ANALYSIS AND MITIGATION OF EM SIDE-CHANNEL ATTACKS ON CHIP-TO-CHIP INTERCONNECTS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2023

Minmin Jiang

School of Engineering

Department of Computer Science

Contents

Word Count: 28893

List of Tables

List of Figures

Acronyms

2.5-D 2.5-dimensional

3-D three-dimensional

A/D analogue-to-digital

AES Advanced Encryption Standard

AHB advanced high-performance bus

AMBA advanced microcontroller bus architecture

ARM advanced risc machine

ASIC application-specific integrated circuit

AVR advanced virtual risc

BGA ball grid array

BRAMs block random access memories

CAT cache allocation technology

CEMA correlation electromagnetic attack

CMOS complementary metal-oxide-semiconductor

CPA correlation power attack

CPU central processing unit

CTS clock tree synthesis

DASH dynamic adaptive streaming over HTTP

DBI data bus inversion

DCMs digital clock managers

DEMA differential electromagnetic attack

DES data encryption standard

DIMM dual in-line memory module

DoM difference-of-means

DPA differential power attack

DRAM dynamic random access memory

DRP dual-rail precharge logic

DUT device under test

DVFS dynamic voltage and frequency scaling

ECDH elliptic curve Diffie-Hellman

EM electromagnetic

EMFI Electromagnetic fault injection

FETs field-effect transistors

FF fast-fast

FIFOs first-in-first-out buffers

FM frequency modulation

FPGA filed-programmable gate array

GE guessing entropy

GPU graphics processing units

GSa/s giga-samples per second

GSM global system for mobile communications

HBM high-bandwidth-memory

HD Hamming distance

HEX Hexadecimal

HTTP hypertext transfer protocol

HW Hamming weight

IC integrated circuit

ICs integrated circuits

IETF Internet Engineering Task Force

IoT Internet of Things

IPs intellectual properties

IVR induction voltage regulator

JTAG joint test action group

LCD liquid crystal display

LDO low-dropout

LEDs light-emitting diodes

LFI Laser fault injection

LLC last level cache

LNA low-noise amplifier

LUT look-up table

LUTs look-up tables

MIA mutual information attack

MIM metal-insulator-metal

MMIA multivariate mutual information analysis

MOS metal-oxide-semiconductor

MOSFET metal-oxide-semiconductor field-effect transistor

MTD measurement-to-disclosure

NAND not-and

NSD normalised standard deviation

P/G power/ground

PC personal computer

PDN power delivery network

PIN personal identification number

PUF physical unclonable function

RAMs random access memories

RDFF random D flip-flop

RDI random delay insertion

RDLs redistribution layers

RF radio frequency

RFID radio-frequency identification

RISC-V reduced instruction set computer - five

RNG random number generator

RO Ring oscillator

ROM read-only memory

RSA Rivest-Shamir-Adleman

RWDFF random write D flip-flop

SABL sense amplifier-based logic

SC switched-capacitor

SCA side-channel attack

SCAs side-channel attacks

SDR software-defined radio

SEMA simple electromagnetic attack

SGX software guard extensions

SMA subminiature version A

SNR signal-noise ratio

SoC system on chip

SPA simple power attack

SR success rate

SS slow-slow

SVI serial voltage identification

TSV through-silicon via

TSVs through-silicon vias

TT typical

TVLA test vector leakage assessment

USRP universal software radio peripheral

WC worst-case

WDDL wave dynamic differential logic

Abstract

This dissertation shows the feasibility of efficiently attacking interposer-based off-chip memory buses with a probe at the optimal placement found by a gradient-based search algorithm. It also shows the feasibility of enhancing security and performance of such interconnects with low-cost delay insertion techniques.

A fast search algorithm that calculates the gradient of normalised standard deviation (NSD) of the emissions is proposed to identify the most efficient attack point of the interposer-based bus. The simulation results on a 64-bit memory bus demonstrate that electromagnetic (EM) attacks at the optimal point require $\times 10$ fewer traces to recover the sensitive key than at the sub-optimal locations.

To protect the bus from EM side-channel attacks (SCAs), two delay insertion methods are proposed. The first one adds delay into the boundary lines of the bus, reducing the correlation between the EM emissions and transmitted data without degrading bus performance. The second one combines a dynamic delay insertion technique with the data bus inversion (DBI) technique, enhancing both bus performance and the resistance against EM attacks. It is demonstrated from the simulation results on an 8-bit interposer-based off-chip memory bus that both methods can lower the signal-noise ratio (SNR) value below 1, making the EM attacks unsuccessful, and the second method improves the worst-case latency by 9.5%. Both methods incurs low area overheads.

Finally, physical EM attacks on a board-to-board connection, as a proof-of-concept, are performed experimentally to validate the proposed methods.

Overall, this dissertation demonstrates efficient EM attacks on an interposer-based off-chip memory bus due to the increasing adoption of 2.5-dimensional (2.5-D) integrated systems and can be extended to wider or on-chip buses. It also proposes superior EM side-channel attack (SCA) mitigation methods that offer the same level of security protection with better performance than other hardware random delay insertion (RDI) methods.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

To my parents, who give me endless love and support

Acknowledgements

I would like to wholeheartedly thank my supervisor, Dr Vasilis F. Pavlidis, for providing guidance and support throughout my PhD period. The meetings and conversations between us were vital in inspiring me to think outside the box from time to time, to form a comprehensive and objective critique to my project. Further, I would like to thank my supervisor for the thoughtful comments and recommendations on this dissertation. Without him, this dissertation would not have been possible.

I also wish to express my sincere gratitude to my co-supervisor, Dr Dirk Koch, for providing invaluable advice during my annual research examinations and the support of FPGA boards regarding my experiments. I would also like to thank my team members Ionannis Papistas and Eleni Maragkoudaki, for showing me the ropes of different simulation tools and the collaborative effort during my research. Furthermore, I would like to thank Dr Mikel Lujan for giving the feedback on my research topic at the very early stage, Dr Barry Cheetham for providing extremely useful advice to my research and career development, Prof. Thomas Thomson for helping with the experimental settings in the lab, Dr James Garside for the invaluable discussion about my research, Dr Nguyen Dao for helping with the mixed-signal simulation and layout design, and the rest of my colleagues from the Advanced Processor and Technology (APT) group for creating such a friendly environment.

I would like to express my appreciation to project EuroExa H2020 and the School of Computer Science for providing me with funding support throughout this research project. I am also thankful to the University of Manchester and all its members of staff for all their support.

Last but not least, I would like to express my heartfelt gratitude to my family and friends, especially Yun Wang, for their constant encouragement and warm support throughout the completion of this dissertation. I am indebted to them for their emotional support, which has lifted my spirits and kept me optimistic throughout the entire process.

Chapter 1

Introduction

1.1 Motivation

With the development of new packaging solutions, in 2.5-D systems, the processor and memory chips can be integrated on the same substrate to scale the system structure and interconnect length for area reduction. Die stacking techniques can further facilitate the integration of discrete dies. So far, high-bandwidth-memory (HBM) has been integrated with graphics processing units (GPU) to meet high bandwidth requirements in the graphic modules that use 2.5-D integration techniques [5].

Research on side-channel attacks (SCAs) on 2.5-D integrated circuits (ICs) still lies at an early stage. Assuming the attackers own full physical access to the circuits, thermal-based SCAs and power-based SCAs on 2.5-D ICs have been exploited [6], [7]. For those two types of attacks, the thermal behaviour or the power consumption of the cryptographic core is modelled and analysed, then the secret key can be estimated through statistical pattern matching [8]. Except for the cryptographic core which encrypts and decrypts the sensitive data, the off-chip interconnects, implemented with higher-level metal wires and used to transfer the encrypted data between different chiplets, deserve further attention.

For the 2.5-D packaging paradigm, the improved interconnections among different components are enabled, where thick and long off-chip interconnects form wide buses to connect two or more dies on the same substrate. Due to technology and performance limitations, the technology node of the off-chip interconnects is not as advanced as the logic cells. Thus, the thick off-chip interconnects exhibit higher coupling capacitance and radiate more electromagnetic (EM) emissions when the current fluctuates. The intentional EM emissions, induced by the data transfer among different dies through

the buses, can become new passive side channels.

EM attacks on on-chip memory blocks have also been performed [9]. In this case, the adversary can not physically access the internals of the memory circuit, and, thus, has to rely on EM emissions for extracting the desired information (e.g. secret key). However, EM emissions in this scenario are also produced from adjacent circuit blocks, significantly hindering the efficiency of an on-chip EM attack. On the other hand, the large physical size of the off-chip memory buses facilitates EM attacks as EM emissions from neighbouring components can be weaker due to the larger physical distance. Dayeol *et al.* [10] performed an off-chip SCA, named *MEMBUSTER*, on the memory bus between the central processing unit (CPU) and the off-chip dynamic random access memory (DRAM). A dual in-line memory module (DIMM) interposer, as shown in Fig. 1.1, inserted between the processor and the DRAM, captures the memory bus signals and finally sends the signals to an analyser for the attack. However, with the adoption of 2.5-D integrated techniques, to the best of the author‘s knowledge, no exploration has been performed to show how best to attack the improved interconnections implemented in the interposer of such a system.

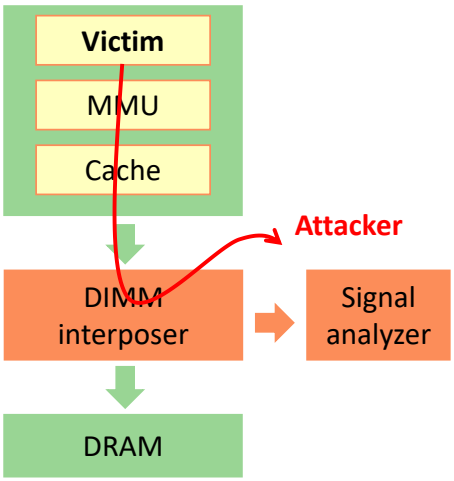


Figure 1.1: Dual in-line memory module interposer implemented in *MEMBUSTER*.

For the protection techniques against SCAs, random delay insertion (RDI) is an effective technique against a correlation based SCAs which belongs to the randomising category (see Chapter 2). RDI countermeasures have been utilised in the datapaths of microprocessors [11], filed-programmable gate array (FPGA) platforms [12], application-specific integrated circuit (ASIC) designs [13], and mask encryption algorithms on microprocessor [14]. These countermeasures effectively reduce the correlation between the assumed power model and measured power consumption, thereby

preventing potential correlation power attacks (CPAs). However, these RDI implementations can degrade circuit performance, or make timing closure more challenging as the timing slack of paths decreases, which is an important limitation, and makes these RDI methods less appealing.

Additionally, existing protection techniques against on-chip components EM attacks include introducing additional sources into the circuit or adapting specific fabrication steps into original integrated circuit (IC) design flow, such as the low-level metal routing technique. In this case, the cryptographic core is routed with low-level metal layers to suppress the critical signatures before they reach the top metal layer [15]. However, low-level metal routing can lead to routing congestion, higher interconnect resistance, and, therefore, performance degradation. Moreover, these techniques are not applicable to off-chip memory buses on interposers as only a few metal layers are available for routing compared to the on-chip interconnect stack that comprises over ten layers in modern fabrication processes. Compared with state-of-the-art mitigation methods against EM attacks, a new security methodology that can enhance immunity against EM attacks for the off-chip interposer-based buses, without degrading the bus performance and being compatible with the complementary metal-oxide-semiconductor (CMOS) technology, is required.

1.2 Contribution

To solve these challenges of efficient EM attacks and efficient countermeasures without performance sacrifice for the interposer-based off-chip memory buses, this dissertation proposes:

- **Efficient probe placement method:** A gradient-search algorithm that can quickly (i.e. $O(N)$) find the optimal probe position and extract the sensitive messages on the interconnects with the minimal measurement-to-disclosure (MTD). Since the voltage drop differences along the bus length direction are negligible, the search space above the bus surface is reduced from three dimensions to two dimensions. In the search along the two-dimensional plane, normalised standard deviation (NSD) value of the EM emissions is adopted to measure the information leakage. For a grid of $N \times N$, this gradient-search algorithm can lower the measurements needed to achieve minimal MTD from N^2 (brute-force measurements) to about N. Based on the attack hotspot identified by this scanning

strategy, considerably lower MTD is needed to obtain the correct key. Moreover, the search algorithm can adapt to different bus widths.

- **Performance-aware interconnect delay insertion scheme:** A scheme that can insert delay to counter EM attacks without compromising the bus latency is proposed. RDI is a proven technique to prevent SCAs on on-chip power networks, but it can impair circuit performance. The bus latency is determined by the lines with the largest cross-coupling capacitance (worst-case), so adding delay to the *boundary lines* with smaller capacitance does not impact the bus performance. This also diminishes the association between the data and the voltage at the probe terminal by altering the transition time of the *boundary lines*. However, the appropriate range of delay to enhance circuit security and preserve performance needs to be established in the experiments.

- **Dynamic delay insertion scheme:** A novel technique that combines the energy-efficient data bus inversion (DBI) technique with dynamic delay insertion is proposed. The specific delay on the *boundary lines* (mentioned previously) can hide the link between EM emissions and data, but the static nature of the method is not suitable for various buses and data types. Also, if the data on the *boundary lines* does not change, the correlation coefficient is unchanged and offers no protection. This dynamic delay insertion scheme adds delay to bit lines based on the Hamming distance (HD) of two consecutive pieces of encrypted data. This way, the delayed lines are hard to detect and reverse engineer. Moreover, unlike the previous static *boundary-line* delay technique, dynamic delay insertion does not worsen and sometimes even improves the bus latency. Furthermore, the delay scheme that produces the delay in both static and dynamic delay insertion methods is the same, which is a low-overhead circuit [16]. For example, based on *UMC* 65 technology, the delay circuit only requires about 160 metal-oxide-semiconductor (MOS) transistors and consumes only 104 μW, compared with a total power of 138 mW for the whole CMOS Advanced Encryption Standard (AES) circuit [17].

1.3 Scope

The main contribution of this dissertation lies in the field of EM SCAs and countermeasures for the 8-bit interposer-based interconnects in 2.5-D systems. The existing

gradient search algorithm, *Hamming distance leakage model* and *correlation EM attack method* are employed to reveal the secret information transmitted through the interconnects. However, the proposed solutions are not limited to the interposer-based interconnects in 2.5-D systems. They can be generalised across diverse interconnect bus types found in 2-D or 3-D systems, accommodating various bus widths and configurations.

To retrieve the correct key from the data transfer on the bus, many effective methodologies ranging from power attacks to EM fault injection attacks have been considered. However, this dissertation is focused on the technique of passively observing the EM emissions, due to the low cost of commercial sensors and its harmlessness in the devices' operations.

In particular, the techniques for EM SCAs mitigation range from modifying the physical layout to shielding the targeted device. However, in this dissertation, the focus is on circuit-level techniques. To provide security enhancements for EM attacks and bus performance, a lightweight scheme is utilised. The relationship between the data and emission is obfuscated by adding delay into specific bus lines, generated by a low-overhead circuit.

Further, the methods and concepts proposed in this dissertation can be combined with other technologies such as masking and hiding to further increase the complexity of the attack. However, their combinations for reaching lower power and area overheads require solving different research challenges, and hence are beyond the scope of this dissertation.

1.4 Dissertation Outline

The dissertation is organised into five main chapters:

- **Chapter 2:** Categorises different types of SCAs, leakage models, adversary models, figures of merits for security evaluation, and countermeasures against SCAs. Specifically, it identifies the detailed attack and protection implementations and issues existing in the current EM SCAs in terms of efficiency, performance maintenance, low area/power consumption, etc.

- **Chapter 3:** Describes the preliminaries of the EM attacks presented in this dissertation, such as the targeted algorithm and hardware platform, attack methodology, etc. Further, it evaluates a gradient-search algorithm for fast location of

the most efficient attack point and discusses its effectiveness in the EM snooping attacks on interposer-based off-chip memory buses.

- **Chapter 4:** Describes and discusses the mitigation schemes to protect the memory buses mentioned in Chapter 3 from EM SCAs. First, a static delay insertion scheme is proposed that can provide the bus lines a certain level of security protection from EM attacks without performance sacrifice. Then, a more powerful dynamic delay insertion scheme is identified to provide the bus lines benefits in both security and performance. The implementation details of these two schemes are presented, followed by the simulation results in both security and bus performance aspects, respectively.

- **Chapter 5:** Evaluates and validates the efficiency of the methods that are proposed in Chapters 3 and 4, with the encryption algorithm implemented on an FPGA board. The EM leakages are collected from an oscilloscope by placing a commercial probe near the interconnects. As a proof-of-concept, experimental attacks on a board-to-board bus, protection scheme of adding delay into two boundary lines, and a protection scheme of dynamically adding delay into different lines are conducted, respectively.

- **Chapter 6:** Summarises and states the main contributions of this dissertation and their implications for future EM attacks. It also discusses future work to further improve the attack efficiency, attack complexity of proposed methods and the potential new methodologies.

1.5 Publications

The ideas, methods, circuits and results presented in this dissertation have been published in peer-reviewed international conferences as listed below:

1. **M. Jiang**, E. Maragkoudaki, and V. F. Pavlidis, "Mitigating EM Side-Channel Attacks with Dynamic Delay Insertion and Data Bus Inversion," *International Symposium on Circuits and Systems*, Austin, pp. 1724–1728, May 2022.

2. **M. Jiang** and V. F. Pavlidis, "Performance-Aware Interconnect Delay Insertion Against EM Side-Channel Attacks," *International Workshop on System-level Interconnect Pathfinding*, Virtual, pp. 25–32, Nov. 2021.

3. **M. Jiang** and V. F. Pavlidis, "A Probe Placement Method for Efficient Electromagnetic Attacks," *International Conference on SMACD and PRIME*, Erfurt, pp. 1–4, June 2021.

An additional paper and contributions during the PhD period which is not part of this dissertation is:

- **M. Jiang**, I. A. Papistas, and V. F. Pavlidis, "Cost Modeling and Analysis of TSV and Contactless 3D-ICs," *Proceedings of Great Lakes Symposium on VLSI*, Beijing, pp. 519–524, Sept. 2020.

Chapter 2

Survey on Side-Channel Attacks

2.1 Chapter Overview

This chapter introduces the concept of SCAs, categories of SCAs, and existing countermeasures to mitigate the attacks. A comprehensive survey on the state-of-the-art SCAs and related mitigation methods are presented in this chapter. In particular, the emphasis is placed on EM SCAs, which is the attack method considered in this dissertation.

2.2 Understanding Side-Channel Attacks

First, what are SCAs? Side-channel represents any physical channel that leaks useful information and which can be monitored by adversaries to recover the sensitive key. The processes used to recover the key are identified as SCA processes. The critical point to distinguish SCAs from other attacks is that the useful information is **not directly leaked from the main communication channel** but from the side channel, such as instantaneous power consumption, EM emissions, acoustic noise, heat, light, etc (as shown in Fig. 2.1).

The SCA was first demonstrated in 1996 by Kocher [18]. The attack's targets, techniques and mitigation methods have changed over recent decades. Along with the development of novel technologies, the attack platform has evolved from smart cards and desktop computers to cloud computing instruments and mobile devices, compared to an earlier survey on classification of SCAs on mobile devices [19]. SCAs can be classified into two main categories: **active attacks** and **passive attacks**, according to whether the attacks interfere with the device operation or not. Each category can be

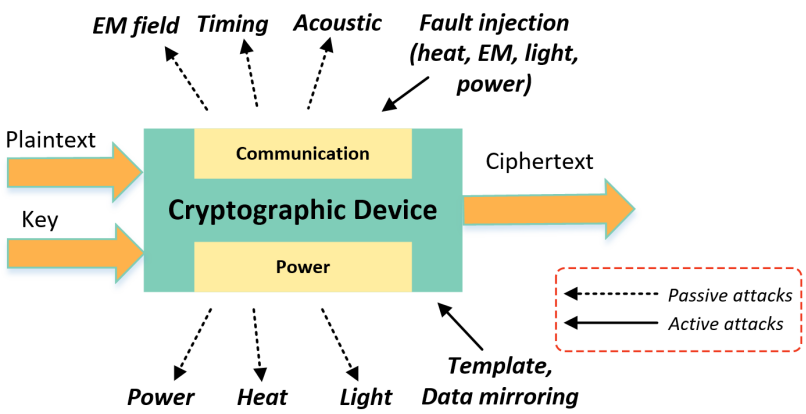


Figure 2.1: Overview of side-channel attacks on a cryptographic device.

further classified based on the targeted information as either hardware-based leakage (physical properties) or software-based leakage (logical properties) attacks. A taxonomy of the SCAs is illustrated in Fig. 2.2 and the overview of SCA methods, costs and targets is provided in Table 2.1. More detailed descriptions of different attack methods will be provided in the next subsections.

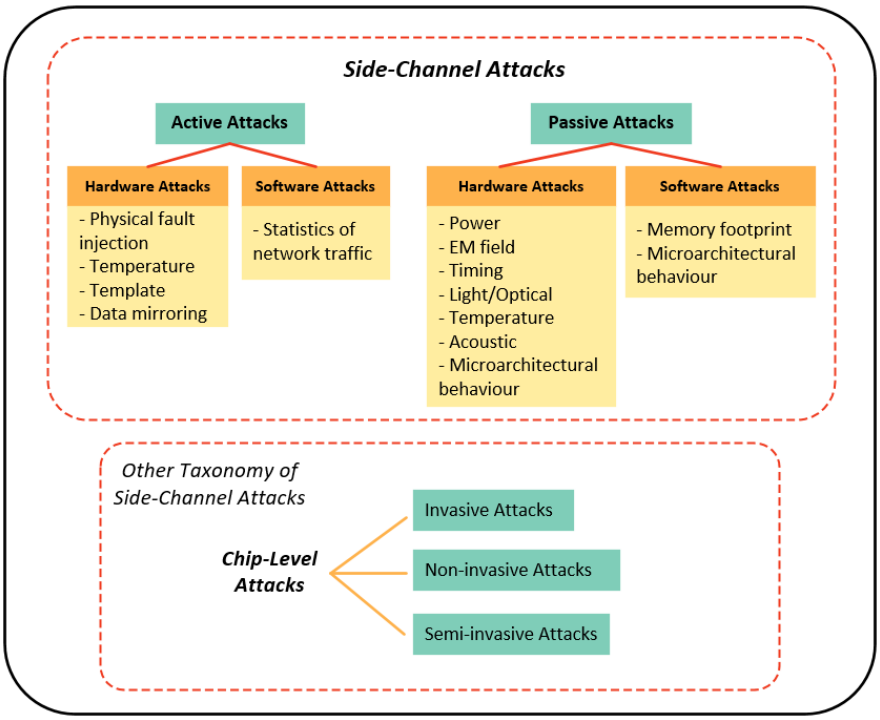


Figure 2.2: Taxonomy of side-channel attacks.

2.2.1 Active Attacks

Active attacks means that the operations or properties of the device may be influenced or even modified by the attackers. For example, attackers can manipulate the algorithm by injecting faults to bypass some mechanism, or interfere with the interconnect communication channels through fault EM radiation injection into the lines to trigger the attacks, where the device can indirectly leak some sensitive information.

Hardware attacks

- *Physical fault injection*: Fault attack/analysis was first proposed and theoretically analysed by Boneh, Demillio, and Liption in 1997 [20]. It was mathematically proved that if stored bits in registers were corrupted by random hardware faults, certain cryptographic protocols like Rivest-Shamir-Adleman (RSA) could be attacked [20]. This attack method was first experimentally validated in 2002 [21], when practical experiments with spike attacks were conducted on a RSA algorithm running on smart cards with no hardware or software protection. Over the past two decades, various fault attacks have been studied through clock/voltage glitching, an electromagnetic pulse or a laser beam.

    1. *Voltage glitching*: Manipulation of the power supply of the device can bypass some security modules [22] or leak some sensitive messages of the firmware code [23], [24], where only minimal modifications to the hardware and low-cost equipment are required for a successful attack.
    2. *Clock glitching*: Clock glitching has been proven effective in cyptographic systems on various embedded devices [25], [26], [27]. These examples assume that the device has an external clock source connected to the *CLK* pin, which is easy for the attackers to access. This assumption is needed because the clock injection timing needs a precise control over the clock period. However, for smaller mobile devices today, internal clocks are normally generated from on-chip circuits, which are harder for the attackers to reach.
    3. *Overclocking*: Overclocking means increasing or decreasing the clock period of the device to make it unreliable. The setup/hold timing constraints for the data launched from the input to the output of the flip-flops cannot be satisfied, where fault data can be launched instead of correct data due

to the timing violations [26]. Overclocking has been successfully utilised to attack a 128-bit hardware AES module implemented on a customised smart card [28].

4. *EMFI*: Electromagnetic fault injection (EMFI) injects faults into devices using a flux of the magnetic field, which was first proposed in 2002 [29] and has been proved as an effective fault injection method against circuits. For example, a software AES implementation on a 32-bit microcontroller based on advanced risc machine (ARM) Cortex-M3 can be attacked by radiating the round counter [30]. EMFI can focus on specific regions (memory blocks, buses, etc.) of the device and, meanwhile, cause less damage to the device compared with other fault injection techniques.

5. *LFI*: Laser fault injection (LFI) belongs to one of the most effective fault injection techniques that can provide high spatial and temporal resolution. It was first proposed by Skorobogatov and Anderson in 2002 [31]. Compared with EMFI, LFI needs chip decapsulation to get access to the silicon die, otherwise, the laser beam cannot pass through the package.

- *Temperature*: Temperature attacks can be enabled by tampering with the environmental temperature of the device (heating or cooling). The correlation between the heat and the circuit activities has been used to attack the sensitive key of the RSA running on advanced virtual risc (AVR) microcontrollers [32]. Even if strong isolation techniques are used in the multi-core platforms, the thermal covert channels can still become side-channels to recover the activities on neighbouring cores [33], [34], or the activities of another user on the same FPGA in Cloud FPGAs [35].

- *Template*: In template attacks, the attackers create a "profile" of the operations of the targeted device with different inputs [36], [37]. It needs full control over the device and copying it, in order to build a template of device operations based on the "points of interest" for the key recovery [38]. Precise modelling of the noise is critical for this form of attack [38].

- *Data mirroring*: Data mirroring, normally used in highly data-reliable systems, copies the data to a redundant storage device in time for any data-recovery requirements. As the copied data is exactly the same as the original data, data mirroring can be a security threat to the system and snooped by side-channel

attackers. For example, not-and (NAND) flash memory mirroring of iPhone 5c can bypass the limitation of passcode retry attempts and the 4-digit passcode can be attacked by brute-forcing in less than a day [39].

Software attacks

- *Statistics of network traffic*: With the burst in usage of websites and the information explosion, the communication channels via the network could compromise data privacy. For example, Tor is a popular communication system that can prevent website fingerprinting by overlapping some web objects to obfuscate the traffic features and, therefore, protect the privacy of the users [40]. However, assuming the attackers can manipulate the packets in the web traffic, the hypertext transfer protocol (HTTP) requests can be manually delayed for hundreds of milliseconds to guarantee that no or little overlapping happens to the web objects for the extraction of website fingerprinting of the users [41]. Moreover, dynamic adaptive streaming over HTTP (DASH) is a popular video streaming method adopted by Netflix, YouTube, etc. However, the characteristic of segment-based data transmission of DASH, brings a new threat of SCAs to video streaming based on the network traffic. The features of video segments can be automatically detected, extracted and stored. When new traffic traces are obtained, the video segment can be identified with an accuracy as high as 90% [42].

2.2.2 Passive Attacks

Passive attacks means the attackers only passively monitor the leaked information from side channels when the device operates. The operations or properties of the device will not be influenced, but the security primitives or premises in the algorithm [43] or the cryptographic implementations in the hardware [44] may be destroyed by the attackers.

Hardware attacks

- *Power*: Power-leakage based SCAs are traditional passive attacks, where the amplitude of the power consumption reflects the internal operations of the device. Consequently, if the current drawn or power fluctuations on the power supply rail is observed during the encryption process of the cryptographic circuit, the sensitive messages can be attacked. Based on trace-collecting strategies and statistical analysis, power attacks can be roughly classified into simple power attack

(SPA) [45], differential power attack (DPA) [46], correlation power attack (CPA) [47], template attack [38] and mutual information attack (MIA) [48].

- *EM field*: EM field is generated when the electric current flowing in the wires changes. The varing EM emissions, radiating from the device can be captured by a nearby-placed probe [49], [50] or a far-placed antenna [51]. Trace-analysis techniques used in power attacks are also applicable to EM attacks, such as a simple EM attack and differential EM attack, etc.

- *Timing*: In programs, the execution of different instructions requires different amounts of time therefore, the total execution time reflects the functional units. If the attacker has sufficient knowledge about the timing requirements of different function units, by using a local program [52] or through a remote observation over the network [53], [54] of the execution time, the execution paths and operations of the users can be reversed by the attackers.

- *Light/Optical*: In optical attacks, the sensitive information can be gleaned from the hard disk activity indicator (light-emitting diodes (LEDs)) [55] or a small number of photons emitted by the transistors on the circuits when they change their states [56] by visual recording using a high-resolution camera. In advanced mobile devices, the minor tilts and turns of the device can be sensed by the ambient-light sensor (popularly employed in advanced mobile devices), where the variations of the sensed information can be further used to attack the personal identification number (PIN) [57].

- *Temperature*: Program execution may affect the physical characteristic of the device (e.g. temperature) and the variations can be measured by fan-based solutions or on-chip/off-chip sensors. It is shown in [58], [32] that the heat radiation of the device (temperature leakage) has a linear relationship with the circuit activities. Meanwhile, the temperature SCAs demonstrate a low-bandwidth characteristic and are thermal-conductivity material dependant [32].

- *Acoustic*: Modern acoustic attacks focus on exploiting the sounds that may come from the speakers, keyboards, headphones, internal components, or the device's mechanical stress when the environmental temperature changes. The validation of acoustic threats was first demonstrated through using keyboards in 2005 [59] and was published in 2009 [60].

- *Microarchitectural behaviour*: Modern computer architectures consist of complex components which can lead to sensitive information leakage, where cache architecture has been shown as a strong side channel in CPU. For example, cache hits and misses can be observed by trace-driven attacks [61], [62].

Software attacks

- *Microarchitectural behaviour*: Cache attacks can also happen as software attacks. For example, the execution time of different instructions can be analysed by time-driven cache attacks, through observing the aggregate number of cache hits and misses [63], [64]. Moreover, the sensitive information of last level cache (LLC) can be touched by the famous "PRIME+PROBE" access-driven attacks [65], [66], [67] and "FLUSH+RELOAD" attacks [68], [69], [70], aiming to extract the cryptographic key.

- *Memory footprint*: The dynamic changes in the usage of memory can reveal the secret information hidden in the program. For example, by tracking the memory changes of modern web browsers, the pages that the user browses can be inferred moreover, even finer-grained user activities can be attacked [71].

Table 2.1: Comparison of state-of-the-art methods of side-channel attacks, attack cost and respective targets. The attack methods are grouped into seven types: SA: simple attack; DA: differential attack; ML: machine learning; SC: statistical correlation; CP: copy/backup; FI: fault injection; MIA: mutual information analysis. Meanwhile, the attack cost for each method is evaluated by SE: single execution, ME: multiple executions and CC: chosen ciphertexts. ✓and ✗indicate whether a physical attack has been performed on the corresponding targets.

Attack	**HW/SW**[1]	**Active/Passive**	**Attack Methods**	**Attack Cost**	**Targets**				
					Code	Website	Communication	Chip	Device
Voltage glitching	HW	Active	ML, FI	ME	✗	✗	✗	[24]✓	✗
Clock glitching	HW	Active	FI, DA	SE, ME, CC	✗	✗	✗	[26], [25]✓	[27]✓
Overlocking	HW	Active	FI, DA	ME	✗	✗	✗	[28]✓	✗
Template	HW, SW	Active	SA, DA, ML, SC	SE, ME	[72]✓	[73]✓	✗	[38]✓	[36], [37]✓
Data mirroring	HW	Active	CP	ME	✗	✗	✗	✗	[39]✓
Network traffic	SW	Active	SC	ME	✗	[41]✓	[42]✓	✗	✗
Power fluctuation	HW, SW	Passive	SA, DA, SC, ML, MIA	SE, ME, CC	✗	[74]✓	✗	[45], [47], [48]✓	[46]✓
Electromagnetic field	HW	Active, Passive	SP, DA, SC, ML	SE, ME, CC	✗	✗	✗	[49], [51], [50], ✓	[75], [76], [77]✓
Timing information	HW, SW	Passive	SC	ME, CC	[78]✓	✗	✗	[52], [53]✓	[54]✓
Light/laser	HW	Active, Passive	FI, SC	ME	✗	✗	✗	[56], [31]✓	[55], [57]✓
Acoustic noise	HW	Active, Passive	SC	ME	✗	✗	✗	✗	[79], [59]✓
Microarchitectural behaviour	HW, SW	Passive	SC	ME, CC	[61], [63], [65]✓	✗	✗	[62]✓	✗

[1] HW/SW: hardware/software attack

2.3 Alternative Characterisation of Side-Channel Attacks

In addition to the categorisation mentioned above, SCAs can be characterised as invasive, semi-invasive, and non-invasive attacks, depending on whether the passivation layer of the device package is removed or not. This classification method applies mostly to the chip-level attacks.

Invasive attacks

Invasive attacks are high-cost. The device under attack needs to be opened or depackaged for the exposure of the silicon die. Expensive equipment is required to directly contact the internal wires or circuit modules. Meanwhile, familiarity with the circuit function blocks and attack experience are also required for the attackers to extract useful information. This is because the complexity of circuits increases with the shrinking feature size.

Non-invasive attacks

Non-invasive attacks are low-cost. The device is monitored or manipulated by the attackers with no physical harm to the device. Moderately-sophisticated equipment and moderate experience are required for the attackers. The targeted device can be probed [80] or be integrated with other test platforms for trace collecting [81] and sometimes no tamper evidence can be found by the users.

Semi-invasive attacks

Semi-invasive attacks fill the gap between invasive and non-invasive attacks. They are inexpensive and repeatable. Less damage is induced to the device; even with the package removed, the passivation layer of the device can remain intact. The attack complexity increases as packaging size shrinks, such as the ball grid array (BGA)-packaged devices.

2.4 Leakage Model

A leakage model is a function used to quantify the leakage of a set of operations from the device via side channels, such as the *Hamming weight/distance*, that are common

leakage models in power SCAs. In the leakage model, the states of the device (e.g. the intermediate values of registers) can be fed as the inputs and the outputs of the function represent the leaked messages. The purpose of building a precise leakage model is to accurately predict the side-channel leakage from a hardware platform for efficient attacks. In other words, a good leakage model can improve the capability of detecting leaked information.

Different physical implementations need different models for efficient leakage explorations. In this section, power analysis is taken as the example to describe different types of leakage models, but the analysis discussed here is adaptable to other leakage types, like EM emissions, sound, etc. Based on the number of operations during leakage observation, the leakage models can be classified into two categories: **univariate leakage** and **multivariate leakage**. Meanwhile, to reduce the technical complexity of leakage modelling, some explorations are provided to answer the question "*Can any data analytic techniques be alternatives to leakage modelling to provide meaningful leakage estimation*"? As this discussion is beyond the scope of this dissertation, it is accepted that leakage modelling is indispensable for efficient SCAs.

2.4.1 Univariate Leakage

In univariate leakage, only one instantaneous leakage (one operation) is observed. Different distinguishers can be used in univariate attacks, such as difference-of-means (DoM), Pearson's correlation coefficient, or the Guassian template, where the same statistics are essentially explored by these distinguishers [82].

Specifically, given an unprotected hardware/software implementation, when the intermediate state Y generates leakage L at time t, the distinguisher targets identifying the critical time point that can provide the maximum possibility to distinguish the correct key hypothesis. For example, if Pearson's correlation coefficient is utilised (CPAs), the goal is to find the key with the highest correlation coefficient between the real measurements and hypothesised values across the whole time period. If DoM is utilised (differential power attack), the goal is to locate the time instance that can maximise the variance between two sets of leakage for all time samples.

For example, *Hamming weight* is a well-known univariate leakage model used in CPAs, where the function returns the Hamming weight (HW) of the input Y. The HW value represents the estimated leakage and can be used to calculate the correlation with the measured leakage L.

2.4.2 Multivariate Leakage

Multivariate leakage is much more complicated because a vector of operations have to be considered. It is normally used in higher-order attacks, where multivariate statistics are required. For example, *HD* is a common bivariate leakage model as the function returns the HD between two operations.

However, in most published work on high-order attacks, the multivariate leakage is usually converted into a univariate signal through some preprocessing steps, which is then followed by established first-order attack techniques. For example, in high-order DPAs on masked encryption algorithms, abs-diff-DPA or product-DPA can be utilised as pre-processing processes prior to a standard DPA [82]. Although the pre-processing functions can be combined with a masked leakage to reveal the secret key, there are unavoidable challenges for attackers [83]: 1) how to select the points of interest?; and 2) how to find the optimal transformation when it varies with different leakage models?

Based on these observations, research has been conducted to evaluate the multivariate statistics directly with pre-processing. For example, multivariate mutual information analysis (MMIA) can be adapted to evaluate the joint statistics directly [84].

2.5 Adversary Model

Based on the familiarity level of the adversaries to the cryptographic system and its execution environment, the adversary models can be classified into three categories: black-box model, white-box model and grey-box model.

2.5.1 Black-Box Model

In the black-box model, the attackers have no or limited control over the execution process and environment. They can only observe the input and output of the cryptographic system, with the algorithms publicly known, as shown in Fig. 2.3.

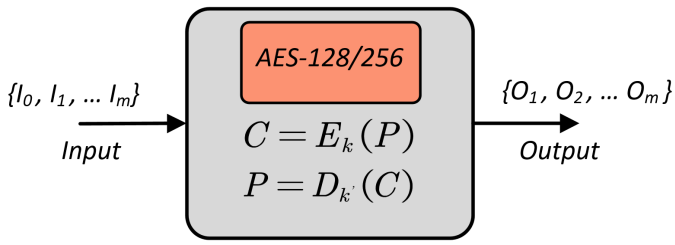


Figure 2.3: Black-box model.

The characteristics of the black-box model are listed as follows: 1) attackers know the encryption and decryption algorithms (including the keys) completely; and 2) attackers can access the software implementation and produce input-output pairs ($\{I, O\}$) by using the algorithm.

Since the attackers have no or very limited knowledge of the internal structure and parameters of the crytographic system, the secrete data can only be inferred by crude methods such as brute-force attacks, or by SCAs. However, the brute-force attack becomes considerably complex if the search space is too large, such as the key candidates for the 128-bit AES using a 128-bit key. The SCA also becomes difficult if the internal information of the system is poorly known, which lower the correlation between the leakage and the sensitive data. Therefore, the black-box attacks are usually less efficient and accurate than the other two models, but more realistic in many scenarios.

2.5.2 White-Box Model

The white-box cryptographic implementations are initially motivated to protect the key in a white-box environment, where the adversaries have full controls over the execution process and environment.

The characteristics of the white-box model are listed as follows: 1) attackers have full accesses to both software and hardware implementations; 2) the intermediate values that are related to the internal behaviours of the circuit can be observed or modified; and 3) the data in memories or caches can be retrieved or altered.

The white-box attacks can provide valuable insights into the advancing technologies of attacks and protections in SCAs, such as identifying the root causes of the SCAs and then suggesting efficient methods to protect the system from SCAs. Furthermore, cryptographic applications that are proven to be practically resistant to white-box attacks, also known as the worst-case (WC) scenarios, can be regarded as secure systems in a completely untrusted execution environment.

2.5.3 Grey-Box Model

The grey-box model lies between the black-box and white-box models, where the adversaries have partial control over the execution process and the environment. The grey-box model is a well-established model in the security of cryptographic systems, which is also the model adopted in this dissertation.

The characteristics of the grey-box model are listed as follows: 1) attackers can

access to the physical implementations of the cryptographic system and have no limitation to use the device; 2) side-channel leakage from the device can be monitored or collected by sending certain patterns of inputs; and 3) faults can be injected into the algorithm or the circuits to reveal the secret information.

