Summary of Yassine Internship

Introduction

Subject

Embedded systems play a crucial role in both everyday life and defense applications, including drones, communication systems, and weapon control. Their widespread adoption, however, makes them vulnerable to hardware attacks—particularly SCA and FIA. SCAs exploit physical vulnerabilities such as power consumption and electromagnetic emissions, while FIAs deliberately disrupt system behavior through fault injection. This research focuses on detecting FIAs using security sensors, specifically addressing POTA where attackers target inactive sensors.

Laboratory Presentation

The LCIS Lab at Grenoble INP in France focuses on embedded systems and cybersecurity research. As part of the CTSYS team, the author contributed to projects aimed at protecting embedded systems from hardware attacks.

Internship Objective

The internship focused on analyzing existing FIA detection mechanisms, specifically delay-based and RO-based sensors, and developing improvements to protect circuits against POTA. Since these attacks target sensors during their inactive periods to bypass detection, the primary aim was to strengthen sensor robustness and enhance embedded system security.

State of the Art

Context

There are two main approaches protect against FIAs: redundancy-based and sensor-based methods. Redundancy-based methods detect faults by comparing duplicate computations but are vulnerable to localized attacks. Sensor-based methods use specialized hardware to monitor system behavior and come in two types: analog and digital. While analog sensors (like time-to-digital converters) offer high sensitivity, they are harder to calibrate and use more power. Digital sensors, including delay-based and RO-based detectors, are preferred for their simplicity and energy efficiency.

Delay-based detectors watch for timing violations and raise alarms when clock delays pass certain thresholds. Yet they struggle against precise attacks like laser or electromagnetic FIAs. RO-based detectors provide better accuracy by using inverter chains to generate frequency signals. These detectors count oscillations

during clock cycles and compare them to baseline values, triggering alerts when variations occur. This approach makes them more resilient against localized attacks.

Problem

Despite their effectiveness, these detectors have their own vulnerabilities. Attackers can physically tamper with the devices, inject faults to manipulate their measurements, or compromise their control software. These risks highlight the need for robust mechanisms to verify detector integrity.

Contribution

The solution proposes continuous sensor frequency validation through RO-based oscillation counting. The system detects attacks by comparing real-time oscillation counts against reference values. This monitoring method strengthens detector resilience, especially against precise attacks like laser FIAs.

Theoretical Study

Objective

The goal of the theoretical study was to develop a system capable of continuously monitoring the frequency of security sensors and detecting anomalies caused by attacks.This system would validate the stability of sensor frequencies over time, ensuring their proper functioning.

Requirements

The system needed to count sensor cycles within fixed intervals and compare the results to predefined reference values. This would allow it to detect deviations caused by attacks or other anomalies.

Functional Design

The system consists of four main components: a cycle counter, a timer, a register, and a comparator. The cycle counter counts the number of sensor cycles within each measurement interval defined by the timer. The register stores the cycle count, and the comparator checks if the count falls within the expected range. If the count deviates from the reference, the comparator triggers an alarm.

Key Formulas

The system relies on two key formulas to determine its parameters. The first formula calculates the minimum number of cycles N required to achieve a desired error margin:

$$N > \frac{F_{clk}}{\Delta f}$$

where F_{clk} is the clock frequency

and Δf is the acceptable error margin

The second formula determines the reference thresholds $X_{k\%}$ for detecting frequency changes:

$$X_{k\%} = \lfloor \frac{N \times (F_{cap} \pm k\%)}{F_{clk}} \rfloor \pm 1$$

where F_{cap} is the sensor frequency

and $k\%$ is the desired detection threshold ($\pm 5\%$ for example)

Implementation & Verification

System Implementation

The system was implemented in VHDL, with each component (timer, counter, register, and comparator) coded separately. The components were then integrated into a complete system and tested on an FPGA. The tests used a clock frequency of 50 MHz and sensor frequencies of 100 MHz, 200 MHz, and 1 GHz to validate the system's functionality.

Results

The system successfully validated the theoretical relationships for N and X. For N, the experiments confirmed that increasing the number of cycles reduces the error margin, as predicted by the formula. For X, the system accurately detected frequency deviations within the specified thresholds ($\pm 10\%$). The global functionality of the system was also validated, with the comparator correctly triggering alarms for out-of-threshold values.

Conclusion & Future Work

Conclusion

The study demonstrated the effectiveness of the proposed system in detecting frequency anomalies caused by attacks. The theoretical models were validated through experiments, confirming the system's ability to enhance the security of embedded systems against FIAs.

Future Work

Future work could focus on testing the system with smaller frequency variations to improve its sensitivity. Additionally, validating the system on physical FPGA hardware would ensure its functionality in real-world conditions. Finally, testing the system under actual attack scenarios, such as laser FIAs, would provide further insights into its robustness and effectiveness.

Glossary

SCA = Side-Channel Attacks

FIA = Fault Injection Attacks

POTA = Power-Off Temperature Attacks

LCIS = Laboratory of System Design and Integration