HARDWARE ATTACK MITIGATION TECHNIQUES ANALYSIS

Samer Moein¹, T. Aaron Gulliver¹, Fayez Gebali¹ and Abdulrahman Alkandari²

¹Electrical and Computer Engineering, University of Victoria, BC, Canada

²Department of Computer, Public Authority for Applied Education and Training, Kuwait City, Kuwait

ABSTRACT

The goal of a hardware attack is to physically access a digital system to obtain secret information or modify the system behavior. These attacks can be classified as covert or overt based on the awareness of the attack. Each hardware attack has capabilities as well as objectives. Some employ hardware trojans, which are inserted during, manufacture, while others monitor system emissions. Once a hardware attack has been identified, mitigation techniques should be employed to protect the system. There are now a wide variety of techniques, which can be used against hardware attacks. In this paper, a comprehensive survey of hardware attack mitigation techniques is presented. These techniques are matched to the hardware attacks and attack criteria they can counter, which helps security personnel choose appropriate mitigation techniques to protect their systems against hardware attacks. An example is presented to illustrate the choice of appropriate countermeasures.

KEYWORDS

Hardware Attack, Hardware Attack Mitigation, Hardware Security, Covert Attack & Overt Attack

1. INTRODUCTION

The use of semiconductor devices in military, financial, economic, and other critical infrastructure has raised significant concerns regarding hardware security. A victim is unaware of the occurrence of a covert attack but may have knowledge of an overt attack. Overt hardware attacks [1] such as deprocessing and reverse engineering are employed to reveal device functionality in order to steal information and copy devices. Further, some overt attacks introduce hardware trojans [2] by modifying Integrated Circuits (ICs) to create abnormal system behaviour, while others monitor system emissions to obtain information. The increasing sophistication of hardware attacks as well as the growing chip complexity makes hardware security a major challenge for the semiconductor industry [3–6].

The design and manufacture of an IC involves multiple processes. These provide numerous opportunities for attacks, and mitigation techniques must be developed to counter them. Figure. 1 shows the approaches to both attacking and defending a chip. Overt attacks, i.e. reverse engineering, deprocessing, and microprobing, allow an attacker to examine the internal structure of a chip. This information can be used to identify chip vulnerabilities for covert at-tacks, i.e.

power, timing, and electromagnetic, or to copy the chip. An attacker can also insert a hardware trojan into a chip to allow an attack to be initiated.

A defender can use destructive techniques to check if malicious modifications have been made to a chip. However, this approach is time consuming and requires significant resources, so it is not practical to examine a large number of chips. In practice, defenders rely on non-destructive techniques to determine if a chip is working probably [7–9]. These techniques can be employed during testing and/or chip operation. Hardware attack mitigation techniques are used to protect a chip during both chip design and operation. These techniques can be used to produce a secure chip when it is being designed. Further, if unexpected behaviour is detected during chip operation, they can be employed to counter any attacks.

The knowledge, skill and resources that modern attackers possess enable them to introduce modifications into the design during the IC life cycle. Many of these modifications are not detected during the testing and deployment phases [4, 10, 11]. Developing mitigation techniques against these malicious attacks begins with their identification and classification. Hardware attacks can be classified as covert or overt [1, 2]. They can also be classified based on the accessibility, resources, and time required for implementation [50]. The classification can be used to determine the system requirements to defend against attacks [12, 13].

Hardware attack mitigation techniques can be divided into two categories, those designed to counter multiple attacks and those developed for single attacks. A number of approaches have been used to counter multiple attacks. Hiding techniques are based on reducing the signal strength or increasing the noise level [53]. Masking techniques make it difficult for an attacker to determine the relationship between chip emissions and the corresponding data or operations [16–19]. Random noise can be employed to decrease the Signal-to-Noise Ratio (SNR) of IC signals [54], and make emissions more independent of the chip operations [53, 55]. Chip emissions can also be masked using asynchronous logic gates [56, 57], or reduced by using low power design techniques [53]. Further, emissions between chip regions can be lowered via design partitioning [20, 21]. Restricting chip access using anti-tampering techniques can prevent an attacker from collecting chip data [22, 23]. Moreover, emission filtering can be used to reduce data leakage [67]. These techniques can be used to counter most covert attacks, as these attacks typically monitor chip emissions. Sensors can also be deployed around a chip to detect anomalies and counter overt attacks [35].

Numerous countermeasures have been proposed for specific hardware attacks. Algorithmic resistance, restricting physical access, randomized computation time [33], and duplicate encryption [26] have been used to counter fault attacks. Time/branch equalization, random delays, and constant time hardware [27] have been used to counter timing attacks. Keyed hash functions, message authentication codes, public key infrastructure, and stream ciphers have been employed to increase the security of JTAG devices and encryption circuitry [29, 30].

Shielding has been used to counter acoustic attacks [15]. Cycling memory with random data can be used to mitigate data remanence attacks [25]. Cache partitioning has been shown to prevent information leakage [44], and sensitive cache lines can be placed in a secure partition [45] to counter cache attacks. Further, a non-deterministic processor can be used to run instructions in random order [47].

Figure 1: Hardware attack and defence strategies.

The contributions of this paper are as follows:

1. A comprehensive survey of hardware attack mitigation techniques is presented.
2. These techniques are matched to the hardware attacks and attack criteria they can counter. These results can help security personnel choose appropriate mitigation techniques to protect their systems against hardware attacks.

The remainder of this paper is organized as follows. Section 2 presents a review of hardware attacks based on attack awareness. Mitigation techniques for covert hardware attacks are presented in Section 3, while techniques for overt hardware attacks are given in Section 4. A discussion of mitigation techniques and some security recommendations are presented in Section 5. Finally, some concluding remarks are given in Section 6.

2. HARDWARE ATTACKS

Hardware attacks aim at physically accessing a system to obtain stored information, determine the internal structure of the hardware, or inject a fault. Several approaches have been proposed to classify hardware attacks based on security levels [69–71], algebraic properties [68], accessibility [66], and resources [1, 2].

In order to evaluate security, tamper protection levels were introduced by IBM [69]. Their classification has six security levels from zero corresponding to a system without any security protection to high for a virtually unbreakable system.

U.S. and Canadian federal government agencies are required to use cryptographic products that have been validated using Federal Information Processing Standards (FIPS) [70] or Common Criteria (CC) [71]. Most CC protection profiles rely on FIPS validation for cryptographic security. FIPS 140-2 or 140-1 validations have four security levels from level 1 which indicates basic security requirements for a cryptographic module to level 4, which indicates physical

security, i.e. an envelope of protection around the cryptographic module to detect device penetration. This classification focuses on cryptographic applications and/or devices.