Due to the emerging development of embedded devices, the implementation details of cryptographic functions or primitives can be hidden/masked/obfuscated through various methods against SCAs. For instance, the secret keys used in *round 0*, *round 1-9* and *the last round* of AES can be separately encoded using a look-up table (LUT) (LUTs 1-3) which is unknown to the attackers (a grey-box model), as illustrated in Fig. 2.4. In this case, the attackers aim to evaluate whether the encoding technique can provide adequate protection to hide the correlation between the leakage with the key. Therefore, the emphasis of grey-box attacks are put on the cryptographic implementation rather than the algorithm itself.

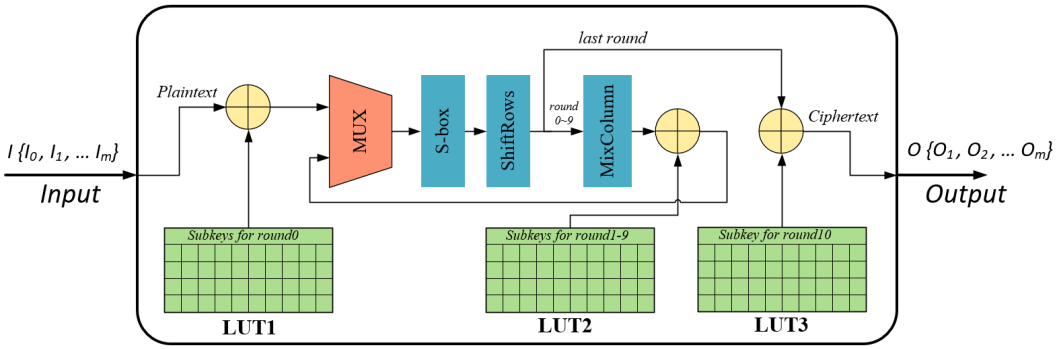


Figure 2.4: Grey-box model.

2.6 Security Validation

Normally, some meaningful figures of merit should be considered for the security evaluation against SCAs for a system. In this dissertation, SNR, test vector leakage assessment (TVLA) and guessing entropy (GE) are considered as the main figures of merit for evaluating the systematic security based on the correlation coefficient values, leakage variances and number of key guesses, respectively.

2.6.1 Signal-Noise Ratio

The first criterion for the security evaluation is SNR. In SCAs, SNR is characterised as an essential parameter to estimate the information leakage in side channels, which

affects the attack capability. In this dissertation and other security related papers, SNR is defined as the ratio between the correlation coefficient of the correct key ρ_{corr} and the maximum correlation coefficient of the incorrect key ρ_{incorr} for all samples in time, as follows [85],

$$SNR = \frac{\rho_{corr}}{\rho_{\max,incorr}}. \tag{2.1}$$

When SNR is greater than 1, the system is regarded as insecure as the correct key can be successfully identified by attackers. However, when SNR drops below 1 in the simulation environment, a system is considered to provide high immunity to SCAs in real-world scenarios with a high probability where noise is considered [85].

2.6.2 Test Vector Leakage Assessment

TVLA is a leakage measurement proposed by Goodwill [86]. It is a generic test methodology without expensive equipment, which can provide the designer or the attacker with an initial evaluation of the internal data dependency in the captured leakage.

Welch's t-test is a widely used scheme in TVLA, where the variance of the mean values of two randomly collected traces are calculated. Take the power attack on the AES algorithm for example, the TVLA based on t-test includes the following three main steps:

1. Collect two sets of power traces with different inputs, where, in one set, both the plaintexts and key are fixed, whilst in the other set, the plaintexts vary randomly and the key remains unchanged.

2. Use the formula $t = \frac{\overline{S_1}-\overline{S_2}}{\sqrt{\frac{\sigma_1^2}{N_1}+\frac{\sigma_2^2}{N_2}}}$ to evaluate whether the statistical information of the two sets of traces are distinguishable, where $\overline{S_1}, \overline{S_2}$ are the mean value of the two sets, σ_1, σ_2 are the standard deviations of the sampling points of the two sets, respectively, and N_1, N_2 are the number of sampling points. The threshold is set to 4.5.

3. If the maximum t-test value at some time points exceeds 4.5, it can be concluded that the leakage is related to the internal data with a possibility of 99.9999999%.

However, TVLA may lose its leakage evaluation function for a low-noise environment [87] and, hence, it is mostly adopted in the attacks happening in a prototype environment rather than a simulation environment.

2.6.3 Guessing Entropy

Guessing entropy (GE), a metric derived from information theory, is employed to quantify the strength of the information leakage from cryptographic circuits. GE is defined as the average number of attempts an attack needs to correctly guess a secret key based on the leaked information [88]. Higher GE indicates a stronger resistance to SCAs, as it implies a larger search space for potential secret values. The aim of EM SCAs demonstrated in this dissertation is to obtain the key K that maximises the subsequent conditional probability,

$$K = arg\max\left(P_r\left[V_m\left(t\right)|H_m\left(t\,|k\right)\right]\right), k \in \left[0, 2^n - 1\right], \tag{2.2}$$

and

$$\begin{aligned} &P_r\left[V_m\left(t\right)|H_m\left(t\,|k\right)\right] \\ &= \sum_{D_{t,t-1}} P_r\left[V_m\left(t\right)|H_m\left(t\,|k, D_t, D_{t-1}\right)\right] \cdot P_r\left[D_t, D_{t-1}\right], \end{aligned} \tag{2.3}$$

where $V_m\left(t\right)$ is the voltage measurement at time point t, $H_m\left(t\,|k\right)$ is the EM leakage hypothesis at time point t based on the guessed key k, and n is the key bit number. D_t and D_{t-1} are, respectively, current data at time point t and previous data at time point $t-1$. $P_r\left[D_t, D_{t-1}\right]$ is the probability of the HD between two consecutive pieces of data, D_t and D_{t-1}. Taking 16-bit bus data ($n = 16$) for example, the probability distribution of the HD between D_t and D_{t-1} is illustrated in Fig. 2.5.

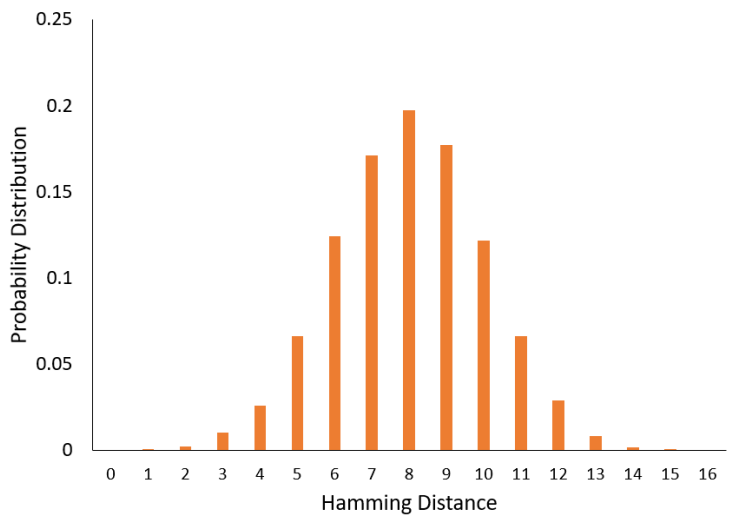


Figure 2.5: The probability distribution of the Hamming distance between D_t and D_{t-1}.

The success rate (SR) of EM attacks, denoted as $SR^{k,D_{t,t-1}}_{V_m(t),H_m(t)} = Pr(K = \Bbbk)$, represents the probability that K corresponds to the correct encryption key $\Bbbk$. GE is determined by sorting the correlation coefficients of all possible keys, represented as vector $\mathbb{C}$, denoted as $\left\{C_{k_1}, C_{k_2,...}C_{k_{2^n}}\right\}$. GE is then defined as the rank of the correct key within the set of all guessed keys [88],

$$GE^{\Bbbk,D_{t,t-1}}_{V_m(t),H_m(t)} = E\left[rank_{\Bbbk}(\mathbb{C})\right], \tag{2.4}$$

where $E\left[rank_{\Bbbk}(\mathbb{C})\right]$ represents the index of the correlation coefficient of the correct key $\Bbbk$ within vector $\mathbb{C}$. The index signifies the relative position of the correct key within all guessed keys.

From Eqs. (2.2) to (2.4), it can be deduced that GE serves as a valuable metric for estimating the average number of key guesses required to recover the correct key.

Comparison of different metrics

The pros and cons of the attack metrics mentioned previously are briefly discussed here before moving to the next section. GE is an information-theoretic metric based on the mutual information of the leakage, whereas TVLA and SNR are statistical metrics based on the variance of the measurements. On the other hand, TVLA can provide a quick pass or fail solution for the evaluation of the system security but cannot provide the quantitative amount of leakage. In contrast, GE and SNR can quantitatively and accurately assess the amount of leakage in a time-efficient manner. Different metrics can be flexibly chosen in different cases according to their respective properties.

2.7 Countermeasures Against Side-Channel Attacks

Once a SCA is detected, a proper mitigation technique can be deployed. Although none of the techniques can provide perfect protections against SCAs, they can make the attacks sufficiently complex and difficult. Existing countermeasures can be broadly classified into either high-level or low-level types based on their abstraction levels. As listing all possible solutions to protect the devices from SCAs would be tedious and requires a long survey, in this section only some typical examples of the two types are introduced with details. Table 2.2 gives an overview of the most important countermeasures, which level they belong to, overhead, security enhancement and whether they have been physically implemented, respectively.

Table 2.2: Countermeasures against side-channel attacks in different levels (high-level and low-level) with their respective overhead, security enhancement and physical application.

Category	Mitigation methods	Overhead	Attack complexity (timing, evaluation metrics)	MTD[1]	Practically applied
High-level	Timing [89]	performance: $< 27\%$ area: 2%	leakage: reduced by 92%	—	RISC-V, FPGA Processor
	Hiding [90]	performance: $> 4.57\times$ energy: $4.5\times$	security: $> 20\times$	> 10240@50MHz	FPGA, Processor
	Masking [91][2]	area: 13.3% frequency: 4.67%	793.2% enhancement of execution cycles	—	FPGA, Processor
	Re-keying [92][2]	protocol: increase 1 pass area: 38.4% performance: 32∼1005 cycles	time complexity 2^{60}	5×2^{44}@20MHz	ASIC Microcontroller
Low-level	Flattening [93]	area: 73.1%	NED: 1.0 to 0.032 NSD: 0.293 to 0.006	—	FPGA, ASIC Processor
	Randomisation [94]	timing: $58.9\times$ power:$1.01\times$	—	5×10^6@17.5-426.5MHz	FPGA, ASIC
	Hiding [95]	area: $< 40\%$ power: $< 35\%$	—	3000@100MHz	FPGA, ASIC
	Isolation [96]	memory: LLC 2%@32bits-core area: $< 15\%$	instructions: 9,475,000 cache accesses: 82,000	—	Cloud
	Shielding [97]	power: $< 5.7\%$ area: very small	SVF[3] < 0.05 STSF[4] < 0.59	—	Processor Microcontroller

[1] MTD: Minimum traces to disclose
[2] Analysis based on third-order techniques
[3] SVF: Side-channel vulnerability factor
[4] STSF: Spatial thermal side-channel factor

2.7.1 High-Level

High-level (or code-level/algorithmic) countermeasures are setting mathematical bounds in the code for the information leakage, which can be shown with a selected list below:

Timing

The running time of a cryptographic algorithm can be a side channel to reveal the secret keys. The rationale is that the execution time varies with different execution paths or cache hits, which is often key-dependant. In order to prevent the time leakage, techniques can be directly applied to the clock. The clock cycles can be misaligned or randomised to hide the timing behaviours. The running time can also be obfuscated by cloning the entire branch, inserting repeated/dummy operations or random interrupts. Moreover, random delays that are generated by a delay producer can be inserted into the communication links between the shared memories.

Hiding

As discussed earlier, the execution time of certain operations may vary depending on the input data (such as modular exponentiation, look-up tables (LUTs), speculative execution and branch prediction), which can be exploited by an attacker to infer secret

information. A possible countermeasure is to use a constant-time function or algorithm that does not exhibit any data-dependant timing variation. For example, a time-difference remover delays the return time until a fixed amount of time has elapsed, regardless of when the computation ends. However, constant-time functions or algorithms are not easy to implement for some cryptographic systems (such as high-speed AES running on general-purpose computers) and may incur a performance overhead. Time-difference removers may increase hardware complexity and system latency.

Masking

A masking scheme is a technique that randomises the data that depends on the secret key with some values (called *masks*) to make it unpredictable. The principle of masking is to split the sensitive variable into several pieces (also called **shares**), such that the original variable can be reconstructed by combining all the pieces. The idea of masking is to prevent an attacker from observing the whole variable at once and thus reduce the amount of information that can be leaked through a side channel. For example, a power SCA can measure the power consumption of a device when it performs cryptographic operations using a key k, and use this information to infer k. To prevent this, k can be split into k_1 and k_2, such that $k = k_1 \oplus k_2$, where $\oplus$ denotes bitwise XOR. Then k_1 and k_2 can be used instead of k in the cryptographic algorithm and are not combined until the end of the computation. Therefore, an attacker can only observe the power consumption of k_1 and k_2 separately, not k itself.

If a sensitive variable is split into $d+1$ shares, d is referred to as the order of masking. The higher the masking order, the more secure the scheme is against SCAs. First-order masking is resistant to simple SCAs but not to DPAs. In order to mitigate high-order DPAs, a high-order masking scheme is required. The hardware-based high-order masking can provide security improvement but with large area overhead, while high-order algorithmic masking can enhance the security and, meanwhile, be area-efficient. For example, a third-order masking scheme that can protect an AES algorithm from a high-order DPA has been proposed, which can also achieve a significant reduction of 88.9% in the execution cycles and 70.5% in the area [91].

Re-keying

The re-keying scheme was first proposed in 2010 [92], which aimed to protect the radio-frequency identification (RFID) tags from SCAs for the fast-growing RFID application market. Due to the limitations of the silicon area and power of the device, the

countermeasure is restricted to the protocol level (e.g. re-keying). The main idea of re-keying is to derive a new session key $k^{'}$ from the master key k and an on-tag generated random value s, where the fresh session key $k^{'}$ is then used to encrypt messages. This implies that the encryption data is never encrypted by the same key, which raises the attack difficulty. The generation process of the random value s was further improved by adding more diffusion or complexity in subsequent works.

2.7.2 Low-Level

Low-level (or physical-level) countermeasures are focused on reducing or eliminating the leakage from the root. In this subsection, these methods are divided into **flattening**, **randomisation**, **hiding**, **isolation** and **shielding**, according to their function.

Flattening

Flattening (or balancing) logic is to use a modified circuit to balance the bit transitions and make the circuit activities constant. For example, if the logic circuit transitions from 0 to 1, there is another complementary circuit transitioning from 1 to 0 at the same time. Therefore, by using adequate balancing techniques, the adversaries can hardly tell which transition pattern occurs. Various logic styles have been developed in recent decades. An example is the dual-rail precharge logic (DRP), it has two outputs (O and O') and operates in *pre-charge* and *evaluate* phases. In the *pre-charge* phase, the logic circuit is set to a state and in the *evaluate* phase, it gives an output O or O'. The circuit is designed to make the two outputs have the same capacitance and would emit the same amount of leakage for a dynamic transition [98]. The later sense amplifier-based logic (SABL) [93] and wave dynamic differential logic (WDDL) [99] both belong to the DRP logic style.

Except for using the standard-cell design mentioned above to flatten the power/current consumption when bits transit, external regulator circuits can be placed between the cryptographic IPs and the power supplies, such as the current equaliser and induction voltage regulator (IVR), to equalise the current/voltage profile.

Randomisation

The cache-based SCAs can be mitigated by cache partitioning and randomisation of mapping between the main memory and the cache, but the architecture design is complicated by the hardware modifications of the processor. This issue can be overcome

by applying an architecture-level randomisation technique that eliminates the SCA on the last-level cache without altering the hardware [100].

This survey [101] reviews various hardware randomisation techniques that can reduce the leakage of information. These techniques include randomising the occupation of flip-flops and data-path elements, random shuffling of sub-operations, inserting random-delay first-in-first-out buffers (FIFOs) into the pipelined architecture, and randomly selecting and executing functional units. These techniques are applicable to different types of hardware platforms (ASIC, FPGA, soft gate array, etc.). For example, in FPGA, four additional random architecture-level methods are available: randomising the location of pipelining registers and datapath elements; adding random power consumption using LUTs, block random access memories (BRAMs) or digital clock managers (DCMs); and reconfiguring LUTs for Sbox randomisation at a fine-grained level. These methods can increase the complexity of physical tampering by reducing the data dependencies of the leakage.

Hiding

Noise injection is a hiding scheme that can lower the SNR of the side-channel information and thus increase the attack complexity. A quantitative analysis shows that the operating frequency and power delivery network (PDN) influence the SNR of the measured supply currents, implying that frequency-dependant noise-injection into the PDN can enhance device security against SCAs [95]. To reduce the area and power overhead caused by the noise injection circuit, the noise can be injected into the signature after it is sufficiently attenuated. More details about this method will be provided in subsection 2.9.1.

Another hiding scheme is dynamic voltage and frequency scaling (DVFS), which is a common technique to decrease energy consumption. For both a multi-level switched-capacitor (SC) based distributed on-chip power delivery system [102] and a Xilinx ZYNQ UltraScale + FPGA platform [94], DVFS can improve the security level of the cryptographic core by introducing a random delay between the input data and the power consumption to obscure their correlations.

Isolation

Isolation is one of the security principles to protect the device from physical tampering attempts. To avoid cache SCAs, Intel developed a technique called cache allocation technology (CAT), which can be adjusted to separate the LLC into a secure and a

non-secure partition and force the secure part to be loaded with the secure page [96]. Moreover, in order to mitigate the software-based SCAs on enclaves, enclaves can be implemented on separate secure cores and, thereby, physically isolated from running on the victim core [103].

To prevent the adversaries from monitoring the current fluctuations from the external power supply pin/grid, a detachable power supply can be used for the cryptographic block. Local capacitors (fully charged) can be connected to the cryptographic circuit and support the encryption process. Afterwards, the capacitors can be re-connected to the external power supply and charged for the next use [104]. Thereby the data-dependant current fluctuations can hardly be observed by the attackers.

Shielding

Addition of shields can prevent the side-channel information leaking from the root. For CMOS devices, all the transistors are fabricated on the substrate layer. According to the material characteristics of the substrate, the back-side substrate layer can have a shielding effect on EM emissions induced by current following through on-chip wires [105]. A thermal-aware shielding technique, 3D-TASCS, can dynamically generate different shielding patterns to conceal the security-related activities on the chip [97].

2.8 Electromagnetic Side-Channel Attacks

An EM SCA is a generic non-invasive attack method that is the main concern in this dissertation, and which was first proposed by Quisquater in 2001 [106]. As EM attacks have the same nature as power attacks, both being induced by current fluctuations, EM-attack analysis also includes simple electromagnetic attack (SEMA), differential electromagnetic attack (DEMA), correlation electromagnetic attack (CEMA), etc. EM attacks have been physically implemented on smart cards, FPGA, chips, Internet of Things (IoT) devices [106], [107] etc. In this section, EM attacks are classified into two types: near-end attacks and far-end attacks, according to the distance between the probing equipment and the targeted device.

The traditional method to distinguish near-end from far-end is: if the location lies in the radiation region of 1 wavelength of the source (near-field), it is near-end; if the location lies in the radiation region farther than 2 wavelengths of the source (far-field), it is far-end, as shown in Fig. 2.6 [108]. In EM SCAs, the near-end attacks require close proximity to the device and the far-end attacks can operate at larger distances,

for example, several metres or even hundreds of metres away from the device. In the following subsections, more details and examples are listed to demonstrate the experimental results of these two groups of EM attacks.

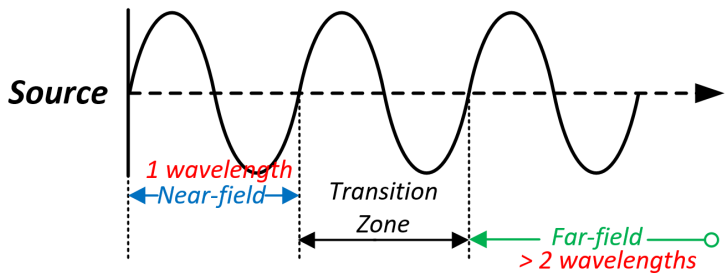


Figure 2.6: Traditional definitions of near-end and far-end.

2.8.1 Acquisition of Electromagnetic Emissions

There are two popular candidates used to capture the EM emissions, which are probe and software-defined radio (SDR). The EM traces can be collected by a simple and low-cost probe that is brought close to the surface of the device under test (DUT), while for the wireless attacks from a distance, the more complex SDR is widely applied.

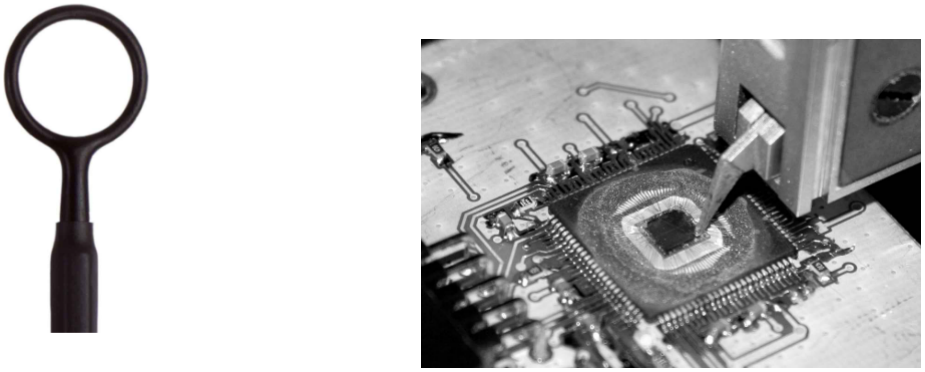
Probe

A probe is an effective tool for the near-field coupling, which can be used to test radiation intensity around these locations or to find the radiating source. It is also the main tool used in this dissertation. The probe can be as simple as a coaxial loop (as shown in Fig. 2.7(a)) or have a high resolution (as shown in Fig. 2.7(b)). In the following, the calculations of the intensity of the EM radiations around the target (chip surface or bus surface) and the amplitude of the detected voltage at the probe terminal are demonstrated, respectively.

The transient EM radiations are proportional to the number of space-time current samples. Assuming the surface current is induced on S, the transient magnetic field radiated by the currents in a homogeneous medium can be calculated by [109]

$$\vec{H}(\vec{r},t) = -\nabla \times \frac{\vec{A}(\vec{r},t)}{\mu} = -\nabla \times \iint_S \frac{\mu \vec{J}\left(\vec{r}',t-\frac{R}{c}\right)}{4\pi R} ds', \tag{2.5}$$

where μ is the permeability, $\vec{A}(\vec{r},t)$ represents the vector potential, $\vec{J}(\vec{r},t)$ is the



(a) (b)

Figure 2.7: (a) A Langer EMV RF-R 50-1 near-field probe [1], and (b) a Lecroy probe with a built-in 30 dB pre-amplifier that can reach a step size of 100 μm [2].

surface current density, R equals the distance between the source point $\vec{r}$ and the observation point $\vec{r}'$, and c refers to the light speed. From Faraday's law, the amplitude of the voltage signal sensed at the probe terminal is determined by the magnetic flux through the probe surface (assuming the effective surface area is S_p), which is given by

$$V(t) = -\frac{d}{dt}\iint_{S_p} \mu_p \vec{H}(\vec{r},t)\, ds, \tag{2.6}$$

where μ_p is the probe permeability and $\vec{H}(\vec{r},t)$ can be calculated from Eq. 2.5. During the emission test, the distance between the probe and the target surface and the angle between the probe and the current direction can affect the magnetic flux through the probe and further affect the voltage amplitude. When the probe is placed in different locations $(l_1, l_2, ...l_n)$ surrounding the target, a total number of n EM traces $(V_{l_1}, V_{l_2}, V_{l_n})$ can be collected.

Software-defined radio

SDR is a promising technology in communication systems, which replaces the majority of hardware components (such as the filters, mixers, dividers, etc.) with software implementations, as shown in Fig. 2.8. It only contains the minimal hardware components to tune in to a wide range of RF frequencies and digitise them with high-speed analogue-to-digital (A/D) converters. The digitised radio frequency (RF) data are entirely processed by advanced digital signal processing techniques, in other words, by

software. Compared with the separate hardware design for different standards in traditional communication systems, SDR can be easily configured to fit multiple standards and is, therefore, more flexible.

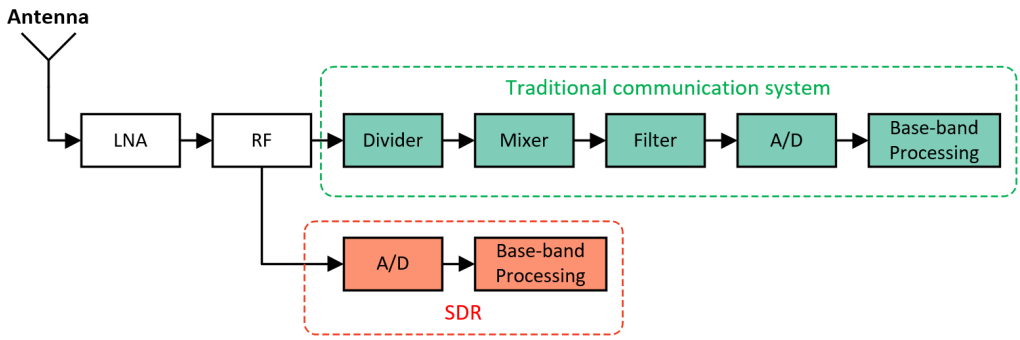


Figure 2.8: Comparison between a traditional communication system and software-defined radio.

Due to the enhanced flexibility of SDR, it is gaining more and more popularity among EM side-channel attackers as it is a perfect candidate to access the radio frequency spectrum from greater distances. With a simple set up, SDR can be used to capture the EM radiations [51]. For more precise sensing systems, more equipment is required, where an SDR can be combined with an antenna and a low-noise amplifier (LNA). Moreover, SDR can sweep a wide range of RF frequencies to help locate the useful EM emissions, which can increase the attack effectiveness and meanwhile, reduce the cost of using multiple radios to sense EM signals with more bandwidths. The often used SDR software and hardware in EM attacks mentioned in the literature include TempestSDR, universal software radio peripheral (USRP) and HackRF [110].

2.8.2 Signal Sampling

Currently, a digital oscilloscope is the main equipment for analysing the captured EM traces. Due to the high-speed A/D converter, the digital oscilloscope can quickly convert the analogue signals obtained from the front-end antennas into digital signals. The upper bound for the sampling rate is often twice the bandwidth of the equipment.

Normally, more frequency components of the EM emissions are prone to be collected, to avoid missing the useful information and meanwhile improve the attack effectiveness. Therefore, a high sampling rate is required for localised EM attacks, and nowadays the sampling rate of a digital oscilloscope can reach as high as 10 gigasamples per second (GSa/s). After the side-channel traces are recorded, the digital

signals can be sent to the computer for statistical analysis or signal processing for further confirmation.

2.8.3 Near-End Electromagnetic Attacks

Near-end EM attacks target direct EM emanations generated by the **intentional** current flow. Normally, sharp transitions happening on the edge of current pulses can generate a wide-band EM spectrum, which can be captured by closely placed sensors. When attacking complex or tiny circuits, high-resolution sensors are required to identify areas of interest and localise the useful signals to improve attack efficiency.

Initially, EM attacks can happen on consumer electronic devices, for example:

- Sensors are attached to the residential power lines of modern TVs (liquid crystal display (LCD) or plasma) to record EM traces when videos are played. The attackers build a database of EM signatures and match the collected traces with the pre-stored database, where the content of the screen can be correctly indicated [75].

- The power cable of the wired PS/2 computer keyboards can be detected, where there is information leaking from the power/data line to the ground line. As the clock frequency of the PS/2 signal is lower than the frequencies of other devices on PCs, the leakage of the PS/2 can be easily filtered out from other signals [77].

- For modern flat-panel displays, compromising emissions can be detected from the cable that connects the display module and the graphic control module, by using a nearby probe combined with an amplifier and a recording device. Furthermore, the bit-transition patterns (equalling colour combinations) that generate significant field levels can be set for attacking the displayed texts [76].

With the increase in functionality and complexity of embedded systems, larger, more compact and complex systems become the new targets of EM attacks. For chip-scale EM analysis, high-resolution sensors are required to identify areas of interest and localise the useful signals to improve attack efficiency. Some examples are listed as follows,

- Ring oscillator (RO) based physical unclonable function (PUF) is regarded as effective and popular solutions to protect IPs/keys on FPGA due to their ability to extract a high PUF response. However, the extraction and location of the

frequencies of RO can be identified through EM attacks, where the RO based PUF can be fully characterised [111].

- The sensitive key of data encryption standard (DES), implemented on an ASIC chip, can also be recovered through EM attacks. For example, the subkey used in the first Sbox of the first round of DES can be attacked by collecting the EM emanations when it operates [112].

Apart from observing the EM emanations passively, EM pulses can also be actively injected into the components of a targeted circuit from the near field. It has been proven in several publications that EM faulty pulses can be introduced into 32-bit microprocessors [113], FPGAs running AES [114], reduced instruction set computer - five (RISC-V) microcontrollers running AES [115], and a Cortex-M microcontroller transferring the data from the flash memory to the data buffer [116], to achieve successful attacks. Meanwhile, the clock speed of the targeted embedded devices has increased from 200 MHz to over 500 MHz. The higher the clock speed, the harder to produce a precise EM injection into the specific clock cycle glitch. State-of-the-art harmonic pulses (up to 1 GHz) can be generated and applied to modern embedded systems, such as the widely-used TrustZone-based secure boot implementation on a multi-core ARM, for the faulty injection attack. In order to demonstrate the effect of EM glitch injection on a state-of-art microcontroller, a register-transfer level fault model has been built [117].

2.8.4 Far-End Electromagnetic Attacks

Far-end EM attacks aim to reveal the operations of and data processed by devices from **unintentional** EM emissions, where the emissions can be amplitude-modulated, angle-modulated, frequency-modulated or unmodulated.

The EM emissions from computer keyboards (PS/2, USB, wireless or laptop) can be captured to recover the sensitive information, such as the keystrokes [77]. Moreover, elliptic curve Diffie-Hellman (ECDH) is a standard public-key encryption algorithm used in OpenPGP (a popular email encryption for all operating systems), defined by the Internet Engineering Task Force (IETF). The decryption keys of ECDH running on PCs can be attacked by capturing the EM emanations from the room next to the personal computer (PC) location [118]. With a carefully chosen ciphertext and the frequency modulation (FM) demodulation technique, the confidential information can be obtained within seconds.

The unauthorised EM leaks from mobile phones, with a different global system for mobile communications (GSM) modules (1G, 2G, 3G, 4G, etc.) [119], can be used to attack the elliptic curve cryptographic algorithm. The running apps can be identified by measuring the intensity of the EM field around the phones for different operating modes. Another example is that the status of the phone camera (front or back) can also be identified by investigating the EM emission patterns of the cameras working at different states.

The mixed-signal chip that runs the AES algorithm can also become the target of far-field EM attacks, as shown in Fig. 2.9. The switching activities of the AES encryption process (clock frequency f_{clock} can produce some noise that can be coupled into the analogue part (radio-frequency circuit) through the substrate and the power supply grid. The noise generated by the AES encryption can be amplified, modulated, and transmitted together with the radio frequency (f_{RF}). This modulated signal can be received by an antenna placed far away from the board (10 metres). Through demodulation, the encryption keys of AES can be attacked.

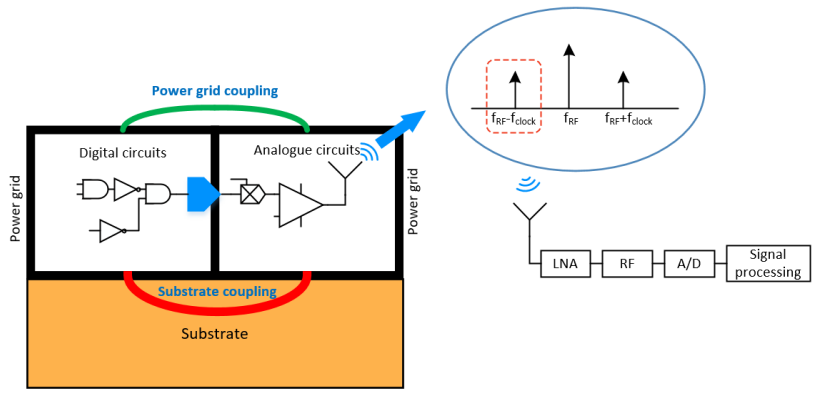


Figure 2.9: Electromagnetic side-channel attack on mixed-signal chips.

At the receiver side, the narrow-band filters are mostly adopted to pick out the useful part from the raw signals. However, in some cases, wide-band filters are used to receive the original signals as much as possible for the purpose of not missing any critical information, where spectrum analysis is needed for post signal processing [120].

2.9 Countermeasures Against Electromagnetic Attacks

With the emergence of EM attacks on cryptography applications, especially due to the available commercial probes and popular IoT area, numerous countermeasures have

been proposed against such threats over recent decades. These promising techniques designed to thwart EM attacks can be divided into two main areas of **hiding** and **masking**, which will be specifically described in the next subsections, respectively. Finally, the comparison of different countermeasures in different attributes (platform, overhead, attack technique, attack modes, etc.) is presented in Table 2.3.

2.9.1 Hiding Techniques

Hiding techniques aim to harden the extraction of useful signals by suppressing the SNR of the EM emission measurements to a very low value. As decreasing the SNR value can be achieved by reducing the signal or increasing the noise, the techniques fall into two main categories: *signal strength reduction* and *noise strength improvement*.

Shielding

Shielding is a basic and most effective technique to prevent EM information leakage, which occurs due to either absorption or multiple reflection. For the far-field attacks that can happen across a wall, the devices can be placed in a shielded room to block the emission-capturing. Shielding an individual component seems very difficult due to the complexity of the circuit; however, internal metal shielding can be inserted into the proper layout of circuits to shield EM radiations from the circuit surface.

For example, the two metal-layer metal-insulator-metal (MIM) capacitor, with a parallel-plate structure, has been proven to be an appropriate shield to reduce the EM emissions by 33 dB and 11 dB from top to bottom [121]. However, this method usually results in area overhead due to the effectiveness of the shielding being closely related to the thickness and size of the shielding layer.

Twisted/distributed power grids

It is known that the EM leakage from the circuit depends on the design of the power delivery network. Normally, the on-chip power/ground (P/G) grids are placed in parallel and the current in the power supply line flows in the opposite direction to the current in the ground line. The magnetic field surrounding these metal lines may be partially cancelled in the far end but add up in the middle end. Therefore, it is possible to sense the underlying data-dependant emanations through fine-grained EM attacks at an optimal position [122].

In order to improve the resilience of the circuits against EM attacks, the topology and characteristics of the power delivery network need to be optimised. For instance, the space between the P/G grids can be adjusted non-uniformly rather than uniformly to reduce the EM emissions [123]. Furthermore, in order to mitigate fine-grained EM attacks, the adjacent P/G grids can be twisted. The loop area surrounded by the power grid and the ground grid is minimised and the magnetic field is thus attenuated, where the overall EM radiation is reduced.

Distributed power delivery networks can also help resist EM attacks. For instance, on-chip regulators can be implemented in the lower-level metal layers, placed close to the cryptographic circuit, and supply power to the cryptographic circuit from the local power grid. The long power delivery interconnects are then reduced to shorter local wires, which result in less EM leakage and make the probing detection more difficult.

Another effective method is to flatten the current fluctuations of the P/G grid by adjusting the impedance of the P/G grid [124]. Extra capacitors can be added to specific nodes to equalise EM profiles, such as the *fluctuation* nodes (nodes with severe fluctuations) or the *antenna* nodes (nodes between the P/G network and supply pins). These capacitors, known as equalisation capacitors, can control the current fluctuations under a fixed threshold. As a result, the MTD improves by 1138x with an area overhead of only 0.62% and a power overhead of 1.36% [124].

Decoupling capacitance

The utilisation of decoupling capacitance can lower the current peak values and hide the correlation between the EM leakage and the data. In the physical implementations, the decaps can be implemented as metal-oxide-semiconductor field-effect transistor (MOSFET) capacitors (which can sufficiently charge the logic block over a small time period) and placed next to the logic gates of the cryptographic computation components. Meanwhile, the amount of decoupling capacitance needs to be carefully chosen, as different amounts of decoupling capacitance provide different suppression effects in the SNR value [125].

However, due to the fact that decaps can become the local source for the charging process in high-frequencies, they may fail to protect the cryptographic intellectual properties (IPs) against SCAs in such cases. The attack can have a higher success rate especially when the resonance frequency of the capacitance is designed close to the clock frequency. For example, the AES running on FPGAs (up to 66 MHz working clock) can be attacked by observing the current flow through the pin renders of the

decap [126]. Moreover, decap can also induce an area overhead as much as 20% [122].

Placement and routing

The top-level metal layers of a chip are highly radiating compared with other lower metal layers [15]. However, the critical signals can only be implemented in the **local** lower-level metal layers due to the high resistance of low-level metal wires. Specialised placement tools like CAD4EM-P [127], are proposed to guide the security-driven placement for the cryptographic cells. Once the critical cells are placed in the lower-level layers, the register-dependant EM leakage emitted from the top layers can be suppressed and blind an adversary from EM attacks.

Furthermore, on-chip regulators (such as buck converters, switched-capacitor regulators and linear regulators) can be packed together with the cryptographic circuit and further attenuate the EM signatures before they reach the top layers. Taking 128-bit AES running on the Atmega microcontroller for example, by combining a lower-level metal routing scheme with EM signature attenuation circuit, the MTD can reach higher than 1 million traces, with an area overhead of 23% and a power overhead of 50% and without performance degradation. For the 256-bit AES running on an ASIC chip, these two techniques can lead to a 100x MTD improvement, with a power overhead of 49% and an area overhead of 36% [128].

Besides the lower-level layer placement and routing, the clock network parameters of the circuit can also be security-driven adjusted during the *clock tree synthesis (CTS)* process. The total clock network is first drawn out from the netlist, then the vulnerable clock nets can be identified according to whether the flip flops related to the clock nets pass the cryptographic calculations or not. As a clock network normally includes parameters, such as clock skew, gate size, load capacitor, etc., by tuning these parameters, a new clock network that can reduce EM leakage can be regenerated without sacrificing other performances [129].

Noise injection

On-chip noise generators can be applied to devices to increase the complexity of attacks by lowering the SNR value. However, the noise addition is not an optimum option due to: 1) large area and power overhead; and 2) noise addition only is not sufficient to hide the useful messages completely. If sufficient samples were collected, it would still be possible to identify the bit value through statistical methods [120].

Therefore, noise addition is normally combined with other methods to provide the system with strong security protection. For example, after the signature attenuation, only a very tiny current injection is required to decorrelate the EM traces with the sensitive bit value [15].

2.9.2 Masking Techniques

Masking is a countermeasure that can be formally proven, where the internal computations are operated with random values (*mask*) to remove the dependencies between the EM signatures and the real data.

The output of the sensitive computation $f(x)$ is replaced by $f_m(x*a)$, where a is the mask value generated by a true random number generator, $*$ represents the operation (such as $\oplus$ in the Boolean masking scheme, $\times$ in the multiplicative masking scheme), and f_m is the masked output function. The input x can be further split into $d+1$ shares, which is d_{th}-order masking. Taking first-order masking for example, x is split into two shares (x_1, x_2, and $x = x_1 * x_2$), then the two shares are sequentially and independently processed with freshly-generated mask values a_1 and a_2, which finally gives an output $f_m(x_1 * a_1, x_2 * a_2)$. By splitting the internal result into d shares that are independent from each other, the attacker cannot obtain the sensitive data-dependant signatures from side channels.

For the sensitive blocks like Sbox in AES or loop PUF, the utilisation of high-order masking can thwart high-order DEMAs [130]. The area overhead induced by masking largely depends on the size and complexity of the mask table.

2.9.3 Limitations of Existing Works

In this section, a comparative analysis of the benefits and drawbacks of the various countermeasures against EM attacks that were discussed in the previous sections is presented. Moreover, the gaps and challenges in the existing literature on this topic are also identified.

Masking is less prevalent than **hiding**, as the former typically necessitates a true number generator to produce random values that are XORed with the sensitive data, which entails more area and power consumption.

