TECHNISCHE
UNIVERSITÄT
DARMSTADT

Novel Power Trace Processing Methods for Side-Channel Analysis of Cryptosystems

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)

von
MEng. Qizhi Tian
geboren in Shanxi, China

Referenten der Arbeit:	Prof. Dr.-Ing. Sorin A. Huss
	Prof. Dr. Ray C.C. Cheung
Tag der Einreichung:	19. April 2013
Tag der mündlichen Prüfung:	23. Mai 2013

Darmstadt 2013
D 17

Erklärung zur Dissertation

Hiermit versichere ich, die vorliegende Dissertation ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 19. April 2013 Qizhi Tian

天道酬勤

Contents

Zusammenfassung

1999 stellte Kocher ein Verfahren vor, um den geheimen Schlüssel eines Kryptosystems zu enthüllen, indem von einer Smartcard ausgehende Seitenkanalinformationen ausgenutzt wurden. Mittlerweile ist die Bedeutung dieses Forschungsfeldes stark angewachsen. Das Motiv hinter diesem Forschungsfeld ist die Bedeutung der Informationssicherheit für eine moderne Gesellschaft gepaart mit einem schier unvorstellbar schnell ablaufenden technologischen Fortschritt.

In dieser Arbeit wird zuerst die von uns vorgestellte Angriffsmethode der Power-Amount-Analyse verallgemeinert und anschließend in eine Methodik zur Power-Analyse überführt, die auf dem in der Telekommunikationstechnik häufig verwendeten weißen Gauß-Rausch-Kanal basiert. Diese Methodik vermittelt zwei wesentliche Konzepte. Zum einen stellt sie eine Möglichkeit bereit, aufgezeichnete Power-Traces zu bearbeiten, um Informationslecks effizienter zu extrahieren und zu separieren, zum anderen können verschiedene Unterscheidungsmerkmale statt nur des Korrelationskoeffizienten angewandt werden.

Um die Angriffsmöglichkeiten zu erweitern, wird danach eine Kleinste-Quadrate-Schätzung auf Basis der Traces vom Leakage-Modell vorgestellt. Basierend auf einem solchen Modell werden die Power-Analysis-Mutationen I und II vorgeschlagen, um eine bessere Angriffsleistung zu erzielen. Ferner wird ein Angriffs-Framework bereitgestellt, das weitere Möglichkeiten zur Schlüsselenthüllung in Kryptosystemen zur Verfügung stellt.

Weiterhin wird eine Reihe von Trace-Vorverarbeitungsverfahren eingeführt, die Fehlausrichtungen in erfassten Power-Traces eliminiert, welche in durch Random-Clock-Verfahren geschützten Kryptosystemen erzeugt wurden. Die Beseitigung der Fehlausrichtung geschieht über horizontales Verschieben und vertikalen Abgleich. Hierzu werden zwei Trace-Vorverarbeitungs-Frameworks bereitgestellt, die sich auf die Vorverarbeitung der falsch ausgerichteten Traces bzw. auf Angriffe fokussieren. Gemäß der jeweiligen Angriffsvoraussetzungen und Implementierungen kann man eine geeignete Trace-Vorverarbeitung und Angriffsmethode für tatsächliche Angriffe selektiv auswählen, um eine effektive Angriffsleistung zu erzielen.

Zuletzt werden alle eingeführten Angriffs- und Vorverarbeitungsmethoden erfolgreich verifiziert und mittels verschiedener kryptographischer Implementierungen, die mit unterschiedlichen Clock-Typen und -Frequenzen laufen, evaluiert. Dies bildet eine brauchbare Grundlage für die Beurteilung der Sicherheit eines Systems, um auch für reale Einsatz-

zwecke sichere Kryptosysteme und Architekturen zu schaffen.

Abstract

In 1999, scientist Kocher proposed a way to reveal the secret key of cryptosystems by exploiting the leaked side channel information from a smart card. Since then, such a research field becomes more and more important. The motive for doing that arises from the interests on the one hand, and the strategic vision behind the information security in modern society accompanied with unimaginable high speed technology development on the other.

In this work, our new proposed attack method, i.e., power amount analysis, is generalized and abstracted firstly, which leads to power amount analysis methodology based on the mostly utilized additive white Gaussian noise channel in the telecommunication field. This methodology conveys two important conceptions. On the one hand, it proposes a way to process the captured power traces to extract and purify the information leakage more efficiently, meanwhile, reduces the dimensionality for the analyzed data resulting to simple calculation in real attacks; on the other hand, various distinguishers may be executed for this attack rather than the calculation of the correlation coefficient.

Second, in order to improve the attack methods, a least squares estimation based trace form leakage model is proposed. Based on such a model, power amount analysis mutation I and II are suggested for perusing better attack performance. Subsequently, an attack framework is given, which provides more possibilities to retrieve keys from cryptosystems.

Third, a series of trace pre-processing methods are proposed to neutralize the misalignment in captured power traces produced from a random clock featured cryptosystem in terms of horizontal alignment and vertical matching. Thereafter, two trace pre-processing frameworks are given concentrating on the misaligned and originally aligned power trace pre-processing and attacks, respectively. According to the different attack requirements and implementations, one can choose appropriate trace pre-processing and attack methods selectively in real attacks to achieve a good attack performance.

Finally, all the proposed attack and trace pre-processing methods and frameworks are successfully verified and evaluated by exploiting different cryptographic implementations running with the different clock types and frequencies, which may be good tools to evaluate the system security for yielding safe cryptosystems and architectures in reality.

Acknowledgements

First and foremost, I would like to show my heartfelt thanks to my supervisor Prof. Dr. -Ing. Sorin A. Huss for his careful academic guidance, support, encouragement, and help of the daily life during my more than four years working in integrated circuits and systems lab. Meanwhile, many thanks to Prof. Dr. Ray C.C. Cheung as my second thesis reviewer for interesting my research topic and giving useful suggestions.

During that time, I am grateful to the former and current colleagues, Abdulhadi Shoufan, Felix Madlener, Adeel Israr, Lijing Chen, H. Gregor Molter, Michael Zohner, Sunil Malipatlolla, Annelie Heuser, Marc Stöttinger, André Seffrin, Hagen Stübing, Thomas Feller, Tolga Arul, Tom Assmuth, Maria Tiedemann, Alexander Biedermann, Zheng Lu, Attila Jaeger. I have learned a lot from them about the research, culture, and serious working attitude. The further discussion, talking, and playing together with you all will be the happy memories in my life. I thank Xueping Wang, Gavin, Huajian Liu, and Xuebing Zhou for careful proof-reading and suggestions. Meanwhile, many thanks to Alexander Biedermann for the German abstract translation.

The beginning days in a new country are difficult, many thanks to Liangliang Zou, Hua Mo, who help me a lot at that time. Meanwhile, thank you all my Chinese friends for making me a happy life here.

Last but not least, I wish to avail this opportunity to express my gratitude and love to my wife, my beloved parents, uncles, aunts, and grandparents, for their long-time support, encouragement, help, and for everything.

List of Abbreviations

AES	Advanced Encryption Standard
ASIC	Application Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BRAM	Block RAM
CFE	Clock Frequency Effects
CLB	Configurable Logic Block
CPA	Correlation Power Analysis
DCM	Digital Clock Management
DES	Data Encryption Standard
DPA	Differential Power Analysis
DPR	Dynamic Partial Reconfiguration
DTW	Dynamic Time Warping
FPGA	Field Programmable Gate Array
GE	Guessing Entropy
GUI	Graphical User Interface
HA	Horizontal Alignment
HD	Hamming Distance
HDL	Hardware Description Language
HW	Hamming Weight
ILM	Instantaneous Leakage Model
IOB	I/O Block
IP	Intellectual Property

LEF	Left Elastic Feature
$LSTM$	Least Squares estimation based Trace form leakage Model
LUT	Look Up Tables
MIA	Mutual Information Analysis
PAA	Power Amount Analysis
$PAA-I$	Power Amount Analysis mutation I
$PAA-II$	Power Amount Analysis mutation II
$PBHA$	Peak position Based Horizontal Alignment
PLM	Process Leakage Model
REF	Right Elastic Feature
SCA	Side Channel Analysis
SMA	SubMiniature version A
SNR	Signal to Noise Ratio
SPA	Simple Power Analysis
SR	Success Rate
$TBHA$	Template Based Horizontal Alignment
USB	Universal Serial Bus
$VHDL$	Very high speed integrated Hardware Description Language
VM	Vertical Matching

List of Figures

List of Tables

Chapter 1

Introduction

Contents

Cryptography has been developed for thousands years, which basically defines a protocol to secure or hide the information during transferring without being known by the third party. Such a field is now divided into two main opposite branches, i.e., cryptography and cryptanalysis.

Cryptography is widely deployed in our daily life from the very beginning of pure encryption or decryption to identification, digital signature, digital envelope, and so on. Meanwhile, the research in such a field features the knowledge from cross-disciplines combining with the mathematics, computer science, physics, statistics, and electronic engineering, etc. For instance, based on the quantum communication in physics, the quantum cryptography was proposed and developed. In a word, the elemental function of cryptography is to fulfill four goals, i.e., information confidentiality, authentication, message integrity, and non-repudiation. Therefore, a lot of efforts are invested to improve the security level in terms of algorithms, architectures, and to ensure that the being transferred information is difficult to be revealed or recovered when facing with malicious attacks.

The cryptanalysis evolves together with the development of cryptography, which, on the contrary, analyzes the cryptosystem to get the hidden information from communication parties. In other words, different kinds of attacks may be mounted on a running cryptosystem. However, some restricted factors must be taken into consideration, e.g., run time of the attack methods, resource usage, and computational complexity, etc. Therefore, to some degree, these factors can be used as the metrics for evaluating attack algorithms in practice.

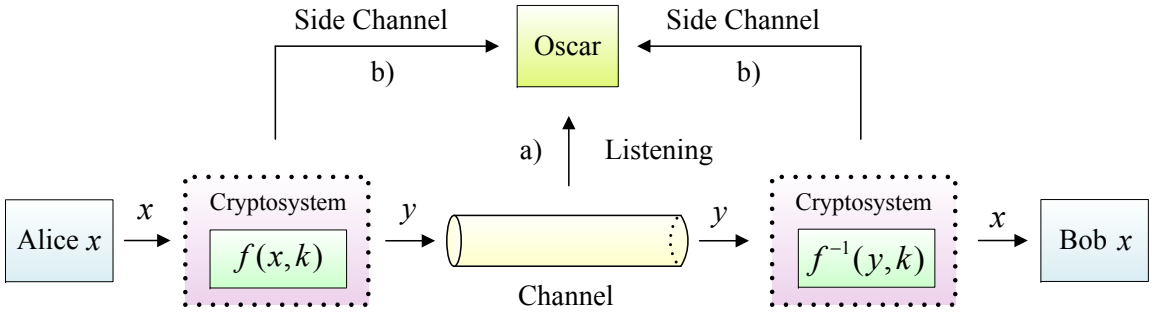

Figure 1.1: An Example in Cryptography

Indeed, both disciplines, i.e., cryptography and cryptanalysis, develop parallel and interact with each other. In order to well protect the running cryptosystem, the mechanism of different attack methods must be well known in advanced. However, in this thesis, we concentrate on the latter, i.e., cryptanalysis on the physical cryptosystem in Side Channel Analysis (SCA) field.

1.1 Cryptography

As written in a lot of literatures, the information communication between Bob and Alice are secured by the cryptographic algorithms running in a cryptosystem and transferred through communication channel, as illustrates in Fig. 1.1. However, there is no absolutely secure channel. A third party Oscar always wants to listen and intercept the information transferring between Bob and Alice. This encryption process may be abstracted as a function $f(\cdot,\cdot)$, where the input x and the key value k are mapped resulting in a encrypted message y, as $y = f(x,k)$. Accordingly, the decryption can be represented by the reversed function $x = f^{-1}(y,k)$. The mentioned process is categorized as symmetric and asymmetric key encryptions. The former features the advantages in terms of opening algorithms, lower calculation complexity, shorter run time, and higher encryption efficiency. Consequently, they are mainly used in the place, where the encryption takes place very often. The typical algorithms are DES and AES, etc., as shown in [ST99], [ST01], and [DR02]. However, the main drawback for the symmetric-key encryption is that the two communication parties must share secret keys. Every time one uses it, a new different key must be generated. After a long time using, the number of keys one must remember increases considerably leading to problems for the key management. Furthermore, some algorithms featuring short key bits can be easily attacked and recovered. For this reason, the algorithm DES has been widely substituted by AES. Therefore, the

asymmetric-key cryptography is taken into the consideration, which is based on very complicate mathematic problems. Consequently, these methods characterize the higher calculation complexity and longer run time. They are mainly exploited in the financial and military systems. Its typical representatives are RSA and ECC, cf. [RSA78] and [KKM08].

According to the purpose and function of cryptographic algorithms, people may have more choices to select the correct one or combination in reality resulting in various cryptosystems. Because of the widespread using, those cryptosystems feature a higher probability for being suffered from malicious attacks. For example, in Fig. 1.1, path a), Oscar, on the one hand, can intercept the secured information y from the message transferring channel; on the other hand, he can also get the information leakage, e.g., the variation of power consumption, magnetic radiation, etc., from the side channel of cryptosystem while the encryption or decryption is running, as shown in Fig. 1.1, path b). After that, he can recover the secret key between Bob and Alice by analyzing the captured side channel information. Therefore, the improvement of security level for the cryptographic algorithms and systems to counteract attacks is a crucial work.

1.2 Cryptosystems

Cryptosystem is the carrier of cryptographic algorithms, which can be implemented with both the software and hardware. As the development of chip technology, a lot of cryptosystems are implemented in the hardware, e.g., embedded systems, which provides the advantages in terms of fast run time, high through put, etc. For this reason, these embedded cryptosystems are widely used in the industry as well as our daily life, e.g., RFID tags, smart cards, mobile phones, satellite communication systems, etc. Therefore, more often the embedded cryptosystems are exposed to a risky condition, i.e., suffering different kinds of attacks. Those attacks can be divided into two parts, i.e., active and passive attacks. The former just leaves the cryptosystem in an extreme condition, where the system is running with malfunctions or errors. These errors can be fully exploited by the adversaries to reveal the secret key, e.g., fault analysis, cf. [BS97] and [BDL97]. The latter does nothing about the cryptosystem, which just collects and attacks the inherent information leakage from the running cryptosystem without causing any damages, i.e., side channel analysis. We will introduce two examples about the threatening of physical cryptosystems in the sequel.

Field Programmable Gate Array (FPGA) is a kind of circuit where the implemen-

tations can be reconfigured and customized by designers with the hardware description languages. In an FPGA-based system, the Intellectual Property (IP) core is usually stored into the on chip or external memory. When the system is powered on, the IP core will be transferred from the memory to the targeted FPGA chip to fulfill given tasks. However, during the data transferring with an unsecured channel, the bit file from IP core may be intercepted by Oscar intentionally. Oscar, on the one hand, can modify the bit file and then send it back to the FPGA chip leading to a malfunction system; on the other hand, he can use this bit file to clone the whole system resulting in big economic loss for the users or manufacturers. In other words, during the file transferring from the memory to the FPGA chip, a lot of attacks can be mounted, cf. [Dri09]. Therefore, a secure way to protect the IP core from being stolen during the transmission has to be taken into consideration.

Maybe FPGA is far away from our daily life, now let us take the smart card as an example. Smart card is so popular, for example, ID card, bank card, visa card, master card, gate control card, and so forth. Nearly every day, you use more than one of the mentioned cards in your life. When you insert your card into the card reader, besides the normal operations for card reading and writing, maybe another party is observing your card, where your input or output data may be intercepted. However, there exists a method, where the power consumption variation during operations is measured, then by means of the mathematic modeling and statistical analysis, the secret key may be revealed in a very short time, cf. [KJJ99]. The very interesting thing is that by detecting the magnetic radiation during the operations, the key may be revealed as well, cf. [AARR03]. The attack methods mentioned before are typical examples of side channel analysis in the path b), Fig. 1.1.

1.3 Side Channel Analysis

The basic idea for side channel analysis is that, without contacting the hardware system directly, the side channel information leakage, e.g., timing, power consumption, magnetic radiation, may be fully exploited to reveal the system key. The first practice in SCA is the Simple Power Analysis (SPA), which directly observes a single power consumption curve to recover some operation sequences or instructions in a running cryptosystem. Different from the SPA attack, the Differential Power Analysis (DPA) needs more power traces to reveal the secret key in cryptosystem by using statistical methods. Both SPA and DPA methods were proposed by Kocher et al. [KJJ99] in 1999, which demonstrates that

the power consumption attack on a physical cryptosystem, i.e., smart card, is feasible. Since then, a lot of scientists are attracted to devote themselves to this research area. After more than a decade's development, this field becomes a hot topic for the hardware security. More and more contributions were yielded. However, these contributions concentrate on two main aspects, i.e., attack methods and countermeasures. For the former, side channel information capture technologies, and leakage model building, etc., are exploited to improve various attacks considering the factors, e.g., run time, resource usage, and reality feasibility, etc. For the latter, different kinds of countermeasures are developed to make the system more secure by utilizing different means in terms of the algorithms, hardware architectures, peripheral components, etc., to counteract malicious attacks.

With regard to the attack methods, Chari et al. published in 2002 a paper on the so-called template attack [CRR02] and [GLRP06], where an extra fully controlled device is required to capture the profiling traces for template generation, after that the probabilities of the noise between the template and the analyzed power traces are calculated for key revealing. In 2004, Brier et al. proposed the Correlation Power Analysis (CPA) method [BCO04]. It can be seen as a transform of DPA attack, where, the correlation coefficient is calculated as a distinguisher between the leakage model and captured power traces. Later, in 2005, the stochastic analysis approach was introduced by Schindler et al. [SLP05] and [LRP07]. Such a method features some similarities to the template attack. Definitely, a fully controlled training device is necessary. Here, the leakage model and the profiling power traces from training device are mapped together by exploiting the least squares estimation algorithm. In 2008, Gierlichs et al. [GBTP08] proposed a new attack method called Mutual Information Analysis (MIA), which combined the knowledge from the information theory, i.e., mutual information, and supplied a generic side-channel distinguisher to reveal a secret key in the embedded cryptosystem. In 2012, we proposed an attack method called Power Amount Analysis (PAA) [TH12e], which relies on the additive Gaussian white noise channel in communication field aiming to attack the cryptosystem by exploiting a large set of time points in the time domain to contribute the information leakage. With respect to the protection of cryptosystems from malicious attacks, a large number of countermeasures were produced resulting in the hardened systems, where the unintentional information leakage is reduced in order to obstacle the diversiform attacks in practice, as detailed in [SP06], [MTT+05], and [SMBY05].

1.4 Thesis Organization

The security of the cryptosystem has been widely concerned and penetrated in our daily life by different of means. Therefore, in this thesis, we concentrate the side channel attacks on the FPGA-based cryptographic implementations, where some new and existing attack methods are proposed and considerably enhanced, respectively, meanwhile, some trace pre-processing approaches are given to access the secret key in cryptosystem efficiently. Finally, the enhanced attack methods and trace pre-processing approaches are combined as a tool for supplying us more choices and possibilities to evaluate the security of cryptosystem. The whole thesis is organized as follows:

Chapter 2 introduces a common architecture of FPGA and the source of power consumption in the running circuits. Then the building of leakage model to mimic the variation of power consumptions is discussed. Finally, the setup of the measurement platform is introduced for the subsequent evaluation of attack methods and trace preprocessing approaches.

Chapter 3 outlines the existing attack methods, as well as some countermeasures to neutralize malicious attacks in practice. Subsequently, in order to evaluate the efficiency of attack methods and the security of countermeasures, some evaluation metrics are introduced for the future using.

Chapter 4 extends and abstracts the definition of our new PAA attack method resulting in PAA methodology, where two key issues are presented. On the one hand, by exploiting the knowledge in communication field a way to preprocess the power traces is discussed, which reduces the dimensionality of analyzed data leading to the cutting down of calculation complexity; on the other hand, more distinguishers are introduced to meet the needs of specific attack requirements.

Chapter 5 introduces our trace form leakage model by means of the least squares estimation, which can improve the performance of PAA attack significantly. Then, based on the thoughts in PAA methodology, two derivative attack methods PAA-I and PAA-II are proposed. Meanwhile, in order to provide more accesses to reveal the secret key in a running cryptosystem, an attack framework is suggested, which combines different leakage models and attack methods.

Chapter 6 discusses our trace pre-processing approaches, i.e., horizontal alignment and vertical matching, in detail, which are mounted to preprocess the misaligned power traces produced from the random clock featured cryptosystem in the time and amplitude domains, respectively. In addition, some technical problems during the parameter adapting phase are discussed. Finally, a software architecture is proposed to deal with

the misaligned power traces easily in reality.

Chapter 7 is an extension of Chapter 6, where the influence factors for trace capturing and attacking are introduced first. Then, by exploiting the autogenetic features in PAA attack, the peak position based horizontal alignment is introduced to speed up the preprocessing of misaligned power traces without the pattern and vertical matching phases. Finally, the frameworks for attacking the misaligned and originally aligned power traces are given, respectively.

Chapter 8 studies and verifies the new proposed attack methods and trace preprocessing approaches in the previous chapters by attacking different implementations of block ciphers, i.e., AES, PRESENT, which are running with different clock types and frequencies.

Chapter 9 concludes the whole thesis and presents the future work.

Chapter 2

Embedded Cryptosystems

Contents

2.1 Introduction

The widely used embedded cryptosystems now face a lot of challenges to the system security. Regardless when and where, the invading of cryptosystems may take place, where the important data or user's information is leaked and intercepted. In order to well protect the embedded system from malicious attacks, one has to get familiar with the existing attack methods. In other words, the well understanding about the mechanism

of attacks may greatly help the designers to revise or create proper countermeasures. Therefore, in order to understand the side channel analysis more easily, the pre-knowledge about the architecture of FPGA and the source of power consumption in the circuits are introduced; then, the building of leakage model is discussed, which is studied by running AES and PRESENT algorithms, respectively. Finally, the measurement setup and the way to capture the power traces are detailed.

2.2 Field Programmable Logic Array

Field programmable logic array is a type of specific circuit, which can be exploited to implement the logic gates, e.g., AND, OR, NOT and XOR, complex combinational functions, and temporal logics, etc. The whole or a part of implementation can be re-designed by the customers exploiting the Hardware Description Language (HDL), e.g., Very high speed integrated Hardware Description Language (VHDL) and Verilog. Generally, the speed of FPGA is slower than the Application Specific Integrated Circuit (ASIC). In addition, it is difficult to fulfill complex designs in FPGA, where much more power is consumed. However, FPGA features many more advantages, e.g., shorter design period; flexible re-programmability; the internal logic designs can be modified for error correction procedure; the costs for such chips are relative low. Therefore, FPGA is widely used to implement the prototype of ASIC design. After different kinds of verifications in FPGA design, the mature one is transferred to ASIC leading to lower costs and risks in the chip design industry. Moreover, in some special industries, FPGA is directly taken as the components of electronic systems aiming to seize the market as soon as possible.

2.2.1 Architecture of FPGA

There are different kinds of FPGA vendors around the world, e.g., Xilinx, Altera, Actel, etc. The two biggest vendors are Xilinx and Altera, both produce FPGAs on the basis of SRAM, where the information will lose when the chip is switched off. FPGAs from Actel mainly feature anti-fuse technology, which embodies the advantages in terms of anti-radiation, higher temperature resistance, lower power consumption, etc. Therefore, FPGAs from Actel are widely used in the military and aerospace fields. The architectures for the FPGAs produced from different vendors are slightly different, however, they share some common principles. The elemental components in FPGA are introduced in the sequel.

The FPGA chip embodies four basic modules, Configurable Logic Block (CLB), I/O

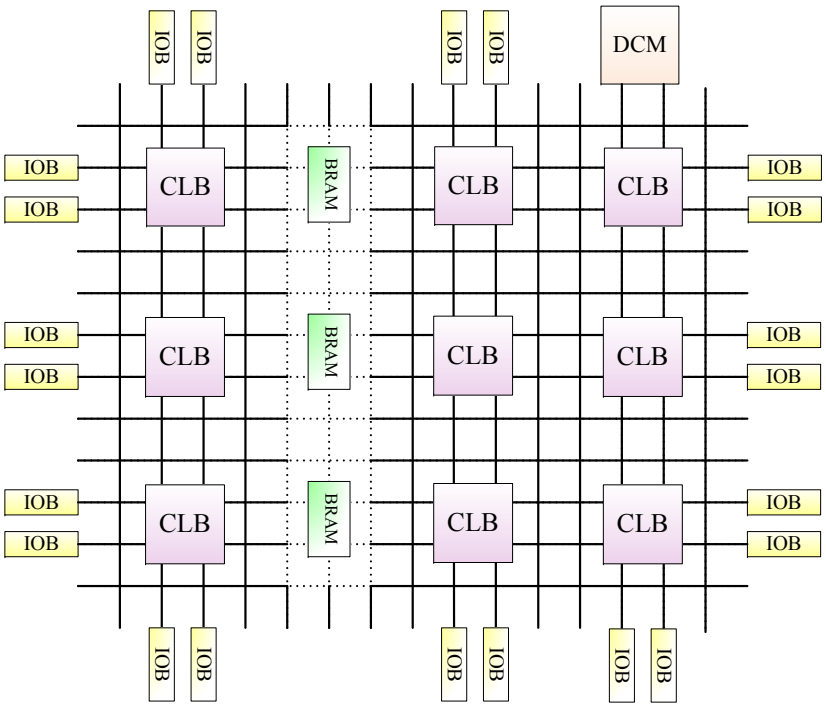

Figure 2.1: Internal Architecture of FPGA

Block (IOB), Block RAM (BRAM), and Digital Clock Manager (DCM), as shown in Fig. 2.1.

CLB is a fundamental logic unit within FPGA. The number of the CLBs varies from device to device. Each CLB contains a configurable switch matrix consisting of 4 or 6 inputs, multiplexers, and flip-flops. They can be used to implement the combinational or sequential logics, as well as the configuration of distributed RAM or ROM. In Xilinx FPGA family, CLB is composed of 2 or 6 slices with additional logics. Each slice has two 4-input Look Up Tables (LUT), user-controlled multiplexers to implement the combinational logic, arithmetic logic, as well as 1 bit register.

IOB is an interface functioning as a bridge to connect the chip's internal logics with external pins, which fulfills different electrical characteristics of the input/output signals. In order to meet the needs of different electrical standards, the IOBs in FPGA are divided into several groups by interface voltage, i.e., different banks. Please note that each bank can only have one specific voltage standard.

BRAMs are embedded into the FPGA bringing more application flexibilities, which can be configured as a single/dual-port RAM, Content Addressable Memory (CAM), and FIFO, etc.

Digital Clock Management is a very important component in the FPGA, which is exploited to manage the clock running in the FPGA chips, e.g., clock frequency divide,

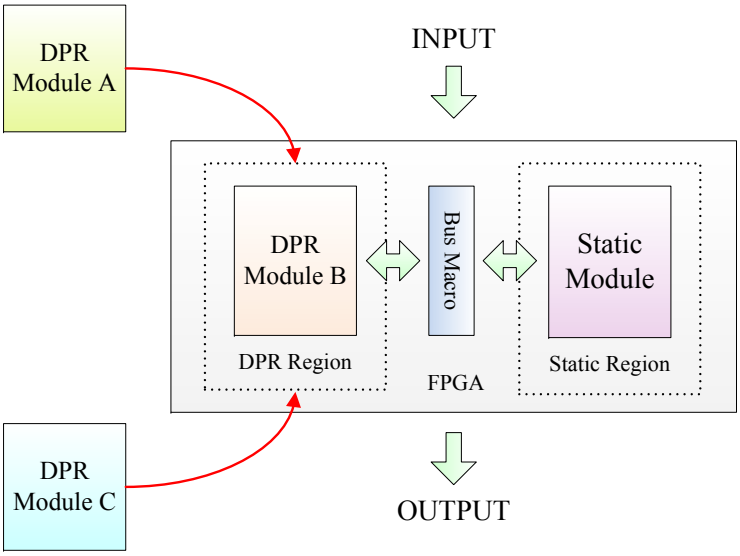

Figure 2.2: Workflow of Dynamic Partial Reconfiguration in FPGA

multiplication, skew elimination, and phase shift. It contains four functions, which are delay locked loop, digital frequency synthesizer, phase shift, and status logic.

2.2.2 Configuration of FPGA

Normally, the implementations, e.g., soft/hard IP cores, are stored into the on-chip RAM deployed in FPGA. Therefore, in order to change the functions or architectures of FPGA, the on-chip RAM can be re-programmed easily. When the FPGA is powered on, either the complied program data can be downloaded to the on-chip RAM or the compiled bit file can be loaded automatically to the FPGA via the flash to fulfill given tasks. There are several modes to configure the FPGA chip, for example, JTAG/boundary-scan configuration mode, master-serial configuration mode, slave-serial configuration mode, master-serial peripheral interface flash configuration mode, and so forth. The user can choose the configuration mode selectively according to the specific requirements.

2.2.3 Dynamic Partial Reconfiguration

Dynamic Partial Reconfiguration (DPR) is a new conception in FPGA technology. It denotes that some parts within the FPGA system can be dynamically reconfigured without interfering with the function of the system while the other parts are still well working.

Generally, FPGA chip is divided into three parts, i.e., dynamic partial reconfiguration region (DPR region), static region, and bus macro, as shown in Fig. 2.2. DPR region must be downloaded at some idle slots during the whole application. In this application,

three DPR modules can be downloaded selectively according to the specific requirements. These three modules can fulfill the same function with different type of implementations, or they all are different functions with different implementations. With regard to the bit stream downloading, one has two possibilities to fulfill this goal. On the one hand, these modules can be downloaded via the interface outside the FPGA; on the other hand, one may do it from the internal memory, where a control logic is needed in the static region to control the sequence of bit stream downloading. Meanwhile, such a region must keep running correctly to manage the input and output data during the whole application. Here, the bus macro is an only bridge for the communication between the DPR and the static region.

As mentioned before, the modules A, B, and C, can be reconfigured with the same function and different implementations. Assume that the modules A, B, and C at a certain time point consume the power consumption, e.g., 3, 5, and 9 units, respectively, while the static module dissipates 15 units power consumption. Then at the same time point, for doing the same task, the power consumption for the system features three possibilities, i.e., 18, 20, and 24 units. Therefore, by choosing the DPR modules randomly, the total power consumption from this system is randomized accordingly. For this reason, the DPR based cryptosystem may be a countermeasure to neutralize the power consumption attacks.

2.3 Power Consumption in CMOS Circuits

In this section, a brief introduction to the source of power consumption in the CMOS circuits will be given, which is an important reason why power consumption attack can be mounted feasibly. Nowadays, the electronic circuits are composed by the enormous number of CMOS transistors, meanwhile, ASICs and FPGAs are no exception. Therefore, the power consumption for the transistors switching in CMOS circuits should be considered carefully. Theoretically, the total power consumption arises from the hardware circuits are divided into two main parts, i.e., dynamic and static power consumptions. Both of them detailed in [WH10] are introduced in the sequel.

2.3.1 Dynamic Power Consumption

A CMOS inverter with a load capacitance shown in Fig. 2.3 is taken as an example. When the value of input V_{in} switches from 1 to 0, the pMOS-transistor conducts, and the capacitance C is charged with load voltage V_{DD} yielding the output value 1 for V_{out}

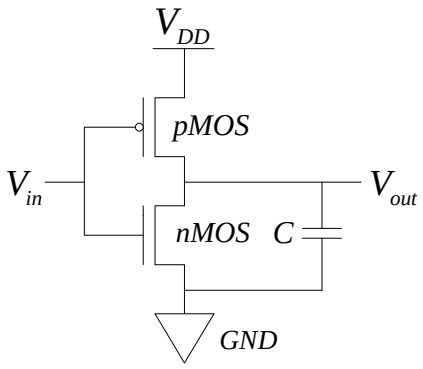

Figure 2.3: CMOS Inverter With a Load Capacitance

accordingly. On the contrary, when V_{in} changes the value from 0 to 1, then the nMOS-transistor switches on, the capacitance C is discharged via the ground line GND leading to the output value 0 for V_{out} accordingly. Because of the input value changing, the pMOS and nMOS transistors are switched on and off alternatively resulting in the capacitance charging and discharging accordingly. The most of the power dissipation for this inverter derives from the capacitance's charging and discharging, which can be calculated as follows:

$$E = \int_0^\infty C\frac{dV}{dt}V_{DD}dt = C \cdot V_{DD} \cdot \int_0^{V_{DD}} dV = C \cdot V_{DD}^2 \tag{2.1}$$

If the gate switches with the frequency f_{sw} during the time period T_c, then the number of certain capacitance charging and discharging is presented by $T_c \cdot f_{sw}$ resulting in average power consumption:

$$P_{sw} = \frac{E}{T_c} = \frac{T_c \cdot f_{sw} \cdot C \cdot V_{DD}^2}{T_c} = C \cdot V_{DD}^2 \cdot f_{sw} \tag{2.2}$$

P_{sw} is so-called *dynamic power consumption*. One finds that the transistors may not switch in every clock period. Therefore, in order to describe the switching frequency f_{sw} more properly, an activity factor α is introduced. Parameter f_{sw} is then depicted as the product of *active factor* α and system clock frequency f leading to the rewrite of dynamic power consumption:

$$P_{sw} = \alpha \cdot C \cdot V_{DD}^2 \cdot f \tag{2.3}$$

The parameter α is a statistic value, which can be roughly estimated. The maximum value of α features 1, i.e., the transistor gates switch in each clock period, otherwise the value is smaller than 1, however, larger than 0.

Another type of dynamic power consumption is called short circuit current. It describes that when the circuit is being pulled up or down, the network states are partially powered on. Consequently, the short circuit current occurs consuming a certain amount of power as well.

Theoretically, the dynamic power consumption consists of two main parts. On the one hand, it is the charging and discharging power consumption arisen from the CMOS gates switching; on the other hand, it is produced from the short circuit current between pMOS and nMOS transistors in the circuit networks.

2.3.2 Static Power Consumption

Static power consumption always exists regardless the CMOS gates switching. The source of the static power consumption arises mainly from the following four parts, subthreshold leakage, gate leakage, junction leakage, and contention current, cf. [WH10, 194-197]. All these leakages occur when the transistors are supposed to be turned off, which consume the power consumption as well. However, comparing with the proportion of dynamic power in total power dissipation, the static power consumption is negligible.

In order to have a further investigation between the dynamic and static power consumption in FPGA chip, the authors in [SKB02] proposed an example in Xilinx FPGA, Vertex-II series. They stated that the static power consumption requires $5 - 20\%$ of the total power dissipation. Consequently, the most portion of power consumption of a running chip derives from the dynamic power dissipation, which is the basic principle to mount power consumption attacks successfully.

2.4 Leakage Model

Leakage model exploits the mathematic approaches to estimate the variation of power consumption for a certain component in the circuits. Especially in side channel analysis, the efficiency of attack methods depends on the quality of the leakage model building. Usually, the estimation of power dissipation for the whole circuit is not necessary. The states changing of the targeted register are focused during the cryptographic operations. When a cryptosystem is running with some cryptographic implementations, a certain amount of registers is exploited and assigned to store and transfer the intermediate values in a binary form. Therefore, a bit value changing from 0 to 1 in a register denotes that a unit power is consumed. For instance, a byte register with 5 bits changing the values from 0 to 1 denotes that 5 units power consumption have been dissipated, which is a small

example to show the basic idea of leakage model building. Consequently, the well selection and carefully building of the leakage model play an important role in side channel analysis field. There are a lot of ways to build the leakage model and to mimic the variation of power consumption in cryptosystem, e.g., Hamming Wight (HW), Hamming Distance (HD), bit model, and zero-value model, etc., cf. [MOP07, pp. 129-135]. However, here, two very common leakage models, i.e., HW and HD, are introduced, respectively, which are widely utilized in practical attacks.

2.4.1 Hamming Weight Leakage Model

Hamming weight leakage model calculates the occurrence number of value 1 for a certain register in the cryptosystem, e.g., an eight bits register with value 11001001 features the Hamming weight value 4. This model just estimates the power consumption of current state for the analyzed register, i.e., the instant power consumption. Therefore, we named it as Instantaneous Leakage Model (ILM), see [TH12e], which has nothing to do with the previous states of the analyzed register. Therefore, it is easy and fast to mount this model into real attacks.

2.4.2 Hamming Distance Leakage Model

Hamming distance model considers the different corresponding positions between the two states of a certain register. For example, a register changes the state from $s_1 = 01101110$ to the state $s_2 = 10010110$ resulting in the Hamming distance value 5. One finds that the HD model estimates the variation of power consumption within a time interval. Therefore, we called it as Process Leakage Model (PLM), as detailed in [TH12e]. Comparing to HW model, the HD model involves the previous states into the calculation, which features higher calculation complexity than HW model. However, experimental results show that the HD model is more powerful than HW model in practice, see [MOP07, pp. 131].

Please note that the HD value can be achieved easily via the calculating of Hamming weight by (2.4). In other words, the Hamming distance can be seen as the Hamming weight calculation between the two exclusive-or-ed states.

$$HD(s_1, s_2) = HW(s_1 \oplus s_2) \tag{2.4}$$

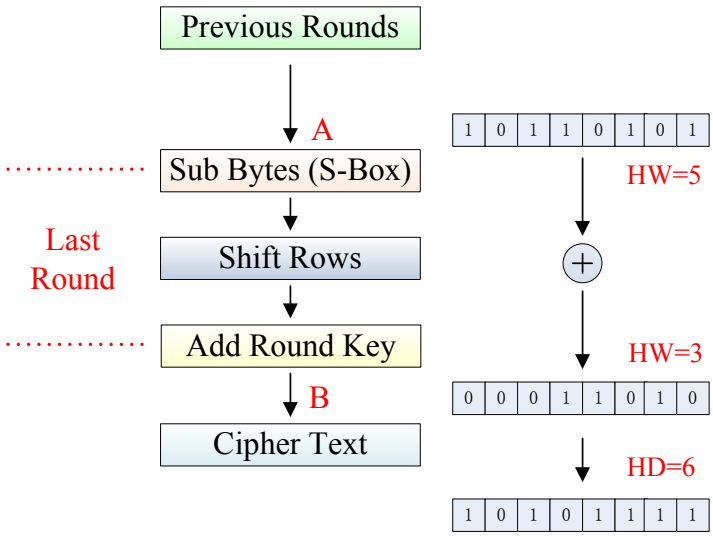

Figure 2.4: Wrokflow of AES-128

2.4.3 Examples in Leakage Model Building

Usually, before mounting attacks, it is better for the adversaries to have the knowledge about the specific implementations in the analyzed cryptosystem. Then, they may focus some weak parts in the hardware architecture to build the leakage model, where the information is considerably leaked. Therefore, in order to explain the mentioned leakage models and exploit them in the subsequent chapters, an AES-128 implementation is taken as an example to show how to build a leakage model for a cryptographic algorithm running in the cryptosystem. It includes two steps, which are analyzed register locating and leakage model selection. Please note that a wrong register targeting and an improper leakage model selection result in lower attack efficiency, successful rate, or fruitless attacks. Therefore, in this section, a short introduction about block cipher AES, light weight algorithm PRESENT, and their leakage model building is given.

