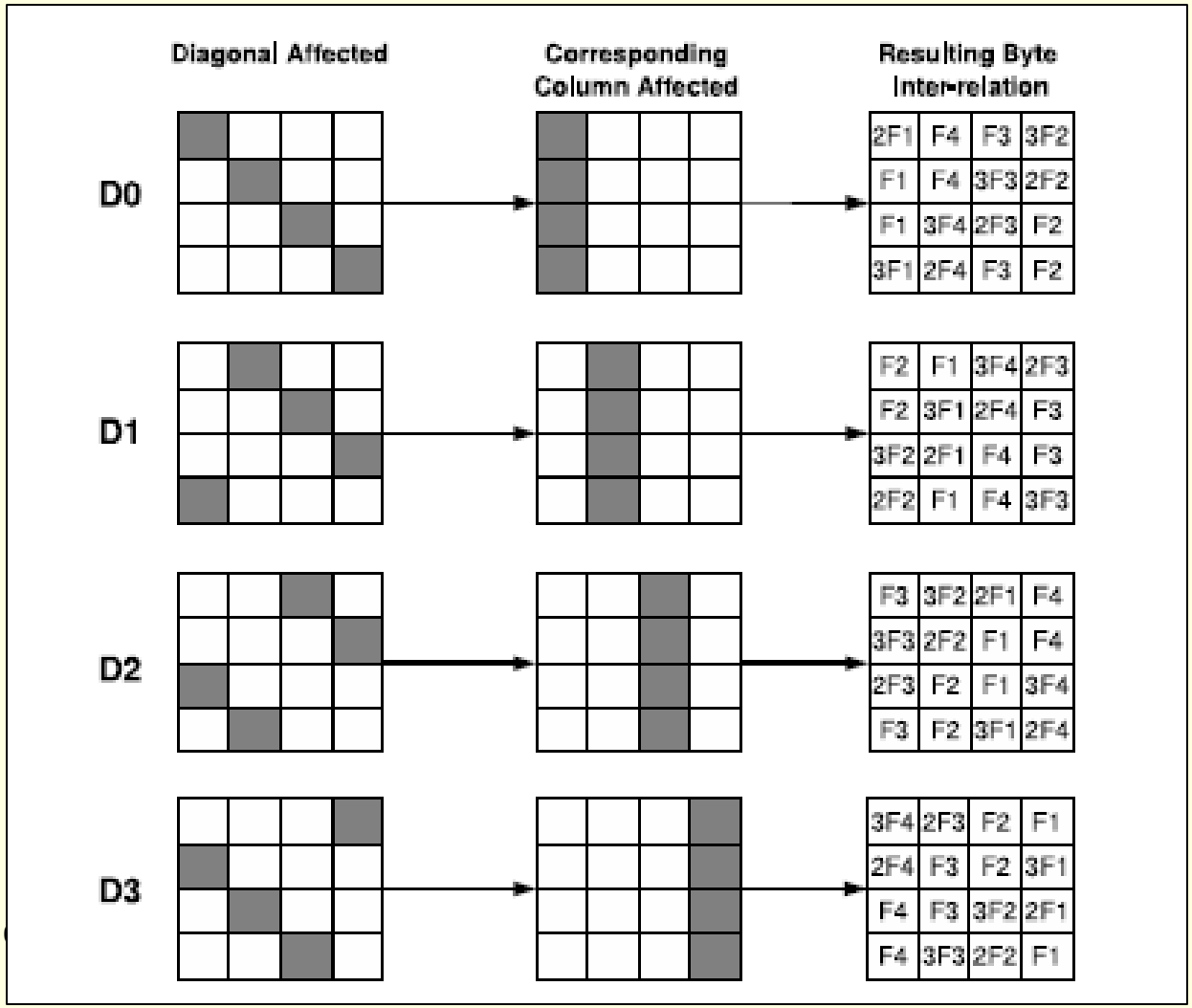
- **Tools Used:**
  - AES Core Implemented on Xilinx Spartan 3E.
  - Agilent Wavefrom (80 MHz)Generator
  - Xilinx Chipscope Pro Embedded Logic Analyzer.

Equivalence of Faults according to M0



Faults induced in Diagonal D_0 at the input of 8th round AES are all equivalent.