A flexible methodology was proposed in [68] to categorize hardware attacks based on their properties. Weights can be assigned based on attack criteria so that detailed comparisons can be made, and as technology changes these weights can be adjusted according to attack and/or defence capabilities. A defender can use this methodology to determine the possible approaches an attacker may use to launch an attack. Variations of the same attack can also be considered. For example, two Deprocessing (DEP) attacks DEP-1 and DEP-2 were considered in [68] where DEP-1 assumes that the attacker uses in-house resources, while DEP-2 assumes the attacker employs outsourcing and so requires fewer resources. In general, the classification of a hardware attack is based on the capabilities and techniques used by the attacker and de-fender. This information can be used by security designers to identify system vulnerabilities and develop countermeasures.

A classification based on attack accessibility was proposed in [66]. This classification divides attacks into three groups: non-invasive, invasive, and semi-invasive. Non-invasive attacks do not require any initial preparation or direct connection to the device. Invasive attacks require direct access to the internal components of the device. Semi-invasive attacks introduced in [73] lie in the gap between non-invasive and invasive attacks. These attacks require a moderate level of accessibility to gain access to the chip surface, but do not require internal physical contact.

In [1, 2], a classification was proposed based on the resources and awareness needed for an attack to succeed. Attacks were classified based on four criteria: Accessibility (A), Re-sources (R), Time (T), and Awareness (W). The awareness criterion (W) divides hardware attacks based on the evidence left of an attack on a system, so there are two categories, covert [2] and overt [1].

2.1. Covert Attacks

A covert attack is when the victim is not aware that it is taking place. This occurs when an attacker is able to obtain information from a device when it is in operation. Figure. 2, illustrates the signals that emanate from a system that can be used by an attacker to compromise a chip. The use of these signals was first employed during World War II after Bell Labs employees noted that whenever a system was activated, signals appeared on an oscilloscope in another part of the lab. They determined that these signals could be used to recover plaintext data [20]. Covert attacks typically require limited accessibility and resources and little time, so often there is no evidence of the attack. Thus, they are simple and so many attackers have the necessary resources and experience to conduct them [12, 13]. As a consequence, many covert hardware attacks exist.

Simple Electromagnetic Analysis (SEMA) attacks examine the electromagnetic emissions from ICs to determine internal operations or obtain secret information. Differential Electro-magnetic Analysis (DEMA) attacks use a statical analysis of these emissions to obtain system information [21]. A Frequency Based Analysis (FBA) attack uses a differential Power Spectral Density (PSD) analysis rather than a time domain analysis [64]. A Simple Power Analysis (SPA) attack examines the power consumption during computations to determine device operations [52]. Differential Power Analysis (DPA) attacks use a statical model of the power consumption during chip operations in conjunction with cryptanalysis techniques to gather secret information [63]. A Timing Attack (TA) exploits the fact that chip operations can take different times to complete. Measuring and analyzing this time can be used to obtain secret information [51]. An Acoustic Attack (ACA) uses a microphone placed close to a chip to determine when encryption operations are being executed, and when the keys change [62]. An Optically Enhanced Position-Locked

Power Analysis (OPLP) attack reveals the current in individual transistors [61]. An Optical Emanation Attack (OEA) monitors the optical signals

Figure 2: Emanating signals from a system.

related to the information being processed [14]. A Covert JTAG port (C-JTAG) attack uses a JTAG port to obtain secret data [30]. A Data Remanence Attack (DRA) aims to retrieve secret information such as that stored in the SRAM of a security processor. At temperatures below a threshold, SRAM contents can be preserved for a period of time even when the power is removed. The attack is triggered when the temperature falls below or rises above certain thresholds [60]. An Advanced Imaging Technique (AIT) attack uses imaging to locate failed transistors or interconnects. This is based on the fact that silicon becomes transparent at Infrared (IR) wavelengths greater than $\lambda = 1100$ nm [66]. A Cache-based (Cache) attack exploits cache hit and miss timing differences to reveal system information such as cryptographic keys [42].

2.2. Overt Attacks

An attack is overt when the victim is aware that it is taking place. In this case, the attacker has one or more of the following goals:

1. Disrupting the system to prevent it from working as expected,
2. Preventing the system from working (denial of service),
3. Reverse engineering the system, to later copy it.

Most overt attacks require significant knowledge, high accessibility to the system, substantial resources, and a long period of time to achieve success. Therefore, the victim will typically be aware of these attacks. Modern chips are complex multilayer devices, so an attack that requires decapsulating a chip to access its internal circuitry is very difficult. These attacks are usually conducted by organizations such as research agencies, government laboratories and universities [1]. A number of covert hardware attacks have been proposed in the literature.

A Fault Injection (FIT) attack inserts signals into a device to alter its operation, typically via the power supply or clock [66]. An Overt JTAG port (O-JTAG) attack uses a JTAG port to obtain test vectors and responses, modify system states, return false test responses, or control Test Mode Select (TMS) and Test Clock (TCK) lines to deceive a tester about the true state of the chip. TMS and TCK attacks require control of TMS and TCK lines to change the voltage on the corresponding inputs [30]. A Fault Analysis (FAT) attack provides inputs to a device and then analyzes the resulting outputs. The inputs can be accidental or intentional and are used to discover secret information [27]. A Microprobing (Micro) attack uses a microprobe station to reveal the internal signals of a chip [66]. Reverse Engineering (RE) attacks extract information about the locations of transistors and interconnections in an IC to determine the functionality. To gain this

information, an attacker progressively removes the chip layers that were formed during fabrication [66]. A Deprocessing (DEP) attack employs the same processes that were used during chip fabrication. There are three types of deprocessing: wet chemical etching, dry chemical etching, and mechanical polishing [65]. Counterfeiting refers to the reproduction or copying of an IC [11].

4. COVERT HARDWARE ATTACK MITIGATION TECHNIQUES

Many mitigation techniques have been proposed to counter covert attacks. Most focus on making chip emissions independent of the operations. Some of these techniques have been designed for a specific attack while others can counter multiple attacks. The covert attack mitigation techniques are described below.

3.1. Hiding

Hiding is a powerful technique that can be used against an attacker attempting to gain information from chip emissions [52, 53]. The following techniques can be used to hide chip emissions.

3.1.1 Noise Generation

The Signal-to-Noise Ratio (SNR) can be reduced by either lowering the signal strength or increasing the noise level. For example, noise generators decrease the SNR which reduces the ability of an attacker to extract information from chip emissions [54]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, TA, ACA, and Cache.

3.1.2 Balanced Logic

Balanced logic is a technique used to make chip emissions independent of the data being processed. For example, Dual-Rail Pre-charged (DRP) logic can be used to create two outputs operating in different phases [53, 55]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, and DPA.