Shielding can mitigate the EM emanations by adding a thick shielding layer, but it can also augment the coupling inductance between the shielding layer and local power grids, which can induce some current that may amplify the emanations.

Decoupling capacitance mainly isolates the main power supply that drives the logic gates that perform the sensitive calculations. However, this approach is more efficacious against power attacks than EM attacks, as EM leakage is often more informative than power traces and more fine-grained decoupling methods are required. To address this gap, capacitors can be added to some specific locations (nodes that have conspicuous current functions and nodes with P/G grid and supply pins) to flatten the EM missions. The main challenge of this method is the design complexity, which demands dedicated EM simulations to find these locations, and the area/power overhead under the layout constraints.

Lower metal layer routing can be integrated with noise injection and a low-dropout (LDO) regulator (a type of linear voltage regulator that can maintain a constant output voltage) [131] to suppress the EM radiations from top metal layers and flatten EM emissions concurrently, but at a high cost. The challenges of this method are the circuit complexity (more function blocks are required), area/power overhead ($> 20\%$), and performance degradation (lower metal layers imply thinner metals with higher resistances). Moreover, routing congestion may arise due to the routing of power lines occupying the resources of the signal lines.

2.10 Aims and Objectives

The aim of this dissertation is to investigate the vulnerability of interposer-based interconnects to EM attacks and to propose effective countermeasures that can prevent or mitigate such attacks without compromising the bus performance. The aim is accomplished by addressing the following objectives:

- **Efficient EM attacks**: to find the optimal location for placing the probe that can capture the maximum amount of information leakage from the interconnects; and, therefore, to minimise the number of measurements required to recover the secret key from the noisy EM signals.

- **Security enhancement**: to propose hardware-based protection schemes for the interposer-based interconnects that can increase their resilience to EM attacks. These schemes aim to obfuscate or decrease the correlation between the information leakage with the sensitive key.

- **No performance degradation**: to not degrade the bus performance (speed) and

in some specific cases, to improve the bus performance by exploiting the benefits of the applied techniques. Therefore, the trade-offs and synergies between security and performance for the interconnects will be explored.

- **Low power and area implementation**: to design and implement effective methods that can achieve lower power and area consumption, especially for the scenarios that have stringent constraints in these aspects.

- **Performance validation**: to evaluate and validate the effectiveness of the proposed methods, experiments will be conducted under different conditions. The performance metrics used to evaluate the results are MTD, SNR and GE. These metrics reflect the ability of the system to resist EM attacks.

2.11 Discussion

The concept and classification of SCAs, which are a type of attack that exploit the physical characteristics of a system, are introduced in this chapter. How the information leakage can be modelled and how the capabilities of the attackers can be defined are explained. Furthermore, three figures of merit (SNR, TVLA and GE) to measure the security level of a system against SCAs are discussed and compared, separately. These methods have different strengths and weaknesses: TVLA can rapidly identify the existence of leakage, while SNR and GE can accurately quantify the amount of leakage.

Additionally, a comprehensive review of the existing literature of the countermeasures against SCAs, with an emphasis on the power SCA as the most prevalent form of attack, is provided in this chapter. The countermeasures are classified into high-level and low-level categories, depending on their implementation level and design complexity. The suitability and compatibility of these countermeasures for various platforms, such as FPGA, ASIC, Microcontroller, cloud infrastructure, data centre, IoT devices, etc., are presented in Table 2.2. Moreover, the main themes of the subsequent chapters, which are the EM attacks and countermeasures, are introduced in this chapter. The state-of-the-art techniques for mitigating EM attacks are summarised in Table 2.3. The limitations and challenges of the current EM protection methods are also discussed, along with the aims and objectives of this dissertation.

This dissertation aims to, for the first time, investigate novel and efficient EM attacks and protections on the interposer-based interconnects in 2.5-D systems that are

widely adopted. An efficient high-resolution EM attack on an off-chip interconnect is presented in Chapter 3, which can exploit the EM leakage from the interposer layer. Two different performance-aware methods to protect the interconnects from EM attacks are proposed in Chapter 4.

Table 2.3: Overview of countermeasures against electromagnetic side-channel attacks. The methods are classified into **hiding** and **masking** two groups. DA: differential attack; SC: statistical correlation; HW: hardware; SW: software; MTD: minimal measurements to disclosure; CPA: correlation power attack; CEMA: correlation electromagnetic attack; DPA: differential power attack; DEMA: differential electromagnetic attack; LDO: low dropout regulator. ✓ and ✗ indicates whether an attack has been physically successful.

Category	Countermeasures	Platform	Area overhead	Power overhead	MTD (other features)	Fault model	Attack technique	CMOS technology	Target	Practically applied	Against attack modes
Hiding	Shielding [121]	Simulation (HW)	-	lower than off-chip regulator	(11-33dB reduction of EM emissions)	-	SC, DA	-	PDN	✗	-
	Decoupling capacitance [126]	FPGA (HW)	20%	-	>20000@66MHz	-	DA	-	AES-128	✓	DEMA
	Twisted power grids [124]	ASIC (HW)	-	-	>52500 (1.51x) @37.5MHz∼75MHz	-	SC	40 nm	AES-128	✓	CEMA
	P/G grid impedance adjustment [124]	Simulation (HW)	0.62%	1.36%	1138x@1GHz	-	DA	-	AES-128	✗	DEMA
	Lower-level metal layer [128]	ASIC (HW)	23% (128bits) 36% (256bits)	50% (128bits) 49% (256bits)	>1 million @50MHz (128bit) 100x@50MHz (256bits)	-	SC	65 nm	AES-128 AES-256	✓	CPA CEMA
	Specialised placement tool [127]	Simulation (HW, SW)	-	10.73% (AES) 10.59% (PRESENT)	5x@20MHz (AES) 5x@20MHz (PRESENT)	-	SC	180 nm	AES-128 PRESENT	✗	CEMA
	Noise injection + LDO [132]	ASIC (HW)	36.9%	32%	>6 million @80MHz	Current signatures	DA	130 nm	AES-128	✓	CPA CEMA
	This work	FPGA (HW)	160 transistors	0.1%	>100,000 @200MHz	-	SC	65 nm	AES-128	✓	CEMA
Masking	Random table masking [130]	ARM processor (HW)	200%[1]	145%[1]	10x@56MHz[1]	-	DA	-	AES-128	✓	3rd-order DPA & DEMA

[1] 2nd-order DPA/DEMA

Chapter 3

Electromagnetic Attacks on Interconnects

3.1 Chapter Overview

In this chapter, the key concepts, attack methods, and modelling required to demonstrate the vulnerability of the interposer-based interconnects to EM SCAs in 2.5-D systems, highlighted in Section 2.11, are introduced.

First, in order to simulate the EM SCAs on interposer-based memory buses, the EM field emanating from the bus lines needs to be observed via a nearby placed probe when data is transferred through the wires. Once sufficient traces are collected, statistical methods can be utilised to investigate the vulnerability of the bus to EM attacks. Preliminaries of an EM attack flow are demonstrated in Section 3.2, where the overview of the AES cipher and CEMA are presented, followed by the introduction of figures of merit to evaluate security and the CEMA attack flow in 2.5-D systems. A simple structure is used to model the interposer-based interconnects, demonstrating the EM radiation from the top metal layers, where the EM field surrounding the wires is also analysed in Section 3.3.

Compared with power SCAs, adversaries can benefit from EM SCAs through space localisation, where the attack hot spot can be found through high-sensitivity EM probes [80]. An algorithm used to determine the optimal position to place the probe, together with the search area reduction offered by the algorithm is described in Sections 3.4.1 and 3.4.2, respectively.

The effectiveness of the algorithm in EM attacks is verified by the simulation results described in Section 3.5. The implementation of mitigation techniques against

EM attacks on such interposer-based buses are the topic of Chapter 4.

3.2 Preliminaries of Electromagnetic Attacks

The EM SCA exploits multiple EM traces by using statistical analysis on the symmetric AES algorithm. The fundamental information on AES, CEMA, and figures of merit, together with the block diagram for the CEMA analysis on 128-bit AES, is presented in the following subsections, respectively.

3.2.1 Advanced Encryption Standard Cipher Overview

AES is an iterated symmetric block cipher that encrypts messages segmented into fixed blocks [133]. The cipher is symmetric because the same key is used for both the encryption and decryption process. Presently, AES is widely used for the encryption of sensitive data and, therefore, becomes the most commonly targeted algorithm of SCAs. When implemented for blocks of 128 bits and a 128-bit (or 256-bit) key, AES normally has ten encryption rounds. For each round, a different 128-bit sub-key is generated by the key-generator, as shown in Fig. 3.1. Within the processing module of each round, there is a combination of 4 processes: Substitution (SBox), Transposition (ShiftRow), MixColumn and XORing with the subkey (AddRoundKey), as shown in Fig. 3.2, while in the last round the MixColumn step is not performed. Outside the processing module of each round, there is a key-generator which expands the original Key for the generation of each subkey.

Encryption

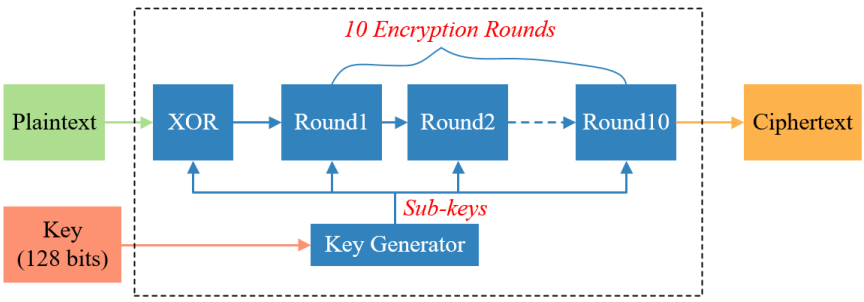


Figure 3.1: Block diagram of the Advanced Encryption Standard algorithm.

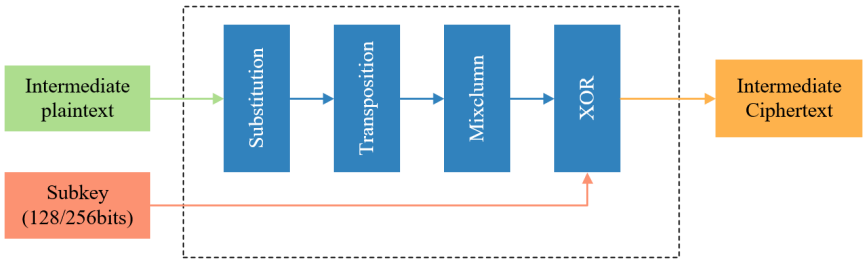


Figure 3.2: Composition of each encryption round.

- *Substitution (SBox):* Within each encryption iteration, the 128-bit intermediate state value is substituted with a corresponding piece of data to obfuscate each byte. The SBox module can be a LUT as shown in Table 3.1, for example, Hexadecimal (HEX) **9A** is replaced with HEX **B8**.

Table 3.1: 128-bit look-up table SBox.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	DB	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

- *Transposition (ShiftRow):* After the substitution process, the 128-bit intermediate state value is arranged as a 4×4 matrix, where the circular shift is applied for each row except for the first row. The first row is always kept fixed. The circular shift moves the bytes of each row to the left 1, 2 or 3 spaces over, depending on which row they are in, and then wraps them around the other side. The byte that is in the first position may end up in the last position after the shift. For example, for a 128-bit state value **C9AFD4F2FBDAC9B692AAD759F56B436A**, the ShiftRow operation for each row is shown in Fig. 3.3.

- *MixColumn:* MixColumn is performed column by column. Each column is multiplied against a constant matrix, where the multiplication results for each byte

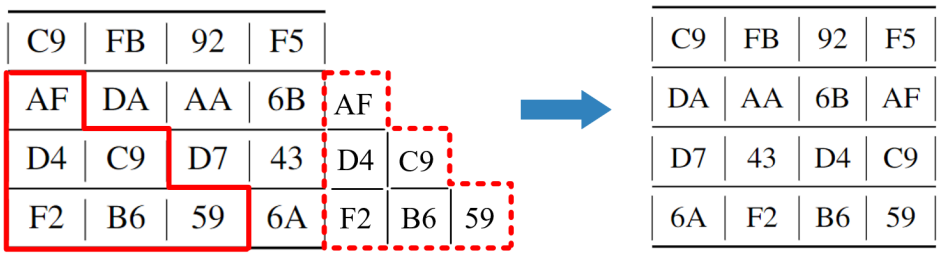


Figure 3.3: Transposition of ShiftRow process.

in the column are XORed together to form the new byte of the intermediate state value, as depicted in Fig. 3.4. For example, the first byte of the new intermediate value b_0 can be calculated by $b_0 = (a_0 \times 2) \oplus (a_1 \times 3) \oplus (a_2 \times 1) \oplus (a_3 \times 1)$. The other three new bytes (b_1, b_2, and b_3) can be obtained, respectively, by multiplying the same four bytes of the intermediate value with corresponding row values of the constant matrix, then XORing the multiplication results.

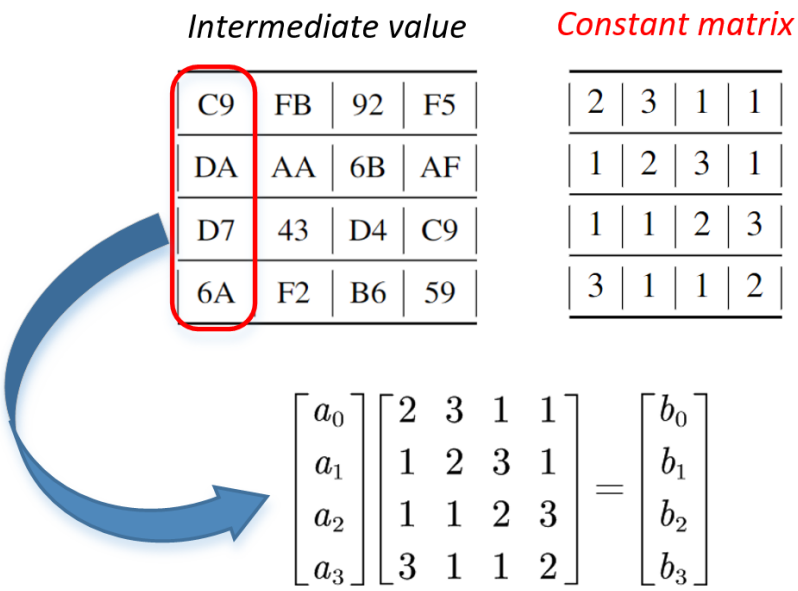


Figure 3.4: Matrix multiplication in MixColumn process.

- *XORing with the sub-key (AddRoundKey):* At the end of each round, the 128-bit intermediate state value is XORed with a current-round 128-bit sub-key generated from the key-generator.

- *Subkey generation:* The initial key (Key) undergoes a mixing technique to generate a series of subkeys, one for each round of encryption. The process of subkey generation is illustrated in Fig. 3.5.

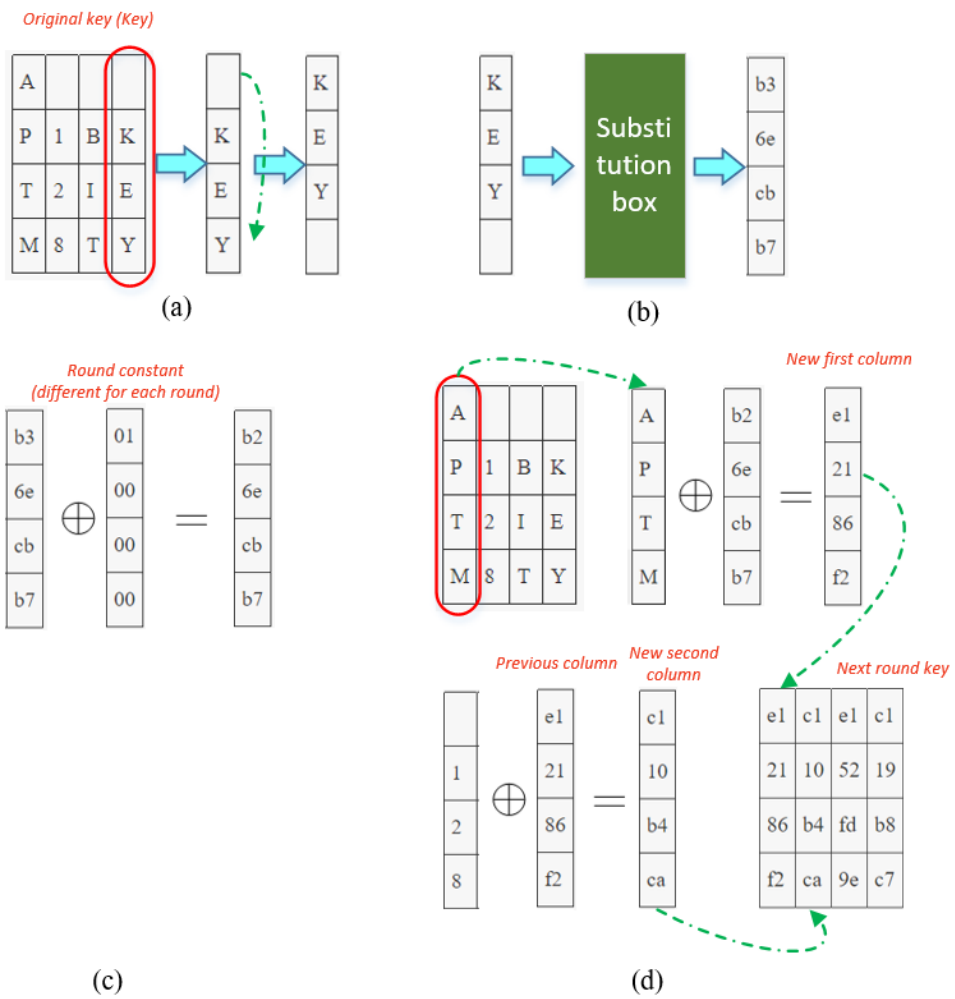


Figure 3.5: Subkey (new round key) generation by expanding the previous round key involves four steps: (a) the bottom byte of the last column of the previous key is moved to the top; (b) each byte is replaced with another byte and the whole column is mapped to another column; (c) the mapped column is then XORed with a round constant that varies for each round; and (d) the new column generated in the previous step is XORed with the first column of the previous key, which produces the new first column of the new round key. The other columns can be obtained by XORing the previous column with the corresponding column of the previous round key.

Decryption

The decryption process of AES is the inverse of the encryption process because the AES cipher is a symmetric cipher. Each decryption round also consists of four operations: Substitution, Transposition, Mixcolumn and XORing with the subkey, while in the final round the Mixcolumn step is not performed. However, the order of these operations is reversed from the encryption process, as shown in Fig. 3.6.

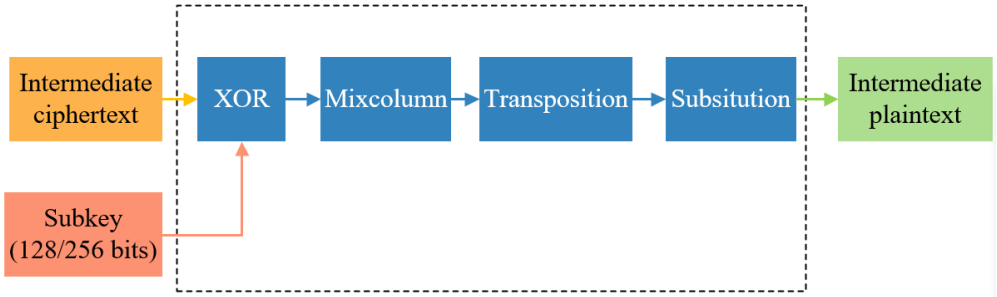


Figure 3.6: Composition of each decryption round.

3.2.2 Correlation Electromagnetic Attack Overview

A correlation based EM attack uses Pearson's correlation coefficient to recover the most probable key by collecting a sufficient number of traces [134]. By linking the measured EM traces with a leakage model, correlation coefficients are calculated to extract the key. The HD model is commonly employed as the leakage model, which assumes that the number of transitions ($0 \rightarrow 1$ or $1 \rightarrow 0$) predicts the magnitude of the EM field. The correlation based EM attack to retrieve the useful message normally has three steps:

- *Calculate the information leakage*: The assumed EM leakage is computed from the number of output transitions where the number of transitions ($0 \rightarrow 1$ or $1 \rightarrow 0$) effectively determines the magnitude of the EM field. By injecting m pre-processed n-bit plaintext vectors $I_m[n-1:0]$ into the AES circuit and sweeping all the possible subkeys, an output vector $H_m[n-1:0]$ can be computed, which is the HW of the XOR result of the plaintext and guessed subkey in the initial round of the AES encryption process [80].

- *Collect the EM traces*: When using near-field coupling techniques, there is no direct electrical connection between the probe and the circuit under test, leading to efficient contactless measurements. The known stream of input vectors $I_m[n-1:0]$ and a fixed secret key $K[n-1:0]$ produce n-bit output vectors $O_m[n-1:0]$. The EM traces generated by the output vectors are measured at the probe terminal as sampled coupled voltages $V_m[n-1:0]$.

- *Calculate the correlation coefficient and attack the key*: In this step, the coupled voltage is sampled and the information leakage is determined, simultaneously, at time point t within the clock period T. For each guessed key k, the correlation

between the assumed leakage and the measured EM traces is calculated by,

$$\rho_{k,t} = \frac{E\left[\left(V_m^t - \overline{V_m^t}\right)\left(H_m^k - \overline{H_m^k}\right)\right]}{\sqrt{Var\left(V_m^t\right) Var\left(H_m^k\right)}},\ k \in [0, 2^n - 1],\ t \in (0, T], \tag{3.1}$$

where the numerator demonstrates the co-variance between the coupled voltage and assumed leakage and Var is the variance of the dataset. When evaluating all of the correlation coefficients $\rho_{k,*}$ at any time point $(*)$ for each guessed key, a maximum $\rho_{k,\max}$ is determined. Therefore, a vector $\{\rho_{0,\max}, \rho_{1,\max}, \ldots, \rho_{2^n-1,\max}\}$ is formed for all 2^n possible keys. The key with the highest correlation coefficient corresponds to the best key candidate. A comprehensive description on correlation power/EM analysis with examples can be found in [135].

3.2.3 Correlation Electromagnetic Attack in 2.5-D Systems

The Sbox is the only non-linear part of the AES algorithm, typically implemented by using LUTs or logic operations, such as bit wise operations. However, either implementation approach entails an overhead in area or dynamic power of the circuit that supports encryption. Therefore, in order to reduce the dynamic power of the cryptographic circuit in 2.5-D integrated systems, the AES module can be implemented by combining a **memory die** (the purple block, a customised ROM-based LUT to perform SBox) with a **logic die** (the yellow block that includes the other AES components except for SBox [3]), and the encrypted application as depicted in Fig. 3.7.

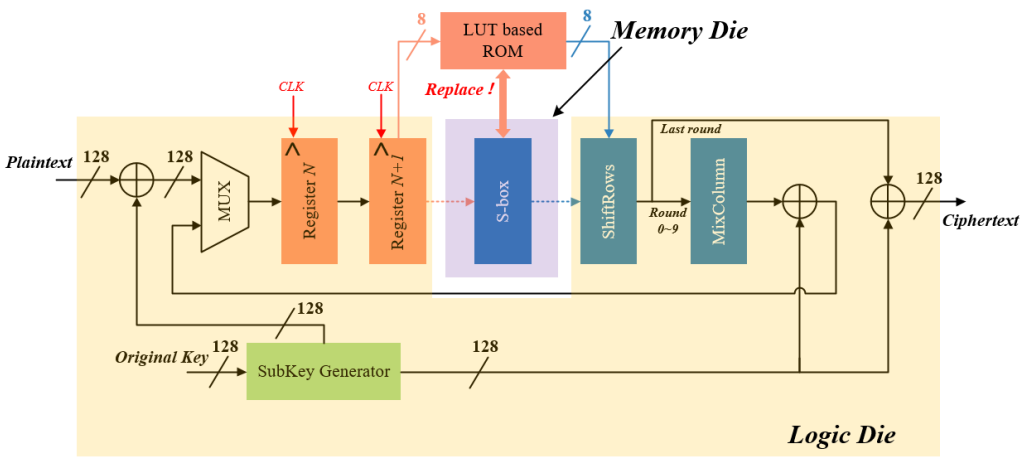


Figure 3.7: Advanced Encryption Standard with SBox (an off-chip ROM [3]).

The ROM receives the address and sends the substitution data through the memory

bus routed within the redistribution layers (RDLs) of the interposer. The value of *Register N+1* is selected as the observation target and the initial round of AES is preferred to be attacked. This choice is because the intermediate result of AES is stored in the *Register N+1* at the end of each round. Thus, the sub-key used in the initial round can be revealed by the EM leakage generated at the edge of the clock when *Register N+1* is updated. Meanwhile, the register value is the memory address being sent to read-only memory (ROM) to find the correct swapped data. The leakage generated by the memory address lines (L_a) is a function of the intermediate value of the register, which depends on the plaintext and the key, denoted as

$$L_a = \psi(PT, k_0) = \psi(HW(PT \oplus k_0)), \tag{3.2}$$

where PT is the plaintext, k_0 is the sub-key byte used in the initial round, $\oplus$ represents the XOR operation, and HW denotes the Hamming weight. In Eq. (3.2), the function ψ maintains a linear relationship with the HW value; the off-chip memory address lines can thereby be snooped to recover the sensitive key.

As depicted in Fig. 3.8, the CEMA analysis of a 128-bit AES cipher implemented in the 2.5-D integrated circuits is demonstrated.

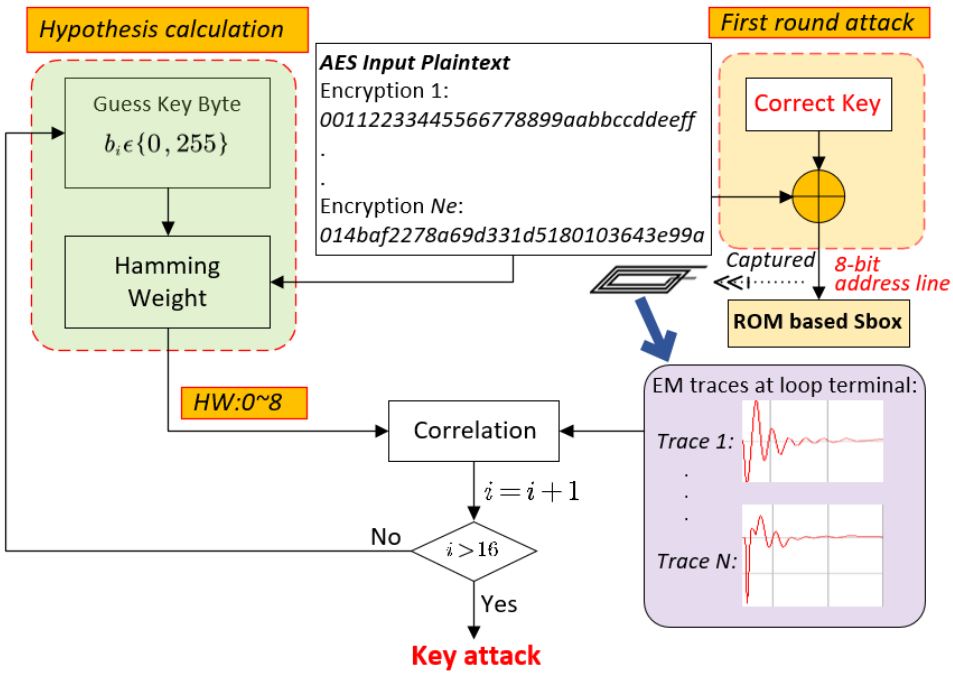


Figure 3.8: Correlation electromagnetic attack of a 128-bit Advanced Encryption Standard algorithm.

The near-field probe can be optimally placed above the bus lines to perform an EM attack using the method introduced in Section 3.4.1. When the probe is placed at the optimal position, a series of plaintexts $\{PT_1, PT_2, ...PT_m\}$ and a fixed key are fed into the system for encryption. After the 128-bit plaintext is XORed with the 128-bit sub-key in the initial round of encryption, the 128-bit intermediate result is grouped into 16 subsets and then sent continuously through the 8-bit memory bus lines for the substitution. The amplitude of the coupled voltage at the probe terminal is correlated with the number of transitions (HW of the XOR result) happening on the bus. By repeating the CEMA method described in Section 3.2.2 for 16 times, the entire 128-bit sub-key can be fully recovered byte by byte. As mentioned in Section 3.2.1, the sub-key used in each round is derived from the original key. Since the derivation process is public, the attacker who knows any sub-key can reverse-engineer the original key.

3.3 Modelling Electromagnetic Field Emanation

Before the details of the modelling are described, two directions that are relevant for this section are defined: the x, y, z-axis of the bus and the horizontal/vertical orientation of the probe. Firstly, the **y-axis** of the bus follows the current direction of the bus, the **x-axis** follows the direction across the bus width, and the **z-axis** follows the direction perpendicular to the bus surface, as shown in Fig. 3.9(a). Secondly, the probe is modelled as a single turn rectangle coil and its orientation depends on how its surface aligns with the bus surface. If the coil surface is parallel to the bus surface, the probe is placed horizontally; if the coil surface is perpendicular to the bus surface, the probe is placed vertically, as shown in Fig. 3.9(b).

As mentioned in the previous section, the targeted EM emanations originate from the top metal routings in the redistribution layers of an interposer. The modelling of an 8-bit interposer-based bus built in the three-dimensional (3-D) field solver tool *ANSYS HFSS* is shown in Fig. 3.10(a) along with related geometry parameters. The dimensions of different layers are chosen based on $UMC65$ technology. The probe is modelled as a single turn rectangular coil with 200 μm length and width equal to a quarter of the length, as shown in 3.10(b), where the coupled voltages at the probe terminal for different switching patterns exhibit the maximum NSD [136] (denoted as the standard deviation of coupled voltages divided by the mean value $\sigma/Vmean$). The bus length is set to 2 mm and the total bus width is set to 40 μm.

To minimise the effect of the initial position of the probe on the amplitude of the

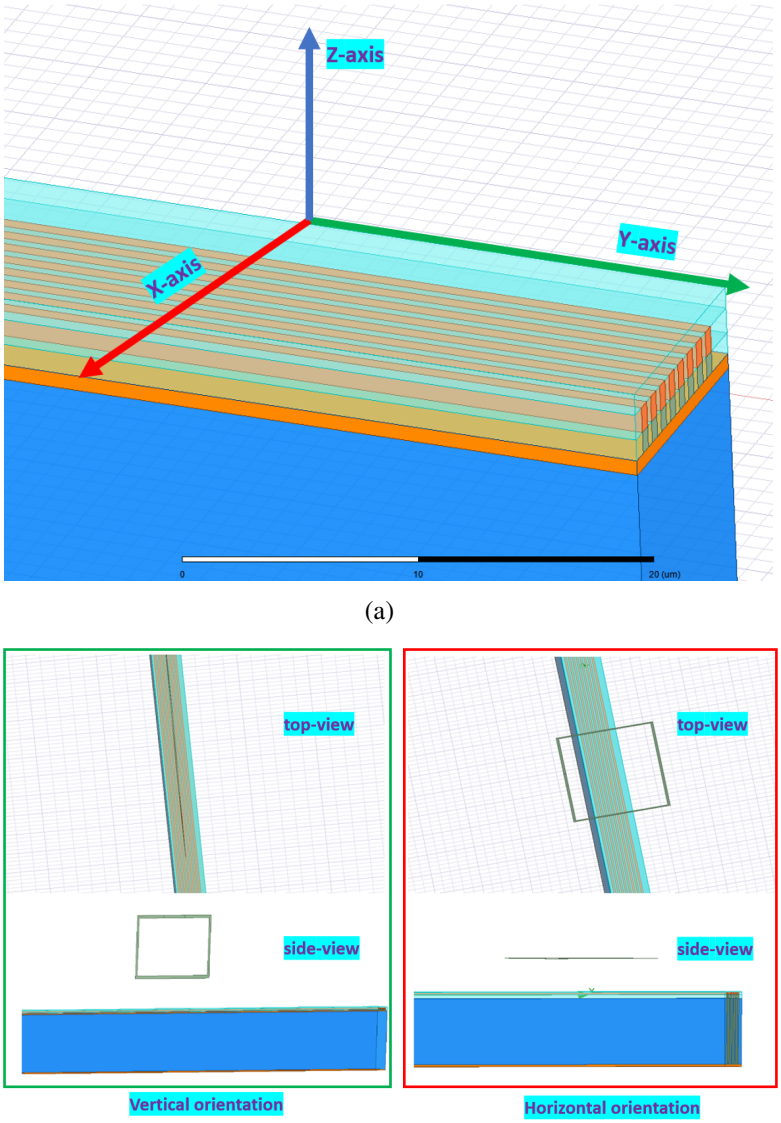
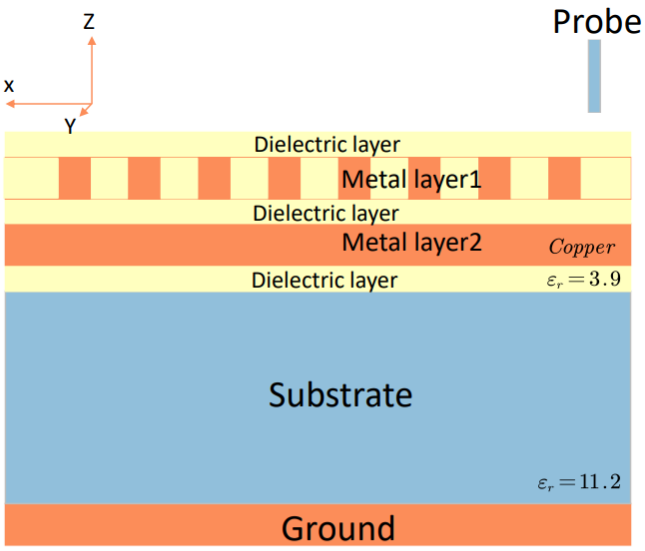


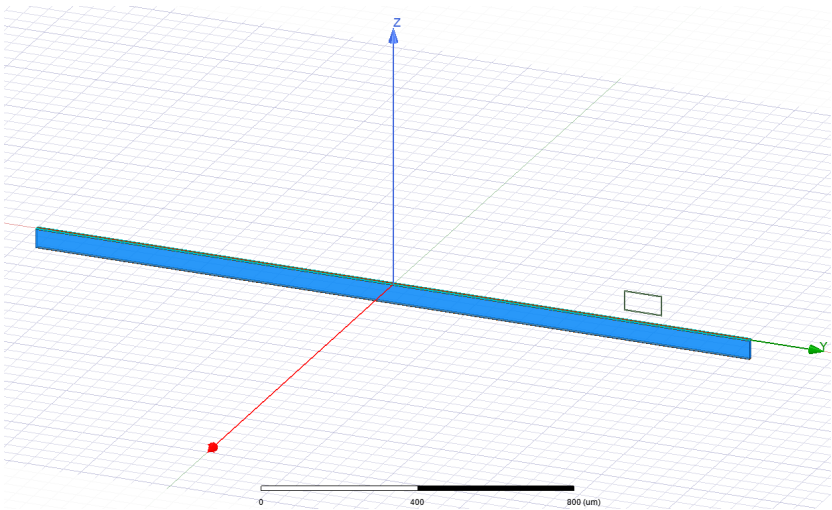
Figure 3.9: Definitions of (a) x-axis, y-axis, z-axis of the bus, and (b) orientations of the probe.

captured EM emissions, the trade-off between possible captured power deviation and relative measurement time for different step sizes along the x-axis and y-axis are shown in Tables 3.2 and 3.3, respectively. The captured power is only measured when the maximum number of transitions (8 transitions) happens on the bus. From Table 3.2, in the x-axis, a step size of 1/10 of the probe width (5 μm) is recommended as it has the lowest amplitude deviation. Due to the short bus width (40 μm) compared to the bus length (2 mm), even for a step size of 5 μm, the simulation time is acceptable. Meanwhile, according to Table 3.3, in the y-axis, a step size of the same length with



Metal layer thickness	Dielectric layer thickness	Substrate thickness	Buslines		
			Width	Space	Length
3 µm	1.5 µm	100 µm	2 µm	4 µm	2 mm

(a)



(b)

Figure 3.10: (a) Cross-section view of the stacking structure, and (b) HFSS modelling of a 2 *mm* bus with a probe (200 μm length, 50 μm width) placed vertically at a distance of 15 μm above the bus (from the probe bottom to the bus surface).

the probe length (200 μm) is recommended as in decreasing the step size, the peak power deviation is not significantly improved rather the simulation time increases.

In EM simulations, the data rate of the bus is 500 MHz and the sampling rate of the trace is set to 10 $Gbps$. A lumped port is applied as the excitation port to assign

Table 3.2: Power deviation and execution time for different step sizes in x-axis.

Step size in fraction of the probe width	Execution time normalised to the case of step size 1	Deviation of peak value for different initial positions in dB
1	1	1.27
1/4	4	1.27
1/6	6	0.77
1/8	8	0.89
1/10	10	**0.57**
1/12	12	0.96

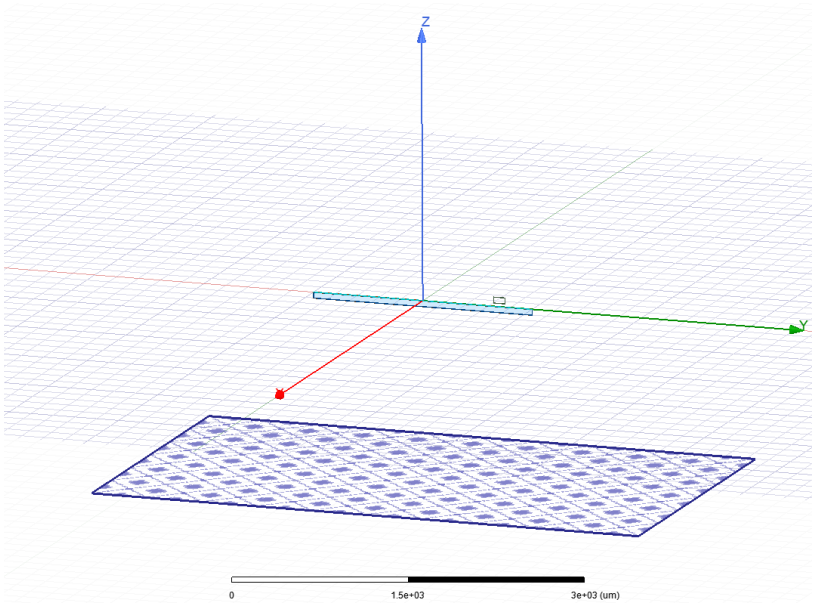
Table 3.3: Power deviation and execution time for different step sizes in y-axis.

Step size in fraction of the probe length	Execution time normalised to the case of step size 1	Deviation of peak value for different initial positions in dB
1	1	**0.41**
1/2	2	0.36
1/3	3	0.72
1/4	4	0.71
1/5	5	0.58
1/6	6	0.67

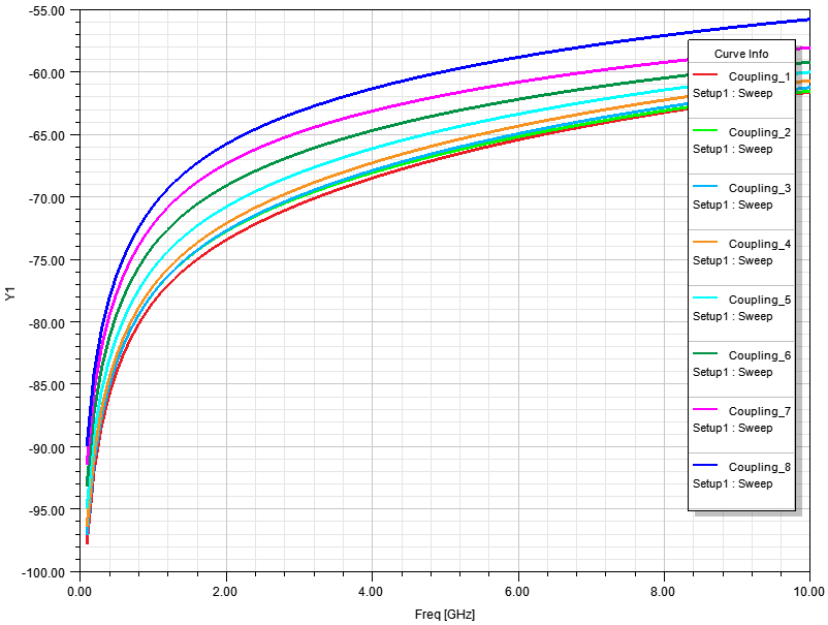
excitation for the model and a Perfect $\vec{E}$ boundary is assigned as the infinite ground plane, as shown in Fig. 3.11(a). Driven Modal is chosen as the simulation type to calculate the modal-based S-parameters in terms of power. The S-matrix solutions reflect the amplitude of power coupled at the probe terminal. The solution frequency is set to 500 MHz and the frequency is swept from 0.1 $Gbps$ to 10 $Gbps$ with a step size 0.1 $Gbps$. The simulation is repeated 256 times to examine the power of captured EM emissions at the probe terminal with different transition patterns. As shown in Fig. 3.11(b), when the number of transitions increases, the power of captured EM emissions increases as well, until it reaches the maximum when 8 lines transit simultaneously.

