Spring 2020

Comprehensive Designs of Innovate Secure Hardware Devices against Machine Learning Attacks and Power Analysis Attacks

Yiming Wen
Old Dominion University, ywen001@odu.edu

COMPREHENSIVE DESIGNS OF INNOVATE SECURE HARDWARE DEVICES AGAINST MACHINE LEARNING ATTACKS AND POWER ANALYSIS ATTACKS

by

Yiming Wen
B.S. June 2013, Central South University (China)
M.S. June 2016, University of South Florida

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
February 2020

Approved by:

Weize Yu (Director)

Chung-Hao Chen (Member)

Cong Wang (Member)

Jiang Li (Member)

ABSTRACT

COMPREHENSIVE DESIGNS OF INNOVATE SECURE HARDWARE DEVICES AGAINST MACHINE LEARNING ATTACKS AND POWER ANALYSIS ATTACKS

Yiming Wen
Old Dominion University, 2020
Director: Dr. Weize Yu

Hardware security is an innovate subject oriented from growing demands of cybersecurity and new information vulnerabilities from physical leakages on hardware devices. However, the mainstream of hardware manufacturing industry is still taking benefits of products and the performance of chips as priority, restricting the design of hardware secure countermeasures under a compromise to a finite expense of overheads. Consider the development trend of hardware industries and state-of-the-art researches of architecture designs, this dissertation proposes some new physical unclonable function (PUF) designs as countermeasures to side-channel attacks (SCA) and machine learning (ML) attacks simultaneously. Except for the joint consideration of hardware and software vulnerabilities, those designs also take efficiencies and overhead problems into consideration, making the new-style of PUF more possible to be merged into current chips as well as their design concepts. While the growth of artificial intelligence and machine-learning techniques dominate the researching trends of Internet of things (IoT) industry, some mainstream architectures of neural networks are implemented as hypothetical attacking model, whose results are used as references for further lifting the performance, the security level, and the efficiency in lateral studies. In addition, a study of implementation of neural networks on hardware designs is proposed, this realized the initial attempt to introduce AI techniques to the designs of voltage regulation (VR). All

aforementioned works are demonstrated to be of robustness to threats with corresponding power attack tests or ML attack tests. Some conceptional models are proposed in the last of the dissertation as future plans so as to realize secure on-chip ML models and hardware countermeasures to hybrid threats.

Dedicated to my mom who gives me encouragement
as well as my best friends who give me company
throughout my fascinating learning experience in United States

ACKNOWLEDGEMENTS

Thanks to unpredictable life and sparks of human nature. Ten years ago, when I was first out for a journey to the society, never did I imagine I would find a career that I would love so much and a life goal that I would fully devote myself to. Thanks to those stories as well as characters included that make me who I am today.

My first special appreciation is devoted to my advisor Dr. Weize Yu. It must be written in my fate to find an eminent mentor like him. His unfailing instruction helps me overcome difficulties one after another. Your accompany beyond academics released my loneliness and we are just like family throughout three years in Norfolk. No words can express my grateful thanks to your patience as my advisor.

Special thanks to all members in my dissertation committee: Dr. Chung-Hao Chen, Dr. Jiang Li, and Dr. Cong Wang. It is my great honor to invite them in my last trip as a Ph.D student and the origin as an engineer. Their rigorous attitude to research and teaching inspires me a lot. I would like to thank them for their invaluable time, patience, comments, and encouragement.

Thanks to my friends, Youwen Min, Zheng Qu, Xushi Wang, and Hongbo Pan. It is already 15 years since we first met. Although we hardly have time to gossip, it is always a great experience to share collected stories every time when I am back home. Their en-couragement and suggestion help me stand firmly for my convictions, and I will treasure our friendship as wealth for a lifetime.

Finally, thanks for the supports and accompany of my mother. I first thought my invitation of your helps you to escape trifles of life. However, when I finish my final word in this work, I realize it is because you are here, that I can focus fully on my work and find a goal to struggle for. I love you, mom!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 HARDWARE SECURITY

What is the trend of computer technology in the next decade? This holds a big question mark for researchers for almost a century. The answer changes dramatically: from a huge monster ENIAC to a cute Alexa set, from the rising of Moore's Law to the prototype of quantum computer, and even from an Intel 4004 with frequency at 104 kHz to an AMD Threadripper 3970 with frequency at 4.6GHz. Questions come and go. The secure system, however, holds its extraordinary vitality in every generation and grows up to a prosperous subject, cybersecurity, today. Except for traditional software secure methods that are still under updating to meet our current Internet environment, some innovate hardware-based threats are recently proposed and request higher demands on hardware designs. Due to the new hardware attacking methods and the increasing distribution of Internet of things (IoT) devices, hardware security is proposed as an extended protection solution of conventional software-based cryptographic system.

Different from conventional software-based threats, hardware threats aim at seeking for vulnerabilities in signal leakage, design defects, and/or insert malicious hardware devices to bypass software countermeasures in higher network layers. Due to the design demand for better performance, most current chip designs tend to think little of protections of hardware signals. Those physical leakages, in return, can be used for mathematical analysis and for

further extraction of confidential information. Consequently, some countermeasures are proposed to hide/encrypt those secrets in physical designs.

Although some models of threats and their countermeasures are presented and proved to be of potential research value to cybersecurity studies, the industries are still not aware of the importance of hardware protections in some ways. First of all, the investment-yield ratio is still the prior consideration of the market. The pursuit of Moore's law is not just for the need of higher performance but also for the commercial competition in the chip market. If a secure hardware design consumes too much system source such as area, power, heating, etc, the reduction of performance will drive consumers to select other brands. Moreover, hardware threats have not caused severe secure information panic. This is mainly because hardware-based attacks are usually stealthy and requires ancillary equipment to extract and analyze physical signals, which makes information stealing from hardware difficult to ordinary people. In addition, some hardware attacks like hardware Trojan can only be implemented during fabrications. This means the reveal of vulnerabilities can only be achieved by designers. The detection of the information leakage would cause excess costs.

Opposite to the sluggish reaction of the market, hardware threats develop greatly due to the growth of advanced technologies. On the one hand, some work already proved that physical threats combined with high-level anti-detection designs are more aggressive to confidential information [8, 9, 10]. On the other hand, machine learning (ML) introduces some innovate vulnerabilities, giving even more challenges to secure hardware designs [11, 12]. As a result, comprehensive hardware security designs with considerations of both hardware and software attacks would be the final solution to cater to the market. Limited by those

conditions, it can be foreseen that future secure hardware devices should be of features that include but not limit to:

- Design on existing functional devices instead of independent secure hardware modules. Although independent hardware modules provide more secure protection mechanisms, the overheads of speed, power, and area require hardware designers to sacrifice some hardware performance to achieve the encryption standard.

- Considerations of jointly resisting side-channel attacks (SCA) and ML attacks. With more IoT devices applied and distributed under unsupervised, leaked physical signal can be targeted and extracted more easily. With the growth of ML technologies, traditional countermeasures to SCA need to be redesigned to confront ML attacks.

- Detection of hardware Trojan need to be aware. Some pre-implemented hardware Trojans tend to negate cryptographic modules.

1.2 SIDE-CHANNEL ATTACKS

Oriented from the natural flaws of switching characteristics in modern complementary metal oxide semiconductors (CMOS), some features or functionality can be analyzed with leaked physical signals. Depending on measurement equipment that is used by attackers, heat, power, time delay, and many more measurement dimensions can be monitored by attackers. Many published papers have proved that side channel attacks are of higher efficiency to modern encryption systems [13, 14, 15, 16]. Among all attacking techniques of side channel attacks, power analysis attacks are most referred and studied. Depending on

complexity of attacks and types of analyzed power, power analysis attacks can be categorized as simple power analysis, differential power analysis (DPA), and leakage power analysis (LPA). Fig. 1.1 shows an experimental result of simple power analysis attacks [1]. In this early study, it has been proved that the Hamming weight of the byte being processed is proportional to the height of the power consumption pulse. However, due to the awareness of power analysis attacks, current hardware designs are usually attached with SCA-resistant designs at either circuit level and architecture level [17]. As a result, DPA and LPA are mainstream researches of current side-channel attacks in which the former focuses on analysis of relationship between confidential information and dynamic power consumption while the latter cares more on mathematical analysis on static power traces. Comparing to DPA attacks, LPA attacks are more dangerous to our cryptographic systems. Djukanovic *et al.* [18] perform LPA attacks on various DPA-resistant logic styles and reveal LPA attacks are effective in extracting confidential information in both CMOS bit sliced circuits and CMOS combinational circuits (e.g., S-boxes).

Conventional countermeasures to power analysis attacks emphasize the reduction of dependency of side-channel leakage and power consumption profile. The power consumption, first of all, can be mitigated by redesigning on CMOS devices. A well designed magnetic tunnel junction CMOS can produce uniform power consumption during operation [19]. By introducing user-defined security metric using constrained state assignment, the power footprint can be encoded [20], making it impossible to reveal the correlation between confidential information and leaked power. A multi-core processor combining Random Dynamic Task Scheduling, Random Dynamic Frequency Scaling, and Random Dynamic Phase Adjustment

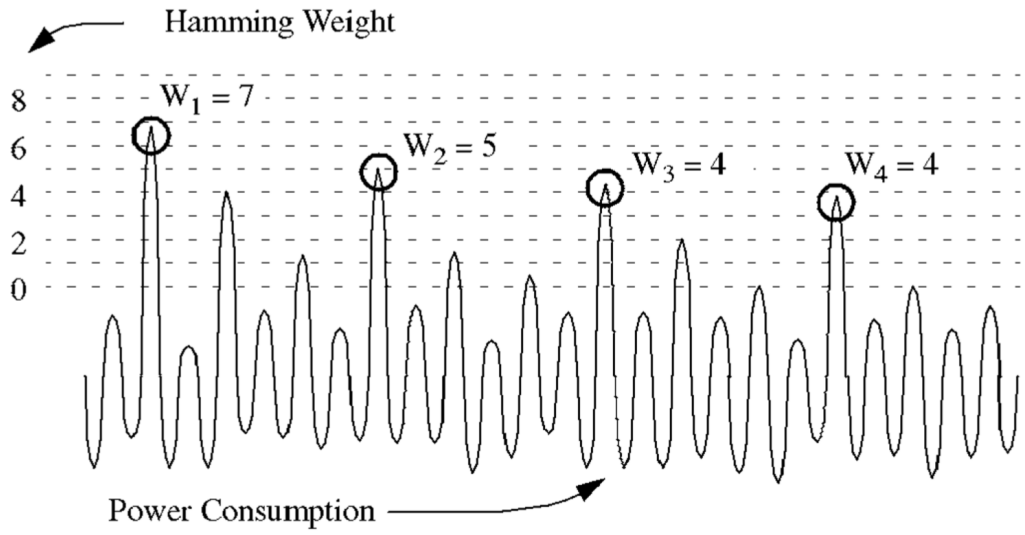


Fig. 1.1: Hamming weight information can be extracted from leaked signal of power consumption [1].

is proved to be of resistance to DPA attacks [21]. Although randomized input can be utilized to arbitrary initial input values on an AES circuit [22], innovate power analysis methods using convolutional neural networks [23, 24, 25] are still challenging hardware designers to push the complexity of secure designs to a higher level. Other than the competition to advanced attacking approaches, another straits in designing countermeasures to SCA is the limitation of power budget and area restriction. Some researchers raise their ideas on low-overhead secure designs to particular side-channel attack method [26, 27]. However, to design an architecture to obstruct all potential physical leakages is still expensive.

1.3 PHYSICAL UNCLONABLE FUNCTION DEVICES

Physical Unclonable Function (PUF) devices are innovate circuit primitives that exploit inherent manufacturing profess variations as natural random variables to realize some secure functions in physical level [28]. According to the specific architecture design, different

physical parameters, such as voltage, time delay, frequency, etc., can be utilized as referred variables in comparison. In this way, binary input sequences drive the designed PUF primitive to generate random outputs, which form the core security parameter called challenge response pairs (CRPs). On the basis of the scale of generated CRPs, a weak PUF is defined as a PUF that generates a countable set of obtainable CRPs, while a PUF primitive that generate relative infinite CRPs is called strong PUF. With their inherent random feature, the internal arithmetic logic and characteristic parameter are usually unpredictable until the measurement collection after fabrication. Thus, each PUF is unique to its circuit layout and is hard to replicate in practice. This makes PUF devices excellent candidacy in security applications such as authentication, key generation, and noise generation [28, 29].

Up to date, variant PUF primitives are invented to meet the demand of different scenarios. Arbiter PUF [30] utilize delay features in logic gates and realize early strong PUF. A SRAM-PUF uses the power-up states of a SRAM cell to realize logical PUF design, making it a practical ID/key generator in micro-controllers. Besides, ring-oscillator (RO) PUFs are inspired by its multi-input and lightweight features. It is also widely used as protection mechanisms in distributed wireless sensor networks.

Nonetheless, associating with the spreading application of PUFs and the design of new PUF primitives, what we cannot ignore are potential risks under attacks and the demand of market. Initially, most proposed PUF designs rely on highly linear comparing logic, which is usually vulnerable to machine learning attacks. Moreover, PUF devices are always implemented as an independent security unit instead of a functional module that promote performance of devices. This will obviously introduce considerable overhead and occupy

system resources. Consider the contradiction to the pursuit of profits, only devices that requires special protection or are exposed to a vulnerable environment will be allocated with protections of PUFs. With the quickening spreading of IoT devices, more intelligent devices will become preys of potential malicious attacks. Consequently, there is an urgent need for a bran-new design standard for PUFs to accommodate the demand of security and overhead.

1.4 MACHINE LEARNING

Machine learning (ML) is an interdisciplinary subject that exploits theories of probability, statistics, approximation, convex analysis, etc. to endow computers machines to imitate human's learning process from prior experience in order to obtain new knowledge or desired functions. As a cutting edge technology in artificial intelligence, ML has been proved to be beneficial in applications of medicine, genetics, data mining, and more promising research fields [31, 32, 33]. According to the demand of tasks, data analysts are capable to design their own neural networks and inject validating learning data that leads our machines to achieve particular functions. Those learning processes are usually categorized as supervised learning and unsupervised learning that are practical in classification, regression, data description, etc [34]. With the massive advent of software development environment/kits [35, 36, 37], to design a customized network is no longer a monopoly to data researchers and engineers. More educational resources are infused into this field, making ML a booming industry and the hottest topic today.

However, just like every story that we learned from our history, a rising technology is a double bladed sword. Except for the excellent performance in the technology revolution,

we can never ignore potential risks since ML is also a dangerous tool for attackers. As hardware security engineers, we know more about this. In this section, more details will be highlighted in how ML is applied in defense and attack mechanisms.

1.4.1 MACHINE LEARNING ATTACKS IN HARDWARE SYSTEM

Early investigations of malicious hardware-based attacks tend to hypothesize that attackers extract physical signals from defects of designs and perform mathematical analysis according to collected information. Recent studies [38, 39] demonstrate machine learning models exhibit extensive potentials in data mining and analysis, extensively reducing the time cost in extracting confidential information from physical signals. Owing to the cruel environment of competition, devices' resistance against machine learning attacks is proposed as a new design criterion to all new secure hardware design. At present, the application of machine learning attacks are usually categorized into two main aspects. One direction of attacks attempts to adopt machine learning as auxiliary tools of mathematical analysis in revealing secret information.

Fig. 1.2 shows an example of profiling attack using ML learning as its modeling tool. The side-channel signals are initially collected from targeted hardware devices. By injecting adequate number of input data and objective physical signals which contains potential correlation to confidential information, a training data set is thus established. In this case, the constructed neural network is regarded as a black box whose learning results may describe the relationship in given training samples. If the training dataset is large enough and the network model is well constructed, the trained model should be capable in imitating the inner logic of the cryptographic chip. By means of ML, attackers will no longer bother to

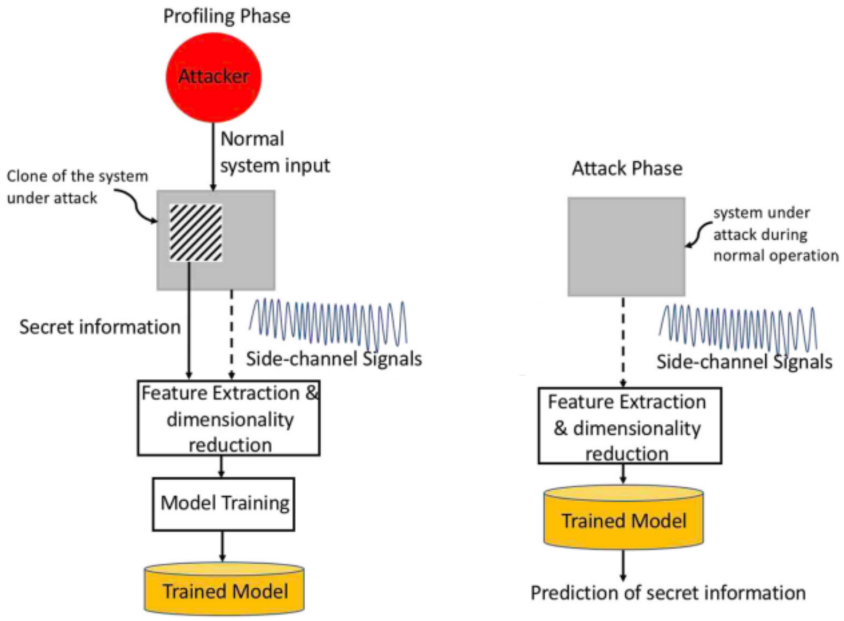


Fig. 1.2: Profiling-based side-channel analysis attack using machine learning as modeling tools: (a) profiling phase; (b) attack phase [2]

attempt different secret keys or build specific mathematical models. A well-trained model is sufficient for attackers to manipulate the hardware system as desired.

Another malicious ML attack model that we have to face today is IC overbuilding, a modeling attack that aims to crack the authentication system of PUF devices. The operation of data extraction and model training are similar to aforementioned side-channel attacks. In practice, SVM, logistic regression, and more advanced DNN architectures are used as cracking models to PUF primitives.

1.4.2 MACHINE LEARNING COUNTERMEASURES IN HARDWARE SECURITY

By contrast, ML possess more extraordinary capabilities in designs of hardware security.

One remarkable application is the detection of hardware Trojans. By implementing secret extracting hardware part, hardware Trojans forms malicious inclusions in hardware system that result in the degradation of performance or the failure of functions in hardware systems [2]. Due to the increasing globalization of industry have been increasing, more third-party manufacturers take part in the production of chips, raising risks that malicious inclusions to be embedded onto the primary design. Hence, the detection of malicious inclusions comes to be a pivotal mechanism in current hardware manufacture. On the basis of the dependence of the usage of golden-chips, fabricated chips that is validated as free of the infection of hardware Trojans, countermeasures to hardware Trojans can be classified as golden-chip methods and golden-chip free methods. Benefiting from machine learning, the workload of data analysis and feature extraction is greatly reduced. By using the divergences in power consumption, [40] exploits support vector machine as a golden-chip based method. An ASIC based implementation method of hardware Trojans is proposed in [41]. Simultaneously, a comprehensive detection using SVM, PCA, and side-channel extraction is introduced by the same team. In addition, more advanced researches of deep neural networks (DNN) participate in the wrestle of confidential information. [42] introduces a method that optimize non-linearity after an initial PCA analysis on hardware Trojans. A BP network is demonstrated to be efficient in accomplishing multiple Trojan detection works individually [43]. This also proves neural networks have satisfying performances in self-learning and data adaptions.

1.5 HARDWARE TROJANS

Hardware Trojans are malicious hardware inclusions that leak secret information, degrade the performance of the system, or cause denial-of-service [2]. With the promotion of globalization and industrial cooperation, the production of a single chip may always involve the engagement of third-party companies. Since the confidential information is always of great appeal to attackers, commercial spies and alike information thieves tend to exploit state-of-the-art technologies and seek unusual attack approaches and vulnerabilities to steal confidential information. Under this background, hardware Trojans and their corresponding detection theories are thus proposed.

Unlike aforementioned non-invasive attack models, like ML attacks and SCA attacks, hardware Trojans is always implemented during fabrication process. For any ML attack models or SCA attack models, approaches of data extraction, methods/tools of analysis, targets of vulnerable architectures are mostly transcendental to designers or security engineers. Based on investigated vulnerabilities or accidents that happens, there are always countermeasures that obstruct the attack procedure. The detection and removal of hardware Trojans, however, is extremely difficult and troublesome. First of all, those malicious components can be very small comparing to the whole layout. Liu et al. propose two malicious inclusion designs that only introduce several diodes to achieve effective data extraction [44]. The scale of Trojans means attackers do not greatly change the function or physical signals. Thus, it is always difficult to be aware of the existence of Trojans.

Current studies of hardware Trojans pay more attention to detection of particular Trojan types. And according to usage of golden chips, chips that are free from Trojan infection,

detection approaches can be categorized as golden chip-based methods and golden chip-free methods. As the name implies, golden chip-based methods require a bunch of chips that are validated free from infection. By using secure reference group, the differences between Trojan-infected chips and golden chips can be easily extracted when comparing physical divergence in between. [40] utilize the frequency domain components of power consumption traces as their reference variable and use support vector machine (SVM) to filter infected chips. A Bayesian method is used to analyze current divergences in two groups [45]. The execution path delay can also be used as examining variable. By using principle component analysis (PCA), a divergence in time delay can be detected to categorize Trojan-infected chips [46]. As for golden chip-free methods, the detection is mostly based on a known parameter dimension that physical parameters may vary between/among target groups. [44] utilize PCA as analyzing tool to map transmission power into three dimensional points and finally classify chip samples into three groups (one Trojan-free and two Trojan infected). Although a great number of papers are proposed on detection approaches of Trojan chips, some defects still exist in their detection procedures. First of all, most Trojan detection methods are proposed with a new Trojan design. Those methods are mostly valid only for the proposed Trojan chips. Moreover, some Trojan detection method only focus on one or two physical parameters. And that is under an assumption that attackers are using the particular Trojan types. As for golden chip-free methods, most proposed work only use PCA or SVM as feature extraction tools. Those methods do not specify the rationality that uses ML models. And in reality, if the Trojan type is unknown, there is no guarantee that the divergence signal can be truly extracted and used for further Trojan detection. Therefore, a

general golden chip-free detection approach is still the ultimate goal in the research domain of hardware Trojans.

1.6 OUR CONTRIBUTION

Consider the current development and bottlenecks of PUF devices illustrated in Section 1.1. The trends of future secure hardware devices should manage to elevate security performance against potential attacks by means of hardware and software simultaneously. In the meantime, the reduction of excess overhead is also an essential consideration. Therefore, we propose our improved designs of secure hardware devices. Our contribution in this work is summarized as follows:

- *Chapter 2* proposes an innovate design of physical unclonable function device against machine learning attacks and side channel attacks.
- *Chapter 3* investigates features of wave dynamic differential logic (WDDL) and propose our design of a new PUF primitive against machine learning attacks and power attacks.
- *Chapter 4* reviews drawbacks of two former designs and introduces a new floorplan for PUF primitives, aiming at giving an ultimate solution to non-invasive attacks.
- *Chapter 6* designs an innovate hardware Trojan for conventional PUF primitive. A statistical method is proposed as new golden chip-free detection approach.
- *Chapter 5* aims at vulnerabilities in profiling attacks and examine the effectiveness of a new attack model that jointly use ML models and SCA models.

- *Chapter 7* summarizes all previous chapters and systematically integrates concepts of designs, vulnerability detection, and performance evaluation into a whole.

- *Chapter 8* exhibits reviews of our current work and future research plans after graduation are mapped out.

CHAPTER 2

DESIGN AN INNOVATE PHYSICAL UNCLONABLE FUNCTION DEVICE WITH VOLTAGE REGULATOR

2.1 MOTIVATION

On-chip workload-aware multi-phase voltage regulators (WAMPVRs) is a countermeasure to power attacks[1]. Oriented from voltage regulation techniques, WAMPVRs aims at mitigating output power profiles while keeping small, fast efficient, high power density, and robust features of initial voltage regulators [48, 49]. By introducing random sequence control scheme, the leakage power can be mitigated to an acceptable level as well as keep a high power conversion efficiency. Converter-gating (CoGa) regulator [3] and converter-reshuffling (CoRe) regulators [50] are two typical regulating techniques. Fig. 2.1 exhibits the base architecture design of CoGa regulator and CoRe regulator. Both architectures adopt similar layout designs. In CoGa regulation scheme, the pseudo-random number generator (PRNG) will act according to the change of output power. When the power demand does not change dramatically, the control signal from the PRNG tends to remain unchanged to maintain a stable output power profile. In CoRe regulators, however, the PRNG will take former control signal as references. By keeping changing the input control sequence, the entropy value of the input sequence is elevated and more randomness can be introduced to protect the output power to be extracted and analyzed.

[1]The content of this Chapter partially has been published in [47].

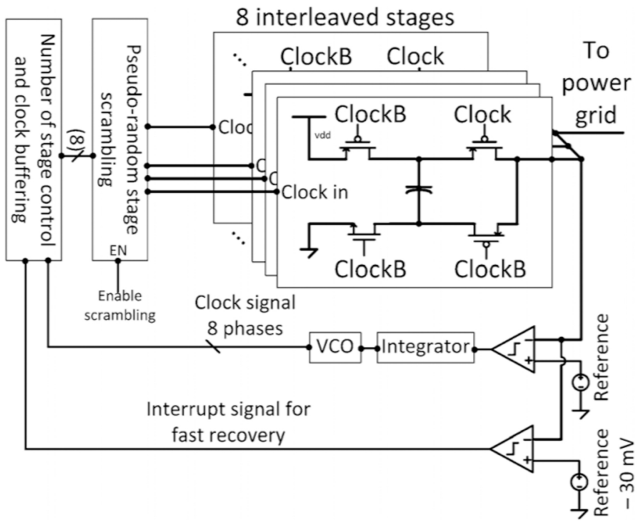


Fig. 2.1: CoGa regulator in [3] (8-phase) introduces PRNG to realize the mitigation of output power profiles.

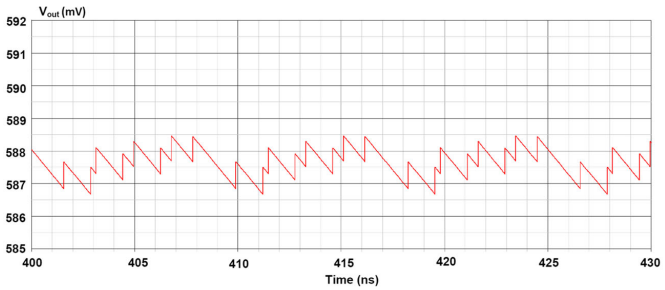
In the aforementioned architecture of CoRe voltage regulator, the input sequence is under controlled by a PRNG, which can be served as a binary sequence input. Besides, all flying capacitors are interleaved with shifted clocks. If we take flying capacitors in each stage as random outcomes of manufacturing fabrication and the capacitance discrimination can be detected and compared, the output power should produce a sensible difference even two identical input sequences are applied on two CoRe voltage regulators. Consequently, CoRe voltage regulator can be easily redesigned as a new PUF. Since the new PUF is designed on a functional device, the overhead of the PUF is greatly reduced compared to traditional SRAM PUF, arbiter PUF, or ring-oscillator (RO) PUF. Those traditional PUF primitives only provide security features which introduce losses in power and area. In this research, we would like to provide an innovate thinking of PUF design, by which we can consider whether we can design and implement PUF primitives on an existing device in our current computer architecture.

2.2 BACKGROUND

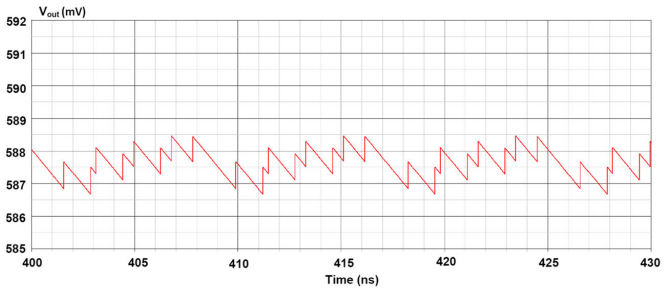
The workload-aware multi-phase voltage regulators (WAMPVRs) like converter-gating (CoGa) voltage regulator [3] and converter-reshuffling (CoRe) voltage regulator [50] are designed based on multi-phase switched capacitor (SC) voltage converters. Integrating WAMPVRs fully on-chip is an efficient solution for reducing the power conversion loss and strengthening the robustness of modern ICs against power attacks [3, 50]. As demonstrated in [3, 50], increasing the total number of phases for the WAMPVRs can result in significant improvements of the power conversion efficiency and the security against power attacks. Accordingly, the designs of on-chip voltage regulators with more than 120 phases have been frequently reported in the recent literatures [51, 52].

In the design of a 2:1 (Input voltage/output voltage = 2:1) 32-phase on-chip SC voltage converter, the simulated output voltage ripples are shown in Fig. 2.2. Case 1 (as shown in Fig. 2.2(a)) and Case 2 (as shown in Fig. 2.2(b)) indicate that different number of activated phases can generate different output voltage ripple signatures. Furthermore, when we compare Case 2 with Case 3 in Fig. 2.2(b) and 2.2(c), under the same number of active phases, the output voltage ripples are also different if the sequences of activation pattern are different.

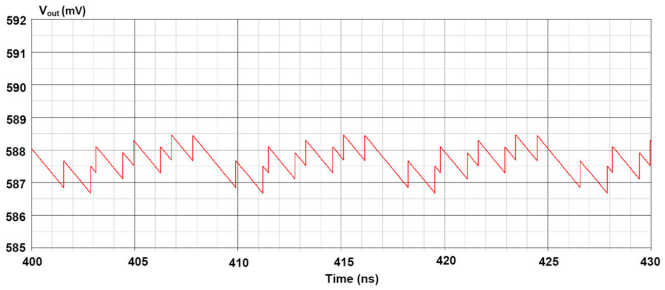
In a multi-phase SC voltage converter, the output voltage ripple is extremely sensitive to the flying capacitance in each sub-phase [3]. Since the flying capacitor in each sub-phase is identically designed, the physical randomness of the flying capacitor induced by the fabrication process enables the multi-phase SC converter to be eligible for building PUF architectures.



(a)



(b)



(c)

Fig. 2.2: Output voltage ripples of a 2:1 32-phase SC converter. a Case 1: Sequence of activation pattern (8 active phases): (7, 12, 13, 18, 20, 25, 27, 31). b Case 2: Sequence of activation pattern (16 active phases): (1, 2, 3, 4, 6, 8, 9, 14, 15, 16, 22, 23, 26, 28, 29, 30). **c** Case 3: Sequence of activation pattern (16 active phases): (2, 5, 6, 9, 10, 11, 14, 16, 19, 23, 24, 26, 28, 29, 30, 32)

2.3 ARCHITECTURE DESIGN

The architecture of a workload-aware multi-phase voltage regulator (WAMPVR)-based strong PUF primitive is devised in Fig. 2.2. Two identically designed 32-phase switched-capacitor (SC) voltage converters: $Block_1$ and $Block_2$ are utilized to build the strong PUF

architecture. The output port of the j^{th} ($j = 1, 2, \cdots, 32$) phase of the SC converter in $Block_i$ ($i = 1, 2$) connects with the switch $W_{i,j}$. Moreover, a 32-bit phase number generator (PNG) is utilized to control the activation behaviors of the switches $W_{i,1}, W_{i,2}, \cdots, W_{i,32}$ to determine the sequences of active phases that are used for building the strong PUF. For example, if only the switches $W_{i,2}, W_{i,5}, W_{i,12}$, and $W_{i,18}$ are turned on by the PNG, the output voltage ripples of *phase* #2, *phase* #5, *phase* #12, and *phase* #18 of the SC converters are selected for generating the PUF response. Since a 32-bit PNG can generate $\binom{32}{0} + \binom{32}{1} + \binom{32}{2} + \cdots + \binom{32}{32} = 2^{32}$ different activation patterns, therefore, the total number of raw challenge-to-response pairs (CRPs) of the WAMPVR-based strong PUF primitive are 2^{32}.

