Trends and Challenges in Ensuring Security for Low-Power and High-Performance Embedded SoCs*

Parisa Rahimi[1], Amit Kumar Singh[1], Xiaohang Wang[2], Alok Prakash[3]
[1] University of Essex, [2] South China University of Technology, [3] Nanyang Technological University
[1] {*pr19863, a.k.singh* }*@essex.ac.uk,* [2]*xiaohangwang@scut.edu.cn,* [3]*alok@ntu.edu.sg*

***Abstract*—In recent years, security, power consumption, and performance have become the important issues in embedded SoCs' design. With the growing number of embedded devices for automotive electronic and electric vehicles, real-time systems, robotics, artificial intelligence, smart technologies, or telecommunication, it is highly likely that these systems will be exposed to attacks or threats. Therefore, it is not easy to implement the security measure of such devices, and it becomes challenging while considering the performance and power issues due to limited available computing resources and often operating on batteries. In this paper, we survey the weaknesses of the embedded SoCs and examine the attacks, power consumption, and performance more closely with the main focus on Physical and Side-Channel attacks, which have not been surveyed previously. Along with the current trends and challenges, upcoming trends and challenges are also elaborated. This paper is intended to help the researchers and system designers in gaining deep insight into designing secure, power-efficient, and high-performance embedded SoCs in the future.**

***Index Terms*—Embedded System, Physical and Side-channel attacks, Power Optimization, High performance, Multi-core systems**

I. Introduction

Embedded systems are developed as a special purpose computer system, which is designed to execute or perform one or more assigned functions, often with a real-time constraint [1]. Further, they have hardware and software implemented as an integrated system in which typically all necessary software functionalities and hardware components are built around a single chip, referred to as system-on-chip (SoC) based solutions [32]. The hardware components usually include a central processing unit (CPU), memory, secondary storage, and input/output ports, frequently alongside other processing elements such as a graphics processing unit (GPU) and even Neural-network Processing Unit (NPU) on the latest high-performance SoCs [49].

The security issue in embedded systems is not new but has quickly become one of the main concerns in every electronic device. There are several emerging factors such as dynamic environment and unverified software that makes embedded systems more vulnerable to attacks [2]. Additionally, it is an undeniable fact that with the advancement of technology, almost all the embedded systems are equipped with the capability of wired or wireless connectivity that further increases the possible risk of the attacks [19] [20].

Along with the security concerns, power consumption is also one of the key design constraints for embedded SoCs such as mobile phones, tablets, wireless sensors, and biomedical devices. The main reasons for analysing power consumption in these systems are due to heat dissipation, limited battery lifetime, size constraints, and costs. At the same time, the increasing demand to achieve high performance on such devices also adds to the design challenges, especially with tighter power budgets. Adding sophisticated security measures on top further exacerbates the power-performance concerns.

However, existing surveys do not consider security, power, and performance jointly, thereby missing the impacts of their confluence in embedded systems. For example, [50] and [51] consider only power, [52] considers only security, and [53] considers only power consumption and performance constraints. Hence, this paper examines several existing works that consider the security of embedded SoCs, while also focusing on the works that consider power and/or performance in tandem with the security challenges. To the best of our knowledge, this is the first paper examining the trends and challenges of embedded SoCs considering security, power, and performance jointly, and providing a future perspective.

The rest of the paper is organized as follows. Section II provides a classification of attacks on embedded SoCs. Sections III, IV, V, VI, and VII examine power analysis, electromagnetic, timing, fault injection, and covert channel attacks, respectively, while also examining respective defence mechanisms and their consideration of power and performance. In Section VIII, we provide envisioned upcoming trends and challenges, while summary and conclusions are inferred in section IX.

II. EMBEDDED SOCS ATTACKS

As shown in Figure 1, security attacks in embedded SoCs can be classified into 2 main categories [3]: A. physical & side-channel/lateral attacks and, B. logical/software-based attacks.

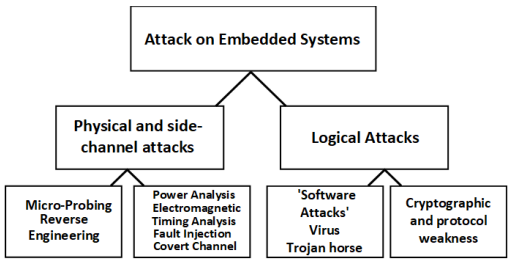

Fig. 1. Different Attacks on Embedded Systems.

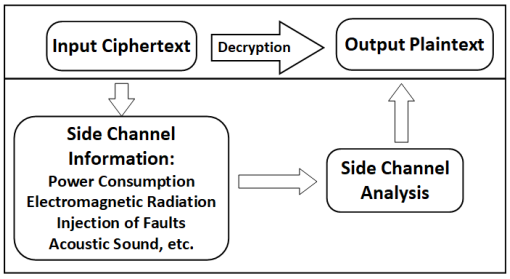

Fig. 2. Side Channel Attacks.

A. Physical and Side-Channel attacks

Physical and side-channel attacks refer to attacks that exploit the system performance and/or recognize properties of the performance [23]. The characteristic of the side-channel attacks is to find the weakness in the design implementation and try to exploit them. They are intended to monitor the properties of the system, while at the same time run the cryptographic operation [8]. Figure 2 demonstrates the side-channel attack process. The side-channel information, power consumption, or electromagnetic radiation can be analysed to identify secret information such as password [9].

