

### AUTOMATING CYBER-PHYSICAL SECURITY VERIFICATION IN THE SOC DESIGN FLOW

Valentin Peltier, Product Marketing Manager

July 2025











## **1. INTRODUCTION**



## A GLOBAL LEADER IN EMBEDDED SECURITY

IoT devices being interconnected, each and every object could be a threat for the whole network.

Therefore, the security of the objects or the devices with their

**lifecycle management is key**, and so is their data. To ensure the integrity of this data, the whole system must be secured and managed. **Trusted devices enable trusted data**.

### ONE DAY, SECURITY WILL BE WORTH MORE THAN THE DEVICES



Secure-IC partners with its clients to provide them with the best end-to-end cybersecurity solutions for embedded systems and connected objects, **from Chip to Cloud** 





## 2. ABOUT PHYSICAL SECURITY

CRYPOGRAPHY IS ROBUST, BUT...



### **SECURITY THREATS**

Cryptography is robust but **physical attack** are here...

- Side-Channel Analysis (SCA)
- Fault Injection Analysis (FIA)
- Cache-Timing Analysis
- Reverse Engineering
- Hardware Trojan





### **A BIT OF HISTORY**

#### **Side-Channel Analysis**

- Introduced in the 1990s by Paul Kocher\*, an American cryptographer
- Recover secret keys by measuring the execution time
- Physical attacks was born!

#### Fault Injection Analysis

- Initially used in hardware testing & reliability engineering, observing how system respond
   → safety-oriented
- Evolution into a security-focused methodology in the 1990s\*\*

\*Paul Kocher, 1996, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems

- \*\*
- R. Anderson & M. Kuhn, 1996, Tamper Resistance a Cautionary Note
- D. Boneh, R. DeMillo, & R. Lipton, 1997, On the importance of checking cryptographic protocols for faults
- E. Biham & A. Shamir, 1997, Differential fault analysis of secret key cryptosystems





#### Side-Channel Analysis

 Measure execution activity of sensitive data into embedded system



 All these characteristics become a potential leakage

#### Fault Injection Analysis

 Disturb embedded system during execution



- Injecting faults at different level:
  - Physical (memory, transistors, logic gates)
  - Micro-Code (skip/jump instruction, read/write)
  - Code (IO/crypto routines, loop statement)







- Secret Key Recovery
- Reverse Engineering

• . . .



#### Fault Injection Analysis

- Denial of Service
- Privilege Escalation
- Secret Recovery
- Reverse Engineering
- • •





. . .

## **COMMON WEAKNESS ENUMERATION**

- MIHW 2021: the first list focused on hardware security
- Created by the Hardware CWE SIG (experts from industry, academia, and government)
- CWE-1300 Improper Protection of Physical Side Channels
- CWE-1247 Improper Protection Against Voltage and Clock Glitches
- CWE-1332 Improper Handling of Faults that Lead to Instruction Skips
- And there are many others...
  - CWE-1319: Improper Protection against Electromagnetic Fault Injection
  - CWE-1384: Improper Handling of Physical or Environmental Conditions



(2025 update in progress)



### **STANDARD CERTIFICATIONS**









#### **Common Criteria & Vulnerability Assessment**

| Range of values*<br>*final attack potential =<br>identification + exploitation | TOE resistant to<br>attackers with attack<br>potential of: | Analyses to be carried out in Secure-IC tools<br>(Catalyzr/Virtualyzr/Analyzr)   | Corresponding AVA_VAN                                       | Corresponding EAL     |
|--|--|--|---|-----------------------|
| 0-15   | No rating  | No analysis required   | AVA VAN.1:<br>vulnerability survey                          | EAL 1                 |
| 16-20  | Basic  | SPA (Single trace analysis)  | AVA_VAN.2:<br>vulnerability analysis                        | EAL2/3                |
| 21-24  | Enhanced-Basic   | DPA ([1st-order] Differential analysis)  | AVA VAN.3:<br>focused vulnerability<br>analysis             | EAL 4                 |
| 25-30  | Moderate   | High-order analysis  | AVA_VAN.4:<br>methodical vulnerability                      | EAL 5 (or4+) analysis |
| 31 and above   | High   | All abovementioned analysis, plus:<br>collision-analysis, stochastic analysis, mutual<br>information analysis, template attack, machine-<br>learning attacks, deep-learning attack | AVA_VAN.5:<br>advanced methodical<br>vulnerability analysis | EAL 6/7 (or 5+)       |

