Technische Universität München

Lehrstuhl für Sicherheit in der Informationstechnik
an der Fakultät für Elektrotechnik und Informationstechnik

High Resolution EM Side Channel Attacks with Multiple Measurement Probes

Robert M. Specht

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines **Doktor-Ingenieurs (Dr.-Ing.)** genehmigten Dissertation.

Vorsitzender:	Prof. Dr-Ing Dr. h.c. Ralph Kennel
Prüfer der Dissertation:	
	1. Prof. Dr.-Ing. Georg Sigl
	2. Prof. Dr.-Ing. Klaus Diepold

Die Dissertation wurde am 23.01.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 02.11.2019 angenommen.

Zusammenfassung

Viele Anwendungen, wie z.B. Payment, Pay-TV und Gesundheitsanwendungen, verwenden Kryptographie um Vertraulichkeit, Integrität und Authentizität sensibler Informationen zu gewährleisten. Die sensiblen Informationen und das schützende kryptographische Schlüsselmaterial selbst sind bevorzugte Ziele von Angriffen. Eine Möglichkeit, diese Angriffe durchzuführen, sind Seitenkanalangriffe. Bei diesen wird die Implementierung eines kryptographischen Algorithmus angegriffen, nicht aber der Algorithmus selbst, z.B. indem der Stromverbrauch oder die elektromagnetische (EM) Strahlung während der kryptographischen Operation beobachtet wird. Lokalisierte EM-Messungen gehören zu einer mächtigen Klasse von Seitenkanalangriffen und verwenden kleine elektromagnetische Sonden von wenigen hundert Mikrometern Innendurchmesser. Durch den kleinen Durchmesser können die Sonden sehr nah an der Chip-Oberfläche platziert werden. Dadurch isolieren die kleinen Messsonden (und somit das aufgenommene Signal) Chip-Bereiche in deren Umgebung vom Rest des Chips. Damit erhöhen sie das Signal-Rausch-Verhältnis (SNR), verglichen zu Strom- oder globalen EM-Messungen, und ermöglichen es mehrere Sonden über dem Chip zu platzieren.

In dieser Arbeit wird der Fokus auf Messungen mit mehreren Messsonden gelegt. Dabei ist es das Ziel, zu ermitteln, ob mehrere Sonden einen mächtigeren Angriff ermöglichen als eine einzelne Sonde. Hierfür werden Angriffe gegen eine symmetrische und eine asymmetrische Chiffre durchgeführt. Diese Angriffe wurden ausgewählt, da sich die Angriffsszenarien bei symmetrischen und asymmetrischen Verschlüsselungsalgorithmen stark unterscheiden können. Die meisten symmetrischen Algorithmen ermöglichen die Erfassung von vielen (tausend) Traces mit demselben Schlüssel, im Gegensatz dazu können asymmetrische Algorithmen durch Gegenmaßnahmen einen Angreifer auf einen Trace beschränken. Um das Potential mehrerer Sonden zu ermitteln, vergleiche ich das Ergebnis der Kombination von mehreren Sonden mit den Ergebnissen der einzelnen Sonden und Strommessungen.

Ein Angriff besteht im Allgemeinen aus 3 Schritten, der Profilingphase, dem Preprocessing und der Auswertung. Während der Profilingphase wird das Verhalten eines Chips mit bekanntem Schlüssel gemessen, sodass der Preprocessing- und Angriffsschritt mit Hilfe des Profilings gezielt angepasst werden können. Angriffe, die eine Profilingphase beinhalten, gehören zu den mächtigsten Seitenkanalangriffen. Trotz Profiling und Preprocessing sind Seitenkanalmessungen (und auch lokalisierte EM-Messungen) typischerweise Messungen mit niedrigem SNR. Deshalb ist es wichtig das Preprocessing so anzupassen, dass idealerweise nur die Informationen über den Schlüssel aus den Traces extrahiert werden und das Rauschen vernachlässigt wird, was zu einer Erhöhung des SNR führt. Hierfür werden hier die maschinellen Lerntechniken Diskriminanzanalyse (LDA) und Hauptkomponentenanalyse (PCA) als Preprocessingtechniken verwendet.