B. Logical Attacks

Logical attacks typically involve sending messages to the device and then observing its behaviour. Typically, the adversary's intention is to trick the device to reveal its keys, access sensitive data, such as passwords, credit card numbers, or the entire identity of a victim's device [5] or to run malicious software [4]. Logical attacks can be classified into two categories: 1) software attacks and 2) cryptographic and protocol's weaknesses attacks. This weakness occurs due to the design and algorithm of the embedded systems.

1) Software attacks: Software attack is a general term that describes attacks launched by software agents like Trojan, viruses, or worms designed to compromise them. The intention of such an attack is to find the vulnerability in the system and then exploit the weakness that stems from flawed design and the algorithm for the embedded system. One of the reasons for frequent software attacks is that they are cheap to implement and do not require complicated infrastructures [21].

2) Cryptographic and protocol weakness attacks: Cryptographic and protocol weakness attacks exploit weakness of cryptographic algorithms, such as asymmetric ciphers, hashing algorithms, and symmetric ciphers, as well as protocol stacks. The main purposes of cryptographic attacks are to gather and access sensitive information or to break into a system [6].

The focus of this paper is physical and side-channel attacks, as it has not been surveyed previously. Under it, micro-proving and reverse-engineering attacks are made difficult as devices are coming sealed, but several other types of attacks have emerged which we have examined.

III. POWER ANALYSIS ATTACKS

Power analysis attacks refer to the analysis of the power consumption during the system's data analysis [10]. In embedded systems, the instantaneous power is dependent on the information and data that is being processed in the system, as well as the process performed by the system. Therefore, by analysing the power consumption of the system when it operates in encryption or decryption, the key can be deduced [11]. Power Analysis attacks can be classified into three categories, simple power analysis, differential power analysis, and correlation power analysis attacks [12].

A. Simple Power Analysis (SPA)

Simple power analysis (SPA) is a technique of side-channel attack that monitors the power consumption in real-time during the operation of cryptographic algorithms and ongoing processes. Power analysis attacks have become a serious security issue for cryptographic devices such as smart cards. As each operation has different power consumption profiles, it indicates the type of function that has been performed [12]. For instance, different functions performed by a microprocessor have different power consumption profiles that attackers analyse and evaluate to identify the executing tasks. Of course, the power consumption during each operation is different, and is dependent on the construction of the microprocessor [13].

B. Differential Power Analysis (DPA)

Differential Power Analysis (DPA) attacks are statistical techniques to exploit the power consumption to identify data information and to extract the secret keys while the cryptographic algorithm is running on the system. DPA attack uses the weakness by analysing the correlation between the power consumption in the chip of a smart card and the encryption key it contains [12]. Differential power attacks can happen in different sections of the chips [14]. DPA discloses few bits of the crypto at each time. This process is continually carried out till eventually, the entire key is known. Successful DPA attacks have been shown on a wide range of encryption algorithms. This attack is stronger than a simple power attack [12] [14].

C. Correlation power Analysis (CPA)

Correlation power analysis (CPA) attack is a new method for DPA. In this attack, the attacker models the power consumption during the analyses step [14]. CPA attack is a powerful analysis attack, capable of breaking a cipher by using a combination of the power consumption of the device and hamming weight of the key-dependent values of the algorithm. These are statistical attacks and use the Pearson correlation modules to correlate data or information [15].

D. Countermeasures against Power Analysis Attacks

Although countermeasures are implemented to prevent power analysis attacks, they have not been proved to be a 100% effective in preventing such attacks [25]. Hence, the main goal of a countermeasure is to make an attack very hard. A good countermeasure makes the time and cost of attacks prohibitively high, thereby making it harder for attackers to succeed [24]. As shown in Figure 3, countermeasures against the power analysis attack can be classified into two categories: hardware-based and software-based [25].

- Hardware-based countermeasures add extra hardware or change the components of the cryptosystem to reduce leakage of sensitive information and data as power [25].

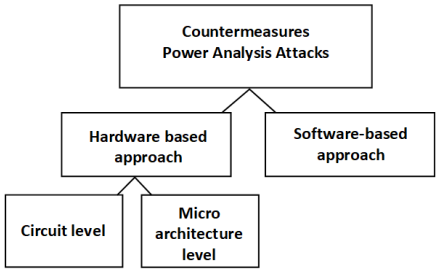

Fig. 3. Classification of countermeasures against power analysis attacks.

Hardware countermeasures can be categorized into two broad categories: circuit level and microarchitecture level countermeasures [15]. Circuit level countermeasures include adding additional components to the circuits or making some changes to the circuits at a macro level [24]. For instance, adding a source of noise or connecting a power line filter to the cryptographic device. Microarchitecture level countermeasures include adding additional components or making some changes at the logic gate level [13] [26].

- Software-based countermeasures, as the name suggests, make modifications to the program or data code implemented on the microcontrollers of the cryptographic systems. For instance, countermeasures for changing the pattern of power consumption may add some wrong instructions randomly inside the encryption process [25].

E. Security, Power consumption and Performance Metrics in Power Analysis Attacks

Table 1 lists the references considering power analysis attacks.In a few studies have been trying to consider just countermeasure, regardless of performance or power consumption, for example, Mutual Information Analysis (MIA), which is a concept presented in the information hypothesis that measures the relationship between two random variables [10].