3.1.3 Asynchronous Logic Gates

Asynchronous logic gates can be used to lower Electromagnetic (EM) emission levels by reducing or eliminating the need for clock synchronization [56, 57]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, and DPA.

3.1.4 Low Power Design

Low power design is a method used to lower the SNR and hide chip emissions to reduce the ability of an attacker to obtain chip information [53]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, ACA, OPLP, OEA, and AIT.

3.2. Shielding

Shielding is an effective method to hide chip emissions. This can be achieved via physical shielding or filtering of chip emissions. Metal layers on the outside of a chip can be used to shield EM emissions. For FBA, a sensor mesh can monitor the chip operations for interruptions or short circuits and raise an alarm if one of these events occurs. Glass shielding, opaque material or black

taping can be used to guard against optical attacks [14]. For ACA, acoustic shielding such as foam can be employed [15]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, ACA, OPLP, OEA, AIT, and Cache.

3.3. Masking (Blinding)

Masking or blinding is a technique used to make it difficult for an attacker to determine the relationship between chip data and emissions. This can be accomplished on a per-gate basis using masking logic, or a per-block basis by randomizing the input data and reversing this operation to obtain the results [16–19]. The input data can also be masked with random data before any operations and the results obtained by removing the mask [27]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, TA, ACA, OPLP, OEA, and AIT.

3.4. Design Partitioning

Design partitioning prevents information leakage between chip regions. For example, regions that operate on plaintext can be separated from those that operate on ciphertext [20, 21]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, TA, ACA, OPLP, and OEA.

3.5. Anti-tampering (Physical Security)

Anti-tampering or physical security is used to limit access by creating a secure zone around a chip. This also reduces the amount of emission data that can be collected [22, 23]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, DPA, TA, ACA, OPLP, OEA, DRA, C-JTAG, FBA, AIT, and Cache. It can also be used to mitigate the overt attacks: O-JTAG, FIT, and FAT.

3.6. Emission Filtering

Hardware and/or software emission filters can be used to reduce the amount of data that is leaked [67]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, ACA, OPLP, and OEA.

3.7. Restricting Physical Access

Restricting access to a device is a simple countermeasure against fault attacks. Encapsulating a device in a tamper-resistant case is an effective means of restricting access [33], which has been successfully implemented [31]. This technique can be used to mitigate the covert attacks: DRA, C-JTAG, and AIT. It can also be used to mitigate the overt attacks: FIT, O-JTAG, and FAT.

3.8. Randomized Computation Time

Randomizing the computation time of chip operations provides protection against fault attacks [33]. This technique can be used to mitigate the covert attack: TA, and the overt attacks: FIT and FAT.

3.9. Deep Sub-micron Technology

Data can be protected using storage devices covered with a top metal layer or constructed with deep sub-micron technology, which makes it difficult for an attacker to access the transistor level or recover data that has been erased [25]. This technique can be used to mitigate the covert attacks: DRA and AIT.

Table 1: Mitigation Techniques for Covert Hardware Attacks

	Hardware Mitigation Techniques													
	SEMA	DEMA	FBA	SPA	DPA	TA	ACA	OPLP	OEA	DRA	AIT	JTAG	Cache	
Randomized Computation Time						✓								
Access	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	
Processor														
Direct Mapped Cache														
Cache Line Locking														
Cache Partitioning														
Prevention													✓	✓
Protocol												✓		
Public/Private Key Pairs												✓		
Test Access Port (TAP) Design												✓		
(PUF)												✓		
Protocol												✓		
Constant time Hardware						✓								
Adding Random Delays						✓								
Equalization														
Cycling Memory with Random										✓				
Technology														
Emission Filtering	✓	✓	✓	✓	✓		✓	✓	✓		✓			
Anti-tampering (Physical Security)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Design Partitioning	✓	✓	✓	✓	✓	✓	✓	✓	✓					
Masking (Blinding)	✓	✓	✓	✓	✓	✓	✓	✓	✓					
Shielding	✓	✓	✓	✓	✓		✓	✓	✓				✓	
Design Power Low	✓	✓	✓	✓	✓		✓	✓	✓					
Logic	✓	✓		✓	✓									
Balanced Logic	✓	✓		✓	✓									
Noise Generation	✓	✓	✓	✓	✓	✓	✓				✓			

3.10. Cycling Memory with Random Data

Memory, i.e. ROM or flash, cells should be cycled 10 to 1000 times with random data when storing sensitive data. This eliminates any noticeable effects arising from the data. Further, EEPROM and flash memory cells should be programmed before erasing them to eliminate any

residual charge from the previous data [25]. This technique can be used to mitigate the covert attack: DRA.

3.11. Time/Branch Equalization

The main countermeasure against timing attacks is to make all chip operations execute in the same amount of time, known as time equalization. For example, multiplication and exponentiation operations should have the same computation time. This prevents an attacker from determining the number of times each operation is executed, or even when an operation is run [26]. A variation of this technique is branch equalization, which makes the time for two branches of a conditional statement the same [27]. Note that time and branch equalization can have a negative impact on system performance. This technique can be used to mitigate the covert attack: TA.

3.12. Adding Random Delays

Adding random delays is another technique used to disguise the computation time for an operation. However, using fixed delays still enables an attacker to infer the system response and/or power consumption for specific operations. Therefore it is better to add random delays only to counter timing attacks [26]. Unfortunately, adding delays has a negative impact on system performance. This technique can be used to mitigate the covert attack: TA.

3.13. Constant Time Hardware

Employing constant time hardware is an easier approach to attack mitigation than adding random delays. In this case, each operation takes the same amount of time regardless of the input [27]. This technique can be used to mitigate the covert attack: TA.

3.14. Operation-Memory Access Prevention

This technique prevents memory access by using arithmetic and logic operations instead of look up tables [42]. Data is preloaded into the cache before it is used so attempts to access the cache always result in a hit, so no information is leaked [43]. This technique can be used to mitigate the covert attack: Cache.

3.15. Cache Partitioning

Cache partitioning prevents information leakage by placing the data in secluded or locked mode. A partitioned cache also separates the cache behaviour of one process from another, which prevents inter-process attacks. Although the cache is still shared, one process cannot access the partitions used by another [44]. This technique can be used to mitigate the covert attack: Cache.

3.16. Cache Line Locking

Cache line locking is an extension of cache partitioning. A partitioned cache is static and prevents sharing of unused cache lines with other processes, which is inefficient. Cache line locking is more flexible since only the sensitive cache lines are in a secluded and or locked partition, leaving the remaining cache lines for general use [45]. This technique can be used to mitigate the covert attack: Cache.