2.4.3.1 Leakage Model Building for AES

AES The workflow of an AES-128 is shown in Fig. 2.4. The plaintext, ciphertext and the key are all 128 bits. The whole process features 10-round operations. In the first 9 rounds, each round embodies four operations, i.e., SubBytes, ShiftRows, MixColumns and AddRoundkey. In the last round, the operation of MixColumns is omitted. Therefore, for the last round, by means of the known ciphtertext and guessed key values, the register states before the SubBytes operation can be easily traced back. Hence, the cryptosystem may be attacked by building the HD leakage model focusing on the last round operation

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
j	1	6	11	16	5	10	15	4	9	14	3	8	13	2	7	12

Table 2.1: Index Table of HD Model in the Last Round

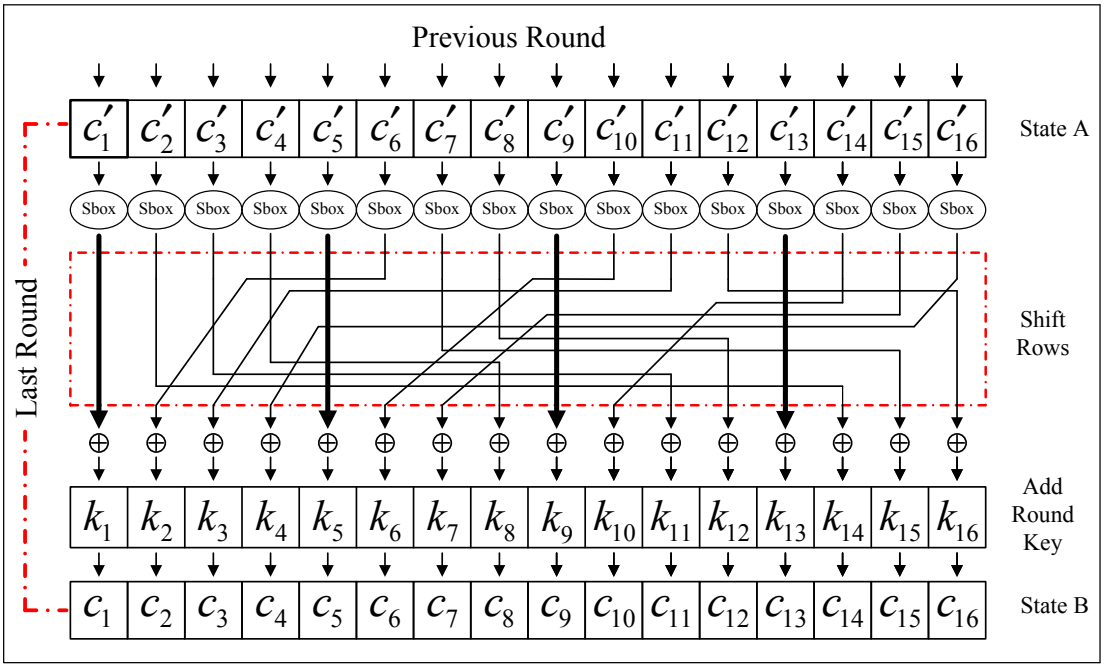

Figure 2.5: Register Scheme for Shift Rows

before and after the SubBytes operation in the sequel.

Leakage Model Building The HD model focusing the register states A and B before and after S-Box in the last round operation is described as follows:

$$\begin{aligned} HD(AB) &= HW(A \oplus B) \\ &= HW(c'_j \oplus c_j) \\ &= HW((Sbox^{-1}(c_i \oplus k_i)) \oplus c_j) \end{aligned} \tag{2.5}$$

where c'_i denotes the output from previous round. i and j are byte index, where $i, j \in [1, 16]$ holds. Because of the ShiftRows operation, the output byte values after SubBytes operation are shifted to different positions, except the bytes 1, 5, 9 and 13, as illustrated in Fig. 2.5. Therefore, in order to attack the i^{th} key byte value, the index values for i and j are different accordingly. For example, for revealing the second byte key, the Hamming distance model is calculated as $HD(AB) = HW((Sbox^{-1}(c_2 \oplus k_2)) \oplus c_6))$,

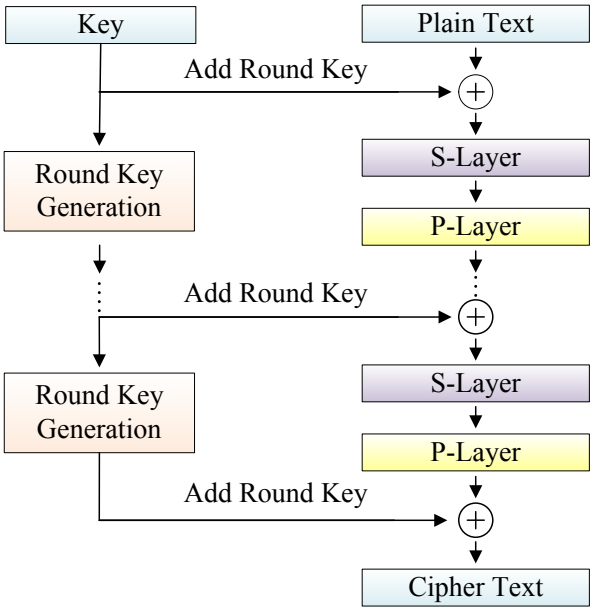

Figure 2.6: Workflow of PRESENT

i.e., the second key byte causes the register states changing in sixth position before and after the S-Box. Then, the corresponding relationship for indexes i and j are listed in Table 2.1. In general, the HD model is exploited to calculate how much power has been consumed during the register states changing from A to B.

The HW model considering the register state A in the last round before S-Box is calculated as follows:

$$HW(A)_j = HW(Sbox^{-1}(c_i \oplus k_i)) \tag{2.6}$$

where the Hamming weight for j^{th} byte in register A is caused by the i^{th} byte key for the ShiftRows operation, where $j, i \in [1, 16]$ holds. The corresponding relationship between i and j is shown in Table 2.1.

2.4.3.2 Leakage Model Building for PRESENT

PRESENT As known, AES is a very common algorithm, which is widely used in cryptosystems. However, it may not meet some special purposes, where the hardware resource is confined, e.g., RFID tags. Therefore, a light-weight block cipher algorithm, PRESENT was proposed in [BKL+] to meet the needs for both the resource and security requirements.

The whole algorithm was proposed on the basis of a SP-network [MOV96]. It consists of totally 31 rounds, where the length of plaintext is 64 bits and the key length

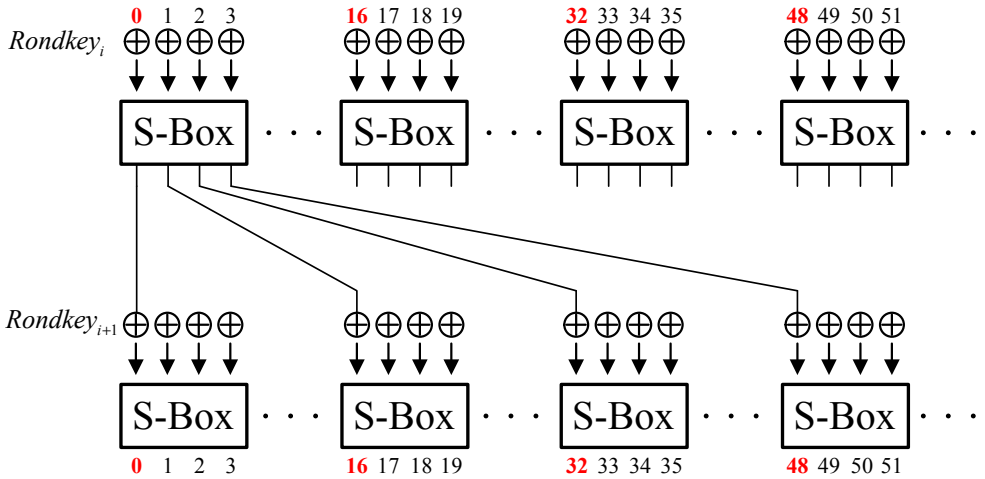

Figure 2.7: Work Scheme of the First Nibble Key

can be chosen selectively, e.g., 80 or 128 bits. The basic workflow for such an algorithm is illustrated as Fig. 2.6. Different from the AES, for the data processing, the whole PRESENT algorithm only features three main operations, for instance, add round key, substitution box (S-Box) and permutation. Before the data getting into the S-Box, it must be exclusive-or-ed with the round key. Then the output feeds to the S-Box, where the data are processed nibble by nibble, i.e., four bits in and four bits out. Therefore, for processing 64 bits input at the same time, 16 S-boxes are required. For attacking this system nibble by nibble, only 16 possible key values for each S-Box are guessed. After that each bit is permuted to another position in a 64 bits register line, as shown in Fig. 2.7. For example, for the first nibble key, after the S-Box operation, these four bits are assigned to the positions, i.e., 0, 16, 32, and 48, in the register line. The whole corresponding relationship for each bit before and after the S-Box is given by the Table 2.2. The completion of the mentioned three steps is taken as a round operation. This round operation runs for 31 times resulting in the ciphertext. For each round, the key is updated by running the key schedule algorithm as the round key, cf. [BKL+].

Leakage Model Building For attacking PRESENT, the Hamming distance model before and after the S-Box is built by (2.7), where p_i and k_i denote the i^{th} nibble plaintext and key, respectively; while c_i defines the output of S-Box.

$$HD = HW(p_i \oplus c_i) \tag{2.7}$$

$$= HW(p_i \oplus Sbox(p_i \oplus k_i)) \tag{2.8}$$

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P(i)	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P(i)	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P(i)	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P(i)	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

Table 2.2: Index Table of Permutation Layer

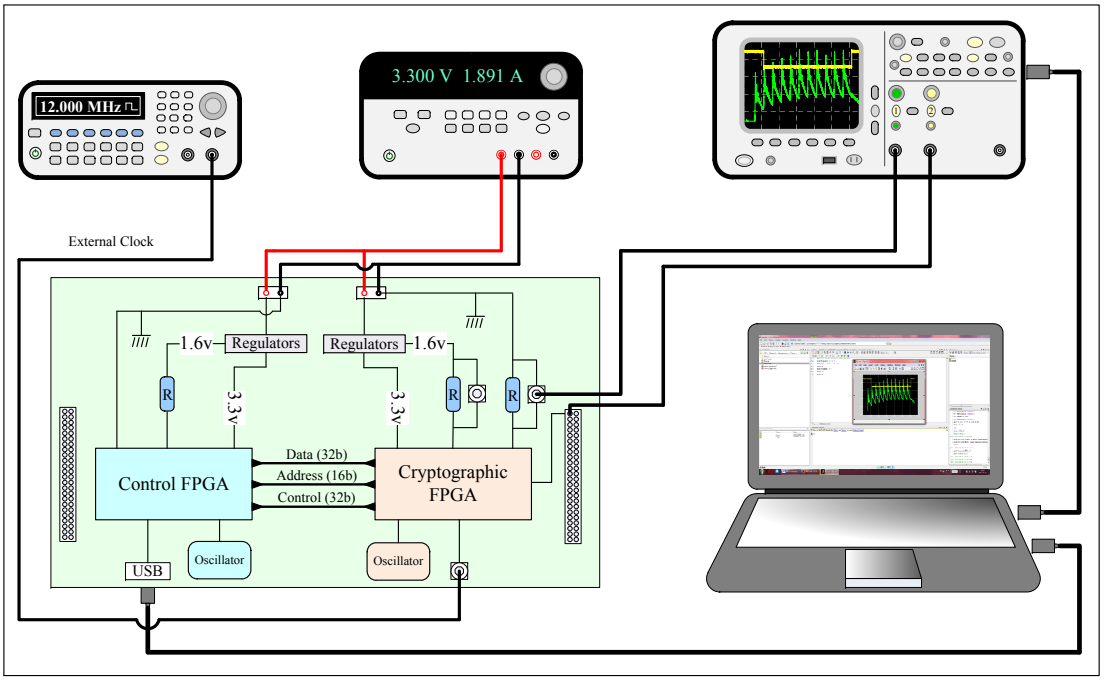

Figure 2.8: Measurement Setup

2.5 Measurement Setup and Power Trace Capture

In order to reveal the secret key of cryptosystems, a good measurement setup for capturing the power consumption in a working cryptosystem is necessary. Such a measurement setup plays an important role for the power consumption analysis, and helps the attacker to achieve high attack success rate by means of an oscilloscope with higher sample rate and resolution. In this section, a power consumption measurement setup is introduced, and some problems occurring in the trace capture phase are discussed as well.

2.5.1 Measurement Setup

As shown in Fig. 2.8, the whole setup is composed by five main parts, i.e., FPGA-development board, DC power supply, function/arbitrary waveform generator, computer, and oscilloscope.

Two FPGAs are deployed into FPGA-development board. One is used as the target

device to implement cryptographic algorithms; the other one is exploited for the board control, including the communication interface, which is a bridge between the target and out-board devices. Between these two FPGAs, there exist three buses, i.e., data, address, and control buses. The control and targeted FPGAs are driven by the on-board oscillators, respectively. However, which are removable. The external clock can be used as well via the SMA (SubMiniature version A) connector from an arbitrary waveform generator.

Function/arbitrary waveform generator is exploited to supply the FPGA board with adjustable higher-quality clock signal in real experiments. Therefore, the waveform generator can be easily adjusted with different clock frequencies. In order to have a stable power supply, a DC power supply is selected for providing the power to both FPGAs. For example, the 3.3V power, which is then divided into two parts by the regulators in the FPGA board: one is an unchanged 3.3V power for the peripheral devices, i.e., memory, configurations; the other one is a 1.6V power to the core chip of FPGA.

Computer is exploited for communicating with the targeted device via the control FPGA. The input and output data are sent and received via an Universal Serial Bus (USB) interface. During the trace capture, the encrypted data is verified with the software to assure the captured power traces are under the correct encryption or decryption operations. Meanwhile, the computer is connected to the oscilloscope via USB socket to store the captured power trace synchronously for the subsequent analysis.

In order to monitor the variation of power consumption produced from the target cryptographic FPGA, an oscilloscope with two channels is necessary, where, one is connected to the FPGA board to execute the trace capture task, and the other one is used to receive the trigger signal. The monitored power consumption is sampled and stored into on-line memory in the oscilloscope. Then, these data are written to the computer via the USB connecter. Usually, the higher sample rate the better quality can be achieved for the captured power traces resulting in the improvement of attack methods.

2.5.2 Power Trace Capture

In power trace capture step, the power consumption of a resister, inserted between the target FPGA and ground line, is measured. One may also measure the power consumption variation for the resister inserted between the 1.6V voltage supply and the core chip of target FPGA. In order to capture the power traces precisely, an extra trigger signal is needed. When the trigger signal is higher/lower than a threshold value, the oscilloscope is then informed to start the power traces recording. Therefore, it is necessary to output a trigger signal deliberately. For example, in the idle state, the trigger signal is high valid,

i.e., value 1 holds. During the encryption or decryption, the trigger signal is changed to 0. Then the capturing can be precisely executed by setting the trigger point in the oscilloscope as falling edge. However, for a real attack, it is difficult to yield or capture an extra trigger signal. The adversaries just record the power consumption from the very beginning when the cryptosystem is turned on. Therefore, they must know exactly the start point of the system, which may also be taken as a trigger signal in general. However, in scientific research, the trigger signal during the running of encryption or decryption can be output intentionally.

Chapter 3

Side Channel Analysis

Contents

3.1 Introduction

Side channel analysis represents the attacks on physical cryptosystems, where the information leakage, i.e., timing, fault, power consumption, magnetic and thermal radiation, etc., may be fully exploited to reveal the secret key in the running cryptographic implementations. As mentioned, a lot of attack methods were published in the last decade. On the one hand, they try to improve the existing approaches in terms of run time, side channel resource usage, and so forth; on the other hand, a lot of new attack methods were proposed by means of new models or conceptions from other disciplines.

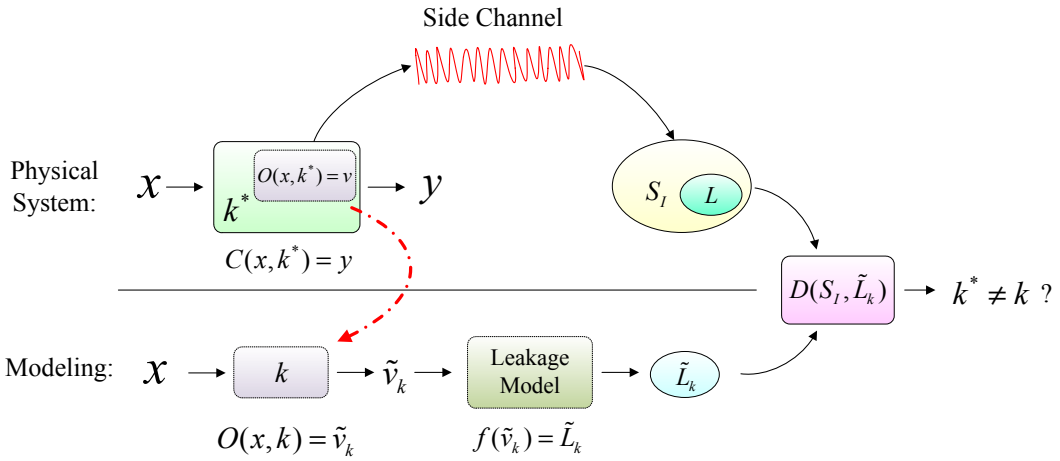

Figure 3.1: Workflow of Side Channel Analysis

The general workflow of side channel analysis is illustrated in Fig. 3.1. A cryptographic algorithm $C(x, k^*) = y$ is running in a physical system, where x and y denote the input and output, respectively, and k^* defines the secret key. During that time, some operations are executed by the function $O(x, k^*)$ resulting in the specific register state v. Meanwhile, the leaked side channel information S_I, e.g., power consumption, is monitored by the adversaries. However, not all the side channel information is useful because of the noise and other useless information. Actually, the adversaries focus on the data dependent information leakage L, where $L \in S_I$ holds. In order to mimic the register state v, the beforehand intercepted input x, output y, and guessed key value k may be exploited by function $O(x, k)$ resulting in the key dependent intermediate register states $\widetilde{v}_k$. Consequently, the leakage model, e.g., Hamming distance or Hamming weight is calculated to estimate the key dependent leakage model $\widetilde{L}_k$ by the function $f(\widetilde{v}_k)$. Finally, the estimated leakage model $\widetilde{L}_k$ and the side channel information S_I are analyzed by $D(S_I, \widetilde{L}_k)$ to reveal the correct key in cryptosystems. $D(\cdot, \cdot)$ denotes the statistical analysis methods, e.g., least squares, correlation coefficient, maximum likelihood, or mutual information, etc.

In this chapter, some existing attack methods are introduced first. Then the effective countermeasures are discussed to neutralize different attacks in practice. Finally, the metrics to evaluate the attack methods and the security of targeted devices are presented.

3.2 Attack Methods

Before we start to discuss the attack methods, a definition called *analysis region* used throughout the thesis is introduced first. It is a portion of the captured side channel

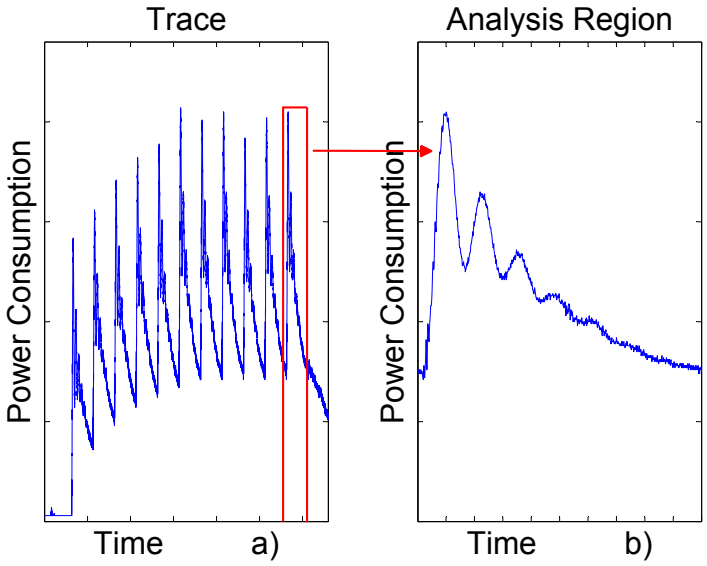

Figure 3.2: Analysis Region

information, which contains the information leakage the adversaries focus on. Take a captured power trace as an example, if the attackers concentrate on the last round operation of a standard AES-128, then the analysis region is the area, where the last round peak exists, as shown in Fig. 3.2. In real attacks, by determining such a region, a lot of time can be saved for analyzing the part where the adversaries are not interested in.

3.2.1 Simple Power Analysis

Simple power analysis fully observes the variation of power consumption with a single power trace to reveal the secret key of a running system or to reconstruct the operation sequence for the cryptographic implementations, see [KJJ99]. Because of the noises produced from the physical hardware, sometimes several power traces are captured for the noise reduction. However, this scheme shows some flaws, especially when the power consumption varies the power values considerably during the data processing phase. In that case, to identify the operation sequence and to reveal the correct key value for the analyzed cryptosystem become difficult. In addition, by mounting this attack, the attackers must have good knowledge about the targeted device and the running algorithms as well as some practical experiences.

3.2.2 Differential Power Analysis

Differential power analysis is a powerful attack method in side channel attack field, which is based on the statistical theories. Therefore, the adversaries may not know the details

of the inner architecture in targeted device, and a large number of power traces are required in the attack phase. The general attack scenario consists of three parts, i.e., side channel information capture, estimated leakage model generation, and attack mounting by statistical analysis. In this attack method, the adversaries tend to classify the power traces according to the values in the estimated leakage model, and then the differences of mean values for the classified power traces are calculated resulting in correct key value.

3.2.2.1 Side Channel Information Capture

In this thesis, the side channel information S_I defines the power traces captured from a running cryptosystem during its encryption or decryption. Therefore, N power traces compose of the trace matrix T with the size $N \times M$, where $T \in S_I$ holds. Each trace embodies M sample points. The power traces are one to one corresponded to the input p and output c, both of them feature the size $N \times 1$.

The power traces must be strictly aligned, which is a prerequisite for mounting DPA attack successfully, cf., [MOP07, pp. 120]. In other words, for a common time point in trace set T, all these power values are produced by the same operation. Otherwise, the misaligned power traces may cause difficulty during the attacks. Unfortunately, if the power traces are misaligned, some pre-processing efforts should be invested in advance to yield the sound power traces for the subsequent attacks.

3.2.2.2 Estimated Leakage Model Generation

The generation of the estimated leakage model $\widetilde{L}$ includes two steps. The first step is targeted register determining. Take an AES-128 as an example, which features 10-round operations. The operation round should be determined in advance, e.g., the first round or the last round. In the second step, the intermediate values $\widetilde{v}$ is calculated by exploiting plaintext p and possible guessed key value $\widetilde{k}$ via function $O(p, \widetilde{k}) = \widetilde{v}$. The intermediate value $\widetilde{v}$ is then mapped by the function $f(\widetilde{v})$ leading to the estimated leakage model $\widetilde{L}$ as follows:

$$\widetilde{L} = \begin{pmatrix} \widetilde{L}_{1,1} & \cdots & \widetilde{L}_{1,K} \\ \vdots & \ddots & \vdots \\ \widetilde{L}_{N,1} & \cdots & \widetilde{L}_{N,K} \end{pmatrix} \tag{3.1}$$

where N denotes the number of the input, and K defines the space of subkey values. In order to generate an efficient and feasible leakage model $\widetilde{L}$ for the subsequent attacks,

the key space K must be chosen carefully. For a register, the more bits one focuses on, the larger key space one may face. For instance, if the adversary focuses a byte register, then the possible key values are 2^8, i.e., $K = 256$. However, if he wants to attack a whole register with 16 bytes, then the key space is a huge and infeasible number, i.e., 2^{128}, which leads to an impossible mission. Therefore, for attacking block cipher AES-128, usually, the attack is mounted byte by byte. In other words, the DPA attack is mounted 16 times for revealing all the key bytes.

3.2.2.3 Attack Phase

The basic conception of DPA attack is that, after some transforming processes, the estimated leakage model $\widetilde{L}$ only contains two values, e.g., 1 and 0. According to these values, the power traces are classified into two groups. Then the mean values of the 1 and 0 featured power traces are calculated, respectively. Finally, the difference R between these two mean values is determined, from which the secret key of targeted cryptosystem can be eventually revealed. The whole attack scenario depicted in [MOP07, pp. 150-152] is outlined in the following.

Each element in $\widetilde{L}$ is compared to a threshold value q and stored back to the same position in $\widetilde{L}$. After that the element $\widetilde{L}_{i,j}$ in $\widetilde{L}$ only contains value 0 or 1. For example, the threshold value $q = 4$. If $\widetilde{L}_{i,j} \geq 4$ then, $\widetilde{L}_{i,j} = 1$ else $\widetilde{L}_{i,j} = 0$, where $i \in [1, N]$ and $j \in [1, K]$ hold. Subsequently, the mean values m_1, m_0 of 1 and 0 featured power traces are separated and calculated by following formulas, respectively.

$$m_{0(i,j)} = \frac{1}{n_{0(i)}} \sum_{s=1}^{N} (1 - \widetilde{L}_{s,i}) \cdot T_{s,j} \tag{3.2}$$

$$m_{1(i,j)} = \frac{1}{n_{1(i)}} \sum_{s=1}^{N} \widetilde{L}_{s,i} \cdot T_{s,j} \tag{3.3}$$

$$n_{1(i)} = \sum_{s=1}^{N} \widetilde{L}_{s,i} \tag{3.4}$$

$$n_{0(i)} = N - n_{1(i)} \tag{3.5}$$

where $i \in [1, K]$ and $j \in [1, M]$ hold. Finally, the difference $R_{i,j}$ can be calculated as (3.6). Then the correct key is identified by the largest difference value in R.

$$R_{i,j} = m_{0(i,j)} - m_{1(i,j)} \tag{3.6}$$

3.2.3 Correlation Power Analysis

Correlation Power Analysis, to some extent, is a variation of DPA attack. Comparing to the trace selection and group in DPA attack, CPA attack exploits the Pearson correlation coefficient as a distinguisher to analyze the relationship between the leakage model $\widetilde{L}$ and the captured power trace set T to yield the correct key featuring the largest correlation value.

Given two random variables $X = [x_1 \dots x_n]$ and $Y = [y_1 \dots y_n]$, both of them contain n elements, in order to evaluate the similarity between X and Y, Pearson correlation coefficient is carried out as follows:

$$\rho = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \tag{3.7}$$

where $-1 \leq \rho \leq 1$ holds. If $\rho > 0$, then X and Y are positive correlated; if $\rho < 0$, then negative correlated; and if $\rho = 0$, then uncorrelated; if $\rho = \pm 1$, then completely positive (negative) related. This is the basic idea of Pearson correlation coefficient.

In CPA attack, the power traces do not need to be classified. The attacks can be mounted directly by calculating the correlation coefficient (Corr) between the estimated leakage model $\widetilde{L}$ and the captured power trace set T by the formula (3.7).

$$\begin{pmatrix} R_{1,1} & \cdots & R_{1,M} \\ \vdots & \ddots & \vdots \\ R_{K,1} & \cdots & R_{K,M} \end{pmatrix} = D_{CPA}\left(\begin{pmatrix} T_{1,1} & \cdots & T_{1,M} \\ \vdots & \ddots & \vdots \\ T_{N,1} & \cdots & T_{N,M} \end{pmatrix}\begin{pmatrix} \widetilde{L}_{1,1} & \cdots & \widetilde{L}_{1,K} \\ \vdots & \ddots & \vdots \\ \widetilde{L}_{N,1} & \cdots & \widetilde{L}_{N,K} \end{pmatrix}\right) \tag{3.8}$$

$$R_{i,j} = Corr(T_{1:N,j}, \widetilde{L}_{1:N,i}) \tag{3.9}$$

The whole attack phase is depicted as follows:

1. Build the estimated leakage model $\widetilde{L}$.
2. Calculate the correlation coefficient between $\widetilde{L}$ and T, as shown in (3.8). The element value $R_{i,j}$ is then calculated by (3.9), where $i \in [1, K]$ and $j \in [1, M]$ holds.
3. Determine the maximum value in R, which indicates the correct key.

In practice, CPA is widely used for its low calculation complexity and high attack efficiency.

3.2.4 Profiling Based Attacks

Template attack and stochastic approach were proposed based on the prior knowledge of the attacked hardware, where an identical fully controlled device is required for profiling phase to build the key dependent template and stochastic model in advance. Both methods share a common conception that between each captured power trace and template or stochastic model the noise vector Z exists, which features multi-Gaussian distribution, as shown in (3.10). Here, for each attacked power trace, the probability of the noise vector Z can be obtained. Then the maximum likelihood method is used for each trace to identify the secret key in cryptosystems.

In this section, the general idea of the template attack and stochastic approach, detailed in [CRR02] and [SLP05], respectively, are outlined.

$$Prob(Z) = \frac{1}{\sqrt{(2\pi)^m |C|}} exp(-\frac{1}{2} Z^T \cdot C^{-1} \cdot Z) \tag{3.10}$$

3.2.4.1 Template Attack

The attack scenario for template attack contains two steps, i.e., profiling and attack phases, as discussed in the following.

Profiling Phase In the profiling phase, the template and the covariance matrix are calculated by means of the profiling traces produced from an identical fully controlled training device as follows:

1. For each subkey value k, N power traces T'_k are acquired, where the mean vector M_k for these captured power traces are calculated as the template:

$$M_k = \frac{1}{N} \sum_{i=1}^{N} T'_{k(i,:)} \tag{3.11}$$

2. Calculate the difference between pairwise $\sum_{i,j}^{K} M_i - M_j$, choose p time points where the large differences are shown.

3. Calculate the noise matrix $Z_k = T'_{k(j,1:p)} - M_{k(1:p)}$, where $j \in [1, N]$ holds. Then compute the covariance matrix $C_k = cov(Z_k)$ with the size $p \times p$.

Attack Phase In the attack phase, for each power trace captured from the targeted device, the probabilities of key dependent noise are calculated by the formula (3.10). Then

the secret key can be eventually revealed by exploiting maximum likelihood estimation, cf. [CRR02] and [GLRP06].

3.2.4.2 Stochastic Approach

Stochastic approach embodies some similar steps with the template attack. The difference between both attack methods is the profiling phase. Here, the leakage model, i.e., Hamming distance, Hamming weight, or bitwise model is required indirectly. In other words, these leakage models are projected into the captured profiling traces by exploiting least squares estimation. Then the estimation of the covariance matrix for the noise is calculated. The attacking phase is the same as template attack, for each trace, the maximum likelihood method is exploited to identify the largest noise probability, which indicates the correct key.

3.2.4.3 Stochastic Model

Profiling Phase In stochastic model, at any time point t, the captured power traces are modeled as $T_{t(x,k)} = L_{t(x,k)} + Z$, where, x denotes the input, and k represents the subkey value; $L_{t(x,k)}$ defines the data dependent part, i.e., leakage part; Z denotes the random noise. In order to estimate the $L_{t(x,k)}$ by $\widetilde{L}_{t(x,k)}$, where $\widetilde{L}_t = \sum_{i=0}^{u-1} \beta_i g_i$ holds, the least square estimation is exploited as follows:

$$\sum_{j=1}^{N_1} (T_{t(x_j,k)} - \widetilde{L}_{t(x_j,k)})^2 = \|T_t - M\widetilde{b}\|^2 \tag{3.12}$$

b is the coefficient, where $b := \beta_0, \ldots \beta_{u-1}$ holds. M consists of the element g, which is formed by the leakage model, e.g., Hamming weight, Hamming distance, or the power values in bitwise, as follows:

$$\mathbf{M} = \begin{pmatrix} 1 & g_{1,2} & \cdots & g_{1,u-1} \\ 1 & g_{2,2} & \cdots & g_{2,u-1} \\ \vdots & \vdots & & \vdots \\ 1 & g_{N_1,2} & \cdots & g_{N_1,u-1} \end{pmatrix} \tag{3.13}$$

where N_1 denotes the number of captured profiling traces, and u is the dimension of the base vector g. Subsequently, the coefficient b is estimated by $\widetilde{b}$ as $\widetilde{b} = (M^T M)^{-1} M^T T_t$. Then $\widetilde{L}_t = M\widetilde{b}$ can be achieved. In the estimation of covariance matrix C, the number of N_2 profiling traces with the fine chosen p time points are exploited, which holds the size $p \times p$, cf. [SLP05] and [LRP07].

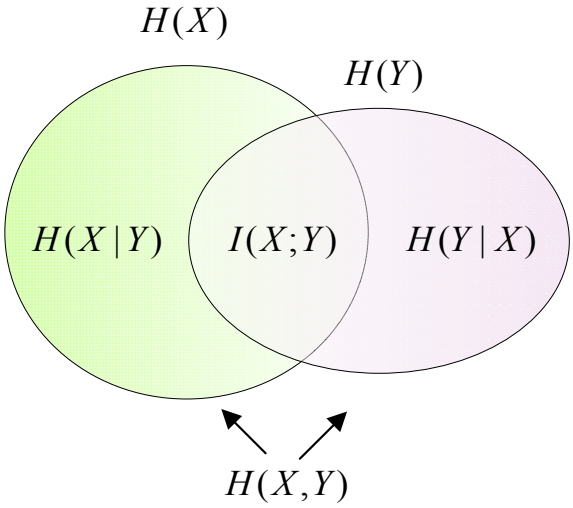

Figure 3.3: Relation Between Entropy and Mutual Information

Attack Phase Both parameters $\widetilde{L}_t$ and covariance C are utilized in the attack phase together with the N_3 power traces captured from the targeted device to depict the probability of noise by exploiting $R_t = \prod_{j=1}^{N_3} Prob(T_t(x_j,k) - \widetilde{L}_t(x_j,k))$ in (3.10), i.e., maximum likelihood estimation. Then the correct key can be determined.

3.2.5 Mutual Information Analysis

Mutual information analysis, proposed in [GBTP08] and [BGP+11], combines the knowledge from information theory, i.e., mutual information, to determinate the correct key value in the physical cryptosystem. A short introduction about such a method, detailed in [CT05, pp.31-48], is given in the following.

3.2.5.1 Mutual Information

There exists a random variable X with the element x. The probability mass function of X is $p(x)$, then the entropy is defined as follows:

$$H(X) = -\sum_{x} p(x) log_2 p(x) \tag{3.14}$$

The entropy measures the average uncertainty of random variable X. The base of the logarithm is 2, which means the unit of the entropy is bit. Consequently, it is widely used in the communication channel theory. $H(X)$ denotes the entropy for a single random variable. one can also define the conditional entropy as $H(X|Y)$, which means that the entropy of X is conditionally under the prior knowledge of another variable Y, as

given in the following:

$$H(X|Y) = -\sum_{x,y} p(x,y) log_2 p(x|y) \tag{3.15}$$

The reduction between $H(X)$ and $H(X|Y)$ is called mutual information, as shown in (3.16). Mutual information measures the mutual dependency between the analyzed two variables X and Y, which features the range $0 \leq I(X;Y) \leq min(H(X), H(Y))$.

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x,y} p(x,y) log_2 \frac{p(x,y)}{p(x)p(y)} \tag{3.16}$$

Therefore, both variables X and Y are independent when $I(X;Y) = 0$ holds; the larger of $I(X;Y)$ the more mutual dependent of X and Y are, which is the fundament that $I(X;Y)$ may be exploited as a distinguisher in the side channel analysis. The relationship between the entropy and the mutual information is visualized in Fig. 3.3, where the $I(X;Y)$ can be obtained in three forms by (3.17). $H(X,Y)$ implies the joint entropy of X and Y, which can be calculated as (3.18).

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{3.17}$$

$$H(X,Y) = -\sum_{x,y} p(x,y) log_2 p(x,y) \tag{3.18}$$

3.2.5.2 Mutual Information Analysis

CPA attack just measures the correlation coefficient ρ between the captured power trace set T and the estimated leakage model $\widetilde{L}$ to reveal the correct secret key in the running cryptographic algorithms. As a consequence, like CPA attack, T and $\widetilde{L}$ can also be analyzed by using mutual information:

$$I(T;\widetilde{L}) = H(T) + H(\widetilde{L}) - H(T,\widetilde{L}) \tag{3.19}$$

However, one finds that the mutual information cannot be calculated by exploiting the elements in T and $\widetilde{L}$ directly, the mass function $p(T)$, $p(\widetilde{L})$, and their joint mess function $p(T,\widetilde{L})$ must be determined in advance.

There are different ways to carry out mass function estimation. A common and easy

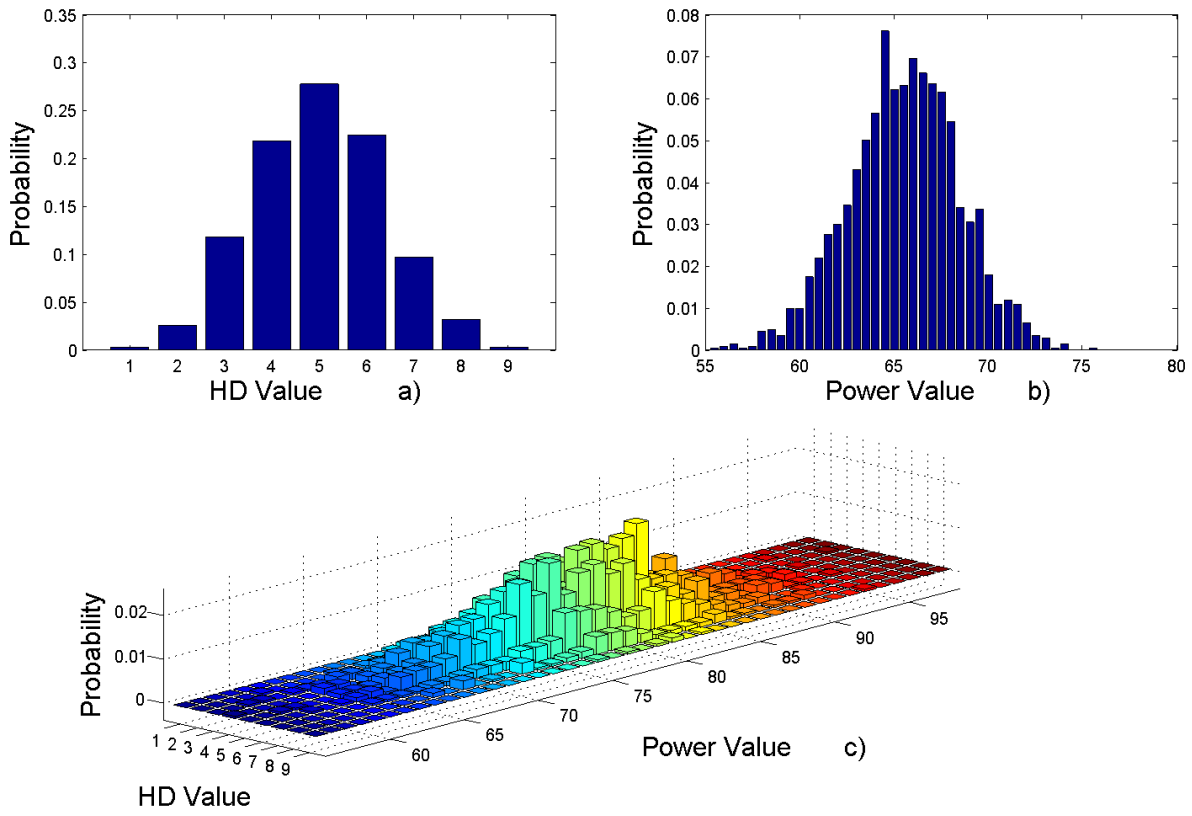

Figure 3.4: Histogram Examples

way is called histogram estimation, where the variables are divided and grouped into several bins, i.e., intervals. Then the number of occurrences for a certain value falling into the same bin is counted, which results in the probability density of the analyzed data. Fig. 3.4 a), b), and c) illustrate the mass functions estimated by histogram. Subsequently, the entropy $H(T)$, $H(\widetilde{L})$, and $H(T, \widetilde{L})$ can be calculated accordingly. Since the adversary has all the elements, the mutual information analysis can be mounted successfully, as detailed in [GBTP08] and [BGP+11]. One finds that the estimation of mass function plays an important role in MIA attack, which gives a direct impact on the attack results. Besides the histogram estimation, the authors of [BGP+11] suggested another way called kernel density estimation to estimate the mass function in practice.