It is evident that most researchers follow up the power consumption, for instance, random instruction injection. Random instruction injection destructs the alignment of power traces with the aim of making the attacks difficult. The result has shown that when one random instruction has injected, the number of power traces at which the key becomes significant has increased up to more than 200 [11]. Masking and hiding the true power trace method is a popular mitigation method used against the power analysis attack. In this countermeasure, the moderate value is hidden with some random values. In this method, an attack has been successfully recovered 64-bit secret key with 256 key searches and 300 power traces [15] such as data masking, table masking, window method, arithmetic, and Boolean masking [14] [15] [24] [25], or cognitive artificial intelligence known as knowledge growing system (KGS). This method is the basis for creating an agent that has the capability to think and act rationally like a human, which is known as the cognitive agent [12]. All these countermeasures [12] [14] [15] [24] [25] have tried to decrease the direct relationship between power consumption and the data, because they directly can affect power consumption. So, they are considering the changes in power consumption, regardless of its performance.

TABLE I
CONSIDERATION OF POWER ANALYSIS ATTACKS IN EXISTING RESEARCH.

Reference	*Attack*	*Defense*	*Performance*	*Power Consumption*
[10]	✓	✓	×	×
[11]	✓	✓	×	✓
[12]	✓	✓	×	✓
[13]	✓	✓	✓	✓
[14]	✓	✓	×	✓
[15]	✓	✓	×	✓
[24]	✓	✓	×	✓
[25]	✓	✓	×	✓
[26]	✓	✓	×	✓

In some cases, the implementation of a countermeasure decreases the performance of the devices due to the extra involved overhead, for instance, Dynamic Voltage and Frequency Scaling (DVFS) aware countermeasure. DVFS-aware is introduced in most modern processors to decrease the power consumption by dynamically adjusting the voltage and frequency at runtime [13]. This DVFS-aware power analysis attack, compared to most existing power analysis attacks that are unaware of DVFS, has increased the accuracy of secret key extraction by 1-22%.

IV. ELECTROMAGNETIC ATTACKS

The Electromagnetic (EM) attack is a side-channel attack and is passive, which means that this attack can exploit the normal functioning of the target device without causing physical damage [16] [17]. Given their nature, the EM side-channel signals are a subgroup of the electromagnetic interference (EMI) produced by the device, they can be really challenging and hard to address [57] [58]. The primary premise of these attacks is that measuring the electromagnetic radiation emitted by a system can lead to exposing sensitive data. In case the deployment of such an attack has to be successful, it requires intimate knowledge of the SoC layout, and the electromagnetic radiation measurement needs to be run as it is in an isolated environment [45].

A. Countermeasures against Electromagnetic attacks

Many strategies, both in hardware and software, have been introduced as countermeasures against EM attacks [45]:

- **Hardware Modifications:** By adding an extra metal layer to the chip that contains the radiation, where it can be placing an active grid on the top of the chip to create more noise into the EM field leading to blurring and emanations [16]. Radiation can also be reduced by technology shrinking and using smaller transistors during chip production [45].
- **Software Modifications:** One of the possible countermeasures is to apply nonlinear key updates, so it can help to prevent attackers from being able to compare and measure the electromagnetic radiation with an encryption procedure [17]. Another possible solution is to make algorithmic changes to the cryptographic primitives so that attacks lead to provably ineffective on the established implementation, for instance, covering data and key with a random cover created at each run [46].

TABLE II
CONSIDERATION OF ELECTROMAGNETIC ATTACKS IN EXISTING RESEARCH.

Reference	*Attack*	*Defense*	*Performance*	*Power Consumption*
[16]	✓	✓	×	×
[17]	✓	✓	×	×
[45]	✓	✓	×	×
[46]	✓	✓	✓	✓
[54]	✓	✓	✓	✓

All these extra measurements have significantly reduced the possibility of the leakage of useful information via the EM side-channel, which will significantly raise the number of samples required for running a successful attack and making the attack impracticable [45]. Even so, it can be said that an attacker with infinite samples still would be able to perform EM attacks on the lower signal. However, there is a possibility that in the near future, the use of an alternative to semiconductor technology will take place over semiconductors as this helps to eliminate the leakage of the information [46].

B. Security, Performance and Power consumption of Electromagnetic Attacks

A brief comparison of different studies show that in most of them the main concern of researchers is to apply a good countermeasure for electromagnetic attacks to decrease the possible risk (Table II). For instance, Logical countermeasures are typically based on power balancing and they follow principles like gate-level masking [16]. Architectural countermeasures on the other hand present amplitude or time distortions to obfuscate the EM trace [45]. Circuit-level physical countermeasures include physical noise injection (NI) [16] [45], signal strength reduction and signal information reduction [17]. Methodologies for signal strength reduction rely on circuit redesign to decrease unintentional emanations. Signal information reduction techniques rely on applying randomization and/or frequent key refreshing within the computation, to significantly decrease the effectiveness of statistical attacks on the available signals [17].