As shown in Fig. 2.3, the mismatches of voltage ripple between $V_{out,1}$ and $V_{out,2}$ are magnified through employing an operational amplifier. Four pipelined SC circuits (*SC circuit* #1, $\cdots$, and *SC circuit* #4, as shown in Fig. 2.3) are utilized to convert the high-frequency voltage ripple mismatch V_a into the critical voltage $V_{a,1}$ for generating the secret authentication data B. Furthermore, each SC circuit has four independent working phases: *charging* phase, *charge-sharing* phase, *output* phase, and *discharging* phase. For example, as shown in Fig. 2.3, if *SC circuit* #1 is in *charging* phase, the switch $S_{1,1}$ will be turned on. Then the positive component of V_a will charge the capacitor $C_{1,2}$ while the negative component of V_a will charge the capacitor $C_{1,1}$. Once the *charging* phase ends, the switch $S_{1,1}$ will be turned off while the switch $S_{1,2}$ will be activated to balance the charge of the capacitors $C_{1,1}$ and $C_{1,2}$. After the *charge-sharing* phase, the *SC circuit* #1 will output the sampled critical voltage $V_{a,1}$ to generate the binary authentication data B by activating the switch

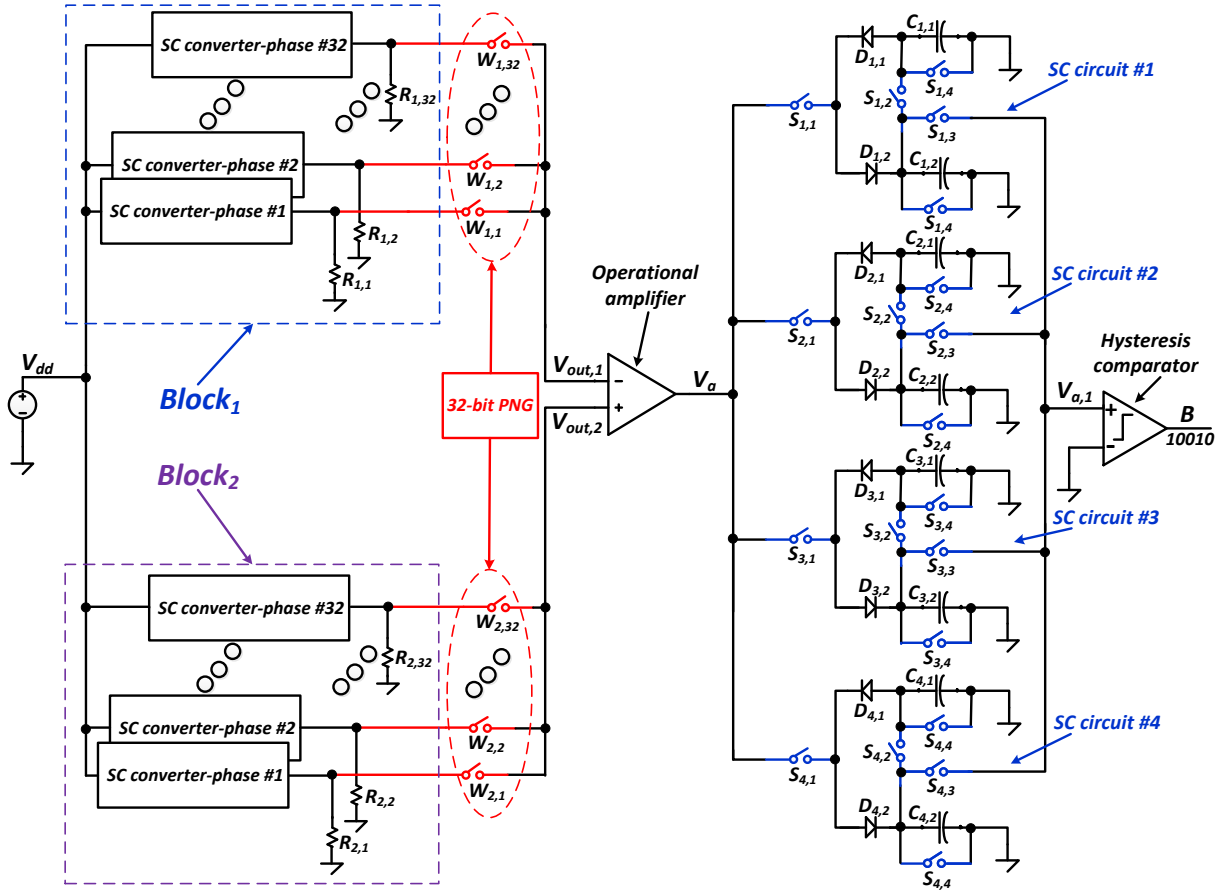


Fig. 2.3: Architecture of the WAMPVR-based strong PUF primitive (the total number X of phases in the original WAMPVR is 64, the resistors $R_{1,1}$, $R_{2,1}$, $R_{1,2}$, $\cdots$, and $R_{2,32}$ are designed with the same resistance R, and the capacitors $C_{1,1}$, $C_{1,2}$, $C_{2,1}$, $\cdots$, and $C_{4,2}$ are also designed with the same capacitance C).

$S_{1,3}$. If the voltage $V_{a,1} \geq 0$ V, the authentication data B outputs logic value "1". Otherwise, $B = 0$. In the final stage (*discharging* phase), the residual charge in the capacitors $C_{1,1}$ and $C_{1,2}$ will be cleared to initialize the next data sampling through turning on the $S_{1,4}$ switches. The main intention of employing *discharging* phase is to break the correlation between the current data and the history data against machine-learning attacks. Please note that in each SC circuit, the behaviors of the switches $S_{h,1}$, $(h = 1, 2, 3, 4)$, $S_{h,2}$, $S_{h,3}$, and $S_{h,4}$ are mutually exclusive. Additionally, as shown in Fig. 2.3, there are two different

kinds of diodes: back-biased diode $D_{h,1}$ and forward-biased diode $D_{h,2}$ exist in each SC circuit. The main role of these diodes is working as a non-linear transformation block to generate the non-linear output response B against machine-learning attacks, which will be fully discussed in Section 2.4.

2.4 EVALUATION

2.4.1 PERFORMANCE EVALUATION

Two most significant metrics that are selected to evaluate the PUF characterization are the inter-Hamming distance (HD) and the intra-HD (reliability) [53, 51, 54]. Inter-HD measures the distinctness between two different PUF devices while intra-HD (reliability) represents the stability of a single PUF device under different temperatures and supply voltages.

In Fig. 2.3, assume the resistors $R_{1,1}$, $R_{2,1}$, $R_{1,2}$, $\cdots$, and $R_{2,32}$ are designed with a high resistance R to reduce the overall power consumption of the WAMPVR-based strong PUF primitive. As a result, under the same process variation, the mismatch rate of these resistors $R_{1,1}$, $\cdots$, and $R_{2,32}$ will be negligible as compared to the mismatch rate of the flying capacitors in the SC converters. Hence, in Fig. 2.3, the output voltage $V_{out,i,j}$ of the j^{th} $(j = 1, 2, \cdots, 32)$ phase of the SC converter in $Block_i$, $(i = 1, 2)$ can be denoted by a function F, as shown below

$$V_{out,i,j} = F\left(C^s_{i,j}, V_{dd}, T_C, t + (j-1)\frac{T_S}{32}\right) \tag{2.1}$$

where C_s is the flying capacitance of the j^{th} phase of the SC converter in $Block_i$. V_{dd}, T_C, T_S,

and t, respectively, are the supply voltage, the environmental temperature, the switching period of the SC converters, and the timing of the 1^{st} phase of the SC converters. Let us assume the supply voltage V_{dd} and the environmental temperature T_C are time invariant. As a result, the critical parameters: C_S, V_{dd}, T_C, and t are mutually independent. Then the output voltage $V_{out,i,j}$ can be approximated as[2]

$$\begin{aligned} V_{out,i,j} &= F\left(C^s_{i,j}, V_{dd}, T_c, t+(j-1)\frac{T_s}{32}\right) \\ &= F_1\left(C^s_{i,j}\right) \times F_2\left(V_{dd}\right) \times F_3\left(T_c\right) \times F_4\left(t+(j-1)\frac{T_s}{32}\right) \\ &\approx \left(\sum_{i_1=0}^{m_1} a_{i_1}(C_{i,j})^{i_1}\right) \times \left(\sum_{i_2=0}^{m_2} b_{i_2}(V_{dd})^{i_2}\right) \times \left(\sum_{i_3=0}^{m_3} c_{i_3}(T_C)^{i_3}\right) \\ &\quad \times \left(\frac{d_0}{2} + \sum_{i_4=1}^{m_4} d_{i_4} cos\left(\frac{2\pi i_4}{T_S}\left(t+(j-1)\frac{T_S}{32}\right)\right)\right. \\ &\quad \left. + \sum_{i_4=1}^{m_4} e_{i_4} sin\left(\frac{2\pi i_4}{T_S}\left(t+(j-1)\frac{T_s}{32}\right)\right)\right) \end{aligned} \tag{2.2}$$

where $F_1(C^S_{i,j})$, $F_2(V_{dd})$, $F_3(T_C)$, and $F_4\left(t+(j-1)\frac{T_s}{32}\right)$, respectively, are the voltage components of $V_{out,i,j}$ that are determined by $C^S_{i,j}$, V_{dd}, T_C, and $\left(t+(j-1)\frac{T_s}{32}\right)$. $\sum_{i_1=0}^{m_1} a_{i_1}(C_{i,j})^{i_1}$, $\sum_{i_2=0}^{m_2} b_{i_2}(V_{dd})^{i_2}$, and $\sum_{i_3=0}^{m_3} c_{i_3}(T_C)^{i_3}$ are the approximated polynomial expansions of $F_1\left(C^S_{i,j}\right)$, $F_2\left(V_{dd}\right)$, and $F_3\left(T_C\right)$, respectively. a_{i_1} $(i_1 = 0, 1, \cdots, m_1)$, b_{i_2} $(i_2 = 0, 1, \cdots, m_2)$, and c_{i_3} $(i_3 = 0, 1, \cdots, m_3)$, respectively, are the coefficients of $(C_{i,j})^{i_1}$, $(V_{dd})^{i_2}$, and $(T_C)^{i_3}$. m_1, m_2, and m_3 are the degrees of the approximated polynomials of $F_1(C^S_{i,j})$, $F_2(V_{dd})$, and $F_3(T_C)$ respectively. $d_0, d_1, \cdots, m_4, e_1, e_2, \cdots, e_{m_4}$ (m_4) are the coefficients (degree) of the

[2]As demonstrated in Fig. 2.2, the output voltage of an SC converter is a periodical signal. Therefore, the voltage component of $V_{out,i,j}$ related with the timing t can be unfolded with Fourier series.

approximated Fourier series of $F_4\left(t+(j-1)\frac{T_s}{32}\right)$. If the supply voltage V_{dd}, the environmental temperature T_C, and hte timing t are fixed, through matching the relationship curve between the capacitance $C^s_{i,j}$ and the output voltage $V_{out,i,j}$, the coefficients a_0, a_1, $\cdots$, and the degree m_1 for $F_1(C^S_{i,j})$ can be unriddled. The coefficients and the degrees of $F_2(V_{dd})$, $F_3(T_C)$, and $F_4\left(t+(j-1)\frac{T_s}{32}\right)$ can also be estimated in a similar way.

Once the complete expression of the output voltage $V_{out,i,j}$ is obtained, the following step is to model the mismatches of output voltage ripple between $Block_1$ and $Block_2$ in Fig. 2.3. Assume the 32-bit PNG in Fig. 2.3 generates the 32-bit binary data $W=(w_1,w_2,\cdots,w_{32})_2$ to select a certain number of active phases of the SC converters for building a strong PUF for authentication by controlling the activation patterns of the corresponding switches.[3] As a result, by using the Kirchhoff's law, the voltages $V_{out,1}$ and $V_{out,2}$ in Fig. 2.3 can, respectively, be derived as

$$\begin{aligned} V_{out,1} &= \frac{\sum_{j=1}^{32} w_j V_{out,i,j}}{R} \times \frac{R}{\sum_{j=1}^{32} w_j} \\ &= \frac{\sum_{j=1}^{32} w_j V_{out,i,j}}{\sum_{j=1}^{32} w_j} \end{aligned} \tag{2.3}$$

$$\begin{aligned} V_{out,1} &= \frac{\sum_{j=1}^{32} w_j V_{out,2,j}}{R} \times \frac{R}{\sum_{j=1}^{32} w_j} \\ &= \frac{\sum_{j=1}^{32} w_j V_{out,2,j}}{\sum_{j=1}^{32} w_j} \end{aligned} \tag{2.4}$$

[3] $w_1, w_2, \cdots, w_{32}$ control the activation behaviors of the switches $W_1, W_2, \cdots, W_{32}$ respectively. If $w_j = 1$, the switches $W_{1,j}$ and $W_{2,j}$ are turned on, and vice versa.

Then the voltage ripple mismatch V_a in Fig. 2.3 is

$$V_a = A_v \left(V_{out,2} - V_{out,1}\right) = A_V \frac{\sum_{j=1}^{32} w_j \left(V_{out,2,j} - V_{out,1,j}\right)}{\sum_{j=1}^{32} w_j} \tag{2.5}$$

For the WAMPVR-based strong PUF primitive in Fig. 2.3, assume the switching period of the SC circuits is designed equal to four times of the switching period of the SC converters and the pulse width of all the switches $S_{1,1}, S_{1,2}, \cdots, S_{4,4}$ in SC circuits is 25%. If $SCcircuit1$ is in *charging* phase, the switch $S_{1,1}$ is in on-state. The voltages V_a^* and V_a^{**} of the capacitors $C_{1,1}$ and $C_{1,2}$ in Fig. 2.3, respectively, are

$$V_a^* = \begin{cases} V_a - V_b & , V_a \geq V_b \\ 0 & , V_a < V_b, \end{cases} \tag{2.6}$$

$$V_a^* = \begin{cases} V_a + V_b & , V_a \leq V_b \\ 0 & , V_a > V_b, \end{cases} \tag{2.7}$$

where V_b is the forward-biased threshold voltage of the diodes $D_{1,1}$ and $D_{1,2}$ in Fig. 2.3. When $SCcircuit$ 1 enters into *output* phase, since the capacitors $C_{1,1}$ and $C_{1,2}$ are designed with the same capacitance C, the critical voltage $V_{c,1}$ in Fig. 2.3 can be denoted as

$$\begin{aligned} V_{a,1} &= \frac{\int_t^{t+T_s} C_{1,1} \frac{dV_a^*}{dt} dt + \int_t^{t+T_s} C_{1,2} \frac{dV_a^{**}}{dt} dt}{C_{1,1} + C_{1,2}} \\ &= \frac{\int_t^{t+T_s} \left(\frac{dV_a^*}{dt} + \frac{dV_a^{**}}{dt}\right) dt}{2} \end{aligned} \tag{2.8}$$

Therefore, if the critical voltage $V_{a,1} \geq 0$ V, the output binary data $B = 1$. Otherwise,

$B = 0$.

Assume N number of WAMPVRs are utilized for building a strong PUF primitive to generate the N-bit binary authentication data $\bar{B}$. As a result, if K strong PUF primitives are selected for evaluating the uniqueness, the inter-HD E is written as [55]

$$E = \frac{2}{K\,(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \frac{\bar{B}_{k_1} \oplus \bar{B}_{k_2}}{N} \times 100\% \tag{2.9}$$

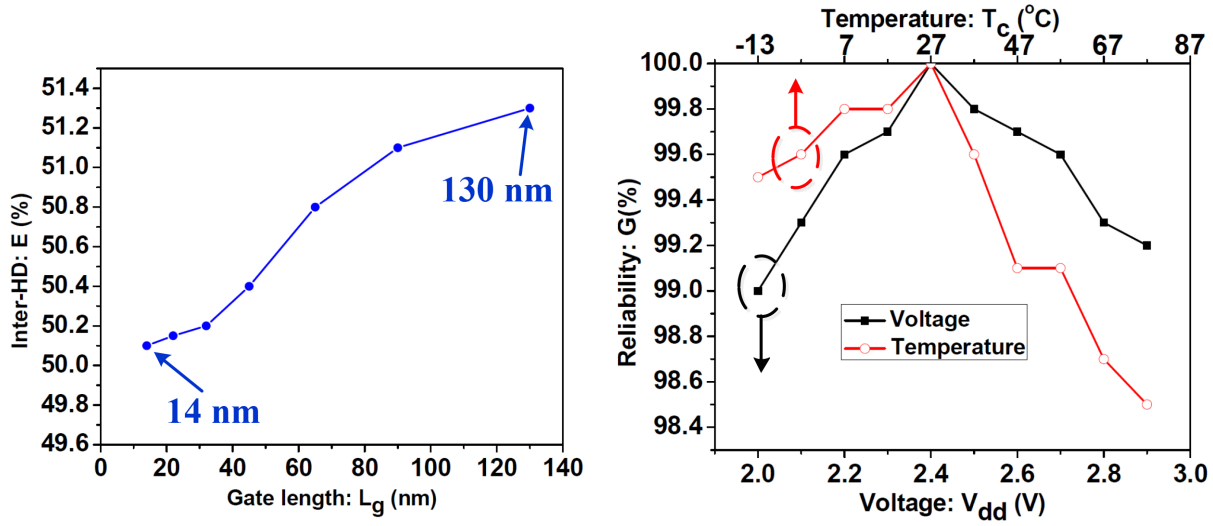
where $\bar{B}_{k_1}$ $(k_1 = 1, 2, \cdots, K-1)$ and $\bar{B}_{k_2}$ $(k_2 = k_1 + 1, k_1 + 2, \cdots, K)$, respectively, are the N-bit binary authentication data generated by the k_1^{th} and k_2^{th} strong PUF primitives.

Similarly, for a single PUF primitive, if M number of different environmental settings are considered, the reliability of the strong PUF primitive G can be expressed as [55]

$$G = \left(1 - \frac{1}{M} \sum_{l=1}^{M} \frac{\bar{B}_0^* \oplus \bar{B}_l^*}{N} \right) \tag{2.10}$$

where $\bar{B}_0^*$ and $\bar{B}_l^*$ are the N-bit binary authentication data generated by the single PUF primitive under the ideal and l^{th} $(l = 1, 2, \cdots, M)$ environmental setting, respectively.

All of the aforementioned parameters in the mathematical model of the designed WAMPVR-based strong PUF primitive are extracted from the 130 nm CMOS technology kits in Cadence. As shown in Fig. 2.4, by applying the Monte Carlo simulation into the aforementioned mathematical model, the inter-HD E of the WAMPVR-based strong PUF primitive is about 51.3% ($L_g = 130$ nm). Furthermore, if the scaling of CMOS technology is considered, through utilizing the mismatch rates of capacitors under different CMOS technologies from [17], the inter-HD E of the technology-scaled WAMPVR-based strong PUF primitive



(a) (b)

Fig. 2.4: Performance evaluation for the designed strong PUF primitive. (a) Inter-HD E versus gate length L_g ($K = 100$ and $N = 32$). (b) Reliability G versus supply voltage V_{dd} and environmental temperature T_c ($M = 50$ and $N = 32$).

can also be predicted. Related experiment results are as shown in Fig. 2.4(a). When the CMOS technology is scaled from 130 nm to 14 nm, the inter-HD, E is improved from 51.3% to 50.1%. That indicates a larger capacitance mismatch rate induced by a shorter gate length enables the WAMPVR-based strong PUF primitive to achieve a better uniqueness. Additionally, the reliability G of the designed WAMPVR-based strong PUF primitive is assessed in Fig. 2.4(b). Concluded from results above, the ideal environmental setting for the strong PUF primitive is: the ambient temperature $T_c = 27$. As shown in Fig. 2.4(b), the worst reliability of the designed WAMPVR-based strong PUF primitive is 98.5% when $V_{dd} = 2.9$ V.

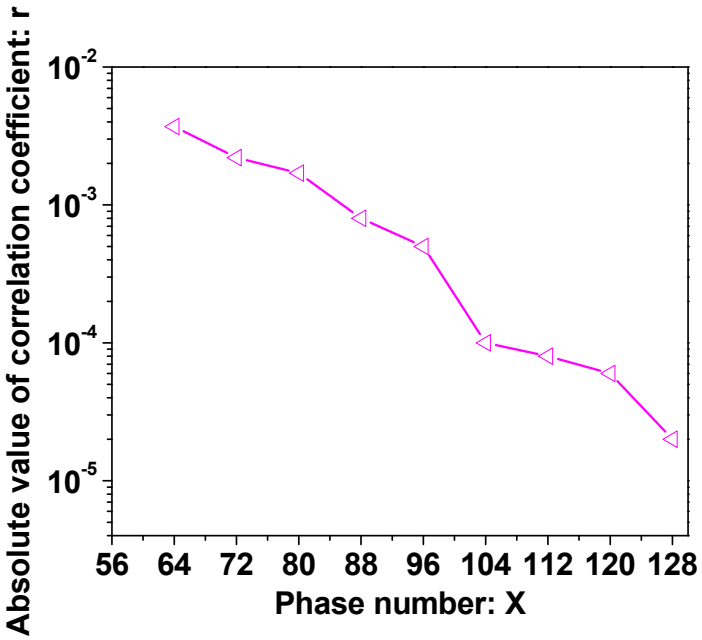


Fig. 2.5: Absolute value r of correlation coefficient between P_{in} and ΔC versus phase number X against side-channel attacks.

2.4.2 SECURITY AGAINST SIDE-CHANNEL ATTACKS

Side-Channel Leakage Analysis

If an X-phase (assume X is even) WAMPVR is utilized for devising a strong PUF architecture, the number of phases in $Block_1$ and $Block_2$ in Fig. 2.3 is $X/2$. Since all the phases in $Block_1$ and $Block_2$ are active all the time, the input power of the WAMPVR-based strong PUF primitive is a constant within a switching period T_s regardless the variations of process, voltage, and temperature (PVT). However, if the mismatches of the flying capacitors in the SC converters induced by the random fabrication process are considered, the total input power P_{in} of the WAMPVR-based strong PUF primitive within a switching period T_s

can be expressed as

$$P_{in} = \sum_{i=1}^{2} \sum_{j=1}^{X/2} C_{i,j}^{S} f_s V_{dd}^2 \tag{2.11}$$

where f_s is the switching frequency of the SC converters.

Since the attacker may leak the mismatches of the flying capacitors in the SC converters through analyzing the input power P_{in}, the absolute value r of correlation coefficient between the input power P_{in} and the capacitance mismatch $\Delta C = C_{2,j}^{S} - C_{1,j}^{S}$ is studied against side-channel attacks. As shown in Fig. 2.5, the correlation coefficient between P_{in} and ΔC is about 0.0037 when the phase number $X = 64$, which indicates a good robustness against side-channel attacks. Moreover, if the phase number X increases, the correlation coefficient between P_{in} and ΔC will be further reduced against side-channel attacks.

Implementation of Side-Channel Attacks

The main intention of implementation of implementing side-channel attacks on the WAMPVR-based strong PUF primitive is unriddling the output response B by analyzing the critical side-channel leakage. If the input power P_{in} of the proposed strong PUF device is tailored as the critical side-channel leakage, the relationship between the input power P_{in} and the output response B needs to be studied when side-channel attacks are executed. Since the random fabrication process and circuit noise conform to normal distributions [55, 56], if the variations of PVT are considered, the input power P_{in} can be further

derived as

$$\begin{aligned} P_{in} &= \sum_{i=1}^{2}\sum_{j=1}^{X/2} C_{i,j}^{S} f_s V_{dd}^2 \\ &= \frac{1}{\sqrt{2\pi X}\sigma_c} \exp\left(-\frac{\left(\sum_{i=1}^{2}\sum_{j=1}^{X/2} C_{i,j}^{S} - X\mu_C\right)^2}{2X\sigma_c^2}\right) \\ &\quad \times \frac{1}{\sqrt{2\pi}\sigma_f} \exp\left(-\frac{(f_s-\mu_f)^2}{2X\sigma_c^2}\right) \\ &\quad \times \left(\frac{1}{\sqrt{2\pi}]\sigma_c} \exp\left(-\frac{(V_{dd}-\mu_v)^2}{2\sigma_v^2}\right)\right)^2 \end{aligned} \tag{2.12}$$

where $\mu_c(\sigma_c)$, μ_f (σ_f), and μ_v (σ_v) are the means (standard deviations) of the flying capacitance, switching frequency, and supply voltage of the proposed strong PUF device, respectively.

So as to model the relationship between the input power P_{in} and the output response B, let us define a function $F^*(P_{in})$ and approximate the function $F^*(P_{in})$ with a polynomial expansion $F^{**}(P_{in})$ as shown below

$$F^*(P_{in}) \approx \sum_{k=0}^{K^*} f_k^* \times (P_{in})^k = F^{**}(P_{in}) \tag{2.13}$$

where K^* is the degree of the approximated polynomial and f_k^* is the coefficient of $(P_{in})^k$. Assume that Z is the number of input power and output response pairs: $(P_{in,1}, B_1)$, $(P_{in,2}, B_2)$, $\cdots$, and $(P_{in,Z}, B_Z)$ of the proposed strong PUF primitive are selected for analysis, then the matching error ΔL between the input power P_{in} and the output response B with the polynomial expansion $F^{**}(P_{in})$ can be expressed as

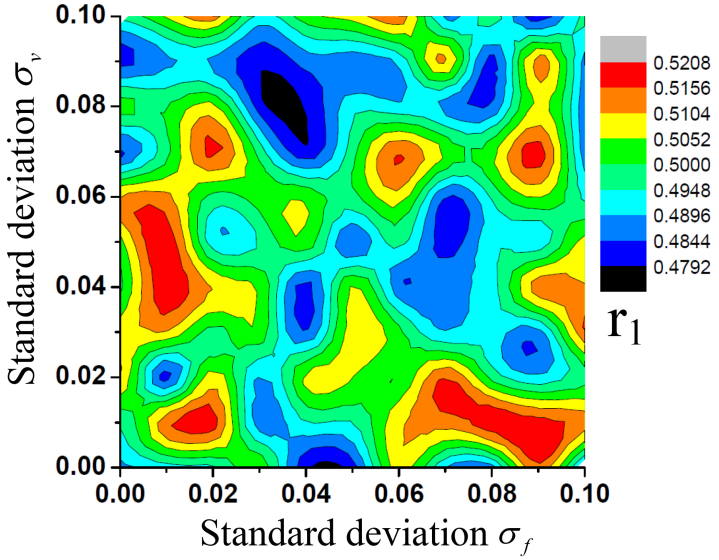


Fig. 2.6: Prediction accuracy r_1 of power attacks versus standard deviations δ_f and δ_v after analyzing 1 million input power and output response pairs (The colors and contours represent the variation values of the prediction accuracy r_1. Since the variation values of the prediction accuracy r_1 are around 0.5 and random, that reflects power attacks are unable to leak critical information on the proposed PUF).

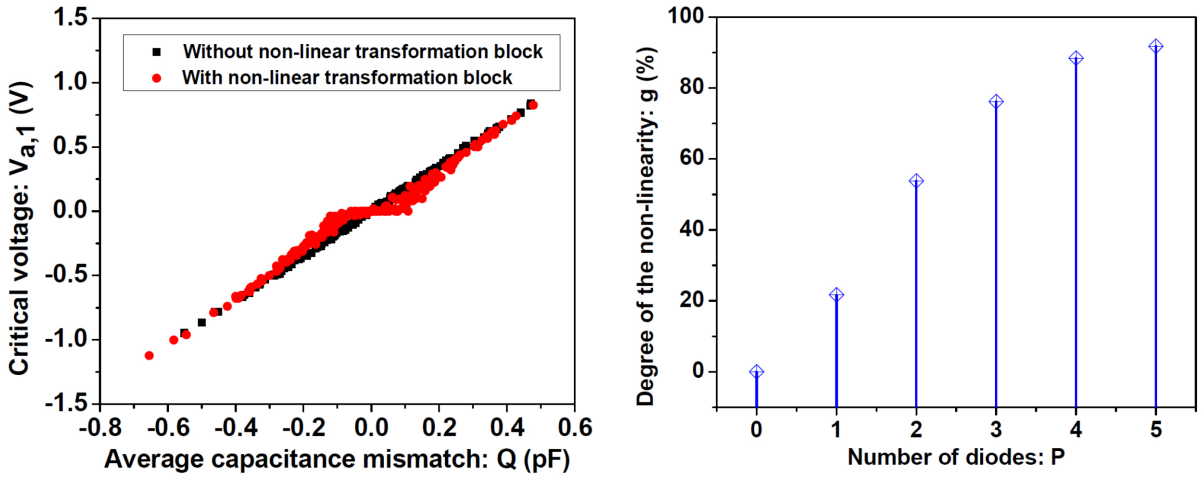
$$\Delta L = \sum_{z=1}^{Z} \left(\sum_{k=0}^{K^*} f_k^* \times (P_{in,z})^k - B_z \right)^2 \tag{2.14}$$

By minimizing the matching error ΔL with

$$\frac{\partial \Delta L}{\partial f_k^*} = \left(2 \sum \left(\sum_{k=0}^{k^*} f_k^* \times (P_{in,z})^k - B_z \right) \right) \times \sum_{k=0}^{K^*} (P_{in,z})^k = 0, \tag{2.15}$$

the optimal K^*, f_0^*, f_1^*, $\cdots$, $f_{K^*}^*$ can be determined.

The Z number of input power and output response pairs: $(P_{in,1}, B_1)$, $(P_{in,2}, B_2)$, $\cdots$, and $(P_{in,z}, B_z)$ of the WAMPVR-based strong PUF primitive with the 130 nm CMOS technology under the standard deviations σ_f and σ_v is simulated in Cadence. As shown in Fig. 2.6, if power attacks are implemented on the WAMPVR-based strong PUF primitive by exploring



(a) (b)

Fig. 2.7: (a) Critical voltage $V_{a,1}$ versus average capacitance mismatch Q against ML attacks. (b) Number of diodes P between the switch $S_{h,1}$ and the capacitor $C_{h,x}$ in Fig. 2.3 versus degree g of the non-linearity of the WAMPVR-based strong PUF primitive.

the input power P_{in} as the critical side-channel leakage, the maximum prediction accuracy of the power attacks is about 0.52 even if 1 million input power and output response pairs are analyzed. That indicates the proposed strong PUF primitive is adequately secure against the advanced power attacks.

2.4.3 SECURITY AGAINST MACHINE-LEARNING (ML) ATTACKS

Non-Linearity Analysis

The degree of the non-linearity between the input challenges and the output response is a critical parameter that affects the robustness of a strong PUF against machine-learning (ML) attacks [57]. For the WAMPVR-based strong PUF primitive in Fig. 2.3, the relationship between the average capacitance mismatch Q and the critical voltage $V_{a,1}$ is studied.