3.3 Countermeasures

In order to neutralize different kinds of malicious attack, more and more countermeasures were proposed. The main idea for the countermeasure is to disconnect or disturb the relationship between the processed data and the leaked side channel information, e.g.,

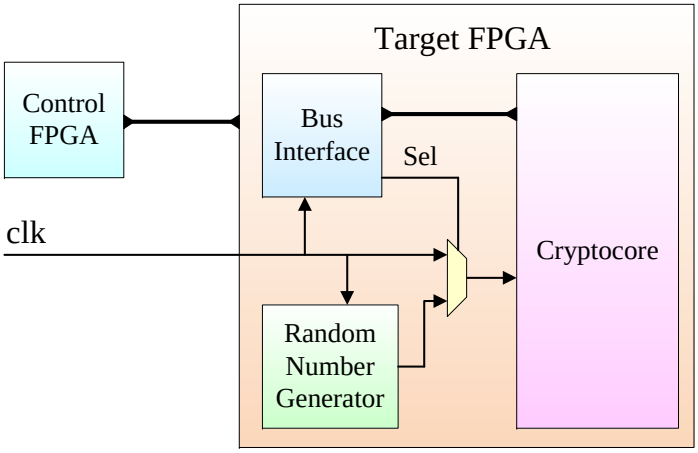

Figure 3.5: Setup of Random Clock Featured Cryptosystem

power consumption. We cannot prevent the system from being attacked completely. However, the improvement of the security level in cryptosystem is feasible.

3.3.1 Hiding

Hiding is a very common countermeasure for mitigating malicious attacks in reality, which features two purposes: on the one hand, randomizing the operations, input clock, etc., of cryptosystem to mislead attackers from the produced misaligned power traces. Regardless whichever method one uses, before carrying out attacks some pre-processing for the power traces is necessary; on the other hand, hiding increases the noise level in the circuits resulting in lower Signal to Noise Ratio(SNR), i.e., the useful information is hidden into the noise, which cannot be discerned easily.

Regardless the CPA, DPA, template attack, or stochastic approach, the prerequisite for all these attacks are the power traces must be aligned. Therefore, the misalignment injection in the captured power traces deliberately may be a good choice to counteract the attacks. There are different ways to achieve this goal, e.g., dummy wait sates inserting, clock frequency varying, etc. Especially in [YWV+05], the authors proposed a countermeasure, where not only the power supply voltage of the cryptosystem changes but also the system clock frequencies vary resulting in the power peak positions shift in the time domain as well as the power values changes in the amplitude domain. However, in all these mentioned countermeasures, only random clock changes the clock frequency fed in the cryptosystem, which impacts on the power consumption in hardware circuits. In other words, the clock frequency variation has direct relationship with the dynamic power consumption in the circuits, as shown in (2.3). Therefore, some phenomenons

accompanied must be considered before mounting the attacks. Here, the random clock featured cryptosystem is illustrated as Fig. 3.5. A random number generator is inserted between the clock line and the cryptocore. A multiplexer is exploited to choose the clock type, i.e., fixed clock or random clock, by the selection signal. The clock frequency in the clock chip is fixed and named as *base clock frequency* f_b, which feeds to the interface and the random number generator, respectively. The output f_r of the random number generator drives the cryptocore resulting in the misaligned power traces. Please note that $0 \leq f_r \leq f_b$ holds. Therefore, the maximum clock frequency of the f_r is f_b, i.e., the increasing of base clock frequency enlarges the range of random clock frequency.

Signal to noise ratio is a widely used terminology in the communication field, which defines the ratio of signal power P_s to noise power P_{noise} by:

$$SNR = \frac{P_s}{P_{noise}} \tag{3.20}$$

It shows the signal strength, which is emerged in the noise. The higher SNR, the easier one can distinguish the signal from the noise. On the contrary, the lower SNR value means the signal is weak and surrounded by too much noise. Therefore, under this condition, the signal cannot be discerned easily, i.e., the information is hidden in the noise.

With regard to the noise adding, theoretically, we have two possibilities to satisfy this aim: on the one hand, the information containing in the captured side channel information can be reduced without changing the amount of noise. However, it is infeasible, because the amount of leaked information cannot be easily controlled, which is blinded to the designers; on the other hand, the noise may be generated and added to the system, which is pretty easy by means of dummy operation inserting and shuffling. With all these methods, more power is dissipated, which, to some extent, emerges the information into the noise leading to lower success rate in real attacks.

The design technologies may help the cryptosystem to resist power consumption attacks. For instance, parallel architecture, dynamic dual-rail logics, etc. The dynamic reconfigurable FPGA mentioned in Chapter 2 can also be used as the countermeasure, where the S-Box implementations are changed randomly during the running encryption resulting in the changes of power consumption.

3.3.2 Masking

Masking is a most-used method in the cryptography field to counteract the power consumption analysis. This method exploits one or several mask value to hide the information

processing. The basic conception is that, the input or intermediate value x involved in the calculation in the cryptographic algorithm is concealed by a random value m using specific operations, e.g., $x * m$, where $*$ denotes the boolean or arithmetic maskings. Boolean masking defines the input or intermediate value x is concealed by exclusive-or with the masking value m, i.e., $x \oplus m$. In contrast, arithmetic masking presents the masking of input or intermediate value by means of arithmetic calculations, e.g., the modular addition or multiplication. Please note that m is known by the algorithm designer. However, the using of masking brings some extra calculations. Later, the masking value m has to be removed at the end of the calculation to yield the correct output, i.e., de-masking. Therefore, in order to remove masking value, one must calculate synchronously how this value changes during the running operations.

The cryptosystem can also be secured by exploiting both masking schemes selectively, cf. [CG00] and [Gou01]. For attacking masking featured cryptosystem, the second or even higher order attacks has to be mounted, which relies on the times of the masking value being used. Therefore, much more efforts are required for breaking such a system, cf. [OMHT06] and [JPS05].

3.4 Evaluation Metrics

How to evaluate the efficiency of attack methods or the security of cryptosystems is a crucial problem for attackers and system designers in practice. People may say my attack method is better than others. However, who knows? We need a standard to evaluate them. In this section, three evaluation metrics are introduced, which are exploited frequently in real attacks.

3.4.1 Time Requirements

Time requirements denote the run time for mounting attack methods or trace pre-processing algorithms in practice, which is a relative value relying on the computation conditions, e.g., the speed of CPU, the volume of memory, etc. However, to some degree, it demonstrates the efficiency of attack and trace pre-processing methods. Usually, the faster run time may help the adversaries to reveal the key of cryptosystems without being noticed within a limited timeframe. However, as a designer, the faster run time may shorten the security evaluation time for the analyzed cryptosystem. Therefore, the optimizing of attack methods or algorithms by using C/C++ codes, parallel, and GPU calculations, etc., improve the time requirements in real attacks considerably.

3.4.2 Success Rate

In order to evaluate and compare the attack methods and trace pre-processing approaches under different attack conditions, the success rate and guessing entropy, etc., are exploited as the metrics, respectively. In real attacks, different parameters are exploited by the adversaries and system designers to show the advantages for their attack methods and countermeasures. However, when comparing them together, it is difficult to determine, which one is the best attack method or countermeasure. Therefore, in [SMY09], the Success Rate (SR) and Guessing Entropy (GE) were proposed, which suggests a way to evaluate the quality of attack methods and implementations in cryptosystems. The general idea of these terminologies is discussed in the sequel.

In side channel analysis field, some factors, e.g., calculation time t, system memory m, and trace quires (usage) q, must be considered, which directly interferes with the attack performance. Therefore, the variation of any factor results in different success rate and guessing entropy. In this chapter, the trace usage q is considered as a mentioned factor.

Success rate, as the name suggests, it measures the probabilities to reveal a sub-key or the global keys in cryptosystem successfully. In practical attacks, each key k is mapped by a function $f : K \rightarrow S$ to the key class $s = f(k)$. $\mathbf{g} := [g_1, g_2, \ldots, g_{|S|}]$ is a guessing vector containing different key candidates. Please note that g_1 is considered as the most likely key candidate. After mounting an attack, the output of the attack is s. If $s \in [g_1, g_2, \ldots, g_o]$, then $R = 1$ is returned, otherwise $R = 0$ holds. Subsequently, the success rate of o^{th} order can be calculated by:

$$SR(q) = Pr(Experiments(R = 1)) \tag{3.21}$$

Usually, if there is no specific notation, the success rate is considered as first order, i.e., if the output $g_1 = s$, then R is assigned with value 1. During the increasing of the power traces with some interval, the attack in each increasing step is mounted to yield the output value R. After running this experiment for several times, the success rate can be achieved with the increasing of trace usage.

Here is an example, for attack one byte of AES-128 by exploiting the total number of 100,000 power trace. The correct key value is 137, i.e., $g_1 = 137$. The experiment is run for 100 times; for each experiment, the number of 1,000 power traces is used; for each attack, the number of 10 power traces is added. Therefore, 100-time attacks can be executed in each experiment. After executing such an experiment for 100 times, the

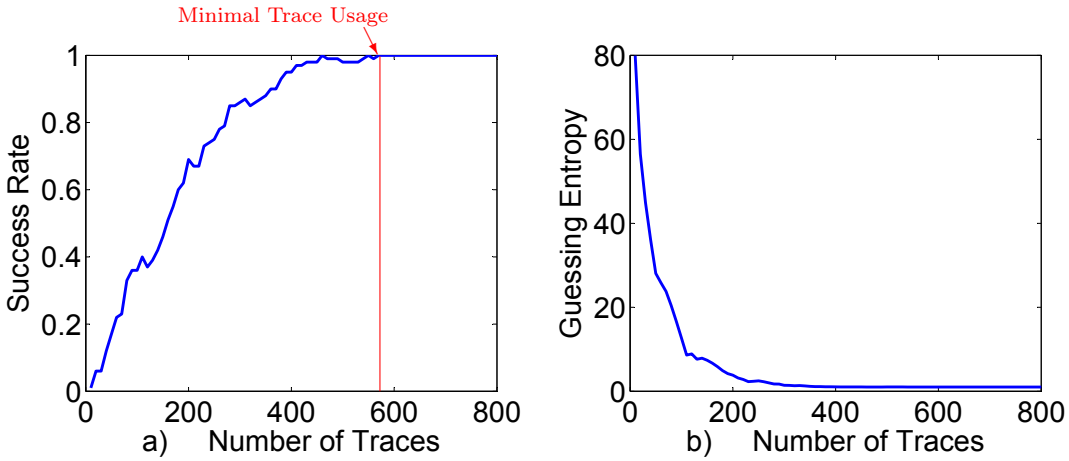

Figure 3.6: Examples of, a) Success Rate, b) Guessing Entropy

probabilities for each increasing step are calculated accordingly. Then, a success rate curve can be achieved, where the x-axis defines the trace usage, and the y-axis shows the success probability. The maximum value for success rate is 1, which denotes this byte is one hundred percent attackable, as illustrated in Fig. 3.6 a). It is clear that the minimum trace usage to reveal such a key byte successfully is 570, which is highlighted in the figure.

3.4.3 Guessing Entropy

Guessing entropy implies the average number of the key candidate to be tested during the attack. For each attack, the rank i of the correct key is returned and assigned as $R = i$. Then the guessing entropy can be calculated from:

$$GE(q) = E(Experiments(R = i)) \tag{3.22}$$

Continuing the previous example, the key space for a byte is 256, i.e., $|S| = 256$. For each attack, we sort all these 256 attack results and find out the index for the correct key value 137, then assign the index value to R. After mounting the experiment for 100-times, the average numbers of the indexes are calculated resulting in the guessing entropy in the y-axis, meanwhile, the x-axis presents the trace usage, as shown in Fig. 3.6 b).

Chapter 4

PAA: Power Amount Analysis Methodology

Contents

4.1 Introduction

Power amount analysis was proposed by us in [TH12e] and [TH12d], where a new way to understand the captured side channel information, i.e., power traces, was presented by means of the so-called Additive White Gaussian Noise (AWGN) channel in the field of telecommunication. Based on such a conception, the mechanism of power dissipation from physical cryptosystems is simulated as an AWGN channel, by which more time points in the captured power traces are exploited to contribute more information leakage

with limited trace usage resulting in a better attack performance in practical power consumption attacks. In comparison to the other attack methods, e.g., DPA and CPA, the PAA attack features the advantages in terms of run time, side channel resource usage, e.g., trace usage, misalignment tolerance, and amplitude fluctuation invariance, which are elaborated in the upcoming sections.

PAA attack arises from a following assumption. There is a register in a running cryptosystem, the states changing in this register from A to B span $0.001ms$, which is monitored and sampled by an oscilloscope running with 100 MHz sample rate. Obviously, there exist 100 discrete sample points in the captured power trace curve to depict such states changing process. If an attacker focuses on the operations during the states changing in the register, theoretically, these 100 time points do contain more or less the useful side channel information leakage, which can be fully utilized to reveal the secret key of the cryptosystem. However, in DPA and CPA attacks, only one time point leaking the biggest information is considered, whereas the other time points are just involved into the calculation as references. In contrast, in template attack and stochastic approach, several time points are being used to contribute the information leakage. However, the calculation complexities for both methods are quite high. Consequently, power amount analysis was proposed to balance the calculation complexity and the contribution of information leakage in practical attacks.

Our new proposed PAA attack exploits the correlation coefficient as a distinguisher to extract the useful information leakage from the targeted cryptosystem. However, one may have more possibilities in the distinguisher choosing. Therefore, in this section, we first detail the mechanism of PAA attack and its autogenetic advantages; then the power amount analysis methodology is introduced, which supplies us a thought to deal with the power traces, i.e., reducing the dimension of the analyzed data before mounting attacks. In this methodology, more statistical analysis methods are exploited as the distinguishers to determine the secret key of physical cryptosystem in practice.

4.2 AWGN Channel in Telecommunication

Telecommunication can be simply represented as the information signal transfer via an electromagnetic carrier. Such a technology now is sufficiently mature through more than one hundred years' development in both theoretical and practical aspects. How to model the telecommunication channel is a very important topic in this field, which to some degree determines the quality of the communication. In order to simplify the real, com-

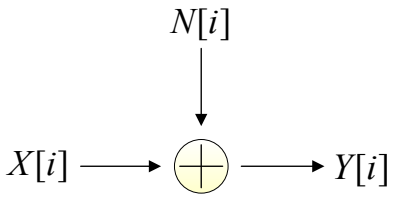

Figure 4.1: AWGN Channel

plicate, multi-noise, communication channels, a lot of models were proposed in the past. Among all these models, there exists a simple and easy but widely used one without considering the fading, interference, dissipation, etc., in real condition, which is called additive white Gaussian noise channel, as depicted in [TV05, p. 167], and [Gol05]. In such a model the noise features a Gaussian distribution, which is added to the signal during its transferring between two communication parties, as shown in Fig. 4.1. Usually, a discrete time AWGN channel is written by:

$$Y[i] = X[i] + N[i] \tag{4.1}$$

where $X[i]$ and $Y[i]$ denote the input and output signals of the channel in the discrete time domain, respectively; and $N[i]$ defines the additive white Gaussian noise while the signal $X[i]$ is passing through this channel. The input $X[i]$ and Gaussian noise $N[i]$ are independent and uncorrelated in the discrete time domain, where $N \sim \mathbf{N}(0, \sigma^2)$ holds, cf. [CT05, p. 261] and [TV05, pp. 29-30]. In other words, for a fixed AWGN channel, it always complies with the rule that whenever the input signal passes through the channel, the statistical characteristic for the noise always holds the mean value 0 and the constant variance σ^2, i.e., it is stable.

4.3 Power Amount Analysis Methodology

Usually, the captured power trace set T contains the information leakage L. However, this information leakage is immersed by a lot of noises from the inner and outside circuits. Different from the existing algorithms, the captured power traces may be pre-processed before mounting attacks, where the information leakage L is abstracted and purified, i.e., de-noising pre-processing. It is no doubt that the pre-processing phase does take some time. Usually one must balance the time requirements, the resource usage and the benefits from the trace pre-processing. The ideal way is to preprocess the power traces without the efficiency losing while mounting power attacks in practice.

In DPA and CPA attacks, by means of the statistical methods even without de-noising pre-processing procedure, the information leakage L can be still identified resulting in a successful key revealing. However, during the maximum information leakage point searching for both methods, the rest time points containing a certain amount of information leakage are not fully utilized leading to the waste of resources.

Therefore, the power amount analysis methodology is suggested in the sequel, by which, the mentioned drawbacks in the existing attack methods can be well handled in the following way.

1. Preprocess the captured power traces without increasing the attack time. Meanwhile, abstract and purify the information leakage leading to the improvement of success rate in real attacks.

2. Employ a large number of time points to contribute the information leakage to improve the attack performance.

Different from the definition of PAA attack we proposed in [TH12e] and [TH12d], PAA methodology extends such a specific definition to a general one, which not only presents a way to understand and preprocess the power traces, but also introduces different distinguishers, e.g., correlation coefficient and mutual information, etc., in the final attacks. Based on this principle, the process of de-noising and more time points calculation are realized by only one operation, which cuts down the calculation complexity by reducing the dimension of analyzed data and omitting the time points traversing step in practice.

4.3.1 Understanding of Power Traces

In order to mount attacks on the physical cryptosystem, the captured power traces are analyzed with the help of mathematic modeling. It is something like the pattern matching, where the leakage model can be seen as the key dependent template and the power traces are taken as the analyzed samples. The correct key will be revealed when the template is well matched with the analyzed samples. Power amount analysis is no exception. Therefore, based on the AWGN channel (4.1), the power dissipation path in a running cryptosystem is abstracted as an AWGN channel. The new hardware model is shown in Fig. 4.2 with the visualized trace examples. Assume the power consumption dissipated from the target hardware component, i.e., a register, is an input of channel model, and the additive Gaussian noise is added while the targeted power dissipation is propagating via this channel. Then the noise added power consumption is captured and sampled by the monitoring devices, e.g., oscilloscope. It is obvious that the power

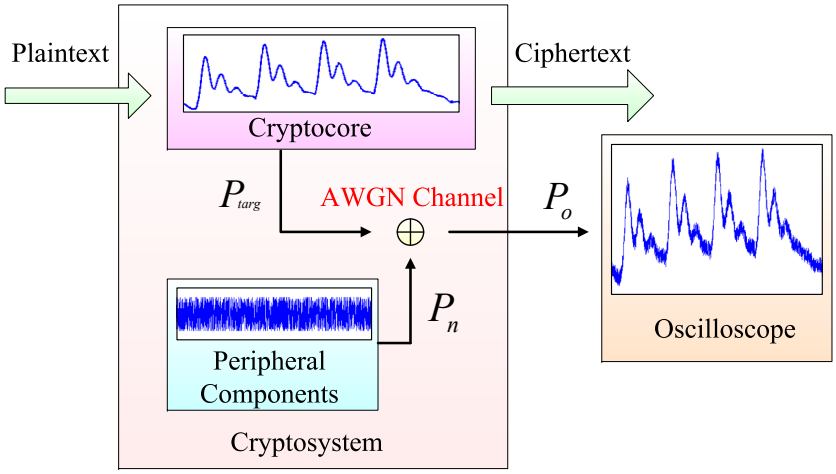

Figure 4.2: Signal and Noise Model for Cryptosystem

trace produced from targeted hardware embodies the pure power dissipation with clear profile. However, for the reason of noise, the captured power trace presents as a curve with an un-smooth profile. The time discrete power trace consists of two parts: one is the power consumption arisen from the target hardware during the process of encryption or decryption, which carries the information leakage correlated to the processed data and the architecture of running implementations; the other one is the noise stemmed from two sources. One is the true noise in the circuits, e.g., the peripheral components, power supply, clock generator, and existing electromagnetic circumstance, etc.; the other one is the non-targeted hardware operations. These noises are assumed featuring additivity, Gaussian distribution, which, meanwhile, are independent of the power consumption from targeted hardware. Consequently, each measured power trace is modeled in the discrete time domain as follows:

$$P_o[i] = P_{targ}[i] + P_n[i] \tag{4.2}$$

where $P_o[i]$ defines the output power consumption from the channel sampled by the oscilloscope; $P_{targ}[i]$ denotes the pure power dissipation produced by the target hardware component during the running of encryption or decryption, which contains the information leakage the adversaries focus on; and $P_n[i]$ implies the additive Gaussian noise, which is independent and uncorrelated with $P_{targ}[i]$ and features the fixed statistical characteristics. The new hardware model in (4.2) considers the time interval spanning a certain amount of time points in each captured power trace. In other words, the new hardware model considers the time points in the rows of trace set T, while DPA and CPA attacks

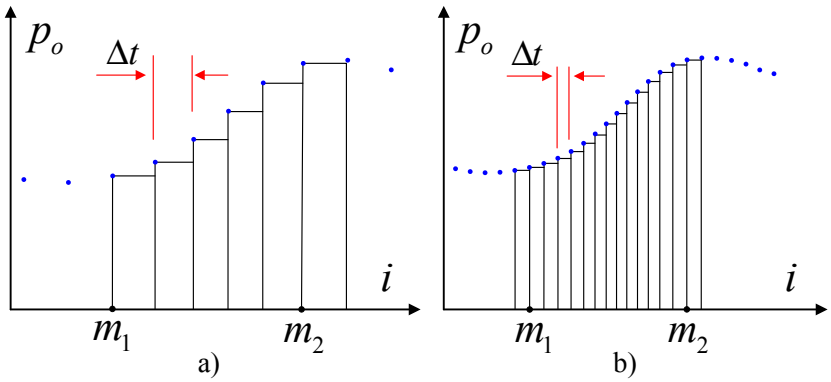

Figure 4.3: Discrete Power Consumption With Right-End Rectangle

concentrate on the columns. For the different power traces, the noise holds the same statistical property, i.e., $P_n \sim \mathbf{N}(0, \sigma^2)$, which is a prerequisite assumption for mounting the power amount analysis methodology feasibly.

4.3.2 Mean and Variance of a Power Trace

In this section, the mean and variance of a power trace are discussed by exploiting the hardware model outlined in (4.2) in the sequel.

Usually, a working cryptographic device dissipates the time continuous power consumption. In order to measure and visualize the behavior of this dissipation, a sampling oscilloscope is exploited to monitor the voltage changing of a resister, who may be inserted between the cryptographic device and the ground line in hardware circuits. The power curve in the vision of the oscilloscope consists of discrete points. Let's focus on a power curve P_o spanning the time from point m_1 to m_2, where $P_o[i]$ is an instantaneous power value with the index i in the discrete time domain.

Given a certain amount of work ΔW operated during a period of time Δt, the average power $\overline{P_o}$ can be calculated from:

$$\overline{P_o} = \frac{\Delta W}{\Delta t} \tag{4.3}$$

For a continuing power curve P_c in the time domain, the average power $\overline{P_c}$ can be calculated directly within a time interval Δt by the function $\overline{P_c} = \int_0^{\Delta t} P_c dt / \Delta t$. The power curve captured from the oscilloscope is composed by the discrete time points. Then the average power consumption $\overline{P_o}$ may be estimated approximately by exploiting integral thought. In other words, for each sampled point, a small area is established by drawing right-end rectangle, as shown in Fig. 4.3 a), then the average power for all these small

areas are calculated and integrated by exploiting the following formulas:

$$\begin{aligned} \overline{P_o} &= \frac{1}{(m_2 - m_1 + 1) \cdot \Delta t}(P_o[m_1]\Delta t + \cdots + P_o[m_2]\Delta t) \\ &= \frac{1}{(m_2 - m_1 + 1)}(P_o[m_1] + \cdots + P_o[m_2]) \end{aligned} \tag{4.4}$$

The average power consumption between m_1 and m_2 is just the mean value of all the sampled time points within $[m_1, m_2]$. In order to estimate the average power more precisely, one may just increase the sample points within the same time interval $[m_1, m_2]$, as illustrates in Fig. 4.3 b). In other words, the more sample points within the time interval $[m_1, m_2]$, the more precise for the average power estimation. Consequently, based on (4.2) the expectation of the power P_o can be rewritten as follows:

$$\begin{aligned} E(P_o) &= E(P_{targ}) + E(P_n) \\ &= E(P_{targ}) \end{aligned} \tag{4.5}$$

Theoretically, $E(P_o)$ denotes the constant part to the power trace, which is stable in the cryptosystem while the input data or key is changing. Such an item carries less information leakage the adversary is finding. Therefore, the key revealing by exploiting $E(P_o)$ is not feasible.

Because the power consumption of the targeted hardware P_{targ} and the noise P_n are independent and uncorrelated in the time domain, meanwhile, $E(P_n) = 0$ and $Var(P_n) = \sigma^2$ hold. $Var(P_o)$ may be easily calculated by:

$$\begin{aligned} Var(P_o) &= Var(P_{targ}) + Var(P_n) \\ &= Var(P_{targ}) + \sigma^2 \end{aligned} \tag{4.6}$$

The variance of the analyzed power trace denotes the degree where the element (time points) of the samples (the trace) deviate away from the average power value $E(P_o)$. Here, in (4.6), the $Var(P_o)$ means that the system's power values change around the average power value within the time points $[m_1, m_2]$, e.g., how much states changing caused power dissipation is consumed for the targeted hardware during that time interval, which attracts more attentions from the attackers. It embodies two portions, i.e., the

$Var(P_{targ})$ and $Var(P_n)$, respectively. $Var(P_{targ})$ is the variation of power consumption, which arises from the targeted register containing the abstracted and purified information leakage, i.e., $Var(P_{targ}) \in L$; the second portion is the variance of noise $Var(P_n)$, which is a constant value. The measuring of the item P_{targ} results in the most interested leakage $Var(P_{targ})$. Subsequently, the adversaries can analyze the relationship between $Var(P_{targ})$ and estimated leakage model $\widetilde{L}$ by means of $D(Var(P_{targ}), \widetilde{L})$ for the final key revealing. However, there is no direct way to separate and measure the item P_{targ} in a noise surrounded running cryptosystem, as well as the noise item P_n. Only the summation P_o of both items can be measured directly. However, one finds that the items $Var(P_{targ})$ and $Var(P_o)$ are in a linear relationship. Therefore, in the attack phase, $Var(P_o)$ and $\widetilde{L}$ are analyzed instead of $Var(P_{targ})$ and $\widetilde{L}$ to yield the correct key.

$$Std(P_o) = \sqrt{Var(P_{targ}) + \sigma^2} \tag{4.7}$$

$$\sqrt{x} = 1 + \frac{1}{2}(x-1) + \frac{1}{8}(x-1)^2 + \cdots \tag{4.8}$$

Theoretically, the standard deviation of P_o cannot be used to reveal the key values. Because the items $Std(P_o)$ and $Var(P_{targ})$ are in a non-linear relationship, as given in (4.7). However, the square root can be expanded to a Taylor series, as given in (4.8). By truncating this Taylor series with the first two items, $Var(P_{targ})$ and $Std(P_o)$ can be approximately taken in a linear relation. Consequently, $Std(P_o)$ may be a substitution for the $Var(P_{targ})$ in the real system attack. However, the attack performance may be not that good.

Now let's concentrate on the power consumption of a resister R, which is driven by the voltage U with the current I. Then the power consumption of the resister is calculated as $P_r = UI = U^2/R$. If the resister has the fixed value R, then the power consumption P_r and U^2 are in the linear relation. However, in power consumption attacks, the power voltage is measured by inserting a resister between the cryptographic chip and the ground line. In fact, the so-called captured power traces present the voltage variation of the resister. Usually, in DPA, CPA, and MIA attacks, the voltage variation curves are directly taken as the power consumption traces. In other words, the voltage U is treated as the square of voltage U^2, which, theoretically, is not that reasonable. In the attack of power amount analysis, the variance calculation for the voltage curves leading to the square of voltage, which can simulate the variation of power consumption veritably.

4.3.3 Attack Scenario

$$Var(T) = \begin{pmatrix} Var(T_{1,1} & \cdots & T_{1,m}) \\ & \vdots & \\ Var(T_{N,1} & \cdots & T_{N,m}) \end{pmatrix} = \begin{pmatrix} V_{1,1} \\ \vdots \\ V_{N,1} \end{pmatrix} \tag{4.9}$$

Relying on the new hardware model and the understanding of power traces, the variance or standard deviation is calculated by spanning a certain amount of the time instants as (4.9). After the calculation of variance, the power trace matrix T becomes a purified leakage vector V, i.e., one dimensionality of the matrix T is reduced, which is exploited for the similarity analysis with the estimated leakage model $\widetilde{L}$. In our original PAA attack, the Person correlation coefficient was utilized as a distinguisher to extract the information leakage in the captured power traces. However, in PAA methodology, the adversaries have more possibilities to choose the statistical analysis methods, i.e., distinguisher $D_{Stat}(\cdot,\cdot)$. The attack scenario of power amount analysis methodology is detailed as following.

1. Build the leakage model $\widetilde{L}$ by using plaintext p or ciphtertext c and subkey value k.

2. Choose the number of m time points in power trace set T within the analysis region. Calculate the variance or standard deviation of each trace in T and store them into the purified leakage vector V, i.e., $V_{i,1} = Var(T_{i,1:m})$, where $i \in [1, N]$ holds, as shown in (4.9).

3. Calculate the results R with the size $1 \times K$ by the selected distinguishers, e.g., difference of means, correlation coefficient, or mutual information, etc., between V and $\widetilde{L}$, as stated in (4.10).

$$\begin{pmatrix} R_{1,1} & \cdots & R_{1,K} \end{pmatrix} = D\left(\begin{pmatrix} V_{1,1} \\ \vdots \\ V_{N,1} \end{pmatrix}\begin{pmatrix} \widetilde{L}_{1,1} & \cdots & \widetilde{L}_{1,K} \\ \vdots & \ddots & \vdots \\ \widetilde{L}_{N,1} & \cdots & \widetilde{L}_{N,K} \end{pmatrix}\right) \tag{4.10}$$

Analysis Method: Difference of Means The purified leakage vector V and the leakage model $\widetilde{L}$ can be analyzed by the method difference of means, where V may be classified according to the values in $\widetilde{L}$ resulting in the correct key values.

Analysis Method: Pearson Correlation Coefficient (Corr) Correlation coefficient can be exploited directly for the similarity analysis, which is the basic idea of PAA attack. Here, the elements in R are computed as $R_{1,i} = D_{Corr}(V_{1:N,1}, \widetilde{L}_{1:N,i})$, where $i \in [1, K]$ holds.

Analysis Method: Mutual Information (MI) Like the correlation calculation, mutual information can also be exploited as a distinguisher here:

$$R_{1,i} = D_{MI}(V_{1:N,1}, \widetilde{L}_{1:N,i}) \tag{4.11}$$

$$= I(V_{1:N,1}; \widetilde{L}_{1:N,i}) \tag{4.12}$$

where $i \in [1, K]$ holds. Please note that the probability mass function for V and $\widetilde{L}$ must be pre-calculated.

Among all these analysis methods, only the correlation coefficient can be mounted directly and easily, whereas, the classification of power traces in difference of means, and the probability mass function calculation of mutual information do take some time and feature higher calculation complexity.

The prerequisite for mounting PAA methodology successfully is the choosing of parameters m, which relies on the sample rate of the oscilloscope and the shape of the captured power traces. The time points can be chosen around the analyzed power peak within the analysis region manually, which is enough to reveal the secret key. Please note that, within a certain range, the more time points are considered the fewer power traces are required to mount this attack. On the contrary, the choosing of the wrong or extreme more time points leads to the cutting down of attack performance. Therefore, on the one hand, more time points can be exploited in the calculation; on the other hand, in order to increase the sampled time points within a fixed time period, an oscilloscope featuring high-resolution is needed.

Theoretically, in PAA methodology, the whole method considers the factor of a time interval in the time domain. Therefore, the process leakage model, e.g., Hamming distance matches this attack method very well. On the contrary, the instantaneous leakage model, e.g., Hamming weight focuses the states changing at some time point. Consequently, the using of such a model in PAA methodology may result in lower success rate.

Usually, DPA and CPA attacks focus on a single time point, therefore, those attacks are classified as *time point attack*. In contrast, PAA methodology is categorized as *time interval attack* for concentrating on the power value changing within a certain time interval.

4.4 Autogenetic Advantages

In practical attacks, comparing to CPA attack, PAA methodology features some nice autogenetic advantages in terms of run time, trace usage, misalignment tolerance, amplitude fluctuation invariance. By means of these characteristics, PAA methodology can be applied in the attacks for both aligned and misalignment injected power traces.

4.4.1 Time Requirements

In PAA methodology, the calculation of variance and standard deviation change the two dimensions trace matrix T to one dimension purified leakage vector V, i.e., one dimensionality of the analyzed data is reduced, which cuts down the calculation complexity during the attack. Therefore, the run time is considerably shortened. As known, most of the mathematic methods focusing on the similarity analysis for two vectors can be exploited as a distinguisher in real side channel attacks. The different distinguisher algorithms feature different time requirements during the computation. However, with the same distinguisher, i.e., correlation coefficient, comparing to CPA attack, PAA attack features shorter run time, as shown in the experimental parts in [TH12e] and [TH12d].

One can benefit as well from the dimensionality reduction for the analyzed data. That's because sometimes the dimension of the analyzed data is higher, e.g., three dimensions, then we have to do the complex calculation. However, the simple calculation is mounted when the analyzed data are of reducible dimensionality. In the upcoming chapter, this thought is used to improve the attack methods considerably without reducing the attack performance.

4.4.2 Trace Usage

The trace usage is a crucial metric for evaluating attack methods. Comparing to DPA, CPA, and MIA attacks, the PAA methodology fully exploits a large number of leakage contained time points to contribute the information. Therefore, theoretically, for accumulating the same amount of information leakage, PAA attack requires fewer power traces, which was proven by us in [TH12d] and [TH12e]. However, we cannot conclude that this feature functions for every case. It should be verified in the future with more real attacks.

$$T = \begin{pmatrix} T_{1,6} & T_{1,7} & T_{1,8} & T_{1,9} & T_{1,10} & T_{1,11} & \dots & T_{1,100} \\ T_{2,6} & T_{2,7} & T_{2,8} & T_{2,9} & T_{2,10} & T_{1,11} & \dots & T_{1,100} \\ T_{3,6} & T_{3,7} & T_{3,8} & T_{3,9} & T_{3,10} & T_{1,11} & \dots & T_{1,100} \end{pmatrix} \tag{4.13}$$

$$T' = \begin{pmatrix} T_{1,6} & T_{1,7} & T_{1,8} & T_{1,9} & T_{1,10} & T_{1,11} & \dots & T_{1,100} \\ T_{2,4} & T_{2,5} & T_{2,6} & T_{2,7} & T_{2,8} & T_{1,9} & \dots & T_{1,98} \\ T_{1,7} & T_{1,8} & T_{3,9} & T_{3,10} & T_{3,11} & T_{3,12} & \dots & T_{1,101} \end{pmatrix} \tag{4.14}$$

4.4.3 Misalignment Tolerance

Assume that the CPA attack is mounted to the power trace matrix T, as shown in (4.13). All the time points do contain the information leakage, where the column 8 supplies the maximum one. In other words, the elements $T_{1,8}$, $T_{2,8}$, and $T_{3,8}$ are the best combination for the key revealing in CPA attack. However, for the misaligned power trace matrix T', as shown in (4.14), it is obvious that the good combination is deliberately broken by the misalignment injection. Meanwhile, CPA attack calculates the correlation coefficient between each column of the power trace and the leakage model $\widetilde{L}$. Therefore, these maximum information suppliers, i.e., $T_{1,8}$, $T_{2,8}$, and $T_{3,8}$ cannot contribute the information leakage together resulting in the descent of coefficient values. In trace matrix T', the time points $T_{1,8}$, $T_{2,6}$, and $T_{3,9}$ do supply the information leakage as well. However, this combination is not an optimum for CPA attack. In order to have the same attack results as before, the trace usage must be increased as a balance. However, in some extreme conditions, the captured power traces are limited, which cannot be increased any more.

In PAA methodology, the variance and standard deviation of each power trace within several time points are pre-calculated. Both methods are based on the statistics. If the number of samples is large enough, a small changing of any element or several elements in the samples cannot give great impact on the final results. Take the trace matrices T and T' again as an example. In trace matrix $T_{1:3,6:100}$, all these time points contribute the information leakage together. Therefore, the leakage strength is stronger than the single time point attack, i.e., CPA attack. In contrast, for misaligned trace matrix T', when calculating the variance within 95 time points, all the maximum leakage points, i.e., $T_{1,8}$, $T_{2,8}$, and $T_{3,8}$ are involved into the calculation and contribute the maximum information leakage together. Comparing to trace matrix T, in the second column two time points $T_{2,99}$ and $T_{2,100}$ are substituted by the elements $T_{2,4}$ and $T_{2,5}$ in trace matrix T'. However, this substitution does not greatly impact the variance value, which still yields

the stable attack results. Therefore, PAA methodology features stronger misalignment tolerance than the time point attack methods, e.g., CPA attack. In other words, PAA methodology can efficiently neutralize a small misalignment during attacks.

This feature has been also tested by us in [TH12e] and [TH12d] by exploiting the artificially generated misaligned power traces. However, in the upcoming experimental chapter, the mentioned feature is verified again with the true misaligned power traces produced from a random clock featured cryptosystem.

4.4.4 Amplitude Fluctuation Invariance

In PAA methodology, the variance or standard deviation calculation features an advantage, which is named as *amplitude fluctuation invariance*, and introduced as following.

$$X' = X + O = [X_1 + o \cdots X_m + o] \tag{4.15}$$

$$Var(X') = Var(X + O) = Var(X) \tag{4.16}$$

Given a vector $X = [X_1, \ldots X_m]$, one can get another vector X' by adding X with a vector O in the amplitude domain, which contains the number of m constant values o, as shown in (4.15). Regardless the value changing in vector O the variance or standard deviation value for vector X and X' are always the same, see (4.16). In a word, all the input elements add a same value leading to the same variance or standard deviation for the output. Therefore, we conclude that moving a power trace in the amplitude domain up or down, the variance and deviation before and after the moving are always the same. Because of this property, for a captured trace set T, moving each trace in the amplitude domain randomly, the yielded purified leakage vector V is always the same. Therefore, when mounting PAA methodology, the attack results R are always the same before and after the moving in the amplitude domain.

We have applied this feature in [TH12d] to neutralize the clock frequency effects proposed in [TH12b]. These effects interfere with the attack results in DPA and CPA attacks, etc., considerably. However, they do not degrade the results of PAA methodology. Therefore, PAA methodology can counteract the clock frequency effects automatically for its amplitude fluctuation invariance.

In practice, the power consumption fluctuated in the amplitude domain intentionally can be taken as a countermeasure to antagonize the power consumption analysis, which may be achieved by exploiting following ways: varying the power supply for cryptographic circuits; feeding the cryptosystem with random clock frequency resulting in

clock frequency effects; exploiting the dynamic reconfigurable technology in FPGA to implement cryptographic algorithms. However, all these countermeasures become invalid when facing with PAA methodology.

4.5 Summary

In this chapter, the definition of PAA attack is extended to PAA methodology. Besides the four merits of PAA attack, in PAA methodology, two viewpoints are conveyed. Firstly, this methodology supplies us a way to process the power traces, where, based on the AWGN channel model, the dimensionality of the analyzed data is reduced resulting in lower calculation complexity; Secondly, for the similarity analysis, not only the correlation coefficient can be used, but also the other algorithms, e.g., difference of means, mutual information, can be involved as a distinguisher to reveal the secret key in physical cryptosystems.

Chapter 5

Trace Form Leakage Model in PAA

Contents

5.1 Introduction

In order to improve the performance of power consumption analysis in practice, the qualified attack methods are needed, where different means are exploited to extract the information leakage from attacked cryptosystem. Meanwhile, without modifying attack methods, a fine chosen leakage model may be selected, which can precisely depict the power consumption variation of the targeted components in a running cryptosystem. Therefore, the leakage model building plays an important role in side channel analysis field, which determines the performance of attack methods.