#### FIPS 140-3 is transitioning to ISO/IEC 19790

#### **ISO 17825**

Table 1 — Associations between non-invasive attack methods and security functions covered by this International Standard

| Crypt<br>ISO St   | ographi<br>tandard   | c Modul<br>s  | e Testin   | ng Fami  | ly-  |   |   | Security functions |             |
|---|--|---|--|--|--|---|---|--------------------|-------------|
|   |  | 19790:2006  | 24759:2008<br>24759:2014   | First Edition<br>(Published 05-15-15<br>Physical Security                | )  |   |   | Symmetric-Key      | AES         |
| 20085-1<br>Prepare 2 <sup>nd</sup> CD)                              | 17825<br>First Edition<br>(Publiched 01-15-16)                           | 19790:2012<br>Second Edition<br>(Published 08-15-12)                                | 24759:2014<br>Third Edition  | Attacks, Mitigation<br>Techniques and<br>Security Requireme              | nts  |   |   |                    | Triple-DES  |
| test tool calibration<br>ods for use in testing                     | Testing methods for the<br>mitigation of non-<br>invasive attack classes | (Corrected 12-15-15)<br>Security  | (Published 03-01-17)<br>Test requirements<br>for   | 20540  |  |   |   |                    | Stream Cip  |
| ation techniques in<br>ographic modules —                           | against cryptographic<br>modules   | cryptographic<br>modules  | cryptographic<br>modules   | (Prepare for Publish<br>Guidelines for Testin<br>Cryptographic Modu      | n)<br>ng<br>nles                                   |   |   | Asymmetric-Key     | Plain RSA ( |
| techniques  | A  |   |  | in their Operation:<br>Environment                                       | 1  | _ | - |                    | RSA PKCS    |
| 20085-2   |  |   |  |  |  |   |   |                    | RSA PKCS    |
| Prepare 4 <sup>th</sup> WD)<br>tool requirements                    | 18367  | 19896-1   | 19896-2  | V1514  | 20543  |   |   |                    | DSA         |
| est tool calibration<br>ods for use in testing<br>n-invasive attack | First Edition<br>(Published 12-15-16)                                    | (Prepare FDIS)<br>Competence  | (Prepare 1 <sup>st</sup> DIS)<br>Competence  | (SP)<br>Competence   | (Prepare 1 <sup>st</sup> DIS)<br>Test and analysis |   |   |                    | ECDSA       |
| ation techniques in<br>ographic modules —<br>: 2 Test calibration   | algorithms and security<br>mechanisms                                    | information security<br>testers and evaluators                                      | information security<br>testers and evaluators   | Laboratories<br>performing IT security                                   | bit generators within<br>ISO/IEC 19790 and         |   |   | Hashing mechanisms | SHA         |
| ods and apparatus   | conformance testing  | <ul> <li>Part 1: Introduction,<br/>concepts and general<br/>requirements</li> </ul> | <ul> <li>Part 2: Knowledge,<br/>skills and effectiveness<br/>requirements for</li> </ul> | testing and evaluation<br>Need this document for<br>Cryptographic Module | ISO/IEC 15408                                      |   |   | RNG and RBG        | Determinist |
|   |  |   | ISO/IEC 19790 testers  | Testing family? ??????   |  |   |   |                    | Non-determ  |
|   |  |   |  |  |  |   |   |                    |             |

