gemalto

# Dynamic program analysis

Pierre.Girard@gemalto.com

RE-TRUST workshop
Meudon, March 19, 2009

# Mission of the day

Give an overview of tools and procedures for dynamic software analysis in an industrial security lab

# Agenda

- Introduction
  - Who, what, why and how

- Analysis
  - Static analysis
    - Software analysis
    - Hardware analysis
  - Dynamic analysis
    - Software tools
    - Input / output tools
    - Hardware  tools
  - Automatic software attack

# WHO ?

# Security labs, part of gemalto R&D

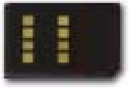
- Mission: ensure that all gemalto products reach the targeted security level
- Activities
  - Research and innovation in cryptography and security
  - Participate to standardisation
  - Security architecture and design of products, protocols, OS, applications, VM, etc…
  - Development and delivery of sensitive pieces of code (crypto. alg.)
  - Preach best practices and train other departments
  - Conduct design specification and code audits
  - Internal or external evaluation of solutions and devices
  - Support and services for customers

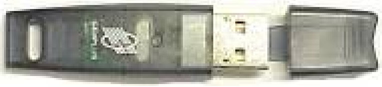
# WHAT ?

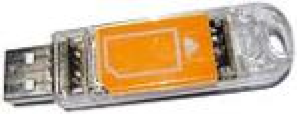
# Cards of course !

# But so many things …

UTOPIA
SAMPLE
TRAVELLER
SIMON SAMPLE
UTO
M UTOPIA
POLICE OF UTOPIA
P<UTOTRAVELLER<<SIMON<SAMPLE<<<<<<<<<<<<<<<<
12345678<8UTO6006107M1411029100660<0000<<<74







# And software, and solutions !

- Desktop PC software
- Server side software
- Operated / hosted software
- Software as a service (SaS)

# WHY ?

# Security

# Software analysis of …

- What we sell
  - Security evaluation of final products

- What we buy to build products or for internal use
  - Verify security claims of vendors
  - Compensate vulnerabilities by our software

- What the hackers produce
  - Understand the exploited vulnerabilities (cloning tools, DeSIMlocking tools, glitchers, unloopers, fake cards, etc.)
  - Hacking tools are protected against analysis !

- …

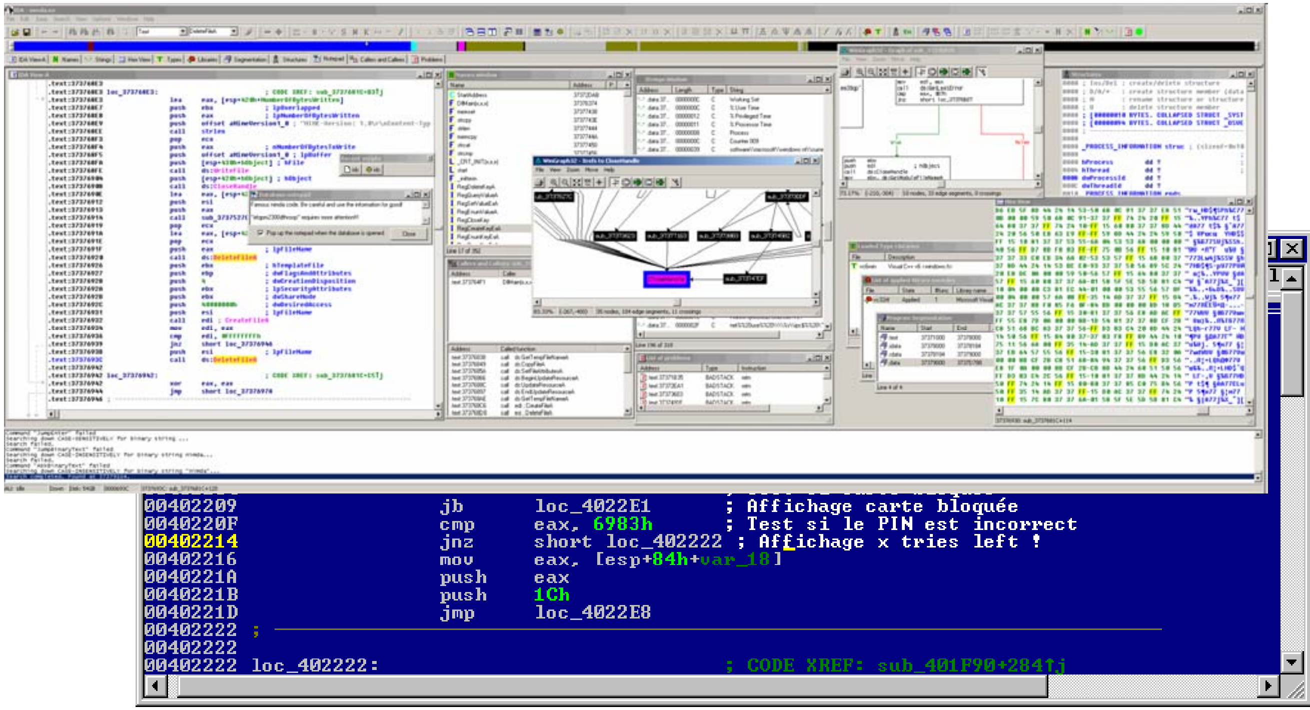
# How ?