Inter-relationships depending on the Diagonals affected



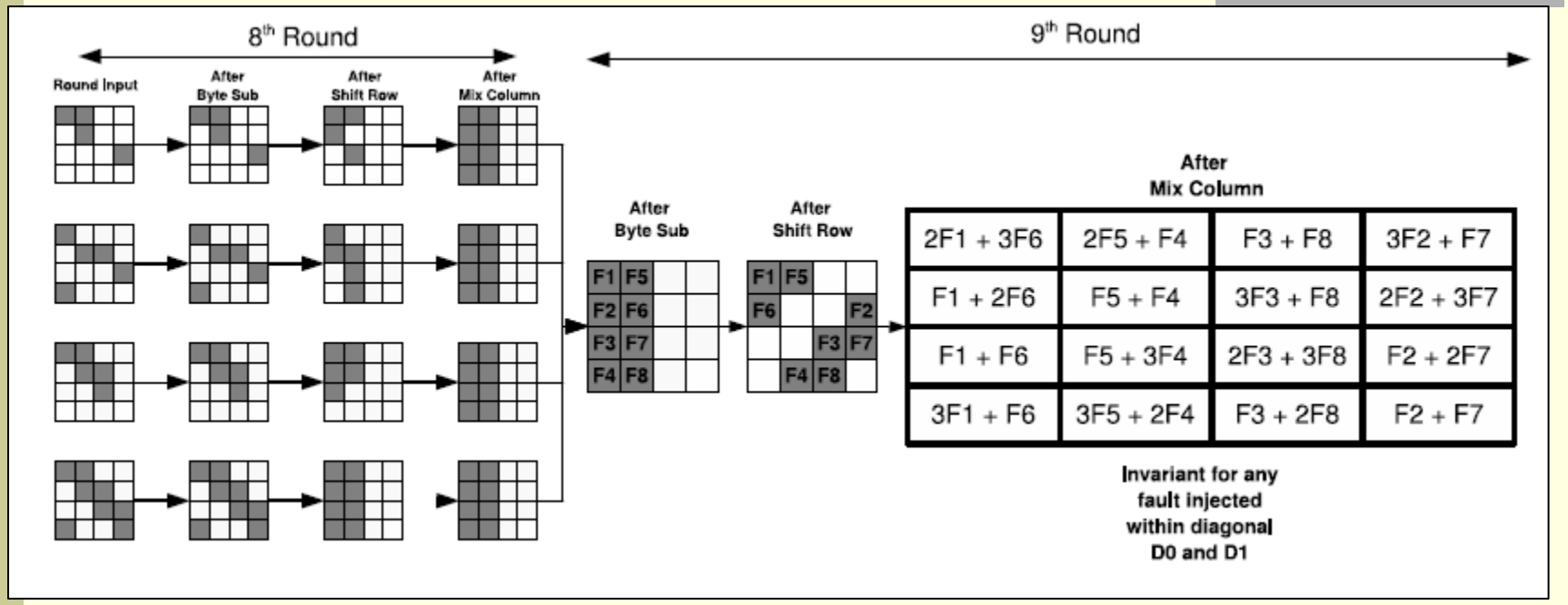
Equations if Diagonal D_0 is affected

$$\mathbf{CT} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} \end{pmatrix} \quad \mathbf{CT'} = \begin{pmatrix} x'_1 & x'_2 & x'_3 & x'_4 \\ x'_5 & x'_6 & x'_7 & x'_8 \\ x'_9 & x'_{10} & x'_{11} & x'_{12} \\ x'_{13} & x'_{14} & x'_{15} & x'_{16} \end{pmatrix} \quad \mathbf{K_{10}} = \begin{pmatrix} k_1 & k_2 & k_3 & k_4 \\ k_5 & k_6 & k_7 & k_8 \\ k_9 & k_{10} & k_{11} & k_{12} \\ k_{13} & k_{14} & k_{15} & k_{16} \end{pmatrix}$$

$$ISB(x_1 + k_1) + ISB(x'_1 + k_1) = 2[ISB(x_8 + k_8) + ISB(x'_8 + k_8)]$$
$$ISB(x_8 + k_8) + ISB(x'_8 + k_8) = ISB(x_{11} + k_{11}) + ISB(x'_{11} + k_{11})$$
$$ISB(x_{14} + k_{14}) + ISB(x'_{14} + k_{14}) = 3[ISB(x_8 + k_8) + ISB(x'_8 + k_8)]$$

- **There are in total 4 such systems of equations for a diagonal D_0.**
- **Each system of equation gives 2^8 keys on an average.**
- **AES key size gets reduced to 2^{32}.**
- **If the attacker does not know which diagonal is affected, then key size is $4.2^{32}=2^{34}$.**

Fault Injected across 2 Diagonals (Fault Model M_1)



$$
\begin{aligned}
a_0 &= 2F_1 + 3F_6 \\
a_1 &= F_1 + 2F_6 \\
a_2 &= F_1 + F_6 \\
a_3 &= 3F_1 + F_6
\end{aligned}
$$

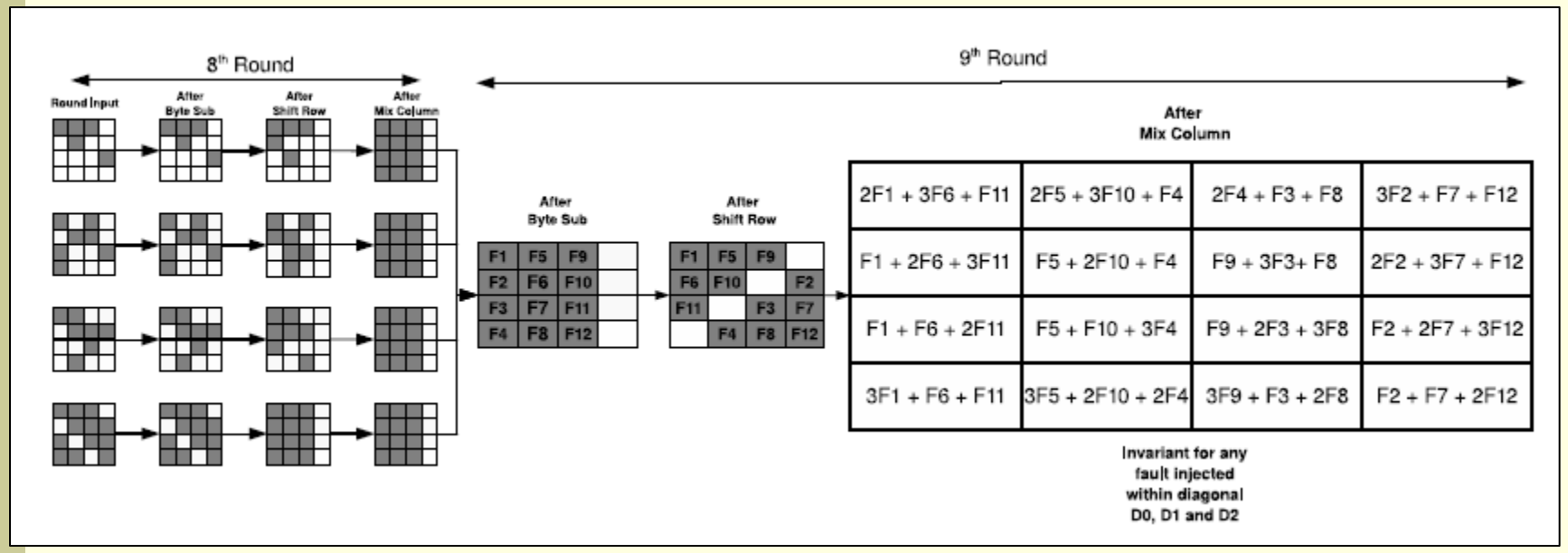
$$
\begin{aligned}
a_1 + a_3 &= a_0 \\
2a_1 + 3a_3 &= 7a_2
\end{aligned}
$$

Equations if Diagonals D_0 and D_1 are affected

$$
\begin{aligned}
a_0 &= ISB(x_1 + k_1) + ISB(x'_1 + k_1) \\
a_1 &= ISB(x_8 + k_8) + ISB(x'_8 + k_8) \\
a_2 &= ISB(x_{11} + k_{11}) + ISB(x'_{11} + k_{11}) \\
a_3 &= ISB(x_{14} + k_{14}) + ISB(x'_{14} + k_{14})
\end{aligned}
$$

- The equation reduces the space of the 4 key bytes of AES to 2^{16}
- Two faulty ciphertexts reduce it to a unique value on an average (experimental result).

