SECURE-IC

THE SECURITY SCIENCE COMPANY

Pre-Silicon Security Evaluation

Security Verification Towards EDA

Pre-Silicon Security Evaluation

CONTEXT

- Banking Payments
- Mobile Phone
- Smartcards
- Computers
- Laptop
- Tablet
- Aerospace & Defense
- ...

Mature Markets / Mature Security

- Automotive
- Factories
- Retail
- Health
- Machine to Machine Communication
- ...

Mature Markets / Emerging Security

- Wearables
- Drones
- Smart Home
- Smart Cities

Emerging Markets / Emerging Security

Pre-Silicon Security Evaluation

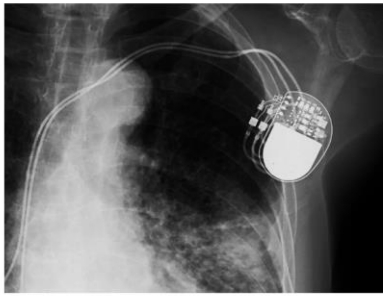
CONTEXT

Pre-Silicon Security Evaluation

CONTEXT

LILY HAY NEWMAN SECURITY 08.09.18 12:30 PM

A NEW PACEMAKER HACK PUTS MALWARE DIRECTLY ON THE DEVICE



CHOO CHIN/GETTY IMAGES

THE FIRST PACEMAKER hacks emerged about a decade ago. But the latest variation on the terrifying theme depends not

A CLEVER ANDROID HACK TAKES ADVANTAGE OF SLOPPY STORAGE

ANGEL GARCIA/BLOOMBERG/GETTY IMAGES

AN ANDROID APP has two choices for where to put its data on a device: internal storage, where it's safe and snug, isolated by the operating system's sandbox, and external storage, where data can move between apps but isn't as protected. Most of the time, that setup works just fine. But when

SECURE-IC
THE SECURITY SCIENCE COMPANY

WHAT KIND OF ATTACKS ?

Security Evaluation

ABOUT SECURITY THREATS

Cryptography is robust but Physical Attacks are here …

Passive Analyses

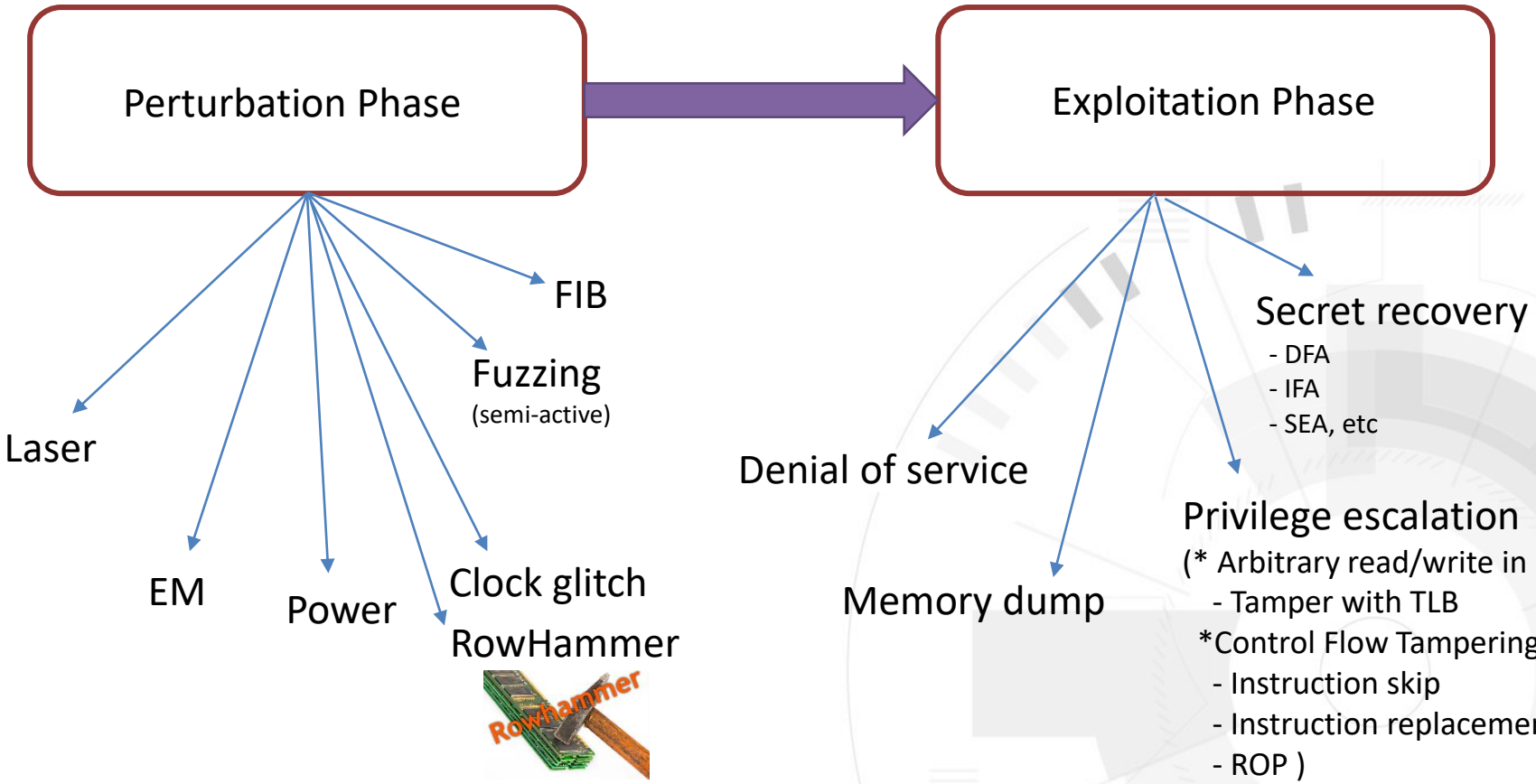
- Do not Interact directly with the target:
  - ➔ Exploit a physical property related to the activity of the sensitive data
- Common analyses: SCA (Side-Channel Analyses)