# Several type of analysis

- Hacker like analysis (low hanging fruit, random search, creativity)
- Penetration testing (test plan, check list, etc.), CC approach
- Security validation: show that counter-measures work
- White box / black box / grey box
- Analysis interpretation : assets identification, security policy, threats, risk analysis

# Static software analysis

# Tools



```
00402209            jb      loc_4022E1       ; Affichage carte bloquée
0040220F            cmp     eax, 6983h       ; Test si le PIN est incorrect
00402214            jnz     short loc_402222 ; Affichage x tries left !
00402216            mov     eax, [esp+84h+var_18]
0040221A            push    eax
0040221B            push    1Ch
0040221D            jmp     loc_4022E8
00402222 ; ---------------------------------------------------------------
00402222
00402222 loc_402222:                         ; CODE XREF: sub_401F90+284↑j
```

- In black box
  - IDA Pro
  - JAD
- In white box
  - Source insight
  - Eclipse + plugins
- Why ?
  - Architecture overview
  - Algorithms and data analysis
  - API used
  - First security feeling (any obfuscation ?)

# Static hardware analysis

# Tools

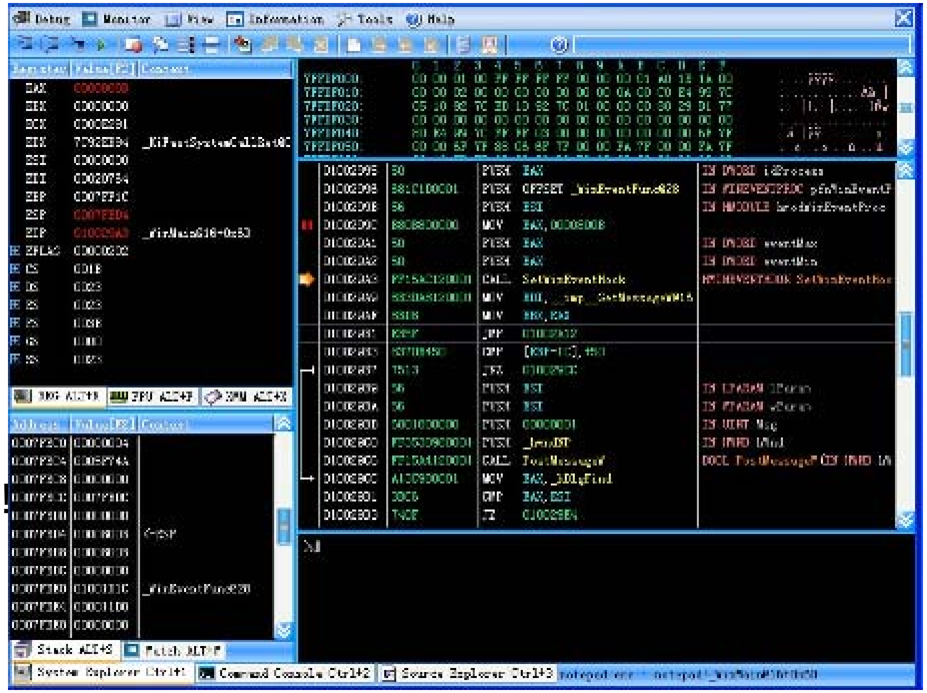
- Mechanical and chemical depackaging

- Optical microscopy

- Why
  - Architecture overview
  - Memories type and size, processor type
  - Sensors and peripherals
  - First security feeling

