CSAW ESC 2023 Qualification Report

TRX Technical Labs - Sapienza

Leonardo Danella
University of Twente
leonardo.danella@hotmail.com

Kristjan Tarantelli
Sapienza University of Rome
tarantelli.kristjan@gmail.com

Francesco Bianchi
Sapienza University of Rome
f.bianchi202@gmail.com

Tiziano Caruana
Sapienza University of Rome
tizianocaruana@gmail.com

Emilio Coppa
Sapienza University of Rome
coppa@diag.uniroma1.it

***Abstract*—This report explores side channel attacks within the domain of Cyber-Physical Systems(CPS), exploring their potential threats, diverse attack methodologies, and the essential mitigation techniques required to curtail the leakage of sensitive information. As CPS continue to integrate physical processes with computational elements, their susceptibility to attacks becomes a pressing concern. Side channel attacks exploit unintended information leakage, such as power consumption patterns, timing variations, electromagnetic emissions, and acoustic signals, to infer critical data. This report provides insights into various attack approaches, including Timing, Power analysis and Cache attacks, offering concrete examples of how attackers might exploit these vulnerabilities. To counter these threats, it discusses mitigation strategies, such as Constant time programming, reducing the search space in s-boxes and cache randomization. Through a comprehensive understanding of side channel attacks and their potential ramifications, this report underscores the importance of holistic defense mechanisms to safeguard CPS infrastructure against emerging security challenges.**

***Index Terms*—side-channel attacks; cyber physical systems;**

I. Introduction

In today's interconnected world, Cyber-Physical Systems (CPS) have become integral components of critical infrastructure, spanning industries from energy and manufacturing to healthcare and transportation. CPS seamlessly blend physical processes with computational power and communication networks, enabling efficient automation, monitoring, and control. However, the fusion of these domains also brings forth a new realm of vulnerabilities, including the risk of side channel attacks that exploit unintended information leakage during system operation. This report explores the landscape of side channel attacks within the context of CPS, shedding light on the potential threats they pose, the diverse attack methodologies they employ, and the imperative mitigation strategies that must be employed to thwart these covert vulnerabilities.

CPS comprise a complex amalgamation of sensors, actuators, control units, and communication channels, all of which operate in concert to achieve specific goals. While these systems offer unprecedented capabilities and efficiency, their intricate nature introduces potential security gaps. One of the most insidious categories of attacks confronting CPS is side channel attacks. Unlike traditional attacks that primarily focus on exploiting software vulnerabilities, side channel attacks hinge on the physical characteristics of a system's operation, including power consumption, execution timing, electromagnetic emissions, and even acoustic emanations. These covert channels often operate beneath the surface, evading conventional security measures and thus targeting the very essence of CPS functionality.

This report aims to provide a comprehensive exploration of side channel attacks in CPS, elucidating the distinct attack vectors, the methodologies they employ, and the countermeasures that can be implemented to safeguard against these insidious threats. By delving into real-world examples and mitigation techniques, this report offers an essential guide to understanding, identifying, and defending against side channel attacks in the intricate landscape of Cyber-Physical Systems.

The subsequent sections of this report are organized as follows:

2) Side Channel Attacks in CPS: This section delves into the concept of side channel attacks, highlighting their relevance and potential impact on CPS security.
3) Different Attack Approaches: Here, the report explores various attack approaches, including power analysis, timing attacks, electromagnetic emissions, and acoustic signals, detailing how each avenue can be exploited by attackers.
4) Examples of Attacks and Mitigations: This section provides concrete examples of side channel attacks within CPS, accompanied by innovative mitigation strategies that can be employed to neutralize these threats.
5) Conclusion: The final section summarizes the key insights of the report, underlining the critical importance of a proactive defense stance against side channel attacks in CPS.

II. Side Channel Attacks in CPS

Side Channel Attacks (SCAs), refers to a category of attacks that exploit unintentional information leakage from a system, in this case, a CPS.

In the area, SCAs are particularly applicable due to the near integration of virtual (cyber) and physical (real-world) components. CPS combines sensors, controllers, communication networks, and physical processes to obtain automation and manipulate targets.

The interactions between these components create possibilities for attackers to exploit the physical characteristics of the device; And that's the reason why these attacks could have serious outcomes like unauthorized access, disruption of operations, data privacy breaches or safety concerns.

III. Different Attack Approaches

Here we are going to list some possible high level concepts of attacks and how they work, while in the next chapter we are going to see some particular exploitation techniques more in depth.
In particular in this chapter we are going to talk about:

1) Power Analysis on Cryptographic Module
2) Timing Attack on CPS Devices
3) Electromagnetic Emission Attack
4) Acoustic Side Channel Attack

1. Power Analysis on Cryptographic Module

Attackers analyze power consumption patterns of CPS devices during execution to deduce information about cryptographic keys, algorithms, or sensitive data. These attacks can be non-intrusive (monitoring power externally) or invasive (tampering with the device).