3.4 Electromagnetic Emission Detection

High spatial resolution can help maximise the coupling for near-field probing [112]. However, this high resolution leads to an extremely large space that must be searched to determine the best probe location for the attack. When the most effective probe location is determined, the target system is attacked fast. If the probe is positioned far from the optimal placement, MTD can require 4.3 times more traces or fail to attack [137]. Consequently, a gradient-search approach is introduced in Section 3.4.1 to facilitate the attack over brute-force search across the overall system area. SNR is used to evaluate the search efficiency of the algorithm, as shown in Section 3.4.2.



(a)



(b)

Figure 3.11: (a) Perfect $\vec{E}$ boundary used as the infinite ground plane, and (b) electromagnetic field amplitude in dB when the transitions on the bus range from 1 (coupling_1) to 8 (coupling_8).

3.4.1 Probe Placement Algorithm

When the signal current flows along the bus lines, the amplitude of the magnetic emissions in different y positions has negligible differences due to the voltage drop. However, the EM field varies significantly along the other two directions (x and z). Consequently, the search space reduces from three dimensions (x,y,z) to two dimensions

(x-z plane). In the search across the x-z plane, the NSD of the emissions is selected to evaluate information leakage.

The evaluation function of the gradient-search algorithm is denoted as $f(x,y,z) = h_{NSD}$, where h_{NSD} is the NSD of the captured power at the probe terminal at each position when the bus data is swept from 0 to 255. The minimum grid cell for the x-z space searching is set to 5 μm according to Table 3.2 and the analysis in Section 3.3.

The search starts from a random position (x_1,y_1,z_1), where NSD is measured at the closest grid cell based on this initial position. Next, NSD is measured, separately, in four adjacent grid cells to the location (x_1,y_1,z_1). The difference of measured NSD between the tested cell and the current cell is regarded as the magnitude of the vector, and the direction of the vector is pointing from the tested cell to the current cell. The addition of the four vectors is the gradient. The next location to be measured is ($x_1 - \Delta \cdot \nabla f$, y_1, $z_1 - \Delta \cdot \nabla f$), where Δ is the step size (learning rate) and ∇f is the calculated gradient. NSD can be then measured in the new grid cell, which is mapped at the new position. In the next iteration, the monitored grid cell is shifted between adjacent grid cells until the edge of the x-z plane or NSD reaches the maximum. The pseudo-code of the probe placement algorithm is listed in Algorithm 1.

Algorithm 1: Gradient-based Search Algorithm of the Optimal Probe Position

```
Initialise the grid resolution (GridResolution) of the x-z plane above the bus;
Initialise the starting position Loc=(x1, y1, z1) of the search;
Initialise the maximal value of normalised standard deviation (MaxNSD);
Initialise the step size of the search (StepSize);
MaxNSD=h_NSD(Loc);
OptimalLoc=Loc;
while true do
    Delta=DeltaCal(Adjacet4Grids());
    Loc=Loc-Delta*StepSize;
    if Loc is within the edge of x-z plane then
        MoveProbe(Loc);
        h_NSD=NSDCal();
        if h_NSD >MaxNSD then
            MaxNSD=h_NSD;
            OptimalLoc=Loc;
        else
            break;
        end if
    else
        break;
    end if
end while
```

There is one issue that needs to be considered for this gradient-based search algorithm: noise. Noise may cause the algorithm to converge to a sub-optimal value instead

of the optimal value. To address this issue, various software and hardware methods are available. Software methods include the stochastic gradient descent method, which introduces randomness to overcome noise. Hardware methods include using a high-resolution probe with a low-noise amplifier to reduce noise or disabling other logic circuits on the board during the encryption process to eliminate interference. However, this section assumes a noise-free environment for the simulation.

3.4.2 Search Area Reduction

In this section, the 8-bit bus described in Fig. 3.12, is taken as an example to verify the quality of the gradient-search algorithm. This bus width is chosen as, typically, EM attacks are launched on parts of wide buses to mitigate the exponential increase in the number of vectors that need to be analysed. Rather a byte-by-byte attack is preferred without considerably degrading the success rate of attacks [137]. A rough scan (3×3) is performed on the x-z plane to estimate the NSD distribution, as shown in Fig. 3.13. In the maps, the vertical orientation exhibits a more significant NSD than the other. Therefore, the sweeping range for x-z plane is 40 $\mu m \times 40$ μm and the probe is assumed to be placed vertically over the x-y plane, as depicted in Fig. 3.12.

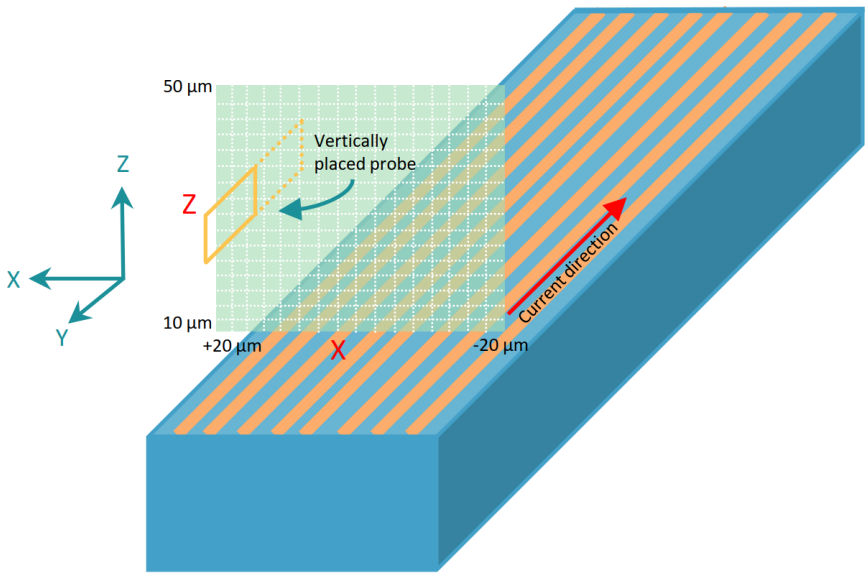
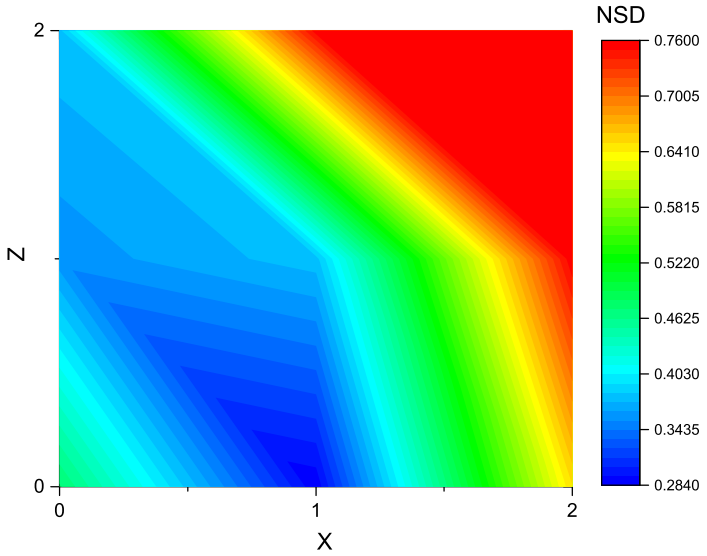
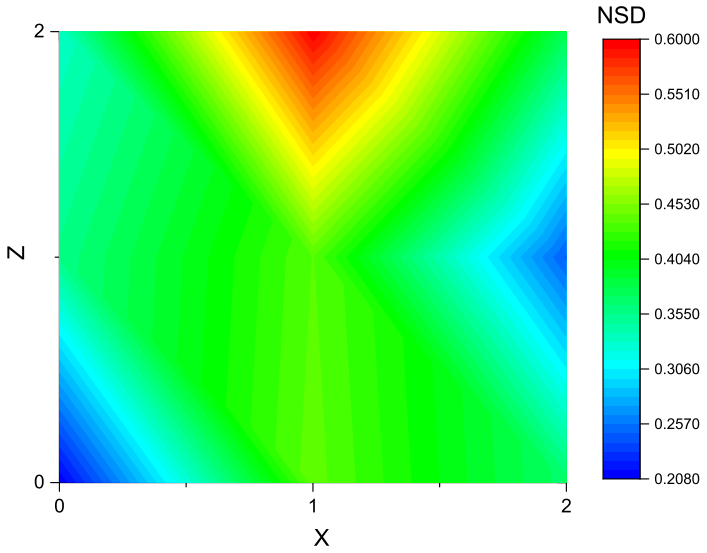


Figure 3.12: Discretised x-z plane where leakage measurements are performed.

When the x-z plane is divided into an 8×8 grid, the NSD map is shown in Fig. 3.14(a). MTD is normally inversely proportional to SNR^2, denoted as $MTD = \frac{k_1}{SNR^2} = \frac{k_2}{NSD^4}$ [138], where k_1 and k_2 are empirical parameters chosen to match the measurements. In this 8-bit bus model, when NSD equals 1.112, MTD is 30, in that case, k_2



(a)

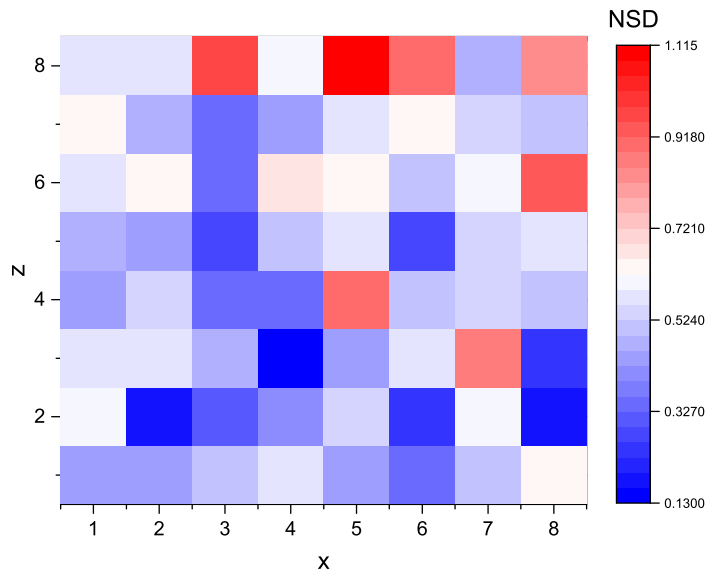


(b)

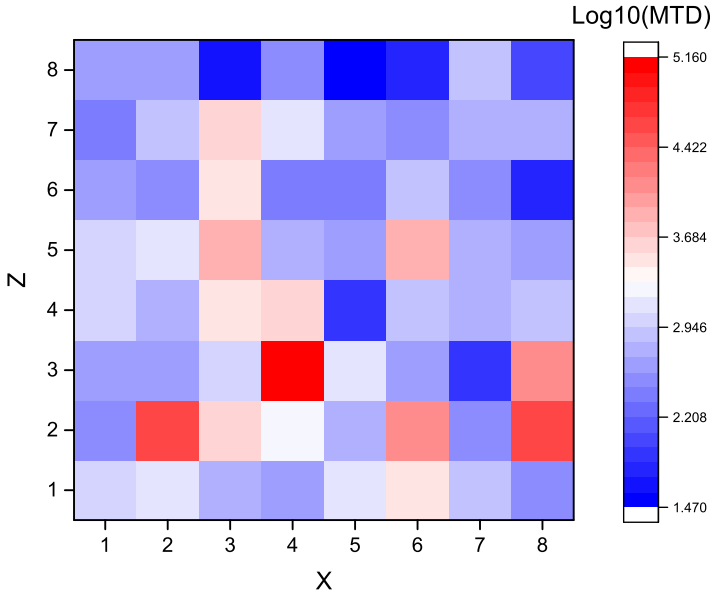
Figure 3.13: Normalised standard deviation distribution in *x*-*z* plane when the probe is placed (a) vertically and (b) horizontally. (The subfigures are not continuous functions, but discrete samples of 9 data points each. The apparent gradient is an artifact of the interpolation method used to smooth the data for visualisation and it does not convey any meaningful information. The purpose of the subfigures is to illustrate the variation of the normalised standard deviations depending on the orientation of the probe.)

is taken as 45.87. As depicted in Fig. 3.14(b), as the NSD increases, the number of traces needed for the EM attack decreases.

Different grid scales are used to verify the algorithm and the average results of



(a)

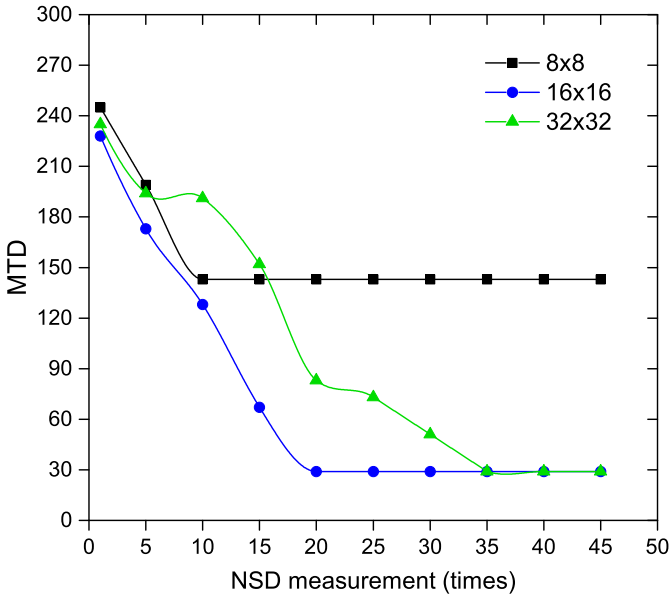


(b)

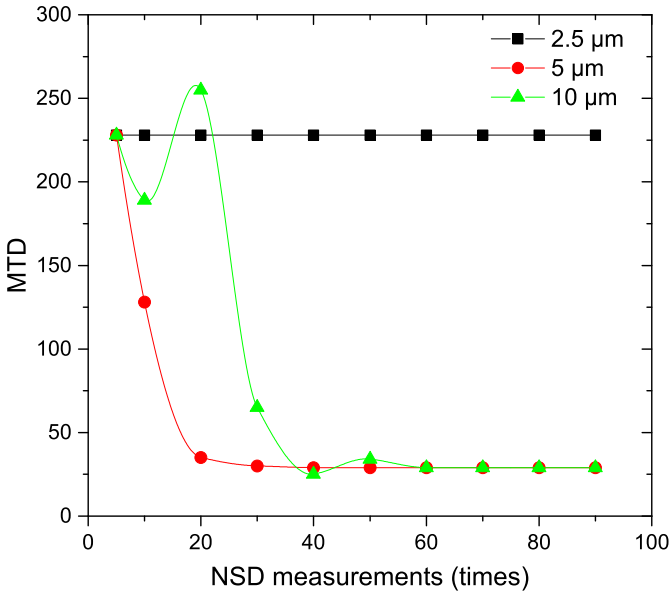
Figure 3.14: (a) Normalised standard deviation map for an 8×8 scan, and (b) heatmap of measurement-to-disclosure (a logarithmic transformation). Higher normalised standard deviation means fewer traces are needed for a successful attack.

multiple experiments are shown in Fig. 3.15. For a grid of $N \times N$, the gradient-search algorithm can reduce the NSD measurements needed to reach minimal MTD from N^2 (brute-force measurements) to approximate N, as shown in Fig. 3.15(a). Furthermore, for the 16×16 grid, the effect of step size Δ on the number of iterations is demonstrated in Fig. 3.15(b). If the step size is too tiny, for example one grid cell (2.5 μm), the search

is prevented from beginning because $Delta * StepSize$ becomes zero due to rounding (line 9 in Algorithm 1). If the step size is too large, for example 4 grid cells (10 μm), the search might miss the optimal location. Thus, given a reasonable step size, the algorithm reduces the search space over the exhaustive brute-force search.



(a)



(b)

Figure 3.15: (a) The search time complexity is reduced from $O(N^2)$ to $O(N)$ for a $N \times N$ grid, and (b) effect of step size on the convergence of the search.

3.5 Simulation Results

To further demonstrate the effectiveness of the gradient-search algorithm, the HD leakage model used in EM SCAs is firstly verified in Section 3.5.1 when the probe is placed at the optimal position, determined by the algorithm. The simulation results of EM attacks on interposer-based memory bus are demonstrated in Section 3.5.2.

3.5.1 Effectiveness of the Algorithm

Based on the gradient-search algorithm described in Section 3.4.1, the optimal EM attack location for this 8-bit bus is found at $(-2.5\mu m, 700\mu m, 45\mu m)$, where the value along the y-axis can be randomly chosen as close as possible to the near-end of the bus. When the probe is placed at this point, frequency sweeping is performed with *ANSYS HFSS* [139]. The S-parameters generated at the frequency domain are exported from *HFSS* and imported into *Spectre Cadence* for transient analysis in the time domain.

When the plaintext is swept from 0 to 255, 256 HD values are recorded in addition to the coupled voltage on the probe. When plotting all 256 peak voltage values for the 256 different plaintexts according to their HD values (0 to 8), as shown in Fig. 3.16, the captured EM trace demonstrates a very good linear correlation with the HD that depends on the plaintext and the key.

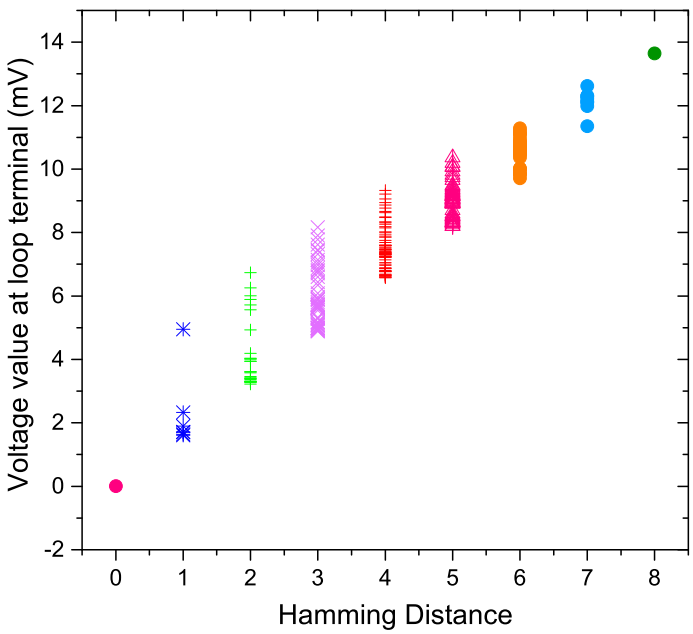
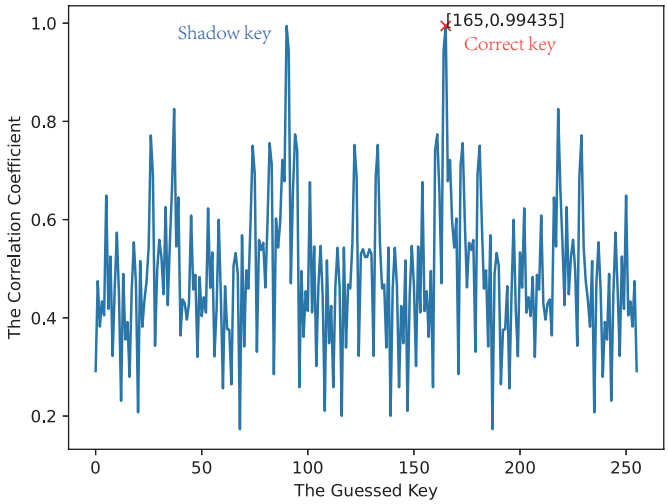


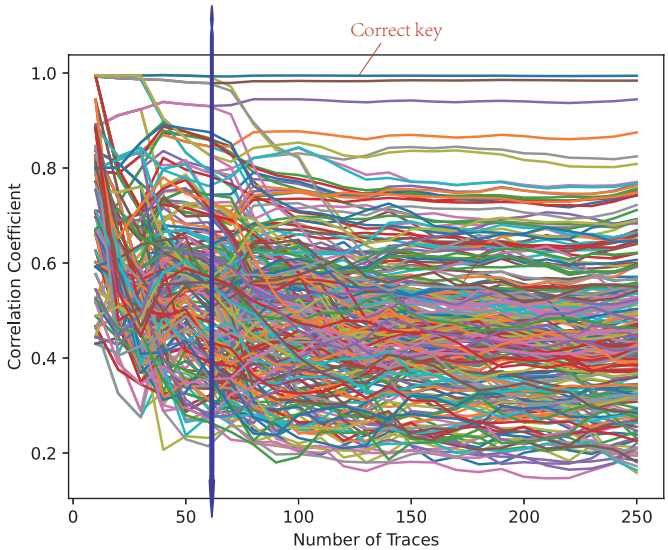
Figure 3.16: Correlation between the coupled voltage and Hamming distance at the optimal position.

3.5.2 Electromagnetic Attack Results

In order to demonstrate that the key can be extracted with this method, the AES encryption process is repeated 256 times for each 8-bit plaintext and an 8-bit fixed key. The attack results are illustrated in Fig. 3.17.



(a)



(b)

Figure 3.17: Attack results of the 8-bit bus. (a) Correlation based electromagnetic attack to discover the subkey in the initial round of Advanced Encryption Standard, and (b) the correct key is distinguished in fewer than 80 traces.

As shown in Fig. 3.17(a), among the correlation coefficients of all the guessed keys (256 keys), the position with the highest correlation corresponds, indeed, to the right key (165). According to Eq. (2.1), the SNR value is calculated as 1.032, which is above 1. There is another similar peak obtained at the position where the symmetric key (90), called the "shadow key", appears. This is because the XOR operation is symmetric. Moreover, the number of traces needed for this successful attack is illustrated in Fig. 3.17(b), where each of the 256 lines corresponds to the probability of the corresponding 8-bit value to be the correct key. The line with the highest correlation corresponds to the correct key.

After the successful attack on an 8-bit bus, the bus width is extended to 64 bits to verify the effectiveness of the algorithm on wider buses. If each byte of the 64-bit key can be attacked individually, the measurements for attacking the whole key can be highly reduced from $2^{64} = 1.84467 \times 10^{19}$ to $8 \times 2^8 = 2048$. This number can significantly increase (decrease) if the probe is sub-optimally (optimally) placed. Far from optimal probe locations completely fail to retrieve even a single sub-key. However, using the proposed algorithm, the correct 64-bit key is generated byte by byte with fewer than 256 traces per byte by placing the EM probe at the optimal position as determined for attacking each byte. The results of the correlation attack for all the sub-keys are shown in Fig. 3.18.

Furthermore, the EM attack results relating to the subkey Byte4 for two different measurement configurations are, respectively, shown in Fig. 3.19(a) and Fig. 3.19(b). When the probe is placed at the optimum location, the sub-key can be efficiently recovered with fewer than 100 traces (MTD=70) and SNR is 1.112, while at the non-optimal position, more than 10$\times$ traces are recorded and yet the correct key (that corresponds to the thick line in Fig. 3.19(b)) exhibits a low correlation coefficient and is not detected where SNR is 0.762.

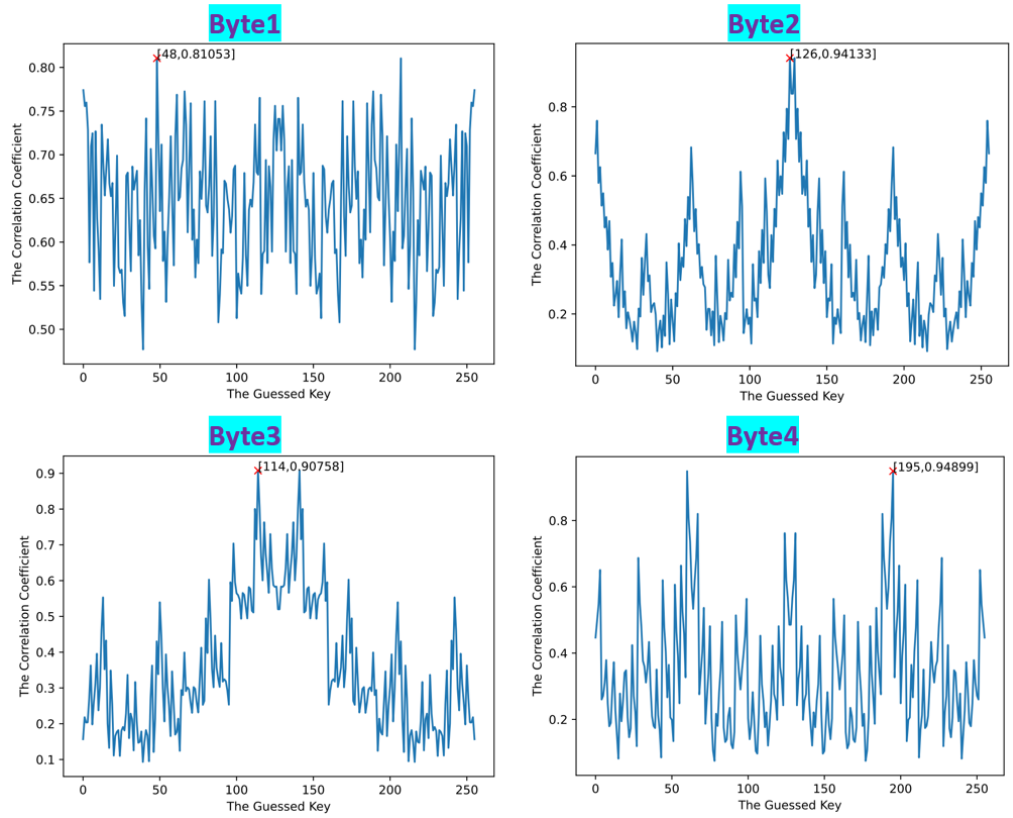
3.6 Chapter Summary

EM emissions have been explored as an effective means for non-invasive SCAs. The leaked EM field from the on-chip memory bus when the data is loaded from the memory has gained considerable attention in previous studies. However, off-chip memory buses have become a new attack target due to the relative ease of access in modern system-in-package technologies, such as 2.5-D integration where processing die and memory die are integrated on one silicon interposer.

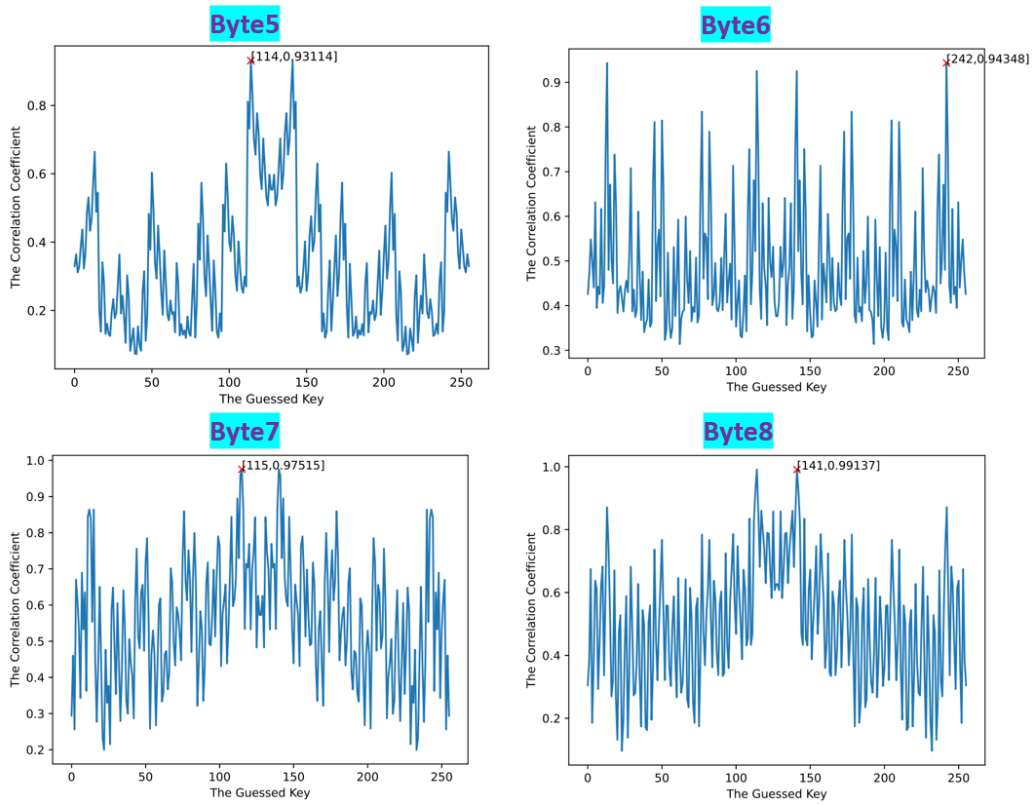
EM snooping attacks on interposer-based off-chip memory buses are investigated in this chapter. A fast and efficient side-channel attack that exploits the high spatial EM field resolution to determine the best attack location quickly is introduced. A gradient-search based algorithm is developed to provide a scanning strategy, which reduces the brute-force search of $N \times N$ space to N. To illustrate this method, an 8-bit bus is used as an example, where the performance of the gradient-based search algorithm is improved from 64 iterations to 8 iterations, which is not impressive. However, this method can be scaled to wider buses. For instance, suppose a bus width of 1024 bits (for advanced high-performance bus (AHB)) and the step size is 5 μm. The grid of the x-z plane above the bus surface will be 1204×1024. If each iteration (measuring the NSD value in each grid cell) takes 3 minutes, then the brute-force search ($O(N^2)$) will take $1024 \times 1024 \times 3$ mins = 3145,728 mins = 2184.5 days to complete, while the gradient-based search ($O(N)$) would take 1024×3 mins = 2.13 days to complete. This is a difference of more than 1025 times, which shows that the gradient-based search algorithm scales much better than the brute-force search.

NSD is preferred as the metric for leakage in the gradient-based search algorithm, which measures the variation of the EM emissions for different plaintexts and identify the optimal attack location with a low MTD value. For a cryptographic system with a fixed key, a high NSD value implies that the EM emissions vary significantly with the plaintexts, which reveals more information and facilitates the attack. Conversely, a low NSD value implies that the EM emissions are relatively constant with the plaintexts, which conceals more information and impedes the attack. The NSD value is influenced by the size of the plaintexts in two ways. First, if the size of plaintexts is too small, the EM emissions will have less variation, leading to a low NSD value and a difficult attack. Second, if the size of plaintexts is too large, the EM emissions will have more noise, resulting in a low NSD value and a difficult attack. Thus, there is an optimal size of plaintexts that maximises the NSD value and enables the most efficient EM attacks. In this section, since the simulation of the EM attacks is conducted in a noise-free environment, the size of plaintexts is chosen to be as large as possible. For the off-chip 64-bit bus scenario, placing the probe at the optimum location reduces the number of traces required for sub-key attacks by $10\times$ compared to other locations.

The implementation details of mitigation techniques against EM attacks on interposer-based memory buses described in this chapter are introduced in Chapter 4. Moreover, the effectiveness of the EM attack method and protection methods on FPGA-based encryption implementations are verified in Chapter 5.

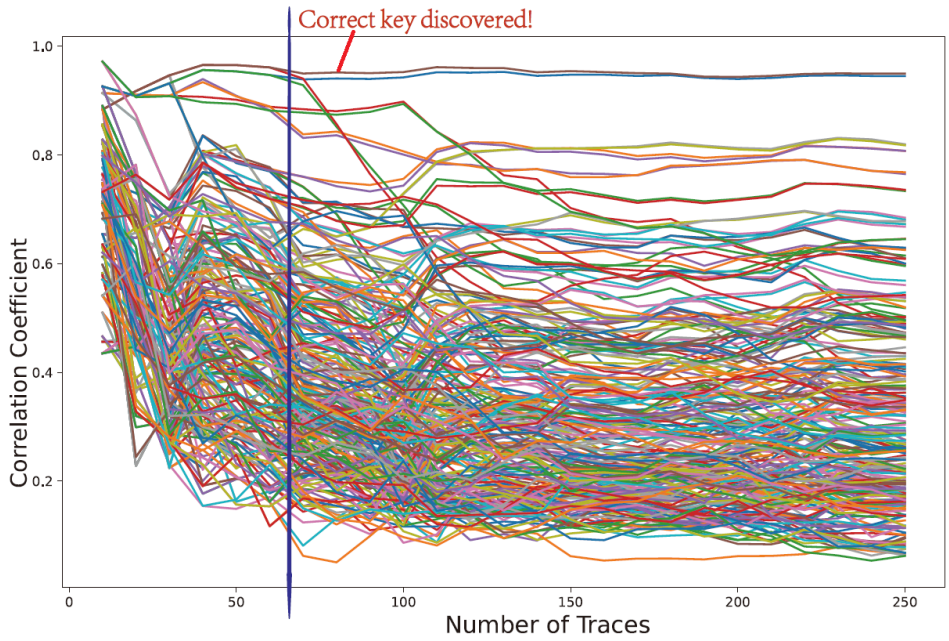


(a)

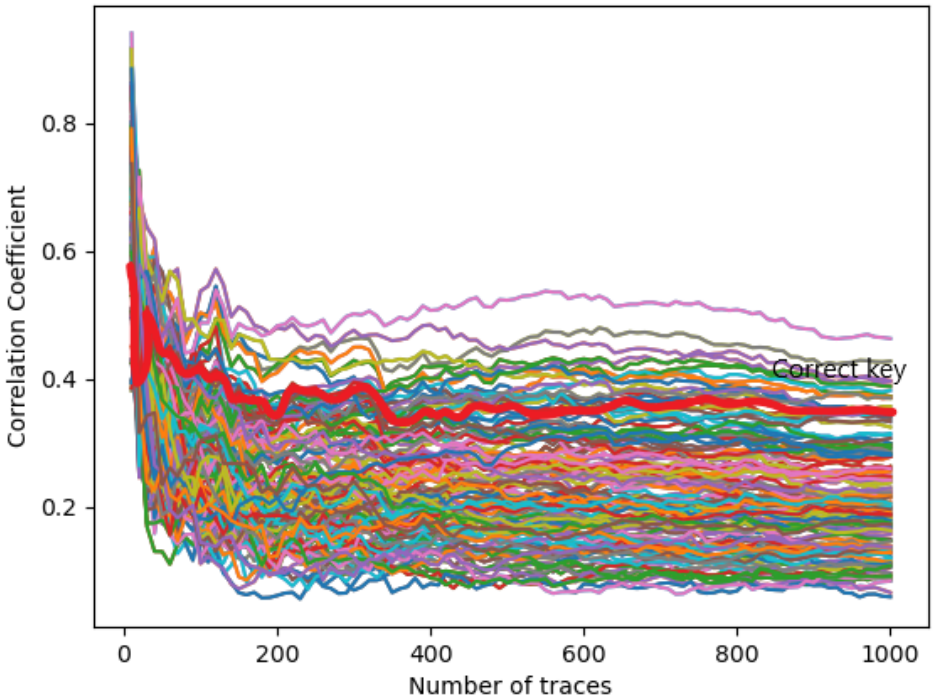


(b)

Figure 3.18: Attack results of the 64-bit bus. (a) Byte1 to byte4, and (b) byte5 to byte8.



(a)



(b)

Figure 3.19: Electromagnetic attack results for Byte4 where the probe is (a) optimally placed as determined by the gradient-search algorithm, and (b) placed 10 μm away from the optimal position.

Chapter 4

Performance-Aware Delay Insertion Methods

4.1 Chapter Overview

In this chapter, novel performance-aware techniques applied to mitigate EM attacks on interconnect buses are introduced. The chapter includes the concepts, mechanism, circuit implementations, simulation experiments and results.

As mentioned in the previous chapter, CPAs can exploit the measured power consumption and recover the key at low cost [140]. Countermeasures against CPAs at circuit level can be **flattening** in terms of the power consumed within critical clock cycles or **randomising** in terms of the processing power to reduce the correlation between the processed data and the consumed power. Random delay insertion (RDI) is an effective technique against CPAs, which belongs to the latter case [141]. RDI can effectively reduce the correlation between the assumed power model and measured power consumption, thereby preventing potential CPAs. Inspired by the idea to randomise power consumption through RDI in CPAs, an effective delay insertion scheme that hinders EM attacks by inserting delay into *boundary lines* of the bus while not affecting the circuit performance is proposed in Section 4.3. Before describing this technique, the core concept used in the delay insertion schemes discussed in this chapter, crosstalk effect, is introduced in Section 4.2.

Furthermore, in order to resolve the application limitations due to the static nature of the method described in Section 4.3, a dynamic delay insertion method, combined with DBI, is proposed in Section 4.4. Finally, a chapter summary is drawn in Section 4.5. The validations for efficient EM attacks on wide buses illustrated in Chapter 3 and

EM-attack mitigation methods will be discussed in Chapter 5.

4.2 Crosstalk Effect on Bus Latency

Crosstalk is associated with the coupling capacitance and inductance between adjacent lines. With decreasing feature sizes, the coupling capacitance between adjacent lines dominates the ground capacitance and the transmission latency of the interconnects becomes largely data-dependant due to the dominant coupling capacitance.

As shown in Fig. 4.1 where the bit lines are treated as *RC* interconnects, the transmission latency of a bus is proportional to the wire resistance and capacitance. Using the typical 50% delay metric in static logic circuits, the transmission latency of an interconnect is estimated by [142],

$$\begin{aligned} T &= 0.4R_tC_t + 0.7\left(R_{buffer}C_t + R_{buffer}C_L + R_tC_L\right) \\ &\approx \left(0.7R_{buffer} + 0.4R_t\right)C_t, \;\; if\; C_L \ll C_t, \end{aligned} \tag{4.1}$$

where R_t, C_t are the resistance and capacitance of the wire, respectively. R_{buffer} is the on-resistance of the driver and C_L is the load capacitance.

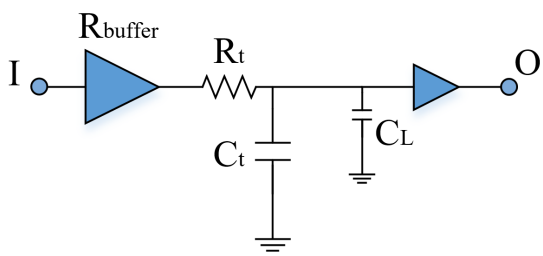


Figure 4.1: Interconnect latency. The latency is determined by the wire resistance R_t and capacitance C_t, on-resistance of the driver R_{buffer}, and load capacitance C_L.

In this model, the bus latency is effectively determined by the capacitance driven by each line C_t, which depends on the switching conditions of the adjacent lines, as its resistance depends on the geometric characteristics and, nominally, is the same for all lines. The capacitance for each line is, respectively, composed of the ground capacitance C_g and coupling capacitance C_m, as shown in Fig. 4.2(a). The interconnection capacitance of the *middle line* I_2 can be calculated by [143],

$$C_{middle} = C_g + C_m\left|\frac{\Delta V_{12}}{V_{dd}}\right| + C_m\left|\frac{\Delta V_{23}}{V_{dd}}\right|, \tag{4.2}$$

where V_{dd} is the voltage of the power supply, and ΔV_{12}, ΔV_{23} is the voltage difference between the *middle line* I_2 with its two neighbours I_1, I_3, respectively.