# Dynamic software analysis with software

# Tools

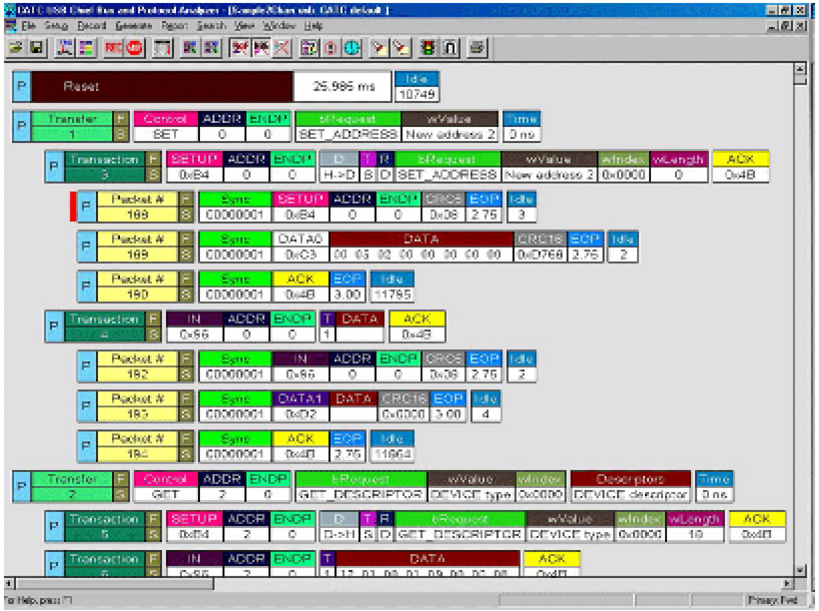


- Debuggers
  - Classic ones
  - SoftICE (unfortunately discontinued !
- Monitoring tools
  - XXMon (filemon, etc)
  - Global monitoring
- Virtualization
  - VirtualBox
  - Allow to control experimental condition an restore quickly a pristine state

# Dynamic software analysis: I/O

# Tools

- Software
  - Wireshark for IP
  - Fuzzing tools
  - Penetration test suites



```
P | Reset | 25.985 ms | Idle 10749
P | Transfer 1 | F S | Control SET | ADDR 0 | ENDP 0 | bRequest SET_ADDRESS | wValue New address 2 | Time 0 ns
  P | Transaction 3 | F S | SETUP 0xB4 | ADDR 0 | ENDP 0 | D H->D | T S | R D | bRequest SET_ADDRESS | wValue New address 2 | wIndex 0x0000 | wLength 0 | ACK 0x4B
    P | Packet # 188 | F S | Sync 00000001 | SETUP 0xB4 | ADDR 0 | ENDP 0 | CRC5 0x08 | EOP 2.75 | Idle 3
    P | Packet # 189 | F S | Sync 00000001 | DATA0 0xC3 | DATA 00 05 02 00 00 00 00 00 | CRC16 0xD768 | EOP 2.75 | Idle 2
    P | Packet # 190 | F S | Sync 00000001 | ACK 0x4B | EOP 3.00 | Idle 11785
  P | Transaction 4 | F S | IN 0x96 | ADDR 0 | ENDP 0 | T 1 | DATA | ACK 0x4B
    P | Packet # 192 | F S | Sync 00000001 | IN 0x96 | ADDR 0 | ENDP 0 | CRC5 0x08 | EOP 2.75 | Idle 2
    P | Packet # 193 | F S | Sync 00000001 | DATA1 0xD2 | DATA | CRC16 0x0000 | EOP 3.00 | Idle 4
    P | Packet # 194 | F S | Sync 00000001 | ACK 0x4B | EOP 2.75 | Idle 11864
P | Transfer 2 | F S | Control GET | ADDR 2 | ENDP 0 | bRequest GET_DESCRIPTOR | wValue DEVICE type | wIndex 0x0000 | Descriptors DEVICE descriptor | Time 0 ns
  P | Transaction 5 | F S | SETUP 0xB4 | ADDR 2 | ENDP 0 | D D->H | T S | R D | bRequest GET_DESCRIPTOR | wValue DEVICE type | wIndex 0x0000 | wLength 18 | ACK 0x4B
  P | Transaction 6 | F S | IN 0x96 | ADDR 2 | ENDP 0 | T 1 | DATA 12 01 00 01 09 00 00 08 | ACK 0x4B
```