Fault Injected across 3 Diagonals (Fault Model M_2)



$$
\begin{aligned}
a_0 &= 2F_1 + 3F_6 + F_{11} \\
a_1 &= F_1 + 2F_6 + 3F_{11} \\
a_2 &= F_1 + F_6 + 2F_{11} \\
a_3 &= 3F_1 + F_6 + F_{11}
\end{aligned}
$$

$$11a_0 + 13a_1 = 9a_2 + 14a_3$$

Equations if D_0, D_1 and D_2 are affected

$$
\begin{aligned}
a_0 &= ISB(x_1 + k_1) + ISB(x'_1 + k_1) \\
a_1 &= ISB(x_8 + k_8) + ISB(x'_8 + k_8) \\
a_2 &= ISB(x_{11} + k_{11}) + ISB(x'_{11} + k_{11}) \\
a_3 &= ISB(x_{14} + k_{14}) + ISB(x'_{14} + k_{14})
\end{aligned}
$$

- The equation reduces the space of the 4 key bytes of AES to 2^{24}
- Four faulty ciphertexts reduce it to a unique value on an average (experimental result).

Experimental Results

Clock Frequency (MHz)	No Fault	Model 0 (M0)	Model 1 (M1)	Model 2 (M2)	Model 3 (M3)
36.0	512	0	0	0	0
36.1	512	0	0	0	0
36.2	512	0	0	0	0
36.3	510	2	0	0	0
36.4	511	1	0	0	0
36.5	508	4	0	0	0
36.6	504	8	0	0	0
36.7	507	5	0	0	0
36.8	490	22	0	0	0
36.9	489	23	0	0	0
37.0	419	79	14	0	0
37.1	448	60	4	0	0
37.2	434	64	13	1	0
37.3	403	94	15	0	0
37.4	408	99	5	0	0
37.5	248	226	38	0	0
37.6	214	205	84	9	0
37.7	128	205	122	57	0
37.8	76	180	133	123	0
37.9	20	122	145	225	0
38.0	158	191	129	34	0
38.1	27	116	185	185	0
38.2	40	127	198	147	0
38.3	26	69	155	257	5
38.4	17	62	137	254	42
38.5	0	20	68	361	63
38.6	0	0	16	319	177
38.7	0	2	20	293	197
38.8	0	1	8	290	213
38.9	0	11	42	368	91
39.0	15	59	107	308	23
39.1	0	2	12	197	301
39.2	0	5	26	339	142
39.3	0	3	11	285	213
39.4	0	0	0	134	378
39.5	0	0	6	138	368
39.6	0	0	0	150	362
39.7	0	0	0	21	491
39.8	0	0	0	18	494
39.9	0	0	0	14	498
40.0	0	0	0	0	512

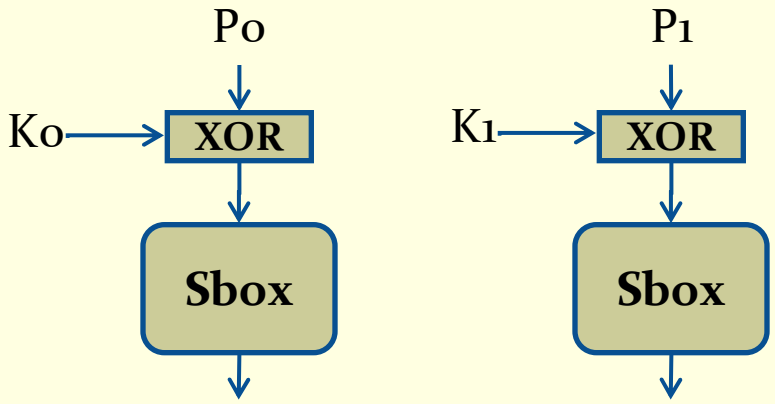
ATTACK REGION

Conclusions

- The work investigates the effect of **multiple byte** faults on AES.
- The fault modeling is based on **diagonals** being affected by random faults.
- The work **extends the coverage** of the attack compared to previous works.
- Shows **experimentally** that multiple byte faults attacks are feasible.
- **Future scope of work:** Efficient **Countermeasures** against these attacks.

Differential Trace Attacks on CLEFIA

Cache Attacks : The Principle



- Power and Time for the (P1 $\oplus$ K1) depends on the previous sbox access.
  - If there is cache hit :
    - (P0 $\oplus$ Ko) = (P1 $\oplus$ K1) =>(K0 $\oplus$ K1) = (P0 $\oplus$ P1)

    Since we know P0 and P1, we can determine (K0 $\oplus$ K1).