Simple Power Analysis (SPA) can be performed by examining a single run of an algorithm and recognizing correlations between power trace and code branches.

Differential Power Analysis (DPA) exploits correlations in the consumption traces by statically analyzing multiple executions of an algorithm.

Correlation Power Analysis (CPA) is a variant of DPA that specifically focuses on correlating power traces with intermediate values or hypotheses.

2. Timing Attack on CPS Devices

These attacks exploit variations in execution time of operations to extract information. For instance, an attacker could infer cryptographic keys by measuring the time taken to execute certain operations.

3. Electromagnetic Emission Attack

By monitoring EM radiation emitted during device operation, attackers can gain insights into the internal operations and data processing. This can lead to the extraction of cryptographic keys and sensitive data.

4. Acoustic Side Channel Attack

Devices emit sound during operation, which can be captured and analyzed by attackers. Certain activities like keystrokes or data processing can be inferred from these acoustic signals.

IV. Examples of Attacks and Mitigations

This section focuses on key aspects of side channel attacks in precise contexts and introduces the subsequent subsections that delve into specific attack approaches and mitigation strategies:

A. Attack: Timing attacks on modular exponentiations

Perhaps the most known example of a timing side-channel is the square-and- multiply algorithm, widely used in cryptography to evaluate (modular) exponentiations. Let us take into consideration the following implementation of the square and multiply algorithm in Python:

```
1 def square_and_multiply(b, exp, mod):
2     t = 1
3     while exp > 0:
4         if exp & 0b1 != 0:
5             t = (t * b) % mod
6         b = (b ** 2) % mod
7         exp >>= 1
8    return t % mod
```

This implementation of the square and multiply algorithm in Python allows us to perform exponentiation efficiently. However, it's important to note that the behavior of this algorithm can vary depending on the inputs. This can be seen particularly in the two edge cases: when the exponent is of the form 2^k, the "if" statement in line 4 is executed only once. Conversely, if the exponent is of the form 2^{k-1}, the same "if" statement is executed for every iteration.
An issue arises when an attacker can measure the algorithm's execution time on different inputs with a fixed exponent. By analyzing this timing information, they can potentially deduce the entire exponent, which can be sensitive information such as a private key or a secret nonce.

B. Mitigation: Constant-time programming

One effective software-based countermeasure to defend against time-based attacks is constant-time programming. In a constant-time routine, the control flow and data accesses are not influenced by the secret inputs. While enforcing constant-time behavior may seem challenging, some tools exist to transform non-constant-time programs into constant-time ones at compile time [1].
A simple countermeasure would be to transform the squaring into a multiplication of b by itself or to rewrite the multiplication as the difference of two squares, as was proposed in [2]. The problem with this approach lies in the fact that the exponentiation is still not constant time, so the Hamming weight of the exponent can still be leaked through timing analysis. A possible solution to this problem is to insert a "dummy" multiplication, making the routine constant time but also slower.

```
def square_and_multiply(b, exp, mod):
    t = 1
    while exp > 0:
        if exp & 0b1 != 0:
            t = (t * b) % mod
        else:
            _ = (t * t) % mod
        b = (b ** 2) % mod
        exp >>= 1
    return t % mod
```

A routine is considered to be constant-time if its control flow and data accesses are not influenced in any way by the inputs that are supposed to remain secret.

C. Mitigation: CRT to exponentiate

Another typical way to optimize the RSA implementation is to use the Chinese Remainder Theorem (CRT) to perform the exponentiation. With CRT, the function $M = C^d mod N$ is computed by first evaluating $M_1 = C^{d_1} \mod N$ and $M_2 = C^{d_2} \mod N$, where d_1 and d_2 are precomputed from d. M1 and M2 are then combined to yield M. RSA with CRT makes the original attack by Kocher [3] inoperative. Nevertheless, a timing attack can expose one of the factors of N, as illustrated by Brumley and Boneh [4].

D. Attack: Power analysis attacks on AES S-Boxes

Using either DPA or CPA, the confidentiality of AES encryption falls. By controlling the input and having access to the power consumption, we can reconstruct the secret key K used by AES.
In the DPA attack, we exploit the fact that, during encryption, an LSB (least significant bit) output of 1 consumes more power than LSB of 0. The difference is not visible to the naked eye, so we need a big sample of data to make this work. Having access to the power consumption of the S-Box lookup, we can then send all plaintext from $[00]*16$, $[01]*16$, ..., $[ff]*16$, and retrieving with the DPA attack the LSB of the S-Box lookup of the first round in AES. This is basically $Sbox[P_i \oplus K_i]$. At this point, we test all the possible value for each K_i and P_i, to find the correct subset of keys.
In CPA, instead, we assume that the power consumption is correlated to the Hamming Weight of input and output. Also in this case, by testing enough plaintext we can derive the number of bit flipped to 1 after the first round, thus with some extra work similar as before, recovering the key or a subset of the key space.