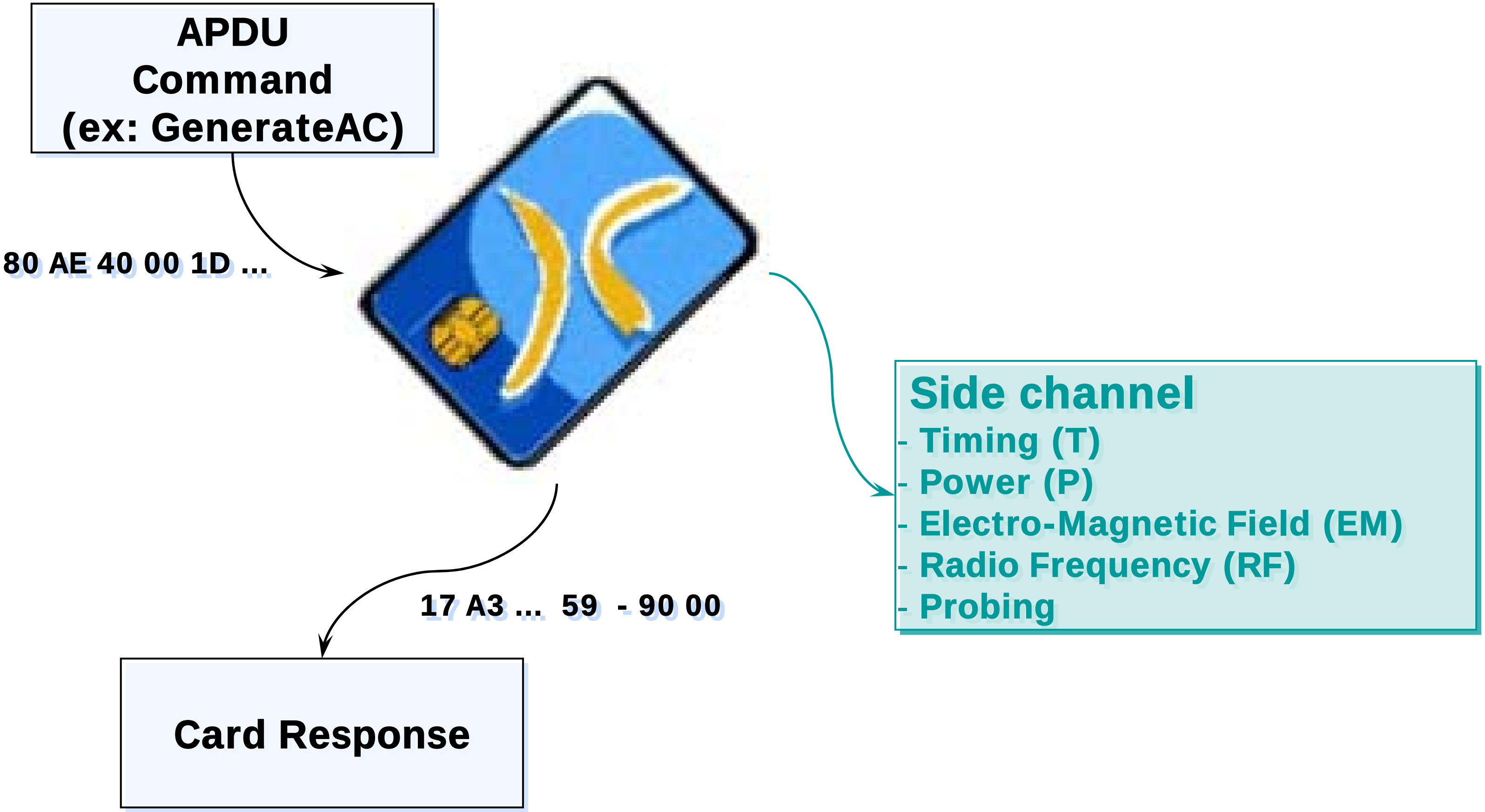
- Hard ware
  - USB chief for USB
  - Proprietary for APDU
  - Proxylab for contactless

# I/O analysis goals

- Traffic analysis
  - Protocols analysis
  - Identify security building block: encryption, randomness, challenge response, message integrity, etc…
- Fuzzing, penetration tools
  - Characterize behaviour and protections
  - Find sensible areas to explore latter on with hand crafted attacks
  - Find directly vulnerabilities

# Dynamic software analysis with hardware

# Side-channel



APDU Command (ex: GenerateAC)