Classes of Cache Attacks

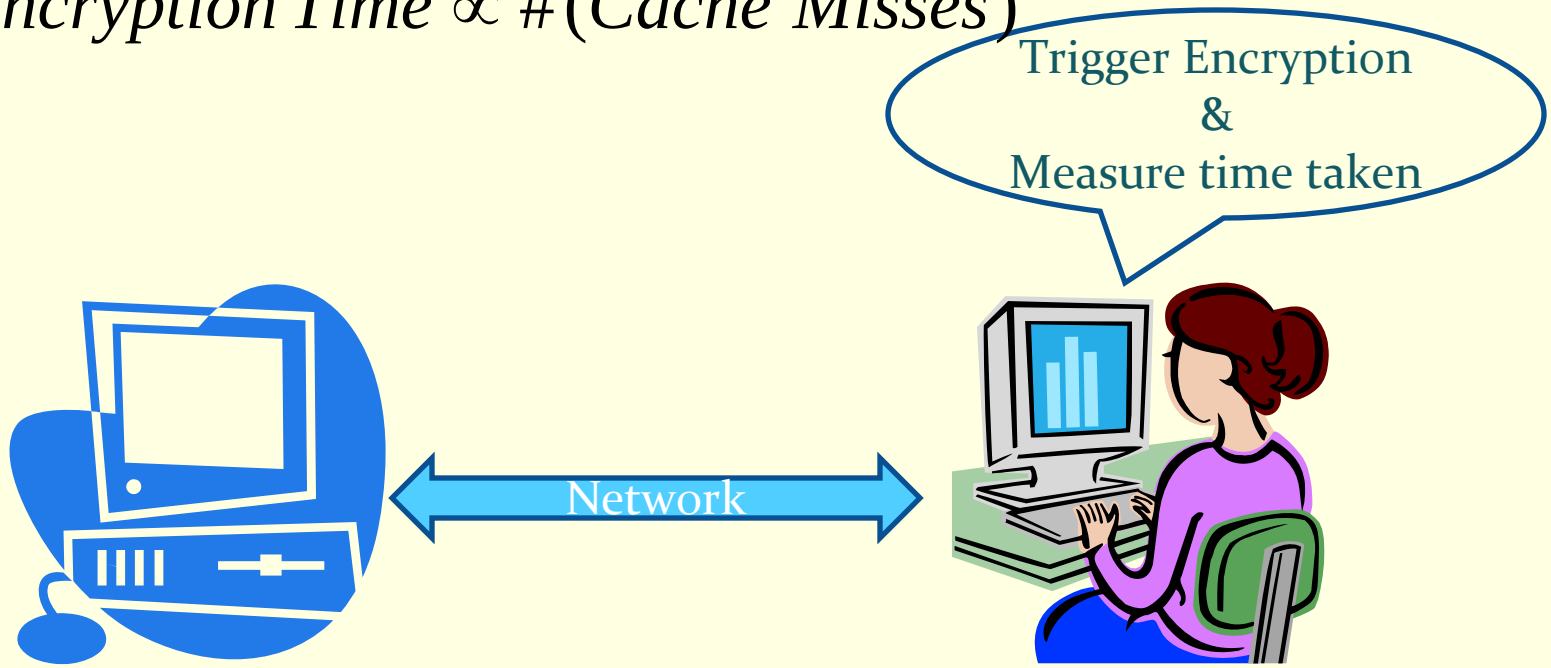
Three ways to identify cache behavior

- Cache Trace Attacks
- Cache Access Attacks
- Cache Timing Attacks

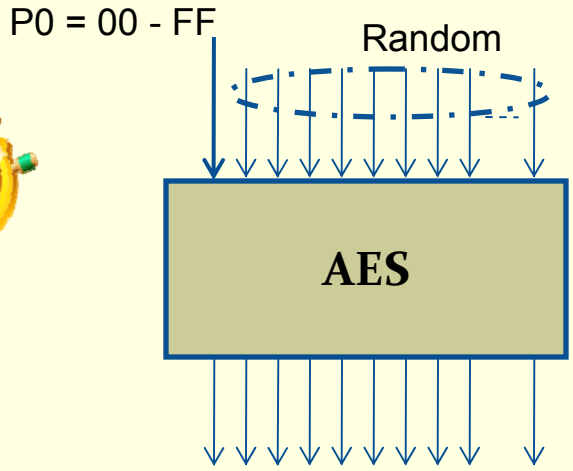
Cache Timing Attack

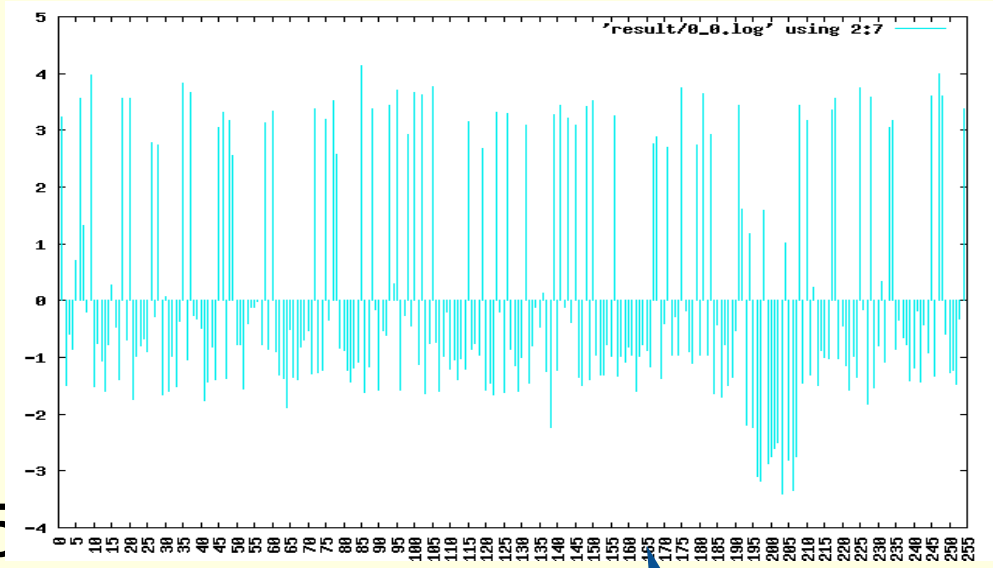
- Based on encryption time measurements.