Usually, in DPA and CPA attacks, Hamming distance and Hamming weight are simple and easy leakage models to mimic the possible information leakage for the targeted hardware components. In contrast, in template attack and stochastic approach, the leakage model exists in the form of key dependent traces. However, in order to generate such models, an extra fully controlled identical device is needed to collect the profiling traces in advance. We, therefore, in [THH12], classify the leakage models into two categories according to the existence form, i.e., *mathematic* and *trace* form leakage models. Here, these two leakage models are abstracted as $\widetilde{L}_{ma} = f_{ma}(x, k)$ and $\widetilde{L}_{tr} = f_{tr}(\cdot, \cdot)$, respectively, where x is the input or output of the cryptosystem, and k defines the possible key

values. The mathematic form leakage model simulates the variation of power consumption by exploiting mathematic method $f_{ma}(x, k)$. For instance, f_{ma} may be a mapping of Hamming Distance or Hamming weight, etc. The trace form leakage model can be achieved in two ways. Firstly, it may be produced by exploiting simulation software or fully controlled hardware to profile the running processes. For example, in the template attack, the profiling trace set T_p can be achieved from a fully controlled identical device, then the trace form leakage model is written as $\widetilde{L}_{tr} = f_{tr}(T_p)$, where f_{tr} denotes the mean value calculation; secondly, the key dependent mathematic form leakage model $\widetilde{L}_{ma}$ may be injected into the power traces leading to the estimated trace form leakage model $\widetilde{L}_{tr}$. For instance, in the stochastic approach, the mathematic form leakage model f_{ma} is mapped into the profiling power trace set T_p as $\widetilde{L}_{tr} = f_{tr}(\widetilde{L}_{ma}, T_p)$, where f_{tr} denotes the calculation of stochastic model.

Given a trace form leakage model $\widetilde{T}_k$, where $\widetilde{T}_k \in \widetilde{L}_{tr}$ and $k \in K$ hold, usually, one has to resort to the maximum likelihood method mentioned in stochastic approach for the secret key revealing, where the calculation is very complicated. However, the situation has changed. We proposed in [THH12] an attack method combining the trace form leakage model $\widetilde{L}_{tr}$ with power amount analysis, where $\widetilde{L}_{tr} = f_{tr}(\widetilde{L}_{ma}, T)$ holds. Please note that the captured power trace set T is exploited directly to build the trace form leakage model. Therefore, the profiling phase is not needed any more. The whole attack relies on the thoughts of PAA methodology, which can preprocess the power traces to extract and purify the inherent information leakage meanwhile to reduce the dimensionality of the analyzed data.

In this chapter, the building of trace form leakage model $\widetilde{L}_{tr}$ is discussed in detail, where the existing mathematic form leakage model $\widetilde{L}_{ma}$ is mapped into such a trace form model by utilizing least squares estimation. Then, an attack framework by exploiting different forms of leakage model and analysis methods in power consumption analysis field is given, where the attack methods can be chosen selectively according to the specific requirements.

5.2 Trace Form Leakage Model Building

In order to understand the mechanism of trace form leakage model building, in this section, the least squares estimation is introduced first. Subsequently, the building of trace form leakage model is elaborated.

5.2.1 Least Squares Estimation in Statistics

The least squares estimation is a widely used method for solving data fitting problems in statistics field, which finds the minimal squares between the observed and estimated data. The general idea of this method detailed in [HTF01] is discussed in the following.

Given an observed variable vector $X^T = (X_0, X_1, \dots X_{u-1})$, where u denotes the dimension of X, the output value Y can be written by the linear equation as:

$$Y = \beta_0 + \sum_{i=1}^{u-1} X_i \beta_i. \tag{5.1}$$

β_0 is an intercept (or bias), which is correspondent to the constant value 1 in X_0, i.e., $X_0 = 1$. Then the linear model from (5.1) is rewritten as:

$$Y = X^T \beta \tag{5.2}$$

This equation may be expanded to the matrix form, then (5.2) is transformed as $Y = X\beta$, and the elements X, β are presented as follows:

$$X = \begin{pmatrix} X_{1,0} & \cdots & X_{1,u-1} \\ \vdots & \ddots & \vdots \\ X_{N,0} & \cdots & X_{N,u-1} \end{pmatrix} \tag{5.3}$$

$$\beta = \begin{pmatrix} \beta_{1,0} & \cdots & \beta_{1,m-1} \\ \vdots & \ddots & \vdots \\ \beta_{u-1,0} & \cdots & \beta_{N,m-1} \end{pmatrix} \tag{5.4}$$

which results in the output matrix Y. Each column in Y can be calculated by $Y_j = X\beta_j$, where $j \in [0, m-1]$ holds, i.e., each column in Y can be linear expressed by the production of matrix X and its counterpart column in β.

$$Y = \begin{pmatrix} Y_{1,0} & \cdots & Y_{1,m-1} \\ \vdots & \ddots & \vdots \\ Y_{N,0} & \cdots & Y_{N,m-1} \end{pmatrix} \tag{5.5}$$

Given the matrix X, Y can be estimated by $\widetilde{Y}$ to find the minimal distance between these to matrices as

$$\| Y - \widetilde{Y} \|^2 = \| Y - X\widetilde{\beta} \|^2 \tag{5.6}$$

$\widetilde{\beta}$ is the estimation of β. One finds that by choosing the estimator $\widetilde{\beta}$, the minimal distance can be achieved. If X is full column rank, where any column of X cannot be written as the linear equation with other columns, i.e., $X^T X$ is nonsingular. Then the unique coefficient estimator $\widetilde{\beta}$ can be calculated as follows:

$$\widetilde{\beta} = (X^T X)^{-1} X^T Y \tag{5.7}$$

Subsequently, by choosing carefully the estimator $\widetilde{\beta}$, Y can be recovered by its prediction $\widetilde{Y} = X\widetilde{\beta}$. In the upcoming section, these calculations are exploited for the generation of trace form leakage model $\widetilde{L}_{tr}$.

5.2.2 Trace Form Leakage Model Building

The generation of key dependent trace form leakage model $\widetilde{T}_k$ by exploiting least squares estimation was discussed by us, in [THH12]. Such a conception was first applied into the power consumption analysis field by stochastic approach as the profiling phase, cf. [SLP05], which is detailed in the sequel.

The key dependent trace form leakage model $\widetilde{T}_k$ may be written as a linear product with fine built matrix A_k and coefficient estimator $\widetilde{\beta}_k$, i.e., $\widetilde{T}_k = A_k\widetilde{\beta}_k$, where $k \in K$ holds. The estimator coefficient $\widetilde{\beta}_k$ can be achieved by finding the minimum distance between the captured power trace set T and key dependent trace form leakage model $\widetilde{T}_k$ as follows:

$$\widetilde{\beta}_k = argmin\| \, T - \widetilde{T}_k \, \|^2 = argmin\| \, T - A_k\widetilde{\beta}_k \, \|^2 \tag{5.8}$$

Subsequently, the estimator coefficient $\widetilde{\beta}_k$ may be calculated by the following formula:

$$\widetilde{\beta}_k = (A_k^T A_k)^{-1} A_k^T T \tag{5.9}$$

$$\mathbf{A_k} = \begin{pmatrix} 1 & A_{1,1} & \dots & A_{1,u-1} \\ 1 & A_{2,1} & \dots & A_{2,u-1} \\ 1 & \vdots & & \vdots \\ 1 & A_{N,1} & \dots & A_{N,u-1} \end{pmatrix} \tag{5.10}$$

It is obvious that an elaboration of A_k determines the success rate for subsequent attacks. Here, the mathematic form leakage model, i.e., Hamming distance, hamming weight, or the register states in bitwise are exploited to build the matrix A_k, as shown in (5.10). The first column of A_k is always a constant value 1, which defines the bias

or intercept in mathematics. However, in power consumption analysis, it represents the non-data dependent part in the captured power traces. The other columns in matrix A_k denotes the data dependent portion. Therefore, $A_{i,1:u-1}$ can be filled by the Hamming distance (weight) values, or the bitwise values of the targeted register. For instance, if $u = 2$, the second column is filled by exploiting the values from the Hamming distance or weight model; if $u = 9$, the bitwise model for a certain register in the cryptosystem is used. In a word, most of the mathematic form leakage model can be exploited here. The power trace set T embodies the number of m time points, as shown in (5.11), i.e., m time points around the power peak in the analysis region are involved into the calculation. Different from the building of the stochastic model, the trace matrix T is directly used into the calculations to establish the trace form leakage model.

$$T = \begin{pmatrix} T_{1,1} & T_{1,2} & \dots & T_{1,m} \\ T_{2,1} & T_{2,2} & \dots & T_{2,m} \\ \vdots & \vdots & & \vdots \\ T_{N,1} & T_{N,2} & \dots & T_{N,m} \end{pmatrix} \tag{5.11}$$

Subsequently, by exploiting the estimator coefficient $\widetilde{\beta}_k$ and matrix A_k, the adversaries can recover the trace form leakage model $\widetilde{T}_k$ as follows:

$$\widetilde{T}_k = A_k \widetilde{\beta}_k \tag{5.12}$$

When drawing a certain row of $\widetilde{T}_k$, the shape of the drawn curve is similar to the power trace in set T, which is the reason why it is called trace form leakage model. The calculation thoughts of this model are the same as the building of the stochastic leakage model. However, the differences are: on the one hand, the exploited power traces are directly captured from the targeted device rather than the identical fully controlled device; on the other hand, the final attack phase is greatly different. In other words, in the trace form leakage model building and the attack phase, the same power trace matrix T produced from the targeted device is utilized. Therefore, an advantage is shown, say, the fully controlled device is not needed any more nor is the profiling step, which leads to a fast calculation.

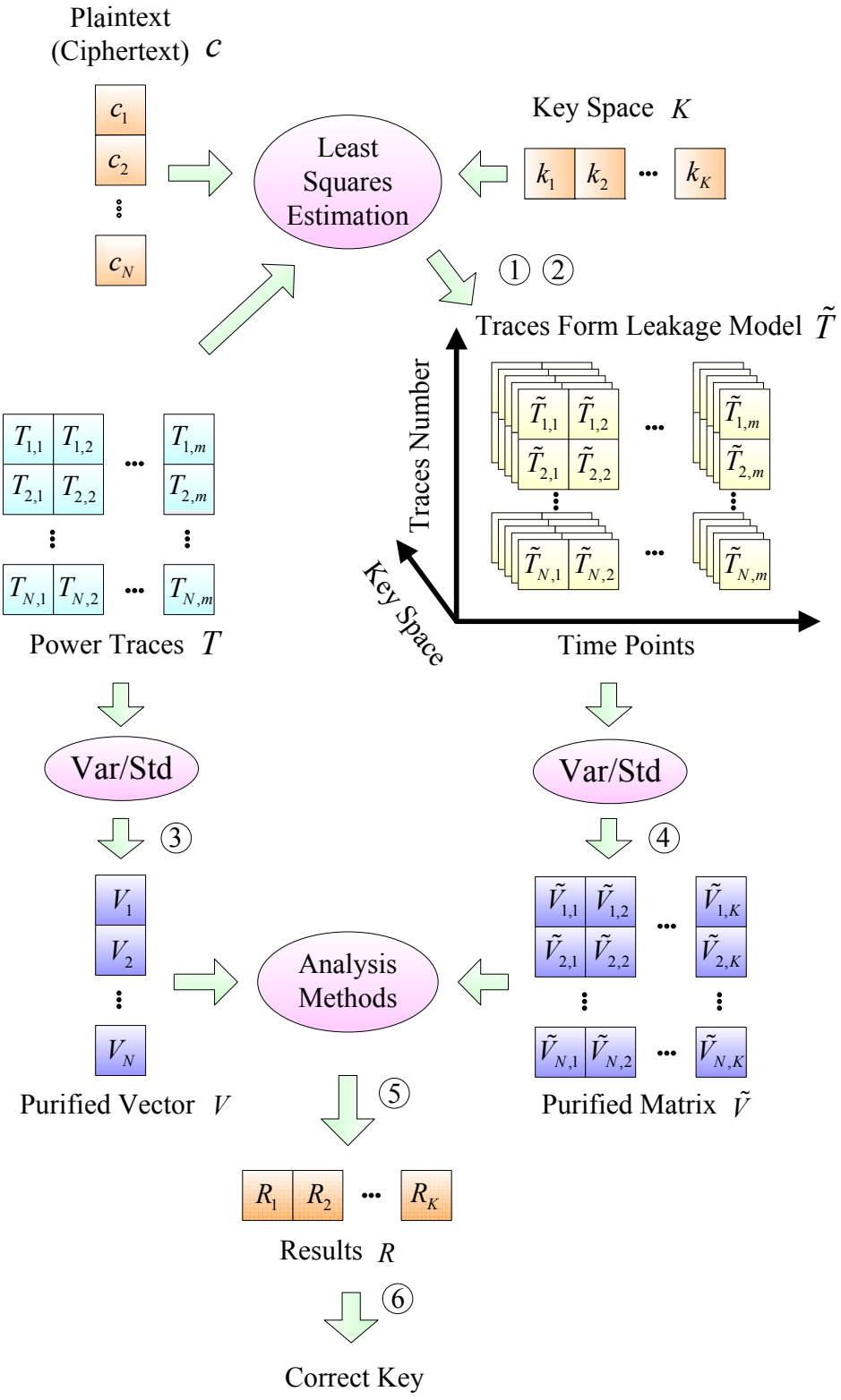

Figure 5.1: Attack Architecture

5.3 Attack Scenario

Now the captured power trace set T and the key dependent trace form leakage model $\widetilde{T}$ are at hand. Then the variance or standard deviation for each trace in both parties is calculated, respectively. Finally, PAA methodology can be mounted.

The whole attack scenario to mount PAA methodology combining with trace form leakage model is given as follows:

1. Choose the number of m time points around the power peak in the analysis region in trace set T.

2. Generate mathematic leakage model $\widetilde{L}_{ma}$, e.g., Hamming distance or bitwise leakage model, for the targeted register. Inject the constructed $\widetilde{L}_{ma}$ into the power trace set T by exploiting least squares estimation algorithm (5.12) to establish the trace form leakage model, i.e., $\widetilde{T} = f_{tr}(\widetilde{L}_{ma}, T)$.

3. Calculate the variance or standard deviation for each captured power trace $T_{i,1:m}$, where $i \in [1, N]$ holds, resulting in the purified leakage vector V holding the size $N \times 1$.

4. Calculate the variance or standard deviation for each estimated trace form leakage model $\widetilde{T}_{i,1:m}$, where $i \in [1, N]$ holds, resulting in the purified estimated leakage matrix $\widetilde{V}$ with size $N \times K$.

5. Analyze the leakage vector V and each column of the purified estimated leakage matrix $\widetilde{V}$ by using different distinguishers, e.g., difference of means, correlation coefficient, or mutual information, to yield the attack results R accordingly.

6. Identify the correct key value in the attacking results R.

The whole attack scenario is visualized in Fig. 5.1. One finds that by applying the thoughts of PAA methodology in real attack, three dimension matrix $\widetilde{T}_k$ is transformed to a two dimension variance matrix $\widetilde{V}$, meanwhile the two dimension trace matrix T becomes a variance vector V. After that, the calculation complexity in the attack phase is cut down considerably.

When building the trace form leakage model, the adversaries can choose $u = 2$, $u = 9$ or other values in a selective manner. No one can conclude which dimension of the matrix A is optimal, the choosing of the parameter u relies on the specific implementations and the hardware architectures. The fine chosen of this parameter results in a higher success

rate.

In order to verify the proposed attack framework, we, in [THH12], mounted CPA and PAA attacks by exploiting the mathematic and trace form leakage models, respectively, on an FPGA-based cryptosystem running with AES-128 cryptographic algorithm. The attack results showed that the trace form leakage model combing with the PAA attack can achieve a better attack performance than those attacks applying directly the mathematic leakage model in practice.

5.4 Attack Framework With PAA-I and PAA-II

In this section, an uniform attack framework is proposed for the key revealing with different leakage models and analysis methods. Meanwhile, two mutation forms of PAA attack are introduced, which are named as PAA-I and PAA-II, respectively. In Fig. 5.2, five choices to mount attacks are detailed as following.

1. Analyze the captured raw traces T and estimated mathematic leakage model $\widetilde{L}_{ma}$ by using correlation coefficient, difference of means, and mutual information selectively, as shown in Fig. 5.2 Analysis 1), which is the basic framework for DPA, CPA, and MIA attacks.

2. Calculate the purified vector V, and the mathematic leakage model $\widetilde{L}_{ma}$ by different means selectively, as shown in Fig. 5.2 Analysis 2), which presents the basic idea of PAA methodology.

3. The power traces T and the estimated trace form leakage model $\widetilde{L}_{tr}$ can be analyzed directly by exploiting the maximum likelihood, as illustrated in Fig. 5.2 Analysis 3). However, before analyzing both parties, the distribution of noise between them will be estimated and a profiling phase is needed. A typical representative for this framework is the stochastic approach.

4. The purified leakage vector V and leakage matrix $\widetilde{V}$ are analyzed, as illustrated in Fig. 5.2 Analysis 4), where different statistic methods are mounted for both parties. Such an attack framework is named as Power Amount Analysis mutation I (PAA-I). The calculation of trace from leakage model does take some time, which, however, can depict the estimated information leakage for the targeted hardware

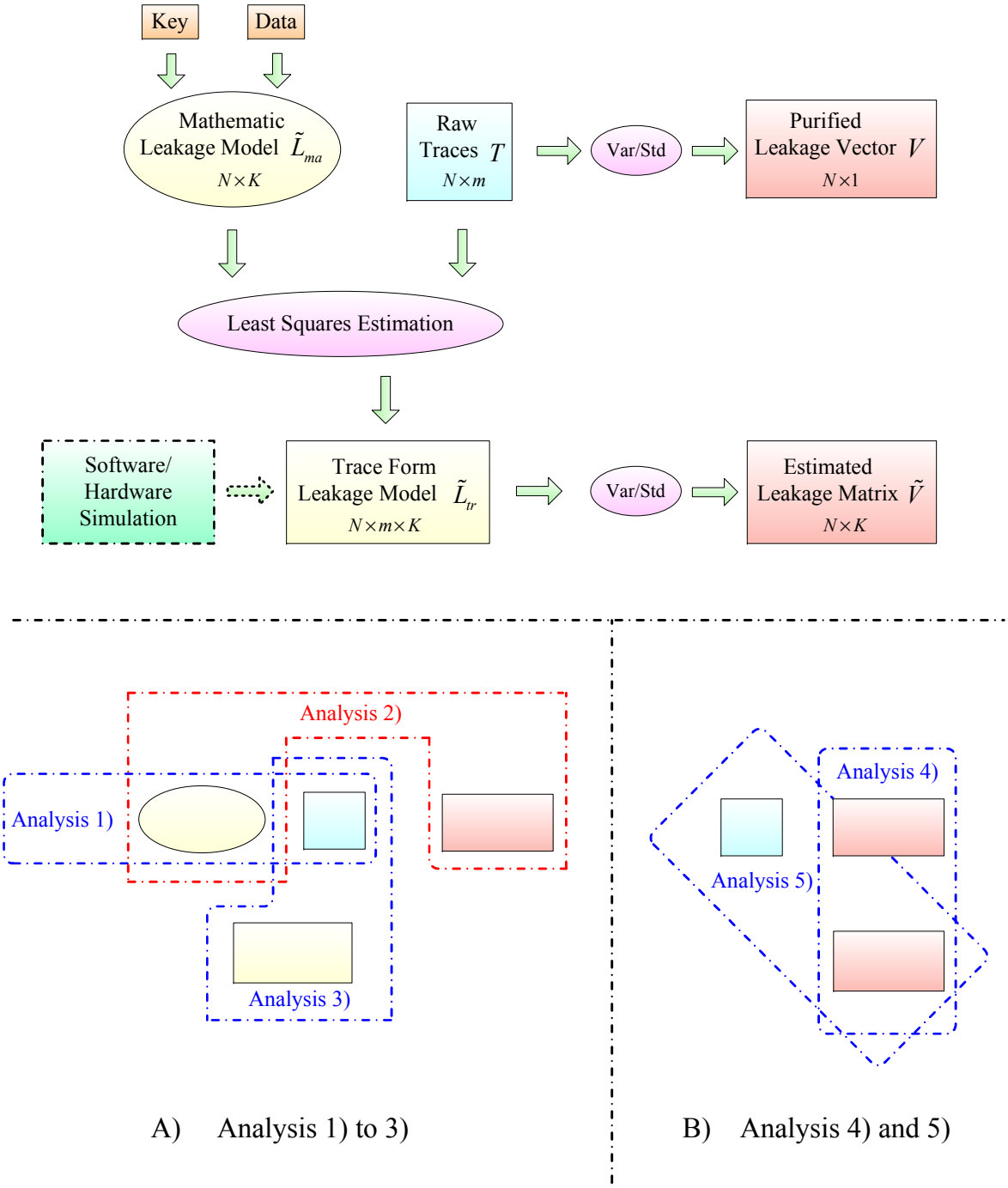

Figure 5.2: Attack Framework

components precisely. Therefore, it is worth doing that for giving in return the higher attack performance.

5. In practice, the purified leakage matrix $\widetilde{V}$ and the captured power trace set T can also be analyzed by exploiting correlation coefficient and mutual information, as presented in Fig. 5.2 Analysis 5). The attack results are achieved as $R_{i,j} = D(T_{1:N,j}, \widetilde{V}_{1:N,i})$, where $j \in [1, m]$ and $i \in [1, K]$ hold. This attack framework is called Power Amount Analysis mutation II (PAA-II). Theoretically, the results from the attack are not satisfactory. That is because the calculation of estimated leakage matrix $\widetilde{V}$ is based on the AWGN model, which considers the time interval for the analyzed data. On the contrary, the way to calculate the results R concentrates on the time instant. Both parties do not well match with each other.

Both attack methods PAA-I and PAA-II are the derivatives of original PAA attack, that is because all these attack methods share a common principle to purify and abstract the leakage information by exploiting a large number of time points in the analyzed data. Therefore, for attack methods PAA-I and PAA-II, they inherit at least two good properties from PAA attack, i.e., stronger misalignment tolerance and amplitude fluctuation invariance. With regard to the items of run time and trace usage: the former needs more time to compute the complicate trace form leakage model; the latter requires some verifications in real attacks in the future.

The proposed attack framework supplies more choices to build the leakage model and to select the analysis methods. One cannot definitely conclude which analysis combination in the framework performs the best attack results. However, we take such a framework as a tool to help the researchers to evaluate the system security in a selective manner.

Chapter 6

Pre-processing of Misaligned Power Trace

Contents

6.1 General Description

Misalignment injection is an effective countermeasure to neutralize power attacks in practice. In order to improve the attack performance during the attacking of misaligned power traces, some pre-processing must be done in advance. For example, in [MOP07, pp. 205-212], the authors proposed several methods, e.g., pattern matching, convolution, and fast Fourier transformation, etc., to deal with the misaligned power traces. Later, van Woudenberg et al. [WWB11] discussed an elastic alignment approach to deal with the

misaligned power traces and to eventually improve the DPA attack by exploiting the Dynamic Time Warping (DTW) algorithm. The suggested algorithm computes the warp path between the reference and the targeted traces, which is a well-known method for aligning utterances in the sound processing field. All these mentioned methods are feasible and useful. However, when trying to attack the misaligned power traces yielded from a random clock armed cryptosystem in practice, the following problems may appear, which must be solved in advance.

First, when dealing with the misaligned power traces, only a small portion in the analysis region of power trace contains the information leakage, which the adversaries focus on. Therefore, the alignment of the whole trace with more efforts investment is unwise. However, how to first efficiently locate the region of interest in power traces for the pattern matching step is a key issue. For instance, an AES encryption curve features the round power peaks in the monitor device. The shapes for the different round peaks look similar. We cannot directly align the peaks without distinguishing the targeted one. When the correlation is exploited to align the power traces, one can find the best matches for the beforehand template in the first, second, or other round peaks by visual observation easily. However, usually, thousands or millions of power traces are captured, and the mentioned mission cannot be executed manually. How to identify the aimed round peak and locate them automatically is a tough job. The authors did not mention it in the previous literatures. Consequently, we proposed a way to align the misaligned power traces partially and dynamically in the time domain in [TSSH12] and [TH12a], where the targeted power peaks are identified firstly according to the building of the leakage model, then the pattern matching is mounted accordingly. It is called Horizontal Alignment (HA).

Second, during the further investigation, one finds that when the base clock frequency is increased in a random clock armed cryptosystem, the power traces present a phenomenon that the power peak positions not only shift in the time domain, but also the power values change in the amplitude domain. We named such a phenomenon as clock frequency effects in [TH12b]. These effects become stronger with a higher base clock frequency. Under that condition, only running the horizontal alignment is not sufficient to improve the attack performance. In other words, an extra trace pre-processing is required to neutralize the clock frequency effects. Therefore, we proposed another method Vertical Matching (VM) operated in the amplitude domain, as detailed in [TH12b]. The running of VM algorithm to the horizontally aligned power traces results in a considerable improvement of attack performance.

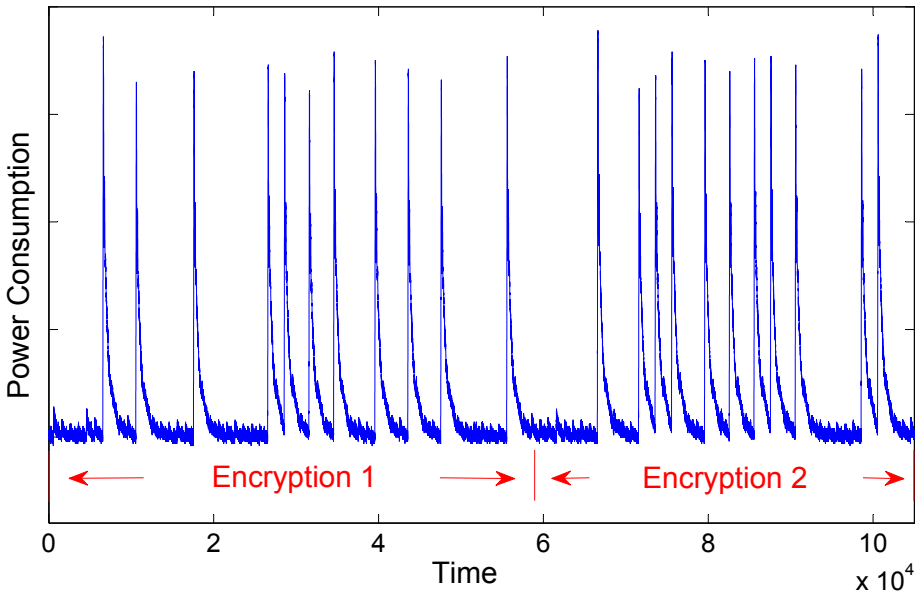

Figure 6.1: Power Trace Behaviors in Random Clock Featured Cryptosystem

In this chapter, the mechanism of the misalignment injection is generalized first, then the horizontal alignment and the vertical matching methods mentioned before are discussed in the sequel, respectively. Finally, a software architecture is suggested to process the misaligned power traces efficiently in practice.

6.2 Misalinged Power Trace Capture

A measurement setup for a random clock armed cryptosystem is illustrated in Fig. 3.5, Chapter 3. When the input is selected as a random clock via the multiplexer, then the misaligned power traces are yielded during the running encryption or decryption, as shown in Fig. 6.1, where the base clock frequency is running at 2 MHz. One finds that the power peaks of the counterparts shift in the encryption 1 and 2 in the time domain randomly. As a result, mounting attacks directly on those misaligned power traces immediately leads to lower success rate.

Obviously, before mounting attacks, the power traces have to be monitored and captured correctly. There exists some technique phenomenons when monitoring a running cryptosystem feeding with a random clock. Those phenomenons are classified by us as left and right elastic features, respectively, as described in the following:

Left Elastic Feature (LEF): The power trace, recoded from the end of the trigger signal during the encryption or decryption, features a static end within the vision of the oscilloscope. Accordingly, the left part of the power trace in the valid trigger range

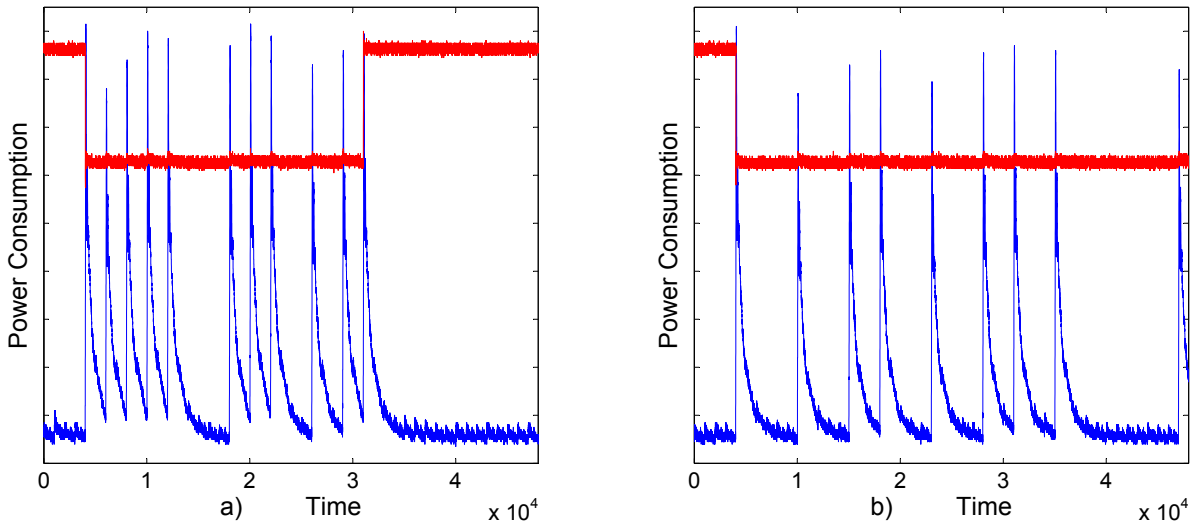

Figure 6.2: Power Trace Capture With REF

demonstrates a length variation.

Right Elastic Feature (REF): The power trace, recoded from the beginning of the trigger signal during the encryption or decryption, features a static beginning within the vision of the oscilloscope. Accordingly, the right part of the power trace in the valid trigger range demonstrates a length variation.

Both technique cases rely on the settings of the monitor device, i.e., oscilloscope. For instance, in case of the trigger signal is set as low valid and falling edge, the power traces being captured feature REF characteristic, otherwise they feature LEF characteristic. Fig. 6.2 a) shows a REF featured power trace with low valid trigger signal. The oscilloscope is set to record the power traces when the falling edge of the trigger signal comes, i.e., at the beginning of the cryptographic operations. At the position of falling edge, the power trace has a fixed start and the right part of the power trace presents a variable length. If the adversary focuses the last round peak during the attack, capturing power traces in REF manner results in a large probability for the last round peak being squeezed out from the vision of the oscilloscope, as illustrated in Fig. 6.2 b). In other words, a lot of invalid power traces without the last round peak are captured. Accordingly, a large amount of the time must be invested to separate the valid power traces from the mixture ones, which considerably reduces the efficiency of the trace pre-processing and enlarges the efforts input during the attack. As a result, in order to reduce the occurrence probability of the invalid power trace being captured, LEF manner is a better way to capture the power traces focusing on the last round peak. In a word, the trace capture style directly impacts the efficiency of the trace pre-processing step.

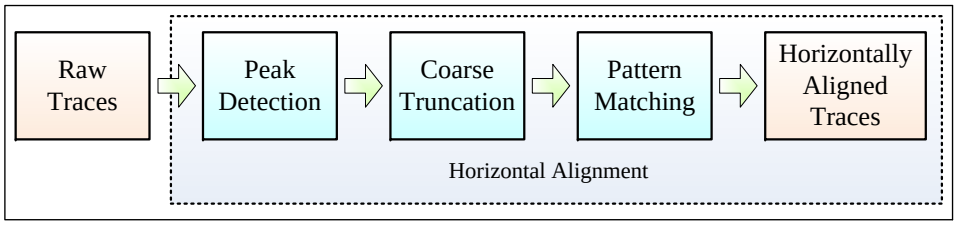

Figure 6.3: Workflow of Horizontal Alignment

Algorithm 1 Threshold Based Peak Location

Require: $T_{i,1:W}$; P^*; W^*
Ensure: t^*
1: Create empty boundary matrices D and C, with size $(:, 2)$.
2: Run Boundary Domain Check Algorithm 2.
3: Compress D into C by exploiting Algorithm 3.
4: Find maximum value (t^*, P_{max}), where $t^* \in [C_{q,1}, C_{q,2}]$.
5: **return** t^*.

6.3 Horizontal Alignment

6.3.1 Workflow of Horizontal Alignment

The workflow of horizontal alignment is divided into three parts, which are peak detection, coarse truncation, and pattern matching, as shown in Fig. 6.3. In the peak position detection phase, the position of the aimed power peak containing the information leakage in the analysis region will be located dynamically. Then, a portion of the power trace around the peak position is truncated, which is the basic idea of coarse truncation. In the pattern matching phase, the cut trace will be matched with the beforehand prepared template by exploiting correlation coefficient, Euclidean distance, or convolution, etc. Only a small portion of the power traces has been truncated resulting in a fast computation in the pattern matching phase. For the whole algorithm, the peak position detection plays an important role, which is the main difference between our methods proposed in [TSSH12] and [TH12a]. Both algorithms concentrate on the threshold and slope based peak position detection algorithms, respectively, which are introduced in the upcoming section.

6.3.2 Threshold Based Peak Detection

As the name suggests, in the threshold based peak position detection algorithm, a threshold power value P^* is required to separate the peak parts in each trace of set T, as shown in Fig. 6.4. Subsequently, another threshold value W^* in the time domain

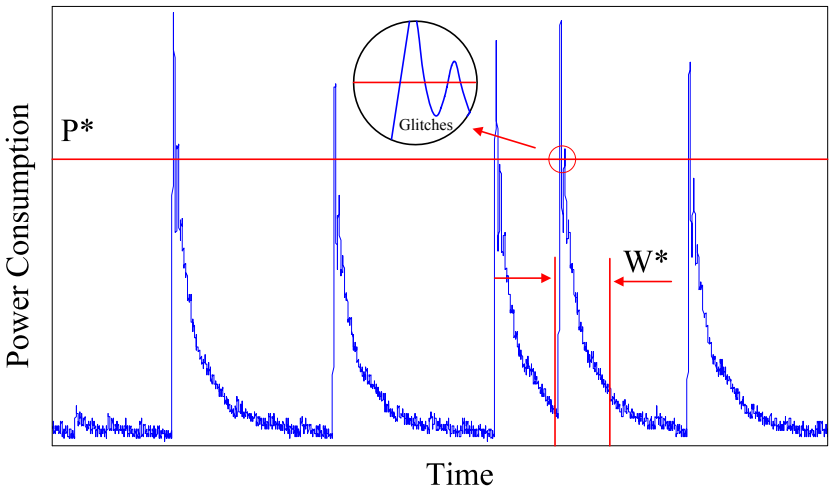

Figure 6.4: Power Value Thresholds

is exploited to distinguish the round peaks from the glitches, which are caused by the random noise in and outside the hardware circuits, as illustrated in Fig. 6.4. Arrays D and C are utilized to store the domains of filtered isolated peaks, both matrices feature uncertainty rows and two constant columns, i.e., $(:, 1:2)$. Because the numbers of isolated peaks in each trace are uncertain, the number of rows in D and C is dynamic accordingly, which increases relying on the specific number of the peaks in each analyzed power trace. Here, $D_{:,1}$ and $C_{:,1}$ denote the left boundary of the domains, whereas $D_{:,2}$ and $C_{:,2}$ store the right ones.

Algorithm 1 presents how to determine the positions of randomly shifted power peaks dynamically. In each trace, the power values of the isolated power peaks, who are larger than the threshold P^*, are determined. Hereafter, the boundaries of the isolated peaks are saved into the boundary matrix D, as shown in Algorithm 2. Then these boundaries are compressed by Algorithm 3 resulting in the compressed boundary matrix C. And the time point corresponding to the maximum power value in boundary $[C_{q,1}, C_{q,2}]$ indicates the target peak position, where q is the peak index the adversaries are focusing on.

Theoretically, the power value P^* must be smaller than and closer to P_{min}, where P_{min} denotes the minimum power peak height in all the power traces. In practice, the threshold value P^* can be easily determined by monitoring the working oscilloscope dynamically during the trace capture phase.

The existence of glitches is highlighted in Fig. 6.4. In order to distinguish the domains caused by the glitches, a domain threshold value W^* in the time domain is required in Algorithm 3, by which the domain D is compressed to matrix C resulting in the one to one correspondence between the number of domains in matrix C and the targeted round

Algorithm 2 Boundary Domain Check

Require: $T_{i,1:W}$, P^*
Ensure: D
1: Search each point in $T_{i,1:W}$
2: **if** $T_{i,1} > P^*$ **then** $D_{1,1} = 1$
3: **if** $T_{i,j} > P^*$ and $T_{i,j-1} \leq P^*$ **then** $D_{:,1} = j$
4: **if** $T_{i,j} \leq P^*$ and $T_{i,j-1} > P^*$ **then** $D_{:,2} = j - 1$
5: **if** $T_{i,W} > P^*$ **then** $D_{end,2} = W$
6: **return** D.

Algorithm 3 Merge Function

Require: Domain Matrix D, W^*
Ensure: Compressed Domain Matrix C
1: Search adjacent domains $[D_{i,1}, D_{i,2}]$ and $[D_{i+1,1}, D_{i+1,2}]$
2: **if** $D_{i+1,1} - D_{i,2} < W^*$ **then** combine two domains into $[D_{i,1}, D_{i+1,2}]$
3: Store combined domain into C
4: **Return** C
Example:
Input: $D = [19, 25], [28, 31], [45, 53]; W^* = 5$
Output: $C = [19, 31], [45, 53]$

peaks. Therefore, given the index number, the peak position can be eventually located by finding the maximum power values within the correspondent domain in matrix C, which indicates the targeted peak position in the power trace. In practice, the threshold value W^* in the time domain can be assigned equal to the width of the round peak, as shown in Fig. 6.4.

After the detection of targeted peak position, the trace will be coarsely extracted from position $t^* - O_l$ with the width W_c. Here, O_l denotes the offset value in time. After that, the pattern matching between the template and the extracted power trace can be mounted.

6.3.3 Slope Based Peak Detection

We study the threshold based peak detection in Algorithm 1 mentioned before with the misaligned power traces running at 2 MHz in [TSSH12], i.e., relatively low base clock frequency. The performance of such an algorithm is efficient. However, when the fed clock frequency is increased, the clock frequency effects take place resulting in the variation of power peaks fiercely. Finding a proper threshold power value P^* to filter out all the isolated power peaks in algorithm 2 is hindered. For instance, in Fig. 6.5 a), the threshold value can be found easily, when the power traces are running at 2 MHz base

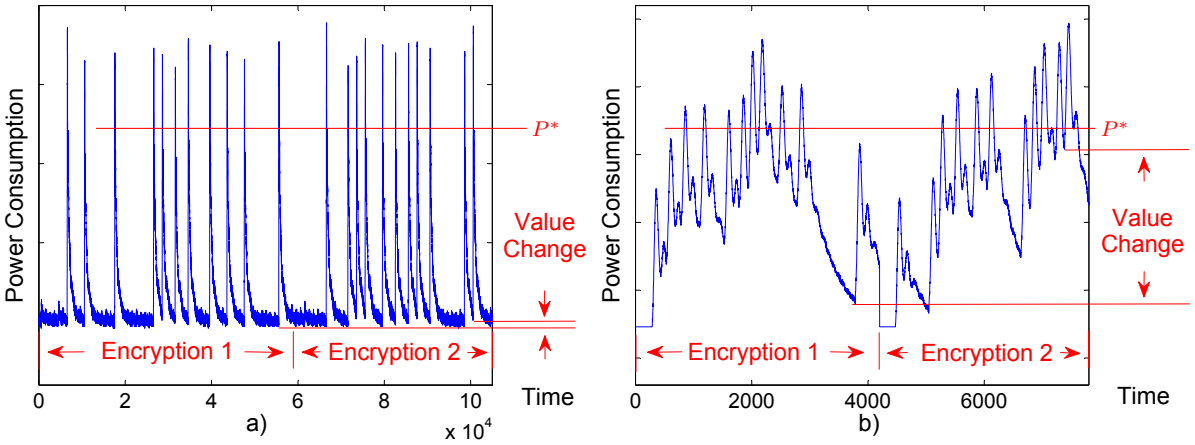

Figure 6.5: Trace Behaviors Running at 2 and 24 MHz Base Frequencies

clock frequency. Whereas, in case of 24 MHz base clock frequency, the threshold determination becomes difficult, as illustrates in Fig. 6.5 b). In other words, the threshold based peak detection algorithm is disarmed when facing with the power traces running at a higher base clock frequency. In order to overcome such a drawback in Algorithm 1, we proposed a slope based peak position detection algorithm in [TH12a]. The general idea is to analyze the slope tendency between any two points in the analyzed power trace without considering the fed base clock frequency in the cryptosystem.