Active Analyses

- Interact directly with the target:
  - ➔ Access to the target
  - ➔ Perturbate its normal behavior
- Common analyses: FIA (Fault Injection Analyses) / Active Probing (FIB)

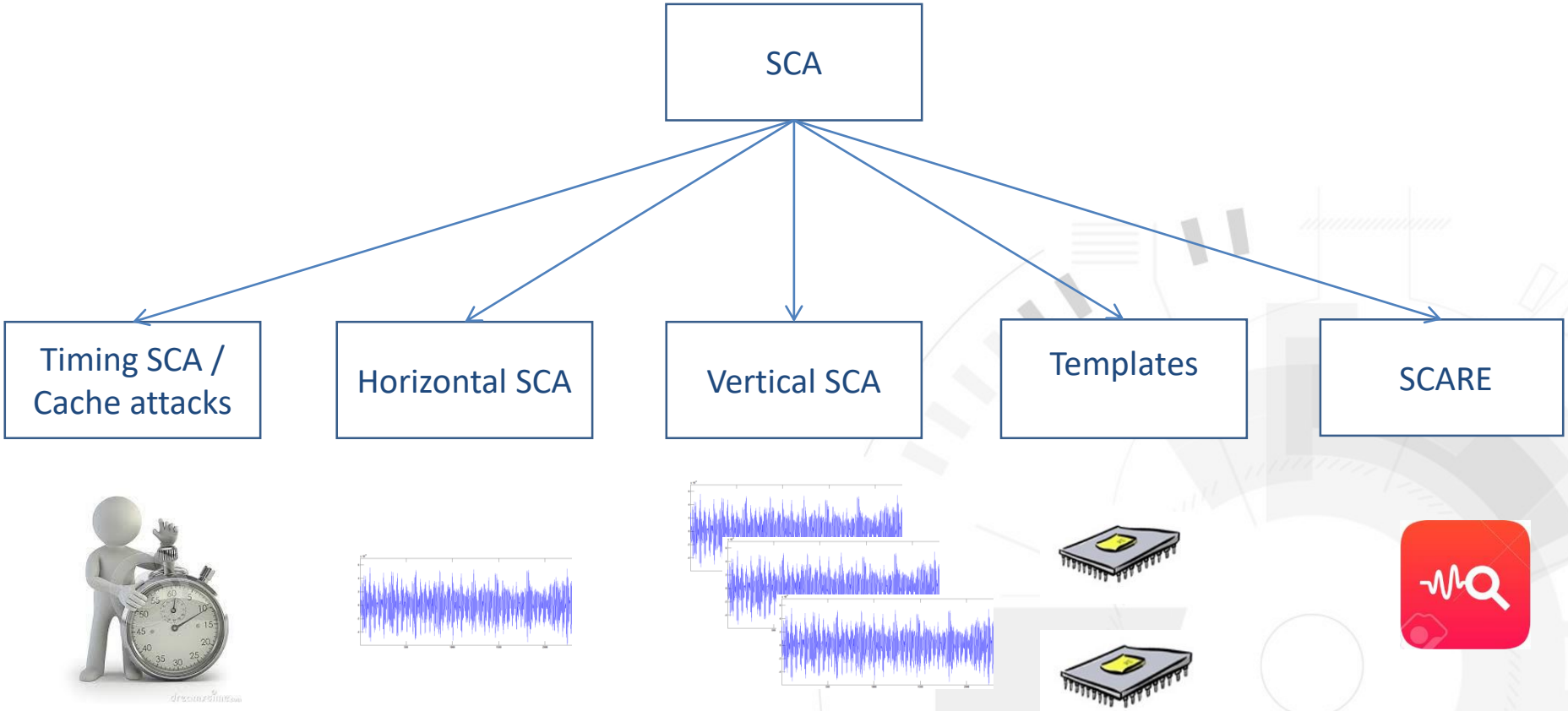
Security Evaluation

ABOUT SECURITY THREATS: Active Analysis



Security Evaluation

ABOUT SECURITY THREATS: Passive Analysis (SCA)