| ocurity functions | Non-Invasive attack methods |            |           |    |
|-------------------|-----------------------------|------------|-----------|----|
|                   | -                           | SP A/SEM A | DP A/DEMA | TA |
| /mmetric-Key      | AES                         | A          | A         | A  |
|                   | Triple-DES                  | A          | A         | Α  |
|                   | Stream Ciphers              | A          | A         | A  |
| ymmetric-Key      | Plain RSA (Key wrapping)    | A          | A         | A  |
|                   | RSA PKCS#1 v1.5             | Α          | A         | A  |
|                   | RSA PKCS#1 v2.1             | NA         | NA        | A  |
|                   | DSA                         | A          | A         | A  |
|                   | ECDSA                       | A          | A         | Α  |
| shing mechanisms  | SHA                         | A          | NA        | NA |
| IG and RBG        | Deterministic               | A          | NA        | NA |
|                   | Non-deterministic           | A          | NA        | NA |





# 3. VERIFICATION TOOLS & AUTOMATION

### ALONG DESIGN FLOW & V-CYCLE



### **SOFTWARE LAYER VERIFICATION**

| <pre>var highlight = function(\$element<br/>if (typeof pattern === 'trian<br/>var regex = (typeof pattern === 'trian<br/>var highlight = function(node) {<br/>var skip = 0;<br/>if (node.nodeType === 3) {<br/>var pos = node.data.search(reges);<br/>if (pos &gt;= 0 &amp;&amp; node.data.lengt)<br/>var match = node.data.match(reges);<br/>var spannode = document.createller<br/>var middlebit = node.splitText(pos);<br/>var middlebit = middlebit.splitText(match);<br/>var middleclone = middlebit.splitText(match);<br/>var middleclone;<br/>monode.appendChild(middleclone);</pre>  |                | 29 | 'use strict';  |
|--|----------------|----|--|
| <pre>var highlight = function(selence<br/>if (typeof pattern === 'trian 'trian'<br/>var regex = (typeof pattern === 'trian'<br/>var skip = 0;<br/>if (node.nodeType === 3) {<br/>var pos = node.data.search(regen).<br/>if (pos &gt;= 0 &amp;&amp; node.data.lengt)<br/>var match = node.data.match(regen).<br/>var spannode = document.createfine<br/>var spannode = document.createfine<br/>var spannode.className = 'highlight;<br/>var middlebit = node.splitText(pos).<br/>var middlebit = middlebit.splitText(set)<br/>var middleclone = middlebit.clonetes(set)<br/>is mode.appendChild(middleclone);<br/>parentNode.replaceChild(spece).</pre>   |                |    |  |
| <pre>if (typeof pattern 'triangle and a second var regex = (typeof pattern 'triangle and 'type and 'type</pre>                           |                |    | var highlight = function(Selement, persons)  |
| <pre>var regex = (typeof pattern</pre>   |                | 32 | if (typeof pattern === 'string' 🛀 procession of the string' 🛀  |
| <pre>var highlight = function(node) {     var skip = 0;     if (node.nodeType === 3) {         var pos = node.data.search(regen).         if (pos &gt;= 0 &amp;&amp; node.data.lengt) {             var match = node.data.lengt) {             var match = node.data.match(regen).             var spannode = document.createllenet {             var spannode.className = 'highlight :             var middlebit = node.splitText(pos).             var endbit = middlebit.cloneese(tree             var middleclone = middlebit.cloneese(tree             var mode.appendChild(middleclone);             var mode.replaceChild(space)</pre>  |                |    | var regex = (typeof pattern string ) and degree and the second string of the second stri      |
| <pre>var highlight = function(node) var skip = 0; if (node.nodeType === 3) { var pos = node.data.search(reges) if (pos &gt;= 0 &amp;&amp; node.data.lengt) var match = node.data.match(reges) var spannode = document.createElement spannode.className = 'highlight'; var middlebit = node.splitText(pos); var middleclone = middlebit.clonetode(true) var middleclone);</pre>   |                |    |  |