$$Encryption\ Time \propto \#(Cache\ Misses)$$



Bernstein's Cache Timing Experiment



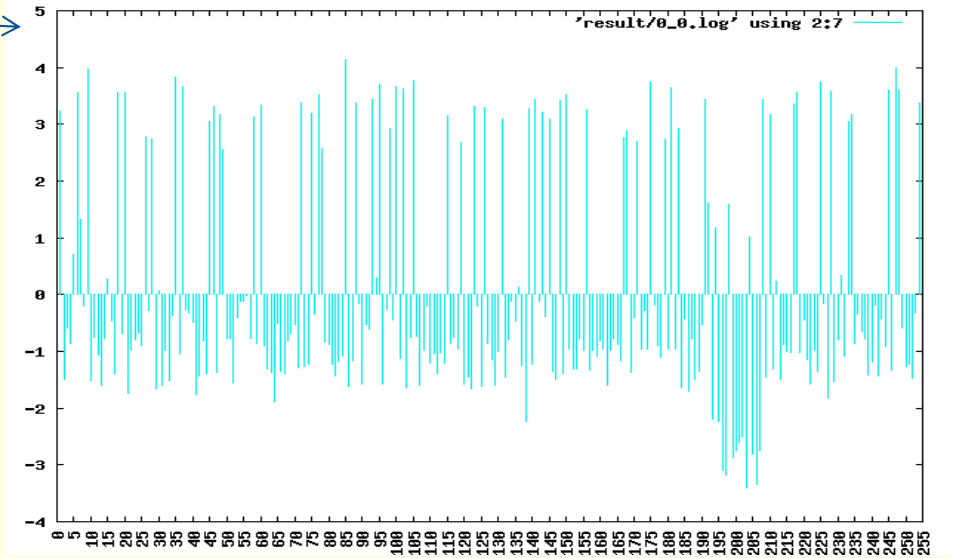


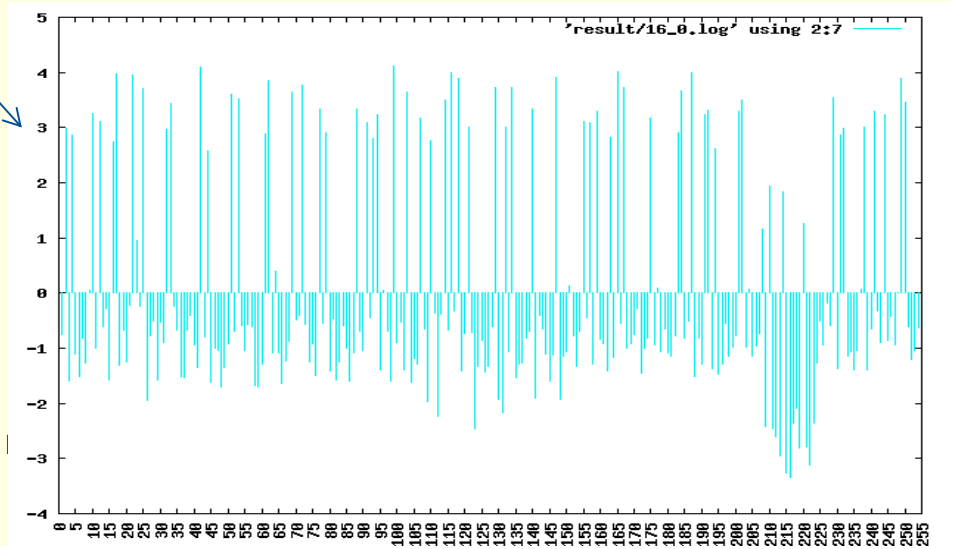
- For P0 = 0 to 255
  - For several runs
    - Vary remaining plaintext bytes randomly
    - AES Encrypt
    - Determine Time For Encryption

Timing Profile for Po

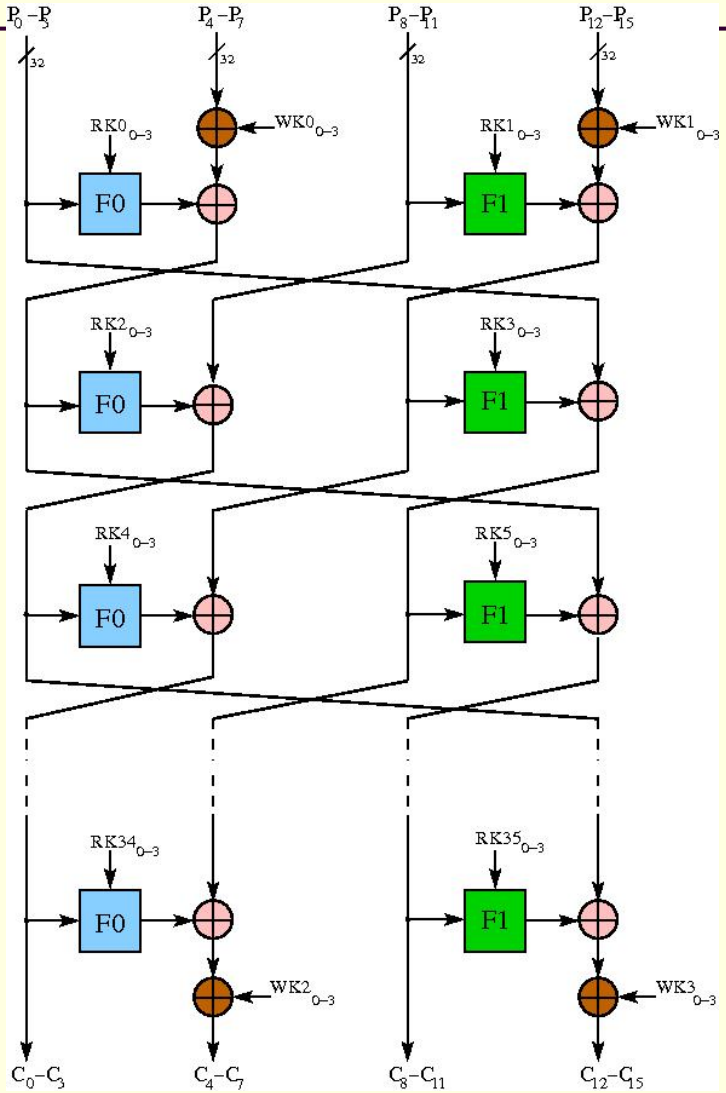
Bernstein's Cache Timing Attack

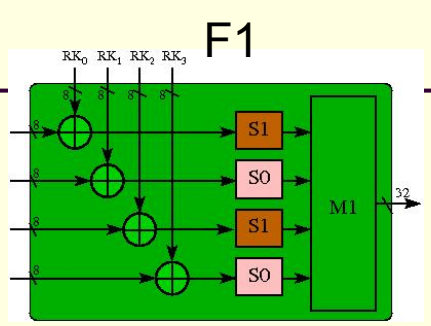
- ***Study Phase :*** Determine the timing profile of a known key.
- ***Attack Phase :*** Determine the timing profile of the unknown key.
- Correlate the two results

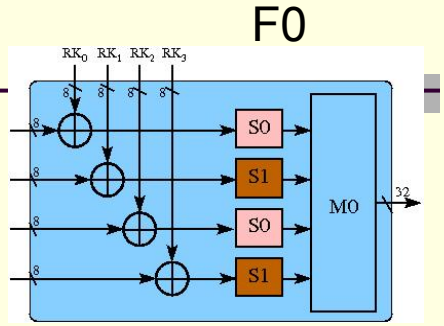




Clefia Structure







- **128 bit block cipher from Sony.**
- **Generalized Feistel Structure**
- **Number of rounds : 18**
- **Whitening Keys added at the beginning and end.**
- **Attacking Clefia requires finding any set of 4 round keys.**
  - **RK0, RK1, RK2, RK3**

