IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Hardware-Level Defenses Against Side-Channel Attacks: Mitigating Power and Timing Vulnerabilities in Cryptographic Chips

Dr. Amrapali Chavan, Om Patil, Krushna Renuke, Yash Raut, Sudarshan Narhare

Department of Artificial Intelligence & Data Science, AISSMS Institute of Information Technology, Pune, India

Abstract: Side channel attacks (SCA) attack the hardware implementation of cryptography, but not the algorithms themselves. The attacks involve monitoring physical parameters that are affected by secret key operations in the hardware implementation and using statistical correlation techniques to recover the keys. Cryptographic chips used in embedded systems, smart cards, hardware security modules (HSMs), and SoCs are especially susceptible due to their limited physical environment. This paper reviews the various hardware techniques for defending against SCAs on power and timing channels. Wave Dynamic Differential Logic (WDDL), hardware masking techniques based on Boolean secret sharing and Threshold Implementations, constant time architecture design, and clock randomization are among those studied. These methods are analyzed for their effectiveness, cost in terms of area and performance overhead, scalability, and residual leaks. A comparative analysis will show the pros and cons of each technique in terms of PPA. Future directions for research include composable defense strategies and leakage verification for SCA-resilient cryptographic hardware.

Index Terms—Side-Channel Attacks; Differential Power Analysis; WDDL; Hardware Masking; Constant-Time Design; Clock Randomization; Cryptographic Hardware Security.

I. *INTRODUCTION*

Traditionally, the assessment of security in cryptographic systems is done based on a black-box model, where a potential adversary can interact with a cipher through its plaintext and ciphertext interfaces alone. Nevertheless, when implementing cryptographic algorithms in hardware, side effects like power consumption and execution time occur, which can be used to obtain secret information. Unintended information paths of this sort are referred to as side-channels, and attacks exploiting side-channel effects are known as side-channel attacks (SCAs).

The existence of timing attack vulnerabilities was established early on, as variations in the execution times of cryptographic algorithms were shown to be able to reveal secret data, especially in public-key encryption schemes such as RSA. Research conducted on power analysis later confirmed that instantaneous power consumption correlated with the values of intermediate data, making it possible to extract secret keys statistically from measurement traces [15], [18]. This made the analysis of hardware security shift from traditional considerations of performance and area into an era where physical leakage became a prime concern.

SCA attacks have been successfully performed on implementations of AES [3], as well as more general cryptographic protocols and embedded systems [10], [2]. In particular, SCA attacks can completely compromise the secret keys of cryptographic schemes, thus violating the security guarantees provided by them. Countermeasures against SCAs can be applied in various levels of abstraction, from software implementation and algorithms to hardware itself. For applications requiring a high assurance level, hardware countermeasures are essential since they operate independently of any software.

Prior work includes research on masking algorithms [5], [7], composable secure hardware designs [1], [13], and advanced SCA evaluation methods [12]. This paper deals with four main categories of hardware countermeasures: dual rail logic (WDDL), hardware masking, constant-time design, and clock randomization. This paper is structured as follows. Section II provides background and prior art review. Section III presents our threat model. Section IV discusses the system architecture. Section V describes the countermeasures. Sections VI, VII, and VIII give a comparative analysis, describe trade-offs, and conclude the paper

II. *BACKGROUND AND RELATED WORK*

The Physical cryptanalysis development is tightly connected to research into leaks from implementation levels of cryptography algorithms. First, it was revealed that secret-dependent behavior of algorithms can be detected via side-channel leaks [15], [16], [18]. Next development included the invention of statistical and machine learning approaches for power traces analysis, becoming one of the essential components of today's side-channel attack frameworks [15], [16], [18]. New discoveries stimulated further advancements in both areas. Recent developments resulted in practical side-channel attacks on embedded systems and microcontrollers using non-invasive measuring techniques [2]. Attacks on proprietary protocols and optimized implementations demonstrate widening attack targets [10].