| <pre>var skip = 0;<br/>if (node.nodeType === 3) {<br/>var pos = node.data.search(regen).<br/>if (pos &gt;= 0 &amp;&amp; node.data.length &gt;= )<br/>var match = node.data.match(regen).<br/>var spannode = document.createElement<br/>spannode.className = 'highlight;<br/>var middlebit = node.splitText(pos);<br/>var endbit = middlebit.splitText(match(a))<br/>var middleclone = middlebit.cloneuce(true)<br/>spannode.appendChild(middleclone);<br/>spannode.appendChild(middleclone);<br/>spannode.appendChild(middleclone);<br/>spannode.appendChild(middleclone);</pre>   |                |    | <pre>var highlight = function(node) {</pre>  |
| <pre>if (node.nodeType === 3) {     var pos = node.data.search(regex);     if (pos &gt;= 0 &amp;&amp; node.data.length &gt; 0)     var match = node.data.match(regex);     var spannode = document.createElement search(regex);     var middlebit = node.splitText(pos);     var middlebit = middlebit.splitText(match(n));     var middleclone = middlebit.cloneWode(true);     var middleclone;;     var spannode.appendChild(middleclone);     var spannode.replaceChild(spannode, replaceChild(spannode, rep</pre> | Figures Inth   |    | var skip = 0;  |
| <pre>var pos = node.data.search(repres) if (pos &gt;= 0 &amp;&amp; node.data.lengta &gt;= 0 var match = node.data.match(repres) var spannode = document.createflement spannode.className = 'highlight'; var middlebit = node.splitText(pos); var endbit = middlebit.splitText(match(repres)) var middleclone = middlebit.cloneMode(true) var middleclone = middlebit.cloneMode(true) var middleclone; var middleclone;</pre>   |                |    | <pre>1f (node.nodeType === 3) {</pre>  |
| <pre>if (pos &gt;= 0 &amp;&amp; node.data.lenges<br/>var match = node.data.match(reges).<br/>var spannode = document.createflement<br/>spannode.className = 'highlight':<br/>var middlebit = node.splitText(pos):<br/>var endbit = middlebit.splitText(match(s));<br/>var middleclone = middlebit.clonelose(true);<br/>spannode.appendChild(middleclone);<br/>spannode.replaceChild(sparnode, compared);</pre>   |                | 18 | var pos = node.data.search(regrate   |
| <pre>var match = node.data.match(tege<br/>var spannode = document.createElement(spannode.className = 'highlight';<br/>var middlebit = node.splitText(pos);<br/>var endbit = middlebit.splitText(match(s));<br/>var middleclone = middlebit.cloneWode(true);<br/>spannode.appendChild(middleclone);<br/>spannode.replaceChild(spannode, match)</pre>  | Store o caller |    | <pre>if (pos &gt;= 0 &amp;&amp; node.data.letters</pre>  |
| <pre>var spannode = document.treater<br/>spannode.className = 'highlight';<br/>var middlebit = node.splitText(pos);<br/>var endbit = middlebit.splitText(match(#));<br/>var middleclone = middlebit.clonelode(true);<br/>var middleclone);<br/>spannode.appendChild(middleclone);</pre>  |                |    | var match = node.data.match(the matthe second secon |
| <pre>spannode.className = highter<br/>var middlebit = node.splitText(pos):<br/>var endbit = middlebit.splitText(match())<br/>var middleclone = middlebit.cloneWode(true)<br/>var middleclone = middlebit.cloneWode(true)<br/>var middleclone = middleclone);<br/>spannode.appendChild(middleclone);<br/>spannode.replaceChild(sparnode, middleclone);</pre>  | 100101-0700    |    | var spannode = document.creatient  |
| <pre>var middlebit = node.spir(text(match(#))) var endbit = middlebit.splitText(match(#)) var middleclone = middlebit.clonelode(true)) var middleclone = middlebit.clonelode(true) var middleclone = middlebit.clonelode(true)) var middleclone = midd</pre>               |                |    | spannode.className = nightaget(pos)  |
| <pre>var endbit = middlebit.spireteneede(true) var middleclone = middlebit.cloneWode(true) var middleclone); spannode.appendChild(middleclone); spannode.replaceChild(sparnode, modelede)</pre>  |                |    | var middlebit = node.spirtiert(match(e))   |
| <pre>var middleclone = middleclone); spannode.appendChild(middleclone); spannode.replaceChild(sparnode.replace</pre>               |                |    | var endbit = middlebit.spirt.cloneWode(trus);  |
| <pre>spannode.appendChild(miduleControl spannode.appendChild(spannode.ap</pre>               | -              |    | var middleclone = middleclone);  |
| -1 441 abit .parentNode.replace  |                |    | spannode.appendChild(midulecechild(spannode, scotterios)   |
|  |                |    | and bit garentNode. replaced   |
| spannode.appendenzze (   |                |    | spannode.appendomzac   |
| var middlectone = man  |                |    | var mlddlecione = middleclone);  |
|  |                |    |  |