6.3.3.1 Slope Analysis

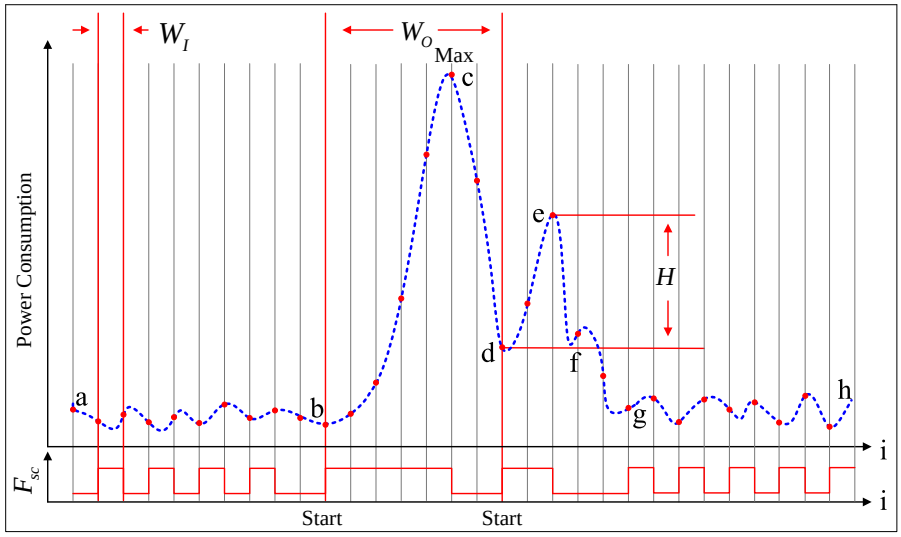

Figure 6.6: Generic Slope of Power Curve

Some basic definitions are given first for understanding and describing the algorithms

easily in the sequel.

Fig. 6.6 illustrates a captured power trace curve ah. The curve bg is assumed as a round peak containing the information leakage. The curves ab and gh behave as the electronic noise. One finds that there exist two obvious peaks in curve bg, i.e., the curves bd and dg with summits c and e, respectively. The peak position of the bigger curve bd is the adversary's target. Please note that by observing a large number of power traces captured from FPGA-based AES implementations, the curve bg is a basic trace element for the most of the captured power traces. Without loss of generality, this curve is taken as our example. Now several definitions are given for the future using.

Slope Characteristic (SC): It depicts the tendency of slope k for a line crosses through two certain points within the interval W_I in the analyzed trace segment. However, the adversaries concentrate the positive-negative characteristic of slope k rather than its exact value.

1. If $k > 0$, then the SC property for these two points and the points in between features positive. The flag F_{sc} is assigned as 1, i.e., $F_{sc} = 1$ holds.

2. If $k \leq 0$, then the SC property for these two points and the points in between features negative. The flag F_{sc} is assigned as 0, i.e., $F_{sc} = 0$ holds.

Fig. 6.6 illustrates an analyzed power trace curve with its flag F_{sc} shown underneath.

Peak Start Position: As the name suggests, it defines the start position of the analyzed power peak. For example, the start position of the bigger and smaller peaks are the points b and d, respectively, as shown in Fig. 6.6.

Peak Height: It measures the height of a peak by calculating the power value difference between the peak start position and its summit. As illustrated in Fig. 6.6, the peak height H for the peak def derives from the power value difference between the start point d and the summit e.

The values in F_{sc} can predict the rising and falling tendency of the analyzed power trace as follows:

1. For the curve segments ab and gh, the flag F_{sc} shows the values 0 and 1 in alternation.

2. For the power peak, the F_{sc} features long continuous value 1 or 0 for the rising and falling parts, respectively. For example, for the rising curve bc, $F_{sc} = 1$ holds. In contrast, for the falling curve cd, $F_{sc} = 0$ holds.

Algorithm 4 SC Calculation

Require: $T_{i,1:W}, W_I$
Ensure: F_{sc}
  Initialize $F_{sc(j)} = 0$, $j \in [1, W]$
  for $j = 1$ to W **do**
    Determine F_{sc} for $T_{i,j}$ within interval W_I
  end for
  Return F_{sc};

Consequently, by extracting the information from the values in flag vector F_{sc}, one can eventually figure out the peak position for the targeted round peak. The whole analysis scenario is discussed in the sequel.

6.3.3.2 General Peak Detection Algorithm

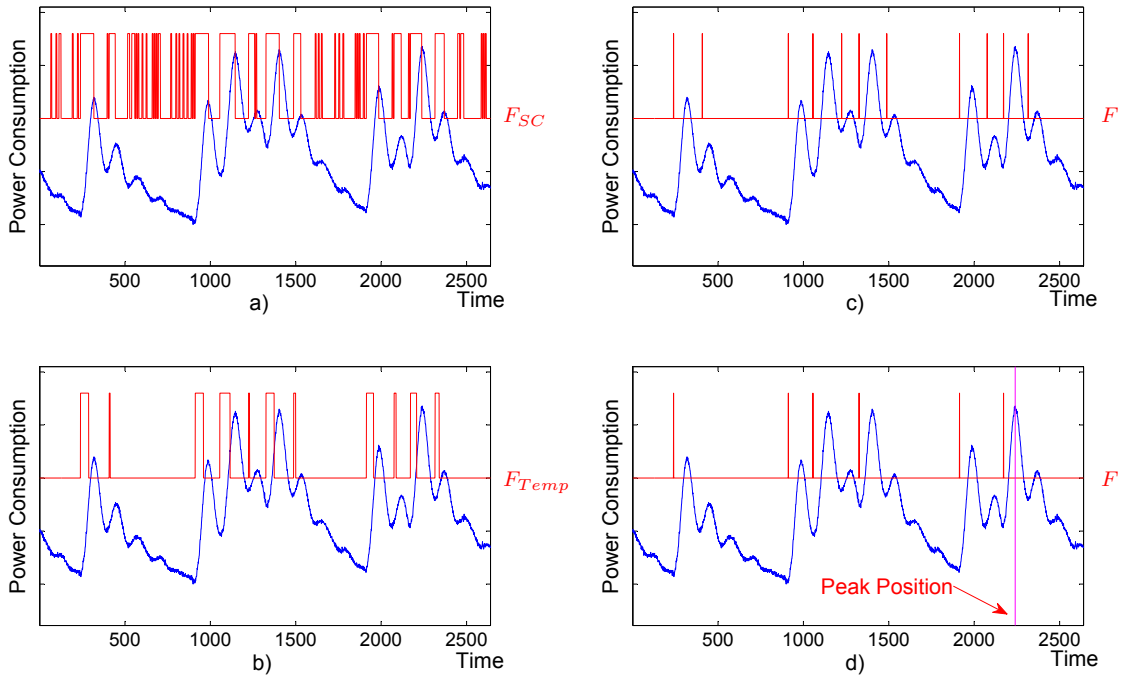

Figure 6.7: Calculation Steps for Slope Based Peak Detection

For each misaligned power trace $T_{i,1:W}$, Algorithm 4 is mounted resulting in flag vector F_{sc}, as outlined in Fig. 6.7 a) with an analyzed power trace presenting as a background. In the rising and falling region, the F_{sc} value features long continuous 1 and 0, respec-

Algorithm 5 Slope Analysis

Require: F_{sc}, W_c
Ensure: F
  Initialize $F_{temp(i)} = 0$, $F_i = 0$, $i \in [1, W - W_c]$
  Fluctuation Filtering:
  for $j = 1$ to $W - W_c$ **do**
    if $F_{SC(j:j+W_c)} = 1$ **then** $F_{temp(j)} = 1$ **end if**
  end for
  Flag Simplify:
  for $k = 1$ to $W - W_c$ **do**
    if $k == 1$ **then**
      if $F_{temp(1)} == 1$ **then** $F_1 = 1$ **end if**
    else
      if $F_{temp(k-1)} == 0$ and $F_{temp(k)} == 1$ **then** $F_k = 1$ **end if**
    end if
  end for
  Return F;

tively. In contrast, in the fluctuation region, the flag vector F_{sc} shows the values 1 and 0 varying alternatively. Please note that the time point interval W_I must be chosen properly, i.e., neither too large nor too small. For example, there exists a vector A=[6, 8, 9, 12, 5, 8, 19, 25, 16, 10, 4, 5], if we choose the points A_1 and A_{12} to establish a line, then the $F_{sc} = 0$ holds, i.e., $W_I = 10$, all the time points $A_{2:11}$ feature the same property. One finds that the points in $A_{5:11}$ form a small peak, however, $F_{sc} = 0$ holds, i.e., the peak from point A_5 to point A_{11} is taken as the noise and filtered out by the algorithm. If we want to keep such a peak, then the value W_I may be chosen smaller. In other words, the choosing of interval value W_I relies on the size of the analyzed peak. Thus, by adjusting the interval W_I, the rising and falling sections of the round peaks can be discerned from the glitches and noise. Later, by analyzing the values in the flag vector F_{sc}, the targeted peak position can be clearly identified for the subsequent algorithms.

Algorithm 5 is exploited to further analyze the flag F_{sc} and to distinguish the targeted peak from the noise resulting in the finding of peak start position in the analyzed power trace. The whole process contains two main steps, i.e., *fluctuation filtering* and *flag simplifying*. In the first step, values 0 and 1 fluctuation parts in F_{sc} will be assigned to 0 after the checking of the value 1 appearing W_c times continuously. In other words, there exists an empty vector F_{temp} with the number of $W - W_c$ zero elements. If any element and the number of W_c elements following are equal to 1 in the flag vector $F_{sc(1:W-W_c)}$, then the element at the same position in F_{temp} is assigned to 1. After this process, the 0, 1 fluctuation parts disappear, and the rests are the regions with long continuous value

Algorithm 6 Peak Position Detection

Require: $T_{i,W}$, F, W_o, P_h
Ensure: t^*
  for $j = 1$ to $W - W_c$ **do**
    if $F_j = 1$ **then** search domain $[j, j + W_o]$ in $T_{i,1:W}$ and calculate the peak height H
    if $H \leq P_h$ **then** assign $F_j = 0$
  end for
  for $t = 1$ to $W - W_c$ **do**
    if $F_t = 1$ **then** detect maximum value (t^*, P_{max}) in the domain $[t, t + W_o]$ of $T_{i,1:W}$
  end for
  Returnt^*

1, which point to the rising parts of each peak, as illustrated in Fig. 6.7 b). The start position of each region with continuous value 1 is the start position of the peak. In the flag simplifying phase, each element in the flag vector F_{temp} is checked. The transition points featuring value changing from 0 to 1 are the peak start positions for the analyzed round peaks, which are assigned to the value 1 in flag vector F, as illustrated in Fig. 6.7 c). One finds now that each flag in F points to a peak start position. However, there exist two obvious peaks, i.e., bigger and smaller ones, and the start position for the former one is focused. Therefore, the identification between the bigger and smaller peak is necessary, which is discussed in the sequel.

In order to distinguish the peaks with different sizes, i.e., the peak height threshold P_h is introduced into Algorithm 6, where W_o denotes the peak start offset. The indicator elements with value 1 in F point to the start positions of each peak in the analyzed power trace. The peak height value H can be calculated for each power peak, then, such a value is compared with the peak height threshold P_h resulting in the separation of bigger and smaller power peaks. At the same time, the indicators, which point to the smaller peaks are removed, i.e., assigned to the value 0, which is visible in Fig. 6.7 d). After that, all the indicators in the flag vector F are in an one to one correspondence with all the targeted peaks. Then the peak positions can be located by searching the biggest pair (t^*, P_{max}) from the peak start position with the offset W_o, where t^* denotes the peak position in the time domain. Then according to the index of the peaks, the adversaries can get the peak position correctly.

The proposed algorithm embodies more parameters than the threshold based algorithm in 1. Consequently, before mounting the algorithm, a trace training is required to get the appropriate parameter settings. As long as one can find the proper parameters, the correct peak position can be detected successfully without considering the fed base clock frequency. Therefore, a visualized software is needed to train the traces and to

determine these parameters properly.

When running the suggested algorithms at hand, if the correct peak position cannot be found, two possible cases may appear. Firstly, the parameters may not be correctly selected; secondly, the analyzed power trace is invalid and not being exploited for the further analysis. Therefore, the analyzed power trace and its counterpart, e.g., input/output (plaintext and ciphertext), must be discarded.

6.4 Analysis GUI

In order to train the power traces in advance to find the proper parameters in the slope based peak detection algorithm and visualize the attack results, a Graphical User Interface (GUI) is created by Matlab. The GUI embodies several basic functions and algorithms mentioned before, which is called *traces view and parameters setting*, as shown in Fig. 6.8. There are two functions in such a GUI. Firstly, it is utilized to visualize the analyzed data or do some basic analysis for the attack results, i.e., maximum values, positions, etc.; secondly, it is exploited to train a certain amount of the analyzed power traces for the parameters adapting in the slope based peak detection algorithm. However, the latter is predominated.

6.4.1 Data Analysis

The basic function for this GUI is to analyze the selected data in the variable space. First, the analyzed variable "traces" is selected from the popup menu in the *Analyzed Traces Information* panel. Then the different information of the being analyzed data are shown in the same panel, e.g., the variable "traces" contains 100 power traces. Each trace embodies 4,200 time points. Besides that, the maximum and minimum values, positions, variance, standard deviation are shown in the *Single Trace Information* panel. The extreme values and its positions are presented in the *Global Information* panel, accordingly. Here, one can decide only one or all traces in this variable is shown in the figure by selecting the radios in panel *Traces Showing*. Additionally, the mean, variance, standard deviation or 3D curves for all the traces can also be drawn by selecting the buttons in panel *Extra Drawing*. For example, in Fig. 6.9, all the traces are shown with gray color except the being analyzed one. At the same time, the mean and standard deviation curves for all these traces can be presented, respectively. In Fig. 6.10, all the traces are drawn with the 3D view.

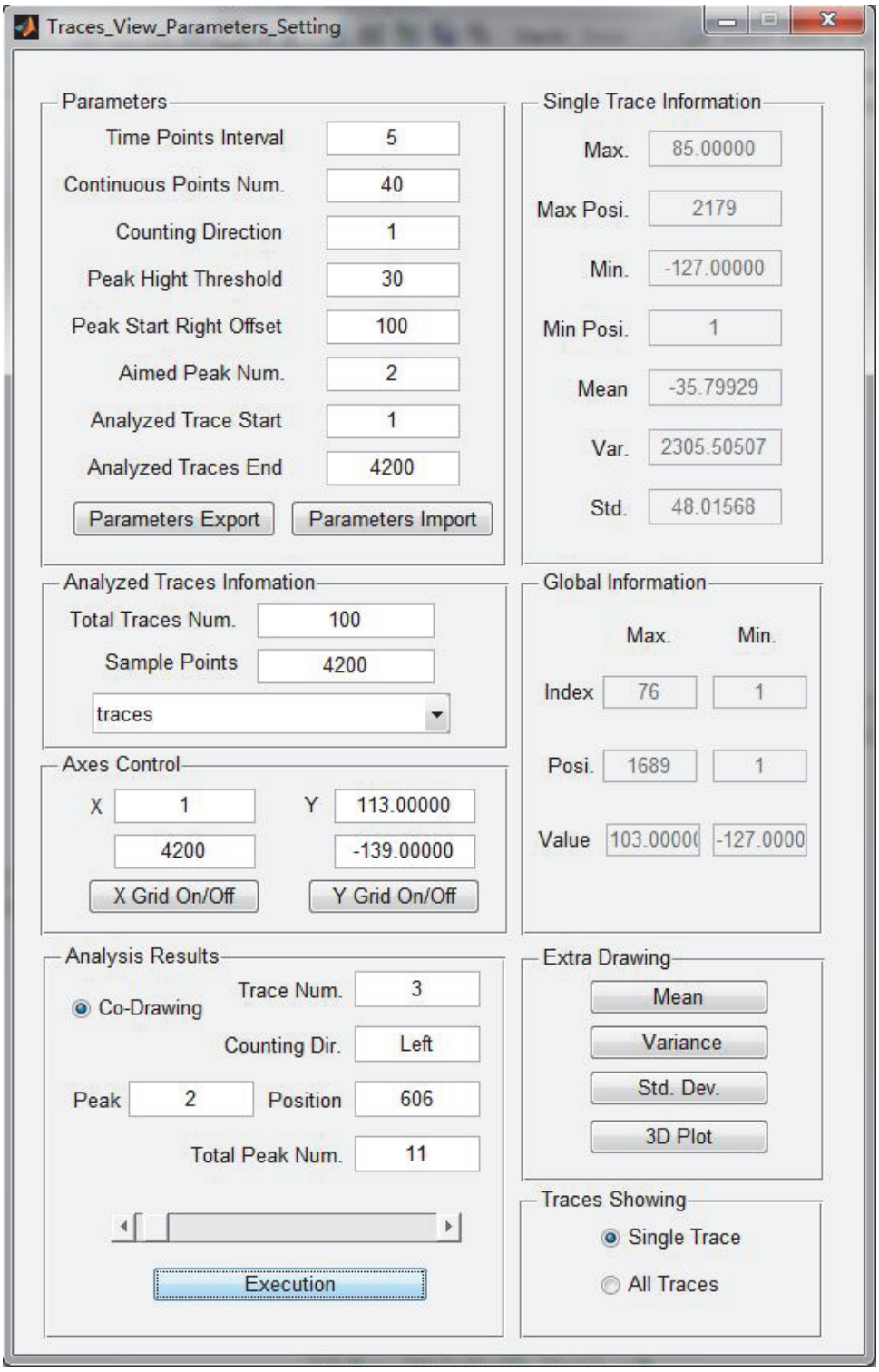

Figure 6.8: Parameter Setting GUI

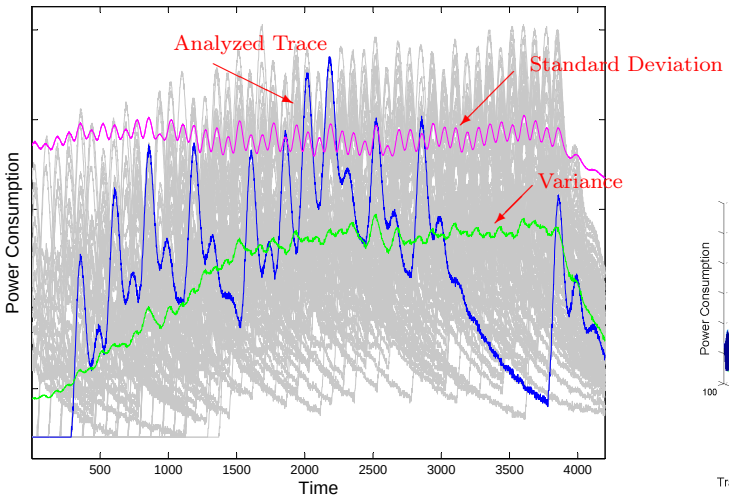

Figure 6.9: Trace View

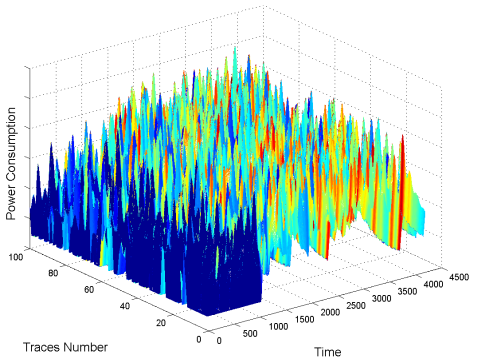

Figure 6.10: Trace Set 3D Show

6.4.2 Parameter Adapting

The main function for this GUI is to adapt the parameters in the slope based peak detection algorithm. In the *Parameters* panel, the parameters, for example, time point interval W_I, continuous point number W_c, peak height threshold P_h, peak start offset W_o, are set in advance for the testing. The start and the end time points of the analyzed traces can also be adjusted, as well as the counting order, i.e., counting from left is 1 and from right is 0; the aimed peak number defines the index of the peak which the adversary wants to locate. After the setting of the parameters, one can press the *Execution* button in panel *Analysis Results* to run the proposed analysis methods. At the same time, a visualized panel shows the progress of peak detection step by step, as shown in Fig. 6.7. The values for determining a set of parameters properly can be changed by training a certain amount of the power traces. All these finely chosen parameters can be exported as a text file for the subsequent using in real attacks, which can also be imported from the previously stored text file.

Take the variable "traces" again as an example. The third trace is focused, where the second peak counting from left side is taken as the aimed peak. The whole trace contains 11 power peaks, and the targeted peak position is at the time point 606. Then the GUI is run for the traces training. The counting direction and the aimed peak number are set as 1 and 2, respectively, where $W_c = 40$, $P_h = 30$, and $W_o = 100$ hold. In order to help the readers to have a further impression about the W_I selection, the parameter W_I is changed with three values, i.e., 2, 5, 70. One finds that only in Fig. 6.12, all the power peaks can be recognized correctly, where $W_I = 5$ holds. With the smaller time point

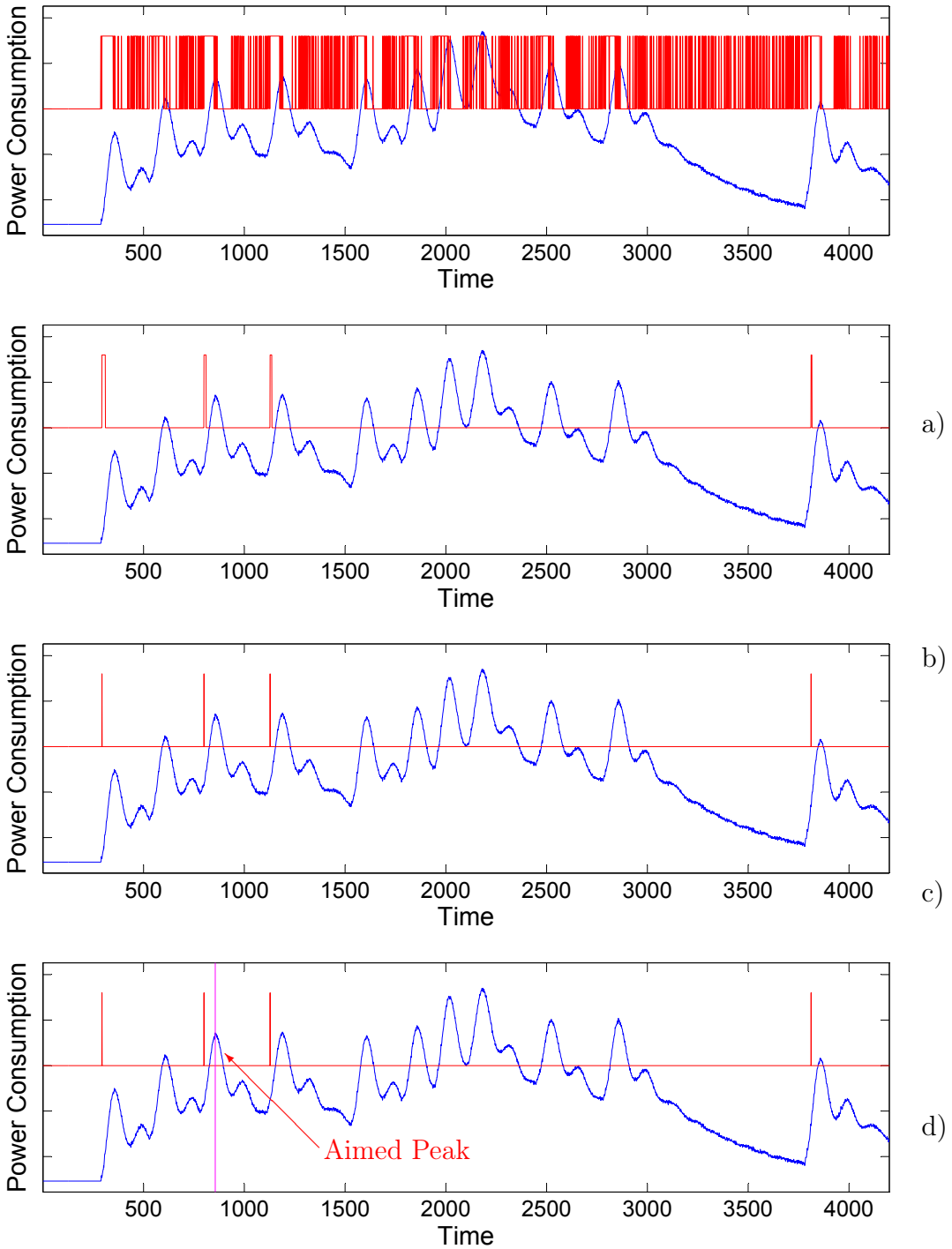

Figure 6.11: Trace Training: $W_I = 2$

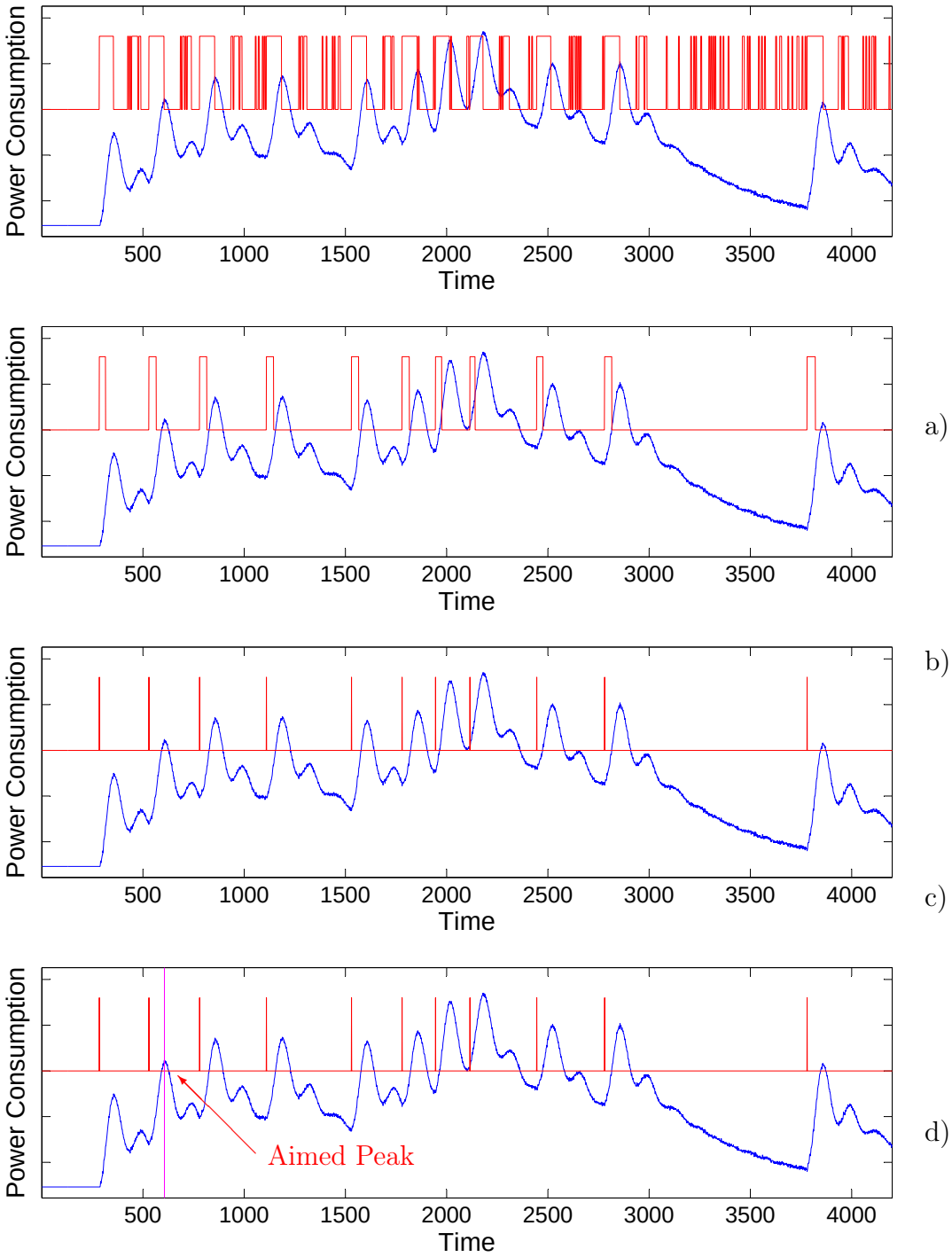

Figure 6.12: Trace Training: $W_I = 5$

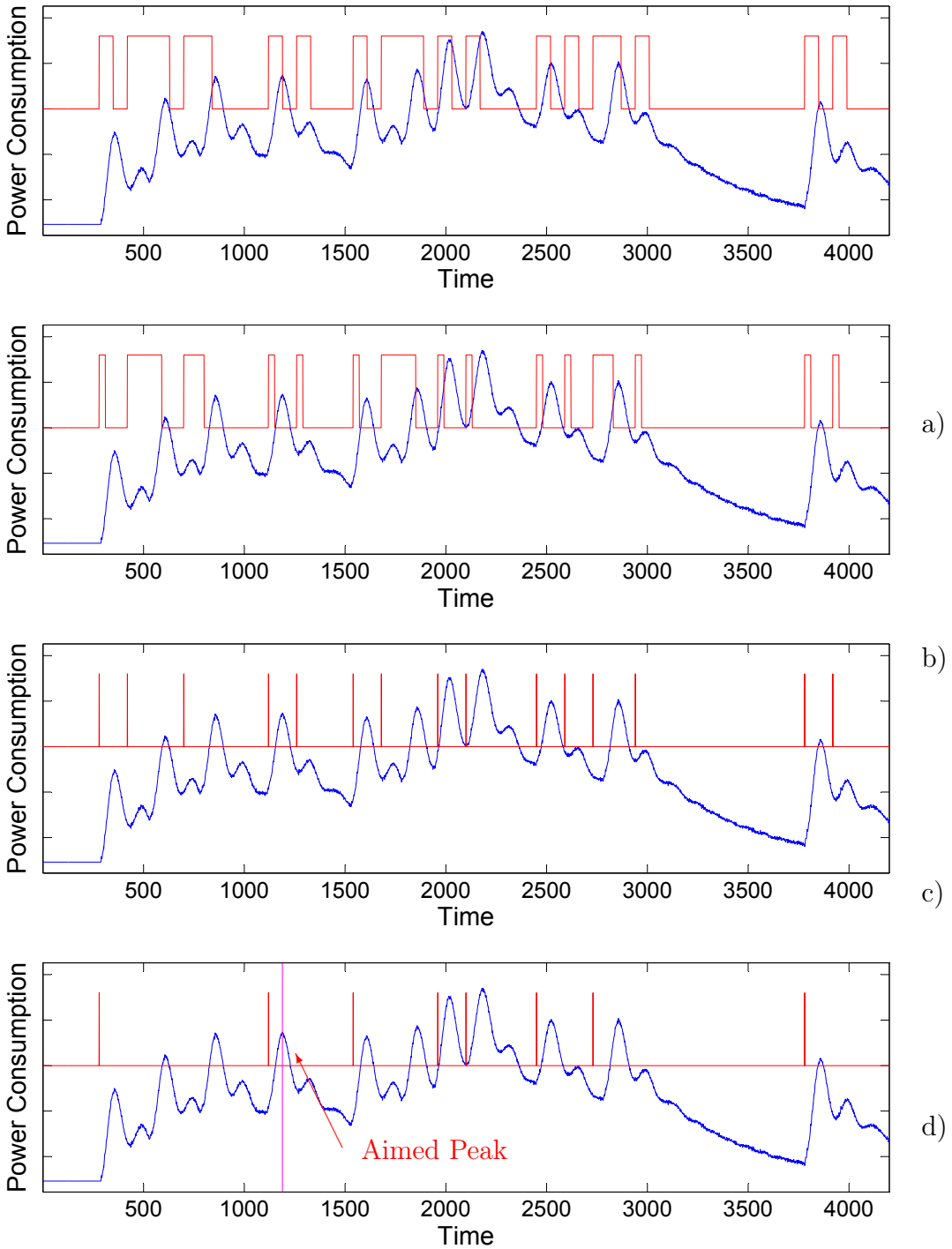

Figure 6.13: Trace Training: $W_I = 70$

interval, e.g., $W_I = 2$, the targeted peak position is lactated by mistake at the position 856, as shown in Fig. 6.11; with the larger time point interval, e.g., $W_I = 70$, the aimed peak position is recognized incorrectly at the time point $1,189$, as illustrate Fig. 6.13. Both parameters $W_I = 2$ and $W_I = 70$ are set either too small or too large, respectively, which are unwise choices for the peak detection. Consequently, after the adjusting of the parameters for training a certain amount of the power traces, a set of parameters for the analyzed power traces can be eventually found. Then the trace pre-processing can be run.

6.5 Clock Frequency Effects

The main portion of power consumption in the CMOS circuits is the dynamic power, which can be estimated by (2.3). Theoretically, the value of capacitance C is fixed when the cryptosystem and all its peripherals are deployed; it is no doubt that voltage regulator can supply the stable power, i.e., V_{DD} is a constant; the switching activity α is a statistical parameter, which, to some extent, may be taken as a constant as well. Only the on system clock frequency f is a random variable in a random clock featured cryptosystem. Therefore, the random clock frequencies have an influence on the variation of dynamic power consumption in the physical cryptosystem directly, which are presented in the captured power traces with the power values of round peaks shifting in the amplitude domain, i.e., clock frequency effects.

For better understanding of the clock frequency effects, assuming in a CMOS circuit, a capacitance charging and discharging requires the time 0.05s and 0.05s, respectively, i.e., the frequency for such a process is 10 Hz. When a cryptosystem is running at 7 Hz fixed clock frequency, for each valid clock, the capacitance can charge and discharge completely. Accordingly, the power traces feature the behavior with stable power peaks. If the clock frequency is increased to 15 Hz, the charging and discharging for the capacitances become incomplete. However, the power peaks are still stable. If the capacitances are fed with a random clock running at 15 Hz base clock frequency, then one finds that with the clock slower than 10 Hz the capacitances can charge and discharge completely. On the contrary, the incomplete and the different amount of power dissipation arisen from the charging and discharging take place when the fed clock is higher than 10 Hz. Both cases are mixed randomly yielding the complete and incomplete charging and discharging for the capacitances alternatively. For example, there exist two types of clock frequencies, i.e., 6 Hz and 15 Hz. Under the former situation, the capacitance dissipates 13 units

Algorithm 7 Search Area Calculation

Require: T
Ensure: V_d
1: Search each time point in each trace of the set T, find its maximum power consumption value, and store it into a vector V_f, where N defines the number of the power traces in T.
2. Find the maximum and minimum peak heights $V_{f(max)}$ and $V_{f(min)}$ in V_f.
3. Calculate $V_d = V_{f(max)} - V_{f(min)}$.
4. **Return** V_d

power; for the latter, the capacitance consumes 20 units power. When there comes a continuous clock with 6 Hz or 15 Hz, and then the capacitance dissipates 26 or 40 units power accordingly. However, if there come two clocks operated at 6 Hz after the 15 Hz, then the capacitance dissipates 33 units power. In other words, the different combinations of the adjacent clock frequencies cause different power dissipation, which behaves in the power traces as the power values changing of round peaks, i.e., clock frequency effects. These effects may be observed from the captured power traces directly. Fig. 6.5 illustrates two encryption traces running at 2 and 24 MHz base clock frequencies, respectively. Fig. 6.5 a) shows the power traces working at 2 MHz base clock frequency, where the power peaks shift only in the time domain, with a little power values variation in the amplitude domain. When the base clock frequency is increased to 24 MHz, in contrast, the power peak positions not only shift in the time domain, but also the power values vary considerably in the amplitude domain, as highlighted in Fig. 6.5 b).

6.6 Vertical Matching

In order to antagonize the clock frequency effects existing in the misaligned power traces, we proposed the vertical matching algorithm resulting in the improvement of attack performance, as detailed in [TH12b]. In this section, a brief introduction about this trace pre-processing method is given.

Without loss of generality, the maximum fluctuation V_d is determined by exploiting Algorithm 7, which defines the maximum difference between the maximum and minimum summit values of the power peaks in set T. Such a value must be calculated in advance, and it implies the search range, which is used for the subsequent algorithm.

In Algorithm 8, each trace is moved in the amplitude interval $[T_{i,1:W} - V_d, T_{i,1:W} + V_d]$. During the moving, the Euclidean distance between the analyzed power trace and the template T_t is determined and stored in a vector Dis, where the minimal value Dis_{j^*}

Algorithm 8 Vertical Matching

Require: $T_{i,1:W}$, $T_{t(1,1:W)}$,
Ensure: T_v
1: Calculate the step interval $s = (2V_d/N)$
2: Calculate $Dis_j = EuDis(T_{i,1:W} - V_d + j \cdot s, T_t)$ for each move, where $j \in [0, 2V_d/s]$
3: Find minimum distance in Dis_j, record its index j^*
4: Store the trace $T_{i,1:W} - V_d + j^* \cdot s$ into trace set T_v
5: **Return** T_v

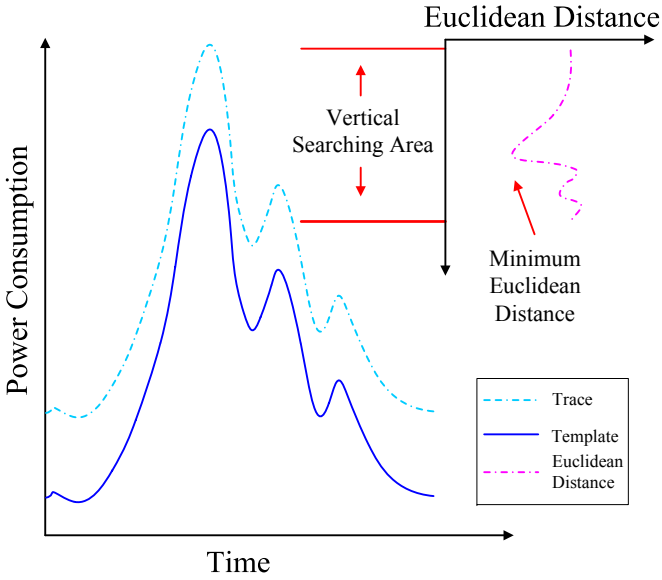

Figure 6.14: Visualization of Vertical Matching

indicates that the both analyzed parts are matched with each other very well. Then the trace with the offset $j^* \cdot s - V_d$ in the amplitude domain is stored into the vertically matched trace set T_v, as shown in Fig. 6.14.

Here is a tip, the integer power values for the captured traces are derived from the sampling and quantizing oscilloscope. Therefore, in practice, the power values are kept as integer after the process of vertical matching, i.e., the step of each moving can be set as 1 or bigger. One may also assign it smaller than 1. However, it increases the calculation time considerably without bringing the improvement of the attack performance.

By exploiting vertical matching, the efficiency of power attacks is considerably improved, which has been presented by us in [TH12b] with the evaluation metrics, i.e., total trace usage and correlation peak height.

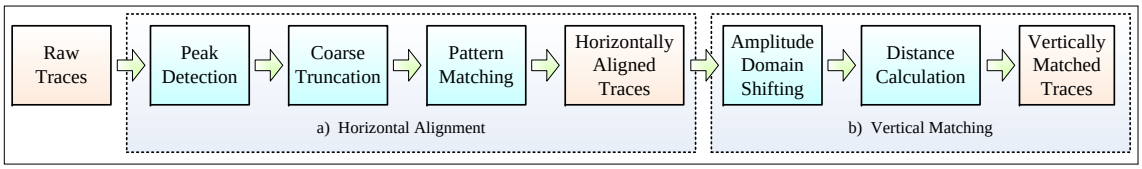

Figure 6.15: Architecture of Trace Pre-processing

6.7 Process Framework

Combining the horizontal alignment with the vertical matching method, the whole scenario of the trace pre-processing is illustrated in Fig. 6.15. Here, for the horizontal alignment, two peak detection methods can be utilized alternatively, which relies on the base clock frequency fed into the cryptosystem. Subsequently, the vertical matching may be mounted accordingly. The whole process to the misaligned power traces can considerably improve the power attacks in reality.