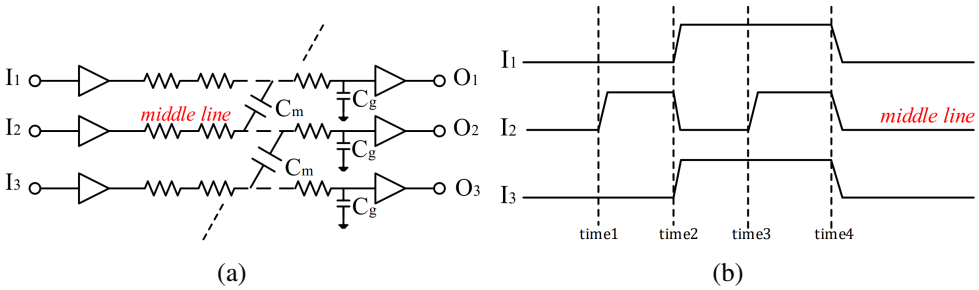


Figure 4.2: (a) Interconnect model of three bit lines of a k-bit bus, and (b) three data transition scenarios: same direction transition, single transition and opposite direction transition.

For an example pattern shown in Fig. 4.2(b), where the three lines switch in the same direction (*time4*), in opposite direction (*time2*), and only a single line transitions (*time1, time3*), the capacitance of the *middle line* corresponds to C_g, $C_g + 4C_m$, and $C_g + 2C_m$, respectively. For interconnect buses due to the Miller effect, the highest capacitance is driven when a line switches to the opposite direction of its neighbours (*time2* in Fig. 4.2(b)). Thus, the worst bus latency among all of the bits is proportional to $C_g + 4C_m$.

4.3 Static Delay Insertion

The methodology comprises an efficient delay insertion scheme that hinders EM attacks, where the delay is inserted into the *boundary lines* of the bus.

As the worst-case bus latency is determined by the lines that drive the maximum cross-coupling capacitance, inserting delay at the *boundary lines* does not affect the circuit performance as these lines always drive a lower capacitance. The inserted delay improves the security strength of the bus against EM attacks due to the reduction of the correlation between EM emissions and transmitted data, making the methodology effective and directly applicable with negligible overhead. The technique is applied to interposer-based off-chip memory buses due to the increasing adoption of 2.5-D integrated systems (although the method is effectively applicable to any interconnect bus). Therefore, the rationale of this static delay insertion scheme, statistical analysis of the added delay on bus performance and security, testbench and simulation results are, respectively, introduced in the following subsections.

4.3.1 Rationale of Static Delay Insertion Scheme

As illustrated in Fig. 4.3, the delay insertion strategy aims to ***skew the transition time of the two boundary wires.*** This intentional skew does not increase the latency of the bus, as the latency of these lines is always smaller than the worst-case latency of the bus.

As the edge lines have only one adjacent line, the capacitance that these lines can drive ranges from C_g to $C_g + 2C_m$ when switching in the same and opposite direction, respectively, to the neighbouring line. Inserting some delay to these lines decreases (increases) the coupling capacitance for switching in the opposite (same) direction. For the case of *time2* in Fig. 4.3, when the delay Δt inserted into the *boundary lines* (I_1, I_3) is greater or equal to their transition time (t_{tran}), the coupling capacitance of the *boundary lines* is reduced. If the skew is selected appropriately, the bus latency of the *boundary lines* (denoted as D_1, D_3) can still be smaller than the worst-case latency (denoted as D_2), as shown in Fig. 4.3. The precise delay to be added is determined by analysing these cases during design time and can be inserted with negligible overhead. Furthermore, the inserted delay decreases the correlation between the processed data and the coupled voltage induced at the probe terminal, which helps improve the resilience against EM SCAs. Therefore, the performance-aware delay insertion can serve as a security countermeasure without degrading the circuit performance.

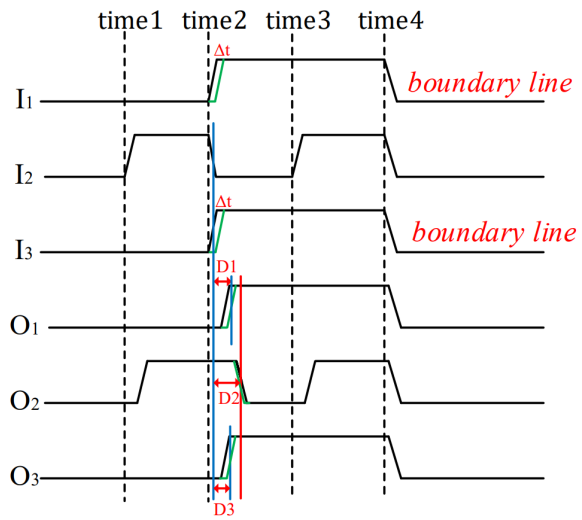


Figure 4.3: Performance-aware delay insertion into *boundary lines* to improve the resilience against electromagnetic side-channel attacks.

In the 8-bit bus case, the worst-case switching pattern is where all adjacent bit lines switch either from 1 to 0 or from 0 to 1, alternatively, between two successive

pieces of data. In this pattern, all but the bit lines at the edges of the bus drive the maximum capacitance $C_g + 4C_m$, while the lines at both edges drive a capacitance $C_g + 2C_m$. Consequently, by inserting a delay that is less than the delay incurred by driving a capacitance of $2C_m$ into two *boundary lines*, which drive a lower capacitance, the latency of the bus does not decrease but security resiliency can be offered by decreasing the correlation coefficient defined in Section 3.2.2. The available range of delays and the effect on the security resilience of the interconnect are discussed for a specific bus on a silicon interposer in the following sections.

4.3.2 Quantifying Effect of Added Delay on Bus Latency

In this subsection, in order to quantify the effect of the delay insertion on the total bus latency, only the worst case is considered. That is, the bus latency is determined by the switching conditions that lead to the maximum capacitance for a bit line. This condition happens where all adjacent bit lines switch in the opposite directions. According to Eq. (4.1) and Eq. (4.2), when delay Δt is inserted into the *boundary lines*, the capacitance of a *boundary line* is

$$\begin{aligned} C_{boundary} &= C_g + C_m \left| \frac{\Delta V_2(0) - \Delta V_1(\Delta t)}{V_{dd}} \right| + C_m \left| \frac{\Delta V_2(0) - \Delta V_3(\Delta t)}{V_{dd}} \right| \\ &= C_g + nC_m,\ 1 \leqslant n \leqslant 2, \end{aligned} \tag{4.3}$$

where $n = 1$ for $\Delta t \geqslant t_{tran}$ and $n = 2$ for $\Delta t = 0$.

By substituting Eq. (4.3) into Eq. (4.1), the total bus latency of the *boundary line* (T_b) can be estimated by,

$$T_b = \Delta t + \left(0.7R_{buffer} + 0.4R_t\right)(C_g + nC_m),\ 1 \leqslant n \leqslant 2. \tag{4.4}$$

When delay Δt is inserted into the *boundary lines*, the coupling capacitance of the line next to the *boundary line* is also reduced, whose latency (T_{nextb}) can be approximated as

$$T_{nextb} = \left(0.7R_{buffer} + 0.4R_t\right)(C_g + nC_m),\ 3 \leqslant n \leqslant 4, \tag{4.5}$$

where $n = 3$ for $\Delta t \geqslant t_{tran}$ and $n = 4$ for $\Delta t = 0$. The bus latency of the remaining middle lines (T_{middle}) can be estimated by,

$$T_{middle} = \left(0.7R_{buffer} + 0.4R_t\right)(C_g + nC_m),\ n = 4. \tag{4.6}$$

When no delay is added, the transmission latency of the middle lines, described by Eq. (4.6) is the greatest across the entire bus since the middle lines drive the highest capacitance ($C_g + 4C_m$). From Eqs. (4.4)–(4.6), if delay Δt, inserted into the *boundary lines*, is properly selected, the latency of the *boundary line* (T_b) does not surpass the bus latency of the middle lines (T_{middle}). Consequently, the overall speed of the bus is not degraded.

4.3.3 Statistical Analysis of the Attacks

To establish the link between the added delay and the security figures of merit, such as the correlation coefficient and *SNR*, a systematic and theoretical analysis is offered in this subsection.

When Δt is inserted into the bus to temporally shift the transition of the lines, the total captured leakage by the probe V_{total}, is denoted as $V_{total} = V_{t-\Delta t} + V_{noise}$, where V_{noise} is the uncorrelated noise generated from neighbouring wires. As shown in Fig. 4.4, $V_{t-\Delta t}$ can be replaced by $V_{max} + V_{\Delta}$, where V_{max} is the maximum coupled voltage at time t, correlated with the leakage model, and V_{Δ} is the voltage difference due to the shifting.

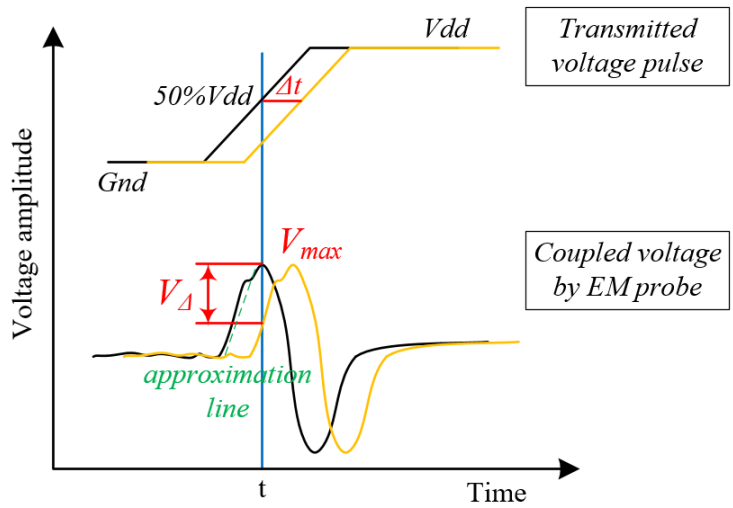


Figure 4.4: The peak value of coupled voltage at the probe terminal has a V_{Δ} difference when delay Δt is inserted to an interconnection (not to scale).

If $Var(V_{\Delta}) \gg Var(V_{\max})$ is assumed, the correlation between the assumed leakage

H and V_{total} can be calculated by [138],

$$\begin{aligned}\rho\left(H,V_{total}\right) &= \frac{E\left(H,V_{total}\right)-E\left(H\right)E\left(V_{total}\right)}{\sqrt{Var\left(H\right)Var\left(V_{total}\right)}} = \frac{\rho\left(H,V_{\max}\right)}{\sqrt{1+\frac{1}{SNR}}}\frac{1}{\sqrt{1+\frac{Var(V_{\Delta})}{Var(V_{\max})}}} \\ &\approx \frac{\rho\left(H,V_{\max}\right)}{\sqrt{1+\frac{1}{SNR}}}\frac{1}{\sqrt{\frac{Var(V_{\Delta})}{Var(V_{\max})}}}.\end{aligned} \tag{4.7}$$

From Eq. (4.7), $\rho\left(H,V_{total}\right)$ is inversely proportional to $\sqrt{Var\left(V_{\Delta}\right)}$. According to [144], the pulse voltage coupled at the probe terminal is denoted as $-MI_p\frac{8t}{\tau^2}\exp\left(-\frac{4t^2}{\tau^2}\right)$, where M, I_p, and τ is the mutual inductance between the probe and the interconnects, peak value of current transmitted on the bus, and pulse width, respectively.

It is assumed that data on all bit lines are launched from a register followed by an I/O buffer since an off-chip bus is investigated. In other words, the register to output path (R2O) is reasonably assumed to exclude combinational paths that may lead to glitches. Furthermore, the inclusion of an I/O buffer can suppress such glitches.

Additionally, the modelling process is used to analytically link the correlation coefficient with the inserted delay. The slew and delay of signal is based on the 10%-90% range. Within this range, V_{Δ} is assumed to change linearly with Δt, and the voltage waveform is approximated by the dashed green line in Fig. 4.4.

The inserted delay Δt used in this section is generated from delay lines [145] and uniformly distributed in group $[0,\Delta_{\min},2\Delta_{\min},...\Delta_{\max}]$, where $\Delta_{\min}$ is the minimum delay and $\Delta_{\max}=k\Delta_{\min}$ (the delay of a line for driving a capacitance of $2C_m$). More details about the circuits that can produce these delays with low overhead are described in Section 4.3.4. Therefore, the variance of V_{Δ} can be estimated by,

$$\begin{aligned}Var\left(V_{\Delta}\right) \approx Var\left(\Delta_t\right) = E\left(\Delta_t^2\right)-\left[E\left(\Delta_t\right)\right]^2 &= \frac{k\left(2k+1\right)\Delta_{\min}^2}{6}-\frac{k\Delta_{\min}^2}{2} \\ &= \frac{k\left(k+2\right)}{12}\Delta_{\min}^2.\end{aligned} \tag{4.8}$$

According to Eq. (4.8), Eq. (4.7) can be further approximated as,

$$\begin{aligned}\rho\left(H,V_{total}\right) \propto \frac{1}{\sqrt{Var\left(V_{\Delta}\right)}} \approx \sqrt{\frac{12}{k\left(k+2\right)\Delta_{\min}^2}} &\propto \frac{1}{\sqrt{k^2\Delta_{\min}^2+2k\Delta_{\min}^2}} \\ &= \frac{1}{\sqrt{\Delta_{\max}^2+2k\Delta_{\min}^2}}.\end{aligned} \tag{4.9}$$

The parameters $\Delta_{\max}$, k, and $\Delta_{\min}$ in Eq. (4.9) play a crucial rule in the effectiveness of the delay insertion technique. By increasing any of these parameters, the correlation between the estimated leakage and measured EM emissions can be reduced. However, there are trade-offs between the security and performance. If $\Delta_{\min} = 0$, no delay is inserted and the correlation coefficient remains unchanged, offering no security benefit; If $\Delta_{\max} > t_{tran}$, the delay inserted to the *boundary line* exceeds the transition time of the middle lines (the worst case) and the bus performance is degraded. Therefore, $\Delta_{\min}$ and $\Delta_{\max}$ in Eq. (4.9) have to be carefully selected to optimise the trade-off between the security and performance. The lower bound $\Delta_{\min}$ (>0) and upper bound $\Delta_{\max}$ ($<t_{tran}$) depend on the specific scenario, as $\sqrt{Var(V_{\Delta})}$ (for a fixed delay Δt) varies across different applications, and so does $\rho(H, V_{total})$ for different scenarios (Eq. (4.7). Moreover, according to Eq. 4.9, $\Delta_{\min}$ has to exceed a specific threshold to ensure that $\rho(H, V_{total})$ is sufficiently small to prevent the attacks, where the threshold is scenario-specific.

As there are diverse side-channel mitigation techniques that have been proposed, for this new *boundary-line* delay insertion technique (even if deterministic), it will be practically infeasible or prohibitively time-consuming for an attacker to guess what delay-insertion pattern has been adopted in the circuit. Therefore, although deterministic, the delay insertion strategy is unknown to the attacker, and can provide the off-chip memory bus resilience against EM SCAs.

4.3.4 Hardware Architecture of Delay Insertion Scheme

The architecture of the proposed static delay insertion scheme is described in Fig. 4.5. Delay is added to two *boundary lines* where the added delay is generated by a delay line and is applied to these interconnect lines. The added delay Δt should be chosen such that the resulting bit line latency does not exceed the worst-case latency where $n = 4$ (see Eqs. (4.4) and (4.6)).

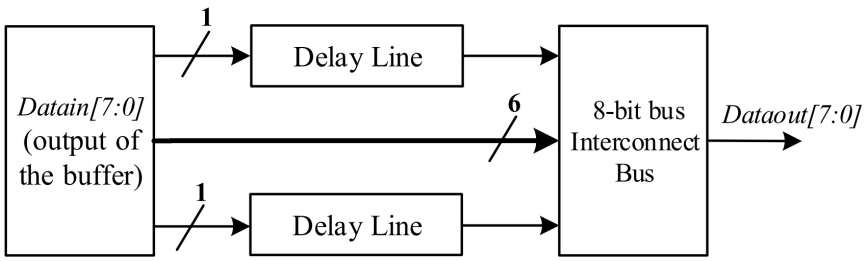


Figure 4.5: The two *boundary lines* are selected as the target lines to be delayed with the inserted delay Δt generated by the delay line.

There has been some research relating to the implementation of the delay logic. A tunable delay line rather than random D flip-flop (RDFF), random write D flip-flop (RWDFF) or random number generator (RNG), is assumed here to produce the desired delay as the implementation of these RDI circuits is complex and can easily induce significant overhead in power and/or area.

With the delay line illustrated in Fig. 4.6, the propagated data pulse can be delayed at a low power cost [145]. When the data pulse *DataPulse_in* propagates from left to right, if signal S_{i-1} is low, INV1 is grounded through two parallel n-MOSFETs, whereas INV1 is grounded through single n-MOSFET when S_{i-1} is high. The parallel MOSFETs in the former case has a lower resistance than the latter case. Therefore, when S_{i-1} equals '0', INV1 has a shorter delay and a longer delay, when S_{i-1} equals '1'. After INV1, the data pulse changes its polarity and the field-effect transistors (FETs) for the pulling-up changes from n-MOSFET to p-MOSFET. If signal S_i is low, INV2 is pulled up to the supply voltage through two parallel p-MOSFETs, which has a lower resistance than being pulled up through one p-MOSFET when S_i is high. Therefore, INV2 has a shorter delay when S_i equals '0' and a longer delay when S_i equals '1'.

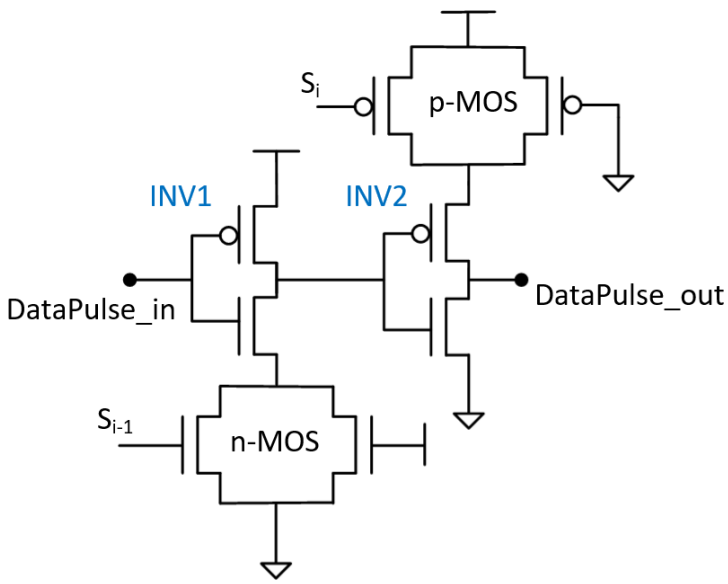


Figure 4.6: Implementation of low-cost delay line.

The inserted delay is determined during the design process with the steps described in Section 4.3.3 (specifically using Eqs. (4.4)–(4.6) and (4.9)). Once the desired delay Δt has been determined, the transistors of the delay line can be suitably sized to produce this delay for the worst transition case.

4.3.5 Simulation Experiments

In this subsection, the simulation environmental setting is first provided. Then the EM attack and circuit performance results are demonstrated where delay is inserted into the *boundary lines*. Furthermore, the robustness of the new delay insertion technique is verified.

Simulation environment settings

The same bus model described in Section 3.3 is used in this section. When the probe is placed vertically over the bus, the S-parameters are generated with a frequency sweep, exported from *HFSS*, and imported into *Spectre* for transient analysis of the interconnect in the time domain. The overall design is simulated using 65 *nm* technology parameters and the nominal voltage V_{dd} is 1.8 V (typical I/O voltage for 65 *nm* technology). To help with the demonstration of the simulation results in the following subsection, the 8-bit bus is depicted in Fig. 4.7 with annotated bit lines.

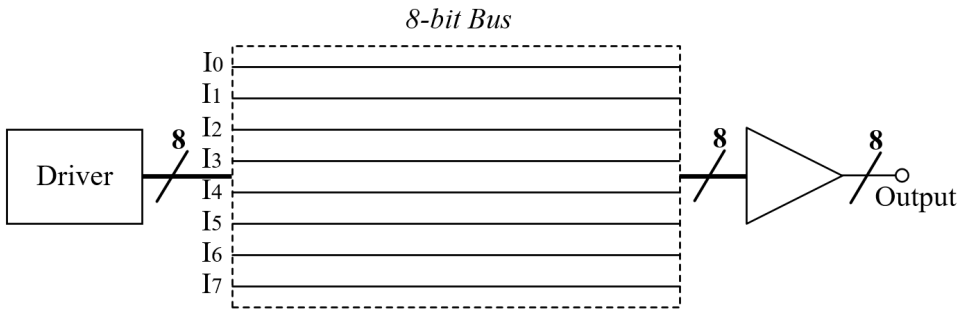


Figure 4.7: Structure of the interposer-based bus with annotated bit lines.

As discussed earlier, the physical device can be deemed resistant to side-channel attacks in the noisy environment if the *SNR* value in a noise-free simulation environment falls below 1. Hence, the impact of critical delay parameters on the circuit security and performance is the main focus of investigation. *SNR*, as defined in Section 2.6, is used as the metric to assess the security and the security results in this dissertation are based on sweeping the 8-bit input.

Signal-noise ratio and performance simulation results

The effect of inserted delay Δt on *SNR* and total bus latency is explored. Meanwhile, the lower bound Δ_{min} and upper bound Δ_{max} can be determined according to the simulation results.

In the WC scenario (all adjacent bit lines switch simultaneously in the opposite direction), Δt is inserted into both *boundary wires* (I_0, I_7 in Fig. 4.7) to shift in time the transmitted data. EM attacks are performed to extract the secret key. The AES algorithm is repeated 256 times for all possible 8-bit plaintexts and a fixed 8-bit key. The correlation coefficient (for both correct key and incorrect key) and *SNR* are, subsequently, plotted as a function of Δt, as depicted in Fig. 4.8.

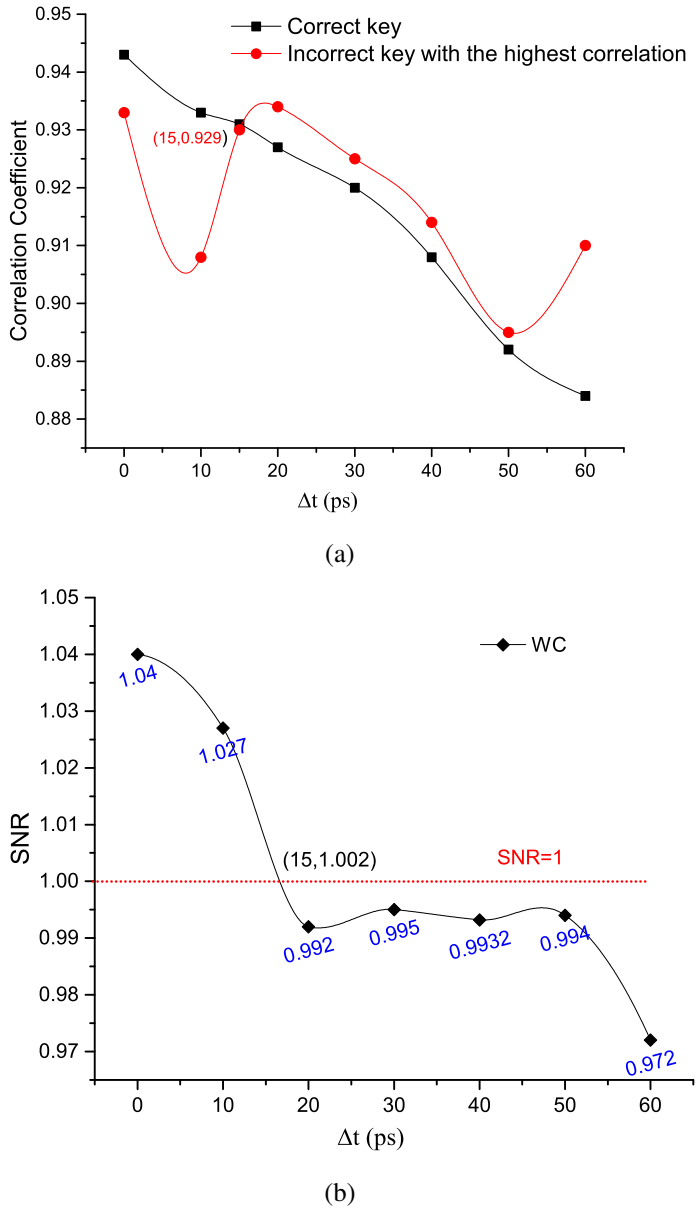


Figure 4.8: (a) Calculation of correlation coefficient for both the correct key and incorrect key when Δt increases, and (b) for the worst-case scenario, with the increase in inserted delay Δt, signal-noise ratio decreases.

As shown in Fig. 4.8(a), the delay insertion is effective only when the lower bound, Δ_{min}, is greater than 15 ps for the bus model shown in Fig. 4.7. In this case, the correlation coefficient of the correct key becomes lower than that of the incorrect key. Moreover, the security improvement of the delay insertion is illustrated in Fig. 4.8(b), using *SNR* as the metric. *SNR* decreases with Δt increases. When Δt exceeds 15 ps, *SNR* falls below 1 (dashed red line), and the EM attacks become unsuccessful.

Note that increasing Δt helps improve circuit security; however, how much Δt can be added without degrading the circuit performance needs also to be addressed. As shown in Fig. 4.9, the total bus latency is calculated from the 50% point of the earliest transition of the output of *Inv1* to the 50% point of the latest transition of the signals at the input of the receiver circuit.

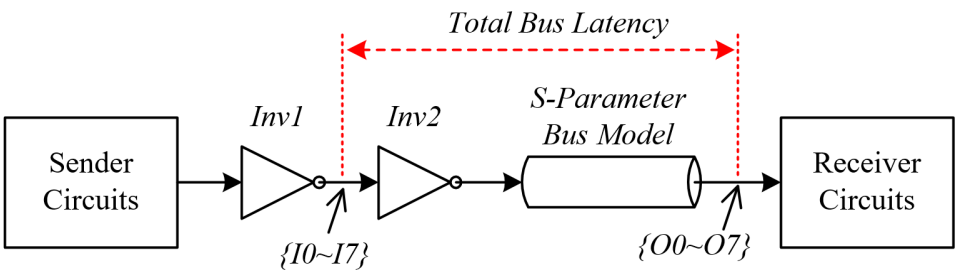


Figure 4.9: The start and end point of the measurement of total bus latency.

The simulated total bus latency with delay insertion is shown in Table 4.1 and Fig. 4.10, where the x-axis is the delay added into I_7 (Δt) and the y-axis is the bus latency. If no delay is added, the total bus latency is 240 ps, which is determined by the worst-case switching pattern scenario of middle lines (e.g. I_3). When the inserted delay Δt increases, the total bus latency remains almost unchanged as the coupling capacitance of I_3 does not change, the latency of which still dominates the total bus latency. When Δt is greater than 70 ps, the *boundary line* I_7 (added with Δt) starts taking over middle line I_3 and dominates the bus latency. Consequently, if Δt increases over 70 ps, for the specific setup, the speed of the bus starts to degrade. Thus, the intersecting point of the two curves (annotated with the square and circle markers) sets the useful upper bound of Δt (Δ_{max}).

As shown in Fig. 4.8(b), 4.10 and Table 4.1, when $\Delta t = 60\,ps$, the *SNR* drops by 6.5% (decreases below 1) and, meanwhile, the total bus latency remains unchanged to sustain the circuit performance (compared with the scenario where no delay is added).

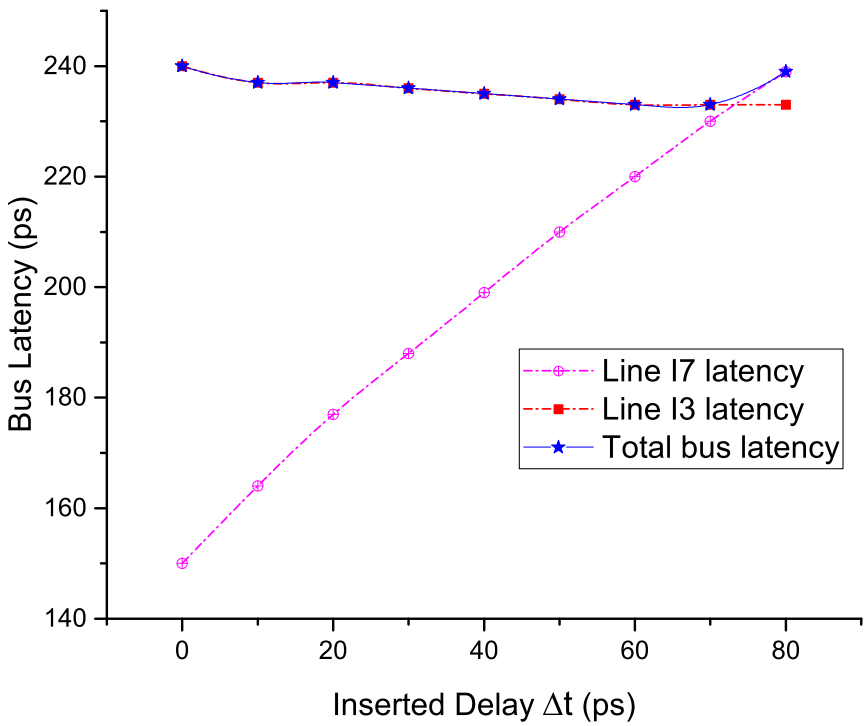


Figure 4.10: Total bus latency vs delay inserted into I_7.

Table 4.1: Added Delay into I_7 vs Total Bus Latency.

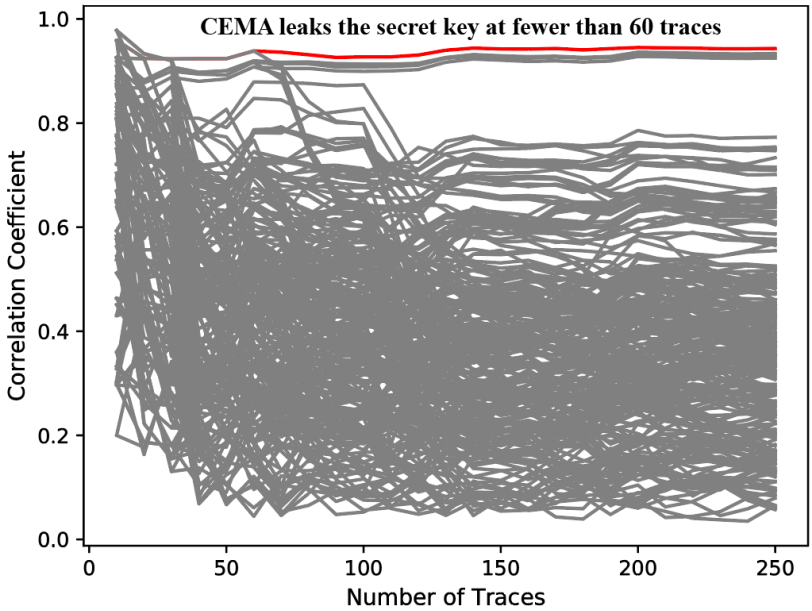
Added delay (ps)	Latency of I_3 (ps)	Latency of I_7 (ps)	Total bus latency (ps)
0	240	150	240
10	237	164	237
20	237	177	237
30	236	188	236
40	235	199	235
50	234	210	234
60	233	220	233
70	233	230	230
80	233	239	239

Signal-noise ratio and security simulation results

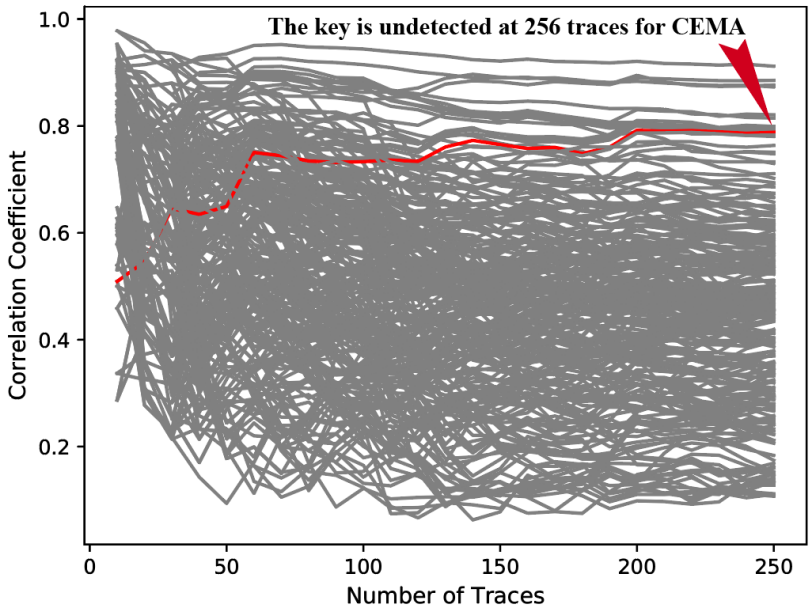
Finally, the traces needed to recover the secret key and SNR values in different attack scenarios is demonstrated. The number of traces needed to attack the secret key for both scenarios, which is widely used in the hardware security field [146], [147], [132], are illustrated in Fig. 4.11. The x-axis is the number of traces needed for a successful attack and the y-axis is the correlation coefficient. The 256 traces in each sub-figure correspond to the probability of the corresponding 8-bit key value.

As depicted in Fig. 4.11(a), when no delay is inserted, the line that corresponds to

the correct key (red line) can be distinguished from other guessed key lines with fewer than 60 traces. While if $\Delta t = 60\,ps$ is inserted into selected interconnects, more than 250 traces are recorded and yet the correct key (that corresponds to the red line in Fig. 4.11(b)) shows a low correlation coefficient and is not detected.



(a)



(b)

Figure 4.11: Correlation coefficient vs number of traces (a) where no delay is added, and (b) where 60 ps is added to the *boundary lines*.

To demonstrate that the proposed methodology is unbiased to any encryption keys, the average *SNR* of unprotected interconnects (with no delay added) and protected interconnects ($\Delta t \in (15\,ps, 70\,ps)$) with different keys is, respectively, depicted in Fig. 4.12. In the interest of space, ten randomly generated keys are listed here. As shown in Fig. 4.12, the proposed technique is not biased to a fixed key, where different delays are inserted into the interconnects, *SNR* for all listed keys falls below 1.

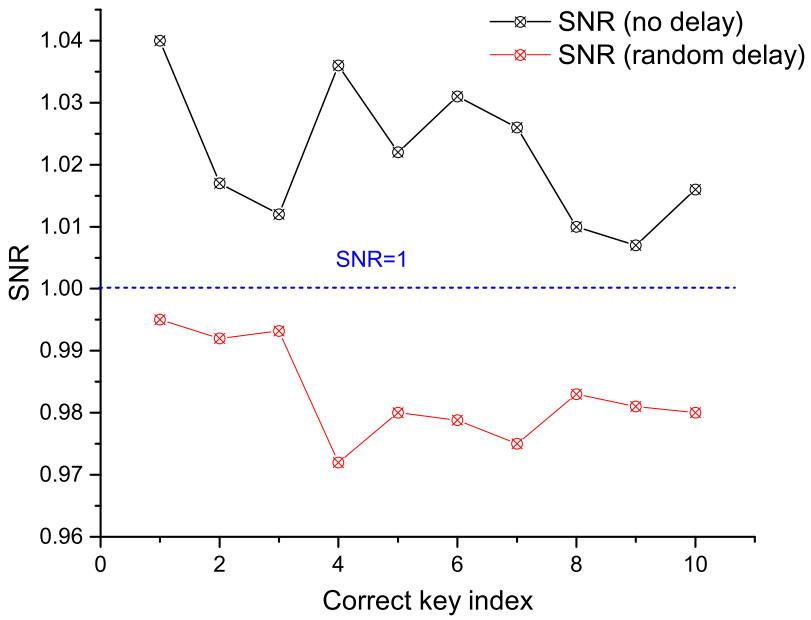


Figure 4.12: Signal-noise ratio comparison between interconnects with no delay and interconnects with a random delay ($\Delta t \in (15\,ps, 70\,ps)$) inserted, where ten different keys are also generated randomly.

4.4 Dynamic Delay Insertion with Data Bus Inversion

In the previous section, a method that introduced a specific delay on the edge lines of an off-chip bus to decrease the correlation coefficient between EM emissions and transmitted data was proposed. Nonetheless, the static nature of the method is not applicable to diverse buses and types of data. In addition, if the edge lines do not transition, there is no gain in decreasing the correlation coefficient and thus this method is ineffective in that case.

With the increasing adoption of 2.5-D integration technology, memory and logic components are integrated on the same substrate and communicate through interconnect buses on the interposer [144]. DBI is typically utilised to decrease the number of

transitions on the bus and, hence, reduce the dynamic power [148] and noise. Additionally, DBI can provide benefits on security enhancement for CPAs, and hinder the correlation between transmitted data and related EM emissions [149]. However, as shown in this section, DBI cannot help if the adversary monitors the control bit that indicates data inversion.

However, this security issue can be resolved by combing DBI with a novel dynamic delay insertion technique, where the information leakage is concealed by adding delay to bit lines based on the HD of consecutive pieces of encrypted data. This way, the delayed lines are selected dynamically and therefore hard to detect and reverse engineer. Moreover, unlike the RDI technique, dynamic delay insertion does not worsen and, in specific scenarios, improves the bus latency. This benefit is obtained by carefully selecting the lines to add delay. Thus, the new security scheme increases resistance to EM attacks for a bus, without compromising the bus performance.

The overview of DBI, the dynamic delay insertion scheme, circuit implementing this delay scheme and corresponding simulation results are analysed in the following subsections, respectively.

4.4.1 Data Bus Inversion Overview

DBI is a technique that uses limited-weight coding to reduce the number of transitions in the circuits and achieve low-power designs [148]. As a general method, DBI is best-in-class applied to the bus to reduce the I/O dynamic power dissipation as the bus may have a very large wire capacitance, especially long off-chip buses.

For an n-bit bus, the DBI technique normally includes the following four steps: 1) Calculate the HD between the current data and the data to be transmitted; 2) Compare the HD value with half of the width (n/2). If the HD value is larger than n/2, the data is reversed and the control bit is set to '1'. The next bus value is the reversed next data value. Otherwise, the control bit is set to '0' and the next bus value is the next data value. The sequence of bus values on an 8-bit bus with DBI applied is shown in Fig. 4.13; 3) If the HD value is smaller than n/2, then the data is sent directly to the bus and the control bit is set to '0'; and 4) The receive circuit recovers the data according to the value of the control bit.

It can be seen from Fig. 4.13 that one extra bit (the control bit) is required in DBI to reduce the bus activities where area overhead increases. However, the DBI technique is an optimal choice as any other coding method needs more than 1 bit to further decrease bus activities.

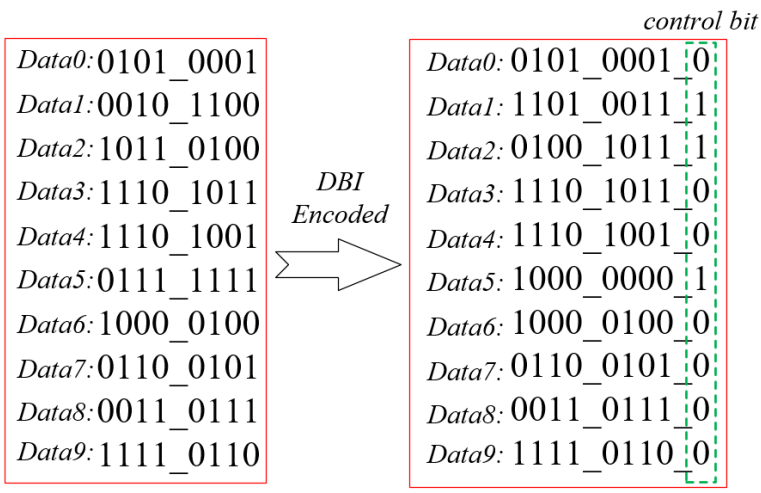


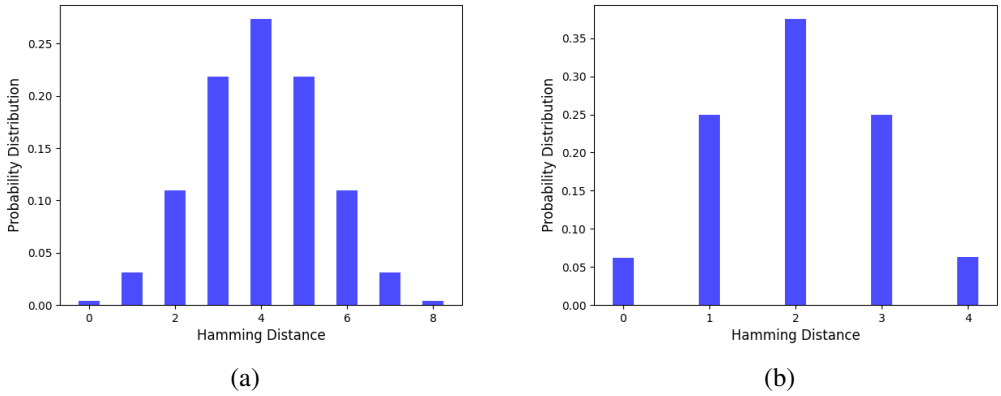
Figure 4.13: Sequence of values on 8-bit interconnects where the data bus inversion technique is applied to ten randomly generated bus data.