NTT Labs, Japan

Timing Attack Results

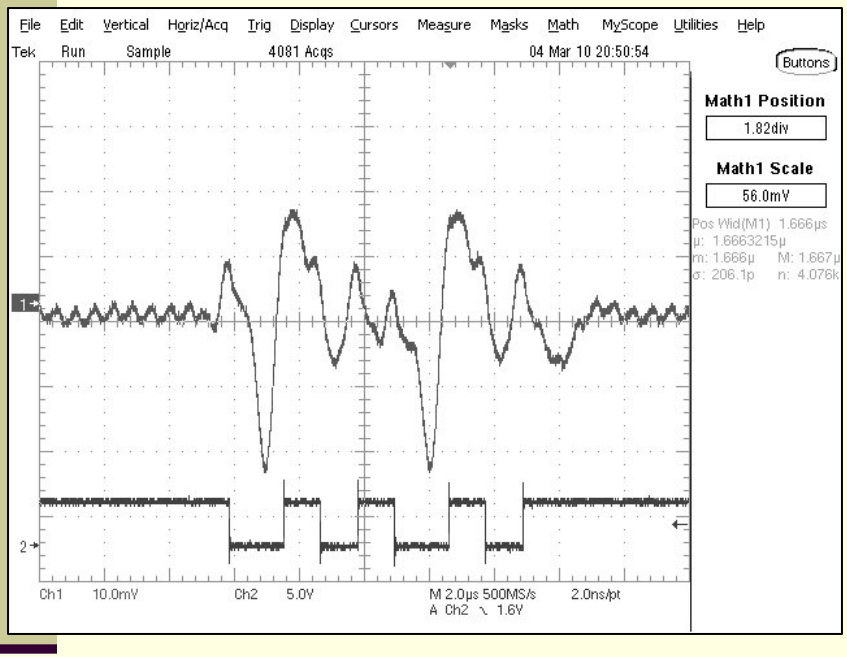
- In around 2^{26} Clefia encryptions the cipher can be shown to break in the face of cache timing attacks
- 3 GHz Intel Core 2 Duo
- 32 kB L1 Cache
- 1 GB RAM
- Linux (Ubuntu 8.04)
- gcc -4.2.4 with O3 optimization.
- Attack Time:
  - First Phase (with known key): 1300 sec
  - Second Phase (with unknown key): 312.5 sec

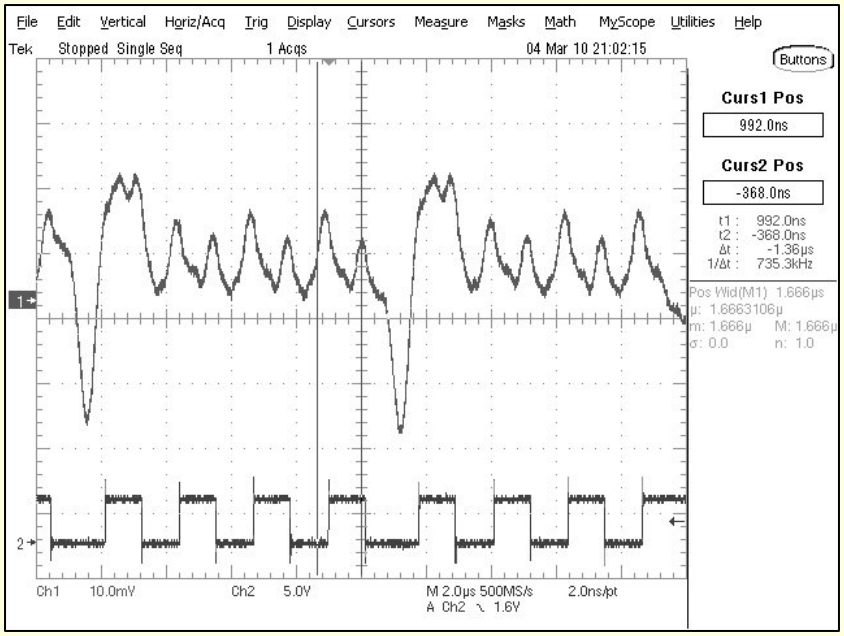
Chester Rebeiro, Debdeep Mukhopadhyay, Junko Takahashi and Toshinori Fukunaga, "Cache Timing Attacks on Clefia", In the Proceedings of Indocrypt 2009.

Trace Attacks

- The attacker has knowledge of the hits and miss patterns of the cache.
- It is a very powerful side channel.
- But the problem is how to obtain this information?
- We observed power consumptions of the device to identify the hit miss pattern.

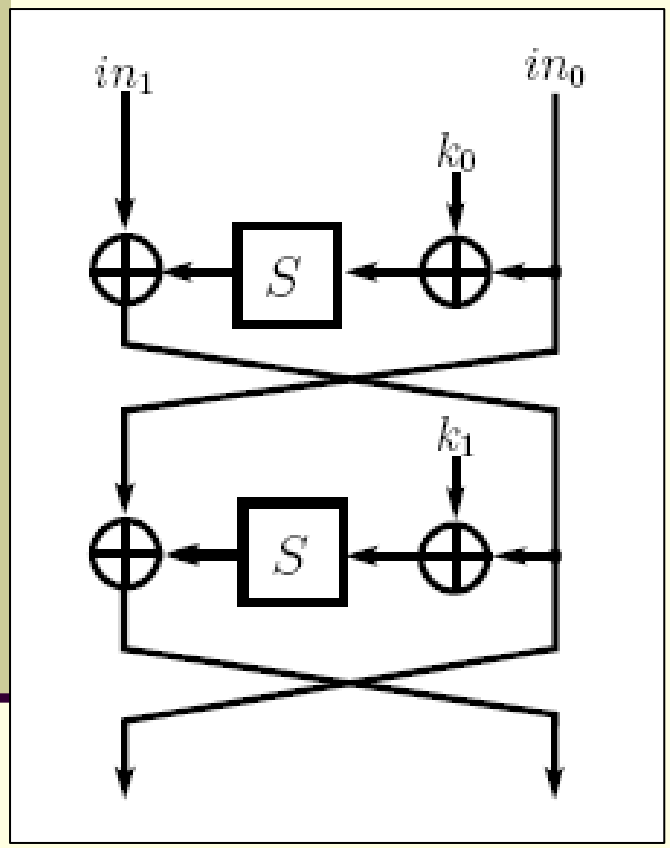
Power profiles and hit-miss patterns





- **Power Consumptions of 4 accesses to the CLEFIA S-Box, S_0.**
- **But the correspondence is not so obvious for the complete cipher.**