Darüber hinaus besteht die Herausforderung bei Messungen mit mehreren Sonden darin, die Informationen der Messdaten effizient zu kombinieren. Diese Kombination kann in verschiedenen Phasen des Angriffs durchgeführt werden. Bei der Analyse der symmetrischen Chiffre, kombiniere ich die Informationen mehrerer Sonden während der profilierten Preprocessingphase mithilfe von LDA. Es wird gezeigt, dass der Angriff auf die symmetrische Implementierung von der Kombination mehrerer Sonden profitiert und weniger Traces gebraucht werden, um den Schlüssel zu extrahieren. Im Vergleich zwischen der Analyse mit einzelnen Sonden und dem kombinierten Ansatz, sinkt die Anzahl der benötigten Traces um den Faktor 5,7. In Relation zum Angriff mit Strommessungen wurde die Anzahl der Traces um den Faktor 238 reduziert. Zusätzlich verbessere ich den State-of-the-Art Angriff, der in diesem Kapitel verwendet wird, um den Faktor 4,2. Darüber hinaus werden Beispiele gezeigt, wie die Informationen mehrerer Sonden bei diesem Angriff kombiniert werden und dass LDA in der Lage ist, die Informationen mehrerer Sonden zu kombinieren.

Während des Angriffs mit nur einem Trace gegen eine asymmetrische Chiffre, wird ein anderes Maß für die Effektivität des Angriffs wie eben gewählt. Dadurch, dass nur ein Trace für die Analyse zur Verfügung steht, wird die verbleibende Anzahl an Schlüsselbits, die für einen erfolgreichen Angriff durch Brute-force ermittelt werden müssten geschätzt, die sog. Brute-Force-Komplexität. Die Brute-Force-Komplexität eines nicht profilierten Angriffs kann in

59 % der Fälle auf 32 Bit oder weniger gesenkt werden, verglichen mit 0 % der Fälle zum Ansatz von Heyszl et al. [Hey14] für eine einzelne Sonde. In dieser Messung wird PCA als Preprocessing- und Dimensionsreduktionsschritt eingesetzt. Mehrere Sonden werden nach der Anwendung von PCA auf die einzelnen Sonden miteinander kombiniert. Allerdings erreicht die Kombination mehrerer Sonden nur vergleichbare Ergebnisse, wie die beste Einzelsonde. Um die Leistung des unprofilierten Ansatzes zu vergleichen, wird zusätzlich ein profilierter Angriff durchgeführt. Damit wird der Mittelwert der Brute-Force-Komplexität von 50 Bit für die beste einzelne Sonde auf 44 Bit im kombinierten Fall reduziert. Diese Verbesserung kann jedoch nur in Kombination mit profilierten Angriffen gezeigt werden, sodass ein Angreifer Zugriff auf ein Gerät mit bekanntem Schlüssel haben muss, um Trainingsdaten zu sammeln.

Abstract

Many applications, e.g., payment, pay-tv, and health care use cryptography to ensure confidentiality, integrity and authenticity of sensitive information. This sensitive information and the protective cryptographic key material are often goals of attacks. One example are side channel attacks, which focus on attacking the implementation of a cryptographic algorithm and not the algorithm itself. Side channel attacks use the observation of the power consumption or the Electro-Magnetic (EM) radiation to extract the cryptographic key. Localized EM measurements belong to a powerful class of side channel measurements. Localized EM measurements use small electro-magnetic probes, e.g., 150 µm inner diameter, which enable a close-to-die distance and the isolation of logic-parts, thereby increasing the SNR. Due to the small probe size it is possible to place multiple probes above the die.