80 AE 40 00 1D ...

17 A3 ... 59 - 90 00

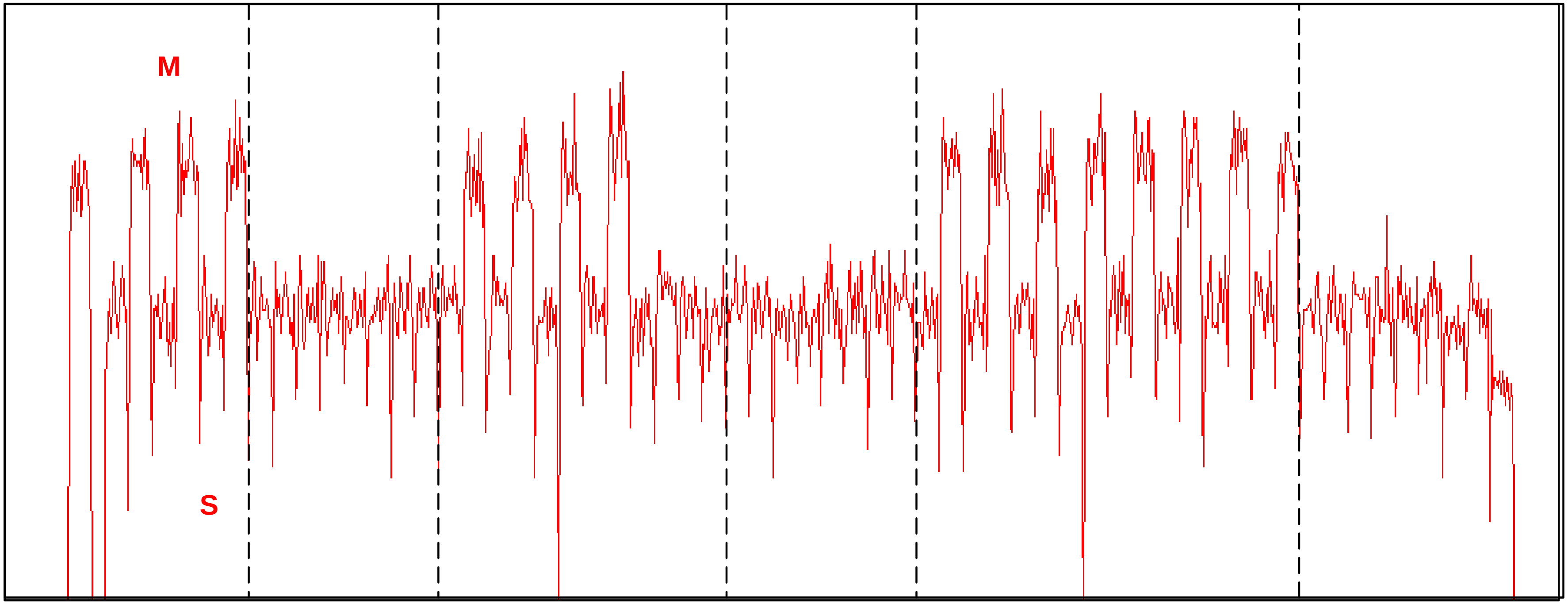
Card Response

**Side channel**

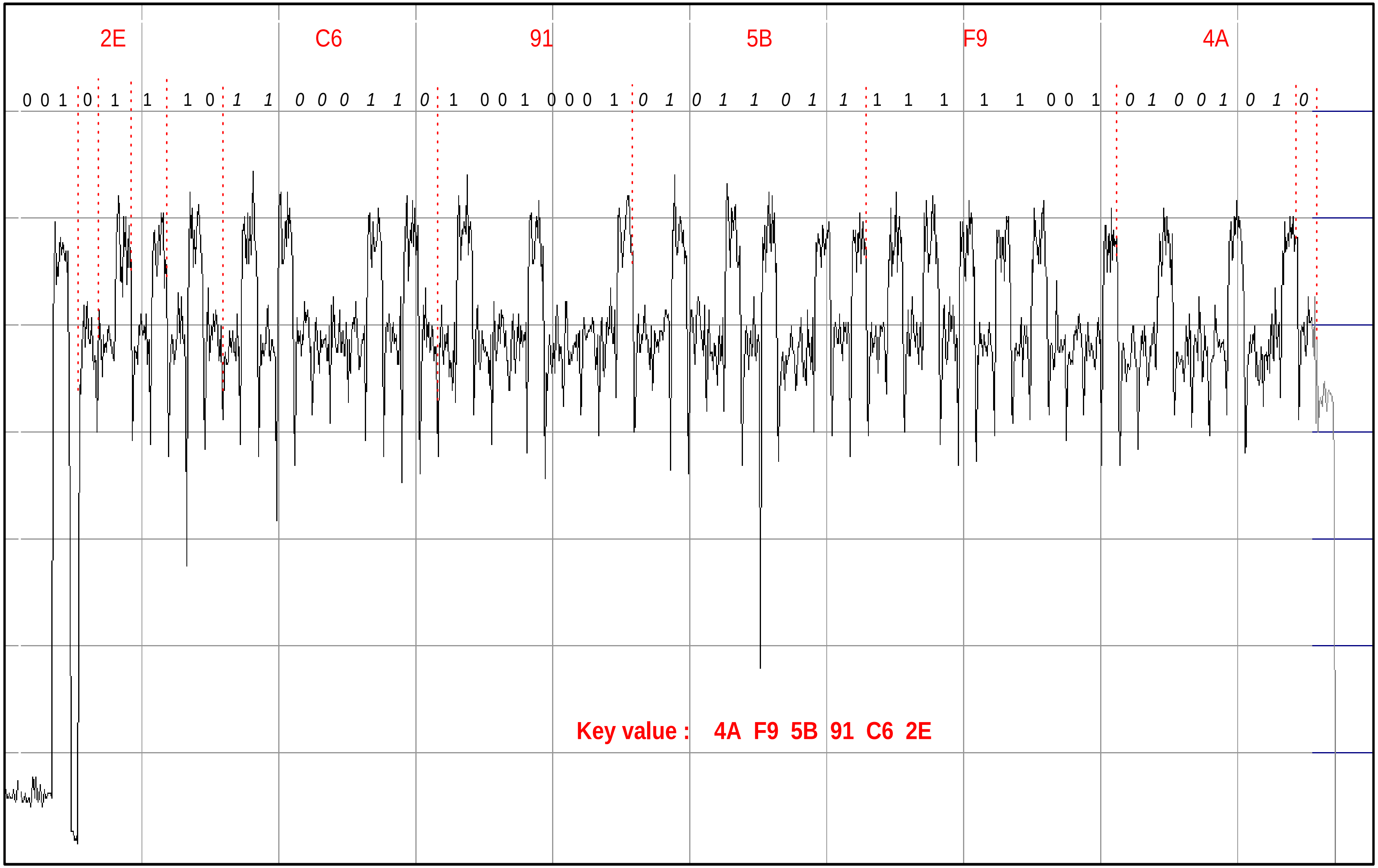
- Timing (T)
- Power (P)
- Electro-Magnetic Field (EM)
- Radio Frequency (RF)
- Probing