The definition of the average capacitance mismatch Q in Fig. 2.3 is

$$Q = \sum_{j=1}^{32} w_j \left(C_{2,j}^S - C_{1,j}^S \right) \tag{2.16}$$

The non-linear relationship between Q and $V_{a,1}$ can be observed in Fig. 2.7(a) when the non-linear transformation block that is consist of diodes $D_{1,1}, D_{1,2}, D_{2,1}, \cdots, D_{4,2}$ (as shown in Fig. 2.3) is enabled. By contrast, a strong linear relationship exists between Q and $V_{a,1}$ if the non-linear transformation block is removed.

If Y number of different Q values: $Q_1, Q_2, \cdots, Q_Y$ are studied, assume the corresponding value of the critical voltage $V_{a,1}$ is: $V_{a,1,1}, V_{a,2,1}, \cdots, V_{a,2,Y}$ $\left(V'_{a,1,1}, V'_{a,2,1}, \cdots, V'_{a,2,Y} \right)$ for the strong PUF with (without) the non-linear transformation block. As a result, the degree g of the non-linearity of the designed WAMPVR-based strong PUF primitive can be estimated as [58]

$$g = \frac{\frac{1}{2Y} \sum_{j_1=1}^{Y} \left(V_{a,1,j_1} - V'_{a,1,j_1} \right)^2}{\left(\frac{\sum_{j_1=1}^{Y} V'_{a,1,j_1}}{Y} \right)^2} \times 100\% \tag{2.17}$$

To enhance the degree of the non-linearity of the proposed strong PUF device, we can increase the number of diodes in the non-linear transformation block. For instance, in Fig. 2.3, only one diode $D_{h,x}$ ($h = 1, 2, 3, 4$ and $x = 1, 2$) exists between the switch $S_{h,1}$ and the capacitor $C_{h,x}$. If larger number of diodes can be inserted, the degree g of the non-linearity of the WAMPVR-based strong PUF primitive will be improved ($g = 91.79\%$ when $p = 5$), as shown in Fig. 2.7(b).

Linear Regression (LR) Attacks

Linear regression (LR) algorithm [58, 59] is a kind of popular machine-learning (ML) algorithms that can be explored to uncover the confidential information of a strong PUF device. For the WAMPVR-based strong PUF primitive as shown in Fig.2.3, there is a 32-bit phase number generator (PNG) $W = (w_1, w_2, \cdots, w_{32})_2$ that is working as the input challenge. Accordingly, the main intention of performing ML attacks on the proposed strong PUF primitive is estimating the relationship between the input challenge W and the output response B. When the LR algorithm is considered for training the challenge-to-response pairs (CRPs), the predicted output response $B*$ of the proposed strong PUF device under the input challenge W can be written as

$$B^* = \sum_{j=1}^{32} W_j \theta_j + \theta_0 \tag{2.18}$$

where $\theta_0, \theta_1, \cdots, \theta_{32}$ are linear coefficients of the LR algorithm.

If n number of CRPs: $(W_1, B_1), (W_2, B_2), \cdots$, and (W_n, B_n) are selected as the training data sets, by considering the lease squares fit rule, the cost function $S(\theta)$ of the LR algorithm can be obtained as

$$S(\theta) = \frac{1}{2n} \sum_{j_1=1}^{n} \left(\sum_{j=1}^{32} w_{j,j_1} \theta_j + \theta_0 - B_{j_1} \right)^2 \tag{2.19}$$

where w_{j,j_1} is the j^{th} bit of the j_1^{th} input challenge W_{j_1}. After repeating the gradient descent

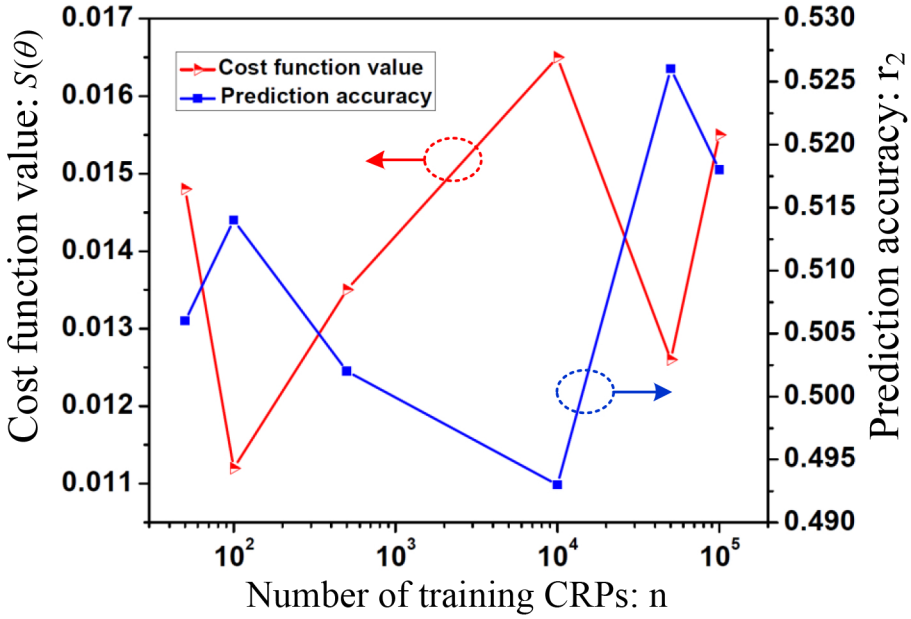


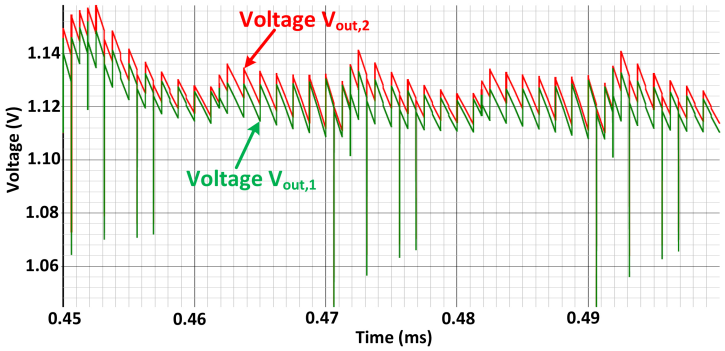
Fig. 2.8: Cost function value $S(\theta)$ and prediction accuracy r_2 versus number of training CRPs n for the WAMPVR-based strong PUF primitive under LR attacks (number of diodes $P = 3$).

algorithm as shown below

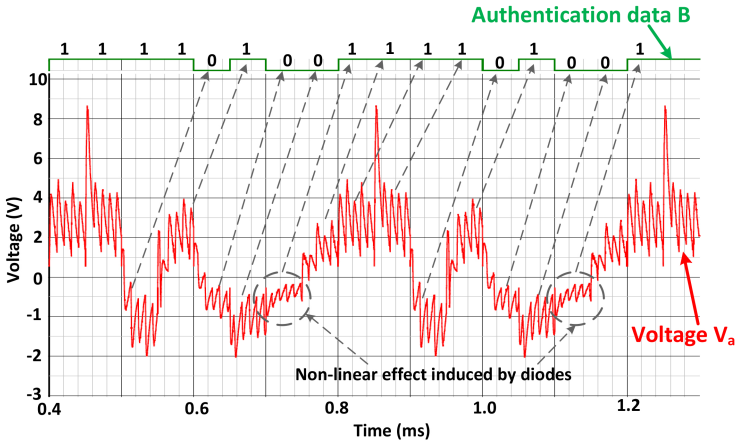
$$\begin{aligned}\theta_j &:= \theta_j - \beta\frac{\partial S(\theta)}{\partial \theta_j} \\ &= \theta_j - \beta\frac{1}{n}\left(\sum_{j_1=1}^{n}\left(\sum_{j=1}^{32} w_{j,j_1}\theta_j + \theta_0 - B_{j_1}\right)^2\right)\sum_{j=1}^{32} w_{j,j_1}\end{aligned} \tag{2.20}$$

where β is the learning coefficient of the LR algorithm, the critical parameters: $\theta_0, \theta_1, \cdots, \theta_{32}$ can be estimated.

Fig. 2.8 shows the variations of the cost function value $S(\theta)$ and prediction accuracy r_2 of the LR algorithm is below 0.53 after enabling $100,000$ number of CRPs, as shown in Fig. 2.8. Consequently, the proposed strong PUF primitive is sufficiently robust against ML



(a)



(b)

Fig. 2.9: Simulated waveforms of the WAMPVR-based strong PUF primitive ($X = 32$). (a) Voltages $V_{out,1}$ and $V_{out,2}$ versus time. (b) Voltage $V_{out,1}$ and binary authentication data B versus time.

attacks.

2.5 CIRCUIT LEVEL SIMULATION

A WAMPVR-based strong PUF architecture is designed and simulated. The waveforms of the voltages $V_{out,1}$ and $V_{out,2}$ in Fig. 2.3 that contain the voltage ripple information are shown in Fig. 2.9(a). By using Monte Carlo simulation, the mismatches of voltage ripple of $Block_1$ and $Block_2$ in Fig. 2.3 induced by the random mismatches of the flying capacitors

in the SC converters can be observed in Fig. 2.9(a) obviously. Additionally, as shown in Fig. 2.9(b), if the voltage V_a (as shown in Fig. 2.3) outputs logic value "0". Moreover, the non-linear effect induced by the diodes can also be observed in Fig. 2.9(b) if the voltage V_a exhibits a small negative amplitude.

2.6 CONCLUSION

A novel strong PUF architecture is designed based on the on-chip workload-aware multi-phase voltage regulators (WAMPVRs).Through exploiting the physical randomness of the flying capacitors in the multi-phase switched-capacitor (SC) voltage converter, the strong PUF primitive we designed achieves a nearly 51.3% inter-HD and 98.5% reliability. Furthermore, in the WAMPVR-based strong PUF architecture we proposed, an approximated constant input power is achieved against side-channel attacks while a non-linear transformation block is utilized to add non-linearity against machine-learning attacks. As demonstrated in the results, for the designed strong PUF primitive, after enabling 1×10^6 (1×10^5) items of data to execute power (machine-learning) attacks, the prediction accuracy is about 0.52 (0.53). By contrast, the prediction accuracy is about 0.98 (0.999) when power (machine-learning) attacks are performed on the conventional PUF design under the assistance of 26×10^3 (39.2×10^3) items of data.

CHAPTER 3

USING BALANCED LOGIC GATES TO DESIGN AN INNOVATE STRONG PUF IN IOT SECURITY

3.1 MOTIVATION

A PUF primitive is designed on a power delivery system as a countermeasure to both machine learning attacks and side-channel attacks[1]. The advantage of the proposed CoRe PUF design is that by attaching additional collection devices of logical signal, the complexity of remodeling is greatly reduced. Besides, since the PUF is redesigned on an existing functional device, the overhead is greatly reduced. Comparing to former PUF designs, the proposed CoRe PUF gives consideration to features of both security and system function. However, the drawback of the proposed PUF design is still obvious. The PUF primitive, after all, needs to gain particular input binary sequence whenever the CRP is recorded or validated. In any case above, the PRNG in the CoRe PUF need to provide particular control signal. When that happens, the power supply must be cut off in order to realize the authentication of PUFs. Another potential vulnerability is that the defensing to machine learning attacks and side-channel attacks is " nonsynchronous". If attackers are able to acquire the PUF behavior with hardware Trojan or other approaches, they can easily categorize collected signal and choose to crack PUF and PRNG one by one. In this section, a new PUF primitive, WDDL-based AES strong PUF, is introduced to mitigate aforementioned

[1]The content of this Chapter partially has been published in [60].

vulnerabilities. Similar to the CoRe PUF, the WDDL-AES PUF is also redesigned on a functional device which eliminate the overhead when introducing a non-functioning part. Comparing to the CoRe PUF in Chapter 2, the new technique can introduce more entropy of input signals against side-channel attacks. And a non-linear function is also added to enhance the non-linearity of internal logic, which aims to reduce the linear correlation of CRPs and vulnerabilities to machine learning attacks.

3.2 ARCHITECTURE DESIGN

The internal architecture of a WDDL-based AES strong PUF primitive is shown in Fig. 3.1. The 128-bit AES[2] cryptographic circuit has 16 number of S-boxes in each encryption round. Since all the S-boxes are fully implemented with WDDL gates, the dynamic power dissipation of each S-box within a clock period is the same under any input data regardless of PVT variations. In the WDDL-based AES strong PUF primitive, 16 number of S-boxes are uniformly divided into four groups: $group_1$, $group_2$, $group_3$, and $group_4$. As shown in Fig. 3.1, the power supply ports of S-box$_{4(i-1)+1}$, $(i = 1, 2, 3, 4)$, S-box$_{4(i-1)+2}$, S-box$_{4(i-1)+3}$, and S-box$_{4(i-1)+4}$ are connected with the line W_i. The 32-bit binary input data $A = (a_1, a_2, ..., a_{32})_2$ is utilized for generating different dynamic power signatures for the corresponding groups, as shown in Fig. 3.1. A tiny resistor $R_{in,i}$ is inserted between the supply voltage source V_{dd} and the line W_i for sensing the current fluctuations induced by the load capacitance mismatches in the WDDL-based S-boxes under different input data. If the resistors $R_{in,1}$, $R_{in,2}$, $R_{in,3}$, and $R_{in,4}$ in Fig. 3.1 are designed with the same resistance, the

[2]The number of CRPs can be further increased if a 196-bit AES or 256-bit AES is utilized for building the WDDL-based AES strong PUF.

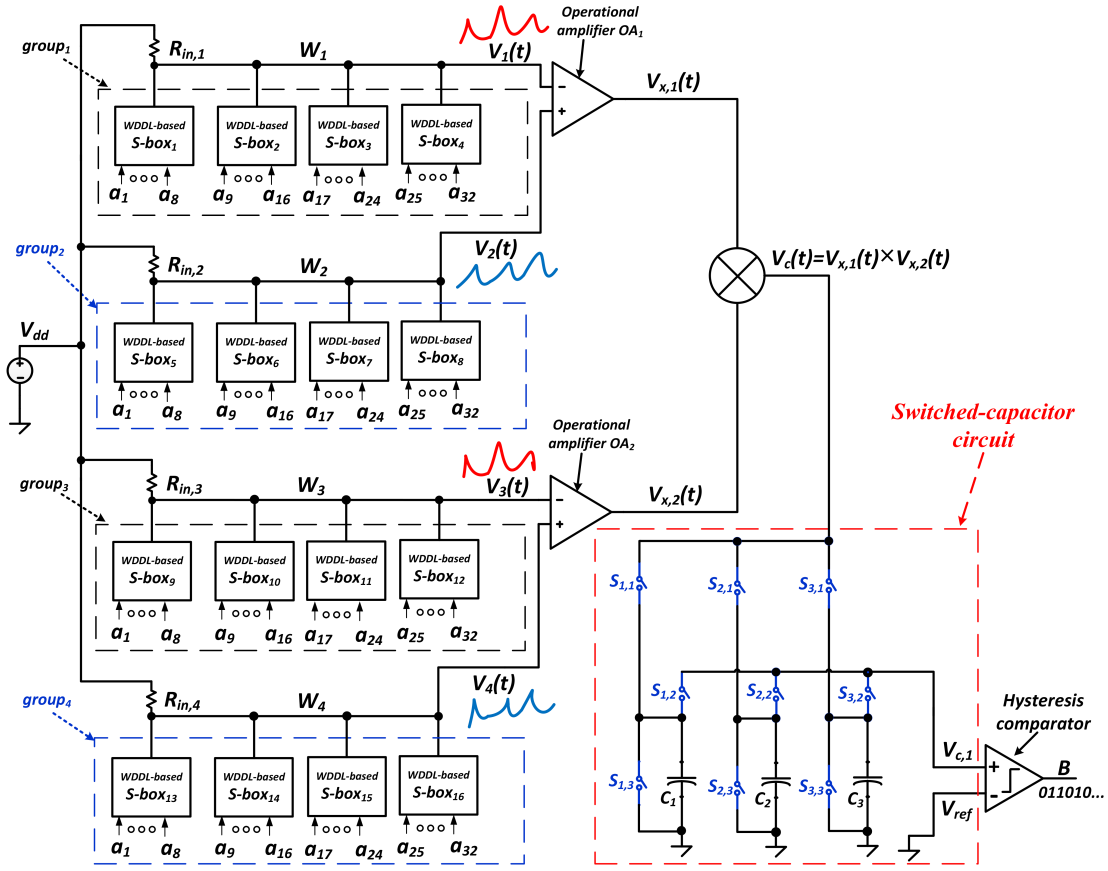


Fig. 3.1: Architecture of a WDDL-based AES strong PUF primitive (the total number N of digital input bits of the AES cryptographic circuit is 128).

differences among the electric potentials[3] $V_1(t)$, $V_2(t)$, $V_3(t)$, and $V_4(t)$ reflect the internal load capacitance mismatches among $group_1$, $group_2$, $group_3$, and $group_4$. In Fig. 3.1, one operational amplifier OA_1 is selected for magnifying the difference between $V_1(t)$ and $V_2(t)$ to generate the critical voltage $V_{x,1}(t)$ while the other operational amplifier OA_2 is utilized to amplify the difference between $V_3(t)$ and $V_4(t)$ for outputting the critical voltage $V_{x,2}(t)$. Moreover, as shown in Fig. 3.1, a non-linear product function is applied on the voltages $V_{x,1}(t)$ and $V_{x,2}(t)$ to generate another critical voltage $V_c(t)$ that can be used for achieving the high non-linear output responses against machine-learning attacks.

[3] $V_i(t)$ is the electric potential of the line W_i where t is the timing, as shown in Fig. 3.1

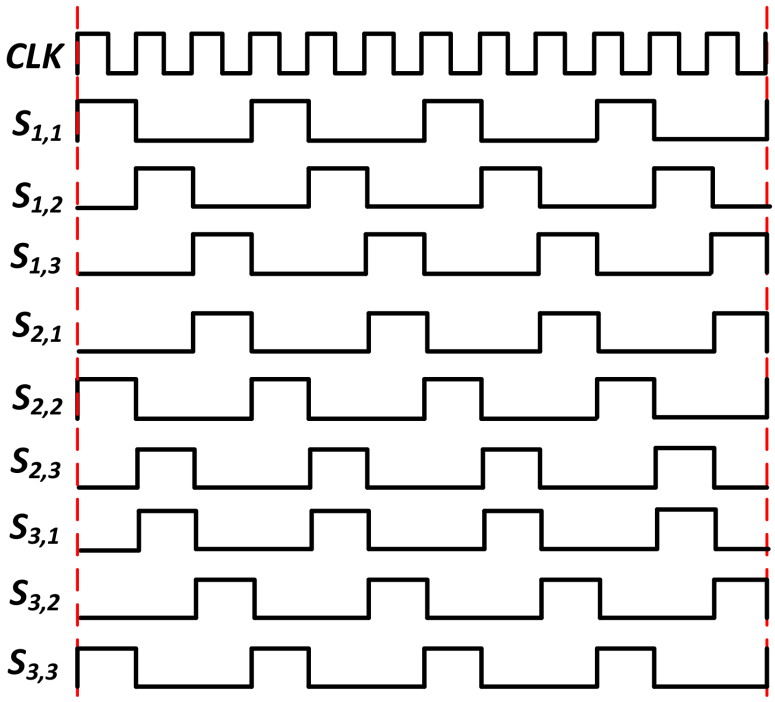


Fig. 3.2: Waveform of control signal of the switch $S_{i_1,j}$ (CLK is the clock signal of the input data A. $S_{i_1,j} = 1$ represents the switch $S_{i_1,j}$ is in on-state, and *vice versa*).

Once the critical voltage $V_c(t)$ is obtained, the following step is converting the analog voltage $V_c(t)$ into the digital confidential data B for authentication. As shown in Fig. 3.1, three flying capacitors C_1, C_2, and C_3 are utilized for sampling the DC component of $V_c(t)$ within each clock period. The waveform of control signal of the switch $S_{i_1,j}$, $(i_1, j = 1, 2, 3)$ is shown in Fig. 3.2. The switched-capacitor circuit in Fig. 3.1 has three basic working phases: *charging* phase, *output* phase, and *discharging* phase. For instance, if the flying capacitor C_1 is in *charging* phase, the switch $S_{1,1}$ is in on-state while the switches $S_{1,2}$ and $S_{1,3}$ are in off-state, as shown in Fig. 3.2. Consequently, the flying capacitor C_1 will have been charged by the critical voltage $V_c(t)$ for a clock period. However, when the *charging* phase ends, the flying capacitor C_1 will be switched to *output* phase. The switch $S_{1,2}$ will be activated to

output the DC component $V_{c,1}$ of $V_c(t)$ within a clock period to generate the binary secret data B by using a hysteresis comparator. If $V_{c,1} \geq V_{ref} = 0$ V, the output binary data $B = 1$. Otherwise, $B = 0$. After the *output* phase, the flying capacitor C_1 will enter into the last working phase: *discharging* phase. The DC voltage $V_{c,1}$ of the flying capacitor C_1 is discharged to 0 V to initialize the next sampling behavior by activating the switch $S_{1,3}$. Please kindly note that only one of the switches ($S_{1,1}$, $S_{1,2}$, and $S_{1,3}$) is active under any timing, as shown in Fig. 3.2.

For the switched-capacitor circuit in Fig. 3.1, the behaviors of the flying capacitors C_1, C_2, and C_3 are mutually exclusive. For instance, when the flying capacitor C_1 is in *charging* phase, the flying capacitors C_2 and C_3 are in *output* phase and *discharging* phase, respectively, as shown in Fig. 3.2. Therefore, the critical analog voltage $V_c(t)$ can be continuously converted into the digital binary data B through utilizing the switched-capacitor circuit.

3.3 PERFORMANCE EVALUATION

In the designed WDDL-based AES strong PUF as shown in Fig. 3.1, the transient power consumption $P_k(A_k^*, t)$ and current $I_k(A_k^*, t)$ of the kth, $(k = 1, 2, ..., 16)$ WDDL-implemented S-box (S-box$_k$) within a clock period of $group_1$, respectively, can be denoted as

$$P_k(A_k^*, t) = f_c V_1^2(t) \sum_{\omega=1}^{\alpha} C_{\omega,k}(A_k^*, t), \tag{3.1}$$

$$I_k(A_k^*, t) = f_c V_1(t) \sum_{\omega=1}^{\alpha} C_{\omega,k}(A_k^*, t) \tag{3.2}$$

where f_c is the clock frequency of the input data and A_k^* is the 8-bit input data of S-box$_k$

that is written as $A_k^* = (a_{8(k-1)+1}, ..., a_{8k})_2$ and $A_{l+4g}^* = A_l^*, (l = 1, 2, 3, 4$ and $g = 0, 1, 2, 3)$, as shown in Fig. 3.1. α is the total number of $0 \rightarrow 1$ transitions happened in the S-box$_k$ under the input data A_k^* while $C_{\omega,k}(A_k^*, t)$ is the load capacitance of the ωth, $(\omega = 1, 2, ..., \alpha)$ WDDL gate in S-box$_k$ that is related with the ωth $0 \rightarrow 1$ transition under the input data A_k^* at the timing t. Since the clock frequency f_c has an approximated linear relationship with the environmental temperature T_e and the supply voltage V_{dd}, respectively, the clock frequency f_c can be approximated as [61]

$$f_c \approx \frac{f_{c,0}(a_0 + a_1 T_e)(b_0 + b_1 V_{dd})}{(a_0 + a_1 T_{e,0})(b_0 + b_1 V_{dd,0})} \tag{3.3}$$

where $f_{c,0}$ is the clock frequency under the ideal environmental temperature $T_{e,0}$ and supply voltage $V_{dd,0}$. a_0 and a_1 (b_0 and b_1) are the coefficients of linear approximation of the environmental temperature T_e (supply voltage V_{dd}).

By utilizing the equation $V_1(t) = V_{dd} - R_{in,1} \sum_{k=1}^{4} I_k(A_k^*, t)$ that is derived from Fig. 3.1, the supply voltage $V_1(t)$ of $group_1$ is approximated as

$$V_1(t) \approx \frac{V_{dd}}{1 + \frac{\sum_{k=1}^{4} \sum_{\omega=1}^{\alpha} f_{c,0}(a_0 + a_1 T_e)(b_0 + b_1 V_{dd}) C_{\omega,k}(A_k^*, t) R_{in,1}}{(a_0 + a_1 T_{e,0})(b_0 + b_1 V_{dd,0})}}. \tag{3.4}$$

Similarly, the supply voltages of the rest groups: $V_2(t)$, $V_3(t)$, and $V_4(t)$ can also be determined by following the aforementioned steps.

Once $V_1(t)$, $V_2(t)$, $V_3(t)$, and $V_4(t)$ are obtained, the critical voltages $V_{x,1}(t)$ and $V_{x,2}(t)$

in Fig. 3.1, respectively, can be determined as

$$V_{x,1}(t) = A_{v,1}(V_2(t) - V_1(t)) + A_{c,1}\frac{V_2(t) + V_1(t)}{2}, \tag{3.5}$$

$$V_{x,2}(t) = A_{v,2}(V_4(t) - V_3(t)) + A_{c,2}\frac{V_3(t) + V_4(t)}{2} \tag{3.6}$$

where $A_{v,1}$ ($A_{v,2}$) and $A_{c,1}$ ($A_{c,2}$) are the differential gain and the common-mode gain of the operational amplifier OA_1 (OA_2), respectively. In Fig. 3.1, if the switching period of the switches $S_{1,1}$, ..., $S_{3,3}$ is designed three times of the clock period of the input data A, the DC component $V_{c,1}$ of the critical voltage $V_c(t)$ within a clock period can be derived as

$$V_{c,1} = \int_{nT_c}^{(n+1)T_c} V_c(t)dt = \int_{nT_c}^{(n+1)T_c} V_{x,1}(t)V_{x,2}(t)dt \tag{3.7}$$

where T_c is the clock period and $n, (n = 0, 1, 2, ...)$ represents the number of the clock period. If $V_{c,1} \geq V_{ref} = 0$ V, the output binary data $B = 1$. Otherwise, $B = 0$.

In order to model the relationship between the input challenge A and the output response B for the proposed WDDL-based AES strong PUF, the next step that needs to be finished is to determine the function $C_{\omega,k}(A_k^*, t)$. So as to do the analysis in a more efficient way, let us write $C_{\omega,k}(A_k^*, t)$ as

$$C_{\omega,k}(A_k^*, t) = C_{\omega,k}^*(A_k^*)C_{\omega,k}^{**}(t) \tag{3.8}$$

where $C_{\omega,k}^*(A_k^*)$ is the component of the ω^{th} load capacitance in S-box$_k$ that is determined by the input data A_k^* while $C_{\omega,k}^{**}(t)$ is the component of the ω^{th} load capacitance in S-box$_k$

determined by the timing t. $C^{**}_{\omega,k}(t)$ can be denoted as

$$C^{**}_{\omega,k}(t) = \begin{cases} 1 & , t = t_\omega + nT_c \\ 0 & , Otherwise \end{cases} \tag{3.9}$$

where t_ω represents the activation timing within the *1st* clock period for the ωth load capacitor that is related with the ωth $0 \rightarrow 1$ transition. Since (3.9) indicates $C^{**}_{\omega,k}(t)$ is a period signal, $C^{**}_{\omega,k}(t)$ can be approximately unfolded with Fourier series as shown below

$$C^{**}_{\omega,k}(t) \approx \frac{c_{0,\omega}}{2} + \sum_{h=1}^{n_1}(c_{h,\omega}cos\frac{2\pi h}{T_c}t + d_{h,\omega}sin\frac{2\pi h}{T_c}t) \tag{3.10}$$

where $c_{0,\omega}$, $c_{1,\omega}$, ..., $c_{n_1,\omega}$, $d_{1,\omega}$, ..., $d_{n_1,\omega}$ (n_1) are the coefficients (degree) of the approximated Fourier series of $C^{**}_{\omega,k}(t)$.If we discretize the time region $[nT_c, (n+1)T_c)$ with n_2 number of timing points, the values of timing t are nT_c, $(n+\frac{1}{n_2})T_c$, $(n+\frac{2}{n_2})T_c, \cdots, (n+\frac{n_2-1}{n_2})T_c$. For a WDDL-based S-box, the total load capacitance is a constant α within a clock period under different input data. As a result, we can obtain $c_{0,\omega} = 2\alpha/n_2$ with

$$\begin{aligned} \int_{nT_c}^{(n+1)T_c}(\sum_{\omega=1}^{\alpha} C^{**}_{\omega,k}(t))dt &\approx \int_{nT_c}^{(n+1)T_c}(\frac{c_{0,\omega}}{2} + \sum_{h=1}^{n_1}(c_{h,\omega}cos\frac{2\pi h}{T_c}t + d_{h,\omega}sin\frac{2\pi h}{T_c}t))dt \\ &= \int_{nT_c}^{(n+1)T_c}\frac{c_{0,\omega}}{2}dt = \frac{c_{0,\omega}}{2}T_c = \alpha\frac{T_c}{n_2}. \end{aligned} \tag{3.11}$$

When the coefficient $c_{0,\omega}$ is obtained, the values of the rest parameters can also be estimated. Before determining the values of the rest parameters, two n_1-dimension vectors $C_\omega = [c_{1,\omega}, ..., c_{n_1,\omega}]$ and $D_\omega = [d_{1,\omega}, ..., d_{n_1,\omega}]$ need to be defined, and two $n_1 \times n_2$ matrices

M_1 and M_2 can be, respectively, set as ...

$$M_1 = \begin{pmatrix} cos(2\pi n) & \cdots & cos(2\pi(n + \frac{n_2-1}{n_2})) \\ cos(4\pi n) & \cdots & cos(4\pi(n + \frac{n_2-1}{n_2})) \\ \vdots & \ddots & \vdots \\ cos(2n_1\pi n) & \cdots & cos(2n_1\pi(n + \frac{n_2-1}{n_2})) \end{pmatrix}, \tag{3.12}$$

$$M_2 = \begin{pmatrix} sin(2\pi n) & \cdots & sin(2\pi(n + \frac{n_2-1}{n_2})) \\ sin(4\pi n) & \cdots & sin(4\pi(n + \frac{n_2-1}{n_2})) \\ \vdots & \ddots & \vdots \\ sin(2n_1\pi n) & \cdots & sin(2n_1\pi(n + \frac{n_2-1}{n_2})) \end{pmatrix}. \tag{3.13}$$

By utilizing (3.9), (3.10), (3.12), and (3.13), we can acquire

$$C^{(0)} + M_1^T C_\omega^T + M_2^T D_\omega^T = C_\omega^{***} \tag{3.14}$$

where the two $n_2 \times 1$ matrices $C^{(0)}$ and C_ω^{***} are $[2\alpha/n_2, 2\alpha/n_2, ..., 2\alpha/n_2]^T$ and $[C_{\omega,k}^{**}(nT_c), C_{\omega,k}^{**}((n+\frac{1}{n_2})T_c), ..., C_{\omega,k}^{**}((n+\frac{n_2-1}{n_2})T_c)]^T$, respectively. If n_1 is set as $n_2/2$, by solving (3.14), the rest coefficients: $c_{1,\omega}$, ..., $c_{n_1,\omega}$, $d_{1,\omega}$, ..., $d_{n_1,\omega}$ can be determined.