3.17. Direct Mapped Cache

Direct mapped cache maps a memory block to a cache line at run time. This can be achieved using the addresses with a remapping table to obtain the cache line indexes. These indexes are hashed into a dedicated partition. An efficient implementation is dynamic memory-to-cache remapping [46]. This technique can be used to mitigate the covert attack: Cache.

3.18. Non-deterministic Processor

A non-deterministic processor can be used to run instructions in a random order while maintaining data coherence and dependencies. The power profile in these processors makes it very difficult for an attacker to obtain information [47]. This technique can be used to mitigate the covert attack: SEMA, DEMA, FBA, SPA, DPA, TA, ACA, OPLP, OEA, AIT, and Cache.

3.19. Secure JTAG Communication Protocol

A secure JTAG communication protocol can be achieved using hash functions, message authentication codes, and stream ciphers. These security primitives can be used to design secure protocols between devices [30]. This technique can be used to mitigate the covert attack: C-JTAG, and the overt attacks: O-JTAG, RE, DEP, and Counterfeit.

3.20. Physically Unclonable Function (PUF)

Existing solutions for network communications such as SSH and SSL are computationally expensive. A simpler approach is to employ a Physically Unclonable Function (PUF) to provide a cryptographic key. The advantage of PUFs is that they are intrinsically unique and inherently random [28]. This technique can be used to mitigate the covert attack: C-JTAG, and the overt attacks: O-JTAG, RE, DEP, and Counterfeit.

3.21. Test Access Port (TAP) Design

JTAG Test Access Port (TAP) design employs hash functions and challenge/response protocols to securely access JTAG infrastructure. It also allows for hierarchical security for JTAG access [29]. This technique can be used to mitigate the covert attack: C-JTAG, and the overt attack: O-JTAG.

3.22. Public/Private Key Pairs

Public/private key pairs can be used for authentication to protect against JTAG attacks. For example, different keys can be employed for JTAG access groups to improve security [29]. This technique can be used to mitigate the covert attack: C-JTAG, and the overt attack: O-JTAG.

3.23. Challenge/Response Protocols

Random number generators can be used to generate challenge messages for use in authentication protocols [29]. This technique can be used to mitigate the covert attack: C-JTAG, and the overt attack: O-JTAG.

Table 1 presents the mitigation techniques associated with covert hardware attacks. Many of these techniques increase noise levels to hide the data signals, while others reduce chip emission levels. The goal is to make it difficult for an attacker to extract the signals from a chip, so chip access

should be limited to only trusted personnel. Some techniques have been designed to protect against multiple attacks, while others are only effective against single hardware attacks such as JTAG, TA, and Cache.

4. OVERT HARDWARE ATTACK MITIGATION TECHNIQUES

Overt attack mitigation techniques are primarily used to prevent an attacker from analyzing the inner structure of a chip. Often an attacker uses an overt attack to understand the chip structure and then use this information in a covert attack. This information can also be used to copy a chip. The overt attack mitigation techniques are described below.

4.1. Error Detection

Error detection codes are used to generate check bits for input data and operation results. If the check bits at the output are incorrect, a fault is detected and the output data is discarded [33]. This technique can be used to mitigate the overt attacks: FIT and FAT.

4.2. Duplicate Operations

Chip operations can be executed multiple times and the outputs considered valid only when they are identical [32]. If the results differ, an alarm is raised. This is not the best solution to defend against fault-based attacks since a fault may still go undetected. It increases the system complexity, but also the resources and time required by an attacker to obtain sufficient data [26], so while implementation is simple, the overhead is high. This technique can be used to mitigate the overt attacks: FIT and FAT.

4.3. Top Layer Sensor Meshes

Sensor meshes are mainly used to protect against microprobing attacks. They are placed above the circuit to detect interruptions and short circuits. If procedures such as selective etching or laser cutting are sensed, an alarm can be raised and countermeasures taken such as erasing non-volatile memory [35]. These meshes can also protect against under-voltage or over-voltage analysis attacks. This technique can be used to mitigate the overt attacks: FIT, Micro, RE, and DEP.

4.4. Clock Frequency Sensor

Robust low frequency sensors are used to detect tampering which slows the clock frequency [35]. If a sensor raises an alarm, countermeasures such as processor reset and bus line and register grounding can be taken. This technique can be used to mitigate the overt attacks: FIT, Micro, RE, and DEP.

4.5. Randomized Clock Signal

This technique can be used to prevent an attacker from predicting the execution time of specific instructions. Most covert hardware attacks require the attacker to predict the time at which a certain instruction is executed. Moreover, processors typically execute the same instructions with a fixed number of clock cycles after each reset, which makes processor behavior predictable. This behaviour simplifies the use of protocol reaction times as a covert channel. Therefore, random time delays should be inserted between any observable action and critical operations that might be subject to an attack. If serial ports are the only observable channels, then random delay routine calls controlled by a hardware noise source can be employed. A

random bit-sequence generator in conjunction with an external clock signal can be used to generate a random internal clock signal to make behavior prediction more difficult [34, 35]. This technique can be used to mitigate the overt attack: Micro.

4.6. Randomized Multi-threading

The predictability of execution cycles in a processor can be decreased by implementing a multi-threaded architecture, which randomly schedules execution on multiple threads [34]. Randomized combinational logic can be used to determine the progression of thread execution in a processor [35]. This technique can be used to mitigate the overt attack: Micro.

4.7. Test Circuit Destruction

Chip testing is done after production, and leaves residual test circuits, which can be exploited by attackers to gain access to buses and control lines. Therefore, the destruction of these circuits is an important attack countermeasure. To achieve this, the test interface for a chip can be placed within the area of another chip on the wafer. Then when the wafer is cut into dies, the connections between the chip and test circuitry are destroyed [35]. This technique can be used to mitigate the overt attack: Micro.

4.8. Restricted Program Counter

The program counter can be used as an address pattern generator to simplify reading the memory contents via microprobing. To counter such attacks, watchdog counters can be used to reset the processor if no jump, call, or return instruction is executed for a number of cycles, but this requires additional circuitry. Another approach is to modify the program counter so that offset counters are employed to cover the entire address space. Each call, jump, or return instruction writes the address of the destination in a register and resets the program counter [35]. This technique can be used to mitigate the overt attack: Micro.

4.9. Encrypted Buses

Encrypted buses can be used to make it intractable for an attacker to obtain chip data. The encryption typically employs a Random Number Generator (RNG) which is initialized at the sender and receiver using a private key [36]. This technique can be used to mitigate the overt attack: Micro.

