GEMPLUS
beyond smart

Round Reduction Using Faults

Hamid Choukri, Michael Tunstall
Security Technologies Department
(hamid.choukri - michael.tunstall) @gemplus.com

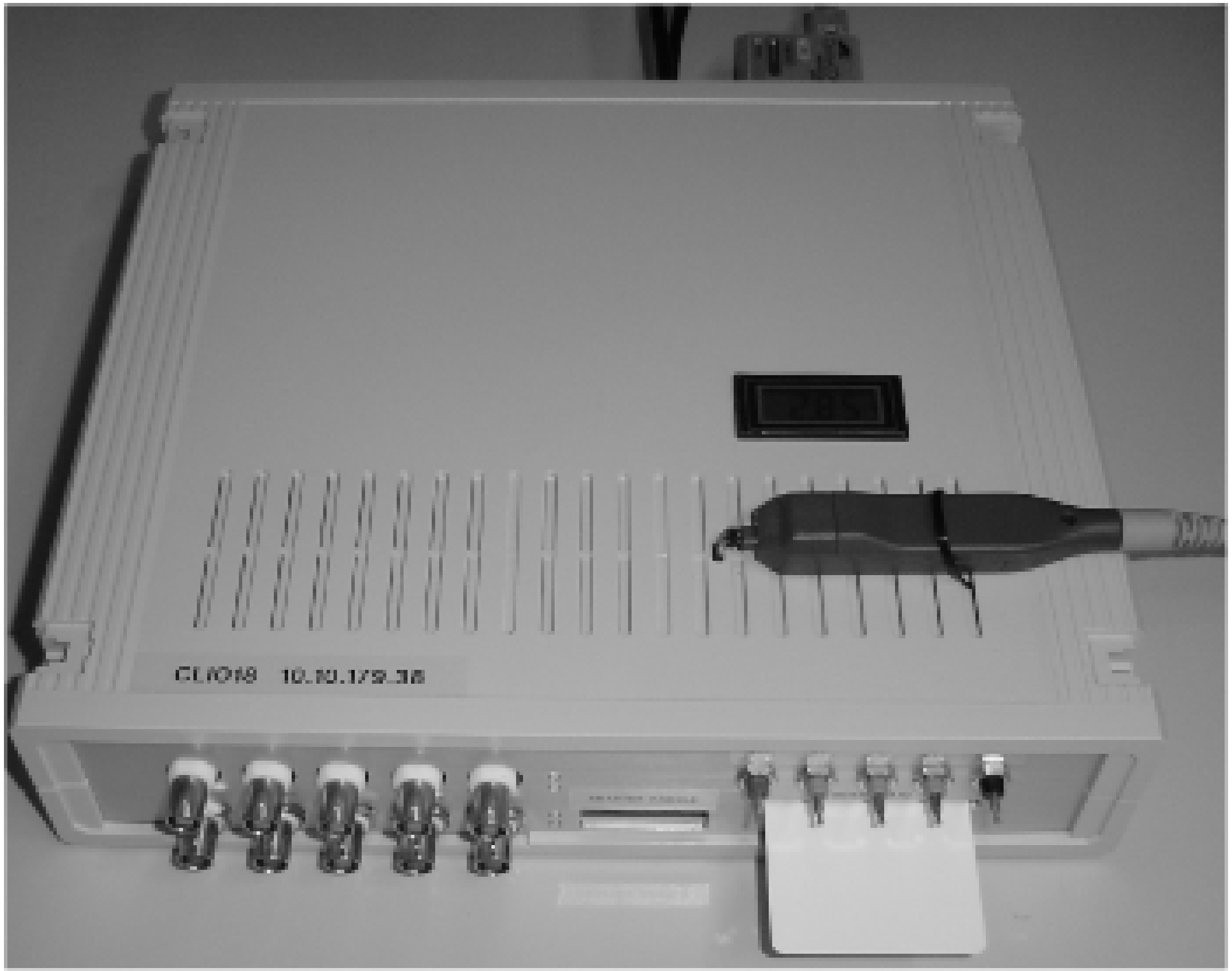
Description

- The objective
  - Break secret keys in very short time.
- The target
  - Secret key algorithms based on a function that is computed iteratively such as the DES (Data Encryption Standard) or the AES (Advanced Encryption Standard).
- The implementation
  - Naïve implementation of AES without counter measures.
- The operating mode
  - A combination of fault attack injection and a cryptanalysis.
  - The fault type is a transient glitch on Vcc (power supply)