Several hardware countermeasures have been designed to deal with those attacks. Masking technique, together with its high-order variants, attempts to decorrelate intermediate secret variables with leaks [5], [7]. Hardware composition techniques have also been created to protect systems against probing and composition-based attacks [1], [13]. Research into cache and timing channel allowed for additional understanding of constant time architecture [12].Therefore, today's state-of-the-art indicates that arms race between evolving attacks and countermeasures goes on.

III. *THREAT MODEL: POWER AND TIMING ATTACKS*

A. Adversary Model

Figure 1 below shows the generalized side channel attack model showing the three major forms of leakage channels available to an attacker, which include power consumption, execution time, and electromagnetic emissions..

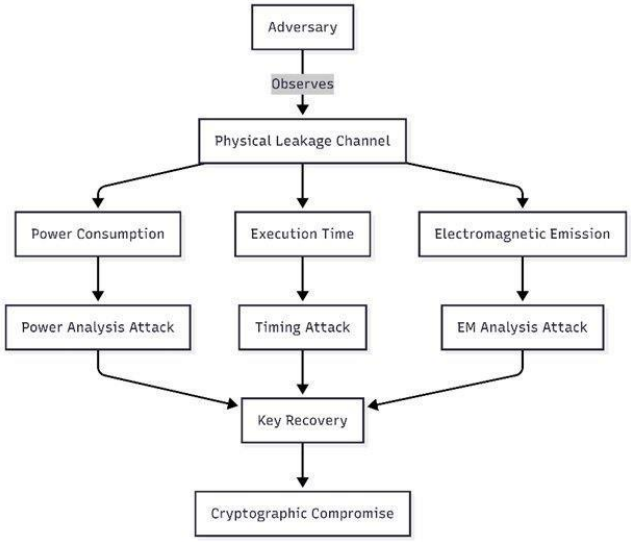

Fig. 1. *Side-channel attack model showing leakage channels and attacks leading to key recovery.*

The attack model employed in this paper employs the traditional non-invasive and passive adversary setting. This entails the adversary having access to the targeted device physically and employing commercially available tools to analyze the energy consumption or execution times of the targeted device. The adversary is aware of the cipher algorithm under implementation and may supply chosen plaintexts or ciphertexts.

B. Simple Power Analysis (SPA)

SPA requires the examination of only one power trace to deduce something about the secret key. For CMOS circuits, the amount of dynamic power dissipated for each switch operation is provided by Equation (1):

$$P_{dyn} = \alpha \cdot C_L \cdot V_{DD}^2 \cdot f \qquad (1)$$

where α is the activity factor (the fraction of cycles where there is a switch), C_L is the load capacitance, V_DD is the supply voltage, and f is the clock frequency. As α is dependent on the data, there is an inherent information leakage in the power trace. In SPA, the attacker correlates the observed characteristics, such as the Hamming weight of data bus changes, to specific key-dependent calculations. SPA is particularly successful when used against schemes that have unique power consumption profiles, like RSA with its square-and-multiply method.

C. Differential Power Analysis (DPA)

DPA is a more sophisticated statistical attack using many power traces and a presumed model of power consumption for recovering secret keys. The attacker divides a collection of N traces into two groups according to a predicted intermediate value that is dependent upon the assumed key hypothesis k*. For an intermediate bit b of V(P, k), the differential trace is represented by Equation (2):

$$D(t) = \frac{1}{|S_1|} \sum_{i \in S_1} T_i(t) - \frac{1}{|S_0|} \sum_{i \in S_0} T_i(t) \qquad (2)$$

where S_1 and S_0 represent the trace index sets where the predicted value of the bit is 1 and 0, respectively. A non-zero spike of D(t) at a certain time indicates that the guessed key k* is correct. Higher order differential power analysis (HO-DPA) utilizes several times or operations.

D. Timing Attacks

Timing attacks take advantage of changes in the execution time dependent upon the data value. In RSA where square-and-multiply algorithm is employed, a bit equal to 1 in the private exponent means an extra multiplication is done resulting in extra execution time. By sending a large number of queries on selected inputs, an adversary can statistically estimate the value of the bits of the exponent.