4.4.2 Effect of Dynamic Delay Insertion and Data Bus Inversion on Security and Bus Latency

Assuming the bus-invert coding technique, proposed in Section 4.4.1 (effectively the origin of DBI), is utilised in the 8-bit interposer-based memory bus shown in Fig. 4.16. If the HD between two consecutive pieces of data is higher than four, the data to be transmitted is inverted. In this case, the maximum HD value will be reduced from 8 to 4 and the probability distribution for the HD value between the current bus data and the next bus data is shown in Fig. 4.14(b). The function ψ in Eq. (3.2) ($L_a = \psi(PT, k_0) = \psi(HW(PT \oplus k_0))$) thereby no longer maintains the linear relationship with HD, as shown in Fig. 4.15. Due to the obfuscation between the bus leakage (L_a) and HD, the DBI technique can provide benefits for security enhancement against SCAs.

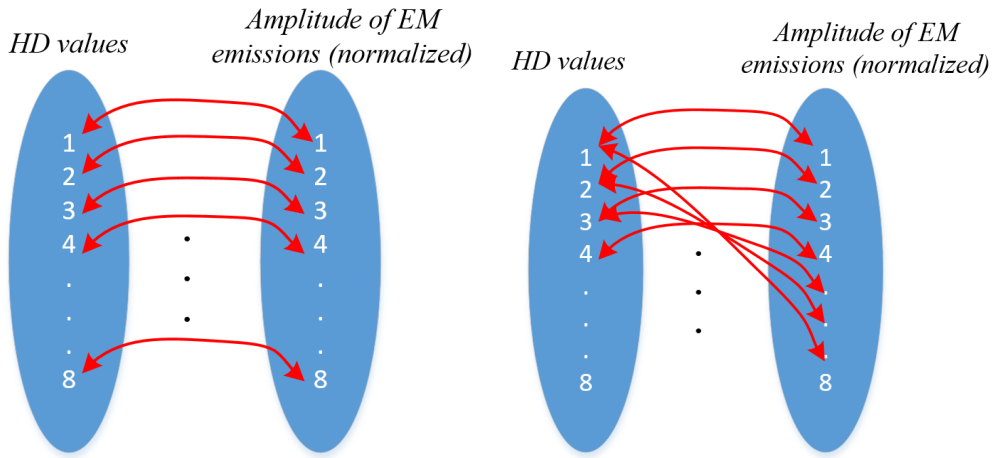
However, there exist limitations of this technique in SCA protection applications that are due to the extra control bit line. From the description in the previous subsection, it is known that this control bit indicates whether the data to be sent is inverted or not. If this bit is monitored by an adversary, DBI can lose its protection as the HD values, which linearly correspond to the data, can be reverse engineered, as will be shown in the latter simulation steps. In this case, a dynamic delay insertion technique can be utilised to fix the gap. The bit lines to be delayed are randomly chosen based on the two consecutive pieces of data at each clock rising edge, where the adversary will find it difficult to identify them.

Furthermore, this new security scheme (dynamic delay insertion combined with



(a) (b)

Figure 4.14: The probability distribution for the Hamming distance value of two consecutive pieces of data (3000 samples) for an 8-bit memory bus (a) without data bus inversion, and (b) with data bus inversion applied.



(a) (b)

Figure 4.15: The correlation between Hamming distance values of two consecutive piece of data and the amplitude of electromagnetic emissions from the bus lines (normalised) is (a) linear with no data bus inversion encoded, and (b) non-linear with data bus inversion applied.

DBI) can also provide benefits on bus performance. This is contributed by the dynamic delay insertion technique as DBI only helps reduce the dynamic power consumption by minimising the bus activity. As described in Section 4.2 and Fig. 4.16, the bus latency is proportional to the wire resistance and wire capacitance C_t, which comprises the ground capacitance C_g and coupling capacitance C_m. C_t varies when bit lines transition,

according to,

$$C_t = C_g + C_m \left| \frac{\Delta V_1}{V} \right| + C_m \left| \frac{\Delta V_2}{V} \right| = C_g + nC_m, \tag{4.10}$$

where $n \in [0,4]$, V is the supply voltage, and ΔV_1, ΔV_2 are the voltage difference between the observed line with its two neighbouring lines, respectively. The bus performance is determined by the line that drives the maximum capacitance (worst case), e.g. as shown in Fig. 4.16, when line I_1 transitions in the opposite direction than I_0, I_2, exhibits the maximum coupling capacitance equal to $4C_m$. When delay is added into lines I_0 and I_2, the coupling capacitance of I_1 is less than $4C_m$. Therefore, the latency of I_1 can be reduced and the bus performance is improved. The proposed dynamic delay insertion method aims to determine the "**victim lines**" that drive the maximum coupling capacitance among all bit lines, then add delay into specific lines to reduce the coupling capacitance of these "**victim lines**". This dynamic delay insertion algorithm will be introduced in the next subsection.

4.4.3 Dynamic Delay Insertion Algorithm

For an 8-bit bus where the DBI technique is applied as considered in this dissertation, there can be a maximum of four transitions for any piece of data. Indeed, if there were more than four ($n/2$, where $n = 8$ is the bus width), for example, $x > n/2$ transitions, according to the DBI method [148], the bus is inverted leading to $(n-x) < n/2$ transitions. Based on this observation, the following four cases list all possible transitions on the bus, as shown in Fig. 4.17. To better describe these transitions, the bus model shown in Fig. 4.16 is used, with the bus lines annotated with indices ($I_0 \sim I_7$).

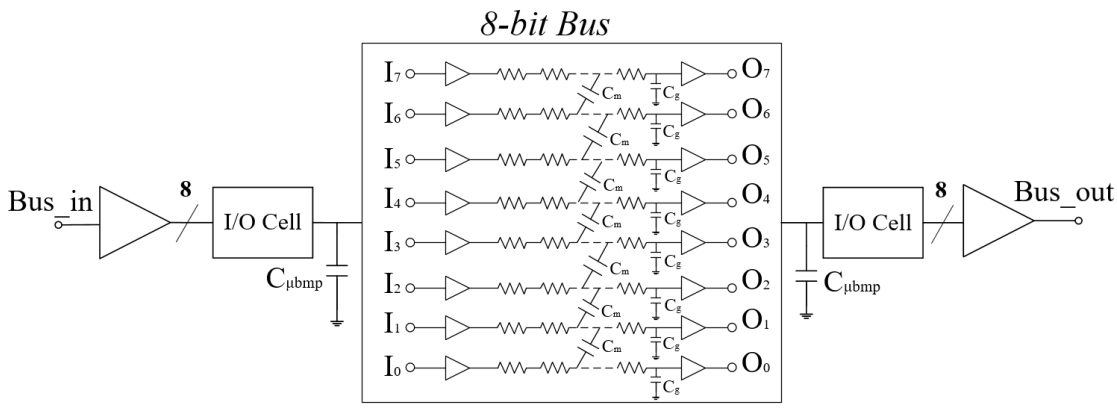


Figure 4.16: 8-bit interposer-based interconnect model.

- ***Case 1***: Up to four individual transitions (which appear alternately on the 8-bit bus), e.g. lines I_1, I_3, I_5 and I_7 transition. All but lines I_0 and I_7 drive a capacitance of $C_g + 2C_m$. Lines I_0 and I_7 drive a capacitance of $C_g + C_m$ if they transition.

- ***Case 2***: Up to two pairs of transitions, e.g. lines I_2, I_3, I_5 and I_6. If the transition is in the same direction, then the lines drive a capacitance of $C_g + C_m$ while for transitions in the opposite direction, the lines drive a capacitance of $C_g + 3C_m$.

- ***Case 3***: Three consecutive transitions and one individual transition, e.g. lines I_4, I_5, I_6 and I_2. If lines I_4, I_5, I_6 switch to opposite directions, the worst-case latency (d_{max}) of the bus is produced due to the maximum capacitance $C_g + 4C_m$ driven by line I_5.

- ***Case 4***: Four consecutive transitions, e.g. lines I_3, I_4, I_5 and I_6. If they switch to opposite directions, lines I_4 and I_5 exhibit the worst-case latency (d_{max}) of the bus as also encountered in the previous case.

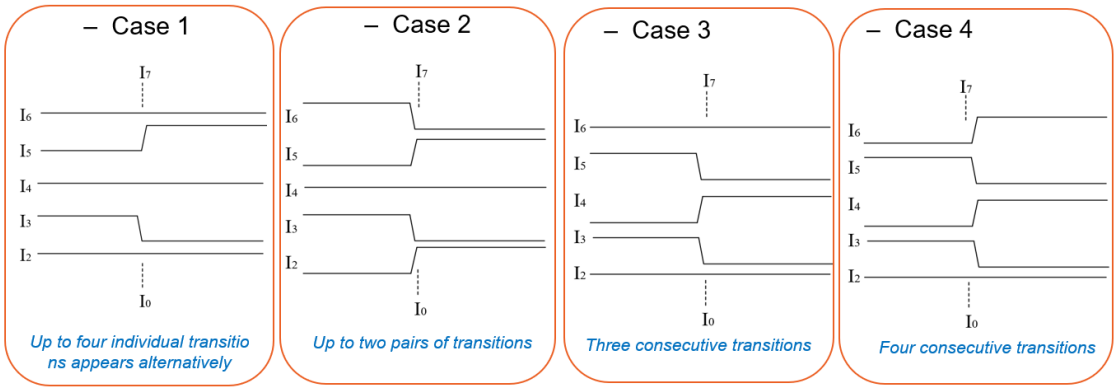


Figure 4.17: Four cases that list all possible transitions.

The pseudo-code of the delay insertion algorithm, listed as Algorithm 2, is explained through the specific examples mentioned in cases 1 to 4. Note that since DBI is applied to the bus, there is an XOR operation between the current and upcoming value of bit line x, denoted as I_{x_t} and $I_{x_{t+1}}$, respectively, where the XOR output is denoted as XOR_x. If line x transitions, $XOR_x = 1$, otherwise $XOR_x = 0$.

First, bit lines I_1~I_6 are considered. In case 1, $XOR_{I_1} = 1$ and $XOR_{I_2} = 0$ (steps 3 and 4), hence Δt is added to line I_1 and to lines I_3, I_5, I_7 as the driven capacitance is $C_g + 2C_m$. In case 2, $XOR_{I_2} = 1$, $XOR_{I_3} = 1$ and $XOR_{I_1} = 0$ (steps 5 and 6), Δt is added

Algorithm 2: Dynamic Delay Insertion Algorithm

```
1:  Input: the XOR result XOR_x between the current data I_{x_t} and upcoming data I_{x_{t+1}} for bit line x
    from the DBI circuit;
2:  for x ∈ [1,6] do;
3:      if XOR_x = 1 and XOR_{x+1} = 0 then
4:          Insert Δt to line x
5:      else if XOR_x, XOR_{x+1} = 1, and XOR_{x-1} = 0 then
6:          Insert Δt to line x
7:      else
8:          Do not insert any delay
9:      end if
10: end for
11: if x = 0 and XOR_0 = 1 then
12:     Insert Δt to line 0
13: else if x = 7 and XOR_7 = 1 then
14:     Insert Δt to line 7
15: else
16:     Do not insert any delay
17: end if
```

to line I_2 and similarly to line I_5 as the inserted delay reduces the driven capacitance to less than $C_g + 3C_m$. In case 3, $XOR_{I_4} = 1$, $XOR_{I_5} = 1$ and $XOR_{I_3} = 0$, Δt is added to line I_4 and similarly to line I_6. For line I_5, $XOR_{I_5} = 1$, $XOR_{I_6} = 1$ and $XOR_{I_4} = 1$, hence no delay is added to line I_5 (steps 7 and 8). Thus, the worst-case latency of the bus d_{max} (latency of line I_5) decreases due to the reduction of the coupling capacitance. Lastly, in case 4, Δt is also selectively added to lines I_3 and I_6 (steps 5 and 6) but not to lines I_4 and I_5 (steps 7 and 8), where the worst-case latency of the bus d_{max} (latency of lines I_4, I_5) drops due to the reduction of the coupling capacitance.

For edge lines I_0 and I_7 (which drive at most a capacitance of $C_g + 2C_m$) whenever a transition occurs, Δt is inserted into these two lines. Note that the original latency of the lines increased by Δt remains lower than the reduced d_{max}.

Succinctly, the target is to add delays to specific lines in cases 3 and 4 such that bus latency decreases and the correlation coefficient drops. For cases 1 and 2, minimum latency is not the target, rather the aim is to respect the reduced worst-case latency d_{max}.

This algorithm can also counter EM attacks on wider buses. For an 8-bit bus, the algorithm improves the worst-case latency, governed by a bit line driving the maximum capacitance of $C_g + 4C_m$. For wider buses, like 32/64/128-bit buses, the algorithm can be adapted to insert delays only when bit lines switch according to cases 1 to 4. That is, no more than four consecutive bit lines switch. If more than four consecutive bit lines switch, the wide bus can be split into small groups with 8 bit lines each, where

the algorithm can be applied to each 8-bit line group. With these approaches, the bus latency does not deteriorate.

4.4.4 Circuit Implementation of Delay Scheme

In this subsection, the specific implementation of the circuit that generates the delay is first described, followed by the evaluation of the hardware cost and power consumption of the delay circuit.

The data to be sent (off-chip) to the SBox (implemented as the ROM-based LUT in Fig. 3.7) are assumed to be launched from a register and then driven by a chain of buffers, as depicted in Fig. 4.18, where the number and size of the buffers can be chosen to satisfy the timing constraints of the bus. The delay is generated by modifying the design of BUF1 (red circle), as depicted in the inset of Fig. 4.18, where the devices of INV0 and M_4~M_7 generate Δt [150]. The delay control signal $\bar{S_x}$ for line x is determined by XOR_x, XOR_{x-1}, and XOR_{x+1}, already available from the DBI and the three gates shown in the figure.

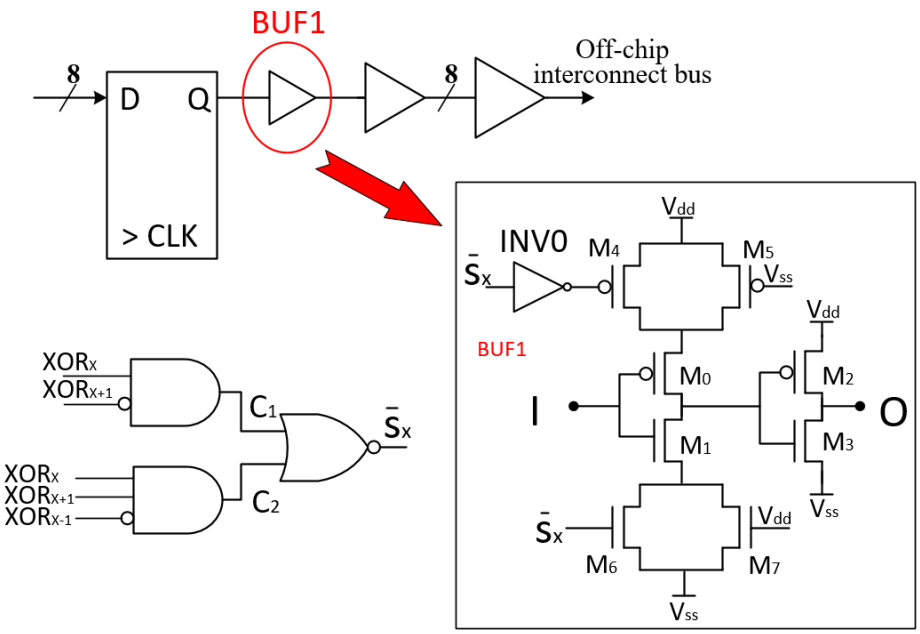


Figure 4.18: Circuit implementation of the proposed delay mechanism.

The first inverter in BUF1 block (INV0) is, respectively, pulled high through M_4, M_5, and M_0 (M_5 and M_0) and low through M_6, M_7, and M_1 (M_7 and M_1) if $\bar{S_x}$ is high (low). Thus, the delay of the inverter increases by Δt, as determined by the size of M_4~M_7, when $\bar{S_x}$ = '0'.

At each clock edge, data I_{x_t} is propagated and $\bar{S_x}$ for $I_{x_{t+1}}$ is evaluated. Thus, the correct operation of the circuit is guaranteed only if the delay for generating signal $\bar{S_x}$ is sufficiently higher than the delay required for data I_{x_t} to propagate through the register (t_{CtoQ}) and BUF1. The circuit that generates $\bar{S_x}$ includes a delay path with XOR, AND, NOR and t_{CtoQ}. The simulated arrival time of $\bar{S_x}$ for three corners, the typical (TT), fast-fast (FF), slow-slow (SS), is, respectively, 110 ps, 94 ps and 158 ps longer than the data propagating through the register and BUF1.

The delay circuit is simulated for a *UMC*65 nm technology [151] and induces a low hardware area cost of about 160 MOS transistors (INV0, M_4~M_7, 2-input AND, 3-input AND, and 2-bit NOR for each bit). Meanwhile, as the calculation of XOR is included in the DBI circuit, the power does not considerably increase either, where the consumed power is $\sim$104 μW, compared with a total power of 138 mW for the CMOS AES circuit [152].

4.4.5 Bus Performance with Delay Inserted

The latency of each bit line is the 50% delay from the output of the register to the far-end of the bus, as shown in Fig. 4.9. The bus latency is determined by the worst-case bit line latency.

Diverse Δt can be obtained by adjusting the size of transistors in BUF1. By adjusting the width of M_7 and M_5, Δt can reach up to 211 ps and 177 ps, where a bit line transitions from 0 to 1 and 1 to 0, respectively.

The simulated total bus latency with delay insertion for the four cases mentioned in Section 4.4.3 is shown in Fig. 4.19, where the x-axis is the delay Δt added to specific bus lines and the y-axis is the worst-case bus latency. As shown in Fig. 4.19, for all four cases, if no delay is added, the worst-case latency of the bus d_{max} is 284 ps, determined by the middle line latency in cases 3 and 4 (e.g., line I_5 in case 3). For the specific setup as illustrated in Fig. 4.19, adding a delay Δt of up to 50 ps for all possible switching scenarios (see cases 1 to 4 in Section 3.4.1), the bus latency reduces to 263 ps.

Based on the premise that Δt is chosen during the design process such that the bus performance does not degrade. The effect of the delay on mitigating EM attacks will be explored in the next subsection.

The physical characteristic of the circuits will vary due to process variation and ageing, which will affect the margin of the inserted delay. However, these two factors are not considered in the delay insertion schemes proposed in this chapter. The effect

of process variation and ageing will be discussed at the end of this chapter and future works in Chapter 6.

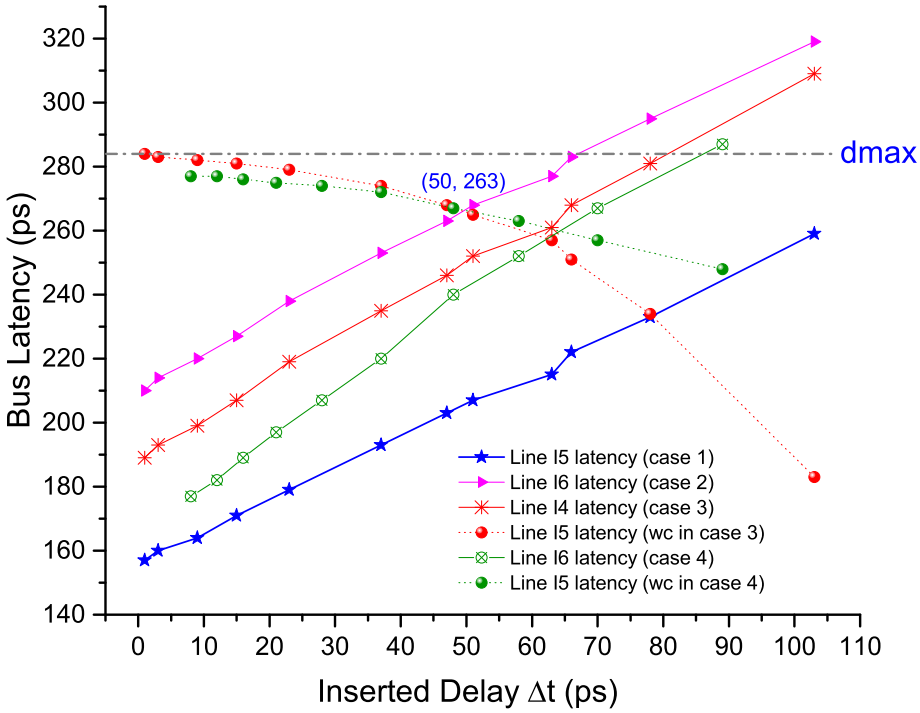
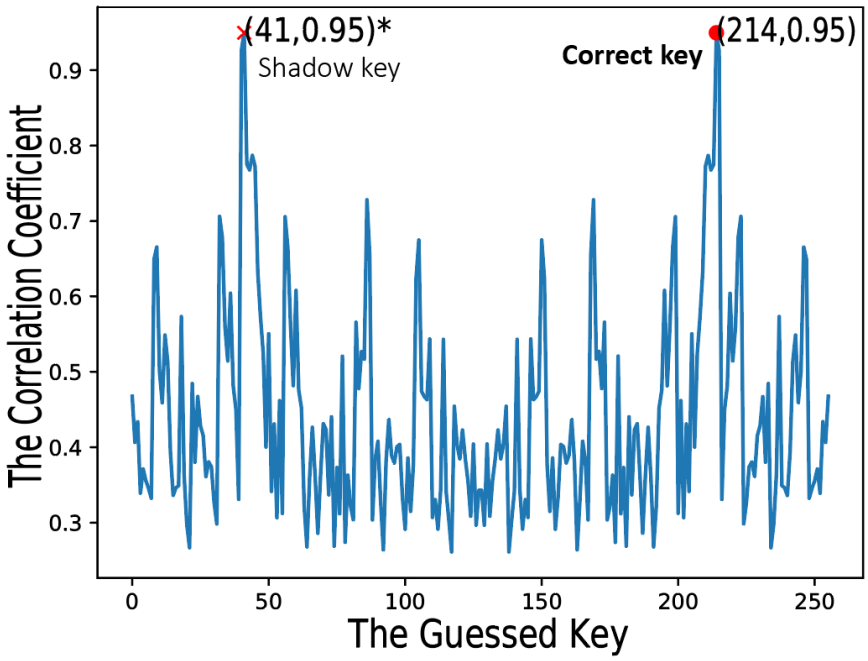


Figure 4.19: Bus latency vs delay inserted into specific bus lines.

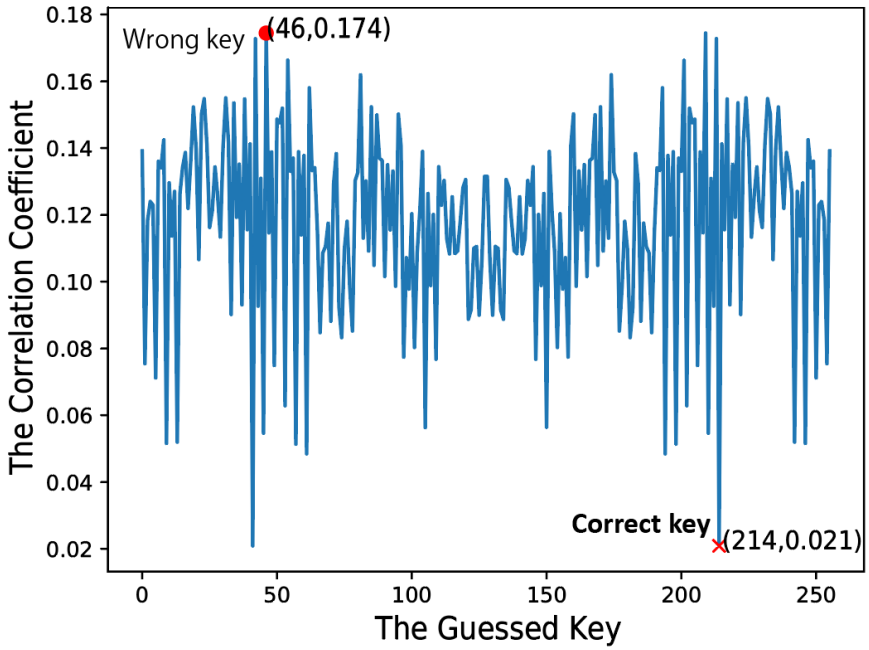
4.4.6 Bus Security against Electromagnetic Attacks

Firstly, the security benefit from the DBI method is analysed. Here, 256 EM traces (simulation) are, respectively, collected where the bus is unprotected and with DBI applied. As shown in Fig. 4.20, for the unprotected bus, the correct key 214 with maximum correlation coefficient 0.95 can be retrieved within 256 traces (50 traces are sufficient to obtain the key and *SNR* is 1.025), while for the bus with DBI applied, the detected key is 46 and *SNR* drops to 0.12, where the correct key 214 is not detected and exhibits a low correlation coefficient.

Indeed, DBI can secure the bus against EM attacks. However, the DBI method requires one extra control bit, which determines whether the data to be transmitted should be inverted. If this control bit line is monitored by the attacker (e.g., through another high-resolution probe), the correct key can be obtained within 256 traces (80 traces are sufficient to obtain the correct key 214 and SNR is 1.103), as shown in Fig. 4.21(a), where the security protection only from DBI does not suffice. To avoid



(a)

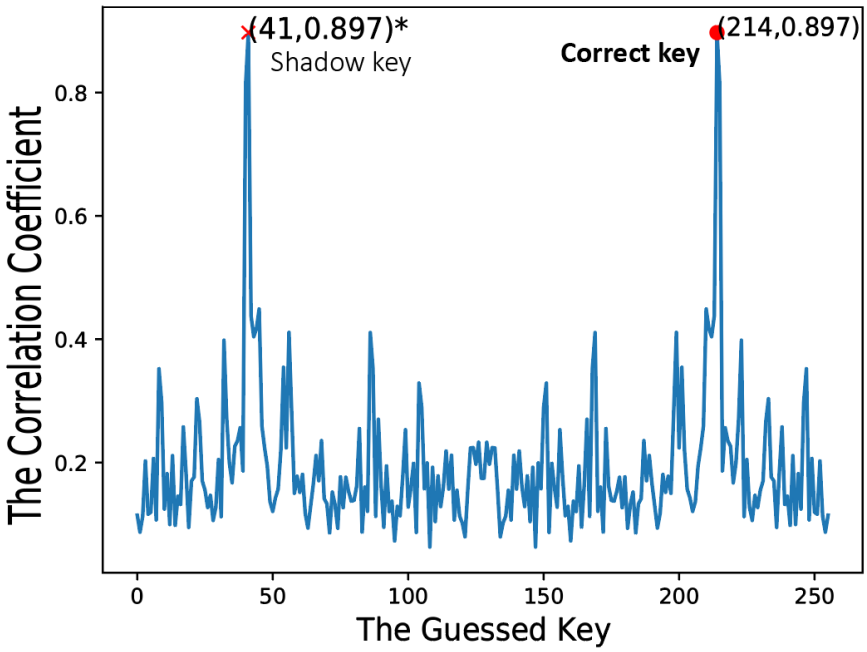


(b)

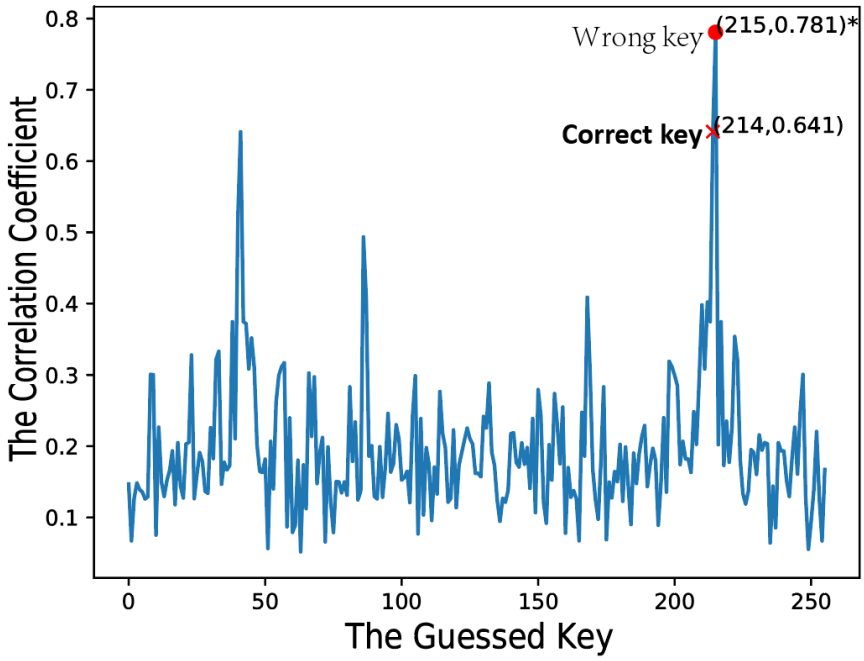
Figure 4.20: Electromagnetic attack results after 256 traces for (a) unprotected bus, and (b) bus where data bus inversion is applied.

this risk, the dynamic delay insertion technique enhances DBI to reinforce protection against such attacks.

Applying delay insertion combined with DBI to the bus lines, where the control



(a)



(b)

Figure 4.21: Electromagnetic attack results after 256 traces for (a) bus with data bus inversion applied, and (b) bus with both dynamic delay insertion and data bus inversion applied. The control bit is monitored by the attacker in both (a) and (b).

bit line for DBI is assumed to be monitored by the attacker, the EM attack result is illustrated in Fig. 4.21(b). The correlation coefficient of the correct key is no longer

the highest within 256 traces and SNR drops to 0.88. As depicted in Fig. 4.19, when Δt equals 50 ps, d_{max} decreases by 9.5% (compared to the scenario with no added delay), offering higher performance in addition to greater security.

4.5 Chapter Summary

This chapter focuses on the protection of interposer-based off-chip memory buses from EM SCAs, which were explored from the attacker's perspective in Chapter 3.

One possible countermeasure against SCAs on on-chip power networks is RDI, which reduces the correlation between the power consumption and the processed data. However, RDI can degrade the circuit performance by introducing random delays. To overcome this limitation, a novel delay insertion scheme that only adds delay to the *boundary lines* of the bus is proposed in this chapter. These lines have lower capacitance than the lines that determine the worst-case bus latency, so adding delay to them does not affect the circuit performance. The added delay enhances the security of the bus against EM attacks by lowering the correlation between EM emissions and data transmission, making this method effective and practical with negligible overhead. It is shown from the simulation results that the technique decreases SNR below 1, which prevents EM attacks, and does not increase the (worst-case) bus latency, maintaining the overall circuit performance. Therefore, the proposed method provides a superior EM SCA mitigation method compared to the state-of-the-art. Theoretical analysis and simulation results demonstrate that the new technique can offer the same level of protection against SCAs with better performance than other hardware RDI countermeasures.

However, this static delay insertion scheme has some limitations in its applicability to different buses and data types. Moreover, if the data on the *boundary lines* remains unchanged, the correlation coefficient is not affected, and no protection is provided in this case.

To address this problem, another innovative technique that combines an energy-efficient data inversion technique with dynamic delay insertion, is then proposed in this chapter. The added delay improves the resistance against EM attacks for the cryptographic circuit without performance degradation and, in some cases, even improves the performance. Simulation results on a set of EM traces, captured from an 8-bit interposer-based off-chip memory bus, confirm the efficiency of the proposed technique by decreasing SNR below 1 and improving the worst-case bus latency by 9.5%.

Moreover, this technique has low area and power overheads, as it only requires 160 MOS transistors and consumes about 104 μW of power, which is less than 1% of the total AES circuit (based on $UMC65$ nm technology).

Nevertheless, the dynamic delay insertion technique has its own limitations. Two issues that affect the delay insertion schemes are: **process variation** and **ageing**. Process variation denotes the variations in the physical parameters of transistors and wires during the fabrication process, which can result in different delay values for the same buffer or wire instances and impair the accuracy and consistency of the delay insertion scheme. Ageing denotes the deterioration of the transistor performance over time due to various mechanisms, which can alter the delay values that are inserted into the bus lines and compromise the stability and reliability of the delay insertion scheme. Moreover, these two issues can interact with each other and cause non-uniform delay degradation across different parts of the circuit, which can undermine the balance and effectiveness of the delay insertion scheme. Therefore, in addition to dynamically selecting the bus lines to insert the delay, different delay values can also be implemented to account for both process variation and ageing issues, for enhanced resilience against EM attacks. This delay insertion scheme will be further discussed in future works of Chapter 6. The following chapter will present the experimental validation of the EM attack and protection methods proposed in Chapters 3 and 4.

Chapter 5

Demonstration of EM Attacks and Protections

5.1 Chapter Overview

This chapter evaluates the efficiency and security of the implementations of techniques described in previous chapters for EM attacks and mitigation. The main contributions of this chapter are as follows: 1) physical attacks on an 8-bit bus through the CEMA method; and 2) experimental demonstration of the effectiveness of the *static* delay insertion technique presented in Chapter 3 and the *dynamic* delay insertion technique presented in Chapter 4.

The experiments illustrated in this chapter are restricted to a low-frequency board-to-board test platform. However, the theory that underpins the demonstrated solutions should protect any communication channel (on-chip or off-chip). As a representative joint test action group (JTAG) interconnect is adopted here, it is believed that these experiments can stand as proof of concept that the delay-insertion solutions can help prevent EM attacks for the off-chip interconnects in a 2.5-D system.

5.2 Experimental Setup

The model of the CEMA attack and countermeasures derived in the previous chapters are validated through measurements on FPGA implementations of the 128-bit AES algorithm. The test platform, experimental settings and test procedures are described in the following subsections, respectively.

5.2.1 Advanced Microcontroller Bus Architecture-Advanced High-Performance Bus

Without loss of generality, the AHB bus standard from the advanced microcontroller bus architecture (AMBA) produced by ARM [4] is utilised as the memory bus to connect the encryption chip and memory chip in the 2.5-D system adopted in this dissertation. AMBA–AHB can interconnect all the modules such as high-performance processors and high-bandwidth random access memories (RAMs) to provide a high communication throughput. Normally, AMBA–AHB consists of four components: **master**, **slave**, **arbiter** and **decoder**, as shown in Fig. 5.1. The data-transfer request (read/write) is initiated by the master components, with the address decoded and responded to by the slave components, where the data-transfer flow is controlled by the logical components such as arbiters and decoders. For simplicity, a particular case (one master, one slave) is discussed here.

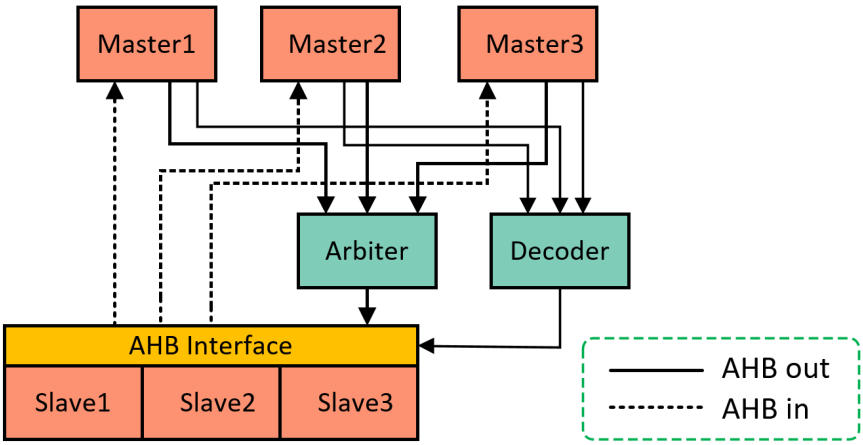
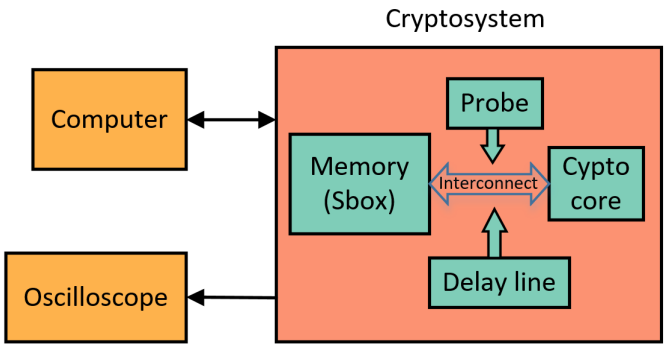


Figure 5.1: Structure of AMBA–AHB standard bus [4] (take the Master reading data from the Slaves, for example).

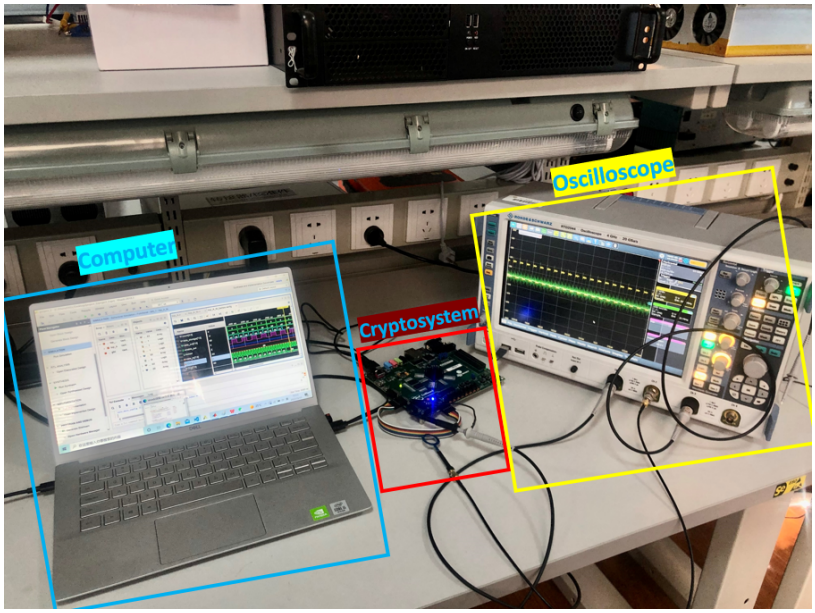
5.2.2 Hardware

To evaluate the effectiveness of the methodologies for and against EM attacks proposed in the previous chapters, the Xilinx Zynq-7000 programmable system on chip (SoC) on a Zedboard, which integrates a dual-core ARM Cortex-9 processor and programmable logic cells up to 6.6 M [153], executes the AES algorithm. Xilinx Vivado 2019.2 tools are used to synthesise the hardware components, which have been implemented using a hardware description language (Verilog) [154], by using 128-bit keys and LUT-based Sbox.

The experimental setup for conducting the EM SCA is illustrated in Fig. 5.2(a). The prepared plaintext stream is sent from the computer to the Zedboard. After the initial parameter settings, the encryption is performed in FPGA and the XORed result of the plaintext and subkey is transmitted through the bus interconnects. A magnetic field probe (9.0 kHz to 3.0 GHz) placed vertically captures the magnetic field emitted from the interconnects. The probe's subminiature version A (SMA) output is connected to an oscilloscope (ROHDE&SCHWARZ, 4 GHz, 20 GSa/s), as shown in Fig. 5.2(b).