4.10. Light Sensor

Light sensors can be employed to prevent chip operation after it has been decapsulated [23]. This technique can be used to mitigate the overt attacks: Micro, RE, and DEP.

4.11. Glue Logic

Glue logic can be used to transform standard building block structures, i.e. the ALU, I/O, registers, and/or CPU circuits to Application Specific Integrated Circuits (ASICs) with a similar logic design. This makes it very difficult for an attacker to find specific signals or circuitry within the IC. Glue logic design can be achieved using special design tools [66]. This technique can be used to mitigate the overt attacks: Micro, RE, and DEP.

4.12. Obfuscation

Obfuscation is a technique that transforms a circuit or design into one that is functionally equivalent but is significantly more difficult to reverse engineer [37–39]. Thus, more resources and time will be required for an attacker to determine chip functions. Obfuscation can also be implemented using PUFs or programmable logic. In this case, the logic is configurable to functionally equivalent designs to conceal the signal paths [40]. This technique can be used to mitigate the overt attacks: Micro, RE, DEP, and Counterfeit.

4.13. Verification Difference

Verification difference is used to test chips by comparing measurements with signature values to detect differences between genuine and altered chips. Altered chips will have a significant difference and thus can be identified. This technique includes power and time delay analysis as well as Scanning Acoustic Microscopy (SAM), IR thermography and X-Ray Fluorescopy (XRF) [58]. This technique can be used to mitigate the overt attacks: Micro, RE, DEP, and Counterfeit.

4.14. IP Watermarking

Intellectual Property (IP) watermarking is a technique similar to paper watermarking and is used to protect against counterfeiting. This is achieved by inserting proprietary information into the IC design. The result is a unique design that includes the watermark within the chip functions. The watermark can be embedded in different abstraction levels of the design making it difficult to detect and/or remove [41]. This technique can be used to mitigate the overt attacks: RE, DEP, and Counterfeit.

4.15. IP Fingerprinting

IP fingerprinting assigns a unique and hidden ID into each instance of the IP [48]. It is typically employed to detect IP overbuilding by a factory. This technique can be used to mitigate the overt attacks: RE, DEP, and Counterfeit.

4.16. IC Metering

IC metering is a set of security protocols that enable designers to gain post-fabrication control of IC properties and use, including remote runtime disabling [49]. A unique ID for each IC is included in the Finite State Machine (FSM) of the design. This is achieved by adding new states and transitions to the original IC FSM to create a Boosted Finite State Machine (BFSM). To bring the BFSM into the initial (reset) state, knowledge of the transition table is required. Since only the designer has this information, it will be difficult for an attacker to generate the input sequences required to bring the BFSM into this state [49]. Another IC metering protocol is based on PUFs [59]. It provides control over all hardware copies and allows counterfeit ICs to be disabled. This technique can be used to mitigate the overt attacks: RE, DEP, and Counterfeit.

Table 2 presents the mitigation techniques associated with overt hardware attacks. RE and DEP are closely related attacks and therefore can be mitigated using the same techniques. Several mitigation techniques have been designed to counter only Micro attacks. Some sets of mitigation techniques can be used to counter all overt attacks. A defender could use one of these sets (based on the target system), to mitigate these attacks. For example, a technique used to detect chip overbuilding can be combined with a technique to prevent an attacker from determining the inner structure of a chip.

5. DISCUSSION

Algorithms that can be used to assess the security of a system against hardware attacks were presented in [68]. These algorithms were developed based on the criteria, relationships, and/or occurrences of hardware attacks. The criteria considered are Accessibility (A), Resources (R), Time (T), and Awareness (W). Each criterion can be divided into sub-levels depending on the application [12, 13] and target system [50]. For example, consider a defender that has discovered a system is vulnerable to a Timing Attack (TA). There can be several variations of this attack, i.e. TA-1 which requires {covert, limited access, limited

Table 2: Mitigation Techniques for Overt Hardware Attacks

		Hardware Mitigation Technique																			
		IC Metering	IP Fingerprinting	IP Watermarking	Verification Difference	Obfuscation	Glue Logic	Light Sensor	Encrypted Bus	Clock Frequency Sensor	Top Layer Sensor Meshes	Restricted Program Counter	Test Circuit Destruction	Randomized Multi-threading	Randomized Clock Signal	Physically Unclonable Function (PUF)	Duplicate Operations	Error Detection	Randomized Computation Time	Restricting Physical Access	Anti-tampering (Physical Security)
Overt Hardware Attack	FAT															✓	✓	✓	✓	✓	✓
	FIT								✓	✓						✓	✓	✓	✓	✓	✓
	Micro				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓							
	RE		✓	✓	✓	✓	✓		✓	✓					✓						
	DEP		✓	✓	✓	✓	✓	✓	✓	✓	✓				✓						
	Counterfeit			✓	✓											✓					

resources, limited time}, TA-2 which requires {covert, limited access, limited resources, medium time}, and TA-3 which requires {covert, partial access, limited resources, limited time}, to succeed. These attacks are illustrated in Figure. 3. A defender has two approaches to protecting the system against these attacks, choosing a technique, which can specifically defend against a TA, or one that can defend against the criteria required by a TA to succeed.

To protect the system against a timing attack, a defender can employ Table 1 to determine which mitigation techniques can be used to protect the system. This indicates that any one of the techniques noise generation, masking (blinding), design partitioning, anti-tampering (physical security), time/branch equalization, adding random delays, constant time hardware, non-

deterministic processor, and random computation time, can be used to counter a timing attack. Some of these techniques can be implemented during the design phase (i.e. design partitioning), while other can be implemented during the operation phase (i.e. noise generation).

To protect a system against a set of criteria, a defender can employ Table 3 to determine appropriate mitigation techniques. The defender can protect the system against a single criterion

Figure 3: Three timing attacks with different criteria.

(i.e. covert or limited resources), or against a combination of criteria (i.e. covert and limited resources).

The advantage of defending against criteria rather than attacks is that this may protect the system against additional attacks. For example, most of techniques described in Sections 3 and 4 can be combined with anti-tampering (physical security) to protect against the situation where an attacker breaches a security zone. Further, several of the techniques in Section 4 can be combined with obfuscation to improve protection against overt attacks. This will help in the case of an attacker defeating a mitigation technique. For example, a defender that employs a PUF to defend against an attack, and an attacker that uses the technique in [72] to bypass the PUF. If the defender becomes aware of this, they can implement an additional mitigation technique such as clock frequency sensor or a technique that denies partial access so this attack will not succeed.