SECURE-IC

THE SECURITY SCIENCE COMPANY

EDA Tools for Design For Security

Pre-Silicon Security Evaluation

EDA: Electronic Design Automation Axes

EDA Axes

Design
- High level Synthesis
- Logic Synthesis
- Schematic capture
- Layout

Simulation
- Transistor / Logic / Behavioral / HW Emulation

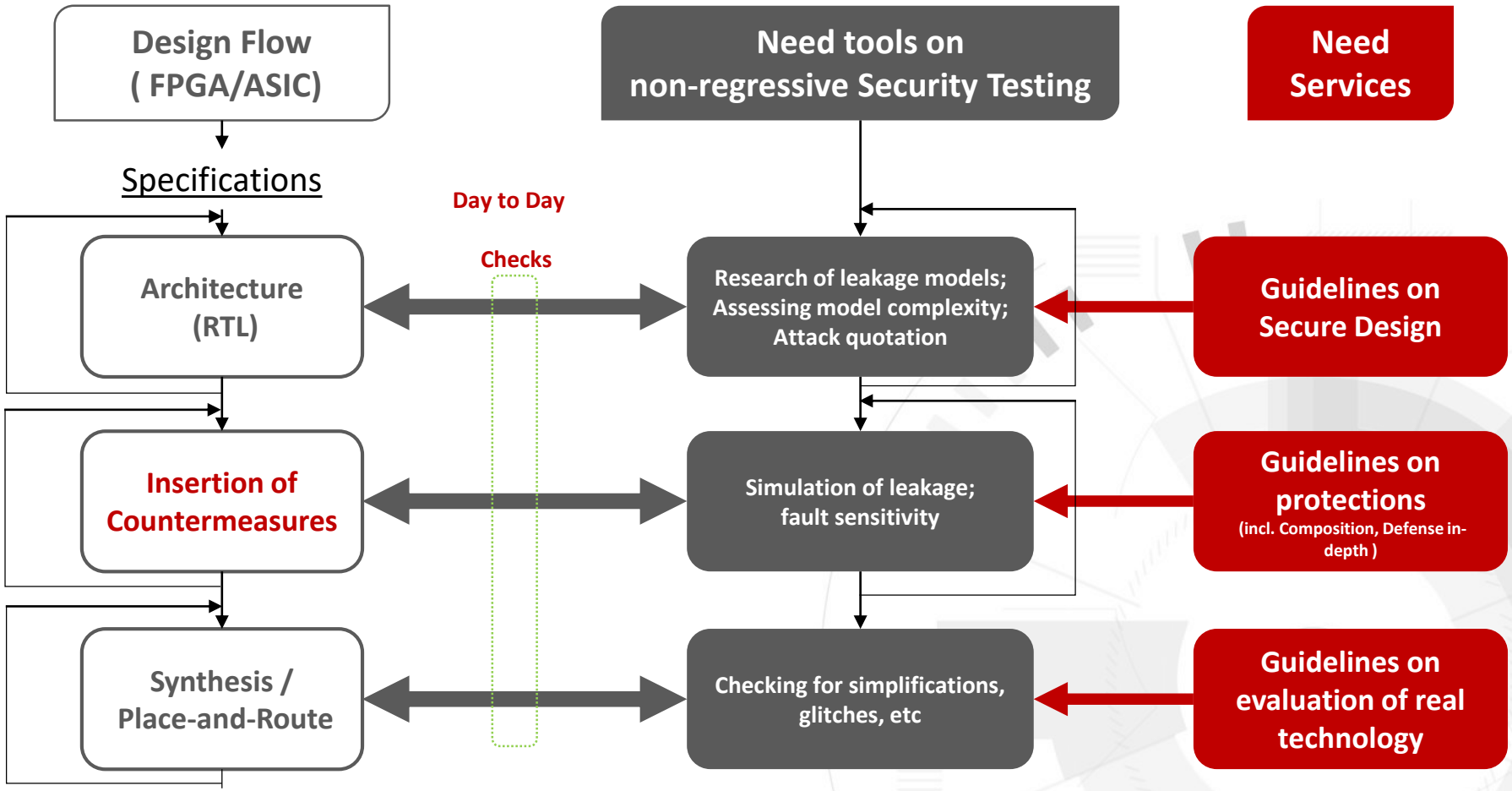
Analysis & Verification
- Functional verification
- Formal verification
- Static Timing Analysis
- Clock Domain Crossing Verification
- Mask Data Preparation

Functional Safety
- Analysis (computation failure in time rate and diagnostic)
- Synthesis (add reliability to structured components)
- Verification (run a fault campaign: error detection)

Security Analysis
- RTL / PS / PR / Layout Security (SCA / FIA and more)
- HW and SW Analysis

Pre-Silicon Security Evaluation

- **Design-for-Security (DFS) approach is being universalized**



Pre-Silicon Security Evaluation

- **Design-for-Security (DFS) approach is being universalized**

Security leaking signal detection by the VIRTUALYZR

```
library design_lib;
use design_lib.aes_pkg.all;

entity cipher is
  port(
    n_reset                 : in  std_logic;
    clk                     : in  std_logic;
    data                    : in  std_logic_vector(127 downto 0);
    key_schedule            : in  std_logic_vector(127 downto 0);
    key_schedule_rdy        : in  std_logic;
    round                   : in  std_logic_vector(3 downto 0);
    cipher                  : out std_logic_vector(127 downto 0));
end entity cipher;

architecture cipher_arch_1 of cipher is

  signal round_value            : std_logic_vector(127 downto 0);
  signal sub_bytes_std_value    : std_logic_vector(127 downto 0);
  signal mix_columns_value      : std_logic_vector(127 downto 0);
  signal shift_rows_value       : std_logic_vector(127 downto 0);
  signal add_key_output_value   : std_logic_vector(127 downto 0);
  signal cipher_round_value     : std_logic_vector(127 downto 0);
  signal cipher_s, cipher_r     : std_logic_vector(127 downto 0);

  attribute keep_hierarchy : string;
  attribute keep_hierarchy of cipher_arch_1 : architecture is "yes";

  attribute equivalent_register_removal : string;
  attribute equivalent_register_removal of cipher_r : signal is "no";
  attribute equivalent_register_removal of cipher_s : signal is "no";


  attribute keep : string;
  attribute keep of round_value             : signal is "true";
```