6.8 Software Architecture

In order to preprocess the misaligned power traces for the subsequent attacks systematically, a software architecture is proposed, as shown in Fig. 6.16.

The whole architecture is composed by several functions and sub-functions, which are detailed in the following.

Function 1. FrequencyDetermining is applied to determinate whether the intensity of the clock frequency effects are strong enough, i.e., when the base clock frequency are higher than a preset threshold value, the program will choose the proper horizontal alignment and vertical matching algorithms accordingly. Usually the threshold value can be achieved by monitoring the trace capture phase online.

Function 2. ThresholdHA is a main function to execute the horizontal alignment by finding a proper threshold to filter the isolated peaks, which embodies three sub-functions, i.e., ThresholdPeakDetect, TracesVerification, and PatternMatching.

Function 3. SlopeHA defines a main function to run the horizontal alignment by analyzing the slope of the analyzed power traces. The whole function contains three sub-functions, i.e., SlopePeakDetect, TracesVerification, and PatternMatching, which are introduced in the sequel.

Function 4. ThresholdPeakDetect implements the peak detection algorithm from exploiting the threshold value P^*, which embodies two sub-functions, i.e., PeakFiltering

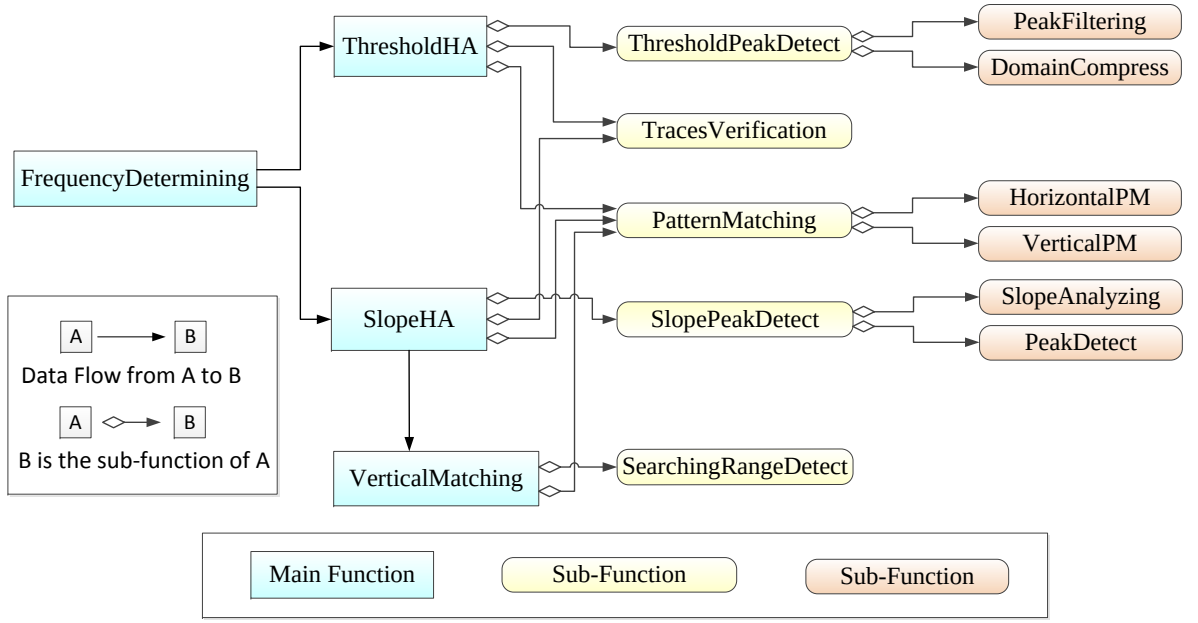

Figure 6.16: Software Architecture for Trace Processing in Matlab

and DomainCompress. Both of them execute the algorithms proposed in Algorithms 1, 2, and 3, respectively.

Function 5. SlopePeakDetect achieves the peak detection algorithm by analyzing the slope characteristics of the power traces, which contains two sub-functions, i.e., SlopeAnalyzing and PeakDetect. Both functions run the algorithms suggested in Algorithms 4, 5, and 6, respectively.

Function 6. TracesVerification is a common subfunction, which is called by the functions ThresholdHA and SlopeHA. The main contribution for such a function is to detect two abnormal cases: on the one hand, there is no peak existing in the analyzed section, it is caused by system or communication errors during the trace capture phase; on the other hand, the power trace does contain the power peaks, however, which is not the correct one the adversary is looking for. The mentioned cases occurring implies that the being analyzed power trace is useless. Therefore, this invalid power trace and its correspondent plaintext (ciphtertext) must be removed from the analyzed data. Then the programm goes on to analyze the next power trace.

Function 7. PatternMatching embodies different pattern matching algorithms, e.g., correlation coefficient, Euclidean distance, etc. According to the specific utilization, which is divided into two sub-functions, i.e., HorizontlPM and VerticalPM. The difference between these two functions is the template moving type. In the former, the template shifts from left to right in the time domain to find the best matched section in the analyzed

trace segment. Consequently, the similarity analysis algorithms, i.e., correlation coefficient, and Euclidean distance can be exploited; for the latter, the template is shifted in the amplitude domain. Therefore, only Euclidean distance can be exploited as the distinguisher. Hereafter, the matched trace section is stored into a new trace matrix, i.e., horizontally aligned or vertically matched power trace matrix for the subsequent attacks. **Function 8. SearchingRangeDetect** calculates the searching range for the template shifting in the amplitude domain in vertical matching phase, as presented in Algorithm 7.

The proposed architecture supplies more possibilities when the misaligned power traces occur. This architecture can help people to access the secret key in cryptosystems with ease and to achieve s better attack performance in practice.

Chapter 7

Combination of Attack Methods and Trace Pre-processing

Contents

7.1 Introduction

In this chapter, some attack factors are introduced first, which have the impact on the trace pre-processing as well as the attack performance. Then the attack methods and trace pre-processing approaches are combined into a framework to provide more choices for the key revealing in physical cryptosystems.

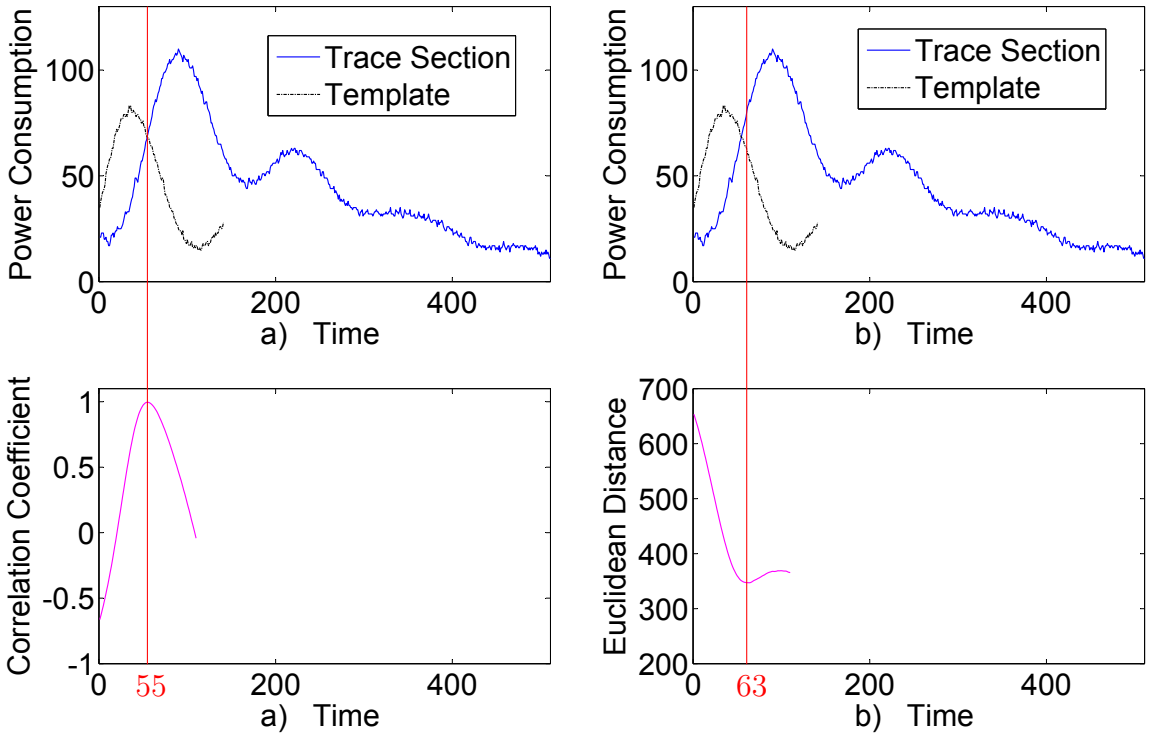

Figure 7.1: Pattern Matching With Different Matching Algorithms

7.2 Factors for Trace Pre-processing

In the trace pre-processing phase, several objective factors, e.g., template length, pattern matching algorithms, sample frequency, and adaption of the oscilloscope, etc., may affect the final attack results in practice, which are discussed in the sequel.

7.2.1 Template Length and Pattern Matching

In the pattern matching phase, the relationship between the template and the analyzed power trace is calculated by exploiting different means as distinguishers to find the best matched section in the analyzed power trace. However, the template length and the pattern matching algorithms do impact the final matched results. On the one hand, by matching the same template using different pattern matching algorithms, different results may be achieved, where tiny variations exist between each matched power trace leading to different attack results. Fig. 7.1 illustrates a mentioned case where the same template moves from left to right in the time domain within 100 time points. For each moving step, the correlation coefficient and the Euclidean distance are calculated, respectively. In order to depict it clearly, an analyzed trace section and a template are shown in the upper part of figure, meanwhile, the similarity calculation results are shown in the lower

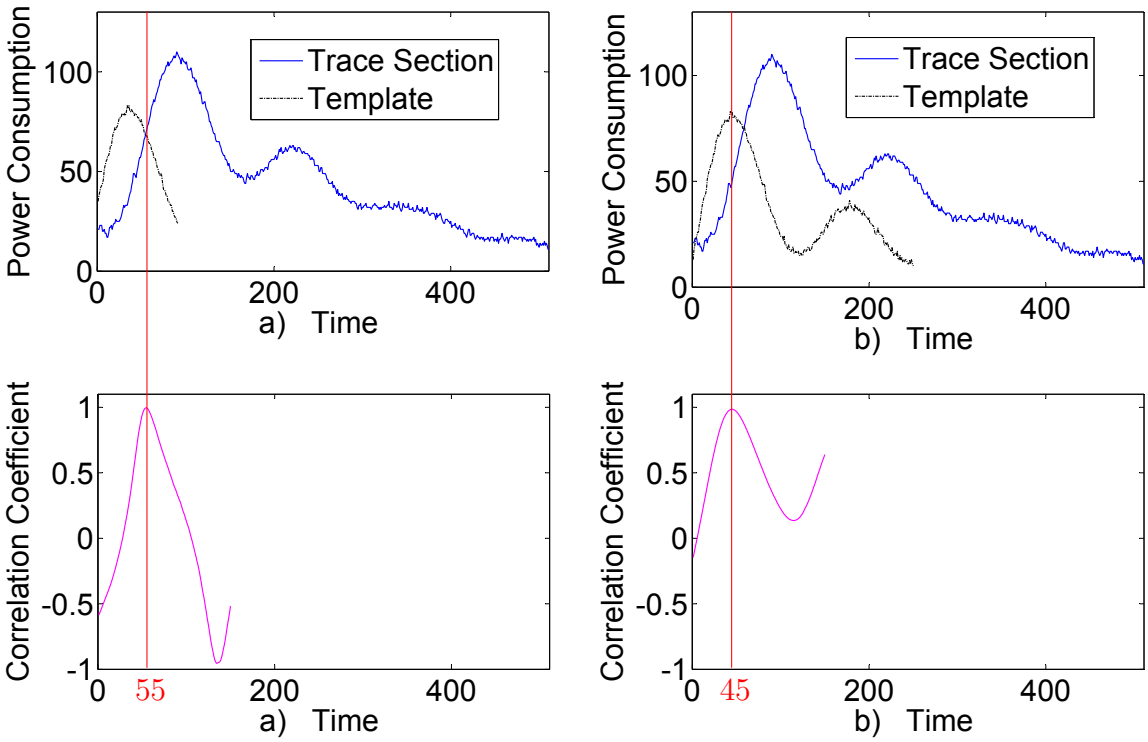

Figure 7.2: Pattern Matching With Different Length of the Template

part. Taking correlation coefficient as a distinguisher, the matched position is located at time point 55. However, in case of the Euclidean distance, the matched position is shifted to the time point 63; on the other hand, when exploiting the same pattern matching algorithm to match the power trace with varying length template, sometimes the results are not exactly the same. For example, the same trace section is matched by using different lengths of template, i.e., 90 and 250 time points hold, respectively, as shown in Fig. 7.2 a) and b). The correlation coefficient is taken as the distinguisher for these two operations. The matched positions are found at the time points 55 and 45, respectively. Consequently, for the time point attacks, such differences for the aligned power traces result in different attack performance. However, this influence may be minimized by applying PAA methodology directly for its concentrating on the time interval rather than the time instant. Therefore, a proper length of the template and a proper pattern matching algorithm affect the performance in trace pre-processing and attack phases.

7.2.2 Sample Frequency

The sample frequency for the oscilloscope plays an important role in the trace capture phase. From Nyquist sampling theorem, in order to reconstruct the being sampled signal

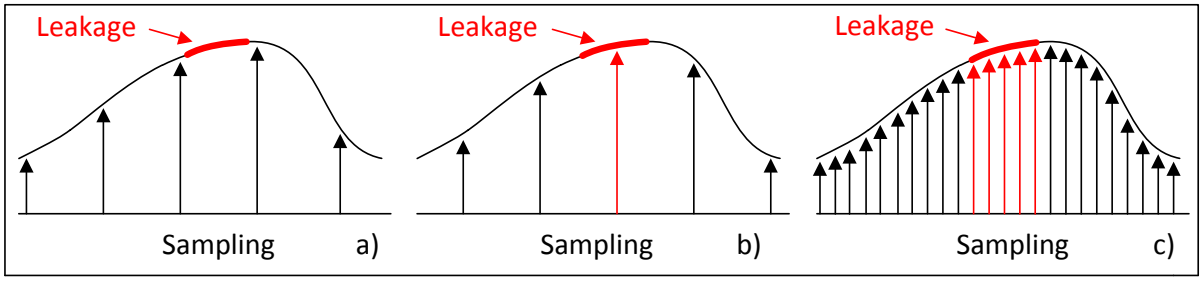

Figure 7.3: Sampling Frequency

from the discrete sample points, the sample frequency must be greater than the twice maximum frequency of the sampled signal, cf., [MA07, pp. 397], which is a basic principle for the oscilloscope working properly. Therefore, before capture the power traces, the sample rate of the oscilloscope and the maximum frequency of the being sampled signal must be checked carefully. Usually, the higher sample rate, the more information can be collected during the capture phase. In other words, an oscilloscope with good quality featuring a higher resolution is a great help for power consumption analysis. Assume that there exists an information leakage in an analyzed power trace segment. If this process is monitored by exploiting oscilloscope with lower sample rate, two possibilities may occur: firstly, as highlighted in Fig. 7.3 a), the leakage lies just in between of the two sampled points, then the sampled discrete curve contains nothing about the information leakage; secondly, fewer information leakage points may be captured, e.g., as illustrated in 7.3 b), where the oscilloscope features the same sample rate as in Fig. 7.3 a). However, only one time point carrying the information leakage is sampled and captured occasionally, which may not include the maximum information leakage point, i.e., most of the information leakage is lost during the sampling. A better way is illustrated in Fig. 7.3 c), where, the sample rate of the oscilloscope is higher, and more time points with the information leakage are sampled and recorded for the subsequent analysis. Therefore, by accelerating the sample rate of the being used oscilloscope, the probability for capturing the points containing the valid information leakage is increased leading to an efficient attack.

DPA and CPA are time point attacks, therefore, the extreme condition in Fig. 7.3 b) may work in such attacks. In contrast, PAA is a time interval attack. Consequently, under the same condition, a lot of time points without the information leakage are exploited resulting in the lower success rate. In other words, PAA attack requires the oscilloscope with a higher sample rate. Usually, in the laboratory, the better attack performance can be achieved by slowing down the input clock of the cryptosystem, where the sample rate of the oscilloscope is unchangeable.

7.2.3 Oscilloscope Adaption

Oscilloscope adaption is also an important factor in the trace capture phase. The speed for trace collection directly affects the final attack efficiency. In order to speed up the trace capture step, a small portion of the whole trace is concentrated by tuning the oscilloscope focusing on the information leakage region, where the adversary is focusing on. The captured power trace is first stored into the online cache of the oscilloscope. Then such data is transferred to the computer or other storage devices. If the whole power trace containing more time points is captured, the heavy burden for the data transferring line will decelerate this process resulting in a longer run time in trace capture phase.

7.2.4 Light, Temperature, and Surrounded Noises

In practice, the light, temperature, and surrounded noises, all may be the influence factors for a running cryptosystem. They have a common feature that a strong intensity of all these factors may vary the power consumption dissipated from the circuits. In other words, with the same input and output, the variations of power consumption stemmed from the cryptosystem are out of the tolerance range resulting in encumbrances for the subsequent attacks.

Strong lights can malfunction the cryptosystem. Let's take an extreme example in fault attack. Usually, the laser is exploited to shine the running cryptosystem to introduce the artificial errors in the running algorithm. Meanwhile, the variation of the power features certain irregular changes. If it occurs during the trace capture phase, the output of cryptosystem may be incorrect, as well as the behaviors of the captured power traces. Then with the wrong output and the abnormal power traces, the leakage model cannot be built correctly, and the right key bytes cannot be achieved resulting in the worse attack performance.

Usually, when the temperature becomes higher, the power dissipation from the cryptosystem is increased accordingly, which makes the temperature of the running system even higher. Therefore, with a higher temperature, for the same input and key, one may get different power traces resulting in an increment of trace usage in practical attacks.

The strong surrounded noises from the other electromagnetic devices also interfere with the working cryptosystem leading to unexpected variation in the power consumption or system.

In order to capture sound power traces containing unnecessary noises in practice, we recommend laying the running cryptosystem in a place, where exist no strong light,

Algorithm 9 Peak Position based Horizontal Alignment

Require: $T_{i,1:W}$, O_l, W_p
Ensure: $T'_{i,1:W_p}$
1: Detect the aimed peak position t^* in $T_{i,1:W}$ by exploiting algorithms: threshold or slope base peak position detection
2: Truncate the power trace from position $t^\circ - O_l$ with width W_p
3: Save the truncated power trace as the aligned trace $T'_{i,1:W_p}$
4: **return** $T'_{i,1:W_p}$.

constant lower temperature, and clean electromagnetic environment.

7.3 PAA Attack on Misaligned Power Traces

In this section, the drawbacks of the misaligned power trace pre-processing with time point attacks are outlined. In contrast, the autogenetic advantages of PAA attack are fully used to overcome the weak points stemmed from the time point attack methods resulting in faster calculation during attacks.

7.3.1 Misaligned Trace Pre-processing With Time Point Attacks

Both methods, horizontal alignment and vertical matching, can considerably improve the power consumption attacks. However, there exist two problems, which should be taken into the consideration. First, by mounting time point attacks, e.g., CPA attack, on the misaligned power traces the different template lengths and the algorithms chosen in the template matching phase yield different matching results; meanwhile, regardless which method being exploited in the pattern matching phase, the traces cannot be aligned exactly, i.e., there always exist alignment errors in both the horizontal alignment and vertical matching phases. These errors are taken as inevitable artificial noises interfering with the attack success rate. Therefore, the performance of time point attacks relies so heavily on the quality of trace pre-processing algorithms; on the other hand, in the vertical matching step, the extra efforts must be invested in this process. Therefore, we proposed a new efficient and effective way without the template trace and the vertical matching step in [TH12c], which is outlined in the sequel.

7.3.2 Peak Position Based Trace Alignment

In horizontal alignment, the captured power traces have to be truncated twice in the coarsely truncation and pattern matching phases, respectively. In order to minimize this

process, the stronger misalignment tolerance characteristic in PAA attack is fully utilized, where the peak position based trace alignment was proposed as Algorithm 9 combining with PAA attack resulting in a faster attack procedure without greatly cutting down the attack performance in practice. In other words, after the peak position finding, the power traces are truncated according to the peak position only once for the subsequent attacks. Without the pattern matching phase, the template is not required anymore, meanwhile a lot of computation time can be saved.

7.3.3 Invalidating of Vertical Matching

The mechanism for vertical matching is to move the power trace up and down in the amplitude domain to find the best matched trace section. Such a process is invalid in the PAA attack for its amplitude fluctuation invariance, which results in a stable attack performance regardless the movement of the power traces.

Based on the autologous features in PAA attack, the pattern and vertical matching steps are omitted together leading to the speed up of attack procedure in reality.

7.4 Attack Frameworks

7.4.1 Attack Framework of Misaligned Power Traces

In order to have a visible framework for attacking the misaligned power traces by applying different attack methods, a complete framework is proposed, which takes the attack methods, the trace pre-processing algorithms, and the fed base clock frequency into consideration.

There are two approaches to mount attacks on such misaligned power traces in practice, as illustrated in Fig. 7.4. Firstly, by exploiting path a), one can mount alignment sensitive attack methods, i.e., DPA, CPA, etc., on the misaligned power traces processed by the Template Based Horizontal Alignment (TBHA). If the clock frequency effects in the captured power traces are fierce, then the vertical matching may be applied to counteract these effects and to eventually improve the attack performance. The vertical matching step in PAA attack is superfluous resulting in short trace pre-processing time. In addition, the attack results are superior to CPA attack after the vertical matching phase, as shown in [TH12e] and [TH12d]; secondly, in order to have the short trace pre-processing time, the power traces processed by the Peak position Based Horizontal Alignment (PBHA) algorithm may be applied without running the pattern matching

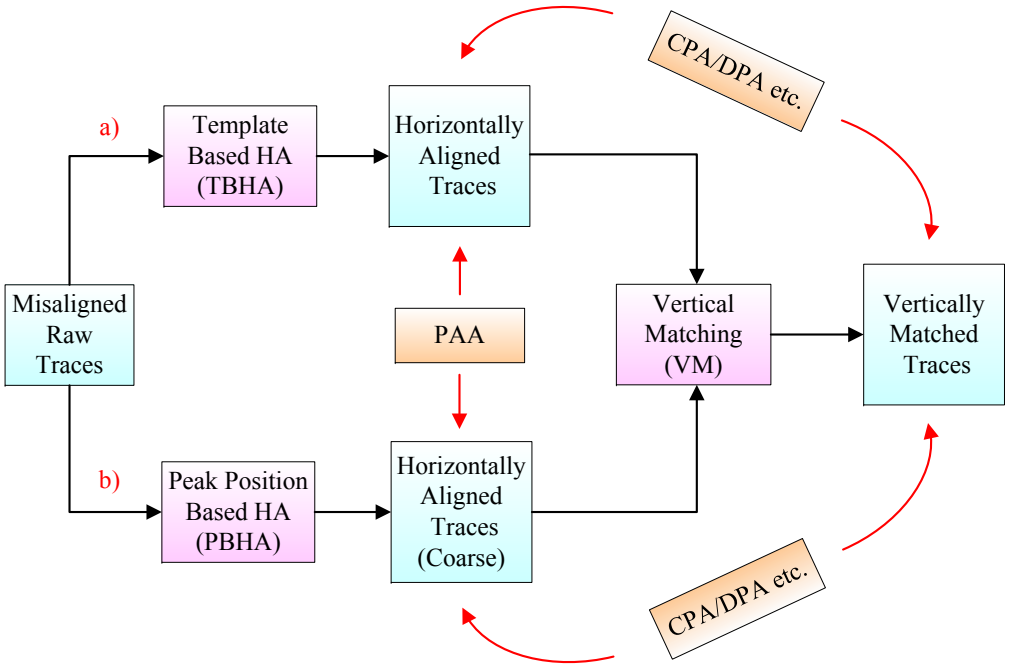

Figure 7.4: Attack Framework for Misaligned Raw Traces

step. Then, PAA attack can be mounted directly on such imprecisely aligned power traces. The attack results are better than or nearly equal to CAP attack after the vertical matching, as shown in [TH12c]. Subsequently, the roughly aligned power trace may be vertically matched when the base clock frequency is higher. CPA or DPA attacks can also be executed on the coarsely horizontally aligned and vertically matched power traces, respectively. However, the attack results are not so good. Comparing to the path a), the calculation for the path b) is simpler and faster. With the proposed attack architecture, the adversaries may access the secret key of cryptosystems in a selective manner. Moreover, this architecture can be exploited for the security evaluation in practice as well.

7.4.2 Attack Framework of Aligned Power Traces

In previous chapter, the trace pre-processing methods are generated for neutralizing the misaligned power traces yielded from a random clock featured cryptosystem. However, whether these methods can be applied to preprocess the originally aligned power traces are unknown. We propose our conceptions as following and verify them in the upcoming chapter by mounting several attacks in reality.

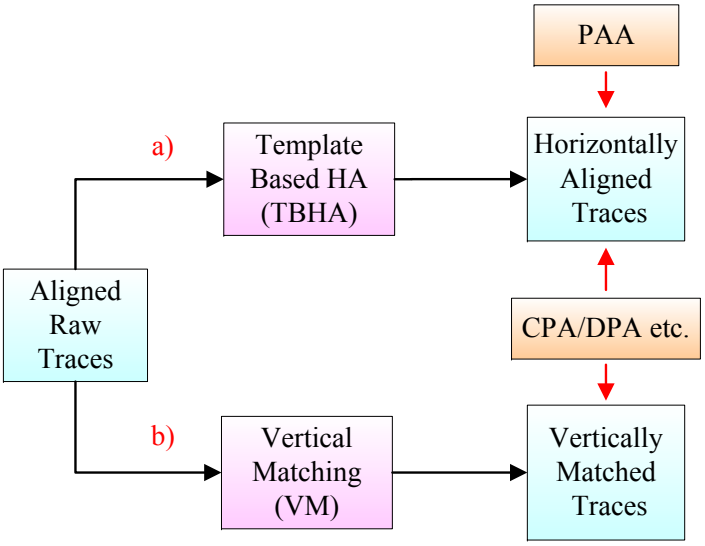

Figure 7.5: Attack Framework for Aligned Raw Traces

Horizontal Alignment Usually, the power traces captured from a running unprotected cryptosystem are assumed as the aligned ones. Therefore, it is time-consuming to run horizontal alignment algorithm on the aligned power traces. In addition, there exist some alignment errors in the horizontally aligned power traces. However, in order to improve the attack performance, we suggest exploiting path a) in Fig. 7.5 under the following conditions: on the one hand, the power traces are captured when the trigger signal is valid, however, the circuit networks may have some delays, which causes the beginning of the trigger signal for different input data may be a bit different. Therefore, there still exists the misalignment in the targeted originally aligned power traces; on the other hand, the clock chip may be of lower quality, then the input clock may contain some vibrations or shifts. If the mentioned cases are strong enough, the adversaries may also collect the power traces with tiny misalignment resulting in a bad attack performance. When one of the mentioned cases occurs, the horizontal alignment may be resorted to improve the attack performance. However, if these interferences are negligible, i.e., they are in a tolerance range, then the system can be attacked directly without spending the time in horizontal alignment. Therefore, whether the pre-processing should be mounted or not relies on the quality of the cryptosystem and the trace capture technology in reality. Here, the horizontal alignment can be mounted easily. Because the power traces are assumed as aligned, the positions of aimed power peaks for different power traces are nearly the same, which can be determined by observing the power traces. Subsequently, the power trace around the known peak position can be directly truncated for the pattern matching phase. Subsequently, the attacks can be executed.

Vertical Matching The reason to run vertical matching algorithm in the aligned traces is that, when the normal cryptosystem is fed by a higher clock frequency, although the input clock frequency is fixed, which may more or less cause the power consumption changing in the circuits. As mentioned in chapter 2, the dynamic power is the main portion of the total power dissipation in CMOS circuits. It does not mean the static power consumption is always stable. They do vary the power consumption within a tiny range when fed with higher clock frequencies. That is to say, in the electronic circuits, some other factors may also affect the power values changing when running at a higher clock frequency. Such tiny variations behave like the clock frequency effects in random clock featured cryptosystem. Therefore, we suggest exploiting vertical matching in path b) to deal with the aligned power traces for achieving a better attack performance, as illustrated in Fig. 7.5.

Chapter 8

Application Examples

Contents

8.1 Introduction

In this chapter, the attack frameworks combining the attack methods and trace pre-processing approaches are studied in real cases. The attack results are then compared accordingly by means of the evaluation metrics introduced in Chapter 3. In order to yield the persuasive experimental results without loss of generality, an FPGA-based cryptosystem running with different implementations and clock frequencies is attacked, respectively. The whole experiments are designed as four separate parts:

1. The performance for attack methods CPA, PAA, and MIA is compared by attacking the same power traces with Hamming distance leakage model.

2. Study the attack framework proposed in Fig. 5.2, Chapter 5, by mounting attack methods CPA, PAA, PAA-I, and PAA-II on the normal power traces with the mathematic and trace form leakage models, respectively.

3. Preprocess the aligned power traces with the trace pre-processing workflow, as illustrated in Fig. 7.5, Chapter 7. Then mount CPA attack on the preprocessed power traces accordingly.

4. Preprocess the misaligned power traces according to the trace pre-processing workflow proposed in Fig. 7.4, Chapter 7. Meanwhile, mount CPA and PAA attacks during the processing steps, respectively.

The success rate, guessing entropy, run time, and minimal traces usage are exploited as evaluation metrics. Before running the specific experiments, the considered platform is introduced firstly.

8.2 Hardware and Software

There are several platforms, which can be exploited to run the cryptographic algorithms for yielding the analyzed power traces, e.g., smart card, micro-controller, FPGA, and ASIC, etc. In order to have flexible choices for the cryptographic implementations and to generate the comparable attack results, the considered platform is side channel attack standard evaluation board (SASEBO) version G [Ins08]. The whole architecture of this board is shown in Fig. 2.8, Chapter 2. Two Xilinx Virtex-II pro FPGAs, i.e., XC2VP30-FG676 and XC2VP7-FG456 are deployed on this development board as the control and cryptographic devices, respectively. By exploiting such a platform, the architectures for the cryptographic algorithms can be modified and customized according to the specific requirements. During the whole experiments, a random number generator may be inserted between the oscillator and the cryptographic FPGA, when a random clock featured cryptosystem is required to yield the misaligned power traces.

The power and the input clock for the board are supplied by an Agilent E3646A DC power supply and a 33250A waveform generator, respectively. The power supply can provide voltage 0-8V with 3A current or voltage 0-20V with 1.5A current. For the waveform generator, it can generate function or arbitrary waveforms with the maximum frequency

of 80MHz.

For trace capture, the Agilent oscilloscope is considered with model number DSO6052A featuring 500 MHz bandwidth and 4 GSa/s sample rate, which contains two channels: one is applied to sample the trigger signal; the other one is used to capture the power consumption dissipated from a working cryptosystem.

The calculation computer features a Quad-Core processor running at 2.8 GHz with 8 GB memory. Indeed, the efficiency of the attack methods sometimes relies on the configuration of calculation computer. The fast CPU and the large memory result in better attack performance in practice.

Xilinx ISE is exploited to edit and compile the VHDL designs, which can be downloaded to FPFA chips to fulfill some given tasks. Matlab is used as the analysis software to preprocess the captured power traces and to mount power consumption attacks. If C/C++ codes or parallel calculation, etc., can be utilized during the attack, the attack results may be achieved faster.

8.3 Comparison of Normal Power Trace Attacks

In this section, CPA, PAA, and MIA attacks are executed on the normal power traces with the same leakage model for the comparison of attack performance.

8.3.1 Experimental Setup

The considered algorithm is AES-128 featuring TBL S-Box as proposed in [RDJ+01]. The Hamming weight concentrating on the register in the last round before and after the S-Box is taken as a leakage model for all subsequent attacks, as detailed in (2.5). Then the CPA, MIA and PAA attacks are mounted, respectively.

The whole system is running at 2 MHz clock frequency. In the attacking phase: for each key byte, the experiments are mounted 30 times with different input data; for each experiment, the maximum number of the trace usage is set to 3,500; in each attack, 10 power traces are added, i.e., 350-time attacks are run in each experiment. Subsequently, the success rate, guessing entropy, and run time are presented and compared, respectively.

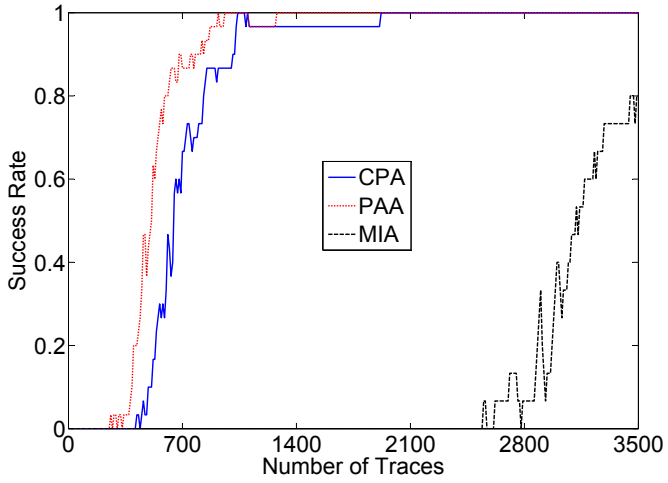

Figure 8.1: Global SR Comparison of CPA, PAA, and MIA Attacks

8.3.2 Experimental Results

8.3.2.1 Success Rate Comparison

Fig. 8.1 shows the global success rate of CPA, PAA, and MIA attacks, respectively. One finds clearly that the PAA attack shows the best attack performance to reveal all the key bytes within 1,000 power traces; To achieve the same goal, CPA attack needs nearly 1,800 power traces, whereas the MIA attack cannot crack all the key bytes within provided power traces.

In order to do a further comparison, the success rate and the guess entropy for each key byte are illustrated in Fig. B.1 and B.2, Appendix B, respectively. Comparing to CPA and MIA attacks, for revealing all the key bytes, PAA attack always features the best performance in terms of the success rate and the guessing entropy; meanwhile, by exploiting MIA distinguisher, the byte nine needs more power traces to be revealed successfully, while the other bytes can be attacked within the given power traces. However, for each key byte attacking, MIA presents the worse attack performance than CPA and PAA attacks.

8.3.2.2 Run Time Comparison

In order to have an intuitive impression about the run time for different attacks, all these attack methods are executed to reveal the first key byte of the analyzed cryptographic algorithm within 10,000 power traces spanning 600 time points in the analysis region. In other words, the size of the power trace set T and the HD leakage model are $10,000 \times 600$

Attack Methods	CPA	PAA	MIA
Run Time	1.06s	0.75s	33.67s

Table 8.1: Run Time Comparison for Different Attack Methods

and $10,000 \times 256$, respectively. For CPA and MIA attacks, the correlation coefficient and mutual information are calculated 600×256 times. However, please note that in MIA attack, the mass function for each analyzed data should be estimated in advance, which is a time consuming task. Whereas in PAA attack, the purified leakage vector V features the size $10,000 \times 1$, then the correlation coefficient is mounted only for 1×256 times resulting in lower calculation complexity. In order to get an average time requirement for the evaluated attack methods, each experiment is repeated for 10 times.

Table 8.1 shows the run time for all the mentioned attack methods. PAA Attack requires the shortest run time, i.e., 0.75s, while the MIA needs the longest one, which is nearly 44 and 31 times of the figures in PAA and CPA attacks, respectively. Under the same condition, comparing to CPA attack, the PAA attack can save 29% run time in practice. There is no double that MIA requires more time because of the higher calculation complexity in the mass function estimation.

From the presented figures, one finds that the PAA and CPA attacks show better attack performance in terms of traces usage and time requirements, whereas, the MIA attack shows an opposite case. Consequently, in the subsequent attacks, MIA attack is out of our consideration.

8.4 Attack Framework Evaluation

In this section, the normal power traces produced from the light-weight block cipher, i.e., PRESENT, mentioned in Chapter 2 running at 7 MHz clock frequency are captured and attacked to evaluate the attack framework proposed in Fig. 5.2, Chapter 5. The considered attack methods are CPA, PAA, PAA-I, and PAA-II.

8.4.1 Experimental Setup

Two leakage models are exploited in the experiments. One is the HD leakage model, where the Hamming distance for the first round of PRESENT algorithm before and after the S-Box will be calculated by (2.8); the other one is the least squares estimation based trace form leakage model. Before calculating the Hamming distance, the bitwise values $p_i \oplus Sbox(p_i \oplus k_i)$ are mapped into the matrix A by (5.10). In the following section, such

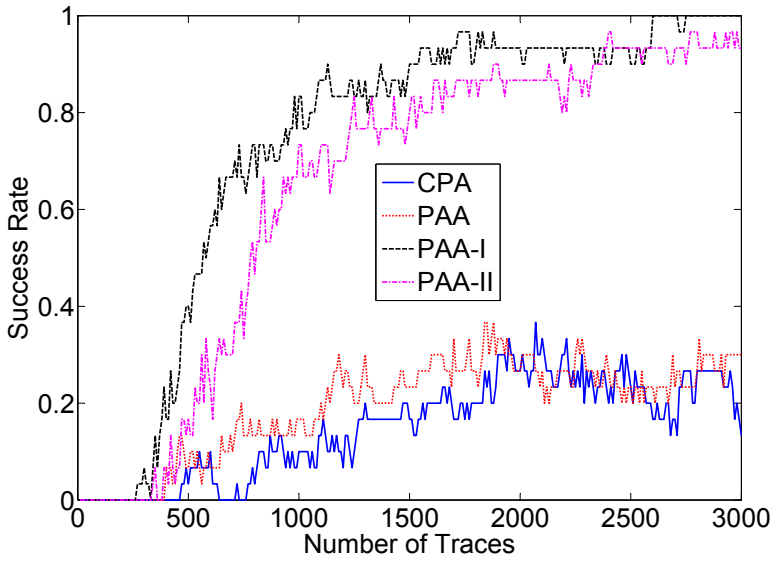

Figure 8.2: Global SR Comparison for CPA, PAA, PAA-I and PAA-II

a model is called Least Squares estimation based Trace form leakage Model (LSTM). Please note that the system is attacked nibble by nibble, i.e., 4 bits. The experiments are divided into three groups as following.

1. Mount PAA-II attack to analyze the power trace set T and the purified leakage matrix $\widetilde{V}$ by exploiting correlation coefficient, as illustrated in Fig. 5.2, Analysis 5).

2. Run PAA-I attack to analyze the purified leakage vector V and the matrix $\widetilde{V}$ by exploiting correlation coefficient, as shown in Fig. 5.2, Analysis 4).

3. Execute CPA and PAA attacks with the HD leakage model, as shown in Fig. 5.2, Analysis 1) and Analysis 2), respectively.

In the attack phase: for each key nibble, the experiments are mounted 30 times with different input data; for each experiment, the maximum number of the traces is set to 3,000; in each attack, 10 power traces are added, i.e., 300-time attacks are executed in each experiment. Subsequently, the evaluation metrics, i.e., success rate and guessing entropy, are presented for the comparison.