(a)



(b)

Figure 5.2: (a) Experimental setup for conducting the experimental electromagnetic side-channel attack, and (b) the mapping of the different blocks to the corresponding instruments, from left to right: computer, cryptosystem, and oscilloscope.

The detailed mapping of each block in the cryptosystem is depicted in Fig. 5.3. The

data from the cryptocore and the Sbox are stored in separate FIFOs in the FPGA and communicated through the wires that connect two JTAGs, which act as the interconnect. The delay line is implemented using a primitive **IDELAY**from the FPGA, which will be discussed in section 5.3.2. In the following experiments, the AES encryption runs at 200 MHz, which is the same as the bus data rate, and the sampling rate of the oscilloscope is set to 10 GS/s.

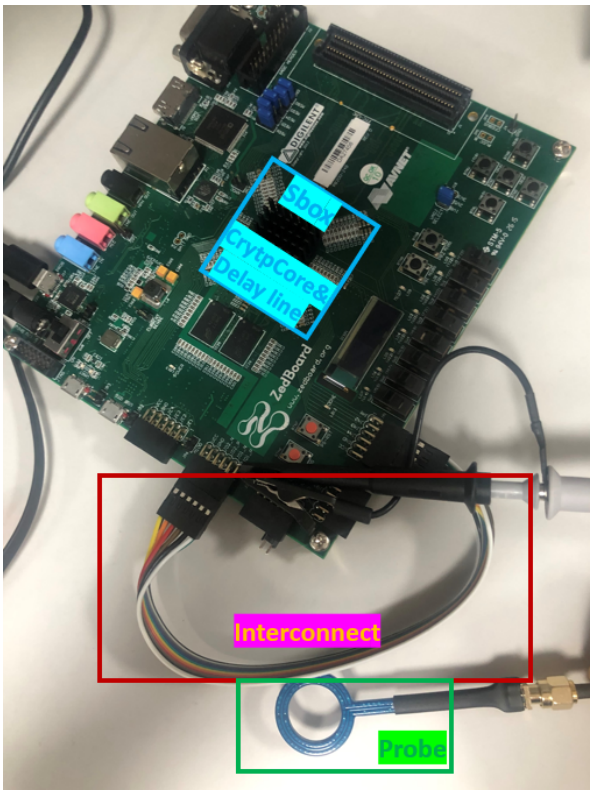


Figure 5.3: Mapping of the cryptosystem on Zedboard.

5.2.3 Obtaining Traces

The set of EM traces (voltage values) can be collected from the Zedboard through several steps. In this section, trace capture, trace alignment technique and trace characterisation will be discussed, respectively.

Zedboard

During the encryption process starts, the plaintexts and keys are read from preloaded files and encrypted by the AES algorithm. The data acquisition timing is synchronised by *enable* signal (Channel 1) via the oscilloscope to monitor the magnetic field

generated by the encryption process. The transient current of *enable* is obtained by measuring the differences between VDD and GND. The time-domain waveforms observed from the magnetic field probe (Channel 2) and the *enable* pin (Channel 1) are shown in Fig. 5.4, respectively.

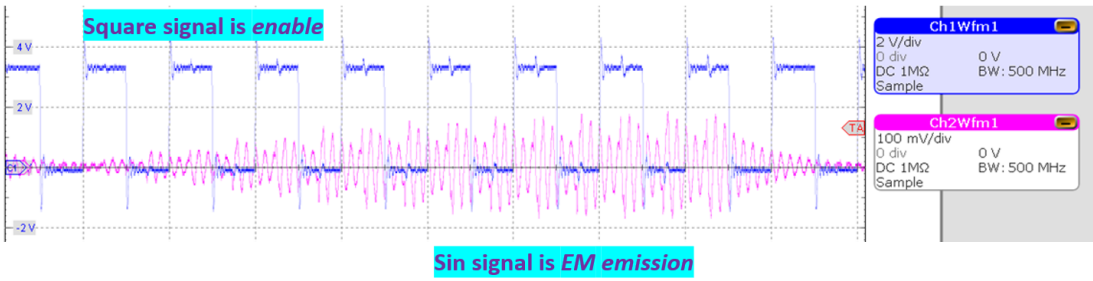


Figure 5.4: Captured electromagnetic traces (Channel 2) and the *enable* signal (Channel 1).

Trace alignment

As shown in Fig. 5.4, the initial-round encryption process of AES happens when *enable* is pulled high, where the EM traces located in this period are useful for the attacks and need to be extracted. Due to the mechanism of CEMA, only the calculation of the correlation between the sampling point and the calculated HD **at the same timing point** for sets of traces can correctly recover the sensitive key. Therefore, one important trace preprocessing step is trace alignment, for mounting a successful and efficient EM attack.

Due to the fact that monitoring of the emanated EM field happens on the board-to-board interconnects, it can be assumed that the radiated EM field is affected less by other logic components operating on the board. Thus, the voltage amplitude captured at the probe terminal when encryption starts is usually higher than that captured in the remainder of the period. Moreover, the round peak detection method has proven that the index of the calculated power peak indicates the position in the time domain that has the maximum power value at low frequencies [155]. Therefore, a peak-searching algorithm is utilised (line 14 in Code Listing 5.1) to extract the points of interest when *enable* is high (the period when encryption is processed). After each peak in each cycle is fixed, an extra 1000 points are selected (line 19, the number of points is chosen based on the overall points existing between two peaks) to form a new .csv file, where the trace segments that include the sensitive information are cut and aligned.

```python
import pandas as pd
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
import numpy as np

enable = pd.read_csv("enable.csv", names=['enable'])
s =  pd.read_csv("EM_emissions.csv", names=['wav'])

df = enable.join(s,how='inner')
#0 1 handling
df['enable']= df['enable'].map(lambda x: 1 if x>2 else 0)

#peak_search algorithm
peaks, _ = find_peaks(df['enable'], height=0)

#find 1000 extra points, from peaks_i-500 to peaks_i+500
emission = []
for i in range(len(peaks)):
    out = df.loc[peaks[i]-500:peaks[i]+500,['wav']].T
    emission.append(list(out.loc['wav']))
np.save('emission.npy',np.array(emission))
```

Code Listing 5.1: Python code used for alignment of trace segments.

Assuming the traces are captured within D *enable* cycles, by using the peak detection mentioned above, the EM trace set is composed of D traces and each trace has M samples. $T(i,M)$ demonstrates the i_{th} EM trace.

Leakage characterisation

For the implementations described in Section 3.2.3, the XORed result of the plaintext and the sensitive key is sent through the off-chip interconnects to find the substitution data. One reason that it is chosen as the attack target is that if each byte of the 128-bit key can be attacked individually, the measurements to recover the overall key can be highly reduced from $2^{128} = 3.403 \times 10^{38}$ to $16 \times 2^8 = 4096$. Furthermore, when the attack on the 8-bit off-chip bus is successful, the techniques can also be extended to wider buses and on-chip buses.

For the EM leakage, 50,000 leakages of the encryptions with randomly generated plaintexts and a fixed key are measured, where the HD value of the bus data is swept from 0 to 8. The linear relationship between the amplitude of the measurements and the HD values is shown in Fig. 5.5. For now, each trace includes 1001 points, which is

essentially the size of a relevant sample of instantaneous measures for EM radiations after trace alignment. To further reduce the dimensions of data sets, the signal-to-noise ratio (different from its definition in Section 2.6 and defined as the ratio between the variance of signal Var_{signal} and the variance of noise Var_{noise} in this subsection) of each point is estimated to identify the best points of interest for the attack, as shown in Fig. 5.6. By combining the results shown in Fig. 5.6 and the peak-searching algorithm proposed in Section 5.2.3, a new data set is chosen that can provide the most information to launch the attacks with the maximum likelihood.

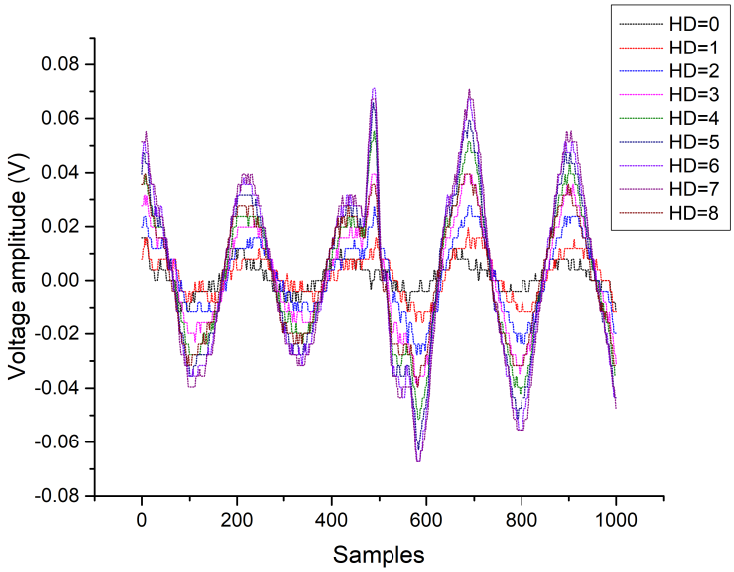


Figure 5.5: The linear relationship between the amplitude of captured voltages and the Hamming distance values (demonstration of the effectiveness of the Hamming weight leakage model).

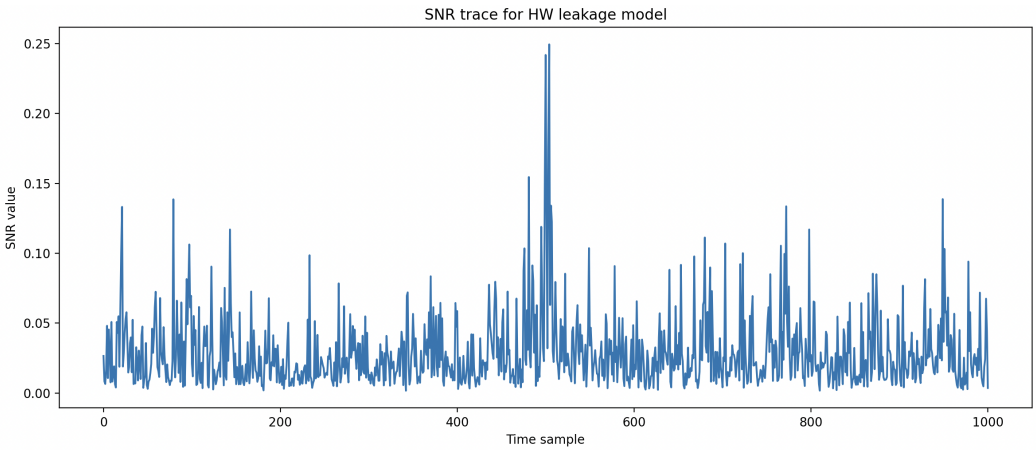


Figure 5.6: Signal-noise ratio value on each leakage point.

5.3 Evaluation and Results

In the following sections, the methods introduced in Chapter 3 and Chapter 4 are validated with experimental results, respectively. The well-known HW leakage model is applied to all cases and the attack method is limited to CEMA. The figures of merit introduced in Section 2.6 are adopted for the evaluation of system security. The results of the EM attack and protection are well described and compared.

5.3.1 First Attack: Electromagnetic Attacks on the Initial Round of Advanced Encryption Standard

Attack description

The dependencies between the EM radiation from the interconnects and the transmitted data is evaluated in the first experiment. Based on the leakage characterisation described in Section 5.2.3, the experimental attacks are implemented on an 8-bit board-to-board bus, where a 128-bit software AES is running on Zedboard. The plaintexts are fed into the memory IP that is integrated on the FPGA board through a .coe file.

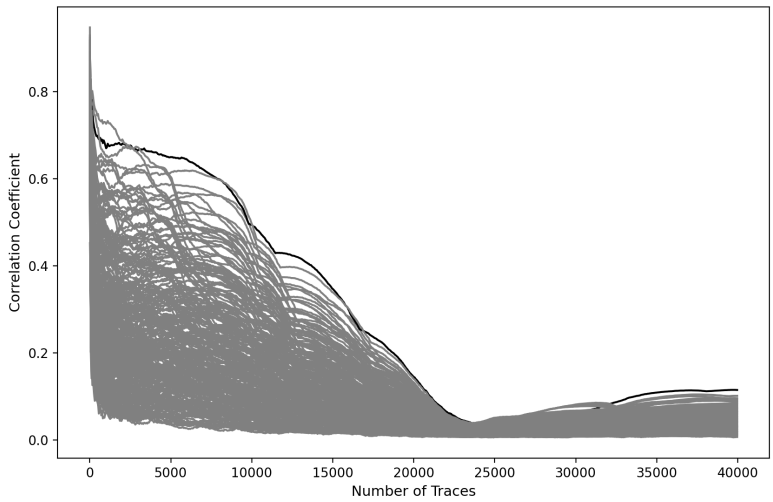
Experimental results

The experiment is produced using 40,000 EM traces (measurements) that are collected from the oscilloscope. As shown in Fig. 5.7, the correct key **HEX1A (=DEC26)** is obtained with less than 35,000 traces with a maximum correlation coefficient of 0.0506. The expression of [26, 0.0506] shown in Fig. 5.7 corresponds to the guessed key and its correlation coefficient, respectively. Furthermore, the SNR value is 1.008, which is higher than 1 demonstrating the security vulnerability of the target.

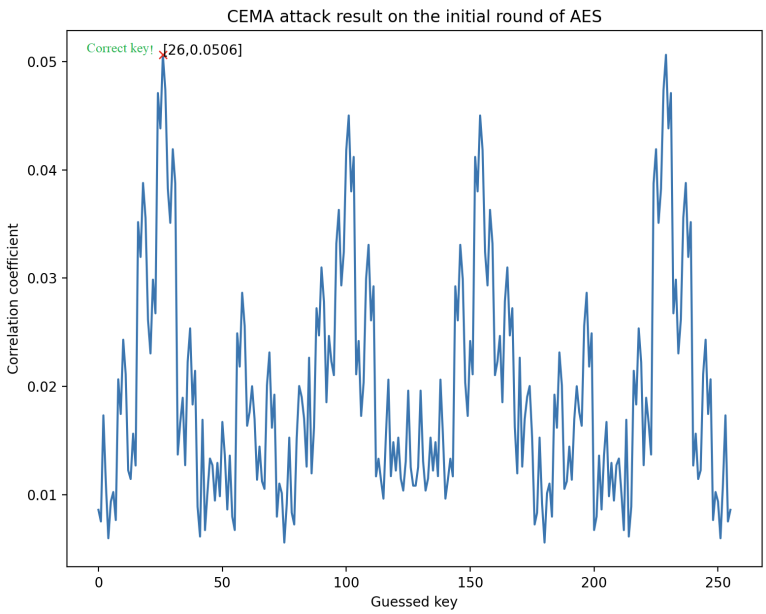
When no technique is applied to the bus, the correct key (that corresponds to the black line in Fig. 5.7(a)) exhibits a high correlation coefficient and can be detected. The required number of measurements is less than 40,000. Thereby, a successful physical EM attack on an 8-bit off-chip bus is demonstrated.

5.3.2 First Protection: Static Delay Insertion

In the next two subsections, the experiment results of two protection schemes against the previous EM attack will be provided with details and then compared. Both protection schemes aim to reduce the dependencies between the leakage and the sensitive



(a)



(b)

Figure 5.7: Attack results of the unprotected 8-bit bus. (a) Correlation coefficient vs. number of traces. The correct key (the black line) can be steadily detected after 35,000 measurements, and (b) the key with the maximum correlation coefficient is the correct key (SNR = 1.008).

data by inserting a certain amount of delay into different bus lines. Thereby, the generation of the required delay will be introduced first.

The I/O instance **IDELAY** is used to generate the required delay and its primitive is shown in Fig. 5.8 [153]. The critical ports that need settings for the delay generation are shown in Table 5.1 and Code listing 5.2.

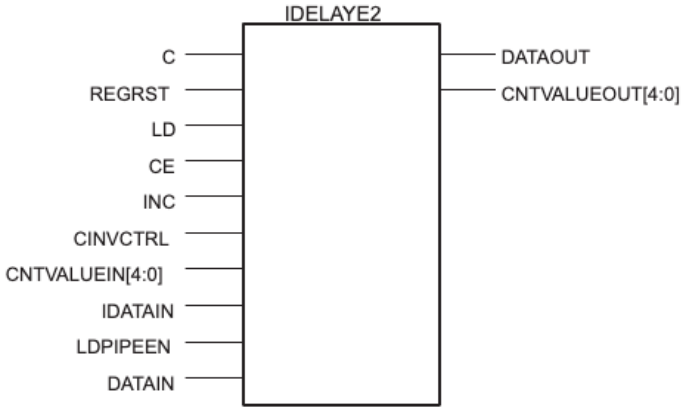


Figure 5.8: IDELAYE2 primitive.

Table 5.1: Parts of primitive ports of IDELAYE2

Port Name	Direction	Width	Function
IDATAIN	Input	1	Data input for IDELAY from the IBUF
LD	Input	1	Load IDELAY_VALUE input (0-31)
DATAOUT	Output	1	Delayed data from the input data
CNTVALUEOUT[4:0]	Output	5	Counter value for monitoring tap value

```
IDELAYE2 #(
   .CINVCTRL_SEL("FALSE"), // Enable dynamic clock inversion (FALSE, TRUE)
   .DELAY_SRC("DATAIN"), // Delay input (IDATAIN, DATAIN)
   .HIGH_PERFORMANCE_MODE("TRUE"), // Reduced jitter ("TRUE"), Reduced power ("FALSE")
   .IDELAY_TYPE("VARIABLE"), // FIXED, VARIABLE, VAR_LOAD, VAR_LOAD_PIPE
   .IDELAY_VALUE(cnt), // Input delay tap setting (0-31)
   .PIPE_SEL("FALSE"), // Select pipelined mode, FALSE, TRUE
   .REFCLK_FREQUENCY(200.0), // IDELAYCTRL clock input frequency in MHz (190.0-210.0, 290.0-310.0).
   .SIGNAL_PATTERN("DATA") )// DATA, CLOCK input signal

IDELAYE2_inst (
   .CNTVALUEOUT(delay_cnt), // 5-bit Counter value output
   .DATAOUT(data_out[j]), // Delayed data output
   .C(1'b1), // Clock input
   .CE(1'b1), // Active high enable increment/decrement input
   .CINVCTRL(1'b0), // Dynamic clock inversion input
   .CNTVALUEIN(1'd0), // 5-bit Counter value input
   .DATAIN(Drg[j]), // Internal delay data input
```

```
19      .IDATAIN(1'b0), // Data input from the I/O
20      .INC(1'b0), // Increment / Decrement tap delay input
21      .LD(1'b1), // Load IDELAY_VALUE input
22      .LDPIPEEN(1'b0), // Enable PIPELINE register to load data
    input
23      .REGRST(1'b0) );// Active-high reset tap-delay input
```

Code Listing 5.2: Verilog instantiation template of IDELAYE2.

The patterns of the delay generated by **IDELAY** are divided into two types: fixed-delay pattern and variable-delay pattern, where the emphasis is put on the latter one. Under the variable-delay pattern, the total amount of delay is determined by *TAP* and the number of *TAP* (.IDELAY_VALUE (VALUE)). *TAP* represents the minimal delay cell and can be calculated by the frequency of reference clock ($RefCLK$), denoted as $TAP = 1/(64 * RefCLK)$. For some frequencies that are commonly used in FPGA, the value of *TAP* is listed as follows,

RefCLK = 200 MHz	TAP = 78 ps
RefCLK = 300 MHz	TAP = 52 ps
RefCLK = 400 MHz	TAP = 39 ps

Protection description

For the first protection scheme, the delay is added to two *boundary lines* to resist EM attacks by shifting the transition time of bit 0 and bit 7. The added delay is swept from *TAP* to *4*TAP*, which ranges from 78 ps to 312 ps. The bus data rate is set to 200 MHz.

Experimental results

The EM attack results when the first protection scheme is applied to the 8-bit bus are illustrated in the following figures. Here GE is defined as the position (ranking) of the correct key among all key candidates. If the guessed key is the correct key, the GE value will be zero (the first place). As shown in Fig. 5.9, when the added delay is 1**TAP*, GE reaches zero and remains at this place after approximately 25,000 traces, while for the other three cases, GE cannot reach zero even with 70,000 traces. Next, SNR, MTD and correlation coefficient are used to further illustrate the attack results.

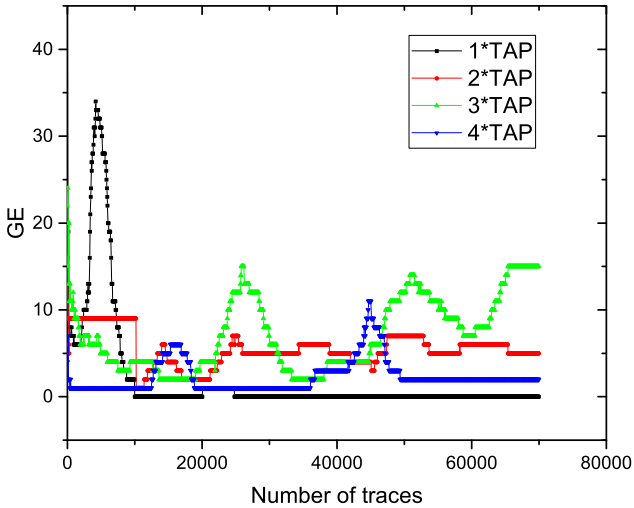


Figure 5.9: Guessing entropy vs number of traces for four delay insertion scenarios.

As shown in Fig. 5.10, when the added delay is 1**TAP*, (complement of) the correct key can be obtained with an MTD of 25,000 and a maximum correlation coefficient of 0.659. This result proves that there is a lower bound for the added delay to provide the effective security protection proposed in Section 4.3.3.

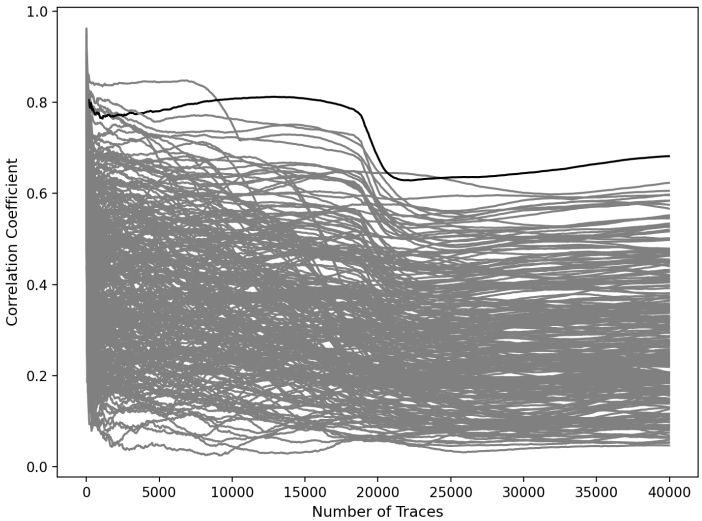
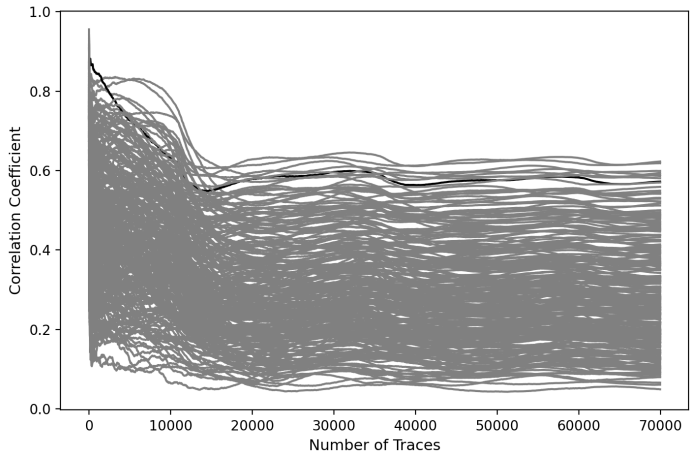


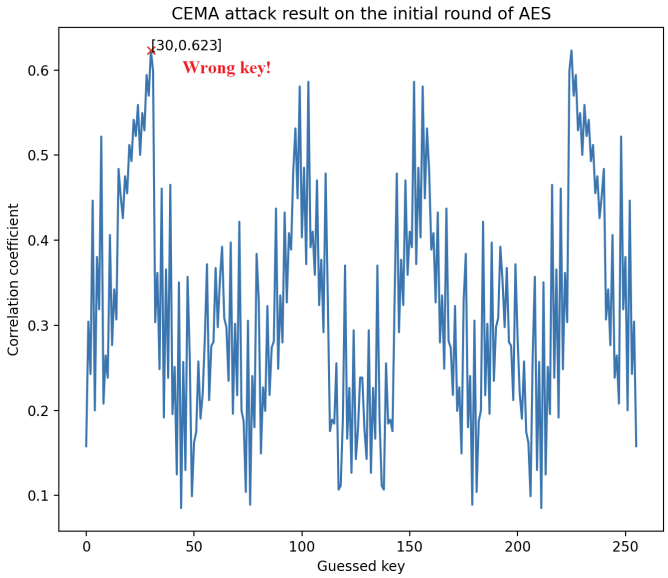
Figure 5.10: Attack results of the 8-bit bus when added delay is 1**TAP*. The complement of the correct key (**DEC224**, the black line) can be detected in approximately 25,000 traces (SNR = 1.063).

When the added delay is more than 2**TAP*, the correct key cannot be extracted even with more than 70,000 measurements, as shown from Figs. 5.11 to 5.13. However,

the attacked wrong keys are close to the correct key (one or two-bit errors), which exhibits the limitations of the *static* delay insertion technique. In the worst case, when the added delay (1**TAP*) is less than the lower bound, the timing pattern between the EM leakage and the sensitive day cannot be disrupted and the correct key can still be extracted (Fig. 5.10). Motivated by this limitation, a second protection scheme is pinpointed to provide stronger security protection against EM attacks.

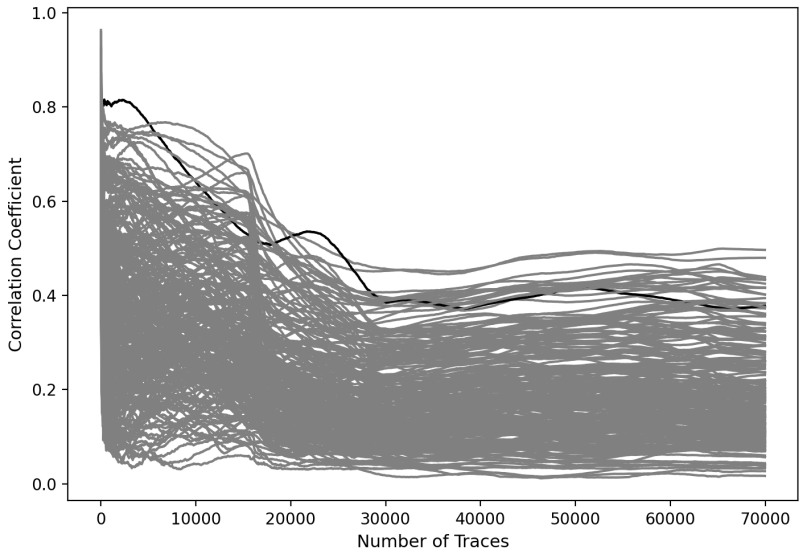


(a)

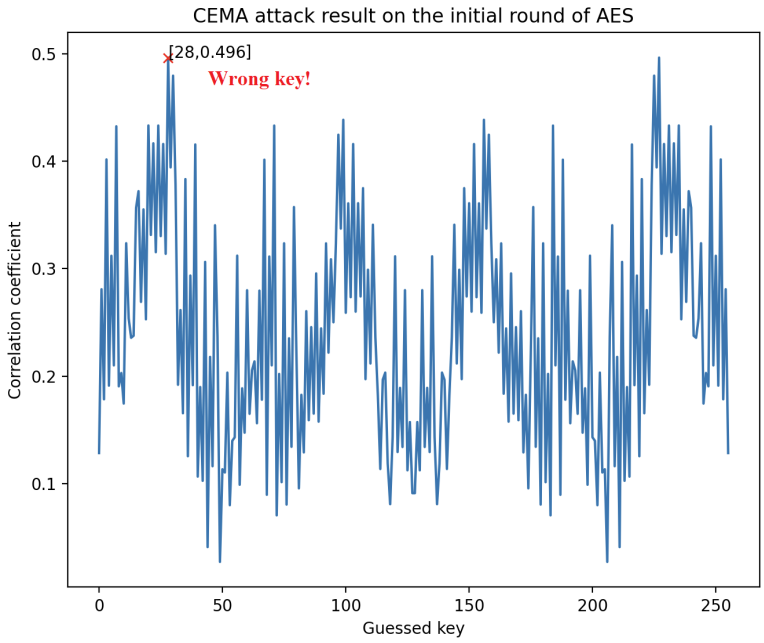


(b)

Figure 5.11: Attack results when added delay is 2**TAP*. (a) The correct key (**DEC31**, the black line) cannot be detected with 70,000 traces, and (b) the wrong key has the maximum correlation coefficient (SNR = 0.941).

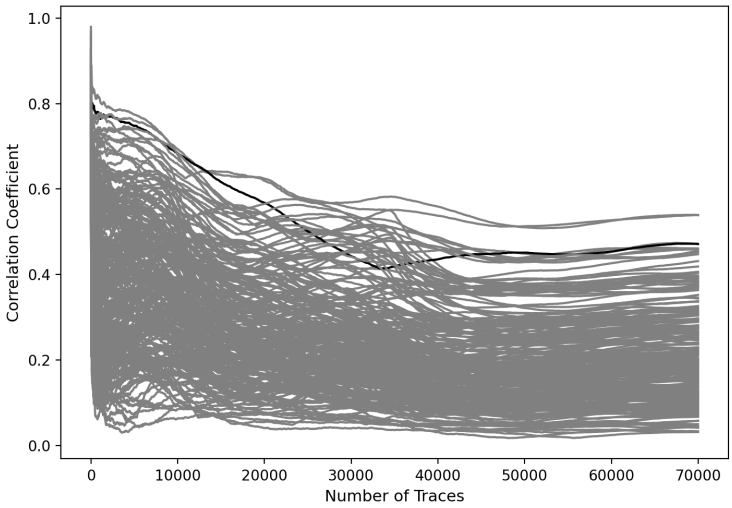


(a)

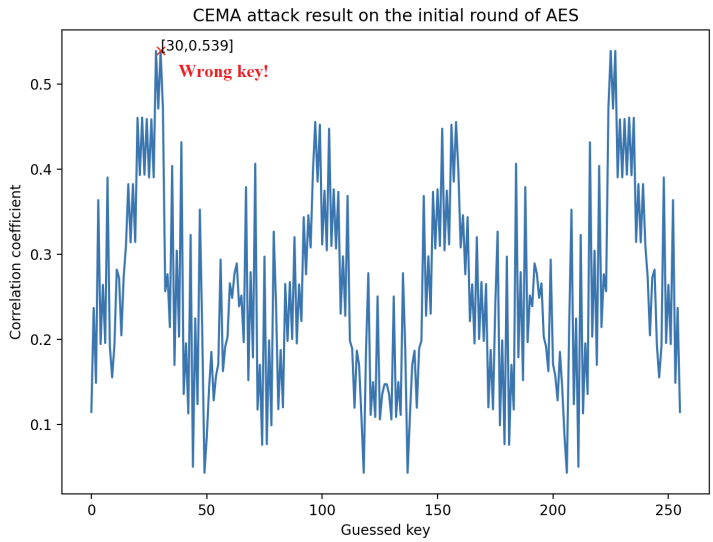


(b)

Figure 5.12: Attack results of the 8-bit bus when added delay is 3**TAP*. (a) The correct key (**DEC31**, the black line) cannot be detected with 70,000 traces, and (b) the wrong key has the maximum correlation coefficient (SNR = 0.818).



(a)



(b)

Figure 5.13: Attack results when added delay is 4**TAP*. (a) The correct key (**DEC31**, the black line) cannot be detected with 70,000 traces, and (b) the wrong key has the maximum correlation coefficient (SNR = 0.908).

5.3.3 Second Protection: Dynamic Delay Insertion with Data Bus Inversion

In this subsection, the experimental results for the attack protection scheme proposed in Section 4.4 are illustrated.

Protection description

For the second protection scheme, the delay is dynamically and non-randomly inserted into the bus lines as determined by the HD of two consecutive pieces of data on the bus where the DBI technique is applied. The Verilog implementation of DBI is shown in Code Listing 5.3, where the HD between *Drg* and the initial data 8'b00 is calculated. If HD is greater than 4, *Drg* is reversed otherwise remains the same.

```
assign data_inv = Drg[0] + Drg[1] + Drg[2] + Drg[3] + Drg[4] + Drg
  [5] + Drg[6] + Drg[7];
always@(*) begin
if (data_inv > 4)
  data_out = ~Drg;
else
  data_out = Drg;end
```

Code Listing 5.3: Verilog implementation of DBI technique on an 8-bit bus.

The appealing feature of this delay insertion scheme is that the delayed bus lines are chosen dynamically, which provides greater spatio-temporal variation and, therefore, makes key extraction more difficult. The block diagram and verilog implementation of the Algorithm 2 proposed in Section 4.4.3 are respectively shown in 5.14 and Code Listing 5.4, where *Din* is the data at time t_i, *Dpre* is the data at time t_{i-1}, D_{xor} represents the XOR result of *Din* and *Dpre*, and S1 is the 8-bit control output. Each bit of S1 acts as the control bit $\overline{S_x}$ in the delay circuit shown in Fig. 4.18, which indicates if this data bit line needs to be delayed or not at time t_i. For instance, if S1[0] is '0', Din[0] will be delayed before reaching the bus.

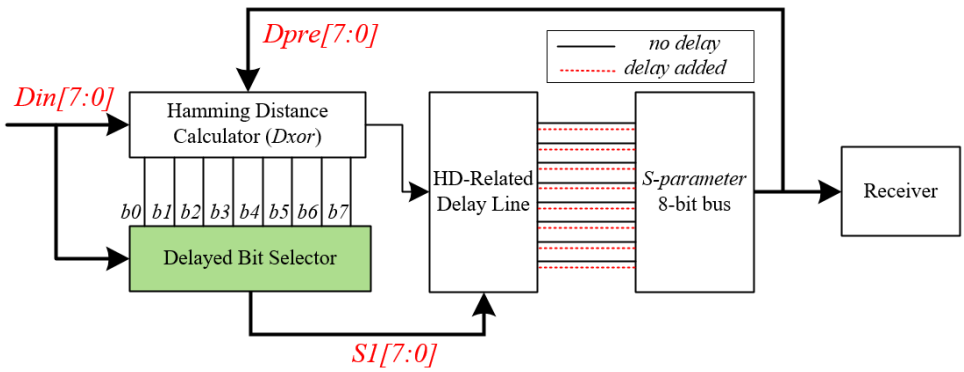


Figure 5.14: Hardware structure of the dynamic delay insertion countermeasure.

```
assign Dxor = Dpre ^ Din;
genvar i;
```

```
generate
   for (i = 1; i < IN_SIZE - 1; i++) begin
assign S1[i] = !((Dxor[i] & ~Dxor[i+1]) || (Dxor[i] & Dxor[i+1] & ~Dxor[i-1]));
   end
endgenerate
assign S1[0] = ~Dxor[0];
assign S1[IN_SIZE-1] = ~Dxor[IN_SIZE-1];
```

Code Listing 5.4: Verilog implementation of the selection of bus lines to be delayed.

Experimental results

For the 8-bit bus protected by the DBI technique, when the control bit is observed by the attacker, the correct key (**DEC31**) can be detected with less than 20,000 traces, as shown in Fig. 5.15. However, if the bus is protected by the DBI technique combined with the dynamic delay insertion technique, even by increasing the number of EM measurements to 100,000 still neither the correct key (**DEC31**) nor the complement of the correct key (**DEC224**) is distinguishable from the wrong keys, as demonstrated in Fig. 5.16.

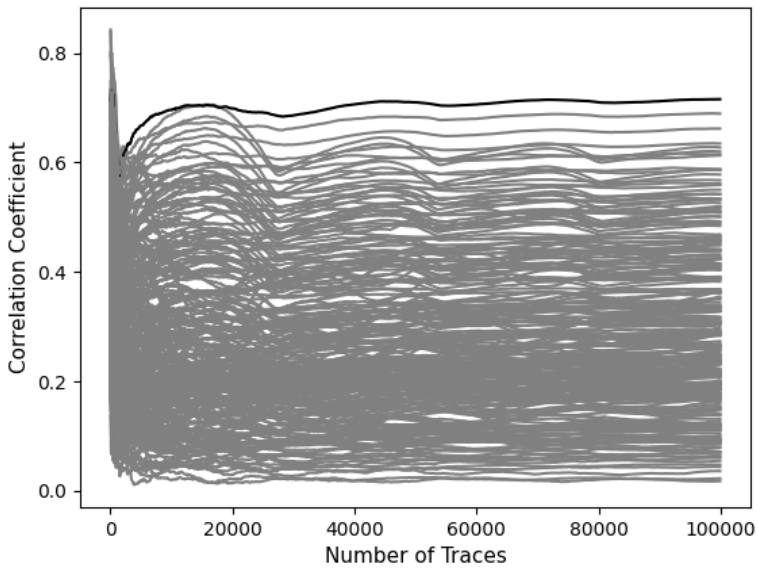


Figure 5.15: Attack result of the bus protected by data bus inversion with the control bit observed by the attacker. The correct key (black line) can be detected in 20,000 traces.

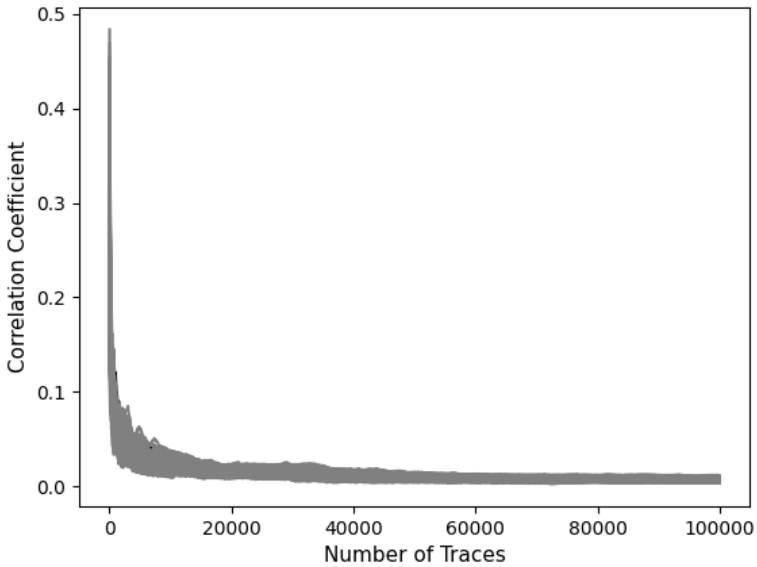


Figure 5.16: Attack result for the bus protected by data bus inversion and dynamic delay insertion. The correct (black line) cannot be detected with 100,000 traces.

The experiments show that the DBI technique alone is not sufficient to protect the circuit, as the secret key can be revealed in less than 20,000 traces if the control bit of DBI is compromised. However, when the DBI technique is combined with the dynamic delay insertion technique, the secret key remains secure even if the control bit of DBI is exposed, as 100,000 traces are not enough to detect it (due to the timing constraint, no more traces can be collected from the experiments). Therefore, the new combined technique improves MTD by at least 5 times and decreases the correlation coefficient by 100 times (from 0.71 to 0.0067). As explained earlier, the delayed lines are selected dynamically in each clock cycle which are determined by the hamming distance between the current data and the previous data transmitted on the bus. The difficulties of reverse-engineering this method are: 1) the delayed lines change every clock cycle and the delayed lines are not observable at the receiver side. Because the added delay is shorter than the transition time of the worst-case (bit line), all the bits are received correctly at the receiver side and synchronised by the register; 2) if the attacker knows that a certain delay scheme has been applied to the bus lines, specific data pattern can be designed to feed to the bus. In this case, a high-cost high-sensitivity EM probe that can distinguish the EM emissions of each single line from other lines is required to identify which bit lines are delayed based on the current data. Moreover, high precision control over the probe and the system (transmitted data can

be modified by the attackers) are also required; and 3) if the attacker has identified the delay insertion pattern through step 2, compensating for the fixed delay is also not easy. One reason is that the delay insertion scheme can inversely benefit from the process variation and ageing, which causes non-fixed delays and may vary across all bit lines, increasing the difficulty for the attacker to fully neutralise the protection scheme. If the delay cannot be accurately compensated, it can help obfuscate the correlation between the data and the emissions. Moreover, if the delay is adjustable, which is left to future works, the randomness in spatial and temporal dimensions further increase the attack complexity.