Concept of Differential Trace Attack



$$\langle in_0 \oplus k_0 \rangle = \langle S[in_0 \oplus k_0] \oplus in_1 \oplus k_1 \rangle$$

- Reduces the key space from 2^{2n} to $2^{n+\delta}$
- In order to reduce the key space further, we take another plaintext, resulting in a hit.
- The corresponding equation is:

$$\langle in'_0 \oplus k_0 \rangle = \langle S[in'_0 \oplus k_0] \oplus in'_1 \oplus k_1 \rangle$$

Concept of Differential Trace Attack

Combining these equations we have the following differential equation:

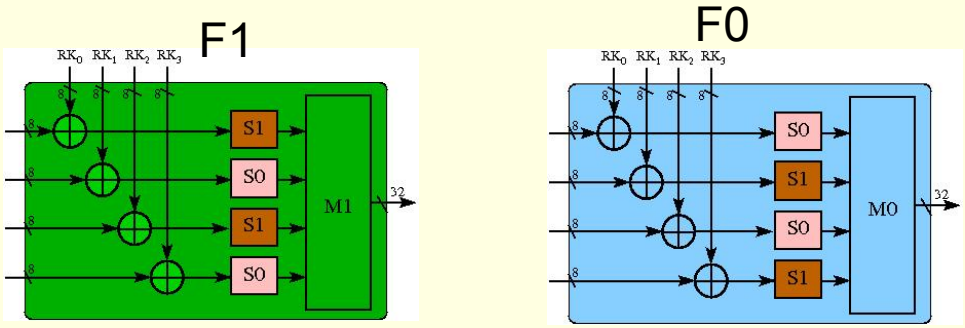
$$\langle in_0 \oplus in_1 \oplus in'_0 \oplus in'_1 \rangle = \langle S[in_0 \oplus k_0] \oplus S[in'_0 \oplus k_0] \rangle$$

The uncertainty of the key now depends on the differential property of the S-Box.

Thus, if f_{avg} is the number of keys on an average that would satisfy the above equation, then the key is reduced to:

$$N = 2^{\delta} f_{avg}$$

Adapting the Attack for CLEFIA



- **Some interesting observations:**
  - Matrices M_0 and M_1 in the F functions does not attain complete diffusion (is not diffusion optimal).
  - If 5 MSBs of the output of each S-Box are known, then 3 bits of F0 and 2 bits of F1 are computable.
  - For a differential pair, the CLEFIA S-Boxes cause 60% in S0 and 50% in S1 input output differentials to be invalid.
  - For a valid input output differential, on an average 1.28 actual values are possible for S0, while it is 1.007 for S1.

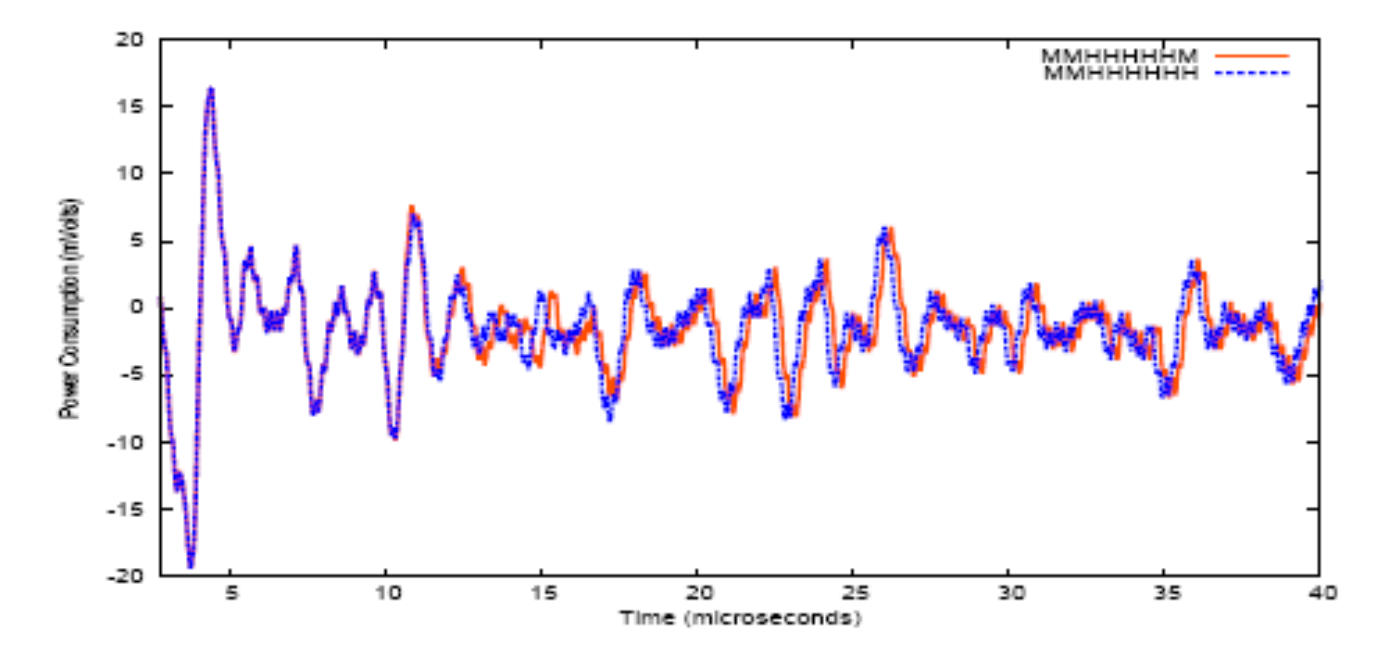
Attack on CLEFIA

- We have developed an algorithm using the above facts to obtain the key in less than 2^{14} encryptions.
- The attack employs the above properties, and the differential Cache Trace technique.
- The Cache Traces patterns are vital for the working of the attack.