Applying the Electromagnetic Attacks countermeasure, such as the Electromagnetic Equalizer (EME), which is an active countermeasure, relies on the result of adjusting the power grid (PG) capacitor to equalize the EM profile to decline the EM leakage. To verify the EME technique, several cryptographic designs have been implemented. In [46], security is improved with area and power overheads of 0.62% and 1.36%, respectively. Furthermore, the well known EM leakage from digital logic is accidentally combined with the radio carrier, which is amplified and then transmitted by the antenna. Typically, when an RF circuit is placed close to the digital circuitry, information and data on the digital circuit's operation leaks into the RF circuit and is broadcast along with regular transmissions [54]. These countermeasures [46] [54] can affect other aspects of the devices, such as power consumption, because electromagnetic and power depend on similar physics, so these studies have tried to consider security, power consumption and performance as well.

V. TIMING ATTACKS

Timing attacks refer to attacks that evaluate the time used by the computing systems during the cryptographic operations, timing measurement and analysis. The security-related process may take different times since it depends on the values of the input information and the secret key [18]. Timing analysis attacks classified into keystroke and (Secure Shell Protocol) SSH timing analysis attacks [10]. Generally discovered sources of timing weakness include:

- Data dependent differences in instruction times.
- Early exit.
- Data dependent code branches.
- Cache access times [18].

A. Countermeasure against Timing attacks

As shown in Figure 4, the countermeasures against the timing attacks can be classified into three categories.

- Application Level Countermeasures.
- Hardware Countermeasures.
- Operating System Countermeasures [27].

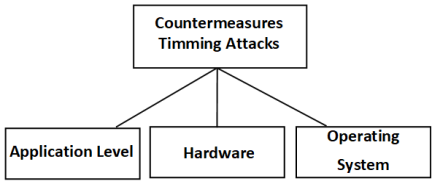

Fig. 4. Classification of countermeasures against Timing attacks.

- **Application Level Countermeasures:** To overcome timing-attacks, countermeasures against the crypto applications are typically very promising and easy to apply and also provide a sufficient level of security. But upon closer inspection, they suffer from few shortcomings, for instance, application-level countermeasures are heavy and not practically sufficient [27]. In addition, for patching the application all of them require to be patched with the same countermeasure, necessitating designers to modify several applications with these bulky countermeasures [28]. This has a huge impact on system performance and energy consumption. On the other hand, this countermeasure method provides a fast and quick fix to stop as well as prevent the timing attacks [43] [44].
- **Hardware Countermeasures:** With a slight modification in the hardware, it can make the attacks less successful and more difficult to happen again, with a little extra effect on the system performance [27].
- **Operating System Countermeasures:** Since the hardware countermeasures are expensive to implement, and application layer countermeasures have a huge overhead, the search is still on for a countermeasure that can be effective and address both issues [28]. Operating system countermeasures may be the best candidate, that use run time monitoring of the system performance using Phasor Measurement Units (PMUs) [18]. For instance, adding some random delays or normalizing all timings to a fixed value, or limiting the frequency of reading the time counter [43] [44].

B. Security, Performance and Power consumption of Timing Attacks

Table 3 shows different aspects of considerations in timing attacks. For example, adding random delays. To counter timing attacks contain adding a random delay that needs an attacker to obtain more timing measurements or performing a fixed response time for all server responses [43], in this methodology has tried to consider both the performance and power consumption along with countermeasure against. To decrease the performance impact of this countermeasure, it has been suggested to just keep the implementations of sensitive processes fixed or adding an unpredictable and fixed delay to the server's response time [43].

Power consumption is considered in a few studies, such as changes in the behaviour of Advanced Encryption Standard (AES) [18]. In cryptographic terms, a method is used to change the behaviour of the underlying AES. This method is a powerful way to security and change the behaviour of encryption in the face of data errors.

In some approaches, performance is examined along with security, for example, countermeasures applied in the hardware however, it can make attacks more challenging with overhead in the performance, like non-cached memory accesses, hardware prefetching, fuzzying clocks, and countermeasures in the operating system [28]. Application level countermeasures, countermeasures in the crypto application is generally easy to implement and prepare sufficient levels of security against timing attacks, such as Countermeasures involving look-Up tables, data-oblivious memory access pattern and constant and random time implementations [55].

Further, in some methodologies such as adopting timing-invariant implementation of the algorithms in the hardware and the software [27] as well as avoid extra reductions [44], power consumption and performance have been disregarded.

TABLE III
CONSIDERATION OF TIMING ATTACKS IN EXISTING RESEARCH.

Reference	*Attack*	*Defense*	*Performance*	*Power Consumption*
[18]	✓	✓	×	✓
[27]	✓	✓	×	×
[28]	✓	✓	✓	×
[43]	✓	✓	✓	✓
[44]	✓	✓	×	×
[55]	✓	✓	✓	×

VI. FAULT INJECTION ATTACKS

Fault injection can be identified by false activation and damage to the hardware module, which leads to an output pattern for the given input to be obtained. In a fault injection attack, a number of variants are involved, which categories into external parameters and environmental conditions [10]. These are the supply voltage, clock, temperature, radiation, and many more to include errors on its components. The outcome of such attacks could be taken advantage of by malicious users to corrupt and damage the secure or non-secure code or information stored on the components, such as memories. Another benefit of the fault injection technique can be used as a pre-cursor to software attacks [29] [39].