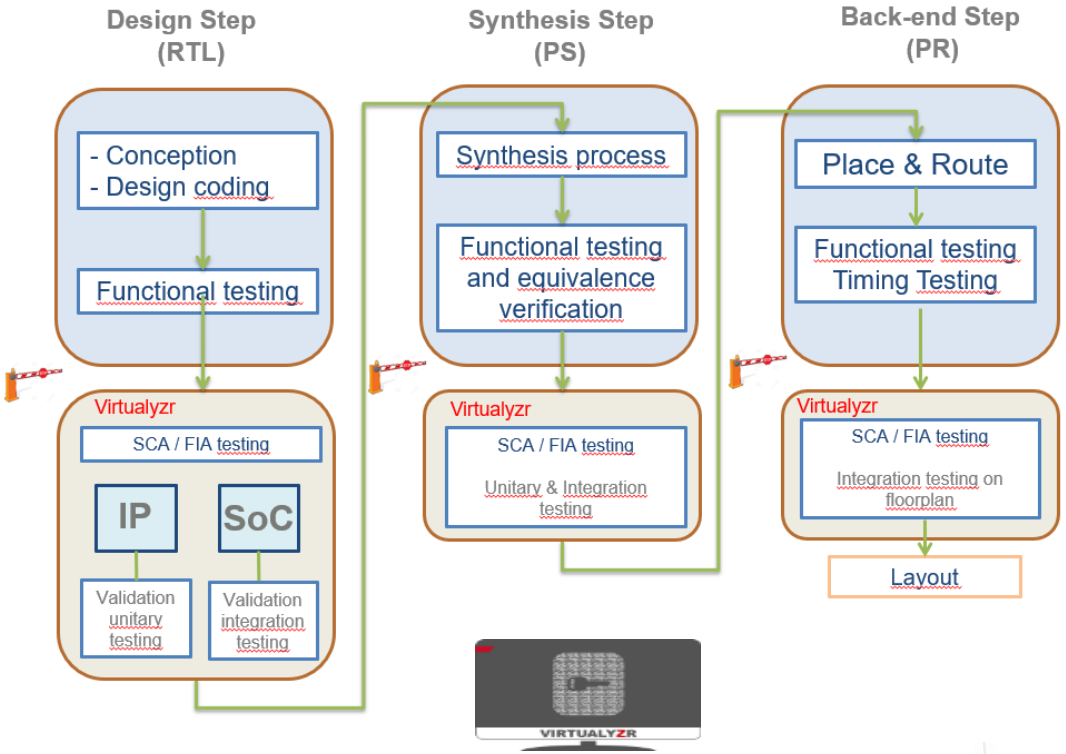
Virtual analysis

Iterative Feeback

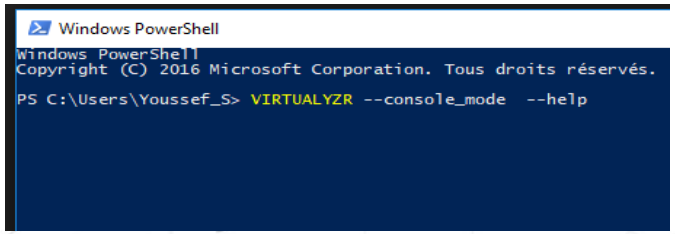
```
library design_lib;
use design_lib.aes_pkg.all;

entity cipher is
  port(
    n_reset                 : in  std_logic;
    clk                     : in  std_logic;
    data                    : in  std_logic_vector(127 downto 0);
    key_schedule            : in  std_logic_vector(127 downto 0);
    key_schedule_rdy        : in  std_logic;
    round                   : in  std_logic_vector(3 downto 0);
    cipher                  : out std_logic_vector(127 downto 0));
end entity cipher;

architecture cipher_arch_1 of cipher is

  signal round_value            : std_logic_vector(127 downto 0);
  signal sub_bytes_std_value    : std_logic_vector(127 downto 0);
  signal mix_columns_value      : std_logic_vector(127 downto 0);
  signal shift_rows_value       : std_logic_vector(127 downto 0);
  signal add_key_output_value   : std_logic_vector(127 downto 0);
  signal cipher_round_value     : std_logic_vector(127 downto 0);
  signal cipher_s, cipher_r     : std_logic_vector(127 downto 0);

  attribute keep_hierarchy : string;
  attribute keep_hierarchy of cipher_arch_1 : architecture is "yes";

  attribute equivalent_register_removal : string;
  attribute equivalent_register_removal of cipher_r : signal is "no";
  attribute equivalent_register_removal of cipher_s : signal is "no";


  attribute keep : string;
  attribute keep of round_value             : signal is "true";
```