E. Mitigation: Search-Space Reduction for S-Boxes

A possible countermeasure is to use S-boxes with high confusion coefficient variance (CCV) in the space partitioned by Hamming weight (HW) classes, as proposed by Legon-Perez et al. in [5]. Bijective $n \times n$ S-boxes resilient to power attacks are hard to find in the space of dimension $2^n!$, specially as n increases, but the novel approach to reduce the search space by HW model equivalence classes allows to generate these S-boxes by class via an algorithm. The CCV theoretically measures the resistance of an S-box against power attacks and remains constant within each HW class.

Algorithm 1 SearchSpaceSboxReduction

Input: S-box s
Integer nss //Number of sets to be swapped
Integer mnos // Max number of outputs that can be swapped
Output: S-box r // HW equivalent with s

1: Select nss weights
2: **for** each k weight **do**
3: create two lists $Inputs[k]$ and $Outputs[k]$ // where each input holds in $Inputs[k], HW(s[input]) = k$
4: **for** each of the selected nss k weights **do**
5: shuffle(Outputs[k], mnos)
6: **for** $p = 0$ to $|C_k| - 1$ **do**
7: $r[Inputs[k][p]] = Outputs[k][p]$
8: return r

F. Attack: Flush+Reload Cache Attack

Cache attacks in the context of CPS can pose significant security threats and specific attack methods may vary depending on the particular CPS.
A well-known cache side-channel attack is Flush+Reload. The attacker monitors cache activity to infer which memory locations have been accessed by a victim process. It can be used to leak sensitive information, such as cryptographic keys, in CPS. This technique is a variant of the Prime+Probe and an attack consists of three phases. In the first phase, the attacker removes the monitored memory line from the cache. Then, in the second phase, the attacker waits for the victim to access the memory line. At last, the attacker reloads the memory line and measures the time it takes to load. If the victim accessed the memory line during the wait phase, the reload is fast because the line is already in the cache. If the victim did not access the memory line, the reload takes much longer as the line needs to be fetched from memory. This timing difference can reveal information to the attacker.
[6] shows how to use this technique to extract the components of the private key from the GnuPG implementation of RSA, by tracing the execution of the victim program and recognising the exponentiation steps. Sequences of Square-Reduce-Multiply-Reduce indicate a set bit of the exponent. Sequences of Square-Reduce which are not followed by Multiply indicate a clear bit.

G. Mitigation: Cache Randomization

Randomizing the mapping between addresses and cache indices disrupts the ability of attackers to easily create minimal eviction sets, crucial for contention-based cache attacks. Still, as [7] shows, randomizing the first-level caches (L1) enhances security but doesn't completely protect against all known attack methods like Prime-Prune-Probe. However, by combining L1 randomization with a lightweight random eviction strategy in higher-level caches, it's possible to mitigate well-known conflict-based cache attacks effectively.

V. Conclusion

In summary, this paper delved into the realm of side channel attacks in Cyber-Physical Systems (CPS). We explored various attack methods highlighting their potential threats to CPS security. By examining concrete examples and mitigation strategies, this paper underscores the importance of proactive defense measures in safeguarding CPS against these evolving attacks.

Depending on the particular setup we will observe in the final phase, our strategy will consist in the use of either software or hardware techniques. In the context of time side-channels and power side-channels attacks, we will construct specific systems using the supplied board to accurately measure fluctuations in time intervals or power usage during crucial operations.

References

[1] P. Borrello, D. C. D'Elia, L. Querzoni, and C. Giuffrida, "Constantine: Automatic side-channel resistance using efficient control and data flow linearization," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, Association for Computing Machinery, 2021.

[2] C. Negre and T. Plantard, "Efficient regular modular exponentiation using multiplicative half-size splitting," *Journal of Cryptographic Engineering*, vol. 7, pp. 245–253, Sep 2017.

[3] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conference*, pp. 104–113, Springer, 1996.

[4] D. Brumley and D. Boneh, "Remote timing attacks are practical," in *12th USENIX Security Symposium (USENIX Security 03)*, (Washington, D.C.), USENIX Association, Aug. 2003.

[5] C. M. Legón-Pérez, R. Sánchez-Muiña, D. Miyares-Moreno, Y. Bardaji-López, I. Martínez-Díaz, O. Rojas, and G. Sosa-Gómez, "Search-space reduction for s-boxes resilient to power attacks," *Applied Sciences*, vol. 11, no. 11, 2021.

[6] Y. Yarom and K. Falkner, "FLUSH+RELOAD: A high resolution, low noise, l3 cache Side-Channel attack," in *23rd USENIX Security Symposium (USENIX Security 14)*, (San Diego, CA), pp. 719–732, USENIX Association, Aug. 2014.

[7] A. Jaamoum, T. Hiscock, and G. Di Natale, "Noise-free security assessment of eviction set construction algorithms with randomized caches," *Applied Sciences*, vol. 12, no. 5, 2022.