A. Countermeasure against Fault injection attacks

The very simple countermeasure against fault injection attacks is to execute twice the same algorithms of the cryptographic for each input, and then check if the results match [29]. This method may be continued to triplicate the implementation of the algorithm in order to get single-error correction capabilities by using the majority voting to the results [39] [40]. The reality is that a motivated attacker can overcome these countermeasures via injecting two weaknesses, one for each implementation of the algorithm. With the recent fault injection methodologies which are currently available, it can influence faults with exact timing while the running of the software routine [41]. In fact, it may be able to inject faults with a single clock cycle in different implements of the same algorithms, although it maybe practically impossible to inject faults in further instruction with a wide range of the fault injection methods which are currently available [42].

B. Security, Performance and Power consumption of Fault Injection Attacks

As table 4 shows, in fault injection attacks a few studies have been done regarding all aspects of devices, such as countermeasure, performances, and power consumption. Implementing some countermeasure against, such as temporal redundancy, software-implemented fault tolerance (SWIFT) techniques, which is a compiler-based transformation that duplicates the instructions in a program and adds comparison instructions at strategic points in the process of code generation [29], restructuring of the algorithm [39]. Hardware-Level which protections are implemented by the chip manufacturer and can be further classified into two categories: **Active protection** such as Light detectors, Frequency detectors, and **Passive protection** for instance Bus and memory encryption, and Microcode-Level Countermeasures, this is a kind of software countermeasures that presented when hardware countermeasures are not sufficient or as cautious protection against future attack methods that may defeat present-generation hardware countermeasures, (e.g. Disabling MSR Interface, Scaling Back Voltage during Enclave Operation, Limiting to Known Good Values, and Multi-variant Enclave Execution), as well as, Software-Level Hardening, e.g. Fault-Resistant Cryptographic Primitives, Application and Compiler Hardening, and Traditional Memory Safety Hardening [41], has tried to decline the possible effect of defence strategies on performances regardless of its power consumption.

In some studies, researchers did not consider power consumption and performance within countermeasures [40] [42]. countermeasure based on on-chip voltage regulators has been suggested where an interleaved switched-capacitor (SC) voltage converter is used, and individual steps are turned on and turned off based on the workload information. For instance, Hardware Trojans that are based on the modification of a logic gate's electrical characteristics in a way that the metal, polysilicon layer, and active area remain unchanged. Trojan has triggered with a rather low probability of $\leq 10^{-6}$ [59], 10% Vdd reduction [40]. In the field of IC security, the typical measures to the protection against fault injection at-

TABLE IV
CONSIDERATION OF FAULT INJECTION ATTACKS IN EXISTING RESEARCH.

Reference	*Attack*	*Defense*	*Performance*	*Power Consumption*
[29]	✓	✓	✓	×
[39]	✓	✓	✓	×
[40]	✓	✓	×	×
[41]	✓	✓	✓	×
[42]	✓	✓	×	×

tacks include the hardware/time redundancy method, physical protection, and the error detection codes (EDC)-based technique. EDC methods are mainly used to transmit data over a communication channel [42]. EDC algorithm, when applied to a sequence of characters in a data packet or message, returns a number that can be used to check the validity of the data stream by another process, e.g. parity check [42].

VII. COVERT CHANNEL

The mechanism which the covert channel uses is not built for communication, like writing and checking if a file is protected to convey 1 or 0. The covert channel uses an insider process that leaks sensitive information to an outsider process [30] [31]. The covert channels can compromise chip-level security. In many communication media seen the covert channel such as timing, heat, or indistinct sound [60] [61]. The heat transfers which is known as thermal covert channels (TCC) can be particularly dangerous [33]. Thermal covert channels spatially can be traced in multi-core systems. Such as any communication system, a thermal cover channel contains a receiver and transmitter [34]. In the transmitter program, the temperature signals produced from sensitive data or information such as user passwords. It can move over the chip by heat transfer among processor cores, finally, the thermal signals reach the destination (receiver) [34]. While thermal signals transmitted with a thermal covert channel throughout a chip, it is highly likely to be degraded or disrupted by the environment temperature variations or the thermal noise [33]. An advanced countermeasure against the thermal covert channel needs to check and scan the signal frequency spectrum, to find positioning affected cores and blocking them [33] [35].

A. Countermeasure against Covert Channel attacks

Before any action to countermeasure against a covert channel, it needs to identify. A covert channel can be identified based on a formal method. To identify covert channels in systems, a wide range of formal methods have been developed. The covert channel happens for two reasons: weaknesses in the system design and design oversights. When covert channels happen because of oversights, they can be fixed once it discovered, but those main to the system can never be eliminated without the system redesigning [36]. If a covert channel was not fixed in the design stage, the best next option is to remove its possible utilize. However, it may cause a very ineffective system, while covert channels often removed entirely just by replacing automated methods with manual methods. If a covert channel cannot be removed, its size or capacity should be decreased [37]. Those capacities can be tolerable depending on the amount of leak-sensitive information. For instance, if the channel is very small so the critical data and information cannot be leaked before it is expired, the capacity of the channel is acceptable. Increasing the channel capacity can lead to some problems because it indicates introducing noise or slowing down system mechanisms, which both cause the limited system's performance [37] [38].

TABLE V
CONSIDERATION OF COVERT CHANNEL ATTACKS IN EXISTING RESEARCH.