Timing attacks in symmetric algorithms occur because of cache-based access patterns, data dependent conditionals and non-constant time look-ups. Timing attacks on AES [12] proved that time taken for accessing tables that are dependent on key values can be used to infer key-dependent addresses, hence reconstructing the whole key. For securing a cipher from all kinds of timing side channels, one must make sure that I(K;T)=0.

IV. *SYSTEM ARCHITECTURE OVERVIEW*

The anti-side-channel attack cryptochip architecture is created based on the four countermeasures proposed in this paper. The architectural structure is split into two layers: core cryptographic path and countermeasures path that includes WDDL cell library, masking logic unit, constant-time ALU, and clock randomization circuits. Entropy for masking and clock randomization PLLs is generated by the true random number generator.

V. *HARDWARE-LEVEL COUNTERMEASURES*

A. Wave Dynamic Differential Logic (WDDL)

Wave Dynamic Differential Logic [7] is an approach based on dual-rail pre-charge logic which attempts to reduce switching activity on CMOS circuits due to data variations. According to the model, any logic signal s can be represented as a complementary pair (s, $\bar{s}$) where only one rail would be transitioning throughout each evaluation period regardless of the function performed. Therefore, switching activity and dynamic power will remain constant regardless of the data variation.

According to WDDL, any gate can be constructed as a pair of complementary cells representing both function and its inverse. In such a way, the AND gate generating output value f=A•B would be able to calculate not only f =A • B but $\bar{f} = \bar{A} + \bar{B}$. The circuit is divided into two stages: pre-charge and evaluation, ensuring that all outputs reach constant values no matter the data. In such a way, WDDL would ensure fixed number of transitions per clock cycle which makes activity coefficient α in equation (1) independent of input data.

However, WDDL suffers from inability to provide exact balance because of unbalance of parasitics in terms of wiring. In case capacitive loads differ, this effect results in additional switching and is referred to as early propagation. Certain layout-based routing approaches could be applied to mitigate this issue. Nevertheless, AES implementations protected with WDDL exhibit significantly higher resistance against DPA attack requiring magnitudes more measurements to be successful [7]. Major drawbacks include double cell area and dynamic power consumption.

B. Hardware Masking

Information theoretical approaches to side-channel protection include hardware masking schemes, where intermediate values are randomized such that it would not be possible to deduce unmasked data from the power traces. The Boolean first order masking scheme uses two shares, a random number r selected uniformly in its range, and v'=v⊕r. At no point during the computation, will the circuit operate using unmasked v. In the d-th order masking scheme, it is possible to express the secret v as $v=v_0 \oplus v_1 \oplus \ldots v\_d$. Therefore, if a side-channel analysis tries to extract information using d intermediate values, then we know it will be impossible since any selection of d-tuple shares will be uniformly distributed, and therefore independent of v.

There are cases when masking algorithms have to be designed in a way to avoid any glitch conditions from causing incorrect share combinations. This can be accomplished through threshold implementations (TI). According to Wegener et al., a TI is a provably secure algorithm that ensures the property of non-completeness, i.e. the circuit should not contain components which depend on the computation of the d+1 shares of a secret variable. TI has been successfully employed on AES and PRESENT.

C. Constant-Time Architectures

Constant time design means that the time taken by each crypto operation is independent of the secret key and data being processed. This involves using only branchless logic at RTL level. The general method for doing so is to calculate both possible results for a condition and then select the right one using data independent multiplexing. The conditional assignment:

$$if\ (key\_bit == 1)\ \ then\ \ result = A\ \ else\ \ result = B \qquad (3)$$

is replaced by:

$$result\ =\ (key\_bit\ \&\ A)\ \mid\ (\sim key\_bit\ \&\ B) \qquad (4)$$

which executes in the same number of clock cycles for any value of key_bit.