Eventually, assume the capacitance mismatch of a WDDL gate induced by the process variations is β. If a Monte Carlo simulation is selected to evaluate the performance of the WDDL-based AES strong PUF, the capacitance function $C_{\omega,k}^{*}(A_k^{*})$ will approximately conform to a normal distribution $N \sim (C_0, (\frac{\beta}{6})^2)$ where C_0 is the identically designed load

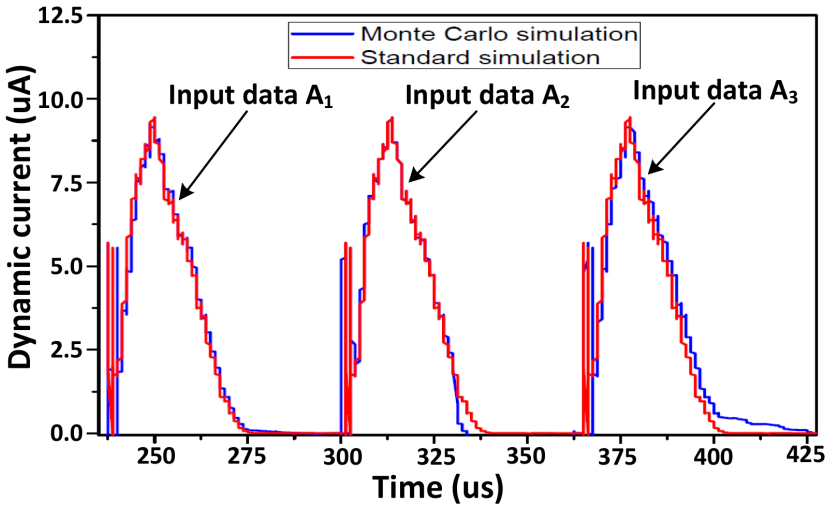
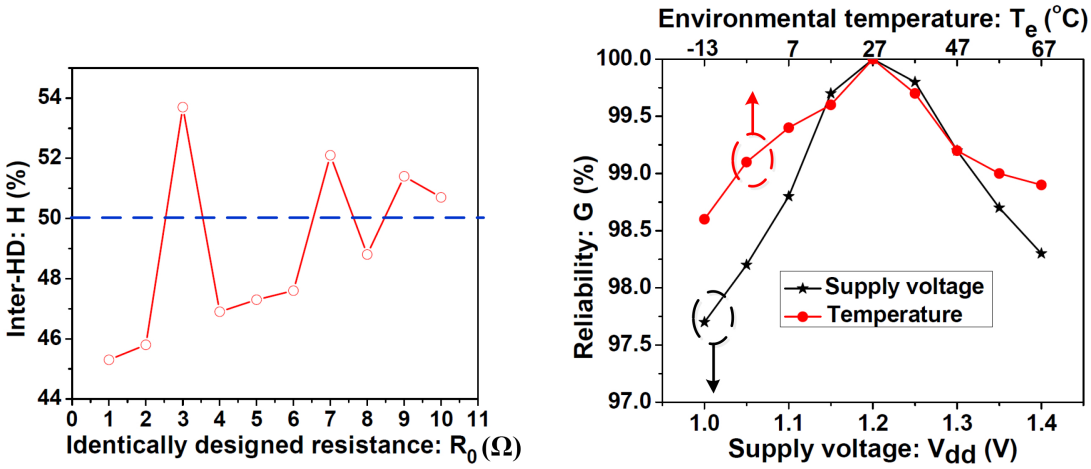


Fig. 3.3: Dynamic current profile of $group_1$ (Four number of WDDL-based S-boxes are used. A_1, A_2, and A_3 are three different 32-bit binary data).

capacitance. Likewise, if the resistance (common-mode gain) mismatch of two identically designed input resistors (operational amplifiers) impacted by the process variations is β_1 (β_2), the resistances of $R_{in,1}, ..., R_{in,4}$ (common-mode gains $A_{c,1}$ and $A_{c,2}$) in Fig. 3.1 will also have an approximated normal distribution $N \sim (R_0, (\frac{\beta_1}{6})^2)$ ($N \sim (0, (\frac{\beta_2}{6})^2)$) where R_0 is the identically designed resistance for $R_{in,1}, ..., R_{in,4}$ under the Monte Carlo simulation.

Therefore, if K number of WDDL-based AES strong PUFs are utilized to generate the K-bit output response B and M number of K-bit WDDL-based AES strong PUFs are used to generate the challenge-to-response pairs (CRPs), the inter-hamming distance (HD) H of the WDDL-based AES strong PUF can be determined by using the aforementioned model. Similarly, if a single WDDL-based AES strong PUF is assessed under different environmental temperatures and supply voltages, the reliability G of the WDDL-based AES strong PUF can also be estimated by considering the environmental temperature T_e and the supply voltage V_{dd} with normal distributions.



(a) (b)

Fig. 3.4: Performance evaluation for the WDDL-based AES strong PUF. (a) Inter-HD H versus identically designed resistance R_0 ($M = 100$ and $K = 10$). (b) Reliability G versus supply voltage V_{dd} and environmental temperature T_e ($M = 50$ and $K = 10$).

A 128-bit AES cryptographic circuit is designed and simulated in Cadence with the 130 nm CMOS technology kits. As shown in Fig. 3.3, when the input data $A = (a_1, a_2, ..., a_{32})_2$ make transitions from A_1 to A_2 and from A_2 to A_3, $group_1$ in Fig. 3.1 exhibits a constant dynamic current if a standard simulation[4] is enabled. That indicates the WDDL-based S-box has a constant dynamic power consumption under different input data regardless of the variations of PVT. However, if a Monte Carlo simulation[5] is performed, the dynamic current signature of the WDDL-based S-box varies when the input data change. The simulation results demonstrate that the random load capacitance mismatches in WDDL gates can achieve different dynamic current signatures for authentication.

By combining the values of the critical parameters that are extracted from the 130 nm CMOS technology node in Cadence with the aforementioned mathematical model, the

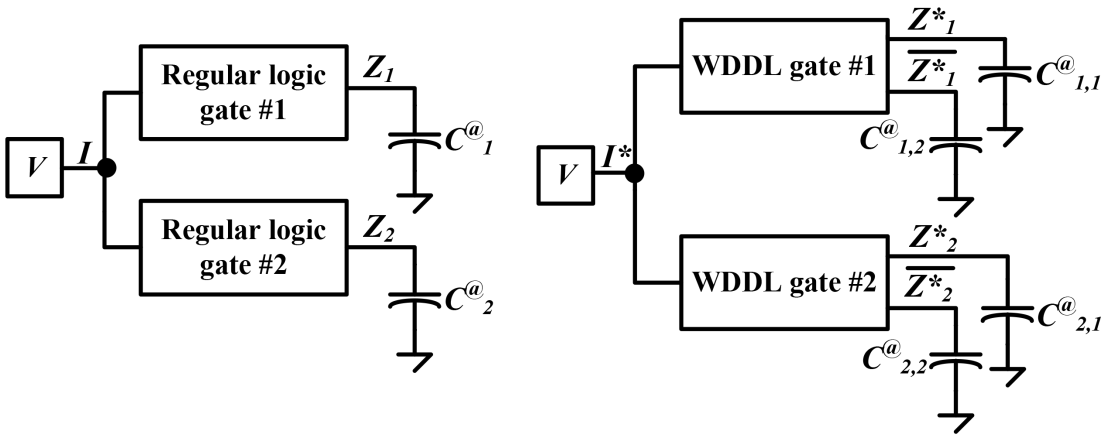
[4]Standard simulation means the simulation neglects the variations of process, voltage, and temperature (PVT).

[5]Monte Carlo simulation represents the variations of process are included in the simulation.

performance of the proposed WDDL-based AES strong PUF can be assessed. As shown in Fig. 3.4(a), the inter-HD H of the WDDL-AES strong PUF approaches the ideal value 50% when the identically designed resistance R_0 increases from 1 Ω to 10 Ω. That indicates a higher R_0 enables the WDDL-based AES strong PUF to achieve a better uniqueness due to a larger variance of physical randomness. In addition, the reliability G of the WDDL-based AES strong PUF is shown in Fig. 3.4(b). The worst reliability G is about 97.7% (V_{dd} =1.0 V and T_e =27 °C) when the ideal environmental setting is: the supply voltage $V_{dd,0}$ =1.2 V and the environmental temperature $T_{e,0}$ =27 °C.

As shown in Fig. 3.1, the proposed WDDL-based AES strong PUF contains 32 input challenge bits: $a_1, a_2, ..., a_{32}$ and 1 output response bit: B. When the strong PUF chip is utilized for authentication, 1000 random input challenges out of the 2^{32} total possible input challenges can be selected for generating 1000 output response bits. Under the 1000 given input challenges, if the attacker predicts the 1000 output response bits with a 0.5 random guess probability, only about 500 output response bits may be matched. In contrast, when the host predicts the 1000 output response bits under the assistance of the stored CRPs, over 970 output response bits can be matched since the worst reliability of the proposed strong PUF is over 97% (as shown in Fig. 3.4(b)).

For the proposed WDDL-based AES strong PUF, its uniqueness is closed to the ideal value (50%) as shown in Fig. 3.4(a). Under such a condition, if 1000 stochastic input challenges are used for creating 1000 output response bits to achieve authentication, the probability of obtaining the same 1000 output response bits for two different strong PUF chips is around $1/2^{1000}$. This indicates it is impossible to receive the same 1000 output



(a) (b)

Fig. 3.5: Two logic gates share the same power supply. (a) Regular logic gates. (b) WDDL gates.

response bits in practice even if millions of chips are manufactured.

Aging issues are inevitable for hardware security primitives like silicon PUFs. Commonly, negative bias temperature instability (NBTI), hot carrier injection (HCI), and time dependent dielectric breakdown (TDDB) [62] are the three critical mechanisms that cause unreliable issues to silicon PUFs. However, as compared to other regular strong PUFs such as arbiter PUFs, the proposed strong PUF is more robust against aging-induced reliability issues. The plausible explanation is that the proposed strong PUF utilizes a large number of logic gates (about 24270) to generate the random load capacitance mismatches. Hence, it is deemed to output the reliable output response even though a small amount of capacitance mismatch is altered by the aging issues.

3.4 ROBUSTNESS AGAINST POWER ATTACKS

To leak the critical information of the WDDL-based AES strong PUF, power attacks

may be deployed by the attacker to estimate the relationship between the input power and the output response of the strong PUF. Hence, so as to demonstrate the proposed PUF is sufficiently robust against power attacks, in this Section, the input power entropy of the proposed PUF is fully analyzed and an advanced power attack is performed on the proposed PUF to explore the relationship between the input power and the output response.

3.4.1 INPUT POWER ENTROPY ANALYSIS

Input power entropy is a significant metric to quantify the robustness of a system against power attacks [63, 64]. For the proposed WDDL-based AES strong PUF in Fig. 3.1, the attacker may perform a power attack on it through monitoring and analyzing its power trace. In order to simplify the power attack, the attacker may dynamically alter the input data A_k^* of the single S-box$_k$ while maintaining a constant input data for the rest of the S-boxes. As a result, only the single S-box$_k$ exhibits the high dynamic power dissipation that may leak the critical information to the attacker. Accordingly, for power attacks, the security of a single WDDL-based S-box dominates the security of the WDDL-based AES strong PUF. As shown in Fig. 3.5(a), two regular logic gates connect with the same power source V. When the output Z_1 of the regular logic gate #1 makes a transition from 0 to 1 and the output Z_2 of the regular logic gate #2 is kept a constant, the transient current I of the power source V is $I = f_c^{@} V C_1^{@}$ where $f_c^{@}$ is the clock frequency of the logic gates and $C_1^{@}$ is the load capacitance of the regular logic gate #1. Hence, $C_1^{@}$ can be determined as $C_1^{@} = I/(f_c^{@} V)$ and the load capacitance $C_2^{@}$ of the regular logic gate #2 can also be estimated in a similar way. That indicates the internal parameters of the regular logic gates can be leaked to the attacker without much effort by utilizing the power analysis.

However, for the WDDL gates in Fig. 3.5(b), the case is quite different. The four different transient current signatures: I_1^*, I_2^*, I_3^*, and I_4^* of the power source V induced by four different output logic transitions are summarized in Table I. By considering Fig. 3.5(b) and Table I, the below equation (3.15) is obtained with

$$f_c^{@} V \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} C_{1,1}^{@} \\ C_{1,2}^{@} \\ C_{2,1}^{@} \\ C_{2,2}^{@} \end{pmatrix} = \begin{pmatrix} I_1^* \\ I_2^* \\ I_3^* \\ I_4^* \end{pmatrix} \tag{3.15}$$

where $C_{1,1}^{@}$, $C_{1,2}^{@}$, $C_{2,1}^{@}$, and $C_{2,2}^{@}$ are the corresponding load capacitance values in Fig. 3.5(b). After solving (3.15), we can obtain

$$C_{1,1}^{@} - C_{1,2}^{@} = I_1^* - I_3^*, \tag{3.16}$$

$$C_{2,1}^{@} - C_{2,2}^{@} = I_2^* - I_4^*. \tag{3.17}$$

Therefore, even if the power analysis is performed on the WDDL gates, a large amount of uncertainty is still existing among the internal parameters: $C_{1,1}^{@}$, $C_{1,2}^{@}$, $C_{2,1}^{@}$, and $C_{2,2}^{@}$.

In a WDDL-based S-box, assume X number of WDDL gates connect with the same power source. As demonstrated above, by using the power analysis, the attacker can only determine the load capacitance mismatch between the output and the complementary output for each WDDL gate. Since the capacitance mismatch of two identically designed load capacitors in the WDDL gates induced by the process variations is β, let us assume the

TABLE 3.1: Four different transient current signatures: I_1^*, I_2^*, I_3^*, and I_4^* induced by four different output logic transitions of Fig. 3.5(b).

Output logic / Transient current	WDDL gate #1		WDDL gate #2	
	Output Z^*_1 0→1	Complementary output $\overline{Z^*_1}$ 0→1	Output Z^*_2 0→1	Complementary output $\overline{Z^*_2}$ 0→1
$I^*=I^*_1$	✓		✓	
$I^*=I^*_2$	✓			✓
$I^*=I^*_3$		✓	✓	
$I^*=I^*_4$		✓		✓

load capacitance $C \in [C_0 - \frac{\beta}{2}, C_0 + \frac{\beta}{2}]$. If N_1 number of different load capacitance values are existing in the WDDL gates, the $j_1 th$, $(j_1 = 1, 2, ..., N_1)$ load capacitance value C_{j_1} and the corresponding probability $P(C_{j_1})$, respectively, are

$$C_{j_1} = (C_0 - \frac{\beta}{2}) + \frac{j_1 - 1}{N_1 - 1}\beta, \tag{3.18}$$

$$P(C_{j_1}) \approx \frac{\frac{1}{\sqrt{2\pi}\frac{\beta}{6}} exp(-\frac{(C_{j_1}-C_0)^2}{2(\frac{\beta}{6})^2})}{\sum_{j_1=1}^{N_1} \frac{1}{\sqrt{2\pi}\frac{\beta}{6}} exp(-\frac{(C_{j_1}-C_0)^2}{2(\frac{\beta}{6})^2})}. \tag{3.19}$$

Assume the load capacitance mismatch of the $i_2 th$, $(i_2 = 1, 2, ..., X)$ WDDL gate in the WDDL-based S-box is ΔC_{i_2}. Then the probability $P^*(C_{j_1}, C_{j_1} + \Delta C_{i_2})$ of the $i_2 th$ WDDL gate with the load capacitance C_{j_1} of the output and the load capacitance $C_{j_1} + \Delta C_{i_2}$ of the complementary output can be written as

$$P^*(C_{j_1}, C_{j_1} + \Delta C_{i_2}) = \begin{cases} \frac{P(C_{j_1})}{\sum_{j_1=1}^{N_2} P(C_{j_1})} & , j_1 = 1, 2, ..., N_2 \\ 0 & , Otherwise \end{cases} \tag{3.20}$$

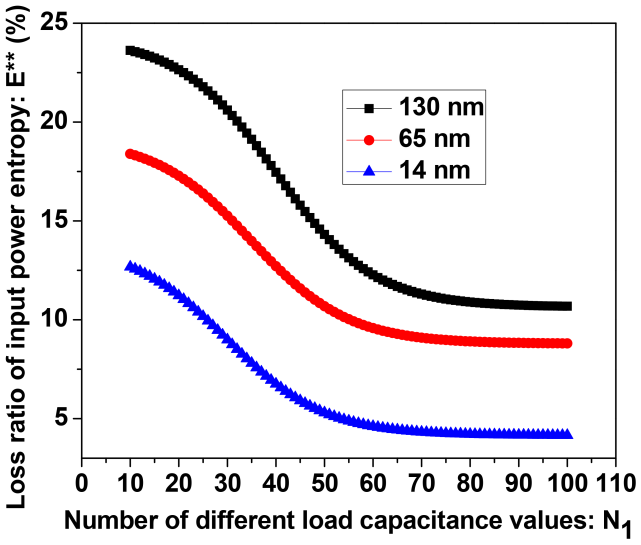


Fig. 3.6: Loss ratio E^{**} of input power entropy versus number of different load capacitance values N_1 for the WDDL-based AES strong PUF against the power attack.

where N_2 and $P(C_{j_1})/(\sum_{j_1=1}^{N_2} P(C_{j_1}))$, respectively, are

$$N_2 = [\frac{\beta - \Delta C_{i_2}}{\beta}(N_1 - 1)] + 1, \tag{3.21}$$

$$\frac{P(C_{j_1})}{\sum_{j_1=1}^{N_2} P(C_{j_1})} \approx \frac{\frac{exp(-\frac{(C_{j_1}-C_0)^2}{2(\frac{\beta}{6})^2})}{\sum_{j_1=1}^{N_1} exp(-\frac{(C_{j_1}-C_0)^2}{2(\frac{\beta}{6})^2})}}{\sum_{j_1=1}^{N_2}(\frac{exp(-\frac{(C_{j_1}-C_0)^2}{2(\frac{\beta}{6})^2})}{\sum_{j_1=1}^{N_1} exp(-\frac{(C_{j_1}-C_0)^2}{2(\frac{\beta}{6})^2})})}. \tag{3.22}$$

Therefore, the input power entropy E of the WDDL-based AES strong PUF after performing the power attack is estimated as

$$E = -\sum_{i_2=1}^{X}\sum_{j_1=1}^{N_1} P^*(C_{j_1}, C_{j_1} + \Delta C_{i_2}) log_2^{P^*(C_{j_1}, C_{j_1}+\Delta C_{i_2})}. \tag{3.23}$$

However, before performing the power attack on the WDDL-based AES strong PUF, the

load capacitance C_{j_1} and complementary load capacitance C_{j_2}, $(j_2 = 1, 2, ..., N_1)$ of the i_2th WDDL gate in the WDDL-based S-box are independent. Hence, the original input power entropy E^* of the WDDL-based AES strong PUF is

$$E^* = -X \sum_{j_2=1}^{N_1} \sum_{j_1=1}^{N_1} P(C_{j_1})P(C_{j_2}) log_2^{P(C_{j_1})P(C_{j_2})}. \tag{3.24}$$

As a result, the loss ratio E^{**} of input power entropy of the WDDL-based AES strong PUF after performing the power attack can be defined as $E^{**} = (E/E^*) \times 100\%$. As shown in Fig. 3.6, the maximum loss ratio of the input power entropy of the WDDL-based AES strong PUF with the 130 nm CMOS technology node is less than 25% after executing the power attack. Moreover, when the CMOS technology node is scaled from 130 nm to 14 nm by using the process mismatch data from [17], the corresponding maximum loss ratio can be reduced below 15%. Accordingly, the WDDL-based AES strong PUF exhibits a good theoretical resilience against power attacks. However, to guarantee the WDDL-based AES strong PUF is sufficiently robust against power attacks in practice, a state-of-the-art power attack is studied on the proposed PUF in Section 3.4.2.

3.4.2 STATE-OF-THE-ART POWER ATTACK ON THE PROPOSED PUF

The main intention of performing power attacks on the WDDL-based AES strong PUF is precisely predicting the output response B by measuring the corresponding input power P_{in} of the PUF. If the mathematical function $f(P_{in})$ is used to map the relationship between the input power P_{in} and the output response B, under the assistance of Taylor series, the

relationship between B and P_{in} can be expressed as

$$\begin{aligned} B = f(P_{in}) &= \sum_{i_3=0}^{\infty} \frac{\mathrm{d}^{i_3} f(P_{in})/\mathrm{d}(P_{in})^{i_3}}{i_3!}(P_{in})^{i_3} \\ &\approx \sum_{i_4=0}^{m} \lambda_{i_4}(P_{in})^{i_4} \end{aligned} \tag{3.25}$$

where m is the approximated degree of the Taylor series and the coefficient $\lambda_{i_4}, (i_4 = 0, 1, ..., m)$ of $(P_{in})^{i_4}$ is

$$\lambda_{i_4} = \frac{\mathrm{d}^{i_4} f(P_{in})/\mathrm{d}(P_{in})^{i_4}}{i_4!}. \tag{3.26}$$

Since the output response B is a binary data, two critical parameters B^* and B^{**} can be utilized by the attacker to predict the output response $B = 0$ and $B = 1$, respectively. The parameters B^* and B^{**} can, respectively, be defined as

$$B^* = \sum_{i_5=0}^{m_1} \lambda^*_{i_5}(P_{in})^{i_5}, \tag{3.27}$$

$$B^{**} = \sum_{i_6=0}^{m_2} \lambda^{**}_{i_6}(P_{in})^{i_6} \tag{3.28}$$

where m_1 (m_2) is the degree of the series $\sum_{i_5=0}^{m_1} \lambda^*_{i_5}(P_{in})^{i_5}$ ($\sum_{i_6=0}^{m_2} \lambda^{**}_{i_6}(P_{in})^{i_6}$) and $\lambda^*_{i_5}, (i_5 = 0, 1, ..., m_1)$ ($\lambda^{**}_{i_6}, (i_6 = 0, 1, ..., m_2)$) is the coefficient of $(P_{in})^{i_5}$ ($(P_{in})^{i_6}$).

Assume n_3 number of input power and output response pairs (IPORP) have been gathered by the attacker to perform the power attack on the proposed PUF. Furthermore, within the n_3 number of IPORP, assume there are n_4 number of IPORP with the output response equals to 0: $(P_{in,1},\ 0)$, $(P_{in,2},\ 0)$, ..., $(P_{in,n_4},\ 0)$ and $n_3 - n_4$ number of IPORP with the

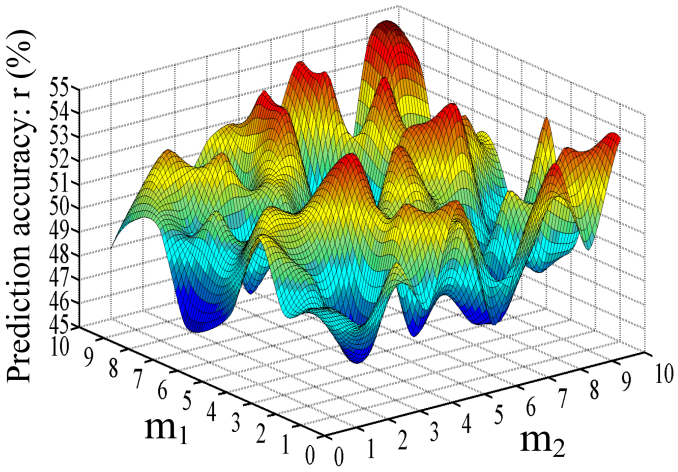


Fig. 3.7: Prediction accuracy r of the power attack under the parameters B^* and B^{**} versus degrees m_1 and m_2 of the series ($n_3 = 1,000,000$).

output response equals to 1: $(P_{in,n_4+1},\ 1)$, $(P_{in,n_4+2},\ 1)$, ..., $(P_{in,n_3},\ 1)$. As a result, if the IPORP $(P_{in,1},\ 0)$, $(P_{in,2},\ 0)$, ..., $(P_{in,n_4},\ 0)$ and $(P_{in,n_4+1},\ 1)$, $(P_{in,n_4+2},\ 1)$, ..., $(P_{in,n_3},\ 1)$ are used for predicting the output response $B = 0$ and $B = 1$, respectively, the two critical parameters B^* and B^{**}, respectively, are modified as

$$B^* = \sum_{j_3=1}^{n_4} \sum_{i_5=0}^{m_1} \lambda_{i_5}^* (P_{in,j_3})^{i_5}, \tag{3.29}$$

$$B^{**} = \sum_{j_4=1}^{n_3-n_4} \sum_{i_6=0}^{m_2} \lambda_{i_6}^{**} (P_{in,j_4})^{i_6}. \tag{3.30}$$

To maximize the prediction accuracy of the power attack on the WDDL-based AES strong PUF, the objective function of the optimization is obtained as

$$\max \quad (B^{**} - B^*)^2 = \Big(\sum_{j_4=1}^{n_3-n_4} \sum_{i_6=0}^{m_2} \lambda_{i_6}^{**} (P_{in,j_4})^{i_6} - \sum_{j_3=1}^{n_4} \sum_{i_5=0}^{m_1} \lambda_{i_5}^* (P_{in,j_3})^{i_5} \Big)^2. \tag{3.31}$$

The input power values $P_{in,j_3}, P_{in,j_4} \in [P_{min}, P_{max}]$ where P_{min} and P_{max} are the minimum power dissipation and maximum power dissipation of the WDDL-based AES strong PUF, respectively. So as to estimate the values of the coefficients $\lambda_{i_5}^{*}$ and $\lambda_{i_6}^{**}$ in (3.31), the partial differential equations of $(B^{**} - B^{*})^2$ need be set to 0. Consequently, we can acquire

$$\begin{pmatrix} 2(B^{**} - B^{*})\frac{\partial(B^{**}-B^{*})}{\partial\lambda_0^{**}} \\ \vdots \\ 2(B^{**} - B^{*})\frac{\partial(B^{**}-B^{*})}{\partial\lambda_{m_2}^{**}} \\ 2(B^{**} - B^{*})\frac{\partial(B^{**}-B^{*})}{\partial\lambda_0^{*}} \\ \vdots \\ 2(B^{**} - B^{*})\frac{\partial(B^{**}-B^{*})}{\partial\lambda_{m_1}^{*}} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.32}$$

Fig. 3.7 shows the relationship between the prediction accuracy r of the power attack on the 130 nm CMOS WDDL-based AES strong PUF and the degrees m_1 and m_2 of the series. After analyzing 1 million number of IPORP, the maximum prediction accuracy r of the power attack is still below 55%. The result reflects it is difficult for the attacker to disclose the confidential information of the WDDL-based AES strong PUF through exploring the power leakage of the PUF.

3.5 RESILIENCE AGAINST MACHINE-LEARNING ATTACKS

Despite the present silicon PUFs may be effective for protecting IoT against some specific malicious attacks like hardware reverse engineering attacks [65, 66], one of the most significant security concerns of the current silicon PUFs is the resilience against machine

learning attacks [57, 58, 67]. As far as we know, high linear relationships exist between the input challenges and the output responses for most conventional silicon PUFs such as ring-oscillator (RO) PUFs [4, 68], arbiter PUFs [69, 57], and clock PUFs [70], which cause them extremely vulnerable to machine-learning attacks. For the WDDL-based AES strong PUF, to assess the corresponding security level against machine-learning attacks, the degree of the non-linearity of the PUF is evaluated and three different deep-learning attacks are performed on the PUF in the following subsections.

3.5.1 ASSESSMENT OF NON-LINEARITY

When it comes to a strong PUF, the degree of the non-linearity between the input challenge and the output response is a critical parameter to reflect the robustness of the strong PUF against machine-learning attacks [71, 57]. Moreover, the linear matching error is a commonly used parameter that represents the degree of the non-linearity of a system [58]. Since $C_{\omega,k}(A_k^*, t)$ represents the ωth load capacitance in S-box$_k$ that is related with the ωth $0 \rightarrow 1$ transition under the input data A_k^* at the timing t of the WDDL-based AES strong PUF, the total load capacitance mismatch $\Delta C_\omega(A_k^*)$ between $group_1$ and $group_2$ in Fig. 3.1 within a clock period is

$$\Delta C_\omega(A_k^*) = \int_{nT_c}^{(n+1)T_c} (\sum_{k=1}^{4} C_{\omega,k}(A_k^*, t) - \sum_{k=5}^{8} C_{\omega,k}(A_k^*, t))dt. \tag{3.33}$$

Similarly, the total load capacitance mismatch $\Delta C'_\omega(A_k^*)$ between $group_3$ and $group_4$ in

Fig. 3.1 within a clock period can also be obtained as

$$\Delta C'_{\omega}(A^*_k) = \int_{nT_c}^{(n+1)T_c} (\sum_{k=9}^{12} C_{\omega,k}(A^*_k, t) - \sum_{k=13}^{16} C_{\omega,k}(A^*_k, t))dt. \tag{3.34}$$

Assume the load capacitance mismatches $\Delta C_{\omega}(A^*_k)$ and $\Delta C'_{\omega}(A^*_k)$ with a linear function are utilized to generate a parameter $V^*_{c,1}$ as shown below to match the DC component $V_{c,1}$ of the critical voltage $V_c(t)$

$$V^*_{c,1} = \gamma_2 \Delta C_{\omega}(A^*_k) + \gamma_1 \Delta C'_{\omega}(A^*_k) + \gamma_0 \tag{3.35}$$

where γ_2, γ_1, and γ_0 are the coefficients of the linear function. If Y number of different input data A^*_k: $A^*_{k,1}$, $A^*_{k,2}$, ..., $A^*_{k,Y}$ are used for evaluating the degree of the non-linearity of the WDDL-based AES strong PUF, the normalized linear matching error ε is

$$\varepsilon = \frac{\sum_{y=1}^{Y} (\gamma_2 \Delta C_{\omega}(A^*_{k,y}) + \gamma_1 \Delta C'_{\omega}(A^*_{k,y}) + \gamma_0 - V_{c,1,y})^2}{2Y(\frac{\sum_{y=1}^{Y} (\gamma_2 \Delta C_{\omega}(A^*_{k,y}) + \gamma_1 \Delta C'_{\omega}(A^*_{k,y}) + \gamma_0)}{Y})^2} \tag{3.36}$$

where $V_{c,1,y}, (y = 1, 2, ..., Y)$ is the yth value of the critical voltage $V_{c,1}$ under the input data $A^*_{k,y}$. By minimizing the linear matching error ε with $\partial\varepsilon/\partial\gamma_z = 0, (z = 0, 1, 2)$, the minimal linear matching error ε_{min} and the coefficients: γ_2, γ_1, and γ_0 can be determined.