Fault configuration

- The chip analysis and tolerance
  - Applied voltage
    - The normal voltage is 5 Volts.
    - The voltage varied from 3 volts to 5 volts.
  - External frequency
    - The normal frequency is 5 MHz
    - The frequency varied from 1 MHz to 5 MHz.
  - Glitch duration.
    - The glitch varied from 1 to 10 clock cycle
- Find optimal configuration for voltage/Frequency/Glitch

Fault Injection Equipment



Fault Target

```
    movlw      0Ah
    movwf      RoundCounter
RoundLabel:

    call       RoundFunction

    decfsz     RoundCounter
    goto       RoundLabel

    call       AddRoundKey
```

```
RoundFunction:
    call    AddRoundKey
    call    ShiftRows
    call    SubBytes
    call    MixColumns
    call    KeySchedule
    ret
```

Sensitive Locations

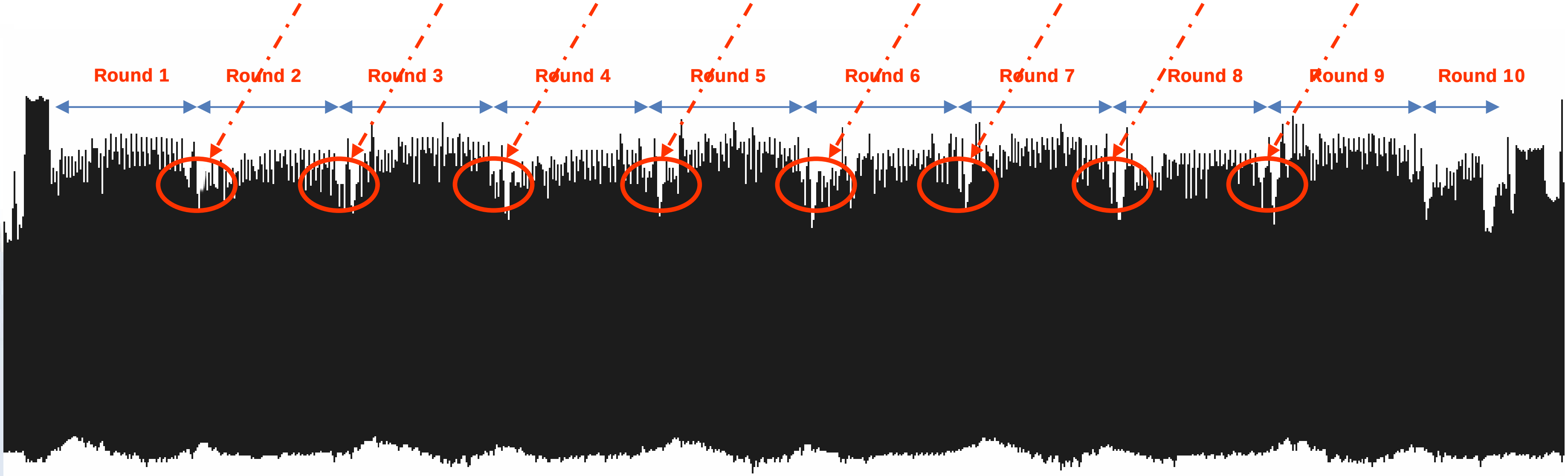
Decrement Task:
RoundCounter <= RoundCounter – 1

Testing Task:
If (RoudCounter == 0)
Status <= 1
Else
Status <= 0

Jump Task:
If (Status == 1)
PC <= PC1
Else
PC <= PC2

Processing Localization

- A naive implementation.
- Rounds are visible in the power consumption.