# RSA attack - reference key

Key value : 00 FF 00 F0 00 0F

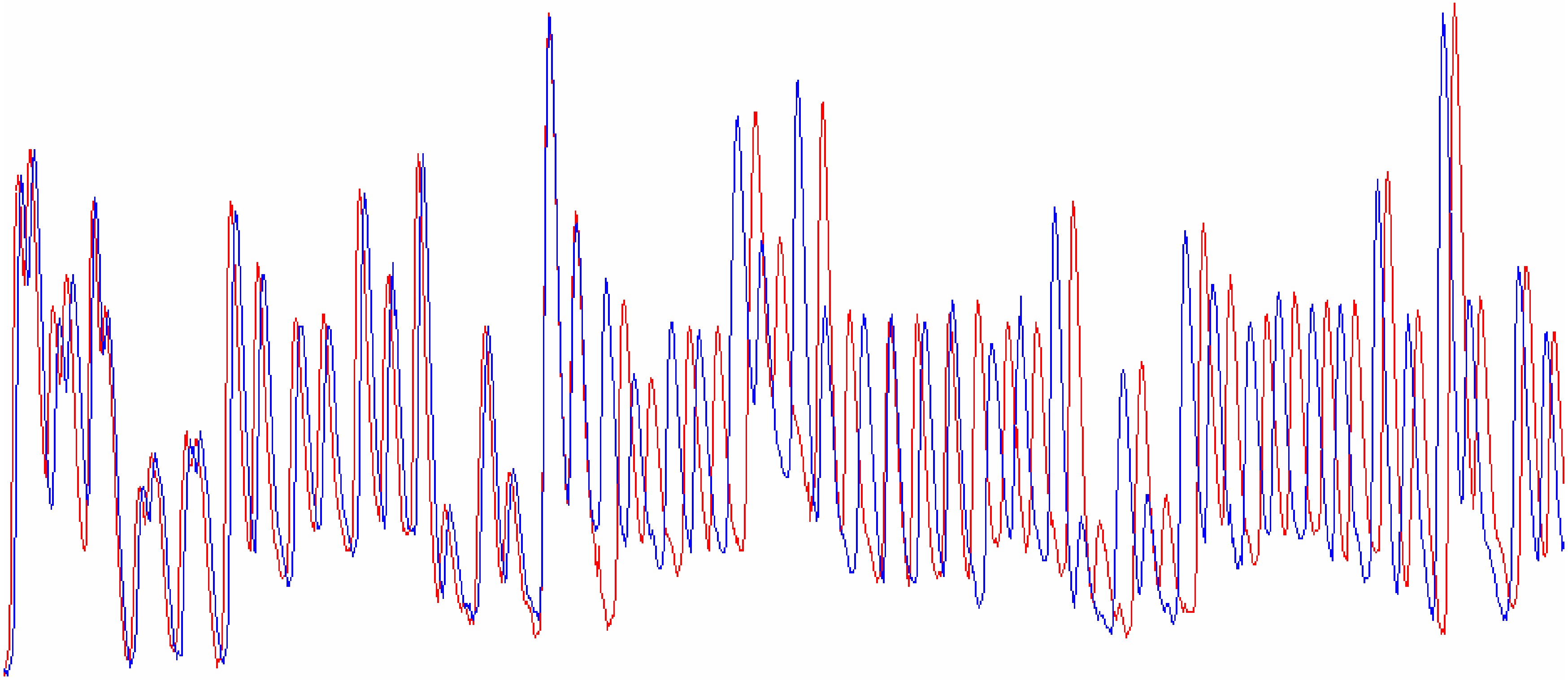


# RSA attack - secret key



# Automatic software attacks

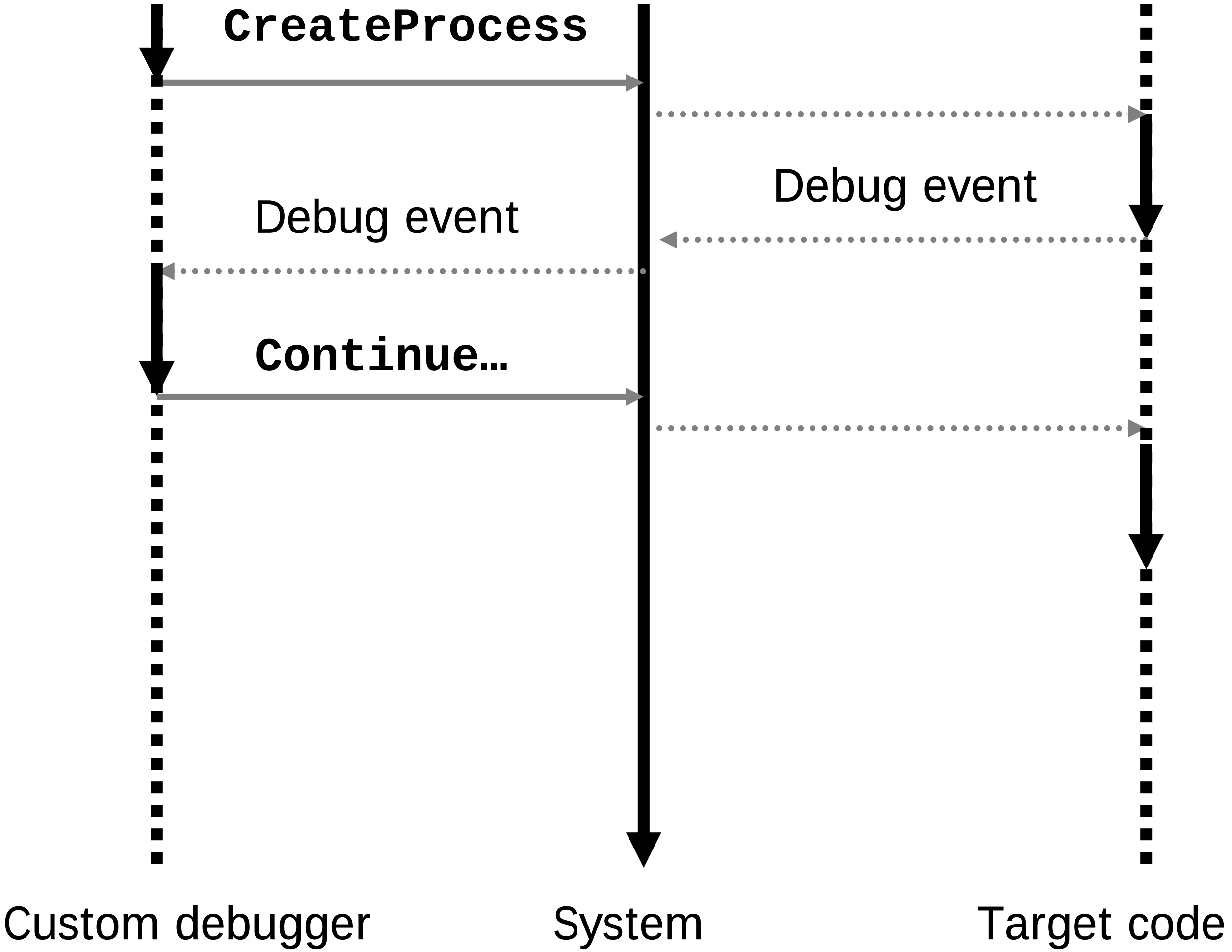
# The idea: use SPA



# What is SPA for software

- We need to measure something which is representative from an execution path
- We chose to record the list of couples (address,opcode) executed by the processor
  - We call this an execution trace
- But how to record this ?
- Basically we wrote a custom debugger