The other cause for timing variations can be memory access. In table-based AES encryption, the SubBytes operation relies on the S-box table for substitution of data bytes. In case the S-box table is located in the cacheable region, then the access to this memory will depend on cache lines, thereby making it susceptible to cache-based side channels. A constant-time approach substitutes this cache access method with a bitsliced implementation of the S-box, which was proposed previously [13].

D. Clock Randomization

Random clocking is a method whereby temporal noise is added to the power trace by means of randomizing the clock rate or the number of operations. The purpose of random clocking is to decouple the clock cycles with the power traces that are recorded, such that the attacker is unable to make use of correlation techniques based on time alignment.

There are two types of random clock implementation: in the first method, a phase locked loop (PLL) or a ring oscillator is modified to create a random clock that has bounded jitter, with the help of TRNG which adds random bits to the input voltage controlled oscillator (VCO). In the second approach, random dummy clock pulses are injected through a secure finite state machine using TRNG.

The effect of random clocking as a singular countermeasure against DPA attacks is somewhat limited since an adversary having sufficient number of traces could synchronize misaligned power traces through various techniques such as dynamic time warping or pattern matching. However, when random clocking is used in conjunction with masking or WDDL, the security is greatly enhanced and the attack becomes computationally impractical. The hardware overhead of this method is very small (5-20% area).

VI. *DESIGN METHODOLOGY AND IMPLEMENTATION FLOW*

The suggested flow for implementing SCA-robust cryptosystems on hardware platforms is presented in Fig. 2. It begins with the identification of threat models, followed by the selection of countermeasures and the implementation of the latter in RTL, and ends with the verification of leakage via the leakage evaluation framework of [13].

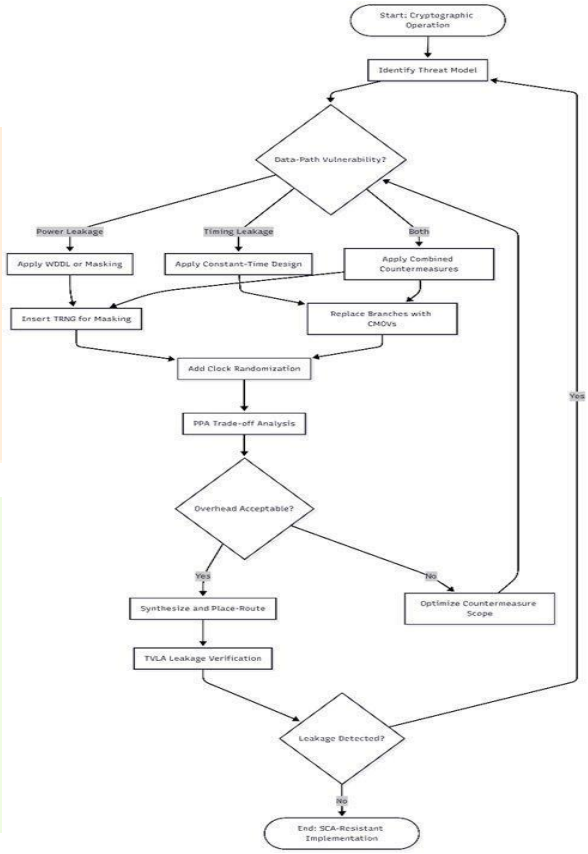

Fig. 2. *SCA-resistant hardware design flow: threats, countermeasures, and TVLA validation.*

VII. *COMPARATIVE ANALYSIS*

Table I compares the four hardware-based defenses examined in this paper based on the following evaluation criteria: the types of attacks that can be mitigated by each defense; the cost in terms of area and power overhead of the hardware required to implement the defense; the complexity of implementing the defense; and the remaining leakage, if any, following deployment of the defense.