8.4.2 Experimental Results

Fig. 8.2 shows the global success rate for the proposed attack methods. Within 3,000 power traces, CPA and PAA attacks present lower global success rate than PAA-I and

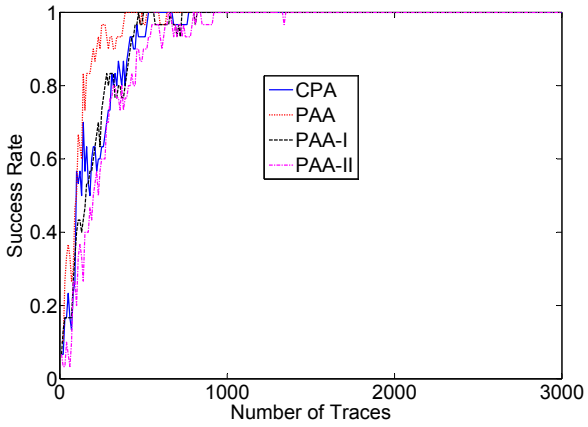

Figure 8.3: SR of Nibble 2

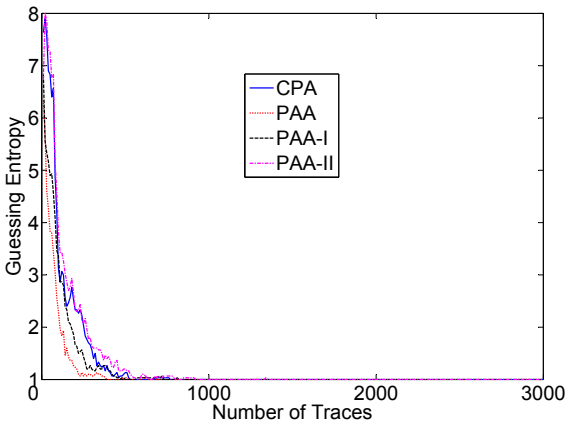

Figure 8.5: GE of Nibble 2

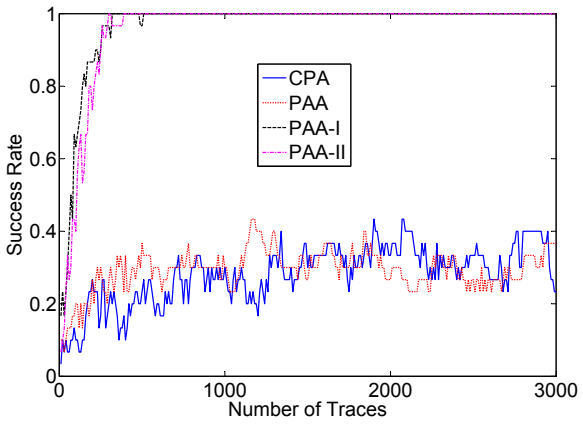

Figure 8.4: SR of Nibble 8

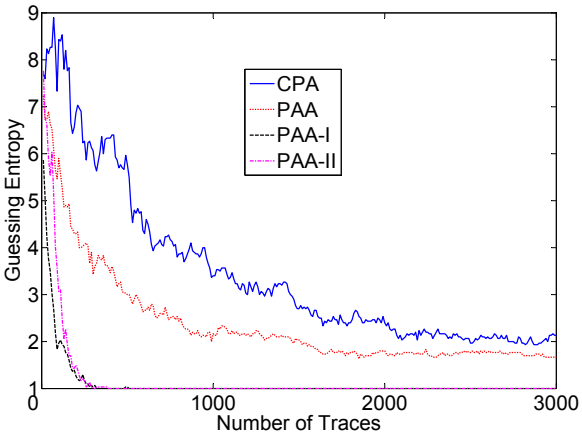

Figure 8.6: GE of Nibble 8

PAA-II attacks, which can be seen obviously in the figure. Among all these attacks, the trace form leakage model improves the attack significantly. However, only the attack PAA-II can reveal all the 16 key nibbles successfully after using 2,750 power traces.

Fig. B.3 and Fig. B.4 in Appendix B show the success rate and guessing entropy for each nibble, respectively. For nibbles 3, 4, 8, and 9, the success rate and the guessing entropy curves for PAA-I and PAA-II surpass their counterparts in CPA and PAA attacks, i.e., the success rate rising rapidly, while the guessing entropy falling fast, which is the reason that the global success rates for PAA-I and PAA-II are superior to CPA and PAA attacks; for the rest key nibbles in each sub-figure, four curves are twisted a bit. Such a phenomenon is illustrated by the larger success rate and guessing entropy figures for nibble 2 and 8 in Fig. 8.3, Fig. 8.4, Fig. 8.5, and Fig. 8.6, respectively. As known, CPA and PAA attacks are powerful, however, there still exist some nibbles, which leak less information resulting in the lower success rate, e.g., nibbles 3, 4, 8, and 9. Therefore, one can resort to PAA-I and PAA-II attacks. Especially in nibble 8, the success curve rises faster than its counterparts in CPA and PAA attacks. The minimal number of the

power traces to attack such a nibble in PAA-I and PAA-II are 520 and 390, respectively, which are rather low.

In this section, the attack framework proposed in Chapter 5 are studied. The experimental figures show that the proposed attack framework is feasible, which provides more choices to reveal the cryptosystem with different leakage models and analysis methods. Meanwhile, it shows that the trace form leakage model is useful, as well as the new attack methods, i.e., PAA-I and PAA-II. Both methods can excavate more hidden information from the power traces. This attack framework may help the adversaries in real attacks as well as the security evaluation for cryptosystems.

8.5 Pre-processing of Aligned Power Traces

In this section, the trace pre-processing algorithms, i.e., horizontal alignment and vertical matching, are exploited to preprocess the originally aligned power traces, respectively. Subsequently, the cryptosystem is attacked by studying the attack framework proposed in Fig. 7.5, Chapter 7, where the CPA attack is executed several times to yield the comparable attack results.

8.5.1 Experimental Setup

The considered cryptographic algorithm is AES-128 featuring PPRM3 S-Box, as detailed in [MS02]. The whole cryptosystem is running at 12 MHz clock frequency.

The Hamming distance leakage model focusing on the register states changing before and after the S-Box in the last round is considered, as shown (2.5), Chapter 2. In order to produce the comparable results, the whole experiment is divided into two parts:

1. Execute CPA attack on the power traces before and after the horizontal alignment, respectively.

2. Run CPA attack on the power traces before and after the vertical matching, respectively.

For each key byte, the experiments are mounted 30 times with different input data; for each experiment, the maximum number of the traces is set to 4,500; in each attack, 10 power traces are added, i.e., 450-time attacks are mounted for each experiment. Subsequently, the success rate and guessing entropy are presented and compared, respectively.

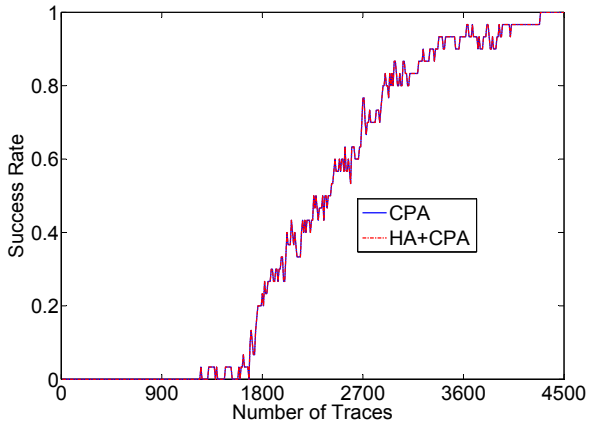

Figure 8.7: Global SR Comparison Before and After HA

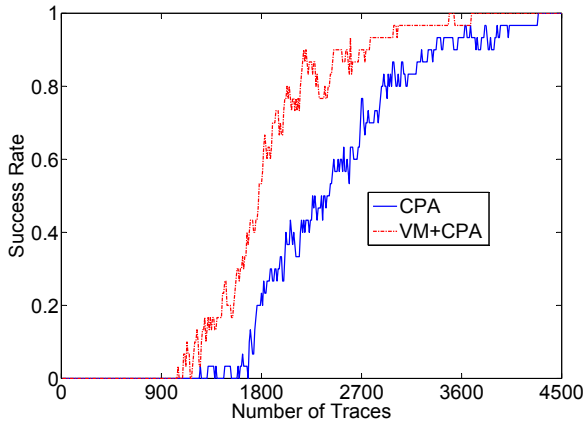

Figure 8.8: Global SR Comparison Before and After VM

8.5.2 Experimental Results

8.5.2.1 CPA Attack Before and After Horizontal Alignment

Fig. 8.7 illustrates the global success rate for mounting CPA attack on the power traces before and after the horizontal alignment. One finds that the two success rate curves are overlapped with each other exactly. It implies that the captured power traces produced from the targeted cryptosystem embody no clock shifting caused misalignment. Therefore, the pre-processing step can be omitted. Fig. B.5 and Fig. B.6 in Appendix B show the success rate and the guessing entropy for each key byte, respectively. One can see clearly the same situation, where both curves in the figures overlap with each other exactly. As mentioned in previous chapter, horizontal alignment for pre-processing the aligned power traces usually is not necessary. Because a tiny clock shifting of the cryptosystem is in a tolerance range. Therefore, the attacks can be mounted directly.

8.5.2.2 CPA Attack Before and After Vertical Matching

Fig. 8.8 shows the global success rate for executing CPA attack before and after the vertical matching. Different from the scene in the horizontal alignment, the presented two success rate curves show a gap. In addition, after the vertical matching, the success rate curve, i.e., VM+CPA, rises faster than its counterpart, i.e., CPA attack before the vertical matching. In other words, the success rate is improved. And the minimal power trace usage for revealing all the 16 key bytes is decreased from 4,290 to 3,690, i.e., 14% power traces are saved. In order to have an intuitive impression, the success rate and the guessing entropy for each attackable byte are illustrated in Fig. B.7 and Fig. B.8, respectively. The success rates increment for bytes 2, 8, 12, 14, and 16 are not that high,

which are nearly the same as their counterparts before the vertical matching. However, for the rest key bytes, the success rates are obviously improved. The experiments show that there are some other factors, which cause the values of the power peaks changing, even if the cryptosystem is driven with a fixed clock. Therefore, when the system is running at a higher clock frequency, the adversaries are recommended to preprocess the aligned power traces by means of the vertical matching to neutralize the power value variation and to achieve an improvement of attack method.

8.6 Attacks on Misaligned Power Traces

In this section, the misaligned power traces produced from a random clock featured AES-128 cryptosystem are attacked by studying the attack framework proposed in Fig. 7.4, Chapter 7.

8.6.1 Experimental Setup

The S-Box type is TBL [RDJ+01]. The base clock chip is running at 7 MHz clock frequency, thus, theoretically, the random frequencies fed to the cryptographic core vary the range from 0 to 7 MHz.

The considered leakage model is Hamming distance focusing on the register states changing before and after the S-Box in the last round by (2.5). The targeted attack methods are CPA and PAA. Before running these attacks the horizontal alignment by using template and peak position based algorithms proposed in Chapter 6 and 7 is run in advance, respectively, resulting in the aligned power traces. From the practical observation, one finds that the clock frequency effects in these misaligned power traces are a bit higher. Therefore, it is necessary to execute vertical matching for the horizontally aligned power traces. The whole experiment is divided into two parts as below.

1. Mount CPA and PAA attacks on the power traces preprocessed by the template based horizontal alignment and the vertical matching, respectively.

2. Execute CPA and PAA attacks on the power traces preprocessed by the peak position based horizontal alignment and the vertical matching, respectively.

For each key byte, the experiments are mounted 25 times with different input data; for each experiment, the maximum number of the traces is set to 4,000; in each attack, 10 power traces are added, i.e., 400-time attacks are mounted in each experiment. Subsequently, the success rate, guessing entropy, run time, and minimal traces usage are

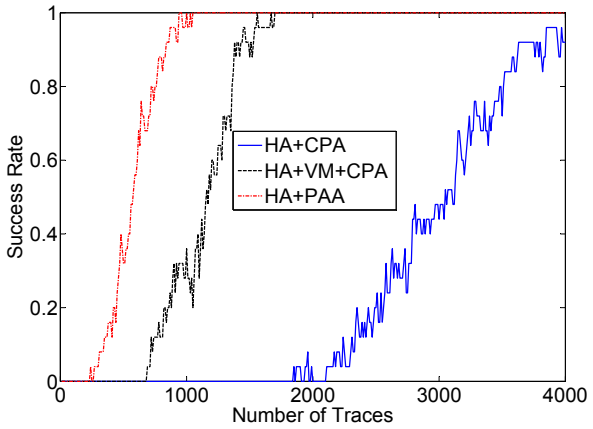

Figure 8.9: Global SR Comparison of CPA and PAA With Template Based Horizontal Alignment

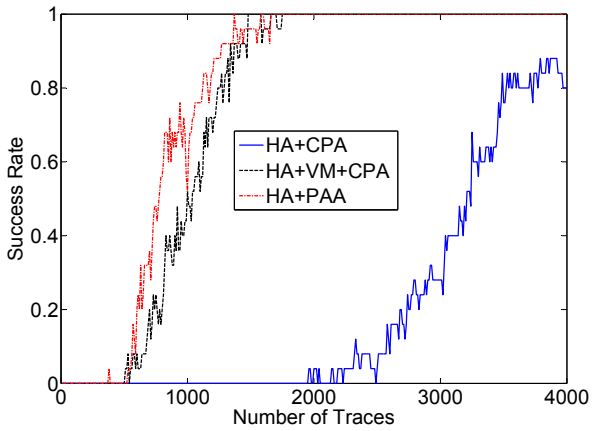

Figure 8.10: Global SR Comparison of CPA and PAA With Peak Position Based Horizontal Alignment

presented and compared.

8.6.2 Experimental Results

8.6.2.1 Template Based Horizontal Alignment

Fig. 8.9 shows three global success rate curves resulted from the proposed attacks on the horizontally aligned and vertically matched power traces, respectively. One finds that within 4,000 power traces, after the horizontal alignment, CPA attack cannot reveal all the key bytes successfully, i.e., the success rate is 92%. In contrast, when mounting CPA attack on the vertically matched power traces, the attack results are improved significantly, i.e., within 1,700 power traces, all the key bytes are recovered, see the curve HA+VM+CPA. Therefore, when the cryptosystem is running with higher random clock frequency, the vertical matching is an efficient way to counteract the clock frequency effects and to eventually improve the alignment sensitive attack methods, e.g., CPA attack. PAA attack shows the best attack performance by just attacking the horizontally aligned power traces without efforts investment in the vertical matching, where for revealing all the key bytes, only 1,050 power traces are consumed. The success rate and guessing entropy for each attackable byte are illustrated in Fig. B.9 and Fig. B.10, respectively. It is clear that the attack performance order for each byte from best to worst shows always like HA+PAA, HA+VM+CPA, HA+CPA.

Table 8.2: Time Requirement and Minimal Trace Usage

Type	Process Type	HA	VM	CPA	PAA	TTR	MTU
TB	HA+CPA	107.31s	-	13.77s	-	121.08s	>4,000
	HA+VM+CPA		40.24s	13.44s	-	160.99s	1,700
	HA+PAA		-	-	7.88s	115.19s	1,050
PB	HA+CPA	13.98s	-	13.52s	-	27.5s	>4,000
	HA+VM+CPA		40.68s	13.53s	-	68.19s	1,750
	HA+PAA		-	-	7.96s	21.94s	1,660
TTR: Total Time Requirement				MTU: Minimal Trace Usage			
TB: Template Based				PB: Peak position Based			

8.6.2.2 Peak Position Based Horizontal Alignment

Fig. 8.10 illustrates the global success rate for CPA and PAA attacks as well. Different from the previous attacks in Fig. 8.9, in these attacks, the power traces are aligned only according to the peak positions, i.e., the alignments are not that precise. Therefore, the pattern matching step in the horizontal alignment is omitted leading to a fast calculation. One finds that after the horizontal alignment, the CPA attack still cannot reveal all the correct key bytes, and the global success rate is reduced from 92% in the template based horizontal alignment to 80% after using 4,000 power traces, see the curve HA+CPA. In other words, the imprecise alignment causes the decline during the attacks. In order to improve the CPA attack, vertical matching may be executed for the horizontally aligned power traces. It is true that the attack results are improved, and the global success rate is almost the same as its counterpart in Fig. 8.9. The number of the minimal used traces to reveal all key bytes is increased from 1,700 to 1,750, as shown in the last column of the Table 8.2. In PAA attack, the global success rate is also reduced, i.e., the rising point starts after the using of 520 traces, however, which still rises faster than CPA attack after the vertical matching. The variations of success rate and guessing entropy for each byte are given in Fig. B.7 and Fig. B.8, respectively. The performance order is the same as the previous attacks in Fig. B.5.

8.6.2.3 Time Requirements and Minimal Trace Usage

In order to show the time requirements in the trace pre-processing and attacking phases, 10,000 power traces are preprocessed and attacked by the proposed attack methods, as shown in Table 8.2. The template based horizontal alignment requires 107.30s to process all the supplied power traces. However, the peak position based horizontal alignment only needs 13.98s. It means that the latter is 7 times faster than the former. Because in the template based horizontal alignment, there exists a pattern matching phase, which takes

some time. In all these attack combinations, HA+VM+CPA in TB takes the longest run time, i.e., 160.99s; whereas HA+PAA in PB requires the shortest run time, i.e., 21.94s. It is true that the peak position based horizontal alignment is faster due to its imprecise alignment. However, the success rates for CPA and PAA attacks are reduced. Although the attack performance is reduced, the minimal trace usage to reveal all the key bytes in PAA attack is as low as 1,660, which is still less than that in the CPA attack after the vertical matching in the template based horizontal alignment. In other words, it is difficult to balance the processing time and the trace usage in practice. Therefore, we suggest the adversaries just choosing the attack framework in an active manner: if the time is limited in reality, they can try to speed up by exploiting peak position based horizontal alignment; if the trace usage is a crucial requirement, the template based horizontal alignment with vertical matching is then selected. In a word, the different possibilities to preprocess the misaligned power traces with different attack methods are provided. One can choose it freely according to the specific implementations and the attack requirements.

8.7 Summary

In order to widely study the attack frameworks and trace pre-processing architectures proposed in previous chapters without loss of generality, in this chapter, different implementations of block cipher running at the different clock frequencies are evaluated. All these application examples prove that the new attack methods and the trace preprocessing architectures work properly and efficiently, which provide researchers more choices in practice to evaluate the security of cryptosystem.

Chapter 9

Conclusions and Future Works

Contents

9.1 Conclusions

In this work, the main contribution is divided into three parts, i.e., attack methods, trace pre-processing approaches, and attack frameworks.

Attack Methods The conception of PAA attack was extended as the PAA methodology based on the thoughts from communication field. Different from the time point attack methods, e.g., DPA and CPA, etc., traversing the time points for searching the maximum information leakage, PAA methodology utilizes a large number of time points in the power trace to contribute the information leakage and purifies them by calculating variance or standard deviation. Such a methodology inherits some autogenetic characteristics, e.g., stronger misalignment tolerance, and amplitude fluctuation invariance, from the original PAA attack, which conveys two useful conceptions. One is the way to purify the information leakage and reduce the dimension in the analyzed data; another one emphasizes that more similarity analysis methods may be run in the attack phase. Later, the least squares estimation based trace form leakage model was proposed, which is exploited fully by the derivative attack methods PAA-I and PAA-II suggested on the basis of PAA methodology. Both attack methods excavate the information leakage from trace form leakage model resulting in an improvement of attack performance in practice. Finally, all these attack methods are combined into an attack framework for supplying more choices when mounting attacks.

Trace Pre-processing A series of trace pre-processing methods were proposed when facing with practical problems for attacking the misaligned power traces. Basically, the template based horizontal alignment is applied for neutralizing misalignment in the time domain dynamically and partially, where the threshold and slope based peak position detection algorithms were given, respectively, for finding the peak position in a selective manner. Later, in order to counteract the clock frequency effects, the vertical matching algorithms were suggested. Both algorithms can handle the misaligned power traces quite well and improve the attack performance considerably. Meanwhile, a software architecture was given to clearly show the readers how to deal with the misaligned power traces easily. Finally, peak position based alignment was proposed, by which, the template and vertical matching phases are omitted for saving a lot of time in the trace pre-processing phase.

Attack Frameworks Different attack methods and trace pre-processing approaches were embedded into the attack frameworks for giving more choices to process and to attack the misaligned and the originally aligned power traces efficiently. Such frameworks were evaluated and verified via the real application examples in Chapter 8 with expected attack performance.

9.2 Future Works

In this thesis, there are still some open questions should be done or solved in the future.

Attack Phase In the leakage model building, for instance, a HD leakage model focusing on a certain byte register features only nine values, i.e., from 0 to 8. However, between the two analysis states S_1 and S_4, there exist some other operations. If all these operations cause the same register states changing, one may build a multi-section leakage model for these specific operations by exploiting some intermediate states, e.g., S_2 or S_3, i.e., the leakage models for states pair (S_1, S_2), (S_2, S_3), (S_3, S_4) are calculated, respectively. Then all these leakage models are summated as the final one for the attack. The accumulation of the multi-section leakage model changes the value range of previous HD model from 0 to 24, which may depict the power value changing more precisely. In other words, such a leakage model is more distinguishable in real attacks. Theoretically, these summated states take place in the different time instants, i.e., the summation value denotes the power variation during a time interval, which fits for the PAA methodology very well and can be exploited in the future to improve the attack performance .

In PAA methodology, several hundreds or even thousands sampled points in time are involved into the calculation. At the moment, these time points are chosen according to the adversaries' experience. However, how to choose them precisely is a difficult task. Therefore, a pre-calculation method is needed for the time points choosing, by which, the number and the start position for these time points are selected properly resulting in the optimum attack results.

In the power consumption attack field, the voltage variation curve is taken directly as the power consumption curve during the attacks. As known, the power consumption P and the square of voltage U^2 are in a linear relation. Therefore, theoretically, the square of voltage curve matches the power consumption oriented leakage model quite well to pursue a better attack performance, which needs a large number of verifications in the future.

Trace Pre-processing Phase Regardless the fixed or random clock featured cryptosystem, from the experiments in previous chapter, one finds that when the input clock is higher, the power traces all feature more or less the power value changing for the power peaks resulting in lower attack performance. However, the relationship between the input clock frequency and the power value variations are not that clear. If one can estimate quantitatively for both parties, then during attacks, the adversaries can decide when to mount vertical matching algorithm; meanwhile, the shifting step in the vertical matching phase can be set more precisely for the time saving.

Summary Side channel analysis is a multidisciplinary research topic. Most of the attack methods are focusing on the leakage model building and distinguisher improvement. However, by exploiting different trace processing methods, one can abstract and purify the information leakage from the power traces resulting in an improvement of attack performance. In the future a lot of mature technologies from other disciplines can be transplanted to such a field, which may supply us various sorts of possibilities to access the information leakage from the physical cryptosystem.

Bibliography

[AARR03] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 29–45. Springer, 2003. (Cited on page 4.)

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004. (Cited on page 5.)

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *Advances in Cryptology - EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997. (Cited on page 3.)

[BGP+11] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual information analysis: a comprehensive study. In *Journal of Cryptology*, volume 24, pages 269–291, 2011. (Cited on pages 33 and 35.)

[BKL+] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer. (Cited on pages 19 and 20.)

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *Advances in Cryptology-CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997. (Cited on page 3.)

[CG00] Jean-Sébastien Coron and Louis Goubin. On boolean and arithmetic masking against differential power analysis. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 1965 of *Lecture Notes in Computer Science*, pages 231–237. Springer, 2000. (Cited on page 38.)

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002. (Cited on pages 5, 31 and 32.)

[CT05] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, second edition, 2005. (Cited on pages 33 and 43.)

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES-The Advanced Encryption Standard*. Springer, 2002. (Cited on page 2.)

[Dri09] Saar Drimer. Security for volatile fpgas. 2009. `http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-763.pdf`. (Cited on page 4.)

[GBTP08] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008. (Cited on pages 5, 33 and 35.)

[GLRP06] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. stochastic methods. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2006. (Cited on pages 5 and 32.)

[Gol05] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005. (Cited on page 43.)

[Gou01] Louis Goubin. A sound method for switching between boolean and arithmetic masking. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 3–15. Springer, 2001. (Cited on page 38.)

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001. (Cited on page 57.)

[Ins08] Research Center For Information Security National Institute. Side channel attack standard evaluation board version g specification. 2008. `http://www.morita-tech.co.jp/SASEBO/en/board/sasebo-g.html`. (Cited on page 100.)

[JPS05] Marc Joye, Pascal Paillier, and Berry Schoenmakers. On second-order differential power analysis. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2005. (Cited on page 38.)

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology-CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999. (Cited on pages 4 and 27.)

[KKM08] Ann Hibner Koblitz, Neal Koblitz, and Alfred Menezes. Elliptic curve cryptography: The serpentine course of a paradigm shift. In *IACR Cryptology ePrint Archive*, volume 2008, page 390, 2008. (Cited on page 3.)

[LRP07] Kerstin Lemke-Rust and Christof Paar. Analyzing side channel leakage of masked implementations with stochastic methods. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 4734 of *Lecture Notes in Computer Science*, pages 454–468. Springer, 2007. (Cited on pages 5 and 32.)

[MA07] Mrinal Mandal and Amir Asif. *Continuous and Discrete Time Signals and Systems*. Cambridge University Press, 2007. (Cited on page 92.)

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007. (Cited on pages 16, 28, 29 and 65.)

[MOV96] A. Menezes, P.C. Van Oorschot, and S. Vanstone. *The Handbook of Applied Cryptography*. CRC Press, 1996. (Cited on page 19.)

[MS02] Sumio Morioka and Akashi Satoh. An optimized s-box circuit architecture for low power aes design. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 2002. (Cited on page 106.)

[MTT+05] Daniel Mesquita, Jean-Denis Techer, Lionel Torres, Gilles Sassatelli, Gaston Cambon, Michel Robert, and Fernando Moraes. Current mask generation: a transistor level security against dpa attacks. In *18th Annual Symposium on Integrated Circuits and Systems Design (SBCCI)*, pages 115–120. ACM, 2005. (Cited on page 5.)

[OMHT06] JElisabeth Oswald, Stefan Mangard, Christoph Herbst, and Stefan Tillich. Practical second-order dpa attacks for masked smart card implementations of block ciphers. In *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 192–207. Springer, 2006. (Cited on page 38.)

[RDJ+01] Atri Rudra, Pradeep K. Dubey, Charanjit S. Jutla, Vijay Kumar, Josyula R. Rao, and Pankaj Rohatgi. Efficient rijndael encryption implementation with composite field arithmetic. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 171–184. Springer, 2001. (Cited on pages 101 and 108.)

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. In *Communications of the ACM*, volume 21, pages 120–126, 1978. (Cited on page 3.)

[SKB02] Li Shang, Alireza Kaviani, and Kusuma Bathala. Dynamic power consumption in $VIRTEX^{TM}$-II FPGA family. In *ACM Int. Symposium on FPGAs*, pages 157–164, 2002. (Cited on page 15.)

[SLP05] Werner Schindler, Kerstin Lemke, and Christof Paar. Stochastic model for differential side channel cryptanalysis. In *Cryptographic Hardware and Embedded Systems - CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005. (Cited on pages 5, 31, 32 and 58.)

[SMBY05] Danil Sokolov, Julian P. Murphy, Alexandre V. Bystrov, and Alexandre Yakovlev. Design and analysis of dual-rail circuits for security applications. In *IEEE Trans. Computers*, volume 54, pages 449–460, 2005. (Cited on page 5.)

[SMY09] F.-X. Standaert, T.G. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Eurocrypt, Lecture Notes in Computer Science*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009. (Cited on page 39.)

[SP06] Kai Schramm and Christof Paar. Higher order masking of the aes. In *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2006. (Cited on page 5.)

[ST99] National Institute Of Standards and Technology. Fips pub 46-3. data encryption standard (DES). 1999. (Cited on page 2.)

[ST01] National Institute Of Standards and Technology. Fips pub 197. advanced encryption standard (AES). 2001. (Cited on page 2.)

[TH12a] Qizhi Tian and Sorin A. Huss. A general approach to power trace alignment for the assessment of side-channel resistance. In *IEEE Int. Conf. on Intelligent Information Hiding Multimedia Signal Processing*, 2012. (Cited on pages 66, 69 and 72.)

[TH12b] Qizhi Tian and Sorin A. Huss. On clock frequency effects in side channel attacks of symmetric block ciphers. In *IEEE Int. Conf. on New Technologies, Mobility and Security*, 2012. (Cited on pages 53, 66, 84 and 85.)

[TH12c] Qizhi Tian and Sorin A. Huss. On the attack of misaligned traces by power analysis methods. In *IEEE Int. Conf. on Computer Engineering and Systems*, 2012. (Cited on pages 94 and 96.)

[TH12d] Qizhi Tian and Sorin A. Huss. Power amount analysis: An efficient means to reveal the secrets in cryptosystems. In *Int. Journal of Cyber-Security and Digital Forensics*, volume 1, pages 99–114, 2012. (Cited on pages 41, 44, 51, 53 and 95.)

[TH12e] Qizhi Tian and Sorin A. Huss. Power amount analysis: Another way to understand power traces in side channel attacks. In *IEEE Int. Conf. on Digital Information Processing and Communications*, 2012. (Cited on pages 5, 16, 41, 44, 51, 53 and 95.)

[THH12] Qizhi Tian, Annelie Heuser, and Sorin A. Huss. A novel analysis method for assessing the side-channel resistance of cryptosystems. In *IEEE Int. Conf. on Intelligent Information Hiding Multimedia Signal Processing*, 2012. (Cited on pages 55, 56, 58 and 62.)

[TSSH12] Qizhi Tian, Abdulhadi Shoufan, Marc Stoettinger, and Sorin A. Huss. Power trace alignment for cryptosystems featuring random frequency countermeasures. In *IEEE Int. Conf. on Digital Information Processing and Communications*, 2012. (Cited on pages 66, 69 and 71.)

[TV05] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005. (Cited on page 43.)

[WH10] Neil H.E. Weste and David Money Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley, 2010. (Cited on pages 13 and 15.)

[WWB11] Jasper G. J. Van Woudenberg, Marc F. Witteman, and Bram Bakker. Improving differential power analysis by elastic alignment. In *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2011. (Cited on page 65.)

[YWV+05] Shengqi Yang, Wayne Wolf, Narayanan Vijaykrishnan, Dimitrios N. Serpanos, and Yuan Xie. Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach. In *ACM/IEEE Design, Automation, and Test in Europe(DATE)*, pages 64–69, 2005. (Cited on page 36.)

Appendix A: List of Publications

A.1 List of Publications

1. Qizhi Tian and Sorin A. Huss, On the Attack of Misaligned Traces by Power Analysis Methods, IEEE Eighth Int. Conf. on Computer Engineering & Systems (ICCES2012), November 2012.
2. Qizhi Tian and Sorin A. Huss, Power Amount Analysis: An efficient Means to Reveal the Secrets in Cryptosystems, Int. Journal of Cyber-Security and Digital Forensics, October 2012.
3. Qizhi Tian, Annelie Heuser, and Sorin A. Huss, A Novel Analysis Method for Assessing the Side-Channel Resistance of Cryptosystems, IEEE Eighth Int. Conf. on Intelligent Information Hiding Multimedia Signal Processing (IIHMSP12), Greece, July 2012.
4. Qizhi Tian and Sorin A. Huss, A General Approach to Power Trace Alignment for the Assessment of Side-Channel Resistance, IEEE Eighth Int. Conf. on Intelligent Information Hiding Multimedia Signal Processing (IIHMSP12), July 2012
5. Qizhi Tian, Abdulhadi Shoufan, Marc Stoettinger, and Sorin A. Huss, Power Trace Alignment for Cryptosystems featuring Random Frequency Countermeasures, IEEE Int. Conf. on Digital Information Processing and Communications (ICDIPC12), July 2012.
6. Qizhi Tian and Sorin A. Huss, Power Amount Analysis: Another Way to Understand Power Traces in Side Channel Attacks, IEEE Second Int. Conf. on Digital Information Processing and Communications (ICDIPC12), July 2012.
7. Qizhi Tian and Sorin A. Huss, On Clock Frequency Effects in Side Channel Attacks of Symmetric Block Ciphers, IEEE Fifth Int. Conf. on New Technologies, Mobility and Security (NTMS12), May 2012.
8. Marc Stoettinger, Sunil Malipatlolla, and Qizhi Tian, Survey of Methods to Improve

Side-Channel Resistance on Partial Reconfigurable Platforms, Design Methodologies for Secure Embedded Systems, November 2010.