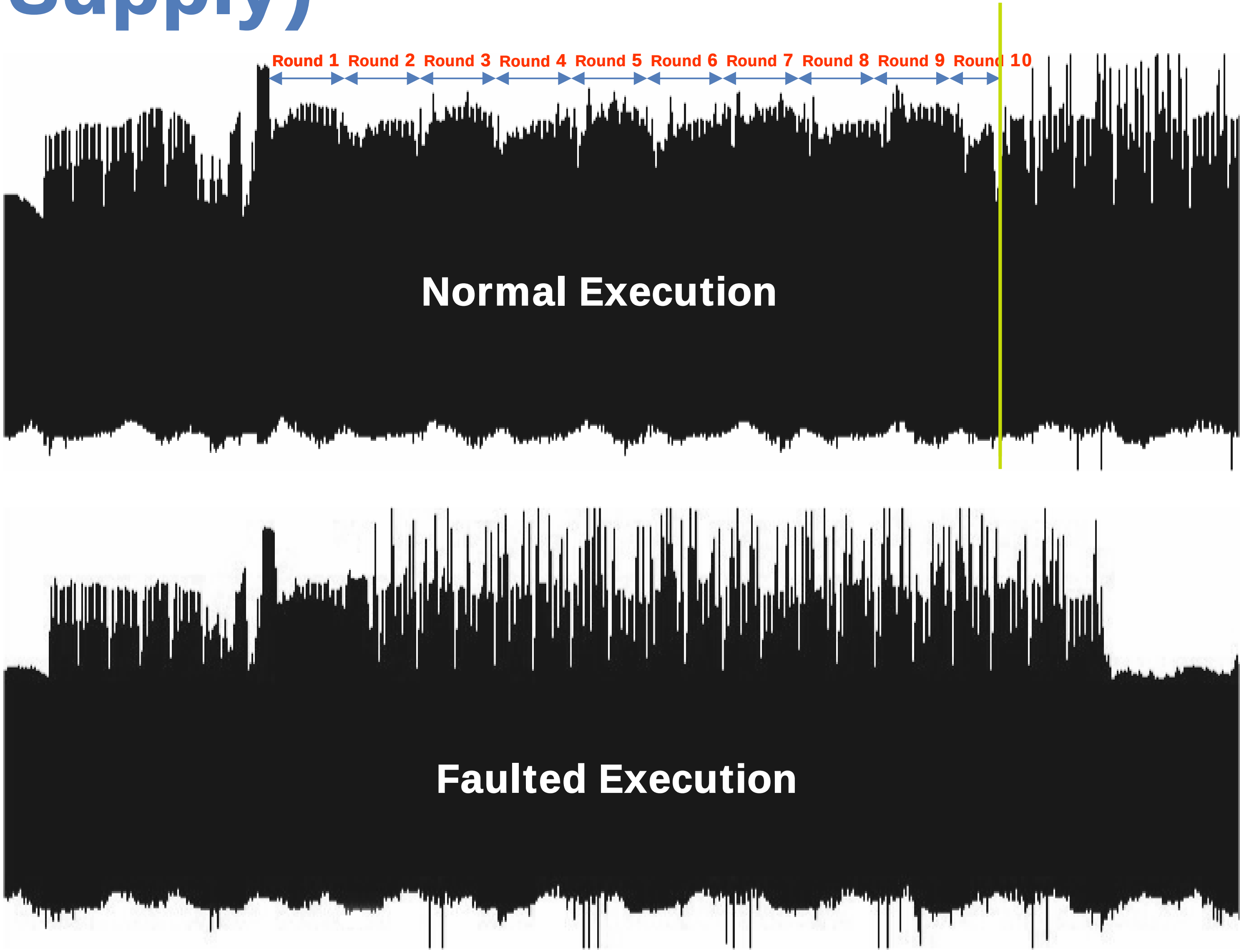


GEMPLUS

The Fault Target

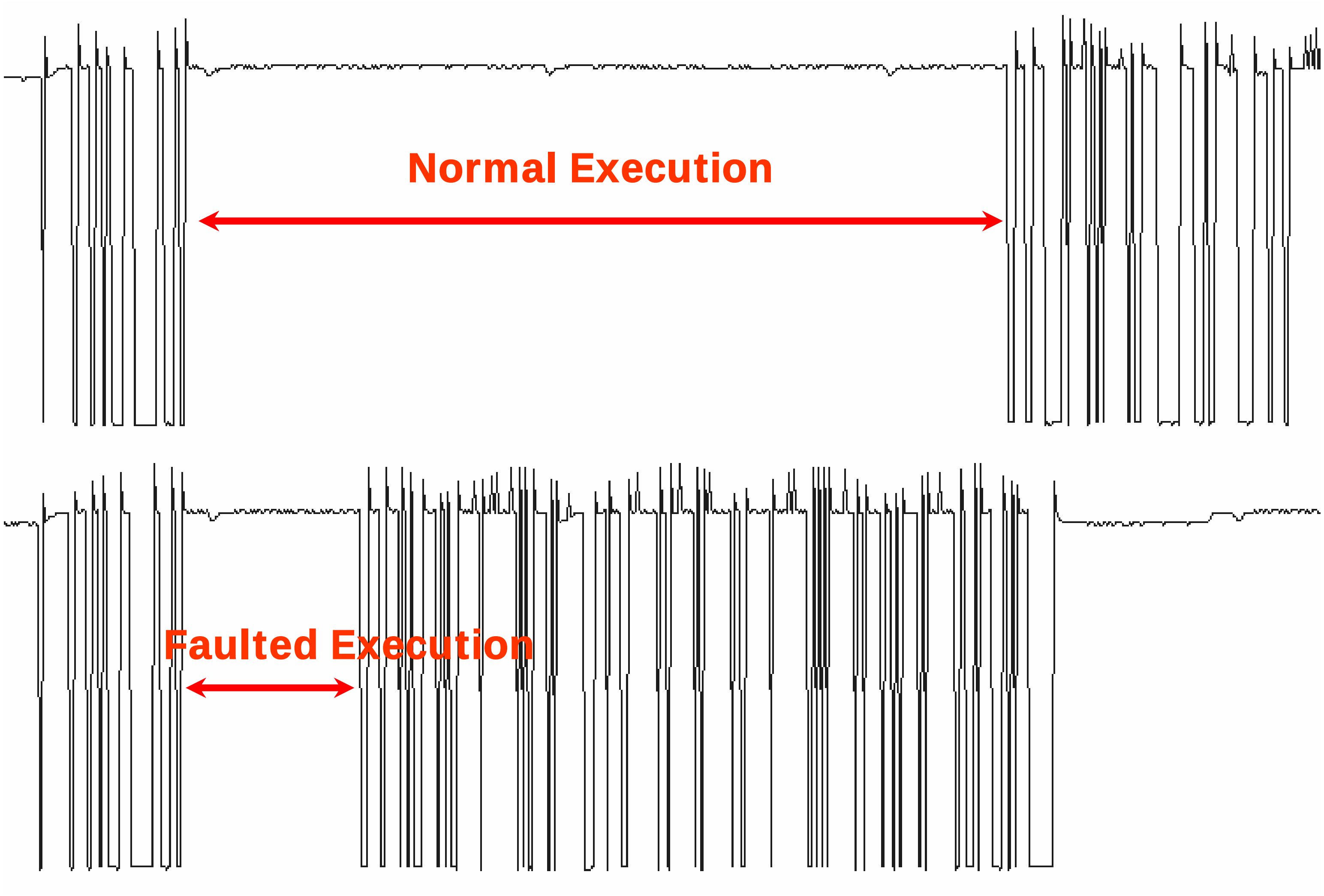
- A glitch was injected at a number of points where the end of the first round was assumed to be.
- This was done with a card with a known key to be able to detect when a successful fault occurred.
- It is also possible to be done with unknown key, but we will have the check IO time execution and the status returned by the card.

Detecting a Fault (Power Supply)



GEMPLUS

Detecting a Fault (I/O Com)



Results interpretation

- 2 faulty cipher-texts, will be:

```
AddRoundKey();
ShiftRows();
SubBytes();
MixColumns();
AddRoundKey();
```

```
AddRoundKey();
ShiftRows();
SubBytes();
AddRoundKey();
```

- Depending on the implementation

Using the Results

- With messages m_1 and m_2, producing cipher texts c_1 and c_2.
- Bytewise exhaustive search for k, in equations:

$$\mathrm{SubBytes}\,(m_1 \oplus k) \oplus \mathrm{SubBytes}\,(m_2 \oplus k) = \mathrm{MixColumn}^{-1}\,(c_1 \oplus c_2)$$

$$\mathrm{SubBytes}\,(m_1 \oplus k) \oplus \mathrm{SubBytes}\,(m_2 \oplus k) = (c_1 \oplus c_2)$$

- Each equation will give 2^{16} possible hypothesis for k.
- In our case the equation to use was known.
- A wrong fault location injection with a faulty result could be easily removed from the acquired result ($P=3.14 \times 10^{-3}$).

Other algorithms

- The attack could be applied to other secret key algorithms since the only difference is in the manner in which the result is exploited.

- As example, the DES reduction to one round give a key-space of 2^{24} to be searched from one corrupt ciphertext.

Counter measures

- Redundancy check of RoundCounter.
- Repeat all or part of the algorithm.
- Add Random delay so that it is difficult to find the correct position.
- Microcontroller with glitch sensor.
- ...

GEMPLUS

Conclusion

- The round reduction is experimentally possible in presence of naïve implementation and without hardware counter measures.
- The attack requires a high degree of control with regard to where the fault take place but relatively little calculation after acquiring the desired corrupt cipher-texts.
- Other fault attacks are possible exploiting the mathematical properties but needs more complex post-treatment.

Thank you

Contacts:

hamid.choukri@gemplus.com

michael.tunstall@gemplus.com or m.j.tunstall@rhul.ac.uk