Pre-Silicon Security Evaluation

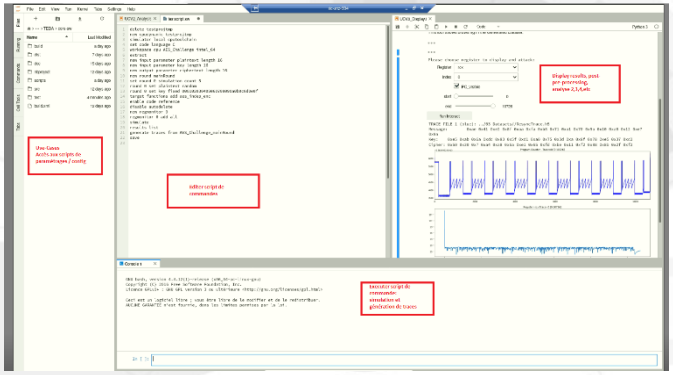
Easy Integration to the Design Life-Cycle



Command Line User Interface
for full automation

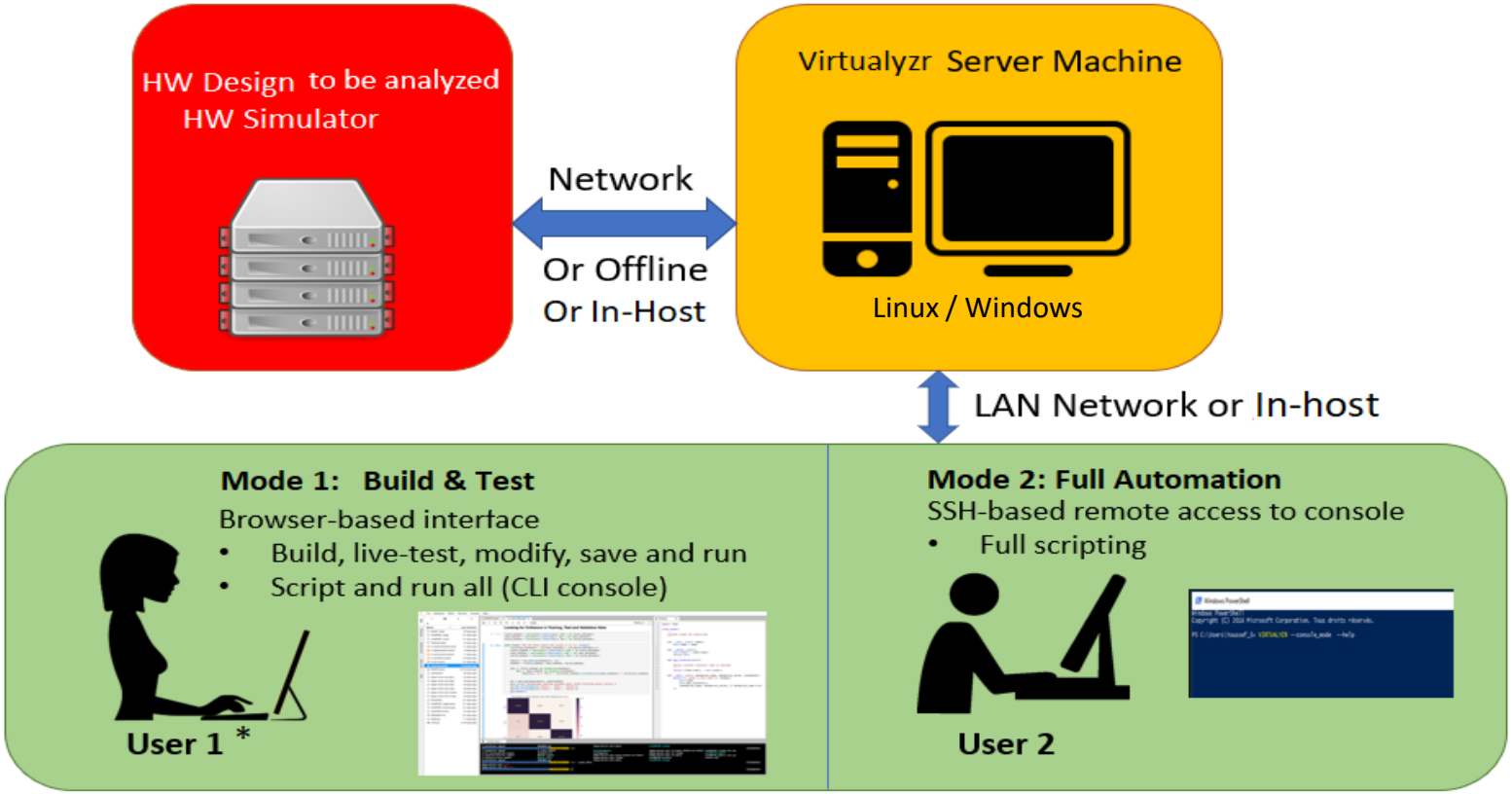


Graphical User Interface
for analysis build & test



Pre-Silicon Security Evaluation

Easy Integration to the Design Life-Cycle



* Or Multiple-users connexions

Pre-Silicon Security Evaluation

Snapshots (Build & Test interface)

Secure-IC

File Edit View Run Kernel Tabs Settings Help

Files Running Commands Cell Tools Tabs

Virtualyzr

Name	Last Modified
UCV9_SCA analysis of a...	20 days ago
UCV7_SCA analysis of ...	20 days ago
UCV8_SCA of a cipher ...	20 days ago
UCV6_SCA analysis of ...	20 days ago
UCV5_SCA analysis of a...	20 days ago
UCV4_SCA analysis of ...	20 days ago
UCV3_Analysis perform...	20 days ago
UCV30_SCA on Secure-...	20 days ago
UCV2_Leakage detecti...	20 days ago