Appendix B: Experimental Results

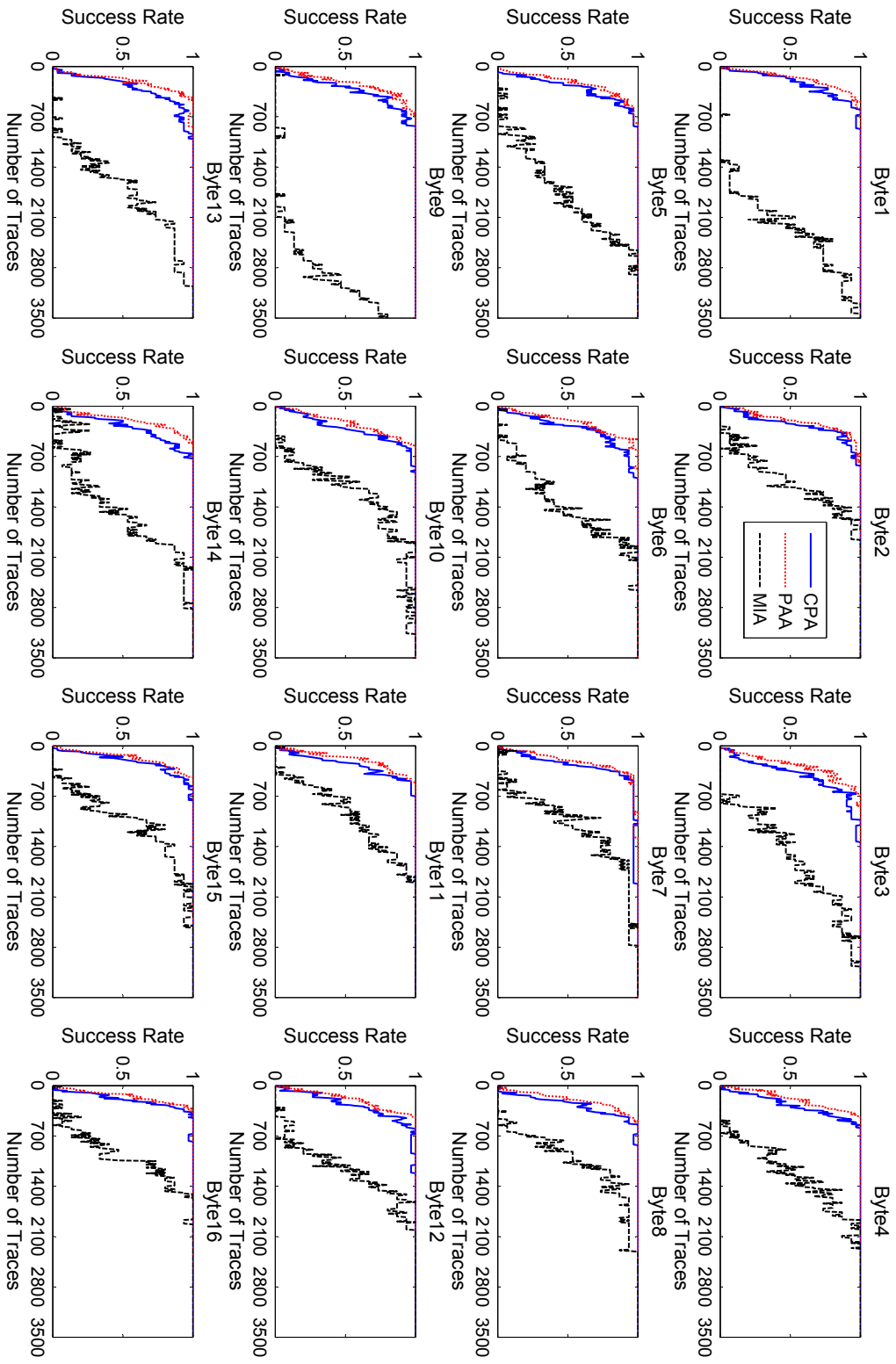

Figure B.1: SR Comparison of CPA, PAA, and MIA

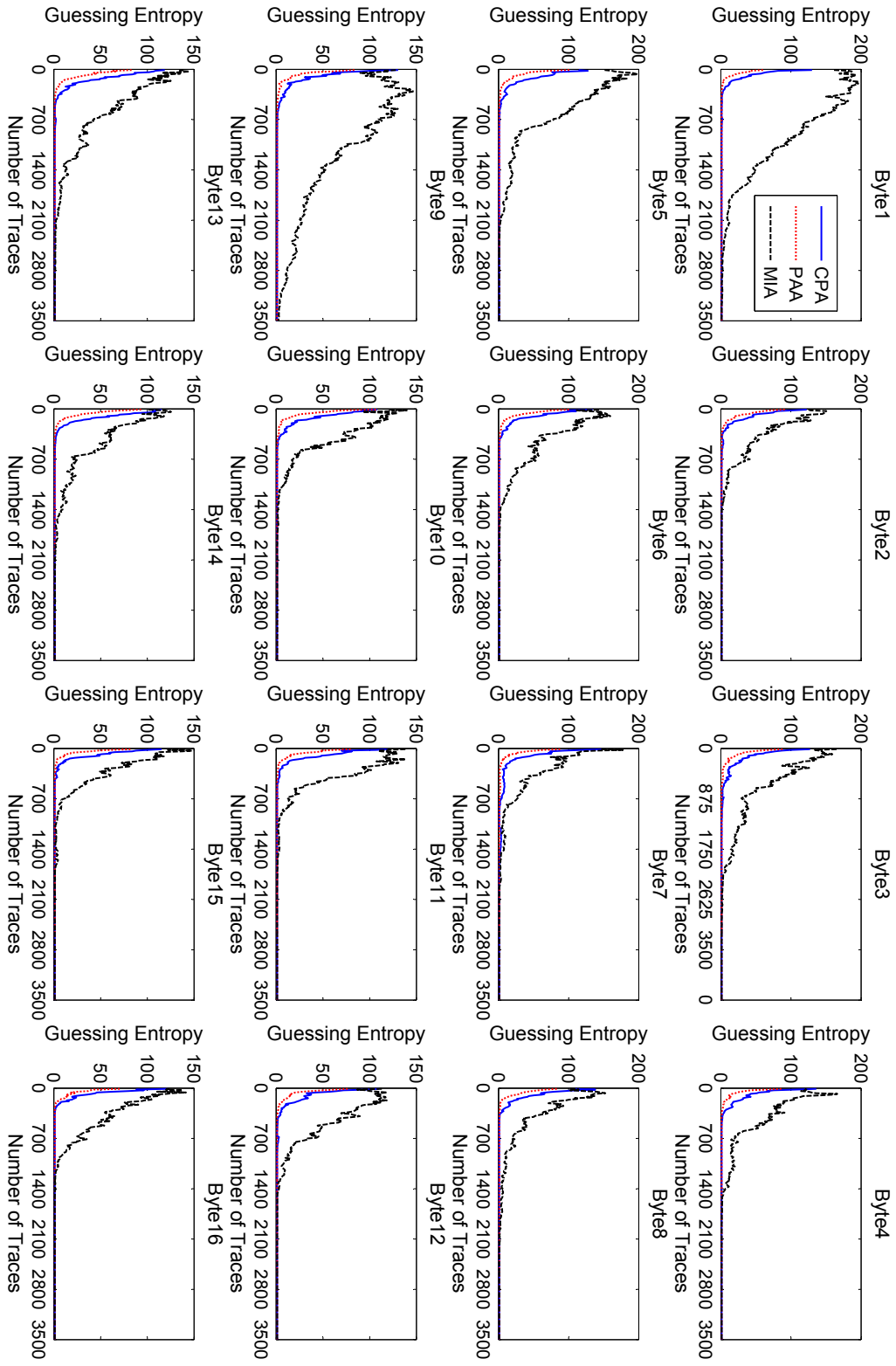

Figure B.2: GE Comparison of CPA, PAA, and MIA

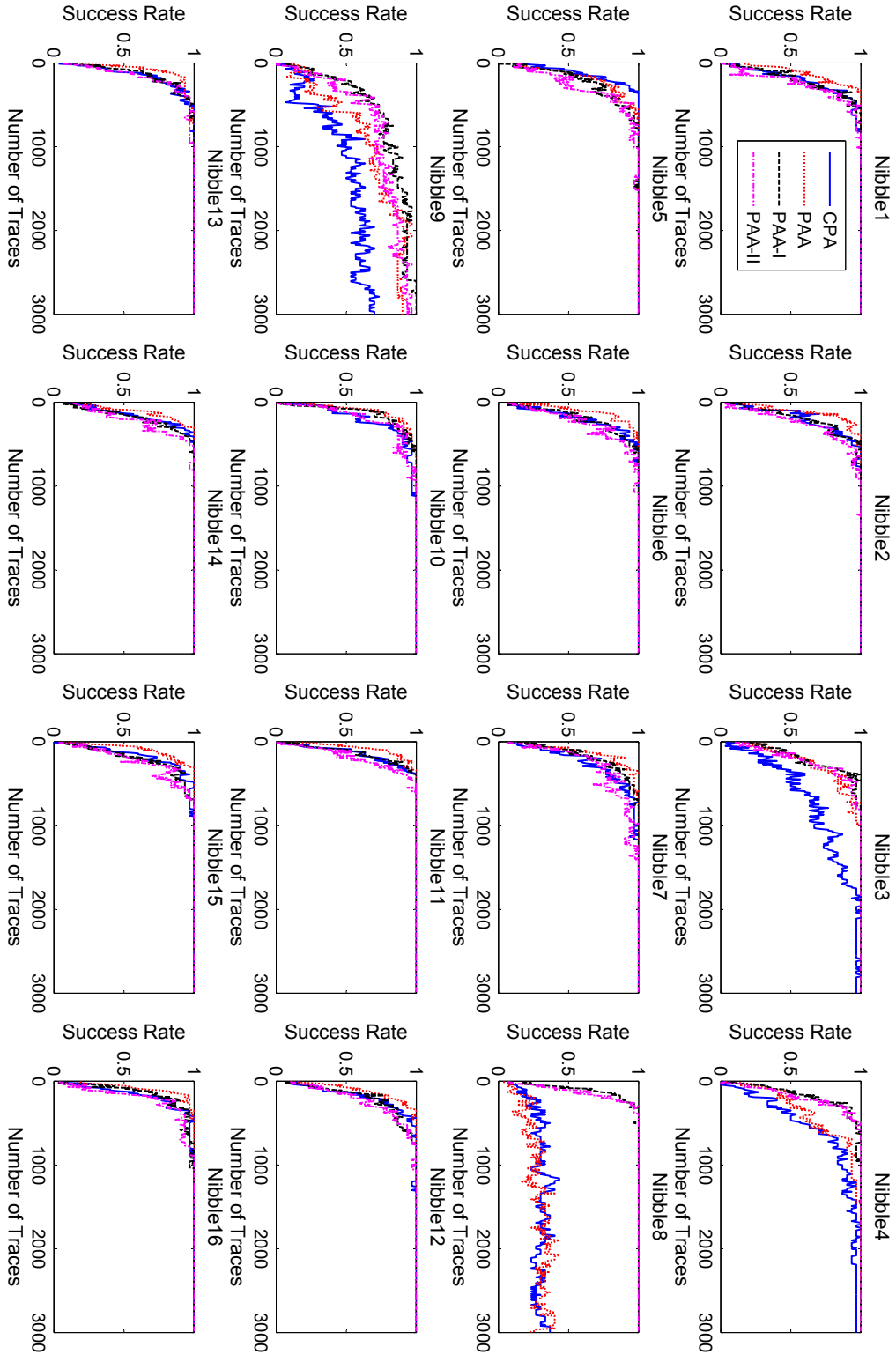

Figure B.3: SR Comparison of CPA, PAA, PAA-I, and PAA-II

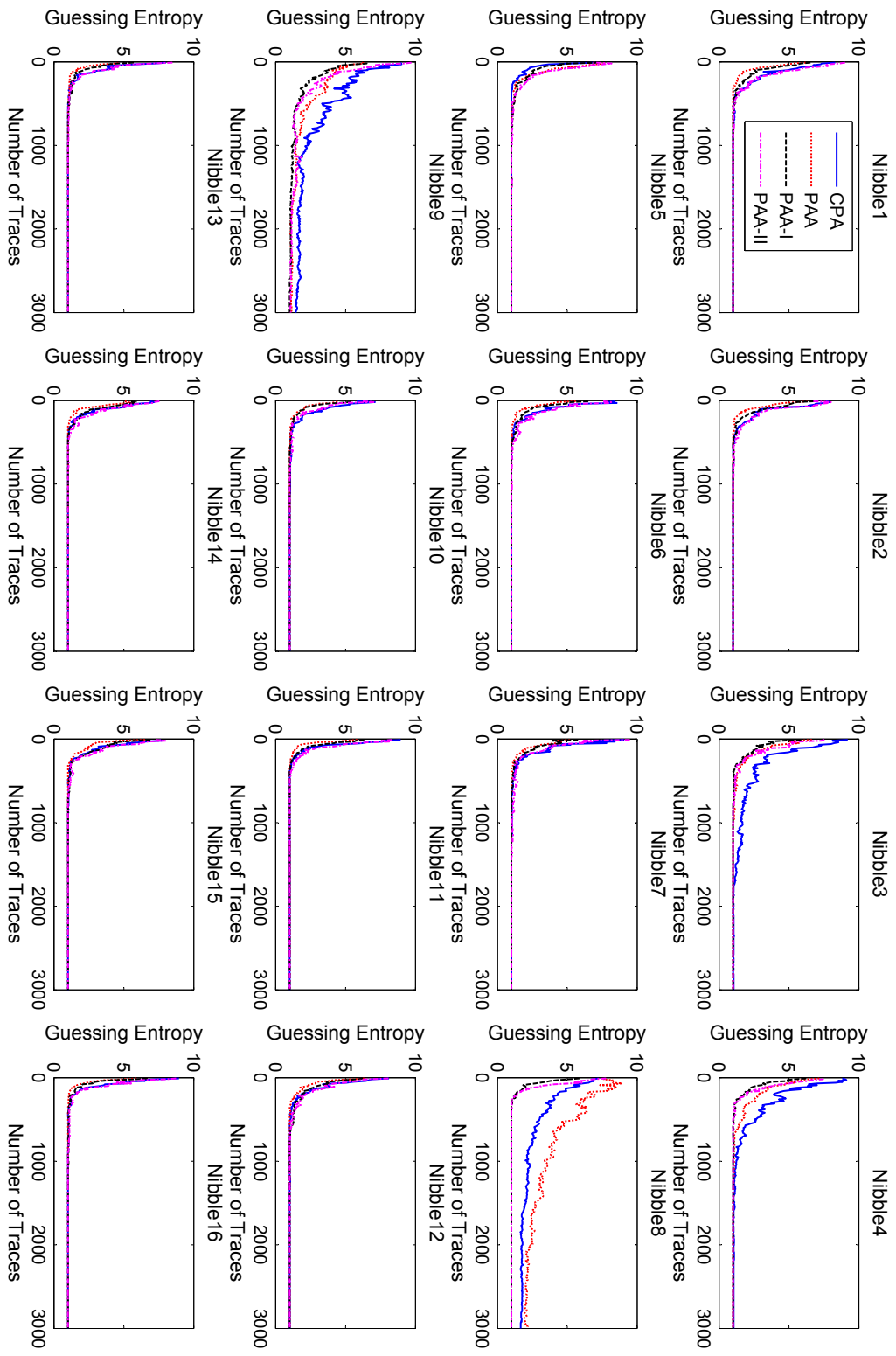

Figure B.4: GE Comparison of CPA, PAA, PAA-I, and PAA-II

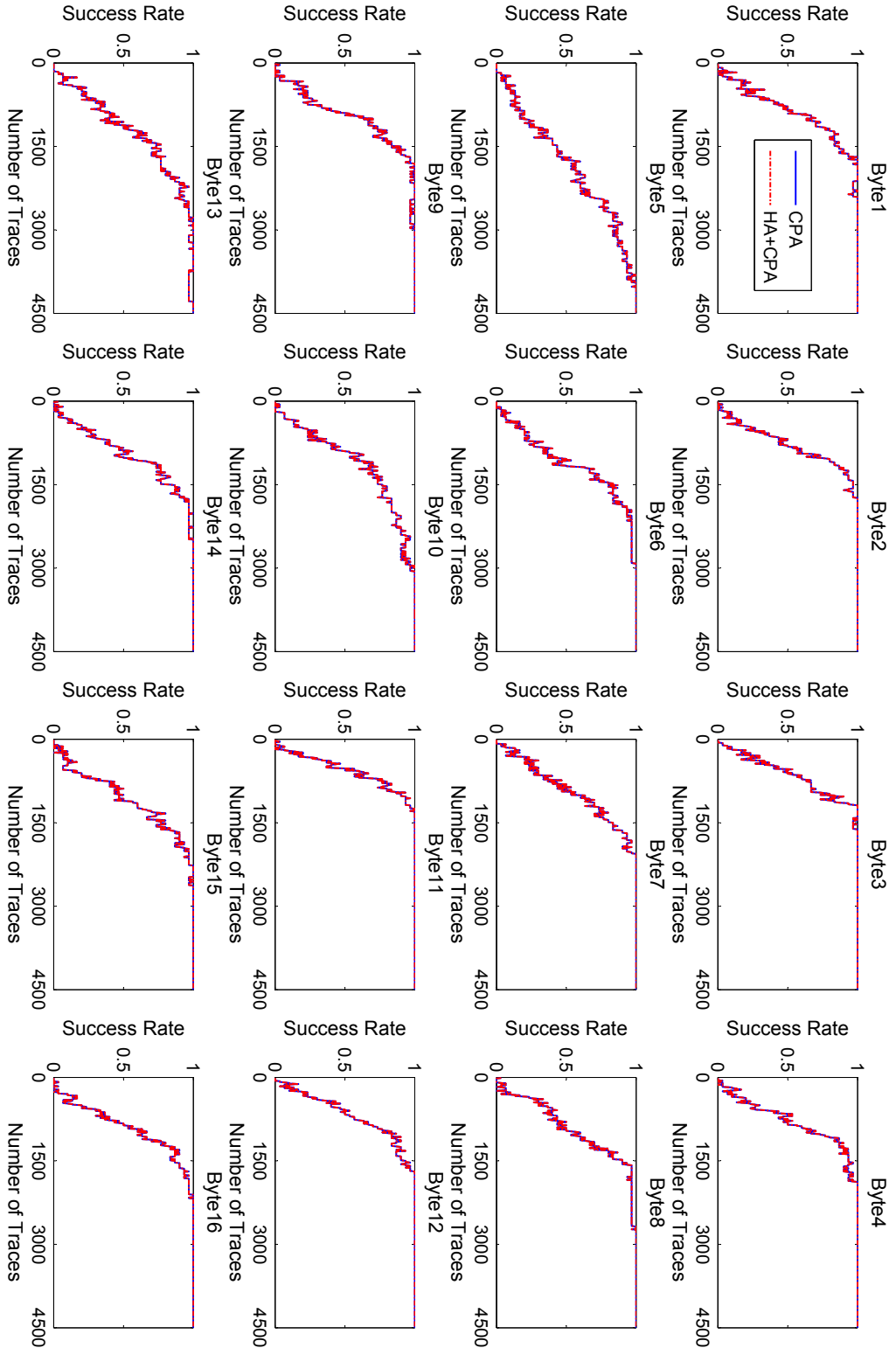

Figure B.5: SR Comparison Before and After HA

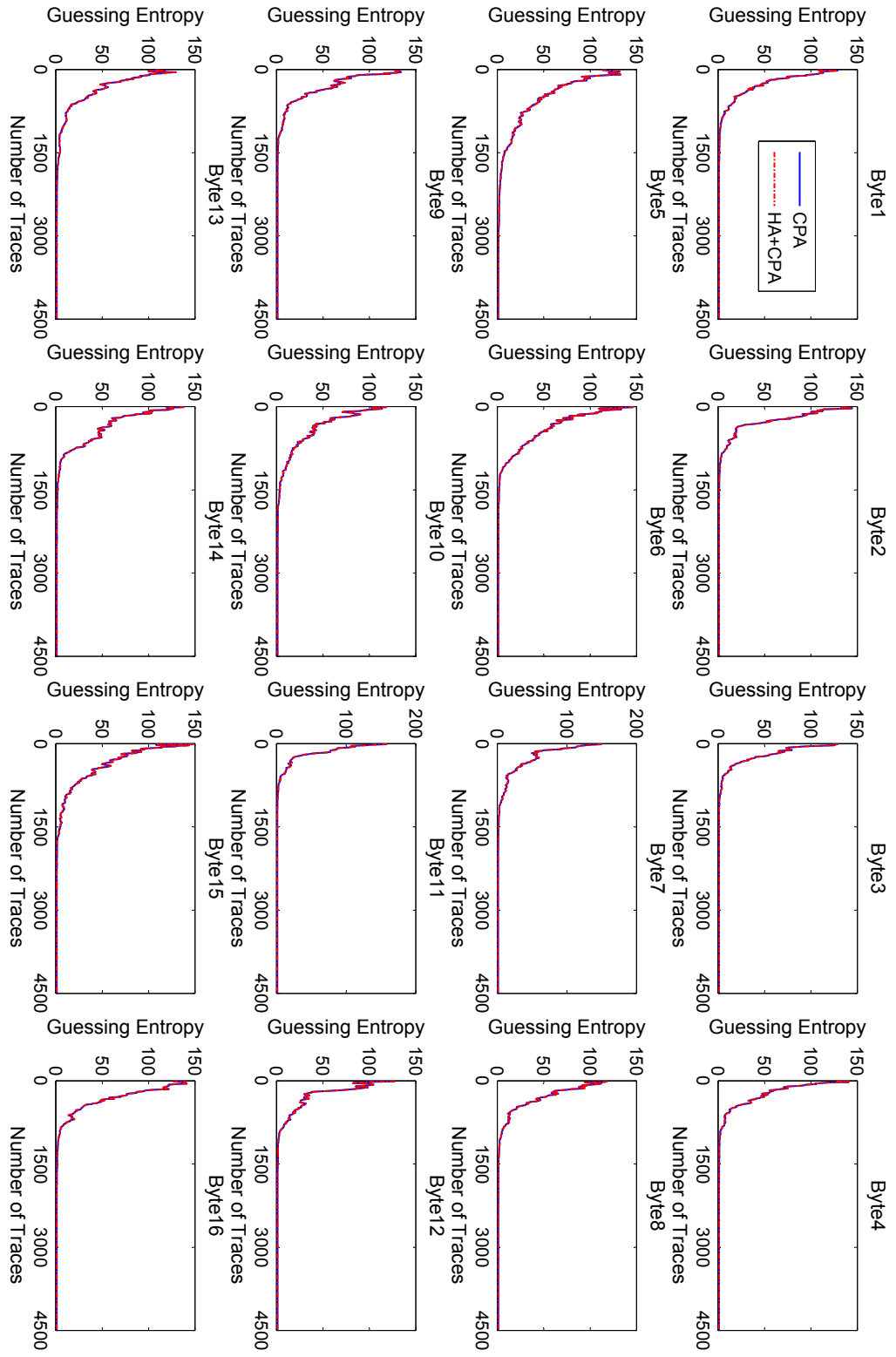

Figure B.6: GE Comparison Before and After HA

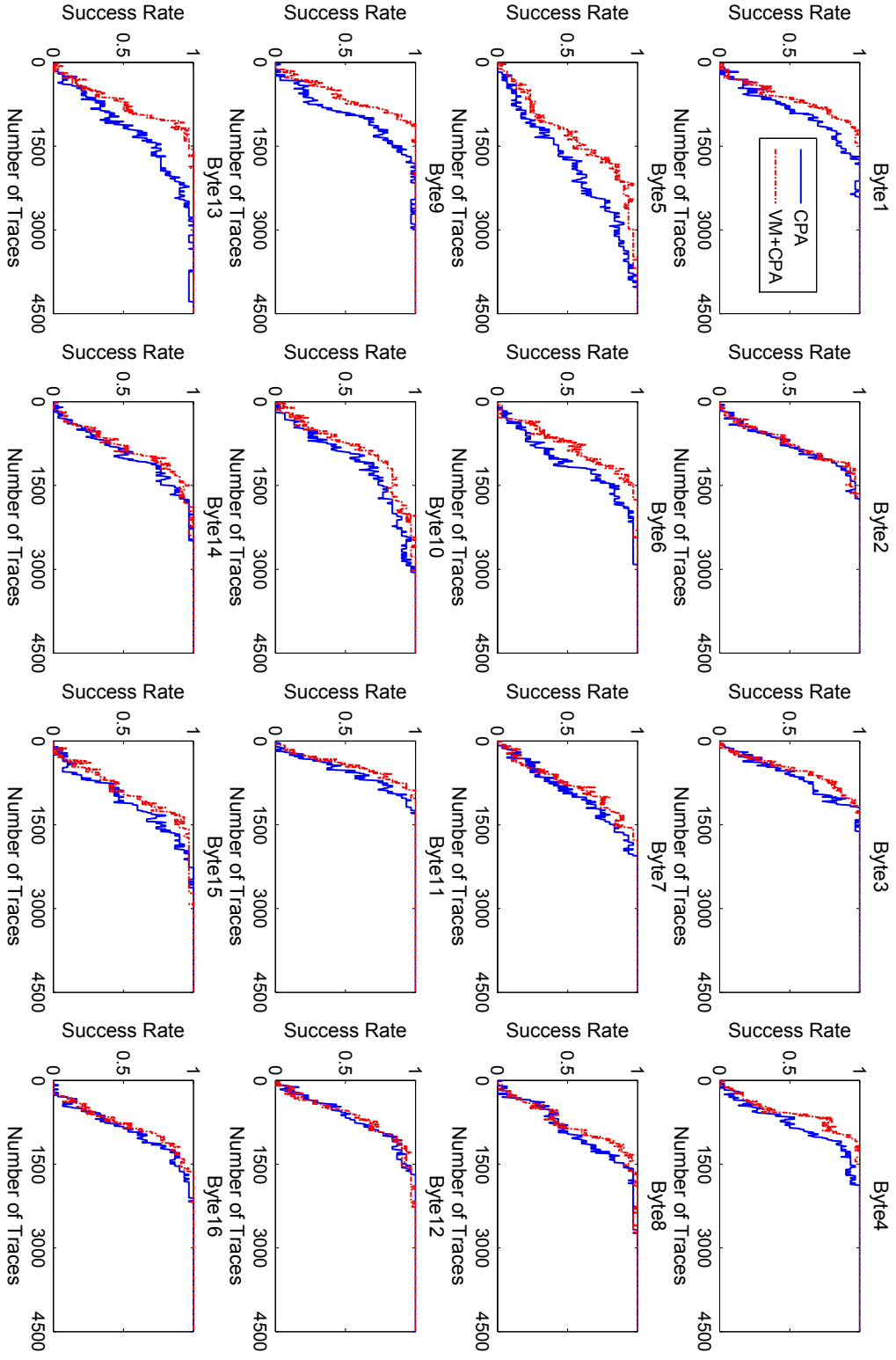

Figure B.7: SR Comparison Before and After VM

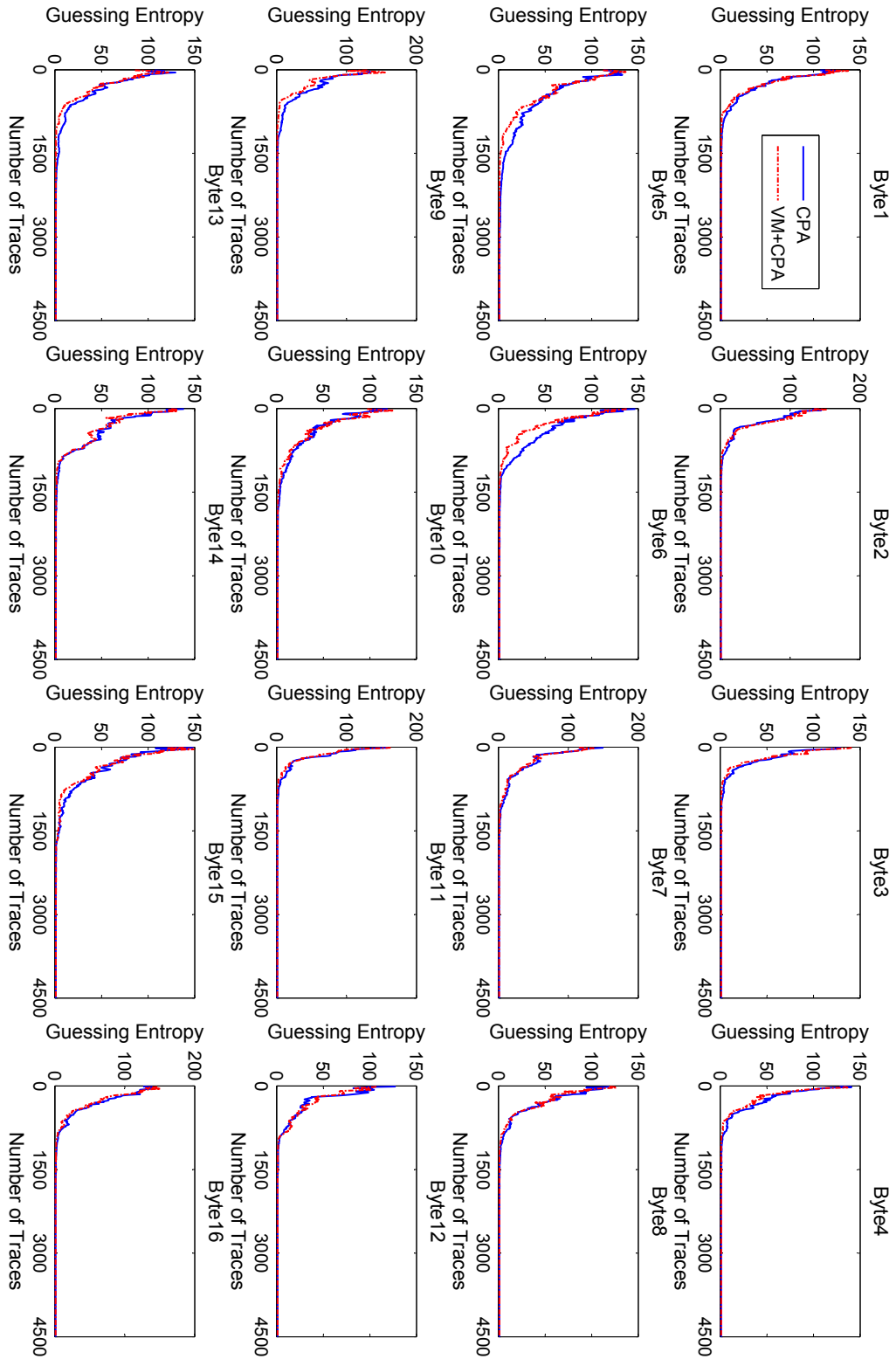

Figure B.8: GE Comparison Before and After VM

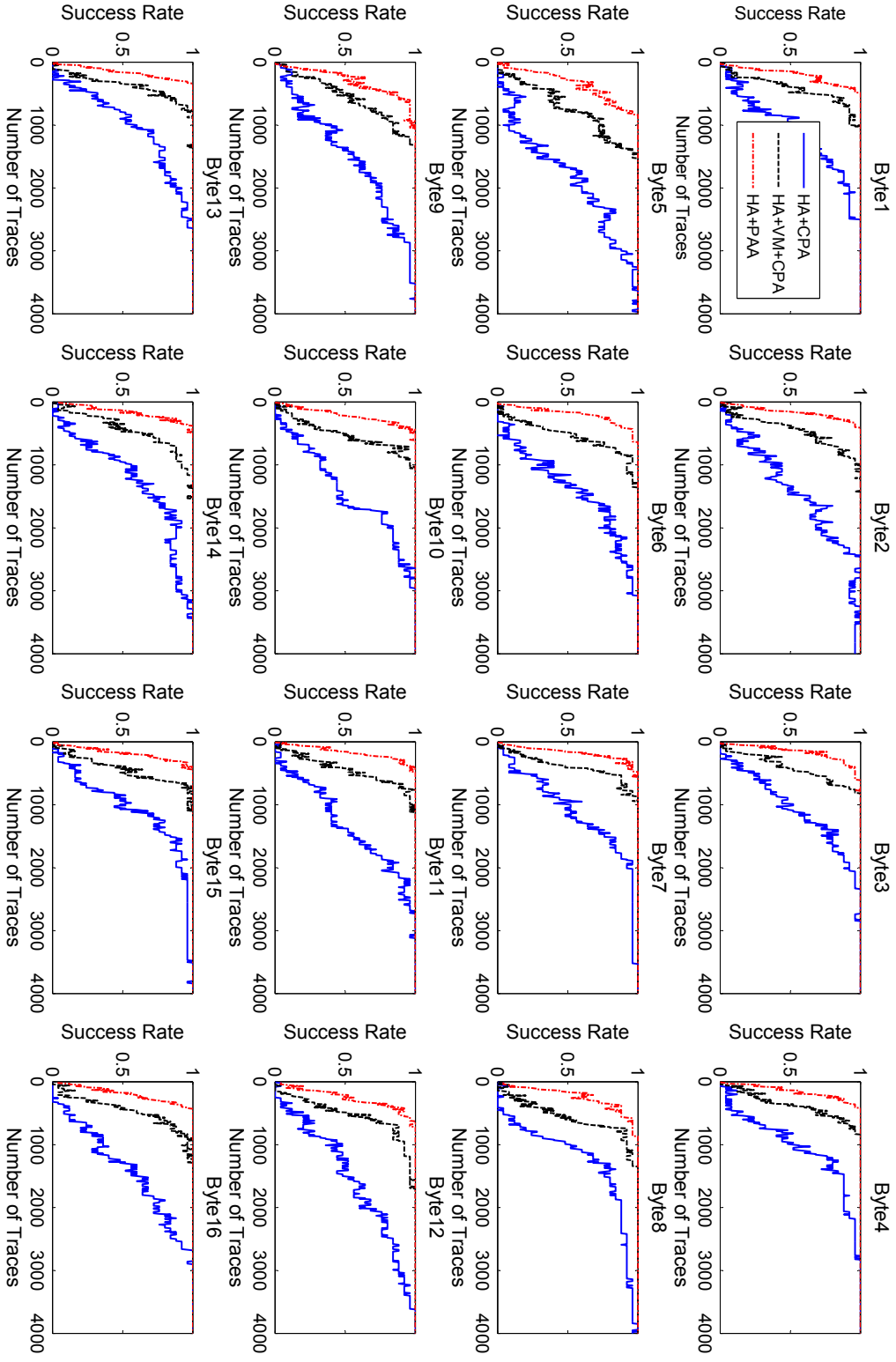

Figure B.9: SR Comparison of Template Based Horizontal Alignment

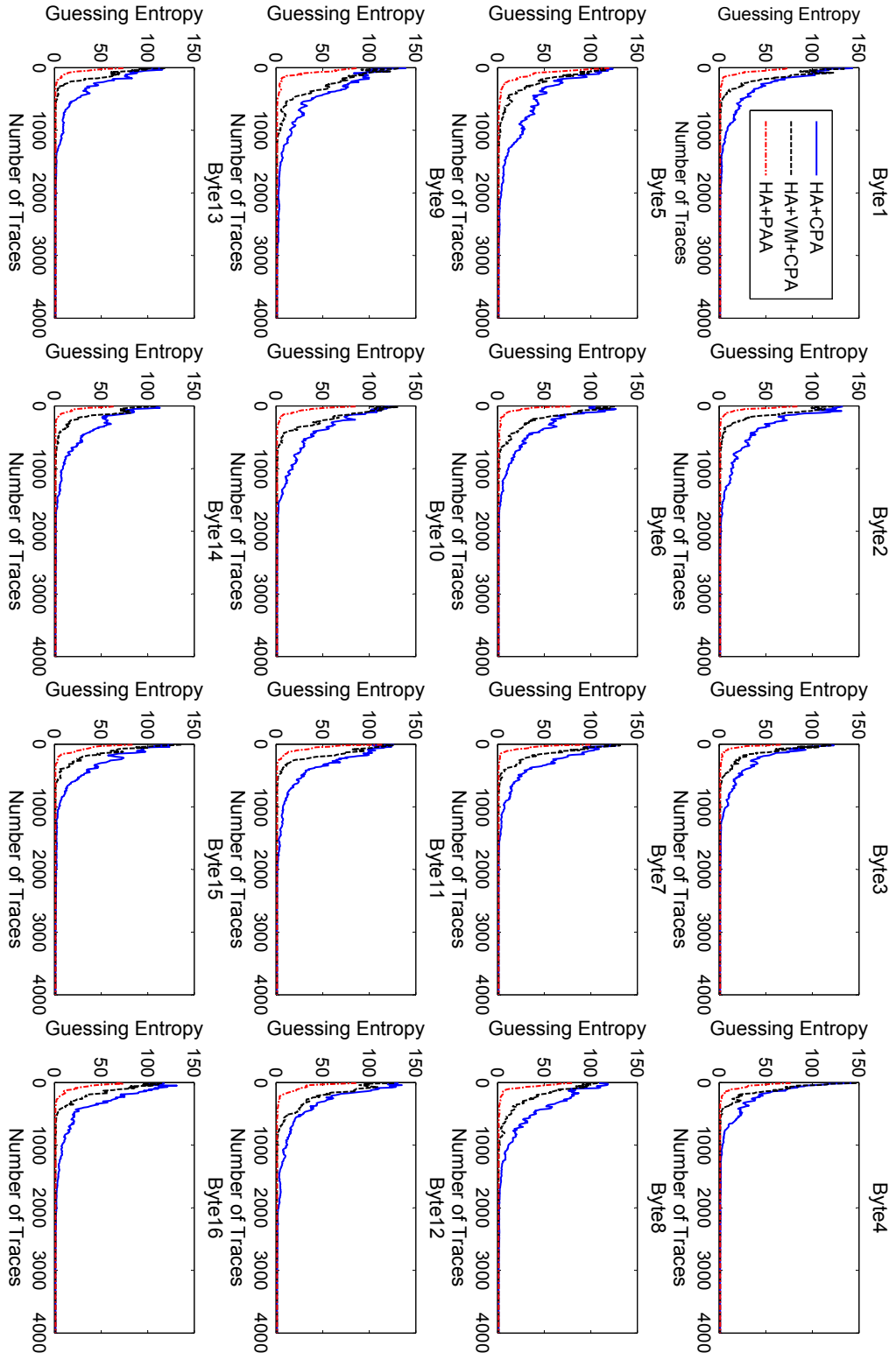

Figure B.10: GE Comparison of Template Based Horizontal Alignment

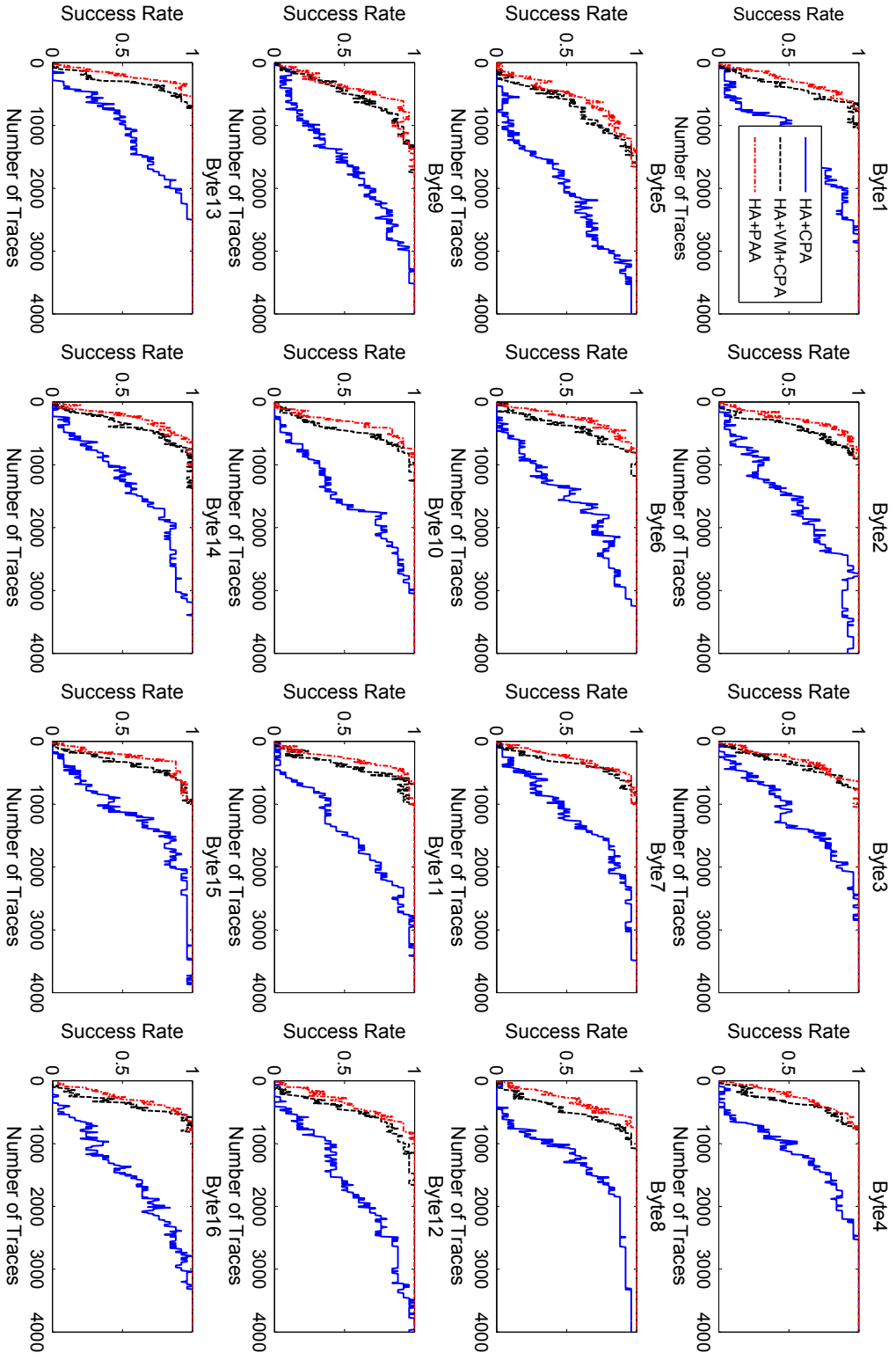

Figure B.11: SR Comparison of Peak Position Based Horizontal Alignment

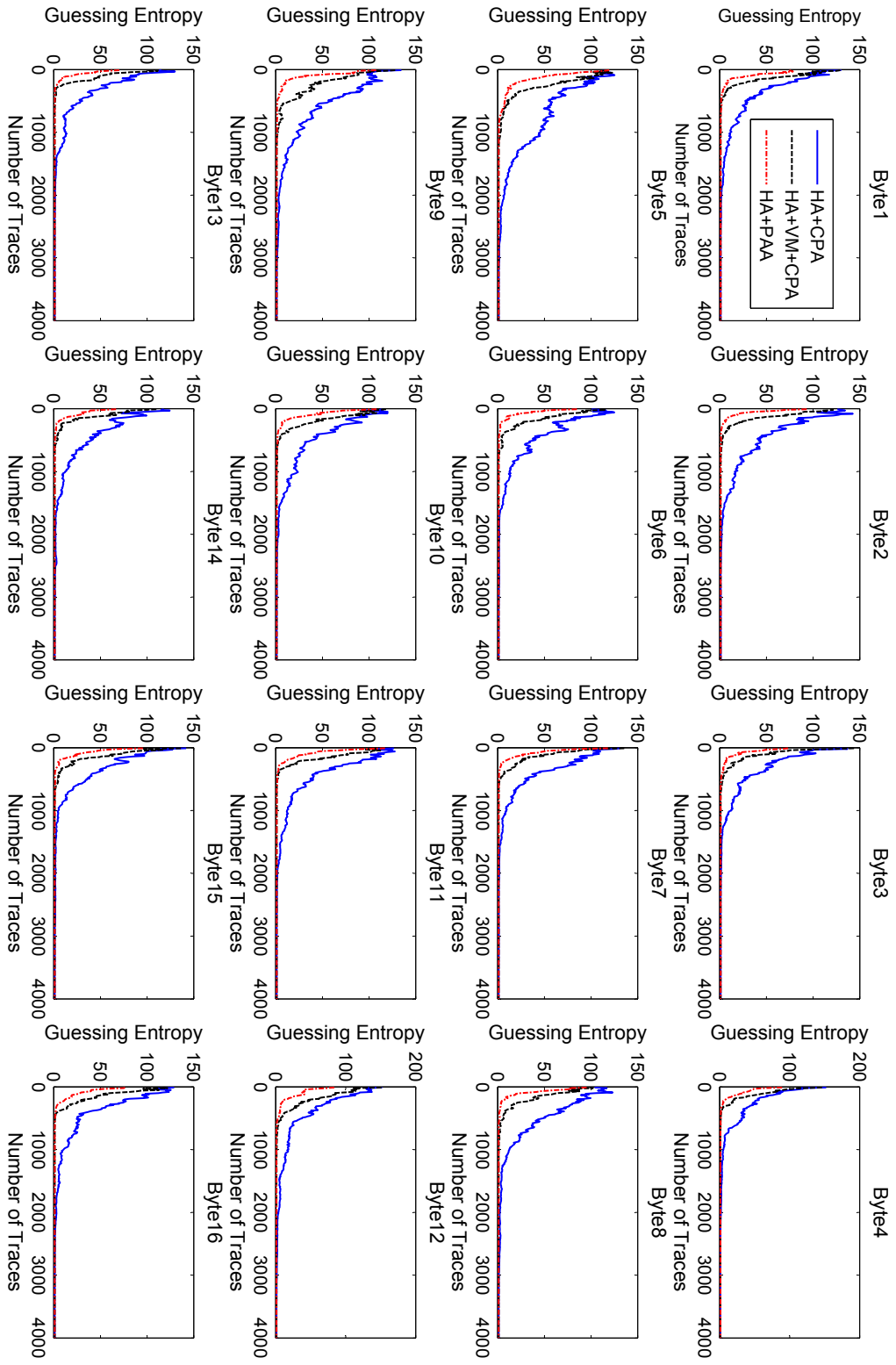

Figure B.12: GE Comparison of Peak Position Based Horizontal Alignment

Qizhi Tian

Gutenbergstr. 5c- 64289 Darmstadt - Germany - Tel: +49 6151 16-6710 - Email: tian_tqz@hotmail.com

Curriculum Vitae

Personal Data

Qizhi Tian

Gutenbergstr. 5c App.26
64289 Darmstadt, Germany

E-Mail: tian_tqz@hotmail.com
Tel: +49 6151 16-6710

Date of Birth: 27.05.1982
Nationality: Chinese

Education

Ph.D.

Technische Universität Darmstadt — Darmstadt, Germany — 2008 - 2013

- Advisor: Prof. Dr.-Ing. Sorin A. Huss
- Institute: Integrated Circuits and Systems Lab, Computer Science Department
- Research Topic: FPGA-based Hardware Security and Side Channel Analysis
- Thesis Title: Novel Power Trace Processing Methods for Side-Channel Analysis of Cryptosystems

Master

Beijing University of Posts and Telecommunications (BUPT) — Beijing, China — 2005 - 2008

- Major: Electronic and Mechanical Engineering
- Topics: FPGA-based Hardware System Design

Bachelor

Beijing University of Posts and Telecommunications — Beijing, China — 2001 - 2005

- Major: Automation
- Relevant Courses: Automation, Digital/Analog Circuits, Theory of Telecommunication, Digital Signal Processing, Signal and System, etc.

Social Experience

Vice President of Association of Chinese Computer Scientists in Germany — 2010 - Present
Vice President of Post Graduate Union in BUPT — 2006 – 2008

Interests

Cycling tours; remote control models; machinery repairing.

Language

Chinese: mother language.
English: good at listening, speaking, reading, writing.
German: B1 class

References

Prof. Dr.-Ing. Sorin A. Huss
Integrated Circuits and Systems Lab
Technische Universität Darmstadt
64289 Darmstadt, Germany
E-Mail: huss@iss.tu-darmstadt.de
Tel: +49-6151-163980

Dr. Abdulhadi Shoufan (Assistant Professor)
Information Security & ECE
Khalifa University
P.O.Box: 1227788, Abu Dhabi,UAE.
E-Mail: abdulhadi.shoufan@kustar.ac.ae
Tele: +971-2-5018574