## PRE-SILICON VERIFICATION (SOFTWARE)

#### SCA at <u>Static</u> Level



Code source (C/C++) verification



 $\succ \text{Timing vulnerability detection & warning} \\ \text{(conditional branch, array indexation, ...)} \quad \int_{\substack{\text{for } i = k - 1 \text{ to } 0 \text{ do} \\ a \leftarrow a^2 \mod n \\ \text{if } d_i = 1 \text{ then}}}^{\text{for } i = k - 1 \text{ to } 0 \text{ do}}$ 

 $a \leftarrow a \times m \mod n$ return a

#### SCA at <u>Dynamic</u> Level



- Binary execution monitoring (emulation)
- Register dumping to rebuild activity (EM/Power traces)
- Cryptanalysis methodologies (NICV, T-TEST, TVLA, CPA, DPA, SPA...)

#### Leakage location on design code

| File              | Function   | Line | Register | Sample |
|-------------------|------------|------|----------|--------|
| aes-challenge-c.c | ark_sb_rnd | 95   | eflags   | 207    |
| aes-challenge-c.c | ark_sb_rnd | 95   | eflags   | 211    |
| aes-challenge-c.c | ark_sb_rnd | 95   | eflags   | 274    |



## PRE-SILICON VERIFICATION (SOFTWARE)

Masking Scheme Checker



- Verify the masking protection coverage
- Wires & mask dependencies
- Correlation on protected vs. unprotected databases



- Mesures memory access times (cache vs. RAM)
- Differences leak sensitive data
- e.g. Time taken during AES encryption is key dependent



**Binary Analysis** 

- Firmware analysis for security checkup
- System (filesystem, system users & utilities)
- Sensitives files (certificates, private keys)
- Security level (system calls, code practice...)

#### **Crypto Misuse**



- Cryptographic integration checkup
- Detect cryptography misuse in a crypto API
- Rule-based analysis engine
- e.g. PKCS11, OpenSSL, Secure-IC iSE neo...



## PRE-SILICON VERIFICATION (SOFTWARE)

#### FIA at Static Injection Level



- Code source (C/C++) modification
- Manual modification before compilation (code instruction, global variable)
- Output execution analysis

#### FIA at Dynamic Injection Level



- Binary execution disturbance (emulation)
- Automated modification during emulation (register corruption, instruction jump/bypass)
- Output execution analysis

#### Leakage location on design code

| File              | Function   | Line | Register | Sample |
|-------------------|------------|------|----------|--------|
| aes-challenge-c.c | ark_sb_rnd | 95   | eflags   | 207    |
| aes-challenge-c.c | ark_sb_rnd | 95   | eflags   | 211    |
| aes-challenge-c.c | ark_sb_rnd | 95   | eflags   | 274    |