UCV29_SCA on functio...	20 days ago
UCV28_FIA EM injectio...	20 days ago
UCV27_FIA Laser injecti...	20 days ago
UCV26_FIA Clock-Glitc...	20 days ago
UCV24_ISO-17825 on ...	20 days ago
UCV25_SCA Cartography	20 days ago
UCV23_ISO-17825 on S...	20 days ago
UCV21_Custom analysi...	20 days ago
UCV22_SCA with sever...	20 days ago
UCV20_Custom analysi...	20 days ago
UCV1_DPA_CPA_LRAUC...	20 days ago
UCV19_Profiling-based...	20 days ago
UCV18_Hardware Leak...	20 days ago
UCV17_High-order SC...	20 days ago
UCV16_SCA with real f...	20 days ago
UCV15_SCA analysis on...	20 days ago
UCV14_SCA analysis of ...	20 days ago
UCV13_SCA analysis of ...	20 days ago
UCV11_SCA analysis of ...	20 days ago
UCV12_SCA analysis of ...	20 days ago
UCV10_SCA analysis of ...	20 days ago

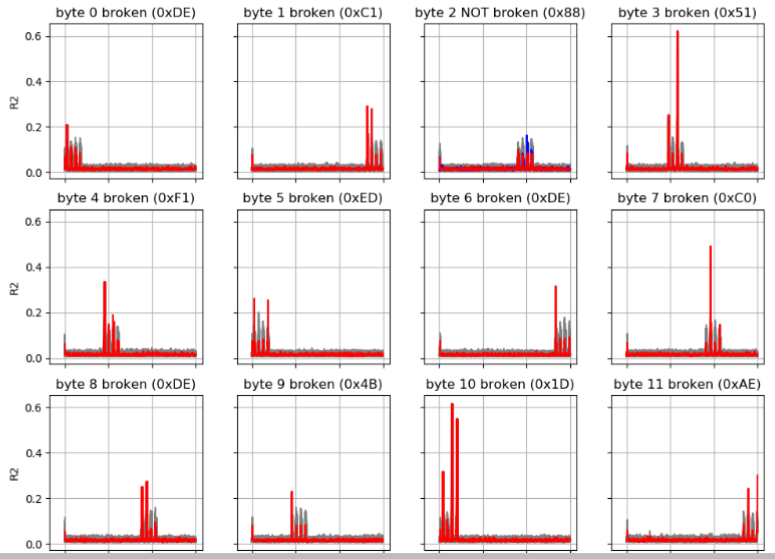
UC04_Analysis-

Markdown Python 2

```
ts = Traces_Object (ts_path)
analysis=LRA_Analysis(ts,lra_path,range(16))
```