As shown in Fig. 3.8, the minimal linear matching error ε_{min} is about 1023% for the 130 nm CMOS WDDL-based AES strong PUF against machine-learning attacks. Additionally, the security of the WDDL-based AES strong PUF against machine-learning attacks can be further predicted when the CMOS technology node is scaled from 130 nm to 14 nm by using

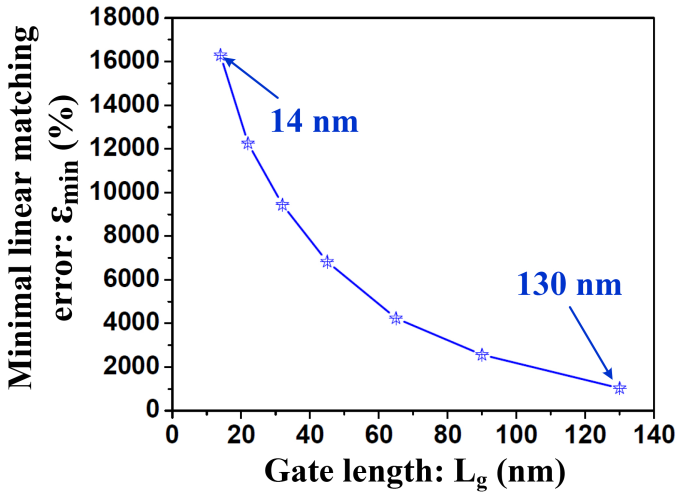


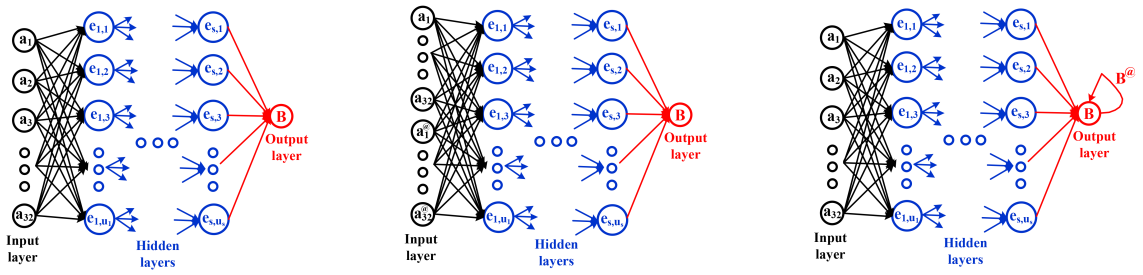
Fig. 3.8: Minimal linear matching error ε_{min} versus gate length L_g for the WDDL-based AES strong PUF against machine-learning attacks ($Y = 100,000$).

the process mismatch data from [17], as illustrated in Fig. 3.8.

3.5.2 DEEP-LEARNING ATTACKS ON THE PROPOSED PUF

Artificial neural network (ANN) attacks are a kind of popular deep-learning attacks that were chosen by prior works [58, 72, 73] to study the resilience of a PUF against machine-learning attacks. In this paper, three different ANN architectures: regular ANN, forward ANN, and backward ANN are designed for performing deep-learning attacks on the WDDL-based AES strong PUF, as shown Fig. 3.9.

For the proposed PUF, the input challenge $A = (a_1, a_2, ..., a_{32})_2$ is a 32-bit binary data and the output response B is a binary bit. Therefore, in the regular ANN as shown in Fig. 3.9(a), a_0, a_1, ..., a_{32} are set as the input layer and B is set as the output layer to model the relationship between the input challenge and the output response of the proposed PUF. However, since the WDDL-based AES strong PUF exploits the characteristic of dynamic



(a) (b) (c)

Fig. 3.9: Three different artificial neural network (ANN) architectures for performing deep-learning attacks on the WDDL-based AES strong PUF. (a) Regular ANN. (b) Forward ANN. (c) Backward ANN..

power dissipation, the output data not only depend on the current input data but also are affected by the history data [74, 75]. Thus, two more advanced ANN architectures: forward ANN (as shown in Fig. 3.9(b)) and backward ANN (as shown in Fig. 3.9(c)) are proposed to further explore the relationship between the input and the output of the PUF. For the forward ANN, the previous 32-bit input challenge $(a_1^{@}, a_2^{@}, ..., a_{32}^{@})_2$ is in conjunction with the current 32-bit input challenge $(a_1, a_2, ..., a_{32})_2$ are set as the input layer. By contrast, the last output response $B^{@}$ is reused to train the backward ANN.

In Fig. 3.9, s number of hidden layers exist in each ANN architecture and the $k_1 th$, $(k_1 = 1, 2, ..., s)$ hidden layer layer has u_{k_1} number of neurons. Assume the weight of the neuron $e_{s,k_2}, (k_2 = 1, 2, ..., u_s)$ that corresponds to the output layer B is x_{s,k_2}. Then the cost function Δf of the regular ANN and forward ANN can be denoted as

$$\Delta f = (\sum_{k_2=1}^{u_s} e_{s,k_2} x_{s,k_2} + L_s - B)^2 \tag{3.37}$$

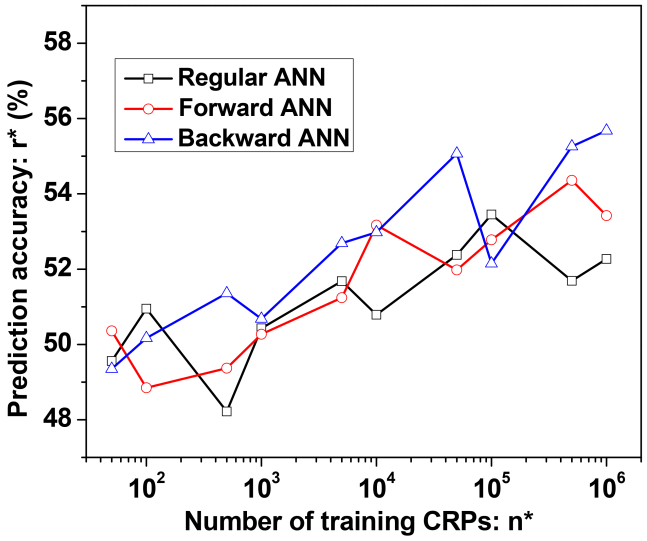


Fig. 3.10: Prediction accuracy r^* versus number of training CRPs n^* for the WDDL-based AES strong PUF under thee different deep-learning attacks ($s = 3$, $u_1 = 15$, $u_2 = 30$, and $u_3 = 20$).

where L_s is the unit bias of the sth hidden layer. Likewise, the cost function Δf^* of the backward ANN is derived as

$$\Delta f^* = (\sum_{k_2=1}^{u_s} e_{s,k_2} x_{s,k_2} + L_s + \Delta x B^{@} - B)^2 \tag{3.38}$$

where Δx is the weight of last output response $B^{@}$. By applying the backpropagation and gradient descent algorithms into the ANNs, the ANNs can be trained with a reasonable number of challenge-to-response pairs (CRPs) to predict the output response of the WDDL-based AES strong PUF under a certain input challenge.

If the ReLU function is selected for the ANNs, as shown in Fig. 3.10, even if 1 million number of CRPs are enabled for training, the prediction accuracies of the three different

deep-learning attacks on the 130 nm CMOS WDDL-based AES strong PUF are below 56%. The result demonstrates that the proposed PUF is adequately robust against the state-of-the-art machine-learning attacks.

3.6 CONCLUSION

A wave dynamic differential logic (WDDL)-based AES strong PUF is proposed as a highly reliable and secure hardware primitive for authentication. By utilizing the WDDL gates and the non-linear product math function, the WDDL-based AES strong PUF primitive achieves a nearly 50.7% inter-HD and 97.7% reliability while maintaining a low loss ratio of input power entropy ($< 25\%$) against power attacks and a huge linear matching error (1032%) against machine-learning attacks.

CHAPTER 4

A NOVEL PUF PRIMITIVE FOR GENERAL PROTECTION AGAINST NON-INVASIVE ATTACKS ON IOT DEVICES

4.1 MOTIVATION

In Chapter 2 and Chapter 3, two innovate PUF primitives are proposed: Converter-reshuffling (CoRe) PUF exploits a mature power delivery device as the fundamental architecture of PUF primitive, while WDDL gate PUF utilizes a SCA-resistant logic gate to accomplish the PUF design[1]. Both designs achieve comprehensive designs against machine learning attacks and side-channel analysis attacks. Meanwhile, by means of adopting existing hardware architecture to design new PUFs, both designs successfully integrated two hardware security primitives and are of resilience to non-invasive attacks. Nonetheless, based on investigations to side-channel attacks, attackers can perform SCA attacks with either input power or output power [77]. In this case, attackers may perform SCA analysis attack according to power traces from both ends and examine the correlation coefficients to select a more vulnerable power signal. This requires us to insert SCA-resistant modules on both ends of vulnerable devices. Simultaneously, in previous designs, only linear regression (LR) attacks and simple machine learning models are examined. According to Fig. 2.7(a), the insertion of hardware devices only induces limited promotion of non-linearity. Consequently, a novel PUF architecture against non-invasive attacks is proposed in this chapter, aiming to impede the non-invasive attacks essentially.

[1]The content of this Chapter partially has been published in [76].

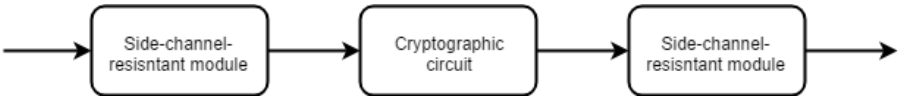
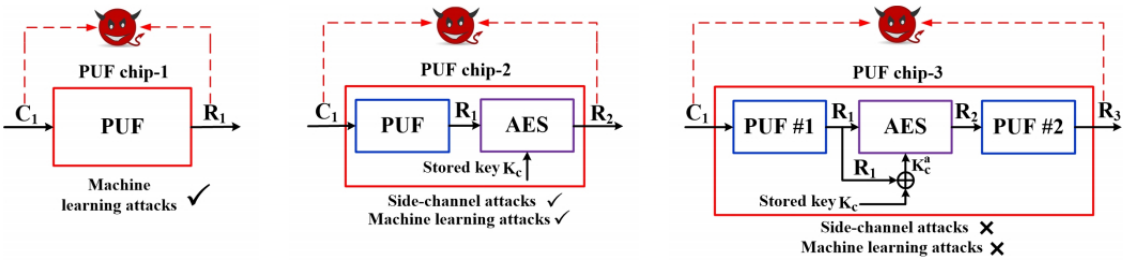


Fig. 4.1: Conceptual model of proposed comprehensive countermeasure to ML attacks and SCA attacks. Both ends of the cryptographic circuit is protected by a ML model. The uncertainty is retained by the keys stored/inserted on the cryptographic circuit.

The conceptual prototype can be found in Fig. 4.1. The conception is based on an assumption that the cryptographic circuit, like AES circuit, is relative non-modelized. With the uncertainty induced by random keys, the cryptographic circuit cannot be modeled by simple machine learning attacks. However, since the operation of AES circuit is proved to be vulnerable side-channel attacks [78], two side-channel-resistant models are thus implemented on both ends of the AES model. Since the uncertainty is introduced by keys of AES circuit, the frontend signal and the backend signal of the entire architecture should retain uncertainty to attackers, which finally result in failures in both machine learning attacks and SCA analysis attacks. In this chapter, we would focus on the prototype design and think of possible vulnerable attack models. At the same time, we would evaluate the robustness to higher level attack models comparing to LR attacks and simple ML attacks.

4.2 WORKING PRINCIPLE OF THE PROPOSED PUF

Fig. 4.2(a) shows a diagram of a conventional PUF chip under machine learning attacks. Since both the input challenge C_1 and the output response R_1 are exposed to the adversary directly, the PUF chip-1 can be cracked by machine learning attacks through training a reasonable number of CRPs. If we explore the design by connecting a conventional PUF circuit in series with an AES circuit to build a new PUF: hybrid PUF, as shown in Fig. 4.2(b),



(a) (b) (c)

Fig. 4.2: Three different PUF chips under machine learning or side-channel attacks. (a) Conventional PUF (PUF chip-1). (b) Hybrid PUF (PUF chip-2). (c) Key-updating (KU) AES-embedded PUF (PUF chip-3).

the adversary may not be able to unriddle the secret information of the conventional PUF via machine learning attacks directly. However, the hybrid PUF in Fig. 4.2(b) is not sufficiently secure. The primary reason is that the output data R_2 of the AES is exposed to the adversary, therefore, the secret key K_c of the AES may be leaked to the adversary by analyzing the correlation between the output data R_2 and a certain physical leakage of the PUF chip-2 if a side-channel attack is executed. Once the secret key K_c of the AES is leaked, the output response R_1 of the conventional PUF will also be disclosed. As a result, the conventional PUF in Fig. 4.2(b) can be uncovered by training the (C_1, R_1) pairs with machine learning attacks ultimately.

So as to eliminate the threats from both side-channel and machine learning attacks, a key-updating (KU) AES embedded PUF is proposed as shown in Fig. 4.2(c). The novel and innovative PUF architecture is secure against machine learning attacks, without increasing the degree of non-linearity between the CRPs. As indicated in Fig. 4.2(c), an AES circuit is embedded between two conventional PUF circuits and the output response R_1 of the PUF #1 circuit is encrypted by the AES circuit to provide the input challenge R_2 to the PUF

#2 circuit. Since the output response R_1 of the PUF #1 and the input challenge R_2 of the PUF #2 are concealed, the adversary is incapable of performing machine learning attacks on either of the two PUF circuits.

Another novel idea is proposed to eliminate the threat of side-channel attacks, which does not rely on the existing countermeasures. It is proposed to add a real-time key-updating function to the architecture that combines the output response R_1 of the PUF #1 with the stored secret key K_c of the AES to create the actual key K_c^a used by the AES circuit ($K_c^a = R_1 \oplus K_c$). This is illustrated in Fig. 4.2(c). Since the input data R_1 and the output data R_2 of the AES are unknown to the adversary and the actual secret key K_c^a of the AES is updating in real-time, the adversary is unable to execute side-channel attacks to reveal the stored secret key K_c.

4.3 ROBUSTNESS AGAINST MACHINE LEARNING ATTACKS

For an m-bit KU AES-embedded PUF as shown in Fig. 4.2(c), the input data/output data of the PUF 1, the AES, and the PUF 2, respectively, are C_1/R_1, R_1/R_2, and R_2/R_3. If the input challenge C_1 is set as $C_1 = (c_{1,1}, c_{1,2}, \cdots, c_{1,m})_2$, the relationship between the input challenge C_1 and the output response R_1 of the PUF 1 can be preliminarily modeled

as

$$R_1 = C_1 \times E_1 = (c_{1,1}, c_{1,2}, \cdots, c_{1,m}) \times \begin{pmatrix} e_{1,1} & \cdots & e_{1,m} \\ e_{2,1} & \cdots & e_{2,m} \\ \vdots & \ddots & \vdots \\ e_{m,1} & \cdots & e_{m,m} \end{pmatrix} = \left(\sum_{i=1}^{m} c_1, ie_{i,1}, \cdots, \sum_{i=1}^{m} c_1, ie_{i,1} \right) \tag{4.1}$$

where E_1 is the $m \times m$ matrix which represents the operation induced by the PUF 1 in Fig. 4.2(c). $e_{i,j}$, $(i, j = 1, 2, \cdots, m)$ is the element of the matrix E_1. Since the output data R_1 of the PUF 1 should be a binary data, the output data $R_1 = (\sum_{i=1}^{m} c_1, ie_{i,1}, \cdots, \sum_{i=1}^{m} c_1, ie_{i,1})$ as shown in (4.1) needs to be normalized by a step function $u(x, \Delta x)$ as shown below

$$u(x, \Delta x) = \begin{cases} 1 & , x \geq \Delta x \\ 0 & , x < \Delta x \end{cases} \tag{4.2}$$

where Δx is the critical point of the step function $u(x, \Delta x)$. As a result, the output data R_1 of the PUF 1 under the normalization of the step function $u(x, \Delta x)$ can be precisely determined as

$$\begin{aligned} R_1 &= u(C_1 \times E_1, \Delta x_1) \\ &= \left(u\left(\sum_{i=1}^{m} c_1, ie_{i,1}, \Delta x_1 \right), \cdots, u\left(\sum_{i=1}^{m} c_1, ie_{i,m}, \Delta x_1 \right) \right) \end{aligned} \tag{4.3}$$

where Δx_1 is the critical point associated with the PUF 1.

Similarly, the accurate output data R_2 of the AES in Fig. 4.2(c) can also be derived as

$$R_2 = u(u(C_1 \times E_1, \Delta x_1) \times A, \Delta x_0) \tag{4.4}$$

where Δx_0 is the critical point related with the AES and A is the $m \times m$ matrix that denotes the math operation processed by the AES circuit as shown below ($ai, j(C_1)$ is the element in matrix A)

$$A = \begin{pmatrix} a_{1,1}(C_1) & \cdots & a_{1,m}(C_1) \\ a_{2,1}(C_1) & \cdots & a_{1,m}(C_1) \\ \vdots & \ddots & \vdots \\ a_{m,1}(C_1) & \cdots & a_{1,m}(C_1) \end{pmatrix}. \tag{4.5}$$

Kindly note that the actual secret key K_c^a of the AES in Fig. 4.2(c) is updated by the internal confidential data R_1 in realtime, thus the element $ai, j(C1)$ varies under a different input challenge C_1.

Once the output data R_2 of the AES is obtained, the m-bit output response R_3 of the KU AES-embedded PUF in Fig. 4.2(c) under the input challenge C_1 is expressed as

$$\begin{aligned} R_3 &= u(R_2 \times E_2, \Delta x_2) \\ &= u\left(u\left(u\left(C_1 \times E_1, \Delta x_1\right) \times A, \Delta x_0\right) \times E_2, \Delta x_2\right) \end{aligned} \tag{4.6}$$

where Δx_2 is the critical point associated with the PUF 2 and E_2 is the corresponding $m \times m$

TABLE 4.1: A CNN structure for modeling PUF primitives.

	Input L	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7	Layer 8	Layer 9
Type		CNN	Max pooling	Dropout	Flatten	Dense	Dropout	Dense	Dropout	Dense
Parameters		K = 3*3		0.25	--		0.75		0.75	
Input		8*16*1	6*14*64	3*7*64	3*7*64	1344	1024	1024	512	512
Output		6*14*64	3*7*64	3*7*64	1344	1024	1024	512	512	2
Activation function		ReLU				ReLU		ReLU		Softmax
Number of nodes	8*16*1	64	64	64	1344	1024	1024	512	512	2

matrix induced by the PUF 2 that is written as ($e^*_{i,j}$ is the element in matrix E_2)

$$E_2 = \begin{pmatrix} e^*_{1,1} & \cdots e^*_{1,m} \\ e^*_{2,1} & \cdots e^*_{2,m} \\ \vdots & \ddots \quad \vdots \\ e^*_{m,1} & \cdots e^*_{m,m} \end{pmatrix}. \tag{4.7}$$

4.3.1 CONVOLUTIONAL NEURAL NETWORK (CNN) ATTACKS ON PUF PRIMITIVES

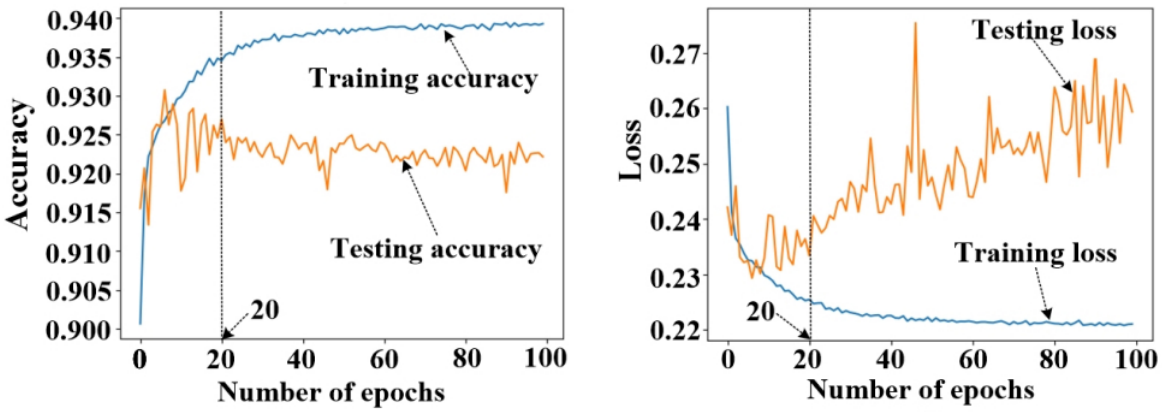
Convolutional neural networks (CNNs) are a kind of advanced machine learning algorithms that can be explored to reveal the secret information. If a CNN attack is performed on a 128-bit regular PUF: arbiter PUF, the detailed CNN architecture for modeling the arbiter PUF is illustrated in Table 4.1. The 128-bit input challenge $C_1 = (c_{1,1}, c_{1,2}, \cdots, c_{1,128})_2$ of the arbiter PUF is transformed into a $8 \times 16 \times 1$ matrix to establish the input training data for the CNNs. Moreover, the most significant bit (MSB) of the output response of the arbiter PUF is selected as the output training data for the CNN structure in Table 4.1.

In layer 1 (convolutional layer) of the CNNs, the $8 \times 16 \times 1$ input matrix is changed

into a $6 \times 14 \times 64$ intermediate matrix, as shown in Table 4.1. The size of the 64 filters (K) and activation function used in layer 1, respectively, are 3×3 and ReLU. In layer 2, a max pooling operation is executed on the obtained $6 \times 14 \times 64$ matrix to generate the $3 \times 7 \times 64$ matrix. The main intention of executing max pooling is reducing the matrix size and maintaining the critical features.

As shown in Table 4.1, in layer 3 and layer 4, a dropout operation and a flatten operation are performed. Then the $1 \times 1 \times 1344$ array is acquired when the flatten operation is finished. The dropout probability that is used in layer 3 is set as 0.25. Furthermore, from layer 5 to layer 9, dense operation and dropout operation are executed in the CNNs alternately. The sizes of the output arrays of layer 5, layer 6, layer 7, and layer 8 are achieved as $1 \times 1 \times 1024$, $1 \times 1 \times 1024$, $1 \times 1 \times 512$, and $1 \times 1 \times 512$, respectively. The corresponding dropout probability in layer 6 and layer 8 is selected as 0.75. Eventually, in the output of layer 9, the $1 \times 1 \times 2$ array that is used for classifying the MSB of the output response as "0" or "1" is created. Kindly note that the activation functions selected in layer 5, layer 7, and layer 9, respectively, are ReLU, ReLU, and Softmax. The optimizer and loss function of the CNN training are chosen with Adam and categorical cross-entropy, respectively. Additionally, the batch size of the CNN training is optimized with 50.

A 128-bit KU AES-embedded PUF and a 128-bit arbiter PUF are simulated in Cadence with the 130 nm IBM CMOS technology kit, respectively. Kindly note that the simulated KU AES-embedded PUF consists of two 128-bit arbiter PUF circuits and one 128-bit AES circuit. In addition, the corresponding CRPs of these two PUFs are also extracted from Cadence simulation. As shown in Fig. 4.3, when the CNN attack is executed on the 128-bit



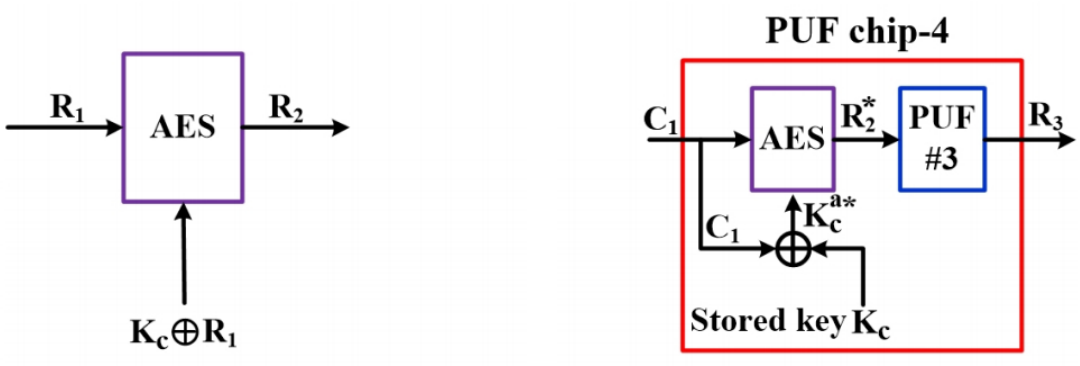
(a) (b)

Fig. 4.3: Training result of the CNN structure in Table 4.1 for modeling the MSB of the 128-bit arbiter PUF (100,000 number of CRPs are enabled for training). (a) Accuracy versus number of epochs. (b) Loss versus number of epochs.

TABLE 4.2: Training Results of the CNN structure for modeling the MSB of the 128-bit KU AES-embedded PUF (number of epochs is 20).

CRPs	Training accuracy	Testing accuracy	Training loss	Testing loss
100,000	0.512	0.509	0.973	0.985
500,000	0.527	0.521	0.954	0.962
1,000,000	0.516	0.518	0.966	0.958

arbiter PUF, after training about 1×105 data, the training (testing) accuracy is obtained as 0.934 (0.927) and the corresponding training (testing) loss is 0.225 (0.234) when the number of epochs is set as 20. By contrast, even if 1 million data are enabled for training, the training/testing accuracy of the CNNs for modeling the 128-bit KU AES-embedded PUF is still around 0.5, as shown in Table 4.2. Hence, the KU AES-embedded PUF we propose exhibits a good robustness against the regular CNN attacks.



(a) (b)

Fig. 4.4: (a) AES in the KU AES-embedded PUF. (b) Equivalent PUF architecture for the KU AES-embedded PUF.

4.3.2 NEW CONVOLUTIONAL NEURAL NETWORK (CNN) ATTACKS ON THE PROPOSED PUF PRIMITIVE

In Section 4.3.1, since the stored key K_c in Fig. 4.2(c) is assumed to be unknown to the adversary, the conventional CNN attacks attempt to disclose the confidential information of the KU AES-embedded PUF by regarding the PUF as a black box. However, if the secret key K_c is leaked, a novel CNN attack may be tailored to model the confidential information of the KU AES-embedded PUF.

When the secret key K_c is leaked, the exact relationship between the input data R_1 and the output data R_2 of the AES in Fig. 4.4(a) will be revealed. In other words, the secret $m \times m$ matrix A associated with the AES is always exposed to the adversary. As a result, only two PUF matrices E_1 and E_2 in (4.6) are unknown to the adversary. To simplify the CNN attack, if the matrix E1 is selected with an m $\times$ m identity matrix I and the critical

TABLE 4.3: Training results of the new CNN attack for modeling the MSB of the 128-bit KU AES-embedded PUF (number of epochs is 20).

CRPs	Training accuracy	Testing accuracy	Training loss	Testing loss
100,000	0.903	0.895	0.261	0.289
500,000	0.911	0.918	0.247	0.232
1,000,000	0.927	0.922	0.195	0.217

point Δx_1 is set as 0.5, (4.6) becomes

$$R_3 = u(u(u(C_1 \times I, 0.5) \times A, \Delta x_0) \times E_{3,3}) \tag{4.8}$$

where E_3 and Δx_3, respectively, are the $m \times m$ matrix and critical point induced by an equivalent PUF circuit. Since $C_1 \times I \times A \times E_3$ is equal to $C_1 \times A \times E_3$, a new PUF architecture as shown in Fig. 4.4(b) can be devised to emulate the KU AES-embedded PUF if the secret key K_c is disclosed.

In Fig. 4.4(b), PUF 3 is the equivalent PUF circuit that is related with the matrix E3 in (4.8). Furthermore, the output data R^2 of the AES in Fig.4.4(a) can be unriddled since the input data C1 and secret key Kc of the AES are open for the adversary. Accordingly, a CNN attack can be performed on the equivalent PUF circuit: PUF 3 because its CRPs: (R_2^*, R_3) are available for training. Once the PUF 3 is cracked, the adversary is able to predict output response R_3 under any input challenge C_1.

Table4.3 shows the results of the new CNN attack on the 128-bit KU AES-embedded PUF if the adversary knows the stored key K_c. The corresponding training/testing accuracy can be over 0.9. Consequently, the most significant security concern for the KU AES-embedded PUF is preventing the secret key K_c from being leaked to the adversary.

4.4 RESILIENCE AGAINST SIDE-CHANNEL ATTACKS

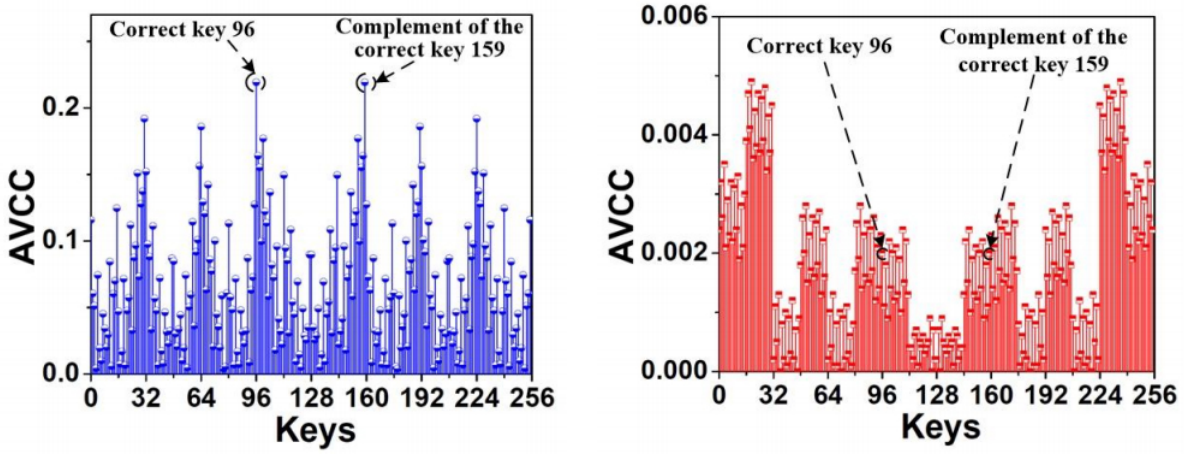
For the KU AES-embedded PUF we propose, the input data R_1 and output data R_2 of the AES in Fig. 4.2(c) are unknown to the adversary. As a result, to implement side-channel attacks on the KU AES-embedded PUF to reveal the stored key K_c, the adversary can only analyze the correlation between the input challenge C_1/output response R_3 in Fig. 4.2(c) and the physical leakages of the PUF chip.

Power attacks [50, 79, 80] are a kind of side-channel attacks that are widely used by the adversary to disclose the secret key of a cryptographic circuit through monitoring the correlation between the processed data and the power dissipation of the cryptographic circuit. For the m-bit KU AES-embedded PUF in Fig. 4.2(c), the input data R_1 of the AES is $\left(u\left(\sum_{i=1}^{m} c_{1,i}e_{i,1}, \Delta x_1\right), \cdots, u\left(\sum_{i=1}^{m} c_{1,i}e_{i,m}, \Delta x_1\right)\right)_2$ as shown in (4.3). Suppose the m-bit stored key K_c of the KU AES-embedded PUF is $K_c = (k_{c,1}, k_{c,2}, \cdots, k_{c,m})_2$, the actual key K_c^a of the AES in Fig. 4.2(c) becomes

$$\begin{aligned} K_c^a &= R_1 \oplus K_c \\ &= \left(u\left(\sum_{i=1}^{m} c_{1,i}e_{i,1}, \Delta x_1\right) \oplus k_{c,1}, \cdots, u\left(\sum_{i=1}^{m} c_{1,i}e_{i,m}, \Delta x_1\right) \oplus k_{c,m}\right)_2 \end{aligned} \tag{4.9}$$

As a result, the real secret key K_c^a will be updated in real-time if different input challenge values: $(c_{1,1}, c_{1,2}, \cdots, c_{1,m})_2$ are enabled.