### **HARDWARE LAYER VERIFICATION**

|  |                  | , <sup>O</sup> Search   |  |  |   |
|--|------------------|---|--|--|---|
| B picarv32.vcd ×   |                  |   |  |  | Ξ   |
| Users > fischerm > Downloads   | > D picorv32     | vcd   |  |  |   |
| File View Settings Help  |                  |   |  |  |   |
| B00 0 000  |                  | ► ₩ H IE →I 0   | <b>E</b>   |  |   |
| Scopes<br>mem<br>handle_axi_arvalid<br>handle_axi_avvalid<br>handle_axi_avvalid<br>handle_axi_rvalid<br>handle_axi_rvalid<br>xorshift64_next<br>uut<br>Variables Filter (context men | i × ¥ ∧ +        | clk<br>mem_axi_awprot [2:0]<br>mem_axi_wvalid<br>mem_axi_wvalid [31:0]<br>mem_axi_wdata [31:0]<br>mem_axi_rready<br>mem_axi_bready<br>mem_axi_awvalid<br>mem_axi_awvalid<br>mem_axi_avvalid<br>mem_axi_arvalid<br>mem_axi_arprot [2:0]<br>mem_axi_arprot [2:0]<br>mem_axi_arddr [31:0]<br>AXI_TEST<br>VERBOSE<br>async_axi_transaction [4:0]<br>axi_test<br>delay_axi_transaction [4:0] | 9<br>9<br>9<br>9<br>9<br>9<br>9<br>9<br>9<br>9<br>9<br>9<br>9<br>9 | 2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>2000<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>20002<br>2 | X 000903c0X0090.X000<br>X 0 X4 X0<br>X 0 X4 X0<br>X 009903c0X00090.X000 |
| 1000000 ps 2000000   | os 300000        | fast raddr  | 600000 ps 70000  | 1 11<br>10 os 8000000 os   | 900000 05   |
| tos://file+ vscode-resource vscode   | -cdo.net/Lisers/ | lischerm/Downloads/oicory32 wrd   | 100000 ps 100000   | a participation participation  | Contraction in the  |
| cos//hie+vscode-resource,vscode  | -xdn.net/Users/I | ischemyDownloads/picory32.vcd   |  |  |   |
| 1000000 to 5000000   |                  |   |  |  | accoscos tx   |
|  |                  |   |  |  |   |
|  |                  |   |  |  |   |





## **PRE-SILICON VERIFICATION (HARDWARE)**





## **PRE-SILICON VERIFICATION (HARDWARE)**





### **AUTOMATED PROCESS**







## 4. CONCLUSION



### Security must be verified—not just designed

• Side-Channel and Fault Injection analysis are critical threats requiring proactive mitigation from RTL to silicon.

### Automation enables true lifecycle coverage

• Embedding physical analysis (SCA, FIA) into standard verification flows ensures early detection and validation throughout the SoC development process.

### Accelerate compliance and reduce risk

 Supports faster certification (CC, ISO, CAVP, FIPS 140-3) and reduces manual effort for design teams up to security sign-off.





## 5. SECURE-IC'S SOLUTIONS



### 3 TOOLS FOR SECURITY VERIFICATION ALONG DEVICE LIFEFCYCLE



CYCLE

LIFE

SECURITY

CYCLE LFE DESIGN









### THANK YOU FOR YOUR ATTENTION

#### CONTACTS

| EMEA     | sales-EMEA@secure-IC.com   |
|----------|----------------------------|
| APAC     | sales-APAC@secure-IC.com   |
| CHINA    | sales-CHINA@secure-IC.com  |
| JAPAN    | sales-JAPAN@secure-IC.com  |
| TAIWAN   | sales-TAIWAN@secure-IC.com |
| AMERICAS | sales-US@secure-IC.com     |

# FOLLOW US ON SOCIAL MEDIA