Reference	*Attack*	*Defense*	*Performance*	*Power Consumption*
[30]	✓	✓	×	×
[31]	✓	✓	×	×
[33]	✓	✓	✓	×
[34]	✓	✓	✓	×
[35]	✓	✓	✓	×
[36]	✓	✓	✓	×
[37]	✓	✓	✓	×
[38]	✓	✓	✓	×

B. Security, Performance and Power consumption of Covert channel Attacks

As table 5 indicates, in covert channel attacks most of the methodologies have been considered performances along with countermeasure against, for instance, DVFS. This technique has confirmed that on average 98% of the TCC attacks can be detected, and the bit error rate of a TCC attack can increase to 92%. The performance penalty caused by the inclusion of the proposed countermeasures is only 3% for an 8×8 system [33]. Dynamic frequency scaling (DFS), which supported by the CPUs where their operating frequency can be changed based on the workload, DFS techniques have demonstrated the feasibility and the potential of achieving a high bandwidth of up to 20 bits/second on Intel Xeon server X5560 [34], countermeasure with scanning and channel-aware jamming [35], the Basic Protocol (BP), in this technique the sender and the receiver must be both ready for the transmission, the main problem of this method is that the receiver sends no feedback to the sender, so the sender doesn't identify whether the receiver has got the information. The Two-Channel Transmission Protocol (TCTP), which aims to improve the deficiencies in BP, the Self-Adaptive Protocol (SAP). SAP means the protocol itself can recognize interference and adjust itself to deliver the information correctly [36]. Duty-Cycle- based on encoding Scheme, e.g. **Single-bit, Single-channel**, in this method, only one frequency has been used in a thermal covert channel at one time, on average, the transmission rate of the proposed thermal covert channel is 30% better than that in. [37] Random traffic padding to reduce the channel capacity [38].

Furthermore, in a few countermeasure implementations, performance has not been considered, such as resource partitioning, isolation techniques for multi-core platforms, that are based on resource partitioning [30], and Ping Tunnel and Smart Covert Channel Tool (SCCT), this method is able of building a hidden overlay network and prepares dynamic overlay routing based on optimized link-state routing [31].

In all of these studies, power consumption has been ignored.

VIII. UPCOMING TRENDS AND CHALLENGES

Embedded SoCs are prevailing in many application domains such as automotive, robotics, and Internet of Things (IoT), and are expected to have increasing penetration in several domains. Considering the widespread adoption of embedded SoCs and their security concerns related to physical and side-channel attacks, we anticipate the following upcoming trends and challenges.

A. Increasing Physical and Side-Channel attacks and defence mechanisms

With the fear of reverse engineering to perform physical and side-channel attacks, devices equipped with embedded SoCs are now coming sealed to make such attacks difficult. However, attackers have been able to come with new types of physical and side-channel attacks, such as covert channel attacks. With the increasing penetration of embedded SoCs of various types in several application domains, e.g. IoT, Industry 4.0 and automotive, it is expected that attackers will continue to exploit unseen vulnerabilities during design time. It could be an enhanced version of an existing attack or even a previously unseen attack. With increasing attack types and their variations, there will be a need to develop relevant defence mechanisms to guard the whole ecosystem of devices equipped with embedded SoCs. The development of defence mechanisms to encounter a variety of attacks will be challenging, as several aspects of information leakage has to be considered.

B. Joint consideration of security, power and performance for defence mechanisms

A robust defence mechanism can be heavy and power-hungry. Therefore, exploration of light-weight mechanisms is expected, as the devices may often have resource and battery constraints. Based on the observations in Sections 3, 4, 5, 6 and 7 that researchers have now started considering security, power, and performance aspects jointly due to the importance of the three metrics, it is expected that such a joint consideration will be widely done in the future. Since there is always a trade-off amongst these metrics and the design space can be huge, it is challenging to find the best design point that can satisfy the security, power, and performance requirements.

C. Hardware- and software-based security

As started earlier in Section 2.2, security solutions for physical and side-channel attacks are mainly based on monitoring SoC properties and then analysing them to perform an attack. The analysis is typically implemented in the form of software tools. With the increasing use of embedded SoCs in various application domains, the enhancement on the software tooling side is expected to continue. In addition, it is expected to have hardware-based solutions as well, though these are costly. These observations indicate that the development of hardware- and software-based defence mechanisms are going to continue hand-in-hand. Further, based on the anticipated increase in attacks in future and current advances in defence solutions, we anticipate that security solutions will jointly consider software and hardware in order to achieve the best protection mechanism at a minimal cost. This requires addressing the challenges of identifying the defence components to be run on the hardware and software. It may also bring the notion of adaptation of the components over the hardware and software depending upon different scenarios. This will require addressing the challenges of identifying adaptation instances and performing them efficiently.

D. Attacks and defences for AI-based solutions

The widespread adoption of AI algorithms will enable their deployment in embedded SoCs. Therefore, it is expected to have new and/or enhanced physical and side-channel attacks focused on AI algorithms [56]. With the increase in attack mechanisms, the development of novel defence mechanisms is expected while addressing the involved unforeseen challenges.

IX. CONCLUSION

In this survey paper, we have examined the different security techniques. These techniques based on different approaches such as performance, and power consumption of embedded SoCs were explored. It is obvious that most of the existing works have tried to implement a good countermeasure to fix the possible risk of attacks regardless of the performance or power consumption of embedded SoCs. Many researchers have considered only performance, or power. Several recent research activities consider both power and performance. However, some countermeasures may have affected other aspects of the system, including performance and power consumption.