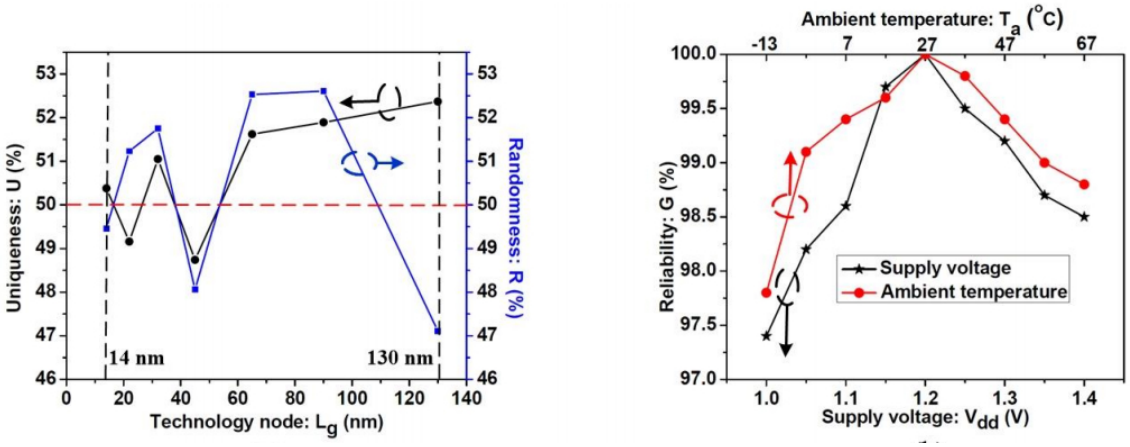
When a power attack is performed on the KU AES-embedded PUF, the adversary may combine the input challenge with the hypothesized keys to predict the power dissipation of the PUF at first. Then the correlation analysis will be executed between the predicted



(a) (b)

Fig. 4.5: Simulations of power attacks (hamming-weight (HW) model is used). (a) Absolute value of correlation coefficient (AVCC) versus possible keys for leaking an 8-bit sub-key of the 128-bit unprotected AES cryptographic circuit after inputting 1,000 number of data. (b) AVCC versus possible keys for leaking an 8-bit sub-key of the 128-bit KU AES-embedded PUF after inputting 1 million number of data.

power and the measured power to estimate the secret key. Fig. 4.5 shows the results of simulated power attacks for (a) a 128-bit unprotected AES cryptographic circuit and (b) the 128-bit KU AES-embedded PUF. As shown in Fig. 4.5(a), the 8-bit secret sub-key 96 of the unprotected AES circuit is disclosed after inputting 1,000 plaintexts of data. However, for the AES-embedded PUF, the secret sub-key 96 is masked from being leaked to the adversary even if 1 million plaintexts are enabled, as shown in Fig. 4.5(b). In addition, the absolute value of correlation coefficient (AVCC) of the correct key 96 in Fig. 4.5(b) is two orders of magnitude lower than the AVCC of the correct key 96 in Fig. 4.5(a). The primary reason is that the actual secret key in the embedded PUF is updating in real-time which greatly weakens the correlation between the processed data and the power dissipation against power attacks.



(a) (b)

Fig. 4.6: Performance evaluation for the 128-bit KU AES-embedded PUF. (a) Uniqueness U and randomness R versus technology node L_g. (b) Reliability G versus supply voltage V_{dd} and ambient temperature T_a ($L_g = 130$ nm).

4.5 PERFORMANCE EVALUATION

Commonly, uniqueness, randomness, and reliability are the three most significant parameters for assessing the performance of a designed PUF [4, 47]. To evaluate the performance of the proposed PUF, A 128-bit KU AES-embedded PUF is designed and simulated in Cadence software with the 130 nm CMOS technology kit. Moreover, Monte Carlo simulations are executed on the designed PUF in Cadence to emulate the random fabrication process. As shown in Fig. 4.6(a), the uniqueness U is improved from 52.4% to 50.4% if the CMOS technology node Lg is scaled from 130 nm to 14 nm; while the randomness R improves from 47.1% to 49.5%. In addition, Fig. 4.6(b) shows the worst reliability of the embedded PUF is about 97.4% when the supply voltage is 1.0 V. The simulation results manifest the proposed PUF has excellent uniqueness, randomness, and reliability.

4.6 CONCLUSION

A novel PUF-AES-PUF architecture in conjunction with a key-updating technique is utilized to design a state-of-the-art PUF primitive that is able to resist non-invasive attacks. The proposed PUF not only has excellent uniqueness (52.4%), randomness (47.1%), and reliability (97.4%) but also maintains a high security level ($>$ 1 million data) against both side-channel and machine learning attacks.

CHAPTER 5

HARDWARE TROJAN-BASED MALICIOUS ATTACKS ON PHYSICAL UNCLONABLE FUNCTION SENSORS

5.1 MOTIVATION

In previous sections, three PUF primitives are proposed. Aiming at mainstream non-invasive attack models, all three models may resolve security problems in their respective domains[1]. According to different application scenarios, we believe those proposed work can adopt most security demands in the protection of physical confidential signals. However, the study of physical unclonable function primitives is only one aspect of the whole domain of hardware security. However, it is always said, "The easiest way to capture a fortress is from within." Is there any way that attackers can bypass all external protection mechanisms and steal confidential messages from within? This talks about another pivotal subject in hardware security, hardware Trojans (HTs).

After considering all potential malicious attacks, a hardware system should be mounted with countermeasures to SCA attacks, ML attacks, and HT attacks, as is shown in Fig. 5.1. In this architecture, a key hardware element should put in the middle of the whole architecture, while its frontend and backend are SCA-resistant models which prevent steals and analyses of physical signal leakages. For the sake of reducing vulnerabilities to modeling attacks, some uncertainty should be injected to the system. In addition, on the basis of

[1]The content of this Chapter partially has been published in [81].

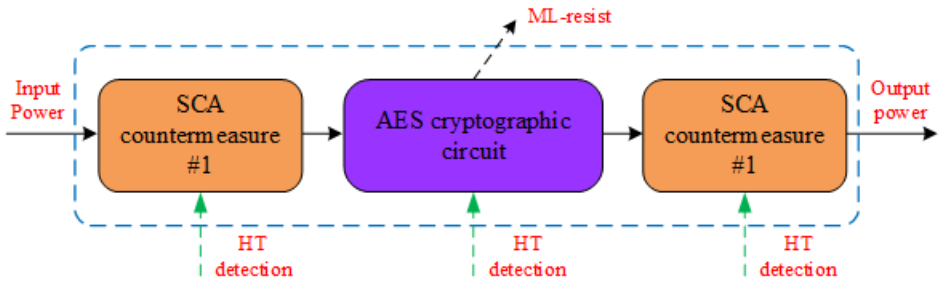
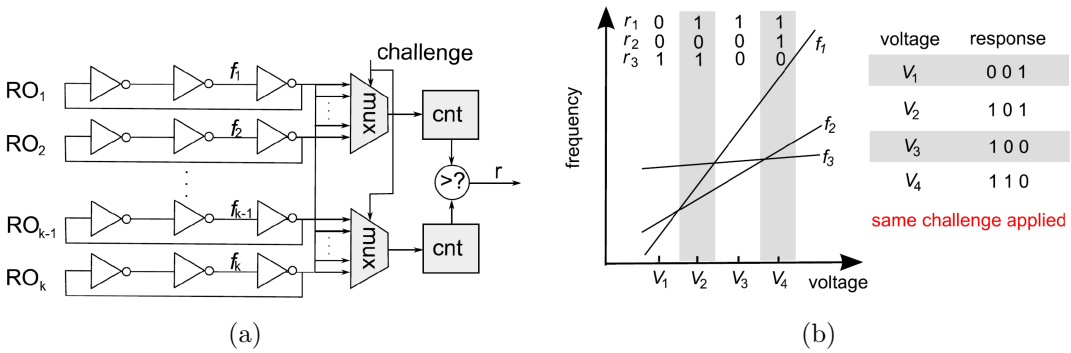


Fig. 5.1: A conceptual sketch of a hardware system that is well protected by countermeasures to ML attacks, SCA attacks, and HT attacks.

the layout and analysis approach, the detection of hardware Trojans can be executed after fabrication or in real-time. Since hardware Trojans are always implemented during fabrications and can be implemented in any chips. In this chapter, we make a more general assumption that hardware Trojans are implemented on a PUF primitive. On the one hand, the design of hardware Trojans need to consider well of the architecture design. If the Trojan is designed for a protection module, attackers do not bother to adjust their design for a new architecture. And this will make their attacks more general and more effective. On the other hand, the implementation of security module is enough for stealing data. Considering the overhead of security modules, hardware devices may not implement redundant countermeasures. On account of the assumption, a hardware Trojan design will be proposed in this chapter, which aims at stealing information processed by a PUF primitive. In the meantime, a detection approach will be presented. The detection is a golden-chip free method which utilize mathematical analysis as its detection tool. To some extent, golden-chip free approach is more practical to conventional method with golden chips. We hope this method may be generalized to adapt to more Trojan detection situations.



(a) (b)

Fig. 5.2: (a) Architecture of a ROPUF [4]. (b) Oscillating frequencies versus supply voltage for a ROPUF under the same input challenge [4]

5.2 REVIEW OF RING OSCILLATOR PUF (ROPUF) SENSOR

5.2.1 VULNERABILITIES IN CONVENTIONAL ROPUF DESIGNS

In a ROPUF, as shown in Fig. 5.2(a), k number of ring oscillators are utilized to output the cipher responses. Despite the k number of ring oscillators are identically designed, the oscillating frequency of each ring oscillator is different under the random fabrication process. As a result, if two multiplexers are used to select two different ring oscillator loops, the unpredictable mismatch of the oscillating frequency of the two selected ring oscillator loops can be converted into the digital cipher data. For example, in Fig. 5.2(a), assume the ring oscillator loops RO_1 and RO_2 are selected by the top multiplexer (mux) and the bottom multiplexer under a certain input challenge, respectively. If the oscillating frequency of RO_1 is higher or equal to the oscillating frequency of RO_2, the output response $r = 1$. Otherwise, the output response $r = 0$.

As shown in Fig. 5.2(b), if the same challenge is applied into the ROPUF, strong linear relationships exist between the supply voltage and the oscillating frequencies. However,

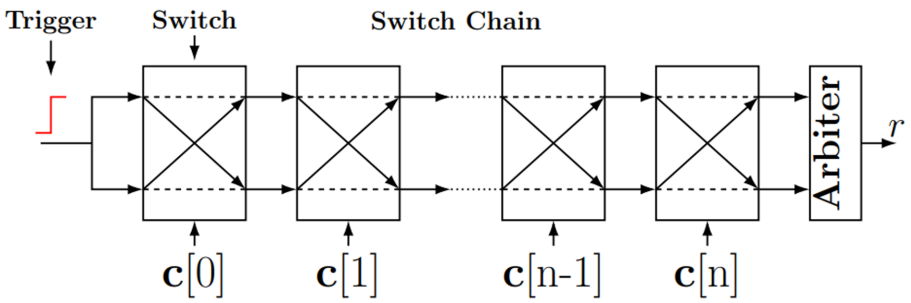


Fig. 5.3: Classic Arbiter PUF using path-swapping switches [5].

due to the affects of the random fabrication process, the slope of the frequency-voltage curve of each ring oscillator in the ROPUF is different. In other words, different supply voltages can achieve different frequency mismatches. Consequently, a one-to-one relationship is established between the supply voltage and the output cipher data under the same input challenge, as shown in Fig. 5.2(b). Moreover, since all the physical quantities like temperature [82], pressure [83], and light luminosity [84] can be sensed and converted into voltage signals without much effort, the ROPUF sensor is able to encrypt the sensed physical quantity against the regular malicious attacks.

5.2.2 MACHINE LEARNING ATTACK ON A PUF PRIMITIVE

In this proposed work, we propose to design a hardware Trojan inclusion to extract critical confidential information in a PUF primitive. Therefore, a pivotal problem in this work is after extracting system information with a Trojan, how to approximate the generation of challenge response pairs (CRPs) with a further training in neural networks.

Some existing works made efforts in attack PUF primitives via convolutional neural networks (CNN). In a paper that is proposed in 2016 [85], authors managed to use machine

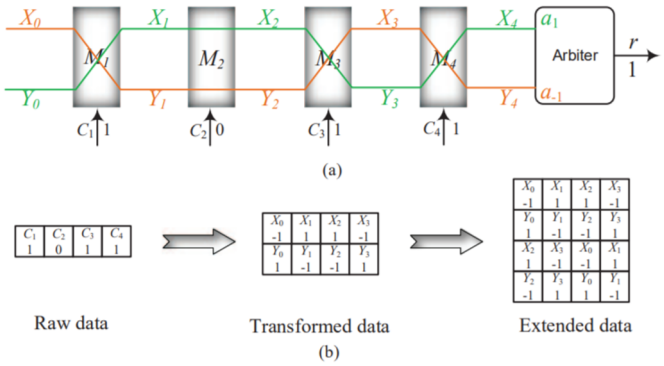


Fig. 5.4: CRPs data transformation and extension [6].

learning algorithm to crack an arbiter PUF using variant machine learning algorithm. The basic architecture of arbiter PUF is proposed in Fig. 5.3. In the proposed architecture, the input signal is $c[0]$ to $c[n]$, which use switch arrays to select signal paths for two input signals. Kindly note that the input signal is split into two paths at the origin, while each switch in the chain will select one signal to pass the top channel and the other signal will pass the lower channel. Due to the fabrication variations among all switches, the overall passing time for two split signals will exhibit a tiny mismatch at the end of the chain. As a result, the arbiter in the end is then used to sense the arrival of trigger signals in two paths and decide the response output at the port r. In such case, for a particular input node $c[i]$, its followed challenge signal $c[i+1]$ will inherit the accumulated time delay in previous challenge node $c[i]$. As a result, adjacent input challenge nodes are of sequential correlation induced by time delay. On the other hand, since the output response r is mechanically generated by the challenge input. The output response will be then of sequential correlation in adjacent output nodes as well. Therefore, in theory, using CNN as attack mechanism to a PUF primitive is rational.

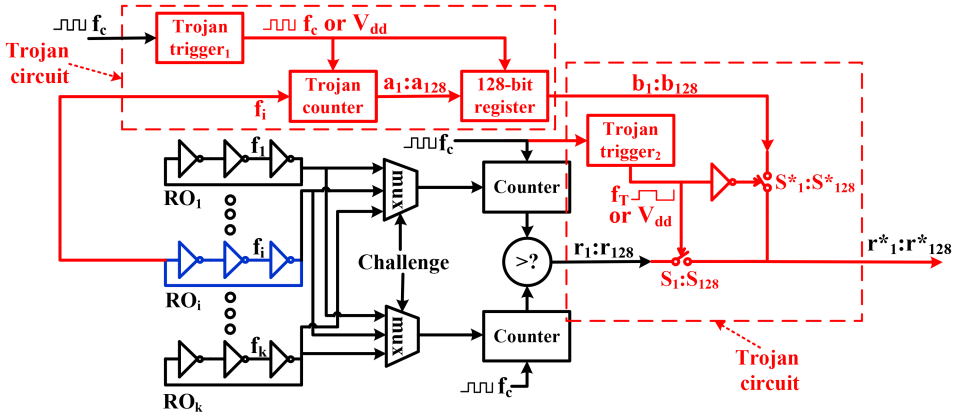


Fig. 5.5: Basic architecture of a 128-bit Trojan-infected ROPUF

In addition to [85], some other works also make efforts in cracking PUF with CNN networks. [6] proposed a CRP transformation and expansion technique for an arbiter PUF. Their extension mechanism is as shown in Fig. 5.4. Their method is to decompose delay signature between two adjacent input nodes, while in a further extension, by shifting bits of X_i and Y_i, the delay signature will be then formatted as two-dimensional image. By using the extended data as input and the response as output, authors apply CNN as attack models for the proposed arbiter PUF and assert this data modeling method is more effective. In this work, we would use a similar CNN architecture to perform modeling attack on the ROPUF.

5.3 DESIGN OF TROJAN-INFECTED ROPUF

The vulnerability of a ROPUF sensor is the high linear relationship between the supply voltage and the oscillating frequency of a single ring oscillator loop. If the critical oscillating frequency in the ROPUF is disclosed to the adversary through utilizing a Trojan circuit, the adversary may estimate the variations of the supply voltage and the physical quantity

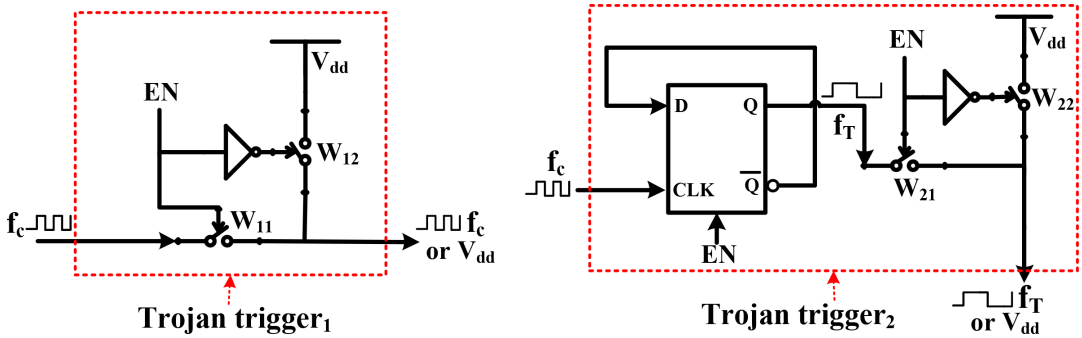


Fig. 5.6: (a) Architecture of the Trojan trigger$_1$. (b) Architecture of the Trojan trigger$_2$

by monitoring the variations of the leaked oscillating frequency.

The basic architecture of a 128-bit Trojan-infected ROPUF is shown in Fig. 5.5. There are two Trojan triggers: Trojan trigger$_1$ and Trojan trigger$_2$ in the embedded Trojan circuit. When the Trojan trigger$_1$ is activated, the enable signal EN in Fig. 5.6(a) will activate the switch W_{11} and deactivate the switch W_{12} to output the original clock signal f_c to activate the Trojan counter and 128-bit register in Fig. 5.5. As a result, as shown in Fig. 5.5, the oscillating frequency f_i of the ith, $(i = 1, 2, ..., k)$ ring oscillator loop RO$_i$ in the ROPUF is extracted by the inserted Trojan counter. The output of the Trojan counter is a 128-bit binary data $(a_1, a_2, ..., a_{128})_2$ or $a_1 : a_{128}$ that includes the information of the oscillating frequency f_i. Subsequently, the 128-bit binary data $a_1 : a_{128}$ is fed into the 128-bit register to generate the 128-bit critical output data $b_1 : b_{128}$, as shown in Fig. 5.5. However, if the Trojan trigger$_1$ is inactive, EN in Fig. 5.6(a) turns off the switch W_{11} and turns on the switch W_{12} to generate the high voltage V_{dd} to deactivate the sequential circuits: Trojan counter and 128-bit register in Fig. 5.5. In such a case, the dynamic power dissipation of the embedded Trojan circuit will be significantly reduced to evade the regular Trojan detection.

Fig. 5.6(b) shows the architecture of the Trojan trigger$_2$.The Trojan trigger$_2$ consists of a D flip-flop, an inverter, and two switches. The primary role of the D flip-flop is achieving the frequency division. When the Trojan trigger$_2$ is activated by the adversary, the enable signal EN in Fig. 5.6(b) becomes 1, thus the D flip-flop will output a new clock signal f_T whose frequency is a half of the frequency of the original clock signal f_c. Moreover, since the EN signal activates the switch W_{21} and deactivates the switch W_{22} in Fig. 5.6(b), the Trojan trigger$_2$ outputs the clock signal f_T. Hence, the clock signal f_T turns on/off the switches $S_1 : S_{128}$ and the switches $S_1^* : S_{128}^*$ in Fig. 5.5 alternately. Ultimately, in Fig. 5.5, the output response $r_1^* : r_{128}^*$ will become $r_1 : r_{128}$, $b_1 : b_{128}$, $r_1 : r_{128}$, $b_1 : b_{128}$, ... once the Trojan trigger$_2$ is activated. This indicates the information of the oscillating frequency f_i can be disclosed covertly in the output response of the Trojan-infected ROPUF.

However, if the Trojan trigger$_2$ is inactive, the EN signal turns off the switch W_{21} and turns on the switch W_{22} in Fig. 5.6(b) to output the high voltage V_{dd}. Under such a condition, in Fig. 5.5, the switches $S_1 : S_{128}$ will be turned on while the switches $S_1^* : S_{128}^*$ will be turned off. Accordingly, the 128-bit output response $r_1^* : r_{128}^*$ of the Trojan-infected ROPUF is equal to the 128-bit output response $r_1 : r_{128}$ of the Trojan-free ROPUF. This means the Trojan-infected ROPUF behaves as a regular ROPUF when the Trojan trigger$_1$ and Trojan trigger$_2$ are inactive.

5.4 RETRIEVAL OF CONFIDENTIAL DATA

5.4.1 DISCLOSURE OF CONFIDENTIAL DATA WITH HARDWARE TROJAN ATTACKS

To successfully uncover the secret information of a 128-bit Trojan-infected ROPUF sensor, the adversary needs to perform regular wireless attacks to obtain the 128-bit input challenge $C = c_1 : c_{128}$ and output response $R^* = r_1^* : r_{128}^*$ of the Trojan-infected ROPUF at first. Next, the adversary may activate the Trojan trigger$_1$ and Trojan trigger$_2$ in Fig. 5.5 to reveal the information of the critical oscillating frequency f_i via wireless communication.

When the Trojan trigger$_1$ in Fig. 5.5 is activated, the Trojan trigger$_1$ will generate the original clock signal f_c to drive the sequential circuits: Trojan counter and 128-bit register. Hence, the analog oscillating frequency f_i is converted into a 128-bit digital data $a_1 : a_{128}$ under the assistance of the Trojan counter. Furthermore, the role of the 128-bit register is delaying the signal $a_1 : a_{128}$ with a clock period. This means the output data $b_1 : b_{128}$ of the 128-bit register is equal to the previous value of $a_1 : a_{128}$.

In Fig. 5.5, if the Trojan trigger$_2$ is activated by the adversary, it will output a new clock signal f_T to control the activation patterns of the switches $S_1 : S_{128}$ and $S_1^* : S_{128}^*$. Kindly note that the frequency ratio between the new clock signal f_T and the original clock frequency f_c is designed as 1:2. Under such a condition, the data sequence that is sent out by the Trojan-infected ROPUF in Fig. 5.5 is $r_1(t) : r_{128}(t)$, $a_1(t) : a_{128}(t)$, $r_1(t+T_c) : r_{128}(t+T_c)$, $a_1(t+T_c) : a_{128}(t+T_c)$, $r_1(t+2T_c) : r_{128}(t+2T_c)$, $a_1(t+2T_c) : a_{128}(t+2T_c), \cdots$ where t is the timing and T_c is the period of the clock signal f_c. Therefore, the information of the critical oscillating frequency f_i can be extracted in the output response of the Trojan-infected

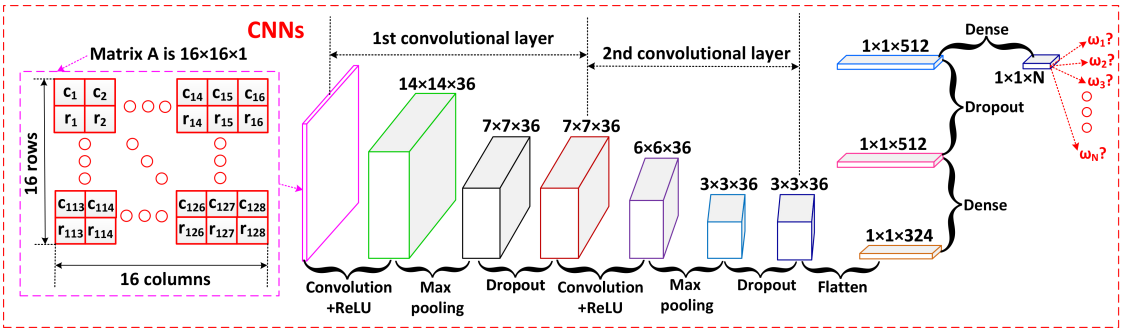


Fig. 5.7: Detailed structure of the CNNs with two convolutional layers for cracking the 128-bit Trojan-infected ROPUF

ROPUF once the Trojan trigger$_1$ and Trojan trigger$_2$ are activated.

5.4.2 DISCLOSURE OF CONFIDENTIAL DATA WITH MACHINE LEARNING ATTACKS

As introduced in Section 5.4.1, under the assistance of the hardware Trojan attacks, the adversary is capable of acquiring the 128-bit input challenge $c_1 : c_{128}$, actual output response $r_1 : r_{128}$, and oscillating frequency information $a_1 : a_{128}$ of the Trojan-infected ROPUF by activating the embedded Trojan circuit. Next, the adversary may use machine learning techniques to model the Trojan-infected ROPUF with the leaked data. Once Trojan-infected ROPUF is cracked by machine learning techniques, the adversary can deactivate the embedded Trojan circuit and is able to predict the variations of the sensed data at any time.

To study machine learning attacks on the Trojan-infected ROPUF, a state-of-the-art machine learning algorithm, CNN, is chosen for modeling the exact relationship among $c_1 : c_{128}$, $r_1 : r_{128}$, and $a_1 : a_{128}$.

The detailed architecture of the CNNs for modeling the Trojan-infected ROPUF is designed as shown in Fig. 5.7. The input challenge $c_1 : c_{128}$ and actual output response $r_1 : r_{128}$

of the Trojan-infected ROPUF are selected for training the CNNs. In Fig. 5.7, a $16 \times 16 \times 1$ matrix A that contains the information of $c_1 : c_{128}$ and $r_1 : r_{128}$ is established as the input training data of the CNNs. Furthermore, the oscillating frequency information $a_1 : a_{128}$ is extracted for generating the output training data of the CNNs. By converting the binary data $a_1 : a_{128}$ into the decimal data ω, we can obtain ω as $\omega = \sum_{j=1}^{128} a_j 2^{j-1}$. Suppose the minimum and maximum values of ω are ω_{min} and ω_{max}, respectively. Therefore, if there are N number of different values for the decimal data ω, the i_1th, $(i_1 = 1, 2, ..., N)$ value ω_{i_1} can be derived as

$$\omega_{i_1} = \frac{i_1 - 1}{N - 1}(\omega_{max} - \omega_{min}) + \omega_{min}. \tag{5.1}$$

As shown in Fig. 5.7, the devised CNNs include two convolutional layers. Firstly, when 36 number of 3×3 filters perform the convolution operation on the $16 \times 16 \times 1$ input matrix A with the ReLU function, the matrix A is converted into a $14 \times 14 \times 36$ matrix. Then the $14 \times 14 \times 36$ matrix is transformed into the $7 \times 7 \times 36$ matrix after executing a max pooling operation to extract the critical features. Subsequently, a dropout operation transforms the $7 \times 7 \times 36$ matrix into another $7 \times 7 \times 36$ matrix with a 0.25 probability. These are the operations that are achieved in the 1st convolutional layer.

In the 2nd convolutional layer, as shown in Fig. 5.7, convolution operation, max pooling operation, and dropout operation are performed on the newest $7 \times 7 \times 36$ matrix to create a $6 \times 6 \times 36$ matrix, the $3 \times 3 \times 36$ matrix, and a $3 \times 3 \times 36$ matrix, sequentially. Kindly note that the size of the filters used in the 2nd convolutional layer is 2×2 and the corresponding dropout probability is set as 0.25.

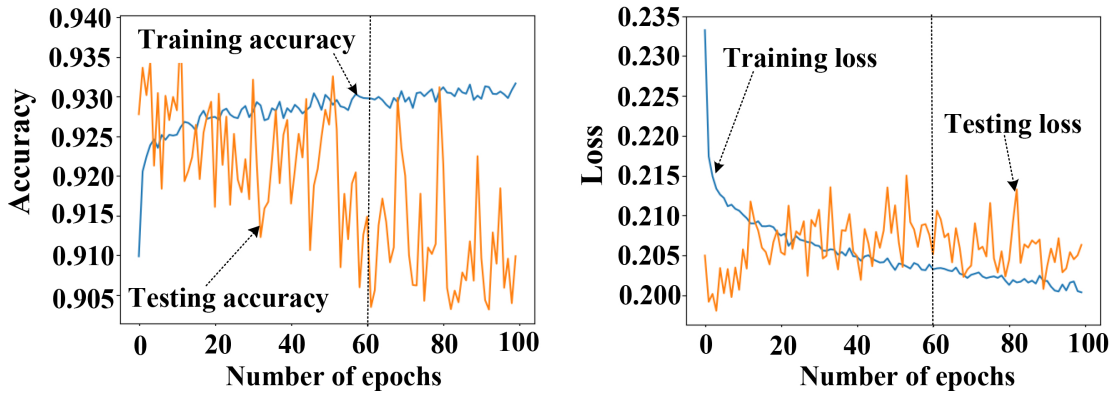


Fig. 5.8: Training result of the devised CNNs with 100,000 number of training data (the batch_size of the training is set as 100 and N is chosen as 10). (a) Accuracy versus number of epochs. (b) Loss versus number of epochs

Once the latest $3 \times 3 \times 36$ matrix in Fig. 5.7 is generated, under the assistance of the flatten operation, the $1 \times 1 \times 324$ array is created. Moreover, after executing dense and dropout operations on the new created $1 \times 1 \times 324$ array sequentially, two intermediate $1 \times 1 \times 512$ arrays are acquired, as indicated in Fig. 5.7. The probability for this dropout operation is set as 0.75. Eventually, when the last dense operation is implemented on the latest $1 \times 1 \times 512$ array with a Softmax function, the $1 \times 1 \times N$ array that is used for classification can be obtained. Kindly note that the $1 \times 1 \times N$ array stores the probabilities: $p_1, p_2, ..., p_N$ for all the possible values of ω_{i_1}: ω_1, ω_2, ..., ω_N to realize classification. For example, if p_1 is the highest probability among all the probabilities: $p_1, p_2, ..., p_N$, the output of the CNNs is selected as ω_1. Similarly, if p_N shows the highest value, the CNNs set the corresponding output as ω_N.

To demonstrate the effectiveness of the CNN attack on the proposed Trojan-infected ROPUF, the training data need to be collected at first. A 130 nm and 128-bit CMOS

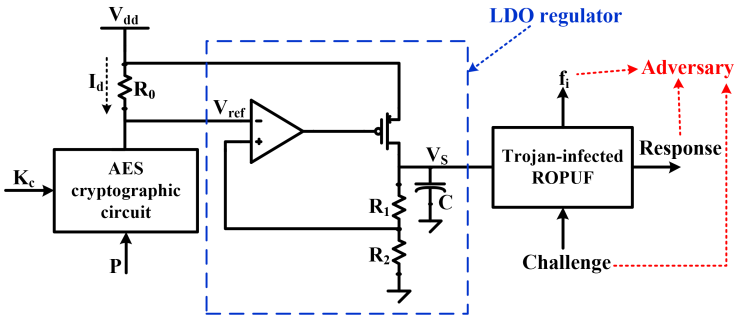


Fig. 5.9: Architecture of the Trojan-infected ROPUF sensor for sensing the dynamic current of an AES cryptographic circuit

Trojan-infected ROPUF is simulated in Cadence. By triggering the embedded Trojan circuit, the input challenge $c_1 : c_{128}$, actual output response $r_1 : r_{128}$, and oscillating frequency information $a_1 : a_{128}$ of the Trojan-infected ROPUF can be extracted from Cadence. As a result, the training data of the CNNs can be acquired from the Cadence simulation. In the CNN training, the loss function and optimizer are chosen with categorical_crossentropy and adam, respectively. After enabling 100,000 data to train the CNNs, as shown in Fig. 5.8, the training (testing) accuracy and loss are about 0.929 (0.910) and 0.203 (0.205), respectively, when the number of epochs is 60. Furthermore, as indicated in Table 5.1, if the number of training data for the CNNs is increased from 100,000 to 200,000, the training accuracy/testing accuracy (TRA/TEA) increases and the training loss/testing loss (TRL/TEL) decreases. However, when the number of training data is further increased from 200,000 to 500,000, the TRA/TEA decreases. That indicates 500,000 training data may cause an "overfitting" problem to the designed CNNs. Therefore, as shown in Table 5.1, machine learning techniques are capable of achieving over 0.94 (0.92) TRA (TEA) on the Trojan-infected ROPUF.