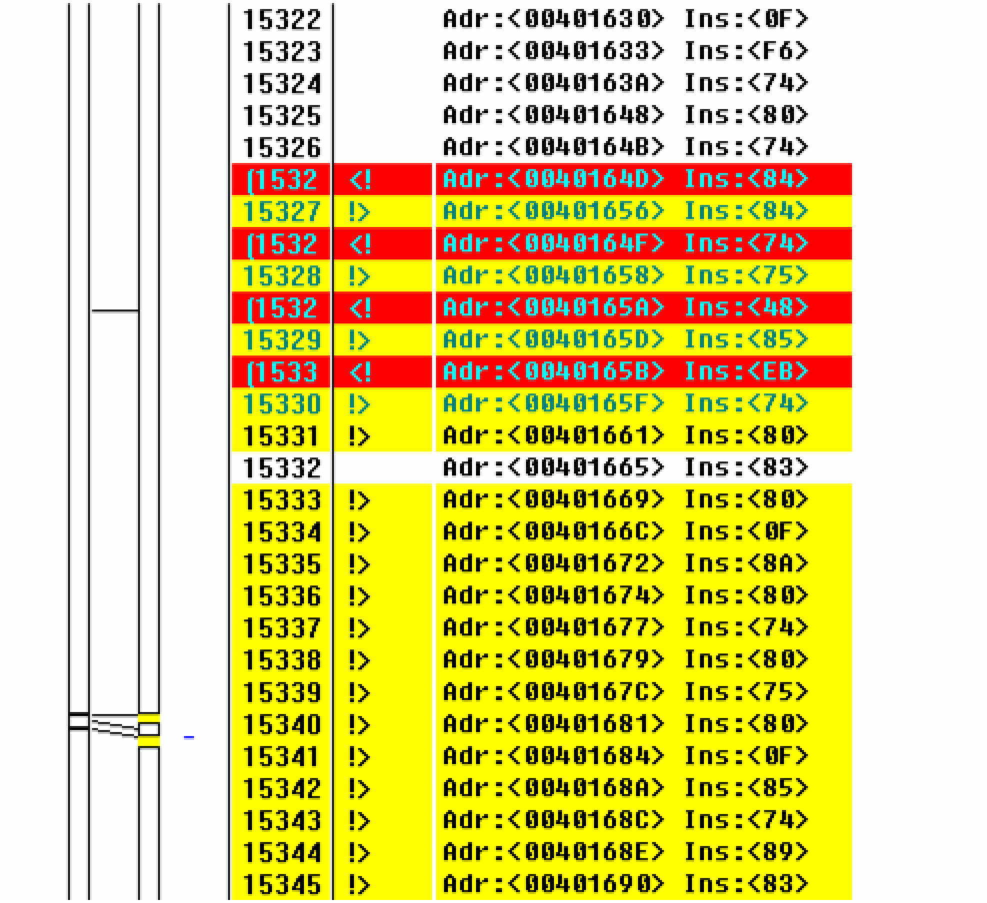
# The Windows' debugging API



# In practice

- The debugging event we need is the execution of a single opcode
- Process
  - Stop the target process
  - Access the saved registers
  - Set the step bit from the debug register
  - Resume  the process

# First result traces



```
15322       Adr:<00401630> Ins:<0F>
15323       Adr:<00401633> Ins:<F6>
15324       Adr:<0040163A> Ins:<74>
15325       Adr:<00401648> Ins:<80>
15326       Adr:<0040164B> Ins:<74>
(1532  <!   Adr:<0040164D> Ins:<84>
15327  !>   Adr:<00401656> Ins:<84>
(1532  <!   Adr:<0040164F> Ins:<74>
15328  !>   Adr:<00401658> Ins:<75>
(1532  <!   Adr:<0040165A> Ins:<48>
15329  !>   Adr:<0040165D> Ins:<85>
(1533  <!   Adr:<0040165B> Ins:<EB>
15330  !>   Adr:<0040165F> Ins:<74>
15331  !>   Adr:<00401661> Ins:<80>
15332       Adr:<00401665> Ins:<83>
15333  !>   Adr:<00401669> Ins:<80>
15334  !>   Adr:<0040166C> Ins:<0F>
15335  !>   Adr:<00401672> Ins:<8A>
15336  !>   Adr:<00401674> Ins:<80>
15337  !>   Adr:<00401677> Ins:<74>
15338  !>   Adr:<00401679> Ins:<80>
15339  !>   Adr:<0040167C> Ins:<75>
15340  !>   Adr:<00401681> Ins:<80>
15341  !>   Adr:<00401684> Ins:<0F>
15342  !>   Adr:<0040168A> Ins:<85>
15343  !>   Adr:<0040168C> Ins:<74>
15344  !>   Adr:<0040168E> Ins:<89>
15345  !>   Adr:<00401690> Ins:<83>
```

# Implemented enhancements

- Track the created processes and threads
- Stop tracing in Windows API
- Don’t debug the target code step by step, but interrupt at end of a linear code section
  - Need to implement a instruction decoder
- Non determinism (e.g active polling)

- Dynamically patch the code when traces differ

- Current tool status
  - Work only on protection at program start-up
  - Just a proof of concept from 2002
  - No plan for further developments

# Conclusion

- Numerous tools are needed for very different types of analysis
- Few tools are really convenient and powerful
- Most of the time custom tools are needed
- Automation is mandatory if you are not a hacker working overnight for free or if you don’t have a lab in low labour cost countries