A close observation of existing works indicate that security, power and performance are considered jointly by only a very few researchers as it is very challenging to consider such a multi-objective optimisation problem and achieve the best results. As such, we have explored security techniques related to each of these major SoCs. We also provide upcoming trends and challenges to be considered to take the physical and side-channel attacks on embedded systems into the next era.

REFERENCES

[1] Jebril et al. Review of Challenges in Embedded Systems Design: Robustness, Predictability, Security. *International Journal of Current Research in Embedded System VLSI Technology* **2017**, *2(1)*, 1–6.

[2] Brown, M. Five Steps to Improving Security in Embedded Systems. *Contemporary Technologies in Automation* **2013**, *3*, 14–22.

[3] Huang et al. A security embedded system base on TCM and FPGA. *2nd IEEE international conference on computer science and information technology* **2009**, 605–609.

[4] Zurawski, R. (Ed.). *Embedded Systems Handbook: Embedded systems design and verification*, **2018**, *6*

[5] Henzinger et al. A survey of side-channel attacks on caches and countermeasures. *Journal of Hardware and Systems Security* **2018**, *2(1)*, 33–50.

[6] Papp et al. Embedded systems security: Threats, vulnerabilities, and attack taxonomy. *13th Annual Conference on Privacy, Security and Trust (PST), IEEE* **2015**, 145–152.

[7] Gashi et al. Diversity for safety and security in embedded systems.*the IEEE International Conference on Dependable Systems and Networks* **2014**.

[8] Rosenberg, J. Security in embedded systems. *Rugged Embedded Systems: Computing in Harsh Environments* **2016**, *3*, 149–205.

[9] An et al. Extension implementation of tcm in the embedded system based on fpga. *In 2013 International Conference on Computer Sciences and Applications, IEEE* **2013**, 749–752.

[10] Ambrose et al. Side Channel Attacks in Embedded Systems: A Tale of Hostilities and Deterrence. *Sixteenth International Symposium on Quality Electronic Design, IEEE* **2015**, 452–459.

[11] Gamaarachchi et al. Power analysis based side channel attack. *arXiv preprint arXiv* **2018**, *1801.00932*, 1–101.

[12] Putra et al. Power analysis attack on implementation of DES. *IEEE ICITSI* **2016**, 1–6.

[13] Lo et al. Power analysis attacks on the AES-128 S-box using differential power analysis (DPA) and correlation power analysis (CPA). *Journal of Cyber Security Technology* **2017**, *1(2)*, 88–107.

[14] Chai et al. A New Power Analysis Attack and a Countermeasure in Embedded Systems. *9th IEEE UEMCON* **2018**, 645–652.

[15] Rai et al. Correlation power analysis and effective defense approach on light encryption device block cipher. *Security and Privacy* **2019**, *2(5)*, e87.

[16] Khan et al. Malware detection in embedded systems using neural network model for electromagnetic side-channel signals. *Journal of Hardware and Systems Security* **2019**, *3(4)*, 305–318.

[17] Sastry et al. Counter Attacking Electromagnetic attacks for securing Embedded Systems. *International Journal of Advances in Science and Technology,* **2012**, *5(1)*, 63–68.

[18] Bogdanov et al. Differential cache-collision timing attacks on AES with applications to embedded CPUs. *In Cryptographers' Track at the RSA Conference, Springer* **2010**, 235–251.

[19] Hanawa et al. Low-power and high-performance communication mechanism for dependable embedded systems. *International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems, IEEE* **2008**, 67–73.

[20] Kartha et al. A Review Paper on Low Power Embedded System Design. *Research Gate* **2014**, 1–6.

[21] Gupta, M. Security challenges and cryptography in embedded systems. *International Journal of Advance Research In Science And Engineering.* **2015**, *4(1)*, 656–662.

[22] Van et al. *Encyclopedia of cryptography and security*; Springer, Boston, MA, **2014**.

[23] Papp et al. Embedded systems security: Threats, vulnerabilities, and attack taxonomy. *13th Annual Conference on Privacy, Security and Trust (PST), IEEE* **2015**, 145–152.

[24] Prouff, E. (Ed.). *Constructive side-channel analysis and secure design*; Springer International Publishing; **2014**.

[25] Bucek, J.; Novotný, M. Differential power analysis under constrained budget: Low cost education of hackers. *Euromicro Conference on Digital System Design, IEEE* **2013**, 645–648.

[26] Hori et al. Evaluation of physical unclonable functions for 28-nm process field-programmable gate arrays. *Journal of Information Processing* **2014**, *22(2)*, 344–356.

[27] Gupta et al. Security and Cryptography. *In Software Engineering for Embedded Systems.* **2019**, 501–547.

[28] Rebeiro et al. Timing channels in cryptography: a micro-architectural perspective, *Springer* **2014**.

[29] Moro et al. Experimental evaluation of two software countermeasures against fault attacks. *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* **2014**, 112–117.

[30] Masti et al. Thermal covert channels on multi-core platforms. *24th USENIX Security Symposium (USENIX Security* **2015**, 865–880.

[31] Kaur et al. Countermeasures for covert channel-internal control protocols. *International Conference on Availability, Reliability and Security, IEEE.* **2015**, 422–428.