TABLE 5.1: Training results of the CNNs with different number of training data (number of epochs is 60 and batch_size is 100)

Number of training data	TRA	TEA	TRL	TEL
100,000	0.929	0.910	0.203	0.205
200,000	0.945	0.928	0.178	0.194
500,000	0.941	0.917	0.183	0.199

5.4.3 DISCLOSURE OF CONFIDENTIAL DATA WITH SIDE-CHANNEL ATTACKS

Side-channel attacks [17, 63, 79] are a kind of powerful non-invasive attacks that can be utilized by the adversary to leak the secret information of ICs through analyzing the correlation between the processed data and the physical leakages of the ICs (*i.e.* power dissipation, electromagnetic emission, and timing information). In order to demonstrate that leaking the critical frequency f_i of the Trojan-infected ROPUF sensor as mentioned in Section 5.3 is sufficient to retrieve the confidential information of IoT, a representative example about implementing side-channel attacks on the Trojan-infected ROPUF sensor is analyzed in this Section.

Advanced encryption standard (AES) is a popular algorithm that can be utilized to encrypt the critical data against the adversary [17, 79]. Fig. 5.9 shows an architecture of the Trojan-infected ROPUF sensor that is used for monitoring the variations of the dynamic current of an AES cryptographic circuit. Assume the secret key and plaintext of the AES cryptographic circuit are K_c and P, respectively. When different plaintexts are inputted into the AES cryptographic circuit sequentially, the dynamic current I_d of the AES cryptographic circuit varies all the time. If a resistor R_0 is selected to sense the variations of the dynamic

current I_d of the AES cryptographic circuit, as shown in Fig. 5.9, the output critical voltage V_{ref} can be denoted as

$$V_{ref} = V_{dd} - I_d R_0 \tag{5.2}$$

where V_{dd} is the voltage of the power source. Moreover, as shown in Fig. 5.9, a low-dropout (LDO) voltage regulator is utilized to manipulate the supply voltage V_S of the Trojan-infected ROPUF. However, since the LDO regulator is controlled by the critical voltage V_{ref}, the relationship between the supply voltage V_S and the critical voltage V_{ref} in Fig. 5.9 is determined as

$$V_S = (1 + \frac{R_1}{R_2})V_{ref} \tag{5.3}$$

where R_1 and R_2 are the biased resistances of the LDO regulator.

As mentioned in Section 5.2, linear relationships exist between the supply voltage and the oscillating frequencies of the ring oscillators of the ROPUF. Therefore, the oscillating frequency f_i of the ith ring oscillator loop RO_i in the ROPUF can be precisely derived as

$$\begin{aligned} f_i &= \lambda_i V_S + \beta_i = \lambda_i (1 + \frac{R_1}{R_2}) V_{ref} + \beta_i \\ &= \lambda_i (1 + \frac{R_1}{R_2})(V_{dd} - I_d R_0) + \beta_i = G(I_d) \end{aligned} \tag{5.4}$$

where λ_i and β_i are the slope and constant of the frequency-voltage curve of RO_i in the ROPUF, respectively. $G(I_d)$ is the linear function that is used to represent the relationship between the dynamic current I_d and the critical frequency f_i. For the Trojan-infected ROPUF sensor as shown in Fig. 5.9, the critical frequency f_i is leaked to the adversary via the inserted Trojan circuit and machine learning techniques. As a result, the adversary may obtain the confidential information of the AES cryptographic circuit through analyzing the variations of the critical frequency f_i.

If the adversary selects the side-channel analysis to retrieve the secret key K_c of the AES cryptographic circuit in the Trojan-infected ROPUF sensor, the correlation between the input plaintext P and the leaked frequency f_i is likely to be explored by the adversary. Assume X number of different plaintexts: P_1, P_2, ..., P_X are inputted into the AES cryptographic circuit, the corresponding dynamic current values of the AES cryptographic circuit are: $I_{d,1}$, $I_{d,2}$, ..., $I_{d,X}$. Then the correlation between P_1, P_2, ..., P_X and $I_{d,1}$, $I_{d,2}$, ..., $I_{d,X}$ can be explored to disclose the secret key K_c with the side-channel analysis. However, since the critical frequency f_i strongly correlates with the dynamic current I_d as shown in (3), the correlation between P_1, P_2, ..., P_X and $G(I_{d,1})$, $G(I_{d,2})$, ..., $G(I_{d,X})$ can also be studied to retrieve the secret key K_c.

As to the Trojan-free ROPUF sensor, conversely, the critical frequency f_i is unknown to the adversary. If the side-channel analysis is performed on the Trojan-free ROPUF sensor, the adversary may analyze the correlation between the input plaintexts of the AES cryptographic circuit and the output cipher responses of the ROPUF to estimate the secret key K_c.

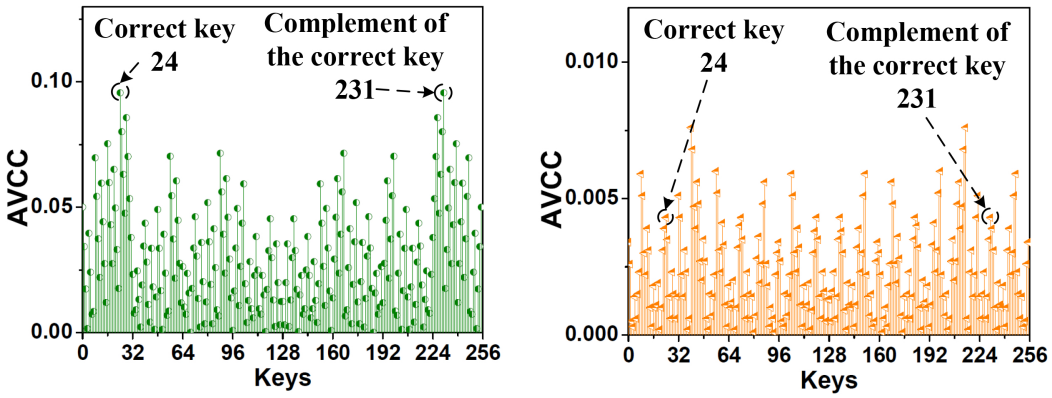


Fig. 5.10: Absolute value of correlation coefficient (AVCC) versus all the possible keys for the ROPUF sensors under the side-channel analysis (Hamming-weight model is used). (a) Trojan-infected ROPUF sensor with 3,000 input plaintexts. (b) Trojan-free ROPUF sensor with 1 million input plaintexts

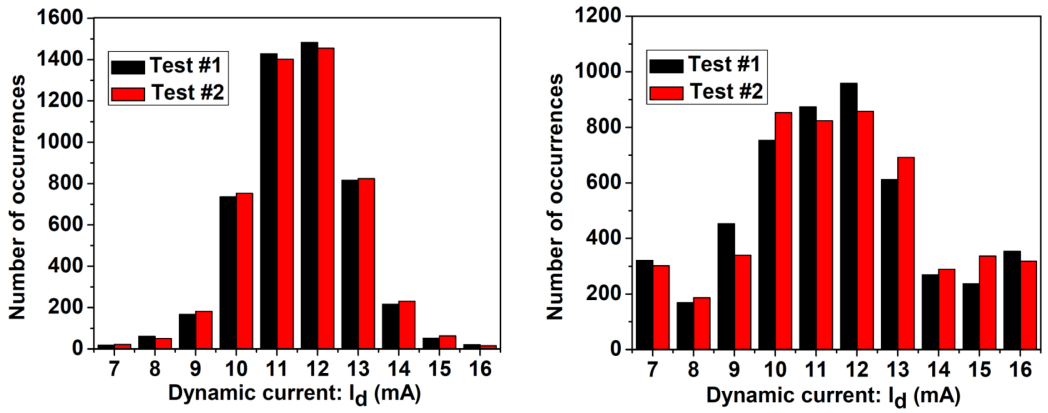
Both the Trojan-free and Trojan-infected ROPUF sensors are simulated in Cadence with the 130 nm CMOS technology kits. A 128-bit AES cryptographic circuit acts as the sensing load for both the Trojan-free and Trojan-infected ROPUF sensors in the simulations. As shown in Fig. 5.10(a), if the side-channel analysis is applied on the Trojan-infected ROPUF sensor by exploring the correlation between P_1, P_2, ..., P_X and $G(I_{d,1})$, $G(I_{d,2})$, ..., $G(I_{d,X})$, only 3,000 plaintexts are sufficient to leak the secret key 24 that corresponds to a substitution-box (S-box) of the AES cryptographic circuit due to the leakage of the critical frequency f_i. By contrast, as shown in Fig. 5.10(b), even if 1 million plaintexts are enabled on the Trojan-free ROPUF sensor, the secret key 24 corresponds to the S-box of the AES cryptographic circuit is successfully masked from being disclosed to the adversary since no critical leakage is available for the adversary.

5.5 TROJAN DETECTION

In a Trojan-free ROPUF IoT sensor, the original physical quantity is sensed and converted into a critical voltage signal at first. Then both the critical voltage signal and the input challenge are encrypted by the Trojan-free ROPUF to generate the output cipher data. Subsequently, the cipher data will be sent to the host via wireless communication. Once the cipher data are received by the host, the values of the original physical quantity can be deciphered through checking the stored look-up table (LUT) of the Trojan-free ROPUF. Since the physical quantity commonly conforms to a Gaussian distribution [17, 86], the host is able to obtain Gaussian distributed sensed data after deciphering the received cipher data.

By contrast, for a Trojan-infected ROPUF IoT sensor, the case is different. As shown in Fig. 5.5, when the embedded Trojan circuit is inactive, the Trojan-infected ROPUF behaves like a Trojan-free ROPUF. As a result, the host is also capable of acquiring Gaussian distributed sensed data. However, once the embedded Trojan circuit is activated, as mentioned in Section 5.4.1, the Trojan-infected ROPUF will send false cipher data to the host. After deciphered the false cipher data, the obtained data are closer to random data that may not conform to a Gaussian distribution.

To verify the effectiveness of the proposed Trojan detection methodology, the architecture as shown in Fig. 5.9 is studied as the representative sample. When two different cases: an AES cryptographic circuit with a Trojan-free ROPUF sensor and an AES cryptographic circuit with a Trojan-infected ROPUF sensor are selected for studying the Trojan detection, the sensed data is the dynamic current I_d of the AES cryptographic circuit. After executing two independent tests: Test #1 and Test #2 on these two cases, the corresponding



(a) (b)

Fig. 5.11: Distributions of the sensed data after executing two independent tests (each test contains 5,000 sensed data). (a) An AES cryptographic circuit with a Trojan-free ROPUF sensor. (b) An AES cryptographic circuit with a Trojan infected ROPUF sensor

distributions of the sensed data are obtained in Fig. 5.11. As indicated in Fig. 5.11(a), the sensed dynamic current I_d of an AES cryptographic circuit with a Trojan-free ROPUF sensor well conforms a Gaussian distribution. Conversely, for an AES cryptographic circuit with a Trojan-infected ROPUF sensor, Fig. 5.11(b) shows the sensed dynamic current I_d complies with a non-Gaussian distribution. Accordingly, by monitoring the distribution of the sensed data in real-time, the Trojan-infected ROPUF sensor chips will be identified.

5.6 CONCLUSION

A hardware Trojan attack is performed in the ring oscillator PUF (ROPUF) sensors of IoT by embedding a sequential Trojan circuit into the chip to leak the critical frequency to the adversary. As demonstrated in the result, after leaking 200,000 critical data by triggering the embedded Trojan circuit, the Trojan-infected ROPUF can be precisely modeled under the assistance of machine learning techniques. Additionally, by monitoring the distribution

of the sensed data received from each IoT sensor in real-time, the Trojan-infected ROPUF sensor chips can be successfully detected.

CHAPTER 6

COMBINING SIDE-CHANNEL ANALYSIS AND MACHINE LEARNING MODELS IN EFFICIENT ATTACKS ON XOR PUFS

6.1 MOTIVATION

In Chapter 2 and Chapter 3, two PUF designs are proposed to enhance the non-linear relationship between input challenge and output response[1]. The main reason is that under the hard limitation of hardware resources, the most effective approach to promote the PUF security is to increase the non-linearity [88, 89, 90]. So as to improve the degree of non-linearity between input challenge and output response against ML attacks, XOR PUFs [91, 92] were proposed by using XOR functions to process the corresponding output response. One thing that needs to be emphasized is that the increasing of algorithm complexity cannot confront all machine learning attacks. However, under the same network scale and cracking time, if a PUF is designed with a more complex non-linear algorithm, attackers may spend more time and effort on the data analysis.

To some extent, a simple convolutional neural network (CNN) model may not be efficient to an XOR PUF with complex non-linear algorithm. The plausible explanation is that the CNN attack is unable to extract the critical correlation among the N input challenge bits of the XOR PUF since the N input challenge bits are mutually independent. However,

[1]The content of this Chapter partially has been published in [87].

if a SCA attack is performed prior to CNN attack, will some non-linearity be reduced or eliminated, and result in a vulnerability to further CNN attacks? In this short section, a hybrid attack model is performed on a XOR PUF to examine its robustness to the novel hybrid attack model.

6.2 PRINCIPLE OF SC-ASSISTED CNN ATTACK

Fig. 6.1 shows the fundamental architecture of a 128-bit arbiter XOR PUF. The 128-bit input challenge $(c_1, c_2, ..., c_{128})_2$ is fed into four identically designed PUF blocks: arbiter PUF #1, arbiter PUF #2, arbiter PUF #3, and arbiter PUF #4, respectively. Due to the affects of random fabrication process, these four PUF blocks generate four different responses: r_1, r_2, r_3, and r_4, as shown Fig. 6.1. Eventually, an XOR operation is performed on r_1, r_2, r_3, and r_4 to create the output response R of the XOR arbiter PUF ($R = r_1 \oplus r_2 \oplus r_3 \oplus r_4$). If a regular CNN attack is executed to model the relationship between the input challenge $(c_1, c_2, ..., c_{128})_2$ and the output response R of the XOR arbiter PUF, the 128-bit input challenge $(c_1, c_2, ..., c_{128})_2$ will be processed by convolution operations. Unfortunately, no significant features can be extracted after applying the convolution operations since there is no correlation among the 128 input challenge bits: $(c_1, c_2, ..., c_{128})_2$.

In order to generate highly correlated input challenge bits to significantly improve the efficacy of CNN attack on the 128-bit XOR arbiter PUF, SC analyses can be utilized to pre-process the 128 uncorrelated input challenge bits: $(c_1, c_2, ..., c_{128})_2$. In the proposed SC-assisted CNN attack, the 128-bit input challenge $(c_1, c_2, ..., c_{128})_2$ is added to a 128-bit intermediate data $(a_1, a_2, ..., a_{128})_2$ to generate the 128 correlated input challenge bits: $(c_1^*, c_2^*, ..., c_{128}^*)_2$ where $c_i^* = c_i \oplus a_i, (i = 1, 2, ..., 128)$. Suppose the input challenge

$(c_1, c_2, ..., c_{128})_2$ is uniformly divided into m groups, the input challenge of the kth, ($k = 1, 2, ..., m$) group can be denoted as $(c_{(k-1)\frac{128}{m}+1}, c_{(k-1)\frac{128}{m}+2}, ..., c_{k\frac{128}{m}})_2$. As a result, the correlated input challenge of the kth group can be derived as $(c^*_{(k-1)\frac{128}{m}+1}, c^*_{(k-1)\frac{128}{m}+2}, \cdots, c^*_{k\frac{128}{m}})_2 = (c_{(k-1)\frac{128}{m}+1} \oplus a_{(k-1)\frac{128}{m}+1}, c_{(k-1)\frac{128}{m}+2} \oplus a_{(k-1)\frac{128}{m}+2}, ..., c_{k\frac{128}{m}} \oplus a_{k\frac{128}{m}})_2$.

To obtain the optimum $(a_{(k-1)\frac{128}{m}+1}, a_{(k-1)\frac{128}{m}+2}, ..., a_{k\frac{128}{m}})_2$ for achieving the target challenge sequence $(c^*_{(k-1)\frac{128}{m}+1}, c^*_{(k-1)\frac{128}{m}+2}, \cdots, c^*_{k\frac{128}{m}})_2$ with the maximum correlation, SC analyses can be deployed to realize the optimization. Since $(a_{(k-1)\frac{128}{m}+1}, a_{(k-1)\frac{128}{m}+2}, ..., a_{k\frac{128}{m}})_2$ is a $\frac{128}{m}$-bit binary data, we can hypothesize $2^{128/m}$ possible values from $(0, 0, ..., 0)_2$ to $(1, 1, ..., 1)_2$ for it. Assume the power consumption of the 128-bit XOR arbiter PUF is P_d. If n number of different $(c_{(k-1)\frac{128}{m}+1}, c_{(k-1)\frac{128}{m}+2}, ..., c_{k\frac{128}{m}})_2$ values are inputted into the 128-bit XOR arbiter PUF, the corresponding n number of different P_d values can be collected. Suppose hamming-weight (HW) model is selected for executing the SC analyses, the correlation between $\sum_{j=1}^{128/m} c_{(k-1)\frac{128}{m}+j} \oplus a_{(k-1)\frac{128}{m}+j}$ and P_d can be studied for estimating the optimum $(a_{(k-1)\frac{128}{m}+1}, a_{(k-1)\frac{128}{m}+2}, ..., a_{k\frac{128}{m}})_2$. When $2^{128/m}$ different correlation coefficients are acquired by hypothesizing $2^{128/m}$ possible values for $(a_{(k-1)\frac{128}{m}+1}, a_{(k-1)\frac{128}{m}+2}, ..., a_{k\frac{128}{m}})_2$, the possible value that corresponds to the highest correlation coefficient is regarded as the optimum $(a_{(k-1)\frac{128}{m}+1}, a_{(k-1)\frac{128}{m}+2}, ..., a_{k\frac{128}{m}})_2$ value. Under such a condition, the 128-bit intermediate data $(a_1, a_2, ..., a_{128})_2$ can be fully determined through applying the SC analyses on each group individually. Moreover, the total computational complexity for estimating $(a_1, a_2, ..., a_{128})_2$ is $m \times 2^{128/m}$.

After executing the SC analyses on the 128-bit XOR arbiter PUF, the new 128-bit input challenge $(c^*_1, c^*_2, ..., c^*_{128})_2$ can be generated. To study the correlation among c^*_1, c^*_2, ..., c^*_{128},

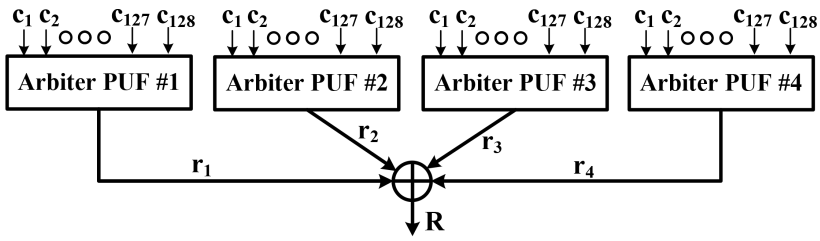
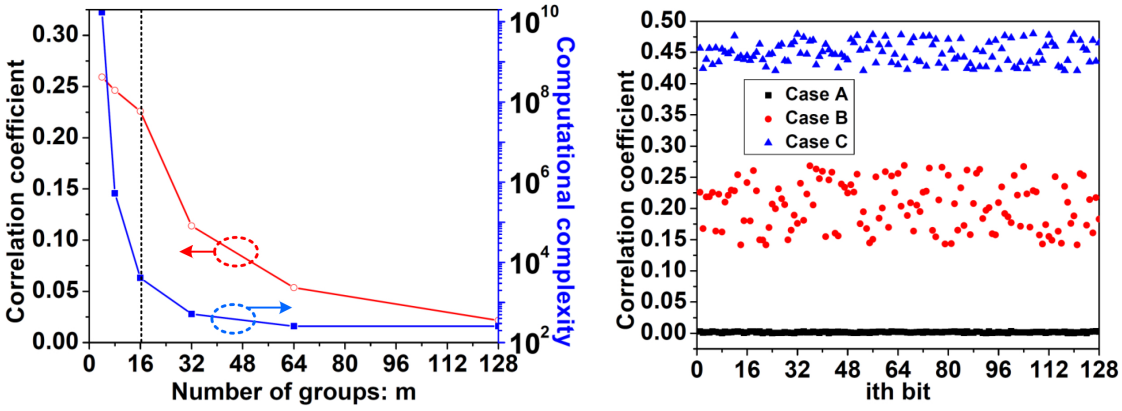


Fig. 6.1: Basic architecture of a 128-bit XOR arbiter PUF (r_1, r_2, r_3, r_4, and R are single bit data)



(a) (b)

Fig. 6.2: Correlation analyses for the original input challenge $(c_1, c_2, ..., c_{128})_2$ and new input challenge $(c_1^*, c_2^*, ..., c_{128}^*)_2$. (a) Correlation coefficient (between c_1^* and $\sum_{i=1}^{128} c_i^*$) and computational complexity versus number of group m. (b) Correlation coefficient verus i_{th} bit for Case A, Case B, and Case C ($m = 16$)

two new correlation coefficients: $r(c_i^*, \sum_{i=1}^{128} c_i^*)$ and $r(c_i^*, \sum_{j=1}^{128/m} c^*_{[\frac{i\times m}{128}]\times\frac{128}{m}+j})$ need to be defined. $r(c_i^*, \sum_{i=1}^{128} c_i^*)$ represents the correlation coefficient between the ith bit: c_i^* and the HW of all the bits: $\sum_{i=1}^{128} c_i^*$. Similarly, $r(c_i^*, \sum_{j=1}^{128/m} c^*_{[\frac{i\times m}{128}]\times\frac{128}{m}+j})$ denotes the correlation coefficient between the ith bit: c_i^* and the HW of the group that includes c_i^*: $\sum_{j=1}^{128/m} c^*_{[\frac{i\times m}{128}]\times\frac{128}{m}+j}$.

The 128-bit XOR arbiter PUF as shown in Fig. 6.1 is designed and simulated in Cadence with the 130 nm CMOS technology kit. The architecture of the four identically designed PUF blocks in Fig. 6.1 is chosen from [93]. Moreover, the SC analyses are executed on the

128-bit XOR arbiter PUF by extracting the correlation between the input challenge and the power consumption of the PUF from the simulation. As shown in Fig. 6.2(a), a balanced correlation coefficient 0.2258 and computational complexity 4096 can be achieved for the new input challenge $(c_1^*, c_2^*, ..., c_{128}^*)_2$ if the number of groups m is chosen as 16. In Fig. 6.2(b), three different cases: *Case A*, *Case B*, and *Case C* are analyzed. *Case A* (*Case B*) reflects the correlation coefficient between the ith bit and the HW of all the bits related with the original input challenge $(c_1, c_2, ..., c_{128})_2$ (new input challenge $(c_1^*, c_2^*, ..., c_{128}^*)_2$). When we compare *Case A* and *Case B*, it is apparent that the correlation coefficients among all the input challenge bits are significantly increased after applying the SC analyses. Furthermore, *Case C* in Fig. 6.2(b) represents the correlation coefficient between the ith bit and the HW of the group that includes c_i^* associated with $(c_1^*, c_2^*, ..., c_{128}^*)_2$. If *Case B* and *Case C* are selected for comparison, it demonstrates that the correlation among the neighbor data in the new input challenge $(c_1^*, c_2^*, ..., c_{128}^*)_2$ is strongly reinforced. This strong correlation can be used for improving the efficiency of CNN attack on the XOR arbiter PUF.

6.3 COMPARISON BETWEEN THE PROPOSED ATTACK AND A REGULAR CNN ATTACK

As demonstrated in Fig. 6.2(a), if the number of groups m is set as 16, the optimum SC analyses can be realized. In such a case, if a CNN attack is executed to process the correlated input challenge $(c_1^*, c_2^*, ..., c_{128}^*)_2$, the $32 \times 4 \times 1$ matrix C^* as shown in Fig. 6.3 is created as the input training data for the CNNs. Kindly note that the challenge bits within group #1, group #2, ... group #16 are placed together in the matrix C^*. The primary intention is maximizing the correlation among the neighbor data for assisting the convolution operations. In the convolutional stage, 32 number of 3×3 filters are performed

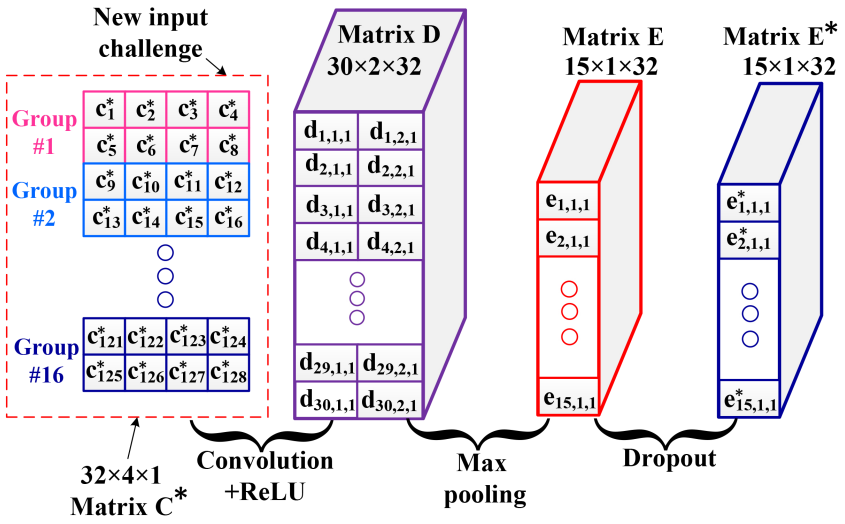


Fig. 6.3: Convolutional layer of the SC-assisted CNN attack for modeling the 128-bit XOR arbiter PUF

on the matrix C^* to generate a $30 \times 2 \times 32$ matrix D with the convolution operations and a ReLU function, as shown in Fig. 6.3. For instance, the element $d_{1,1,1}$ of the matrix D in Fig. 6.3 can be written as

$$d_{1,1,1} = max\{0, \sum_{h=1}^{6} d^*_{1,1,1,h} c^*_h\} \tag{6.1}$$

where $d^*_{1,1,1,1}$, $d^*_{1,1,1,2}$, ..., $d^*_{1,1,1,6}$ are the 6 parameters of the filter that are related with the element $d_{1,1,1}$. After executing the max pooling operation, the $30 \times 2 \times 32$ matrix D is transformed into the $15 \times 1 \times 32$ matrix E, as shown in Fig. 6.3. Then a dropout operation is used for converting the $15 \times 1 \times 32$ matrix E into another $15 \times 1 \times 32$ matrix E^* with a 0.2 dropout probability. These aforementioned steps are achieved in the convolutional layer of the CNNs.

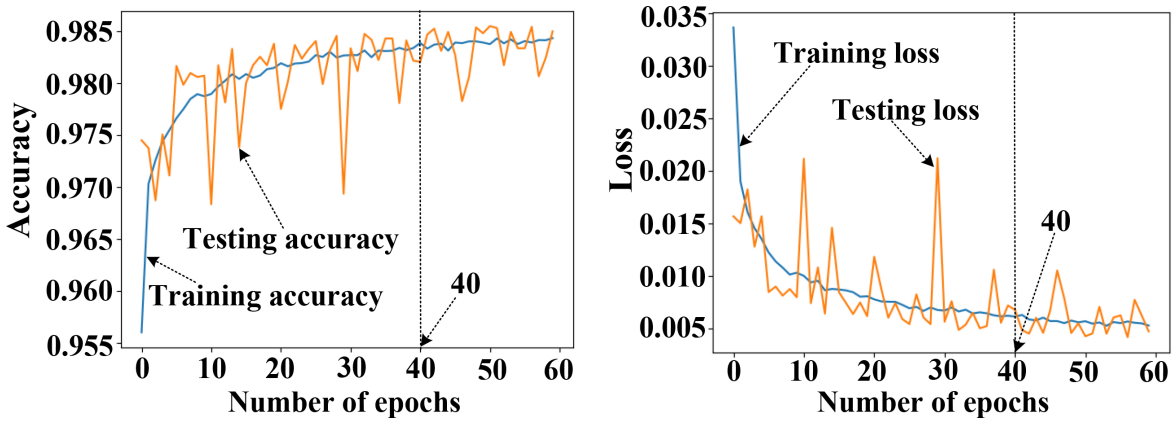
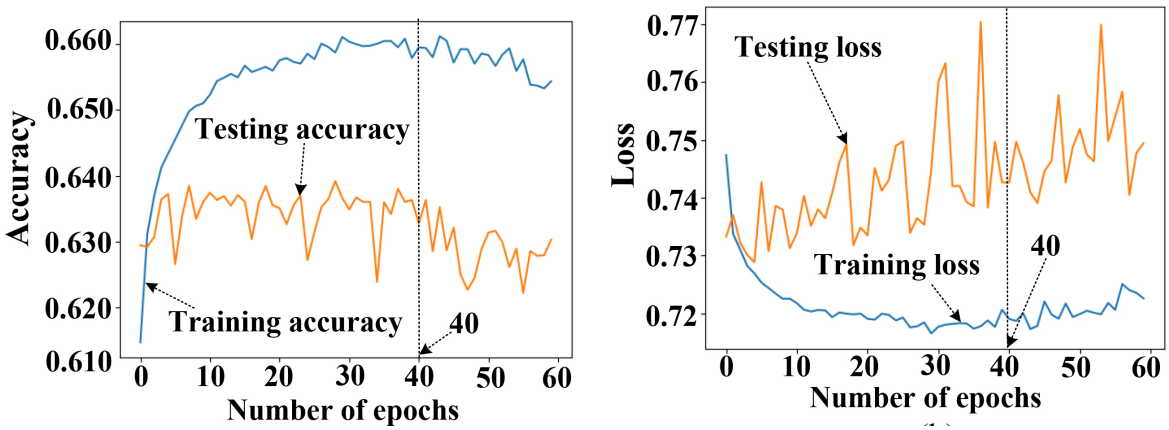


Fig. 6.4: Training result of the SC-assisted CNN attack on the 128-bit XOR arbiter PUF. (a) Accuracy versus number of epochs. (b) Loss versus number of epochs

So as to classify the output response R of the 128-bit XOR arbiter PUF with a high logic value "1" or a low logic value "0", the $15 \times 1 \times 32$ matrix E^* needs to be converted into a $1 \times 1 \times 2$ array ultimately. To achieve this goal, in the designed CNNs, the $15 \times 1 \times 32$ matrix E^* is converted into the $1 \times 1 \times 480$ array, a $1 \times 1 \times 720$ array, a $1 \times 1 \times 720$ array, and a $1 \times 1 \times 2$ array by using a flatten operation, a dense operation, a dropout operation, and a dense operation, respectively. The dropout probability of the last dropout operation is set as 0.8 and a Softmax function is used in the final dense operation.