6. CONCLUSION

Determining the possible hardware attacks against a system is a critical step in developing a defence strategy. Once an attack has been identified, an appropriate mitigation technique should be employed to protect the system. Many hardware attacks are covert, in which case a de-fender will not be aware of the attack. Therefore, it is critical to develop mitigation techniques to counter these attacks. Several overt attacks have been developed to gain information about a system, which can later be used in a covert attack, or to make a copy (counterfeit), which is a major concern. Some mitigation techniques can counter multiple attacks, while others have been developed to counter single attacks. Physical security creates a secure zone around a chip to limit the data an attacker can collect from emissions, and is an effective technique against many covert attacks.

Table 3: Hardware Attack Mitigation Techniques and the Corresponding Hardware Attack Criteria

		Criteria										
		Awareness		Accessibility			Resources			Time		
		Covert	Overt	Limited Access	Partial Access	Full Access	Limited Resources	Moderate Resources	Excessive Resources	Short Time	Medium Time	Long Time
Hardware Attack Mitigation Techniques	Noise Generation	✓		✓	✓		✓			✓	✓	
	Balanced Logic	✓		✓	✓		✓			✓	✓	
	Asynchronous Logic Gates	✓		✓	✓		✓			✓	✓	
	Low Power Design	✓		✓	✓		✓	✓		✓	✓	
	Shielding	✓		✓	✓		✓	✓		✓	✓	
	Masking (Blinding)	✓		✓	✓		✓	✓		✓	✓	
	Design Partitioning	✓		✓	✓		✓	✓		✓	✓	
	Anti-tampering (Physical Security)	✓	✓	✓	✓		✓	✓		✓	✓	
	Emission Filtering	✓		✓	✓		✓	✓		✓	✓	
	Error Detection		✓		✓		✓			✓	✓	
	Restricting Physical Access	✓	✓		✓		✓	✓		✓	✓	
	Duplicate Operations		✓		✓		✓			✓	✓	
	Randomized Computation Time	✓	✓		✓		✓			✓	✓	
	Top Layer Sensor Meshes		✓		✓	✓	✓	✓	✓	✓		✓
	Clock Frequency Sensor		✓		✓	✓	✓	✓	✓	✓		✓
	Deep Sub-micron Technology	✓			✓		✓	✓			✓	
	Cycling Memory with Random Data	✓			✓		✓				✓	
	Time/Branch Equalization	✓			✓		✓				✓	
	Adding Random Delays	✓			✓		✓				✓	
	Constant Time Hardware	✓			✓		✓				✓	
	Operation-Memory Access Prevention	✓		✓			✓			✓		
	Cache Partitioning	✓		✓			✓			✓		
	Cache Line Locking	✓		✓			✓			✓		
	Direct Mapped Cache	✓		✓			✓			✓		
	Non-deterministic Processor	✓		✓	✓		✓	✓		✓	✓	
	Secure JTAG Communication Protocol	✓	✓		✓	✓	✓		✓		✓	✓
	Physical Unclonable Function (PUF)	✓	✓		✓	✓	✓		✓		✓	✓
	Test Access port (TAP) Design	✓	✓		✓		✓				✓	
	Public/Private Key Pairs	✓	✓		✓		✓				✓	
	Challenge/Response Protocols	✓	✓		✓		✓				✓	
	Randomized Clock Signal		✓			✓		✓				✓
	Randomized Multi-threading		✓			✓		✓				✓
	Test Circuit Destruction		✓			✓		✓				✓
	Restricted Program Counter		✓			✓		✓				✓
	Encrypted Buses		✓			✓		✓				✓
	Light Sensor		✓			✓		✓	✓			✓
	Glue Logic		✓			✓		✓	✓			✓
	Obfuscation		✓			✓		✓	✓			✓
	Verification Difference		✓			✓		✓	✓			✓
	IP Watermarking		✓			✓			✓			✓
	IP Fingerprinting		✓			✓			✓			✓
	IC Metering		✓			✓			✓			✓

REFERENCES

- [1] S. Moein and F. Gebali. Quantifying overt hardware attacks: Using ART schema. In *Computer Science and its Application, Lecture Notes in Electrical Engineering*, vol. 330, Springer, pp. 511–516, 2015.
- [2] S. Moein, F. Gebali, and I. Traore. Analysis of covert hardware attacks. In *J. Convergence*, vol. 5, no. 3, pp. 26–30, 2014.
- [3] M. Banga and M. Hsiao. A region based approach for the identification of hardware trojans. In *Proc. IEEE Int. Workshop on Hardware-Oriented Security and Trust*, pp. 40–47, 2008.
- [4] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. In *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.
- [5] M. Rostami, F. Koushanfar, and R. Karri. A primer on hardware security: models, methods, and metrics. In *Proceedings of the IEEE*, Vol. 102, Issue. 8, pp. 1283–1295, 2014.
- [6] S. Moein, S. Khan, T. A. Gulliver, F. Gebali, and M. W. El-Kharashi. An attribute based classification of hardware trojans. in *Proc. Int. Conf. on Computer Eng. and Sys.*, pp. 351–356, 2015.
- [7] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia. MERO: A statistical approach for hardware trojan detection. In *Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, vol. 5747, Springer-Verlag, pp. 396–410, 2009.
- [8] S. Saha, R. S. Chakraborty, S. S. Nuthakki, Anshul, and D. Mukhopadhyay. Improved test pattern generation for hardware trojan detection using genetic algorithm and boolean satisfiability. In *Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, vol. 9293, Springer, pp. 577–596, 2015.
- [9] S. Moein, J. Subramnian, T. A. Gulliver, and F. Gebali, and M. W. El-Kharashi, Classification of hardware trojan detection techniques. In *Proc. Int. Conf. on Computer Engineering and Sys.*, pp. 357–362, 2015.
- [10] S. Adee. The hunt for the kill switch. In *IEEE Spectrum*, vol. 45, no. 5, pp. 34–39, 2008.
- [11] K. M. Goertzel and B. A. Hamilton. Integrated circuit security threats and hardware assurance countermeasures. In *Crosstalk - Real Time Inform. Assurance*, pp. 33–38, 2013.
- [12] S. Moein and F. Gebali. Quantifying covert hardware attacks: Using ART schema. In *Proc. Adv. in Inform. Science and Computer Eng.*, pp. 85–90, 2015.
- [13] S. Moein and F. Gebali. A formal methodology for quantifying overt hardware attacks. In *Proc. Adv. in Inform. Science and Computer Eng.*, pp. 63–69, 2015.
- [14] J. Loughry and D. Umphress. Information Leakage from Optical Emanations. In *ACM Trans. Inform. and Sys. Security*, vol. 5, no. 3, pp. 262–289, 2002.
- [15] D. Genkin, A. Shamir, and E. Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology, Lecture Notes in Computer Science*, vol. 8616, Springer, pp. 444–461, 2014.
- [16] S. Chari, C. S. Jutla, J. R. Rao and P. Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology, Lecture Notes in Computer Science*, vol. 1666, Springer-Verlag, pp. 398–412, 1999.
- [17] L. Goubin and J. Patarin. DES and differential power analysis the “duplication” method. In *Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, vol. 1717, Springer-Verlag, pp. 158–172, 1999.
- [18] M.-L. Akkar and C. Giraud. An implementation of DES and AES, secure against some attacks. In *Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, vol. 2162, Springer-Verlag, pp. 309–318, 2001.
- [19] J. D. Golic and C. Tymen. Multiplicative masking and power analysis of AES. In *Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, vol. 2523, Springer-Verlag, pp. 198–212, 2003.
- [20] J. Friedman. Tempest: A signal problem. In *NSA Cryptologic Spectrum*. [Online]. Available: https://www.nsa.gov/public_info/_files/cryptologic_spectrum/tempest.pdf
- [21] J.-J. Quisquater and D. Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *Smart Card Programming and Security, Lecture Notes in Computer Science*, vol. 2140, Springer-Verlag, pp. 200–210, 2001.
- [22] R. Anderson and M. Kuhn. Tamper resistance - a cautionary note. In *Proc. USENIX workshop on Elec. Commerce*, pp. 1–11, 1996.
- [23] M. Tehranipoor and C. Wang. *Introduction to Hardware Security and Trust* (Eds.). Springer, 2012.