Obtaining Cache Trace Patterns from the Power Profiles

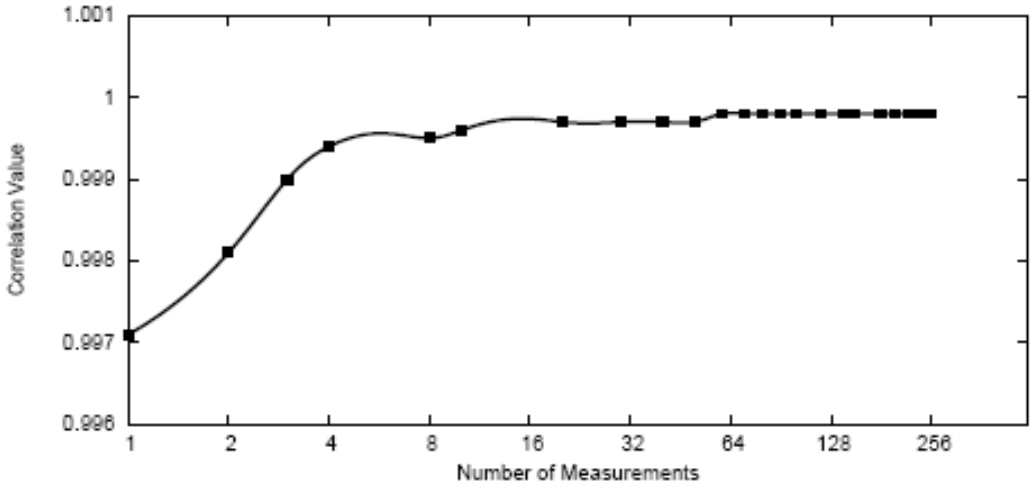
- Test Platform:
  - Xilinx XC2VP30 FPGA on the **SASEBO** side channel attack evaluation board.
  - 300 MHz PowerPC-405 core
  - 16 kB two way set associative data cache.
  - 32 kB of the FPGAs block RAM configured as the processor memory.
  - CLEFIA's reference code from SONY was run on PowerPC (http://www.sony.net/clefia)

Power Profiles for two first round access patterns



- **The difference is not so obvious as for the single S-Box seen earlier.**
- **However correlation analysis seems to pick up the small difference.**

Correlation Analysis with no. of measurements



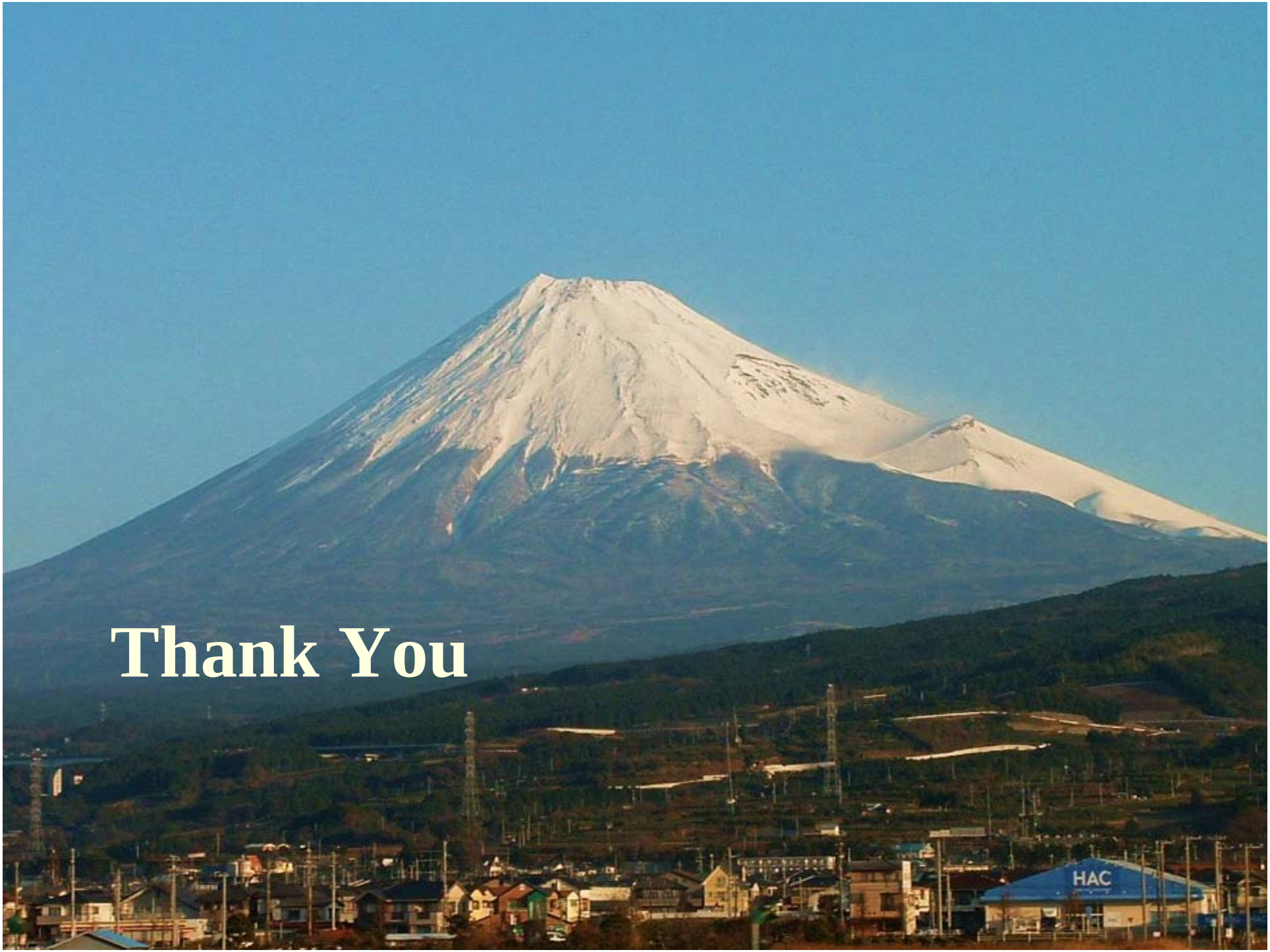
- **The power profiles for the same Hit Miss patterns show a strong correlation:**
  - It increases from 0.997 to 1 with the number of measurements (as shown above)
  - For two different patterns it is around 0.8

Classification of Hit Miss Patterns

- **This helps us to classify the Hit Miss patterns based on their power consumption:**
  - for example the first round has 64 Hit Miss patterns.
  - We were able to create 64 different power profiles, corresponding to each Hit Miss pattern
  - This classification helps to identify an unknown Hit Miss pattern from an observed power profile

Present Activities

- We have developed a counter-measure for CLEFIA to prevent the cache attacks:
  - **idea:** the entire table fits in one cache line.
- We are presently working on Formal Models for cache attacks.


Thank You
HAC