TABLE I Comparative Analysis of Hardware SCA Countermeasures

Countermeasure	Threats Mitigated	HW Cost (Area / Power)	Impl. Complexity	Residual Leakage
WDDL / Dual-Rail Logic	DPA, SPA	~100% area; ~50–100% power	High (routing)	Low–Moderate
Hardware Masking	DPA, HO-DPA	~20–40% area; ~15–30% power	Moderate–High	Low (order-d secure)
Constant-Time Design	Timing Attacks	~5–15% power; low area	Moderate	Very Low
Clock Randomization	Timing, DPA (partial)	~5–20% area	Low–Moderate	Moderate

The WDDL approach provides an architectural solution to DPA by removing data-dependent switching within cells. It relies on physical principles of complementary capacitance balance instead of randomness. However, its security guarantee decreases in real-world scenarios because of routing-related asymmetries; thus, WDDL becomes vulnerable to attacks that utilize residual glitch-based leakage. In addition, the approximately twofold area overhead incurred is a major cost associated with large cryptographic cores.

Hardware masking, especially TI, yields the most powerful theoretical security properties among all approaches mentioned. TI guarantees provable resistance to side-channel attacks until a certain attack order d. The only drawback is that TI imposes requirements regarding on-chip randomness; for example, a first-order TI of AES needs 128-bits of fresh randomness for each encrypted block. The amount of randomness and required circuitry exponentially increase with higher orders of masking. TI is usually unfeasible for embedded systems.

Constant-time design solves the problem of timing attacks effectively and with a relatively small overhead. Its impact on power analysis is modest since removing branches will eliminate only some SPA-visible control flow instructions, but it will not decorrelate data and power consumption. Thus, constant-time design is a required base method, which can be applied together with hardware masking or WDDL.

Clock randomization can be viewed as a low-cost supplementary solution that adds complexity to trace alignment in statistical side-channel attacks. Security provided by this approach is practical but not formal. Clock randomization in conjunction with hardware masking or WDDL requires adversaries to decompose masked values and align traces simultaneously.

VIII. *PERFORMANCE VS. SECURITY: PPA TRADE-OFFS*

The cost of implementing SCA countermeasures at the hardware level is associated with increased power consumption, performance reduction, and area usage - collectively known as PPA metrics in VLSI circuit design.

The WDDL countermeasure is the most expensive in terms of both area and power. The dual-rail cell library doubles the number of gates of any combinational logic. A study on the 90 nm CMOS AES implementation found that the overhead was about 100% for area and 50-100% for power [7]. As for performance, there is a noticeable decrease because pre-charging followed by the evaluation cycle halved throughput without the help of pipelining.

The overhead for hardware masking countermeasures consists of the overhead associated with implementing the logic used for masked calculations. A first-order implementation of the masked AES algorithm adds 20-40% to the area and 15-30% to power overhead against unprotected hardware. Higher-order masking algorithms will require much larger overheads. The added TRNG is also part of the system overhead cost.

Constant-time changes are the least expensive countermeasures, especially if an ASIC is built for constant time execution from scratch. The overhead is 5-15% increase in power and marginal area increase if replacing conditional branches and using branchless algorithms and algebraic S-boxes. The bitsliced AES is efficient on parallel computing architecture [4].

Clock randomization introduces minimal overhead in the form of a TRNG, but the biggest PPA cost comes from modifying the PLL. Jitter within ±10-20% of the nominal clock period guarantees good desynchronization performance while causing less than 5% performance overhead.

IX. *CONCLUSION*

In this paper, we review hardware-based side-channel countermeasures for cryptographic chips. We focus mainly on protection from power analysis and timing-based attacks. While there are several different approaches to preventing side channels, including WDDL, hardware masking, constant-time design, and clock randomization, each addresses different aspects of side channel leakage and provides its own set of security guarantees. WDDL seeks to achieve an equal number of switchings, but achieving this goal might be

difficult in modern CMOS circuits. On the other hand, hardware masking, especially using Threshold Implementations, ensures high formal security but at the cost of extra area usage and randomness requirements. Constant-time design prevents any timing-based leakage at the architecture level and is a must in any cryptographic system implementation. Finally, clock randomization can serve as an additional low-cost method that does not provide any formal security guarantees but adds to attack complexity.