- [24] W. Cilio, M. Linder, C. Porter, J. Di, S. Smith, and D. Thompson. Side-channel attack mitigation using dual-spacer dual-rail delay-insensitive logic (D3L). In Proc. IEEE SoutheastCon, pp. 471–474, 2010.
- [25] S. Skorobogatov. Data remanence in flash memory devices. In Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science, vol. 3659, Springer-Verlag, pp. 339–353. 2005.
- [26] H. Bar-El. Introduction to side channel attacks. White Paper, Discretix Tech. Ltd. [online]. Available: <http://gauss.eecs.uc.edu/Courses/c653/lectures/SideC/intro.pdf>
- [27] Y. Zhou and D. Feng. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. IACR Cryptology ePrint Archive, 2005.
- [28] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In Proc. of ACM Design Automation Conf., pp. 9–14, 2007.
- [29] C. J. Clark. Anti-tamper JTAG TAP design enables DRM to JTAG registers and P1687 on-chip instruments. In Proc. Int. Workshop on Hardware-Oriented Security and Trust, pp. 19–24, 2010.
- [30] K. Rosenfeld and R. Karri. Attacks and defenses for JTAG. In IEEE Des. Test, vol. PP, no. 99, pp. 1–18, 2013.
- [31] IBM. (2008) CCA basic services reference and guide for the IBM 4758 PCI and IBM 4764 PCI-X cryptographic coprocessors. [Online]. Available: <http://www-03.ibm.com/security/cryptocards/pdfs/bs330.pdf>
- [32] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In Advances in Cryptology, Lecture Notes in Computer Science, vol. 1294, Springer-Verlag, pp. 513–525, 1997.
- [33] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache. Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. Proc. IEEE, vol. 100, no. 11, pp. 3056–3076, 2012.
- [34] S.W. Moore. Multithreaded Processor Design. Kluwer, 1996.
- [35] O. Kommerling and M. G. Kuhn. Design principles for tamper-resistant smartcard processors. In Proc. of the USENIX Workshop on Smartcard Tech., pp. 9–20, 1999.
- [36] L. Changlong, Z. Yiqiang, S. Yafeng, and G. Xingbo. A system-on-chip bus architecture for hardware trojan protection in security chips. In Int. Conf. of Electron Devices and Solid-State Circuits, pp. 1–2, 2011.
- [37] E. Castillo, U. Meyer-Baese, A. Garcia, L. Parilla, and A. Lloris. IPP@HDL: Efficient intellectual property protection scheme for IP cores. IEEE Trans. VLSI Systems, vol. 16, no. 5, pp. 578–591, 2007.
- [38] R. S. Chakraborty and S. Bhunia. Security against hardware trojan through a novel application of design obfuscation. In Proc. ACM Int. Conf. on Computer-Aided Design, pp. 113–116, 2009.
- [39] R. S. Chakraborty and S. Bhunia. Hardware protection and authentication through netlist level obfuscation. In Proc. IEEE/ACM Int. Conf. on Computer-Aided Design, pp. 674–677, 2008.
- [40] J. B. Wendt and M. Potkonjak. Hardware obfuscation using PUF-based logic. In Proc. IEEE/ACM Int. Conf. on Computer-Aided Design, pp. 270–277, 2014.
- [41] M. Ni and Z. Gao. Watermarking system for IC design IP protection. In Proc. Int. Conf. on Commun., Circuits and Systems, pp. 1186–1190, 2004.
- [42] D. A. Osvik, A. Shamir, and E. Tromer. Cache attacks and countermeasures: The case of AES. In Topics in Cryptology, Lecture Notes in Computer Science, vol. 3860, Springer-Verlag, pp. 1–20, 2006.
- [43] E. Brickell, G. Graunke, M. Neve, and J.-P. Seifert. Software mitigations to hedge AES against cache-based software side channel vulnerabilities. In IACR Cryptology ePrint Archive, 2006.
- [44] D. Page. Partitioned cache architecture as a side-channel defense mechanism. In IACR Cryptology ePrint Archive, 2005.
- [45] Z. Wang and R. B. Lee. New cache designs for thwarting software cache-based side channel attacks. In Proc. Int. ACM Symp. on Computer Architecture, pp. 494–505, 2007.
- [46] Z. Wang and R. B. Lee. A novel cache architecture with enhanced performance and security. In Proc. Int. IEEE/ACM Symp. on Microarchitecture, pp. 83–93, 2008.
- [47] D. Page. Defending against cache-based side-channel attacks. In Information Security Technical Report, vol. 8, no. 1, pp. 30–44, 2003.
- [48] A. E. Caldwell, H.-J. Choi, A. B. Kahng, S. Mantik, M. Potkonjak, G. Qu, and J. L. Wong. Effective iterative techniques for fingerprinting design IP. In IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 23, no. 2, pp. 208–215, 2004.
- [49] Y. Alkabani and F. Koushanfar. Active hardware metering for intellectual property protection and security. In Proc. USENIX Security Symp., pp. 291–306, 2007.