To train the CNNs that are used for executing the SC-assisted CNN attack, the corresponding loss function and optimizer are chosen as binary_crossentropy and adam, respectively. The batch size for the CNN training is optimized with 100. As shown in Fig. 6.4(a), after enabling 150,000 training data, the SC-assisted CNN attack is able to achieve a 0.984 (0.982) training (testing) accuracy on the 128-bit XOR arbiter PUF when the number of epochs is chosen as 40. Fig. 6.4(b) shows the corresponding training (testing) loss is about



(a) (b)

Fig. 6.5: Training result of the regular CNN attack on the 128-bit XOR arbiter PUF. (a) Accuracy versus number of epochs. (b) Loss versus number of epochs

0.006 under such a training (testing) accuracy.

If a regular CNN attack is performed on the 128-bit XOR arbiter PUF, the input training data needs to be replaced with the original input challenge $(c_1, c_2, ..., c_{128})_2$. This means the elements of the $32 \times 4 \times 1$ matrix C^* in Fig. 6.3 are substituted with c_1, c_2, ..., c_{128}. After training the CNNs associated with the regular CNN attack with 150,000 data, the training (testing) accuracy is about 0.658 (0.633) if the number of epochs is set as 40, as shown in Fig. 6.5(a). Moreover, Fig. 6.5(b) reflects a much higher training (testing) loss will occur under the regular CNN attack.

6.4 CONCLUSION

A novel side-channel (SC)-assisted convolutional neural network (CNN) attack is proposed in this letter to model non-linear PUFs: XOR PUFs. By utilizing SC analyses to add strong correlation among the input training data, the training/testing accuracy of the CNN

attack is improved over 0.98. By contrast, the training/testing accuracy of a regular CNN attack is about 0.64 without the assistance of the strong correlation.

CHAPTER 7

CONCLUSION

Physical unclonable function (PUF) research is a core topic in the domain of hardware security. Workload aware multi-phases voltage regulator (WAMPVR) PUFs utilize capacitance differences as compared parameter and introduce serial diode sets to increase the non-linearity in voltage signals. Wave dynamic differential logic (WDDL) PUF introduce an existing side-channel-resistant hardware component into PUF design. By utilizing a non-linear function on circuit level, the power entropy is greatly promoted, enhancing the PUF resilience to ML attacks and SCA attacks. Although both PUF designs are proved to be robust to non-invasive attacks. The usage of non-linear function, however, introduces more randomness in circuit function, providing additional stability and robustness against more advanced ML attack models.

Compared to WAMPVR PUF and WDDL PUF, the conceptual design of PUF-AES-PUF architecture considers more vulnerable leakage paths of confidential information. If a cryptographic circuit is planed in the middle of the circuit, both input power trace and output power trace are obscure to attackers. While one end of both PUFs are not open to attackers, attackers can no longer perform modeling attacks solely on a single protection mechanism. The uncertainties induced by secret key also enhance the PUF robustness to deep neural network (DNN) models.

Suppose the PUF primitive is implemented with an unwilling Trojan inclusion in the

fabrication process, the relationship between input physical signal and output digital responses are revealed to attackers. Comparing to an integral PUF protection, the Trojan inclusion eliminates most uncertainties on devices, resulting in a vulnerability to ML modeling attacks and succedent SCA attacks. If the existence of Trojan can be aware of, since the Trojan will alter the electrical signal in an unnatural manner, the sensed data will no longer conform to Gaussian distribution. Thus, the statistical method is effective in the detection of hardware Trojans.

In addition, to some floorplans with simple or single protection mechanisms, a profiling attack model jointly using ML attack models and SCA attack approach can easily penetrate the secure hardware model. With the participance of SCA approach, more circuit features are extracted from leakage signals. The possibility to crack non-linear XOR PUF is thus promoted to 0.98, comparing to the accuracy of 0.65 under a simple CNN attack model.

CHAPTER 8

FUTURE WORK

8.1 MACHINE LEARNING-BASED PHYSICAL UNCLONABLE FUNCTION (PUF)-LIKE MODULES

In the whole dissertation, all subjects are around the concept to enhancing the internal algorithm logic/non-linearity against ML attacks and/or SCA attacks. To a certain extent, the enhancement of algorithm does increase the robustness to both attack models. However, the comparison logic, after all, is based on two sets of hardware modules. Any increments in algorithm complexity will result in additional consumption in area, power budget, etc. As an engineer in hardware security, our ultimate goal is to use resources as little as possible to resolve potential threats ultimately. Therefore, based on the same topic of PUF, is there any solution that holds confidentiality, flexibility, and expansibility at the same time? Under this assumption, we believe the combination of ML algorithms and PUF primitives can be a promising research topic in this area.

First of all, same as all cryptographic-purpose system, the architecture of all PUF primitive are public, while the concrete variable/key details of inner components are kept as secret information. This theory does not only facilitate the information exchange in cryptographic system, but also enables hardware designers take advantage of known secret inner components in their architecture designs. However, in view of current hardware architecture and hardware security theory, the bottleneck of PUF industry is the rigid restriction of area and power budget stipulate the scale of PUF as well as its internal algorithm complexity.

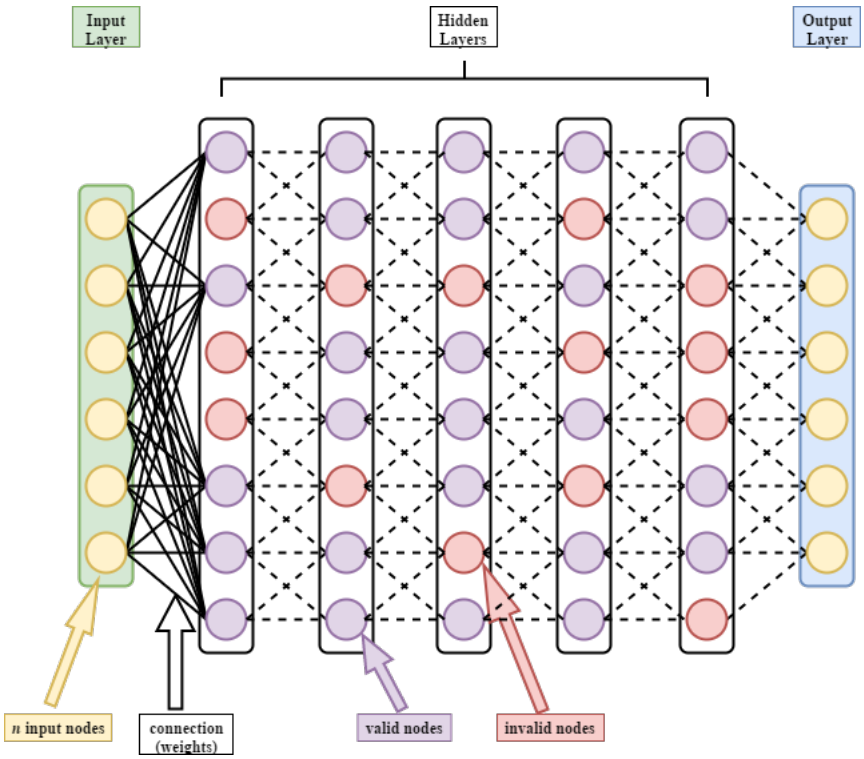


Fig. 8.1: Conceptual floorplan of neural network PUF-like module.

Under this situation, a PUF primitive may always be compromised with a modeling attack under a finite time complexity. In addition, from the perspective of all proposed PUF designs, all uncertainties are fixed when the PUF layout is decided. In accompany with a finite algorithm complexity, the series of Taylor expansion cannot be too large, resulting in vulnerabilities in differential cryptanalysis and linear cryptanalysis. And that is also why we introduce artificial uncertainty dimensions in Chapter 4. Furthermore, with the advent of machine learning algorithm and quantum computers in the foreseeable future, all current cryptographic algorithms with fixed complexity or finite uncertainties may be cracked with more advanced modeling algorithms.

In view of all aforementioned problems, one promising future work is to exploit machine

learning architecture as PUF-like hardware architecture. Please kindly note that the conceptual architecture is not simply from manufacture uncertainties as conventional designs. As is shown in Fig. 8.1 is conceptual floorplan of neural network PUF-like module. Similar to regular PUF architecture, the input layer and the output layer stipulate the dimension of the challenge signal and response signal. The hidden layer forms the internal arithmetic logic of the PUF module. Different from regular neural network architecture, not all nodes in hidden layers are utilized, as well as their corresponding connection weights. Comparing to conventional PUF architecture, the conceptual PUF-like module has following features. First, the architecture is analogous to field programming gate array (FPGA). Since not all internal logic/arithmetic nodes will be used, hardware designers can decide which nodes/weights are to be used in either training process or validation process. Besides, the growth of machine learning algorithms makes it possible to embed NN-based hardware module into chip layouts. Once the NN module is embedded onto hardware architectures, Node weights of the network module can be rewritten in execution. This means the network module can be designed for multi-purposes. If the system asks for a key or an authentication code, the neural network can terminate the regular work for several cycles and generate some pairs of CRPs. This greatly reduces the overhead that introduces a non-systematical functional unit into hardware designs. Last but not least, more uncertainty dimensions will be introduced with this design. On the one hand, the number of input nodes and output nodes are uncertain. Hardware designers can design their specific mappings from challenges to responses, which gives great degree of freedom in security designs. On the other hand, since the usage of nodes are not fixed. Even attackers are able to compromise one network architecture,

security supervisors can always alter the network structure or change connection weights as new safety strategies. Although some details in realizations are not well considered, this conceptual architecture can still be one promising research directions in the future.

8.2 LOW/REDUCED OVERHEAD PHYSICAL UNCLONABLE FUNCTION DEVICES

Similar to every innovate idea and designs in hardware industry, PUF primitives, after all, are still hardware devices and is of commodity value attributes. Except for the infinite pursuit on security attributes, another crucial design principle of PUF is the reduction of excess overhead on area, power, and all other physical dimensions that may degrade the overall performance of the computer system. Though protection mechanisms on the software system also requires the actual impact as few as possible, comparing to the total resources on hardware architectures, software security strategies, like AES and RSA cryptography, possesses relative unlimited computational resources and can be optimized on the algorithm level. A more real question that the PUF industry has to face is no matter how efficient the PUF can be exploited as hardware security strategies, if the degradation of performance is not negligible, any PUF-based security chip will be weak in market competitions. As the final tradeoff in the market place, PUF, as an additional security module, is the first one to be sacrificed. Eventually, the last but not least barrier before PUF devices can be negotiated to large-scale applications is the overhead problem.

In previous sections, most words underline the promotion of security of PUF against potential attacking models, to be specific, ML attacks and SCA attacks. In Chapter 2 and Chapter 3, we manage to exploit existing hardware architectures or mechanisms as the

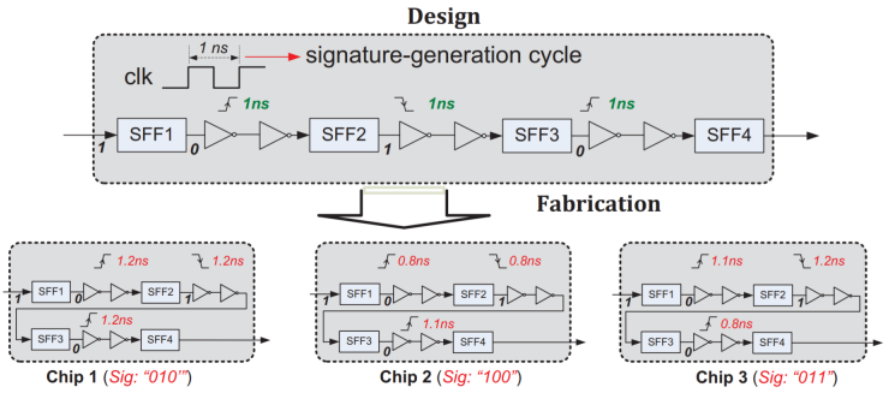


Fig. 8.2: Illustration of PUF realization using on-chip scan structure [7].

basis of innovate PUF primitives. And in Chapter 4, our strategy is to use existing low-overhead designs (RO PUF) and essential security modules (AES cryptographic circuit) to reconstruct a PUF logic. Since all architecture designs utilize existing modules on system, no discussions are developed in details on further reducing consumption on hardware resources in hardware system, though we equally believe the overhead problem is not negligible in more severe security markets in the touchable future. However, considering hardware threats are already proposed and studied by both researchers and "evils behind computers", we cannot wait for the emergence of attacks, and not until then did we equip our hardware system with PUF primitives. Therefore, we choose to investigate the feasibility of most accessible architectures at first and regard the deduction of resource usage as future works.

According to accessible literatures, early discussions on low overhead PUF designs are no later than 2008 [94], though designs of PUF primitives can be traced back to seven years before [95]. Early works also do not mention too much on power/area budgets, while they mostly assert the execution time of security authentication is a small proportion comparing to the time of system usage. A meaningful work on PUF overhead exhibits an experimental

result on a *scanPUF* [7], as is shown in Fig. 8.2. In the proposed work, the PUF is realized on a scan mechanism on two flip-flop paths. And users use experiments on FPGA implementation to prove the innovate PUF consume 11.17% excess area compared to conventional RO-PUF, while the overall power budget is elevated by 9.65%. In a more recent work, by introducing mechanism of self-regulation and reconfiguration, [96] designs an inverter-based PUF with zero-overhead stabilization scheme. Based on our current work, the security attribute is still our priority in PUF designs, in which we are aiming to an ultimate and general solution to all current malicious attacks in hardware system. As a result, one possible research work in the future should be the overhead reduction of our current PUF-AES-PUF architecture, since it is believed to be most robustness in all our current designs. Another possible work can be the arithmetic redesign on PUF comparison. As is stated in previous sections, the non-linearity is a key factor in evaluating PUF performance. We may think of a low-overhead comparison system that balance the resource budget and the non-linearity simultaneously.

REFERENCES

[1] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, pp. 541–552, May 2002.

[2] R. Elnaggar and K. Chakrabarty, "Machine learning for hardware security: opportunities and risks," *Journal of Electronic Testing*, vol. 34, no. 2, pp. 183–201, 2018.

[3] O. A. Uzun and S. Köse, "Converter-gating: A power efficient and secure on-chip power delivery system," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 2, pp. 169–179, 2014.

[4] Y. Gao, H. Ma, D. Abbott, and S. F. Al-Sarawi, "PUF sensor: Exploiting PUF unreliability for secure wireless sensing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2532–2543, 2017.

[5] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1200–1205, 2005.

[6] J. Zhang and L. Wan, "Cmos: Dynamic multi-key obfuscation structure for strong pufs," *arXiv preprint arXiv:1806.02011*, 2018.

[7] Yu Zheng, A. R. Krishna, and S. Bhunia, "Scanpuf: Robust ultralow-overhead puf using scan chain," in *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 626–631, Jan 2013.

[8] S. Ghandali, T. Moos, A. Moradi, and C. Paar, "Side-channel hardware Trojan for provably-secure SCA-protected implementations," *arXiv preprint arXiv:1910.00737*, 2019.

[9] K. S. Subraman, A. Antonopoulos, A. A. Abotabl, A. Nosratinia, and Y. Makris, "Demonstrating and mitigating the risk of an FEC-based hardware Trojan in wireless networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2720–2734, 2019.

[10] V. Govindan, S. Koteshwara, A. Das, K. K. Parhi, and R. S. Chakraborty, "ProTro: A probabilistic counter based hardware Trojan attack on FPGA based MACSec enabled Ethernet switch," in *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pp. 159–175, Springer.

[11] J. Delvaux, "Machine-learning attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF–FSMs," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2043–2058, 2019.

[12] N. A. Hazari, A. Oun, and M. Niamat, "Analysis and machine learning vulnerability assessment of XOR-inverter based ring oscillator PUF design," in *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 590–593, IEEE.

[13] M. Yan, R. Sprabery, B. Gopireddy, C. Fletcher, R. Campbell, and J. Torrellas, "Attack directories, not caches: Side channel attacks in a non-inclusive world," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 888–904, IEEE.

[14] C. Glowacz and V. Grosso, "Optimal collision side-channel attacks," 2019.

[15] D. Schepers, A. Ranganathan, and M. Vanhoef, "Practical side-channel attacks against WPA-TKIP," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pp. 415–426.

[16] M. Schwarz, "Software-based side-channel attacks and defenses in restricted environments part," 2019.

[17] W. Yu and S. Köse, "A lightweight masked AES implementation for securing IoT against CPA attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 11, pp. 2934–2944, 2017.

[18] M. Djukanovic, L. Giancane, G. Scotti, A. Trifiletti, and M. Alioto, "Leakage power analysis attacks: Effectiveness on DPA resistant logic styles under process variations," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 2043–2046, May 2011.

[19] S. D. Kumar and H. Thapliyal, "Security evaluation of MTJ/CMOS circuits against power analysis attacks," in *2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, pp. 117–122, Dec 2017.

[20] R. Agrawal and R. Vemuri, "On state encoding against power analysis attacks for finite state controllers," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 181–186, April 2018.

[21] J. Yang, F. Dai, J. Wang, J. Zeng, Z. Zhang, J. Han, and X. Zeng, "Countering power analysis attacks by exploiting characteristics of multicore processors," *IEICE Electronics Express*, p. 15.20180084, 2018.

[22] Y.-J. Kang, K.-H. Kim, and H. Lee, *Scrambler Based AES for Countermeasure Against Power Analysis Attacks*, pp. 152–157. Springer, 2019.

[23] B. Hettwer, S. Gehrer, and T. Güneysu, "Profiled power analysis attacks using convolutional neural networks with domain knowledge," in *International Conference on Selected Areas in Cryptography*, pp. 479–498, Springer.

[24] A. Raychowdhury, "Machine learning in profiled side-channel attacks and low-overhead countermeasures," 2019.

[25] G. Yang, H. Li, J. Ming, and Y. Zhou, "Convolutional neural network based side-channel attacks in time-frequency representations," in *International Conference on Smart Card Research and Advanced Applications*, pp. 1–17, Springer.

[26] R. Giterman, M. Wicentowski, O. Chertkow, I. Sever, I. Kehati, Y. Weizman, O. Keren, and A. Fish, "Power analysis resilient SRAM design implemented with a 1% area overhead impedance randomization unit for security applications," in *ESSCIRC 2019 - IEEE 45th European Solid State Circuits Conference (ESSCIRC)*, pp. 69–72, Sep. 2019.

[27] D. Das, S. Maity, S. B. Nasir, S. Ghosh, A. Raychowdhury, and S. Sen, "High efficiency power side-channel attack immunity using noise injection in attenuated signature domain," in *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 62–67, May 2017.

[28] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *2007 44th ACM/IEEE Design Automation Conference*, pp. 9–14, IEEE, 2007.

[29] Y. Wen and W. Yu, "Machine learning-resistant pseudo-random number generator," *Electronics Letters*, vol. 55, no. 9, pp. 515–517, 2019.

[30] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525)*, pp. 176–179, June 2004.

[31] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and structural biotechnology journal*, vol. 13, pp. 8–17, 2015.

[32] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," *Nature Reviews Genetics*, vol. 16, no. 6, pp. 321–332, 2015.

[33] J. Gao, "Machine learning applications for data center optimization," 2014.

[34] P. Harrington, *Machine learning in action*. Manning Publications Co., 2012.

[35] P. N. Ramkumar, H. S. Haeberle, S. M. Navarro, A. A. Sultan, M. A. Mont, E. T. Ricchetti, M. S. Schickendantz, and J. P. Iannotti, "Mobile technology and telemedicine for shoulder range of motion: validation of a motion-based machine-learning software development kit," *Journal of shoulder and elbow surgery*, vol. 27, no. 7, pp. 1198–1204, 2018.

[36] J. Gu, Y. Liu, Y. Gao, and M. Zhu, "OpenCL caffe: Accelerating and enabling a cross platform machine learning framework," in *Proceedings of the 4th International Workshop on OpenCL*, pp. 1–5, 2016.

[37] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[38] X. Xu and W. Burleson, "Hybrid side-channel/machine-learning attacks on PUFs: A new threat?," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, 2014.

[39] B. Hettwer, S. Gehrer, and T. Güneysu, "Applications of machine learning techniques in side-channel attacks: a survey," *Journal of Cryptographic Engineering*, pp. 1–28, 2019.

[40] T. Iwase, Y. Nozaki, M. Yoshikawa, and T. Kumaki, "Detection technique for hardware Trojans using machine learning in frequency domain," in *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pp. 185–186, Oct 2015.

[41] M. Muehlberghuber, F. K. Gürkaynak, T. Korak, P. Dunst, and M. Hutter, "Red team vs. blue team hardware Trojan analysis: Detection of a hardware Trojan on an actual ASIC," in *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, (New York, NY, USA), Association for Computing Machinery, 2013.

[42] C. Wang, S. Zhao, X. Wang, M. Luo, and M. Yang, "A neural network Trojan detection method based on particle swarm optimization," in *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pp. 1–3, Oct 2018.

[43] Jun Li, Lin Ni, Jihua Chen, and E. Zhou, "A novel hardware Trojan detection based on bp neural network," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 2790–2794, Oct 2016.

[44] Y. Liu, Y. Jin, A. Nosratinia, and Y. Makris, "Silicon demonstration of hardware Trojan design and detection in wireless cryptographic ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1506–1519, 2016.

[45] X. Chen, L. Wang, Y. Wang, Y. Liu, and H. Yang, "A general framework for hardware trojan detection in digital circuits by statistical learning algorithms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 10, pp. 1633–1646, 2016.

[46] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *2008 IEEE International workshop on hardware-oriented security and trust*, pp. 51–57, IEEE, 2008.

[47] W. Yu, Y. Wen, S. Köse, and J. Chen, "Exploiting multi-phase on-chip voltage regulators as strong PUF primitives for securing iot," *Journal of Electronic Testing*, vol. 34, no. 5, pp. 587–598, 2018.

[48] E. Alon and M. Horowitz, "Integrated regulation for energy-efficient digital circuits," in *2007 IEEE Custom Integrated Circuits Conference*, pp. 389–392, Sep. 2007.

[49] Wonyoung Kim, M. S. Gupta, G. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in *2008 IEEE 14th International Symposium on High Performance Computer Architecture*, pp. 123–134, Feb 2008.

[50] W. Yu and S. Köse, "A voltage regulator-assisted lightweight AES implementation against DPA attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 8, pp. 1152–1163, 2016.

[51] Y. Lu, J. Jiang, and W.-H. Ki, "A multiphase switched-capacitor DC–DC converter ring with fast transient response and small ripple," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 2, pp. 579–591, 2016.

[52] Y. Lu, J. Jiang, W.-H. Ki, C. P. Yue, S.-W. Sin, U. Seng-Pan, and R. P. Martins, "20.4 a 123-phase DC-DC converter-ring with fast-dvs for microprocessors," in *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, pp. 1–3, IEEE.

[53] Z. He, M. Wan, J. Deng, C. Bai, and K. Dai, "A reliable strong PUF based on switched-capacitor circuit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 6, pp. 1073–1083, 2018.

[54] V. P. Yanambaka, S. P. Mohanty, and E. Kougianos, "Making use of manufacturing process variations: A dopingless transistor based-PUF for hardware-assisted security," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, pp. 285–294, 2018.

[55] M. Wan, Z. He, S. Han, K. Dai, and X. Zou, "An invasive-attack-resistant PUF based on switched-capacitor circuit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 8, pp. 2024–2034, 2015.

[56] W. Yu and S. Köse, "Implications of noise insertion mechanisms of different countermeasures against side-channel attacks," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE.

[57] M. S. Alkatheiri and Y. Zhuang, "Towards fast and accurate machine learning attacks of feed-forward arbiter PUFs," in *2017 IEEE Conference on Dependable and Secure Computing*, pp. 181–187, IEEE.

[58] X. Xu, A. Rahmati, D. E. Holcomb, K. Fu, and W. Burleson, "Reliable physical unclonable functions using data retention voltage of SRAM cells," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 903–914, 2015.

[59] C. Zhou, K. K. Parhi, and C. H. Kim, "Secure and reliable XOR arbiter PUF design: An experimental study based on 1 trillion challenge response pair measurements," in *Proceedings of the 54th Annual Design Automation Conference 2017*, p. 10, ACM.

[60] W. Yu and Y. Wen, "Leveraging balanced logic gates as strong pufs for securing iot against malicious attacks," *Journal of Electronic Testing*, pp. 1–13, 2019.

[61] S. Tao and E. Dubrova, "Temperature aware phase/frequency detector-basec RO-PUFs exploiting bulk-controlled oscillators," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pp. 686–691, IEEE.

[62] M. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor, "An aging-resistant RO-PUF for reliable key generation," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 3, pp. 335–348, 2015.

[63] W. Yu and S. Köse, "Time-delayed converter-reshuffling: An efficient and secure power delivery architecture," *IEEE Embedded Systems Letters*, vol. 7, no. 3, pp. 73–76, 2015.

[64] S. Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos, and Y. Xie, "Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach," in *Design, Automation and Test in Europe*, pp. 64–69, IEEE.

[65] M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and S. Devadas, "Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 37–49, 2014.

[66] S. Wei, J. B. Wendt, A. Nahapetian, and M. Potkonjak, "Reverse engineering and prevention techniques for physical unclonable functions using side channels," in *Proceedings of the 51st Annual Design Automation Conference*, pp. 1–6, ACM.

[67] W. Che, M. Martinez-Ramon, F. Saqib, and J. Plusquellic, "Delay model and machine learning exploration of a hardware-embedded delay PUF," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 153–158, IEEE.

[68] R. Hesselbarth, F. Wilde, C. Gu, and N. Hanley, "Large scale RO PUF analysis over slice type, evaluation time and temperature on 28nm xilinx FPGAs," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 126–133, IEEE.

[69] D. P. Sahoo, R. S. Chakraborty, and D. Mukhopadhyay, "Towards ideal arbiter PUF design on xilinx FPGA: A practitioner's perspective," in *2015 Euromicro Conference on Digital System Design*, pp. 559–562, IEEE.

[70] Y. Yao, M. Kim, J. Li, I. L. Markov, and F. Koushanfar, "ClockPUF: Physical unclonable functions based on clock networks," in *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 422–427, EDA Consortium.

[71] B. Chatterjee, D. Das, and S. Sen, "RF-PUF: IoT security enhancement through authentication of wireless nodes using in-situ machine learning," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 205–208, IEEE.

[72] N. Alimohammadi and S. B. Shokouhi, "Secure hardware key based on physically unclonable functions and artificial neural network," in *2016 8th International Symposium on Telecommunications (IST)*, pp. 756–760, IEEE.

[73] L. Santiago, V. C. Patil, C. B. Prado, T. A. Alves, L. A. Marzulo, F. M. França, and S. Kundu, "Realizing strong PUF from weak PUF via neural computing," in *2017 IEEE international symposium on defect and fault tolerance in VLSI and nanotechnology systems (DFT)*, pp. 1–6, IEEE.

[74] W. Yu and S. Köse, "Security implications of simultaneous dynamic and leakage power analysis attacks on nanoscale cryptographic circuits," *Electronics Letters*, vol. 52, no. 6, pp. 466–468, 2016.

[75] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti, "Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 2, pp. 355–367, 2009.

[76] Y. Wen, S. F. Ahamed, and W. Yu, "A novel puf architecture against non-invasive attacks," in *2019 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*, pp. 1–5, June 2019.

[77] G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle, "Machine learning in side-channel analysis: a first study," *Journal of Cryptographic Engineering*, vol. 1, no. 4, p. 293, 2011.

[78] T. Roche, V. Lomné, and K. Khalfallah, "Combined fault and side-channel attack on protected implementations of AES," in *International Conference on Smart Card Research and Advanced Applications*, pp. 65–83, Springer, 2011.

[79] C. Tokunaga and D. Blaauw, "Securing encryption systems with a switched capacitor current equalizer," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 23–31, 2009.

[80] W. Yu and S. Köse, "Exploiting voltage regulators to enhance various power attack countermeasures," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 2, pp. 244–257, 2016.

[81] W. Yu and Y. Wen, "Malicious attacks on physical unclonable function sensors of internet of things," in *2019 IEEE 28th North Atlantic Test Workshop (NATW)*, pp. 206–211, May 2019.

[82] U. Rührmair, J. Martinez-Hurtado, X. Xu, C. Kraeh, C. Hilgers, D. Kononchuk, J. J. Finley, and W. P. Burleson, "Virtual proofs of reality and their physical implementation," in *2015 IEEE Symposium on Security and Privacy*, pp. 70–85, IEEE.

[83] J. Tang, R. Karri, and J. Rajendran, "Securing pressure measurements using sensor-PUFs," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1330–1333, IEEE.

[84] K. Rosenfeld, E. Gavas, and R. Karri, "Sensor physical unclonable functions," in *2010 IEEE international symposium on hardware-oriented security and trust (HOST)*, pp. 112–117, IEEE.

[85] R. Yashiro, T. Machida, M. Iwamoto, and K. Sakiyama, "Deep-learning-based security evaluation on authentication systems using arbiter puf and its variants," in *International Workshop on Security*, pp. 267–285, Springer, 2016.

[86] O.-X. Standaert, E. Peeters, G. Rouvroy, and J.-J. Quisquater, "An overview of power analysis attacks against field programmable gate arrays," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 383–394, 2006.

[87] W. Yu and Y. Wen, "Efficient hybrid side-channel/machine learning attack on XOR PUFs," *Electronics Letters*, vol. 55, pp. 1080–1082(2), October 2019.

[88] R. Kumar and W. Burleson, "On design of a highly secure PUF based on non-linear current mirrors," in *2014 IEEE international symposium on hardware-oriented security and trust (HOST)*, pp. 38–43, IEEE, 2014.

[89] M. Kalyanaraman and M. Orshansky, "Novel strong PUF based on nonlinearity of mosfet subthreshold operation," in *2013 IEEE international symposium on hardware-oriented security and trust (HOST)*, pp. 13–18, IEEE, 2013.

[90] A. Vijayakumar and S. Kundu, "A novel modeling attack resistant PUF design based on non-linear voltage transfer characteristics," in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 653–658, IEEE, 2015.

[91] Y. Bai, L. Wu, X. Wu, X. Li, X. Zhang, and B. Wang, "PUF-based encryption method for ic cards on-chip memories," *Electronics Letters*, vol. 52, no. 20, pp. 1671–1673, 2016.

[92] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF modeling attacks on simulated and silicon data," *IEEE transactions on information forensics and security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[93] M. Hossain, S. Noor, and R. Hasan, "Hsc-iot: A hardware and software co-verification based authentication scheme for Internet of Things," in *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pp. 109–116, IEEE.

[94] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of puf-based "unclonable" rfid ics for anti-counterfeiting and security applications," in *2008 IEEE International Conference on RFID*, pp. 58–64, April 2008.

[95] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[96] D. Li and K. Yang, "A self-regulated and reconfigurable cmos physically unclonable function featuring zero-overhead stabilization," *IEEE Journal of Solid-State Circuits*, vol. 55, pp. 98–107, Jan 2020.

VITA

Yiming Wen
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529

Yiming Wen received the B.S. degree in safety engineering from Central South University, Beijing, China in 2013. In 2016, he receives the M.S. degree in University of South Florida in 2016. Begin from Fall 2017, he joins the Center for Cybersecurity Education and Research (**CCSER**) and pursue his Ph.D. degree in Department of Electrical and Computer Engineering in **Old Dominion University** (Norfolk, Virginia). He focuses on researches related to hardware security, including PUF designs, hardware Trojans, and machine learning attacks.