In [12]:
```
keyfound=analysis.Compute_R2_Dx256_max_over_R(show=True)
```

```
Analyze LRA max over R:  99% |>>>>>>>>>> | ETA:   0:00:00 Elapsed Time: 0:00:10
```



aes_fsm.vhd

```
20 library ieee;
21 use ieee.std_logic_1164.all;
22 use ieee.numeric_std.all;
23 library design_lib;
24 use design_lib.aes_pkg.all;
25
26 entity aes_fsm is
27   port (
28     clk                 : in  std_logic;
29     n_reset             : in  std_logic;
30     start               : in  std_logic;
31     round               : out std_logic_vector(3 downto 0);
32     ke_ready            : out std_logic;
33     done                : out std_logic);
34 end entity aes_fsm;
35
36 architecture aes_fsm_arch_1 of aes_fsm is
37
38     constant init_st : std_logic_vector(1 downto 0) := "00";
39     constant ke_st : std_logic_vector(1 downto 0) := "01";
40     constant dn_st : std_logic_vector(1 downto 0) := "10";
41 --  type state is array (init_st,ke_st,dn_st);
42 --  signal pr_state : state;
43 --  signal nx_state : state;
44   signal pr_state : std_logic_vector(1 downto 0);
45   signal nx_state : std_logic_vector(1 downto 0);
46
47   signal current_round : std_logic_vector(3 downto 0);
48
49   attribute keep_hierarchy : string;
50   attribute keep_hierarchy of aes_fsm_arch_1    : architecture is "yes";
51
52   attribute keep : string;
53   attribute keep of pr_state        : signal is "true";
54   attribute keep of nx_state        : signal is "true";
55   attribute keep of current_round   : signal is "true";
56
57 begin
58
59   ----------------------------------------------------------------------
```

Terminal 1

```
    soc list
    target
    target add <design_filter>
    target add all
    uiwatcher
    unload
    workspace
    workspace ip <workspace_name> <ip_definition_level>
    workspace list
    workspace soc <soc_plugin> <workspace_name> <soft_name>
[virtualyzr]
```

Pre-Silicon Security Evaluation

Your Gain

- Provides a security verification layer: It runs hands-in-hands with functional verification workflow
- Compliance with **ISO/IEC 17825, 20085** and evaluation standards **CC, ISO/IEC 15408 and FIPS 140**
- Detects, characterizes and extracts the security vulnerabilities from the design:
  IP and SoCs / cryptographic and non cryptographic targets / FPGA, ASIC, eFPGA
- Allows considering best analysis conditions (white box analysis, free noise, no jitter, etc).
- Allows performing security checkpoints at different design levels (behavioural level, netlist level)
- Allows checking countermeasures by fixing issues in masking scheme for instance:
  self-masking, variable time operation, simplification, etc.
- Allows more fidelity with the final technology
- The evaluation is cheaper as no measurement equipment or platforms are required.
- Improve and put forward the DFS (Design for security) approach.

Pre-Silicon Security Evaluation

Differentiator Features

- Seamless flow from analysis of software, to mixed SW/HW, pure HW, netlist, GDSII, with the same interface
- Identification and characterization of security issues, annotated directly in the design as inputted by the user
- Simultaneous security and low power objectives /or security and safety (ISO 26262)
- Interactive API with the tool + full automation
- Latest distinguishers: collision, LRA, machine learning, etc.
- Delivered with many Use Cases on representative analyses, for a fast learning curve.

SECURE-IC

THE SECURITY SCIENCE COMPANY

THANKS FOR YOUR ATTENTION

CONTACT

EUROPE	sales-EU@secure-IC.com
APAC	sales-APAC@secure-IC.com
JAPAN	sales-JAPAN@secure-IC.com
AMERICAS	sales-US@secure-IC.com