- [50] S. Moein, F. Gebali, T. A. Gulliver, and M. W. El-Kharashi. Hardware attack risk assessment. In *Proc. Int. Conf. on Computer Engineering and Sys.*, pp. 346–350, 2015.
- [51] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology, Lecture Notes in Computer Science*, vol. 1109, Springer-Verlag, pp. 104–113, 1996.
- [52] P. C. Kocher, J. M. Jaffe, and B. C. Jun. Differential power analysis. In *Advances in Cryptology, Lecture Notes in Computer Science*, vol. 1666, Springer-Verlag, pp. 388–397, 1999.
- [53] P. C. Kocher, J. M. Jaffe, B. C. Jun, and P. Rohatgi. Introduction to differential power analysis. *J. Cryptographic Eng.*, vol. 1, no. 1, pp. 5–27, 2011.
- [54] P. C. Kocher, J. M. Jaffe, and B. C. Jun. Using unpredictable information to minimize leakage from smartcards and other cryptosystems. US Patent No. 6,327,661, 2001.
- [55] J. M. Jaffe, P. C. Kocher, and B. C. Jun. Balanced cryptographic computational method and apparatus for leak minimization in smartcards and other cryptosystems. U.S. Patent No. 6,510,518, 2003.
- [56] G. Taylor, S. Moore, R. Anderson, R. Mullins, and P. Cunningham Improving smart card security using self-timed circuits. In *Proc. IEEE Symp. on Asynchronous Circuits and Systems*, pp. 211–218, 2002.
- [57] Z. C. Yu, S. B. Furber, and L. A. Plana. An investigation into the security of self-timed circuits. In *Proc. IEEE Symp. on Asynchronous Circuits and Systems*, pp. 206–215, 2003.
- [58] F. E. McFadden and R. D. Arnold. Supply chain risk mitigation for IT electronics. In *Proc. IEEE Int. Conf. on Tech. for Homeland Security*, pp. 49–55, 2010.
- [59] R. Maes, D. Schellekens, P. Tuyls, and I. Verbauwhede. Analysis and design of active IC metering schemes. In *Proc. Int. Workshop on Hardware-Oriented Security and Trust*, pp. 74–81, 2009.
- [60] S. Skorobogatov. Low temperature data remanence in static RAM. University of Cambridge Computer Laboratory Technical Report 536, 2002.
- [61] S. Skorobogatov. Optically enhanced position-locked power analysis. In *Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, vol. 4249, Springer-Verlag, pp. 61–75, 2006.
- [62] A. Shamir and E. Tromer. (2004) Acoustic cryptanalysis on nosy people and noisy machines. [online]. Available: <http://www.tau.ac.il/~tromer/acoustic/ec04rump/>
- [63] J. Zhang, D. Gu, Z. Guo and L. Zhang. Differential power cryptanalysis attacks against PRESENT implementation. In *Proc. Int. Conf. on Advanced Computer Theory and Engineering*, pp. V6-61–V6-65, 2010.
- [64] C. C. Tiu. A new frequency-based side channel attack for embedded systems. Ph.D. Dissertation, University of Waterloo, Waterloo, ON, 2005.
- [65] L. Wagner. Failure Analysis of Integrated Circuits: Tools and Techniques (Eds.). Springer, 2012.
- [66] S. Skorobogatov. Physical attacks and tamper resistance. In *Introduction to Hardware Security and Trust*, M. Tehranipoor and C. Wang (Eds.), Springer, New York, NY, pp. 143–174, 2012.
- [67] T. S. Messerges, E. A. Dabbish and R. H. Sloan. Examining smart-card security under the threat of power analysis attacks. In *IEEE Trans. Computers*, vol. 51, pp. 541–552, 2002.
- [68] S. Moein, F. Gebali, and T. A. Gulliver. Hardware attacks: An algebraic approach. In *J. Cryptographic Engineering*, pp. 1–13, 2016.
- [69] D. G. Abraham, G. M. Dolan, G. P. Double, J. V. Stevens. Transaction Security System. In *IBM Systems Journal*, vol. 30, no. 2, pp. 206–229, 1991.
- [70] U.S. Department of Commerce. Security requirements for cryptographic modules. [online]. Available: <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>
- [71] Common Criteria Evaluation and Validation Scheme. [online]. Available: <http://www.niap-cc-evs.org/>
- [72] Y. Oren, A.-R. Sadeghi, and C. Wachsmann. On the effectiveness of the remanence decay side-channel to clone memory-based PUFs. In *Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, vol. 8086, Springer-Verlag, pp. 107–125, 2013.
- [73] S. Skorobogatov. Semi-invasive attacks: A new approach to hardware security analysis. In *Technical Report UCAM-CL-TR-630*, University of Cambridge, Computer Laboratory, 2005.

AUTHORS

Samer Moein received the B.Sc. degree and the M.Sc. degree from Kuwait University, Kuwait, in 2004 and 2011, respectively, and the Ph.D. degree from the University of Victoria, Victoria, BC, Canada, in 2015, all in computer engineering. He is currently a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Victoria. His research interests include computer security, cryptography, and cryptoprocessors.

T. Aaron Gulliver received the Ph.D. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada, in 1989. From 1989 to 1991, he was a Defence Scientist with the Defence Research Establishment Ottawa, Ottawa, ON, Canada. He has held academic appointments with Carleton University, Ottawa, and the University of Canterbury, Christchurch, New Zealand. He joined the University of Victoria in 1999, where he is a Professor with the Department of Electrical and Computer Engineering. In 2002, he became a fellow of the Engineering Institute of Canada. In 2012, he was elected as a fellow of the Canadian Academy of Engineering. From 2000 to 2003, he was Secretary and a member of the Board of Governors of the IEEE Information Theory Society. He is currently an Area Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. His research interests include information theory and communication theory, algebraic coding theory, multicarrier systems, smart grid, and security.

Fayez Gebali received the B.Sc. (Hons.) degree in electrical engineering from Cairo University, the B.Sc. (Hons.) degree in mathematics from Ain Shams University, and the Ph.D. degree in electrical engineering from the University of British Columbia. He is a Professor with the Department of Electrical and Computer Engineering, University of Victoria, where he is currently Department Chair. His research interests include parallel algorithms, networks-on-chip, 3-D integrated circuits, digital communications, and computer arithmetic. He held an NSERC Postgraduate Scholarship from the University of British Columbia.

Abdulrahman Alkandari received the B.Sc. and M.Sc. degrees in computer engineering from Kuwait University, Kuwait, in 2004 and 2011, and the Ph.D. degree in computer science from the International Islamic University Malaysia in 2014. He is an Assistant Professor with the Department of Computer Science, Public Authority for Applied Education and Training (Basic Education College). His research interests include intelligent systems, traffic engineering, algorithms, smart phone applications, IoT, smart cities, and wireless sensor.