[32] APereira et al. Towards a metamodel for a requirements engineering process of embedded systems. *VI Brazilian Symposium on Computing Systems Engineering (SBESC), IEEE* **2016**, 93–100.

[33] Huang et al. Countermeasures against the thermal covert channel attacks targeting many-core systems. *IEEE Design Automation Conference (DAC), IEEE* **2020**, 1–6.

[34] Alagappan et al. DFS covert channels on multi-core platforms. *IFIP/IEEE International Conference on Very Large Scale Integration, VLSI-SoC* **2017**, 1–6.

[35] Wang et al. Combating Enhanced Thermal Covert Channel in Multi-/Many-Core Systems With Channel-Aware Jamming. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **2020**, *39(11)*, 3276–3287.

[36] Lin et al. Robust and efficient covert channel communications in operating systems: design, implementation and evaluation. *IEEE International Conference on Software Security and Reliability Companion* **2013**, 45–52.

[37] Long et al. Improving the efficiency of thermal covert channels in multi-/many-core systems. *Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE* **2018**, 1459–1464.

[38] Vateva-Gurova, T.; Suri, N. On the Detection of Side-Channel Attacks. *IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)* **2018**, 185-186.

[39] Barenghi et al. Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. *Proceedings of the IEEE* **2012**, *100(11)*, 3056–3076.

[40] Kumar et al. Parametric trojans for fault-injection attacks on cryptographic hardware. *Workshop on Fault Diagnosis and Tolerance in Cryptography, IEEE* **2014**, 18–28.

[41] Murdock et al. Software-based fault injection attacks against Intel SGX. *IEEE Symposium on Security and Privacy (SP)* **2020**, 1466–1482.

[42] Liu et al. Fault injection attack on deep neural network. *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* **2017**, 131–138.

[43] Wu et al. Eliminating timing side-channel leaks using program repair. *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis* **2018**, 15–26.

[44] Schindler, W. Exclusive exponent blinding may not suffice to prevent timing attacks on RSA. *International Workshop on Cryptographic Hardware and Embedded Systems,Springer* **2015**, 229–247.

[45] Selvaraj et al. Electromagnetic induction attacks against embedded systems. *Asia Conference on Computer and Communications Security* **2018**, 499–510.

[46] Wang et al. Electromagnetic equalizer: an active countermeasure against EM side-channel attack. *The International Conference on Computer-Aided Design.* **2018**, 1–8.

[47] Whitnall, C.; Oswald, E. A critical analysis of ISO 17825 ('Testing Methods for the mitigation of non-invasive attack classes against cryptographic modules'). *In International Conference on the Theory and Application of Cryptology and Information Security, Springer, Cham.* **2019**, 256–284.

[48] Feng et al. Effects of non-invasive brain stimulation on headache intensity and frequency of headache attacks in patients with migraine. *A systematic review and meta-analysis. Headache: The Journal of Head and Face Pain* **2019**,*59(9)*, 1436–1447.

[49] Dutt et al. Toward smart embedded systems: A self-aware system-on-chip (soc) perspective. *ACM TECS* **2016**,*15(2)*, 1–27.

[50] Singh et al. Dynamic energy and thermal management of multi-core mobile platforms: A survey. *IEEE Design & Test*, **2020**, *37(5)*, 25–33.

[51] Ehret et al. A Survey on Hardware Security Techniques Targeting Low-Power SoC Designs.*IEEE HPEC*, **2019**, 1–8.

[52] Manifavas et al. Embedded systems security: a survey of EU research efforts. *Security and Communication Networks* **2015**, *8(11)*, 2016–2036.

[53] Paulino et al. Improving Performance and Energy Consumption in Embedded Systems via Binary Acceleration: A Survey. *ACM Computing Surveys (CSUR)* **2019**, *53(1)*, 1–36.

[54] Camurati et al. Screaming channels: When electromagnetic side channels meet radio transceivers. *ACM SIGSAC Conference on Computer and Communications Security* **2018**, *53(1)*, 163–177.

[55] Van Goethem et al. The clock is still ticking: Timing attacks in the modern web. *ACM SIGSAC Conference on Computer and Communications Security* **2015**, 1382–1393.

[56] Shafique et al. Robust machine learning systems: Challenges, current trends, perspectives, and the road ahead. *IEEE Design & Test* **2020**, *37(2)*, 30–57.

[57] Sangodoyin, S., et al. Side-Channel Propagation Measurements and Modeling for Hardware Security in IoT Devices. *IEEE Transactions on Antennas and Propagation* **2020**, *69(6)*, 3470–3484.

[58] Werner, F.T., et al. An Efficient Method for Localization of Magnetic Field Sources That Produce High-Frequency Side-Channel Emanations. *IEEE Transactions on Electromagnetic Compatibility* **2021**, *14*, 1–13.

[59] Zhao, Y., et al. On hardware-trojan-assisted power budgeting system attack targeting many core systems. *Journal of Systems Architecture* **2020**, *109*, 101757.

[60] Huang, H., et al. Detection of and Countermeasure against Thermal Covert Channel in Many-core Systems. *IEEE TCAD* **2021**.

[61] Wang, J., et al. Combating Enhanced Thermal Covert Channel in Multi-/Many-Core Systems With Channel-Aware Jamming. *IEEE TCAD* **2020**, *39(11)*, 3276–3287.