It is important to note that there is always a trade-off between security strength and the costs incurred by countermeasures. Thus, while one method could suffice on paper, no single method would be enough in practice to protect a cryptographic system.

REFERENCES

[1] G. Cassiers, B. Gregoire, I. Levi, and F.-X. Standaert, "Hardware private circuits: From trivial composition to full verification," IEEE Trans. Comput., vol. 71, no. 3, pp. 584–597, Mar. 2022.

[2] L. Wouters, J. Gierlichs, and B. Preneel, "On the susceptibility of Texas Instruments SimpleLink platform microcontrollers to non-invasive physical attacks," IACR Trans. Cryptographic Hardware and Embedded Systems, vol. 2022, no. 1, pp. 317–356, 2022.

[3] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "A cautionary note regarding evaluation of AES hardware implementations," in Proc. AES Candidate Conf., 2022, pp. 133–147.

[4] M. Azouaoui et al., "Bitslice masking and improved shuffling," IACR TCHES, vol. 2022, no. 2, pp. 140–165, 2022.

[5] A. Rezaei Shahmirzadi and A. Moradi, "Second-order SCA security with almost no fresh randomness," IACR TCHES, vol. 2021, no. 3, pp. 708–755, 2021.

[6] J. Breier, X. Hou, and S. Bhasin, "SNIFF: Reverse engineering of neural networks with fault attacks," IEEE Trans. Reliab., vol. 71, no. 4, pp. 1527–1539, 2022.
[7] P. Sasdrich et al., "Low-latency hardware masking with application to AES," IACR TCHES, vol. 2020, no. 2, pp. 300–326, 2020.
[8] D. Kubovčík and J. Brederlow, "Combined power and fault analysis attacks," in IEEE HOST, 2023, pp. 55–65.
[9] A. Abdulgadir et al., "A side-channel-resistant implementation of ASCON," ACM JETC, vol. 19, no. 1, pp. 1–22, 2023.
[10] L. Chmielewski and L. Batina, "SCA strikes back," in IEEE EuroS&PW, 2023, pp. 142–151.
[11] F. Berti et al., "Protected modes of operation and masking," IACR TCHES, vol. 2023, no. 2, pp. 112–145, 2023.
[12] J. Kramer, M. Fyrbiak, and C. Paar, "Cache-side-channel countermeasures survey," ACM Comput. Surv., vol. 56, no. 4, pp. 1–38, 2024.
[13] G. Cassiers and F.-X. Standaert, "Towards globally secure masked circuits," IACR TCHES, vol. 2021, no. 2, pp. 1–24, 2021.
[14] G. Cassiers and F.-X. Standaert, "Composing masked gadgets with probe isolating non-interference," IEEE TIFS, vol. 15, pp. 2542–2555, 2020.
[15] R. Benadjila et al., "Deep learning for side-channel analysis and ASCAD," J. Cryptographic Engineering, vol. 10, no. 2, pp. 163–188, 2020.
[16] A. Heuser et al., "Machine learning for side-channel analysis: Advances and challenges," ACM Comput. Surv., vol. 55, no. 1, pp. 1–36, 2023.
[17] V. Grosso, O. Bronchain, and F.-X. Standaert, "Hardware masking vs. glitches," IACR TCHES, vol. 2023, no. 3, pp. 95–120, 2023.
[18] O. Bronchain and F.-X. Standaert, "Breaking masked implementations with few traces using deep learning," IEEE TIFS, vol. 17, pp. 1903–1916, 2022.
[19] M. Nassar et al., "Efficient countermeasures against higher-order side-channel attacks," IEEE Trans. Comput., vol. 73, no. 2, pp. 345–358, 2024.
[20] J. Balasch, F.-X. Standaert, and G. Cassiers, "Composability and formal verification of side-channel secure hardware," IACR TCHES, 2025.