5.4 Discussion

In this chapter, the initial proof-of-concept experiments on Zedboard are conducted to evaluate the EM attacks and the proposed protection methods are demonstrated. It have been shown from the results that an unprotected 8-bit bus is vulnerable to EM attacks, as the correct key can be recovered with less than 35,000 traces by using the NSD value to locate the optimal attack spot. The *static* delay insertion scheme can offer some protection to the bus if the delay exceeds the lower bound. DBI technique alone is not effective if the attacker can access the control bit, as the key can be revealed with less than 30,000 traces in the experiments. The combination of a *dynamic* delay insertion method combined and the DBI technique can significantly enhance the bus security as it increases MTD by 5x and reduces the correlation coefficient decreases by 100x. The experiments in this chapter only demonstrate the feasibility of the methods proposed in Chapter 3 and Chapter 4. The verification of bus performance improvement is beyond the scope of this dissertation, as it requires high-speed bus links and expensive hardware. Finally, the *dynamic* delay insertion scheme (combined with the DBI technique) is compared with other state-of-the-art countermeasures in Table 2.3 of Chapter 2.

Chapter 6

Conclusion and Future Work

6.1 Contribution Summary

In this dissertation, the concepts, methods, and implementations for efficient EM attacks and protections on interposer-based interconnects in 2.5-D systems have been proposed, for the first time. In this dissertation, state-of-the-art implementations in scenarios where the sensitive data is transmitted through the interconnects in the initial round of AES encryption have been demonstrated to: 1) attack the sensitive keys with 10 $\times$ fewer traces at the attack hotspot, compared to suboptimal probe locations; 2) provide the same level of protection against EM attacks with better bus performance than other RDI techniques and less area overhead; and 3) enhance security protection against EM attacks for the cryptographic circuit without performance degradation and, in specific scenarios, even improve performance.

To achieve these results, a gradient-search algorithm that uses NSD value as the function is first proposed. The optimal probe position can be found within *O(N)* iterations. The proposed search algorithm is scalable to different bus widths.

RDI techniques can protect against SCAs by hiding the link between the leakage and the processed data. However, RDI may worsen the bus performance by inserting delay to the worst-case line. To resolve this issue, a *static* delay insertion technique is proposed that adds delay to two *boundary lines*. These lines do not have the worst case as they always drive a lower coupling capacitance than the maximum across all the bus lines. For the off-chip 8-bit interconnect scenario, when a delay of up to 70 ps is added into the *boundary lines*, the SNR drops below 1 and the total bus latency does not increase. Although the delayed lines are deterministic, it is applicable. One reason is that the process variation and ageing can shift the inserted delay during the fabrication

or over time. Therefore, the delay added to the bus is not necessarily static. The *static* delay insertion technique can actually benefit from these physical characteristics of the chip. The other reason is that there are various side-channel mitigation techniques, it will be hard or costly for an attacker to guess what delay-insertion pattern is used in the circuit. Therefore, even though deterministic, the delay insertion strategy is unknown to the attacker and can offer the off-chip memory bus the required resilience to EM SCAs.

Further, the *static* delay insertion technique can be improved to a *dynamic* delay insertion technique, that improves both immunity against EM attacks and the bus performance. This *dynamic* delay insertion technique determines which bit line should be delayed before a new data reaches the bus. The sensitive information is concealed by adding delay to bit lines based on the HD of two consecutive pieces of encrypted data. This method lowers the worst-case coupling capacitance and, therefore, no bus performance degradation occurs and for specific patterns the performance improves while security is enhanced. In short, this *dynamic* delay insertion method has both a temporal and spatial element. Even if the attacker may guess the temporal component of the method (e.g. a deterministic delay), the spatial component is unlikely to be guessed as which bit lines will be delayed cannot be determine by the attacker.

The proposed solutions in Chapters 3 and 4 are also applicable to various buses in 2-D and 3-D systems, not only to the interposer-based interconnects in 2.5-D systems. The attack method proposed in Chapter 3 is more feasible for scanning the bus surface in 2-D systems than in 2.5-D and 3-D systems, because the on-board buses are more accessible and the bus lines have larger width and length, resulting in more detectable EM radiations for a successful attack. However, if the data signals are routed in the middle layers of the board, a more sensitive probe integrated with a larger amplifier may be needed to capture the EM leakage. For the 3-D systems, multiple chiplets are normally stacked in two ways [156]: **through-silicon via (TSV)** stacking and **contactless** stacking, where the buses are TSVs and wireless links, separately. Probing the EM radiations from through-silicon vias (TSVs) is feasible, but it may require some special techniques and equipment, such as the semiconductor parameter analyser [157]. Furthermore, a coaxial shield TSV (a central signal conductor surrounded by an outside concentric ground shell) can be used to reduce the EM radiations [158], making the probing more challenging. For the contactless stacking systems, such as the conductors based 3-D systems, the EM radiations from the conductors are employed to transfer power or data between different chiplets. These EM radiations are theoretically

probeable but practically constrained by several factors, such as the short distance and low communication power between the inductive pairs. Therefore, probing the EM radiations in 3-D systems depends on various factors, such as the type and structure of the TSVs/conductors, the frequency and power of the signal, and the sensitivity and resolution of the probe. However, probing the EM radiations from 3-D systems does not guarantee a successfully attack on them. To date, to the best of our knowledge, there are no published works of EM attacks on 3-D systems. Nevertheless, EM attacks could pose a threat to 3-D systems in some scenarios, such as military or aerospace applications, where the devices are exposed to hostile EM environments. Therefore, it might be interesting to investigate the vulnerability and resilience of 3-D systems to EM SCAs in the future. The protection methods proposed in Chapter 4 can be directly applied to 2-D and 3-D systems. Due to their low power and area overhead characteristics, these techniques can benefit the compact 3-D systems more by enhancing their security within the power and area constraints.

6.2 Future Work

Future work can benefit from four areas of research and development: 1) enhancement of *dynamic* delay insertion method; 2) application of a DVFS technique to counter EM attacks; 3) removal of the delay added to the buslines; and 4) EM attacks on AES by injecting faults into the serial voltage identification (SVI) bus between the CPU and voltage regulator. The details of each area are as follows:

- Enhancement of *dynamic* delay insertion: as discussed in the last part of Chapter 4, process variation and ageing can cause different delay values or change the delay value over time, thus affecting the effectiveness of delay insertion schemes, such as altering the margin of the effective delay in the *dynamic* delay insertion scheme. If a tunable delay scheme can be applied to the *dynamic* delay insertion method, the effects of process variation and ageing can be reduced, and enhanced security can be provided to protect the bus. This way, the attacker cannot determine the amount of delay that is added to the bus, and which bus line is delayed. The **non-constant delay** can be obtained by using multifinger circuits [159].

- Application of a DVFS technique to counter EM attacks: DVFS is a common technique to reduce the power consumption in integrated circuits. For countering EM attacks, DVFS can be utilised on the bus voltage supply based on the

transition pattern of bus data. For instance, for more data transitions with high HD values, the voltage can lowered; for less data transitions with low HD values, the voltage can be raised. This way, the magnitude of the coupled voltage at the probe terminal can be decorrelated with the sensitive data.

- Removal of the delay added to the buslines: locating the shifted peak value in the amplitude domain dynamically can align the power traces in the time domain [160]. This method can be adopted to preprocess the *static* misaligned EM traces (added delay is fixed) and thus defeat the security protection. For the *dynamic* misalignment of EM traces, a new algorithm can be developed to identify the delay insertion pattern and compensate for the shifted time.

- EM attacks on AES by injecting faults into SVI bus between the CPU and voltage regulator: it has been shown that hardware fault injection can be applied to the SVI bus and manipulate the voltage regulator, which can then extract the sensitive key of the RSA algorithm running in Intel software guard extensions (SGX) [161]. Although Intel stated that opening the case and tampering the internal hardware to compromise SGX was beyond the scope of SGX, EM fault injection requires less access to the silicon die (chip package removal) and can have a precise spatial and timing resolution.

Bibliography

[1] "LANGER EMV-Technik." https://www.langer-emv.de/en/index.

[2] J. Heyszl, D. Merli, B. Heinz, F. D. Santis, and G. Sigl, "Strengths and Limitations of High-Resolution Electromagnetic Field Measurements for Side-Channel Analysis," in *International Conference on Smart Card Research and Advanced Applications*, pp. 248–262, Springer, Nov. 2012.

[3] C. Teegarden, M. Bhargava, and K. Mai, "Side-Channel Attack Resistant ROM-based AES S-Box," in *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 124–129, June 2010.

[4] "*ARM*." https://developer.arm.com/, 1985.

[5] H. Lee, K. Cho, H. Kim, S. Choi, J. Lim, and J. Kim, "Electrical Performance of High Bandwidth Memory (HBM) Interposer Channel in Terabyte/s Bandwidth Graphics Module," in *2015 International 3D Systems Integration Conference (3DIC)*, pp. TS2.2.1–TS2.2.4, Aug. 2015.

[6] J. Knechtel and O. Sinanoglu, "On Mitigation of Side-Channel Attacks in 3D ICs: Decorrelating Thermal Patterns from Power and Activity," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, June 2017.

[7] J. Dofe and Q. Yu, "Exploiting PDN Noise to Thwart Correlation Power Analysis Attacks in 3D ICs," in *2018 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*, pp. 1–6, June 2018.

[8] V. V. Rao, A. Sasan, and I. Savidis, "Analysis of the Security Vulnerabilities of 2.5-D and 3-D Integrated Circuits," in *2022 23rd International Symposium on Quality Electronic Design (ISQED)*, pp. 1–7, Apr. 2022.

[9] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and Electromagnetic Analysis: Improved Model, Consequences and Comparisons," *Integration*, vol. 40, pp. 52–60, Jan. 2007.

[10] D. Lee, D. Jung, I. T. Fang, C.-C. Tsai, and R. A. Popa, "An Off-Chip Attack on Hardware Enclaves via the Memory Bus," in *29th USENIX Security Symposium (USENIX Security 20)*, Aug. 2020.

[11] C. Clavier, J.-S. Coron, and N. Dabbous, "Differential Power Analysis in the Presence of Hardware Countermeasures," in *Cryptographic Hardware and Embedded Systems — CHES 2000*, pp. 252–263, Aug. 2000.

[12] M. Bucci, R. Luzzi, M. Guglielmo, and A. Trifiletti, "A Countermeasure against Differential Power Analysis based on Random Delay Insertion," in *2005 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3547–3550, IEEE, May 2005.

[13] S. Moore, R. Anderson, R. Mullins, G. Taylor, and J. J. Fournier, "Balanced Self-Checking Asynchronous Logic for Smart Card Applications," *Microprocessors and Microsystems*, vol. 27, pp. 421–430, Oct. 2003.

[14] M. Tunstall and O. Benoit, "Efficient Use of Random Delays in Embedded Software," in *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems*, pp. 27–38, May 2007.

[15] D. Das, M. Nath, B. Chatterjee, S. Ghosh, and S. Sen, "STELLAR: A Generic EM Side-Channel Attack Protection through Ground-Up Root-cause Analysis," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 11–20, May 2019.

[16] M. Jiang, E. Maragkoudaki, and V. F. Pavlidis, "Mitigating EM Side-Channel Attacks with Dynamic Delay Insertion and Data Bus Inversion," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1724–1728, IEEE, May 2022.

[17] S. Lu, Z. Zhang, and M. Papaefthymiou, "1.32GHz High-Throughput Charge-Recovery AES Core with Resistance to DPA Attacks," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*, pp. C246–C247, June 2015.

[18] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Advances in Cryptology — CRYPTO '96*, pp. 104–113, Aug. 1996.

[19] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard, "Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices," *IEEE Communications Surveys Tutorials*, vol. 20, pp. 465–488, Dec. 2018.

[20] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," in *Advances in Cryptology — EUROCRYPT '97*, pp. 37–51, May 1997.

[21] C. Aumüller, P. Bier, W. Fischer, P. Hofreiter, and J.-P. Seifert, "Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 260–275, Springer, Aug. 2003.

[22] C. Gerlinsky, "Breaking Code Read Protection on the NXP LPC-Family Microcontrollers," *RECON, Brussels, Belgium*, 2017.

[23] J. Obermaier and S. Tatschner, "Shedding too much Light on a Microcontroller's Firmware Protection," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, (Vancouver, BC), USENIX Association, Aug. 2017.

[24] C. Bozzato, R. Focardi, and F. Palmarini, "Shaping the Glitch: Optimizing Voltage Fault Injection Attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, p. 199–224, Feb. 2019.

[25] T. Fukunaga and J. Takahashi, "Practical Fault Attack on a Cryptographic LSI with ISO/IEC 18033-3 Block Ciphers," in *2009 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pp. 84–92, Sept. 2009.

[26] M. Agoyan, J.-M. Dutertre, D. Naccache, B. Robisson, and A. Tria, "When Clocks Fail: On Critical Paths and Clock Faults," in *International conference on smart card research and advanced applications*, pp. 182–193, Springer, Apr. 2010.

[27] J. Balasch, B. Gierlichs, and I. Verbauwhede, "An In-depth and Black-Box

Characterization of the Effects of Clock Glitches on 8-bit MCUs," in *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 105–114, Sept. 2011.

[28] N. Selmane, S. Guilley, and J.-L. Danger, "Practical Setup Time Violation Attacks on AES," in *2008 Seventh European Dependable Computing Conference*, pp. 91–96, May 2008.

[29] J.-J. Quisquater, "Eddy Current for Magnetic Analysis with Active Sensor," *Proceedings of Esmart, 2002*, pp. 185–194, Sept. 2002.

[30] A. Dehbaoui, A.-P. Mirbaha, N. Moro, J.-M. Dutertre, and A. Tria, "Electromagnetic Glitch on the AES Round Counter," in *Constructive Side-Channel Analysis and Secure Design*, pp. 17–31, Mar. 2013.

[31] S. P. Skorobogatov and R. J. Anderson, "Optical Fault Induction Attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, pp. 2–12, Aug. 2003.

[32] M. Hutter and J.-M. Schmidt, "The Temperature Side Channel and Heating Fault Attacks," in *International Conference on Smart Card Research and Advanced Applications*, pp. 219–235, Springer, Nov. 2014.

[33] R. J. Masti, D. Rai, A. Ranganathan, C. Müller, L. Thiele, and S. Capkun, "Thermal Covert Channels on Multi-core Platforms," in *24th USENIX Security Symposium (USENIX Security 15)*, pp. 865–880, Aug. 2015.

[34] D. B. Bartolini, P. Miedl, and L. Thiele, "On the Capacity of Thermal Covert Channels in Multicores," in *Proceedings of the Eleventh European Conference on Computer Systems*, EuroSys '16, Apr. 2016.

[35] S. Tian and J. Szefer, "Temporal Thermal Covert Channels in Cloud FPGAs," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '19, p. 298–303, Feb. 2019.

[36] D. Karakoyunlu and B. Sunar, "Differential Template Attacks on PUF enabled Cryptographic Devices," in *2010 IEEE International Workshop on Information Forensics and Security*, pp. 1–6, Dec. 2010.

[37] O. Choudary and M. G. Kuhn, "Template Attacks on Different Devices," in *Constructive Side-Channel Analysis and Secure Design*, pp. 179–198, Apr. 2014.

[38] S. Chari, J. R. Rao, and P. Rohatgi, "Template Attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 13–28, Springer, Aug. 2002.

[39] S. Skorobogatov, "The Bumpy Road towards iPhone 5c NAND Mirroring," *arXiv preprint arXiv:1609.04327*, Sept. 2016.

[40] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," tech. rep., Naval Research Lab Washington DC, 2004.

[41] G. He, M. Yang, X. Gu, J. Luo, and Y. Ma, "A Novel Active Website Fingerprinting Attack against Tor Anonymous System," in *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 112–117, May 2014.

[42] J. Gu, J. Wang, Z. Yu, and K. Shen, "Traffic-based Side-Channel Attack in Video Streaming," *IEEE/ACM Transactions on Networking*, vol. 27, pp. 972–985, Apr. 2019.

[43] P. Belgarric, P.-A. Fouque, G. Macario-Rat, and M. Tibouchi, "Side-Channel Analysis of Weierstrass and Koblitz Curve ECDSA on Android Smartphones," in *Topics in Cryptology - CT-RSA 2016*, pp. 236–252, Mar. 2016.

[44] T. Kubota, K. Yoshida, M. Shiozaki, and T. Fujino, "Deep Learning Side-Channel Attack Against Hardware Implementations of AES," in *2019 22nd Euromicro Conference on Digital System Design (DSD)*, pp. 261–268, Nov. 2019.

[45] S. Mangard, "A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion," in *Information Security and Cryptology — ICISC 2002*, pp. 343–358, Nov. 2003.

[46] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology — CRYPTO' 99*, pp. 388–397, Aug. 1999.

[47] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, pp. 16–29, Aug. 2004.

[48] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual Information Analysis: A Generic Side-Channel Distinguisher," in *Cryptographic Hardware and Embedded Systems – CHES 2008*, pp. 426–442, Aug. 2008.

[49] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, pp. 251–261, May 2001.

[50] Y. Hori, T. Katashita, A. Sasaki, and A. Satoh, "Electromagnetic Side-Channel Attack against 28-nm FPGA Device," *Pre-proceedings of WISA*, p. 84, Aug. 2012.

[51] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon, "Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, p. 163–177, Oct. 2018.

[52] R. Hund, C. Willems, and T. Holz, "Practical Timing Side Channel Attacks against Kernel Space ASLR," in *2013 IEEE Symposium on Security and Privacy*, pp. 191–205, May 2013.

[53] Z. H. Jiang, Y. Fei, and D. Kaeli, "A Complete Key Recovery Timing Attack on a GPU," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 394–405, Mar. 2016.

[54] B. B. Brumley and N. Tuveri, "Remote Timing Attacks are still Practical," in *Computer Security – ESORICS 2011*, pp. 355–371, Sept. 2011.

[55] M. Guri, B. Zadov, and Y. Elovici, "LED-it-GO: Leaking (A Lot of) Data from Air-Gapped Computers via the (Small) Hard Drive LED," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 161–184, July 2017.

[56] J. Ferrigno and M. Hlaváč, "When AES Blinks: Introducing Optical Side Channel," *IET Information Security*, vol. 2, pp. 94–98, Sept. 2008.

[57] R. Spreitzer, "PIN Skimming: Exploiting the Ambient-Light Sensor in Mobile Devices," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, pp. 51–62, Nov. 2014.

[58] J. Brouchier, T. Kean, C. Marsh, and D. Naccache, "Temperature Attacks," *IEEE Security Privacy*, pp. 79–82, Apr. 2009.

[59] S. Young, "Researchers Recover Typed Text using Audio Recording of Keystrokes," *UC Berkeley NewsCenter*, 2005.

[60] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard Acoustic Emanations Revisited," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, pp. 1–26, Nov. 2009.

[61] O. Acıiçmez and Ç. K. Koç, "Trace-Driven Cache Attacks on AES (short paper)," in *International Conference on Information and Communications Security*, pp. 112–121, Springer, Dec. 2006.

[62] J.-F. Gallais, I. Kizhvatov, and M. Tunstall, "Improved Trace-Driven Cache-Collision Attacks against Embedded AES Implementations," in *Information Security Applications*, pp. 243–257, Aug. 2011.

[63] D. J. Bernstein, "Cache-Timing Attacks on AES," 2005.

[64] D. A. Osvik, A. Shamir, and E. Tromer, "Cache Attacks and Countermeasures: The Case of AES," in *Topics in Cryptology – CT-RSA 2006*, pp. 1–20, Feb. 2006.

[65] F. Liu, Y. Yarom, Q. Ge, G. Heiser, and R. B. Lee, "Last-Level Cache Side-Channel Attacks are Practical," in *2015 IEEE Symposium on Security and Privacy*, pp. 605–622, May 2015.

[66] G. Irazoqui, T. Eisenbarth, and B. Sunar, "S$A: A Shared Cache Attack That Works across Cores and Defies VM Sandboxing – and Its Application to AES," in *2015 IEEE Symposium on Security and Privacy*, pp. 591–604, May 2015.

[67] M. Kayaalp, N. Abu-Ghazaleh, D. Ponomarev, and A. Jaleel, "A High-Resolution Side-Channel Attack on Last-Level Cache," in *Proceedings of the 53rd Annual Design Automation Conference*, DAC '16, June 2016.

[68] D. Gullasch, E. Bangerter, and S. Krenn, "Cache Games – Bringing Access-based Cache Attacks on AES to Practice," in *2011 IEEE Symposium on Security and Privacy*, pp. 490–505, May 2011.

[69] G. Irazoqui, M. S. Inci, T. Eisenbarth, and B. Sunar, "Wait a Minute! A fast, Cross-VM Attack on AES," in *Research in Attacks, Intrusions and Defenses*, pp. 299–319, Sept. 2014.

[70] Y. Yarom and K. Falkner, "FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack," in *23rd USENIX Security Symposium (USENIX Security 14)*, pp. 719–732, Aug. 2014.

[71] S. Jana and V. Shmatikov, "Memento: Learning Secrets from Process Footprints," in *2012 IEEE Symposium on Security and Privacy*, pp. 143–157, May 2012.

[72] D. Gruss, R. Spreitzer, and S. Mangard, "Cache Template Attacks: Automating Attacks on Inclusive Last-Level Caches," in *24th USENIX Security Symposium (USENIX Security 15)*, pp. 897–912, USENIX Association, Aug. 2015.

[73] M. Schwarz, F. Lackner, and D. Gruss, "JavaScript Template Attacks: Automatically Inferring Host Information for Targeted Exploits," in *NDSS*, Feb. 2019.

[74] Q. Yang, P. Gasti, K. Balagani, Y. Li, and G. Zhou, "USB Side-Channel Attack on Tor," *Computer Networks*, vol. 141, pp. 57–66, Aug. 2018.

[75] M. Enev, S. Gupta, T. Kohno, and S. N. Patel, "Televisions, Video Privacy, and Powerline Electromagnetic Interference," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, pp. 537–550, Oct. 2011.

[76] M. G. Kuhn, "Electromagnetic Eavesdropping Risks of Flat-Panel Displays," in *International Workshop on Privacy Enhancing Technologies*, pp. 88–107, Springer, May 2004.

[77] M. Vuagnoux and S. Pasini, "Compromising Electromagnetic Emanations of Wired and Wireless Keyboards," in *USENIX security symposium*, vol. 1, Aug. 2009.

[78] J. Seibert, H. Okhravi, and E. Söderström, "Information Leaks Without Memory Disclosures: Remote Side Channel Attacks on Diversified Code," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 54–65, Nov. 2014.

[79] P. Cheng, I. E. Bagci, U. Roedig, and J. Yan, "SonarSnoop: Active Acoustic Side-Channel Attacks," *International Journal of Information Security*, vol. 19, pp. 213–228, Apr. 2020.

[80] M. Jiang and V. F. Pavlidis, "A Probe Placement Method for Efficient Electromagnetic Attacks," in *SMACD / PRIME 2021; International Conference on SMACD and 16th Conference on PRIME*, pp. 1–4, July 2021.

[81] Y.-i. Hayashi, N. Homma, T. Sugawara, T. Mizuki, T. Aoki, and H. Sone, "Non-Invasive EMI-based Fault Injection Attack against Cryptographic Modules," in *2011 IEEE International Symposium on Electromagnetic Compatibility*, pp. 763–767, Aug. 2011.

[82] A. Moradi and O. Mischke, "On the Simplicity of Converting Leakages from Multivariate to Univariate," in *Cryptographic Hardware and Embedded Systems - CHES 2013*, pp. 1–20, Aug. 2013.

[83] G. Fumaroli, A. Martinelli, E. Prouff, and M. Rivain, "Affine Masking against Higher-Order Side Channel Analysis," in *Selected Areas in Cryptography*, pp. 262–280, Aug. 2011.

[84] B. Gierlichs, L. Batina, B. Preneel, and I. Verbauwhede, "Revisiting Higher-Order DPA Attacks: Multivariate Mutual Information Analysis," in *Topics in Cryptology - CT-RSA 2010*, pp. 221–234, Mar. 2010.

[85] I. Levi, A. Fish, and O. Keren, "CPA Secured Data-Dependent Delay-Assignment Methodology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, pp. 608–620, Aug. 2017.

[86] B. J. Gilbert Goodwill, J. Jaffe, P. Rohatgi, *et al.*, "A Testing Methodology for Side-Channel Resistance Validation," in *NIST non-invasive attack testing workshop*, vol. 7, pp. 115–136, Sept. 2011.

[87] F.-X. Standaert, "How (Not) to Use Welch's T-Test in Side-Channel Security Evaluations," in *Smart Card Research and Advanced Applications*, pp. 65–79, Nov. 2019.

[88] F.-X. Standaert, T. G. Malkin, and M. Yung, "A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks," in *Advances in Cryptology-EUROCRYPT 2009: 28th Annual International Conference on the Theory and*

Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings 28, pp. 443–461, Springer, Apr. 2009.

[89] A. Fell, H. T. Pham, and S.-K. Lam, "TAD: Time Side-Channel Attack Defense of Obfuscated Source Code," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pp. 58–63, Jan. 2019.

[90] Z. Chen, A. Sinha, and P. Schaumont, "Using Virtual Secure Circuit to Protect Embedded Software from Side-Channel Attacks," *IEEE Transactions on Computers*, vol. 62, pp. 124–136, Dec. 2011.

[91] Y. Wang and Y. Ha, "A Performance and Area Efficient ASIP for Higher-Order DPA-Resistant AES," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, pp. 190–202, Apr. 2014.

[92] M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni, "Fresh Rekeying: Security against Side-Channel and Fault Attacks for Low-Cost Devices," in *Progress in Cryptology – AFRICACRYPT 2010*, pp. 279–296, May 2010.

[93] K. Tiri, M. Akmal, and I. Verbauwhede, "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards," in *Proceedings of the 28th European Solid-State Circuits Conference*, pp. 403–406, Sept. 2002.

[94] B. Hettwer, K. Das, S. Leger, S. Gehrer, and T. Güneysu, "Lightweight Side-Channel Protection using Dynamic Clock Randomization," in *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*, pp. 200–207, Aug. 2020.

[95] X. Wang, W. Yueh, D. B. Roy, S. Narasimhan, Y. Zheng, S. Mukhopadhyay, D. Mukhopadhyay, and S. Bhunia, "Role of Power Grid in Side Channel Attack and Power-Grid-Aware Secure Design," in *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, May 2013.

[96] F. Liu, Q. Ge, Y. Yarom, F. Mckeen, C. Rozas, G. Heiser, and R. B. Lee, "CATalyst: Defeating Last-Level Cache Side Channel Attacks in Cloud Computing," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 406–418, Mar. 2016.

[97] P. Gu, D. Stow, R. Barnes, E. Kursun, and Y. Xie, "Thermal-Aware 3D Design for Side-Channel Information Leakage," in *2016 IEEE 34th International Conference on Computer Design (ICCD)*, pp. 520–527, Oct. 2016.

[98] K. Mai, "Side Channel Attacks and Countermeasures," in *Introduction to Hardware Security and Trust*, pp. 175–194, Springer, Aug. 2011.

[99] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, pp. 246–251, IEEE, Feb. 2004.

[100] H. Wang, H. Sayadi, T. Mohsenin, L. Zhao, A. Sasan, S. Rafatirad, and H. Homayoun, "Mitigating Cache-based Side-Channel Attacks through Randomization: A Comprehensive System and Architecture Level Analysis," in *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1414–1419, Mar. 2020.

[101] N. Mentens, "Hiding Side-Channel Leakage through Hardware Randomization: A Comprehensive Overview," in *2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, pp. 269–272, July 2017.

[102] W. Yu, O. A. Uzun, and S. Köse, "Leveraging On-Chip Voltage Regulators as a Countermeasure against Side-Channel Attacks," in *Proceedings of the 52nd Annual Design Automation Conference*, DAC '15, June 2015.

[103] M. Van der Maas and S. W. Moore, "Protecting Enclaves from Intra-Core Side-Channel Attacks through Physical Isolation," in *Proceedings of the 2nd Workshop on Cyber-Security Arms Race*, pp. 1–12, Nov. 2020.

[104] A. Shamir, "Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies," in *Cryptographic Hardware and Embedded Systems — CHES 2000*, pp. 71–77, Aug. 2000.

[105] L. Lin, D. Zhu, J. Wen, H. Chen, Y. Lu, N. Chang, C. Chow, H. Shrivastav, C.-W. Chen, K. Monta, and M. Nagata, "Multiphysics Simulation of EM Side-Channels from Silicon Backside with ML-based Auto-POI Identification," in *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 270–280, Dec. 2021.

[106] J.-J. Quisquater and D. Samyde, "ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards," in *Smart Card Programming and Security*, pp. 200–210, Sept. 2001.

[107] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, "Semi-Invasive EM Attack on FPGA RO PUFs and Countermeasures," in *Proceedings of the Workshop on Embedded Systems Security*, pp. 1–9, Oct. 2011.

[108] WIKI, "Electromagnetically short antennas." `https://en.wikipedia.org/wiki/Near_and_far_field`.

[109] A. E. Yilmaz, J.-M. Jin, and E. Michielssen, "Time Domain Adaptive Integral Method for Surface Integral Equations," *IEEE Transactions on Antennas and Propagation*, vol. 52, pp. 2692–2708, Oct. 2004.

[110] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, "A Survey of Electromagnetic Side-Channel Attacks and Discussion on Their Case-Progressing Potential for Digital Forensics," *Digital Investigation*, vol. 29, pp. 43–54, June 2019.

[111] D. Merli, J. Heyszl, B. Heinz, D. Schuster, F. Stumpf, and G. Sigl, "Localized Electromagnetic Analysis of RO PUFs," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 19–24, June 2013.

[112] L. Sauvage, S. Guilley, and Y. Mathieu, "Electromagnetic Radiations of FPGAs: High Spatial Resolution Cartography and Attack on a Cryptographic Module," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 2, pp. 1–24, Mar. 2009.

[113] T. Hummel, "Exploring Effects of Electromagnetic Fault Injection on a 32-bit High Speed Embedded Device Microprocessor," Master's thesis, University of Twente, 2014.

[114] A. Dehbaoui, J.-M. Dutertre, B. Robisson, and A. Tria, "Electromagnetic Transient Faults Injection on a Hardware and a Software Implementations of AES," in *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 7–15, Sept. 2012.

[115] A. Dehbaoui, J.-M. Dutertre, B. Robisson, P. Orsatelli, P. Maurine, and A. Tria, "Injection of Transient Faults using Electromagnetic Pulses Practical Results on

a Cryptographic System," 2012. Journal of Cryptology ePrint Archive: Report 2012/123.

[116] A. Menu, S. Bhasin, J.-M. Dutertre, J.-B. Rigaud, and J.-L. Danger, "Precise Spatio-Temporal Electromagnetic Fault Injections on Data Transfers," in *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pp. 1–8, Aug. 2019.

[117] N. Moro, A. Dehbaoui, K. Heydemann, B. Robisson, and E. Encrenaz, "Electromagnetic Fault Injection: Towards a Fault Model on a 32-bit Microcontroller," in *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 77–88, Aug. 2013.

[118] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer, "ECDH Key-Extraction via Low-Bandwidth Electromagnetic Attacks on PCs," in *Topics in Cryptology - CT-RSA 2016*, pp. 219–235, Feb. 2016.

[119] V. Gustov and A. Levina, "Electromagnetic Fields as a Sign of Side-Channel Attacks in GSM Module," in *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, Apr. 2021.

[120] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM Side—Channel(s)," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, pp. 29–45, Aug. 2003.

[121] A. W. Khan, T. Wanchoo, G. Mumcu, and S. Köse, "Implications of Distributed On-Chip Power Delivery on EM Side-Channel Attacks," in *2017 IEEE International Conference on Computer Design (ICCD)*, pp. 329–336, Nov. 2017.

[122] M. Wang, V. V. Iyer, S. Xie, G. Li, S. K. Mathew, R. Kumar, M. Orshansky, A. E. Yilmaz, and J. P. Kulkarni, "Physical Design Strategies for Mitigating Fine-Grained Electromagnetic Side-Channel Attacks," in *2021 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–2, Apr. 2021.

[123] A. Kumar, C. Scarborough, A. Yilmaz, and M. Orshansky, "Efficient Simulation of EM Side-Channel Attack Resilience," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 123–130, Nov. 2017.

[124] C. Wang, Y. Cai, H. Wang, and Q. Zhou, "Electromagnetic Equalizer: An Active Countermeasure Against EM Side-channel Attack," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, Nov. 2018.

[125] T. Katashita, A. Satoh, K. Kikuchi, H. Nakagawa, and M. Aoyagi, "Evaluation of DPA Characteristics of Sasebo for Board Level Simulations," *proceedings of COSADE*, 2010.

[126] A. U. Danis and B. Ors, "Differential Power Analysis Attack considering Decoupling Capacitance Effect," in *2009 European Conference on Circuit Theory and Design*, pp. 359–362, Aug. 2009.

[127] H. Ma, J. He, Y. Liu, L. Liu, Y. Zhao, and Y. Jin, "Security-Driven Placement and Routing Tools for Electromagnetic Side-Channel Protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, pp. 1077–1089, Sept. 2021.

[128] D. Das, J. Danial, A. Golder, N. Modak, S. Maity, B. Chatterjee, D.-H. Seo, M. Chang, A. L. Varna, H. K. Krishnamurthy, S. Mathew, S. Ghosh, A. Raychowdhury, and S. Sen, "EM and Power SCA-Resilient AES-256 Through $> 350\times$ Current-Domain Signature Attenuation and Local Lower Metal Routing," *IEEE Journal of Solid-State Circuits*, vol. 56, pp. 136–150, Nov. 2021.

[129] H. Ma, J. He, M. Panoff, Y. Jin, and Y. Zhao, "Automatic On-Chip Clock Network Optimization for Electromagnetic Side-Channel Protection," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, pp. 371–382, May 2021.

[130] C. H. Gebotys, "A Table Masking Countermeasure for Low-Energy Secure Embedded Systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 740–753, July 2006.

[131] "*LDO*." `https://en.wikipedia.org/wiki/Low-dropout_regulator`.

[132] A. Singh, M. Kar, V. C. K. Chekuri, S. K. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "Enhanced Power and Electromagnetic SCA Resistance of Encryption Engines via a Security-Aware Integrated All-Digital LDO," *IEEE Journal of Solid-State Circuits*, vol. 55, pp. 478–493, Oct. 2019.

[133] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," 1999.

[134] D. Real, F. Valette, and M. Drissi, "Enhancing Correlation Electromagnetic Attack using Planar Near-Field Cartography," in *2009 Design, Automation & Test in Europe Conference Exhibition*, pp. 628–633, Apr. 2009.

[135] "*Correlation Power Analysis*." `https://wiki.newae.com/Correlation_Power_Analysis`.

[136] O.-X. Standaert, E. Peeters, G. Rouvroy, and J.-J. Quisquater, "An Overview of Power Analysis Attacks Against Field Programmable Gate Arrays," *Proceedings of the IEEE*, vol. 94, pp. 383–394, Jan. 2006.

[137] V. V. Iyer and A. E. Yilmaz, "An Adaptive Acquisition Approach to Localize Electromagnetic Information Leakage from Cryptographic Modules," in *2019 IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS)*, pp. 1–6, Mar. 2019.

[138] S. Mangard, "Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness," in *Cryptographers' Track at the RSA Conference*, pp. 222–235, Springer, Feb. 2004.

[139] "*HFSS*." `https://www.ansys.com/products/electronics/ansys-hfss`.

[140] R. Novak, "Side-Channel Attack on Substitution Blocks," in *International Conference on Applied Cryptography and Network Security*, pp. 307–318, Springer, Oct. 2003.

[141] I. Levi, O. Keren, and A. Fish, "Data-Dependent Delays as a Barrier Against Power Attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp. 2069–2078, July 2015.

[142] H. Bakoglu and J. D. Meindl, "Optimal Interconnection Circuits for VLSI," *IEEE Transactions on Electron Devices*, vol. 32, pp. 903–909, May 1985.

[143] K. Hirose and H. Yasuura, "A Bus Delay Reduction Technique Considering Crosstalk," in *Proceedings of the conference on Design, automation and test in Europe*, pp. 441–445, Jan. 2000.

[144] V. F. Pavlidis, I. Savidis, and E. G. Friedman, *Three-Dimensional Integrated Circuit Design*. Newnes, 2017.

[145] M. Fujino and V. G. Moshnyaga, "An Efficient Hamming Distance Comparator for Low-power Applications," in *9th International Conference on Electronics, Circuits and Systems*, vol. 2, pp. 641–644, IEEE, Sept. 2002.

[146] M. Alioto, S. Bongiovanni, M. Djukanovic, G. Scotti, and A. Trifiletti, "Effectiveness of Leakage Power Analysis Attacks on DPA-Resistant Logic Styles Under Process Variations," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, pp. 429–442, Aug. 2013.

[147] N. Chawla, A. Singh, N. M. Rahman, M. Kar, and S. Mukhopadhyay, "Extracting Side-Channel Leakage from Round Unrolled Implementations of Lightweight Ciphers," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 31–40, IEEE, May 2019.

[148] M. R. Stan and W. P. Burleson, "Bus-invert Coding for Low-power I/O," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 3, pp. 49–58, Mar. 1995.

[149] M. A. Vosoughi, L. Wang, and S. Köse, "Bus-Invert Coding as a Low-Power Countermeasure Against Correlation Power Analysis Attack," in *2019 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*, pp. 1–5, IEEE, June 2019.

[150] E. Maragkoudaki and V. F. Pavlidis, "Energy-Efficient Time-based Adaptive Encoding for Off-Chip Communication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, pp. 2551–2562, Aug. 2020.

[151] M. Xuan, "United Microelectronics Corporation UMC." `http://umc.com/`, 1980.

[152] S. Lu, Z. Zhang, and M. Papaefthymiou, "1.32GHz High-Throughput Charge-Recovery AES Core with Resistance to DPA Attacks," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*, pp. C246–C247, June 2015.

[153] "*XILINX*." `https://www.xilinx.com/`, 1984.

[154] "*Aoki Laboratory*." `http://www.aoki.ecei.tohoku.ac.jp/crypto/`, 2007.

[155] Q. Tian and S. A. Huss, "A General Approach to Power Trace Alignment for the Assessment of Side-Channel Resistance of Hardened Cryptosystems," in

2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 465–470, July 2012.

[156] M. Jiang, I. A. Papistas, and V. F. Pavlidis, "Cost Modeling and Analysis of TSV and Contactless 3D-ICs," in *Proceedings of Great Lakes Symposium on VLSI*, pp. 519–524, Sept. 2020.

[157] Q. Zeng, J. Chen, and Y. Jin, "Effect of Radiation on Reliability of Through-Silicon via for 3-D Packaging Systems," *IEEE Transactions on Device and Materials Reliability*, vol. 17, pp. 708–712, Dec. 2017.

[158] C.-H. Cheng and T.-L. Wu, "An Ultracompact TSV-Based Common-Mode Filter (TSV-CMF) in Three-Dimensional Integrated Circuits (3-D ICs)," *IEEE Transactions on Electromagnetic Compatibility*, vol. 58, pp. 1128–1135, Apr. 2016.

[159] A. F. Tong, W. M. Lim, K. S. Yeo, C. B. Sia, and W. C. Zhou, "A Scalable RFC-MOS Noise Model," *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, pp. 1009–1019, Apr. 2009.

[160] Q. Tian and S. A. Huss, "A General Approach to Power Trace Alignment for the Assessment of Side-Channel Resistance of Hardened Cryptosystems," in *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 465–470, July 2012.

[161] Z. Chen, G. Vasilakis, K. Murdock, E. Dean, D. Oswald, and F. D. Garcia, "VoltPillager: Hardware-based Fault Injection Attacks against Intel SGX Enclaves using the SVID Voltage Scaling Interface," in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 699–716, USENIX Association, Aug. 2021.