In this thesis I carry out simultaneous measurements with multiple measurement probes. Afterwards, I assess if multiple probes lead to a more powerful attack, which I evaluate using an attack with multiple localized EM probes against an implementation of one symmetric and one asymmetric cipher. The attack scenarios may differ significantly for symmetric and asymmetric algorithms. Most symmetric algorithms allow the capturing of many (thousands) traces with the same key. However, implemented countermeasures in the asymmetric case can restrict an attacker to one side channel observation, e.g., blinding countermeasures. To demonstrate the efficiency of multiple probes, I compare the result of using multiple probes to the results of separate evaluations of each probe and the result of the power side channel attack.
An attack commonly consists out of 3 steps, profiling, preprocessing, and the attack evaluation. During the profiling step the behavior of a device is observed with practical measurements and a known secret; hence, the preprocessing and attack step can be specifically adapted with profiling. Attacks, which include a profiling step belong to the most powerful side channel attacks. Nonetheless, side channel measurements are typically low-SNR measurements. Thus,

a preprocessing step can be implemented, which ideally extracts the information about the key and neglects the noise, which results in increasing the SNR. To achieve this increase in SNR, I apply the machine learning techniques Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) in this thesis. Furthermore, the challenge for multiprobe measurements is to efficiently combine the information of the measurement data. The combination can be performed at different stages of the attack. When attacking the symmetric cipher I combine the information of multiple probes during the profiled preprocessing phase. I show that the attack on the symmetric implementation can benefit from the combination of multiple probes. I demonstrate that, compared to analyzing the probes separately, the combined multiple probe attack reduces the number of required traces by factor 5.7. Compared to the power side channel attack, the number of required traces was reduced by factor 238. Additionally, I enhance the current state-of-the-art profiled attack, used in this chapter, to require factor 4.2 less number of traces to break the implementation.
Furthermore, I show examples on how the information of multiple probes is combined during this attack and that LDA is capable of combining the information of multiple probes.

During the single trace attack against an asymmetric cipher, I use the remaining number of key bits, which have to be brute-forced for a successful attack, the brute force complexity as a measure for the effectiveness of the attack. As a preprocessing and dimension reduction step, I use PCA in the attack against the asymmetric cipher.
I can lower the brute force complexity of the single trace unprofiled attack against an asymmetric cryptographic implementation in 59 % of cases to 32 bit or lower with the application of PCA, compared to 0 % of cases to the approach of Heyszl et al. [Hey14] for a single probe. When combining multiple probes, the combination reaches comparable results to the best single probe. To compare the performance of the unprofiled approach, I carry out a profiled attack. Therefore I reduce the mean in the keys brute force complexity from 50 bits for the best single probe to 44 bits in the combined case. However, I can only show this improvement by combination for profiled attacks, such that an attacker must have access to a device with known key to collect training data.

Contents

Chapter 1

Introduction

Protecting sensitive data against unauthorized access and manipulation is of central importance in many applications, e.g., in pay-tv, payment or health. To protect sensitive data, encryption algorithms are used to provide confidentiality, integrity and authenticity for the data. Commonly in cryptography the encrypted transmission of data is modeled by a communication between the two parties, Alice and Bob, depicted in Figure 1.1. These two parties want to communicate secure. The attacker (Eve), who can read (or even modify) all messages between Alice and Bob, wants to get knowledge of the messages by observing the ciphertext. The goal of an encryption algorithm (cipher) is that Eve cannot gain any information about the message, without knowing the secret key. To ensure secure encryption algorithms Kerckhoffs described six design principles in 1883. The todays most relevant principle is:

"A cryptosystem must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience." [Ker83]

Hence, the only secret has to be the key and everything else e.g., the algorithm itself should be assumed to be public knowledge. Multiple publications show that proprietary ciphers are likely to be reverse engineered and possibly broken, e.g., A5/1, HiTag2 and Crypto [Noh08, Ver12, Bih00].
The exact mathematical backgrounds of these algorithms were not publicly known, until a reverse engineering. After the knowledge was pubilc, the encryption algorithms did not withstand modern cryptoanalytic attacks. Cryptoanalytic attacks belong to logical attacks. During a logical attack an attacker can access and observe one (or more) communication interfaces and the received/transmitted messages, shown in Figure 1.1. This kind of attack is applicable to a wide range of systems, especially in case of wireless systems an attacker is capable of observing (or even modifying) communication. However, most (standardized) modern cryptographic algorithms were intensively

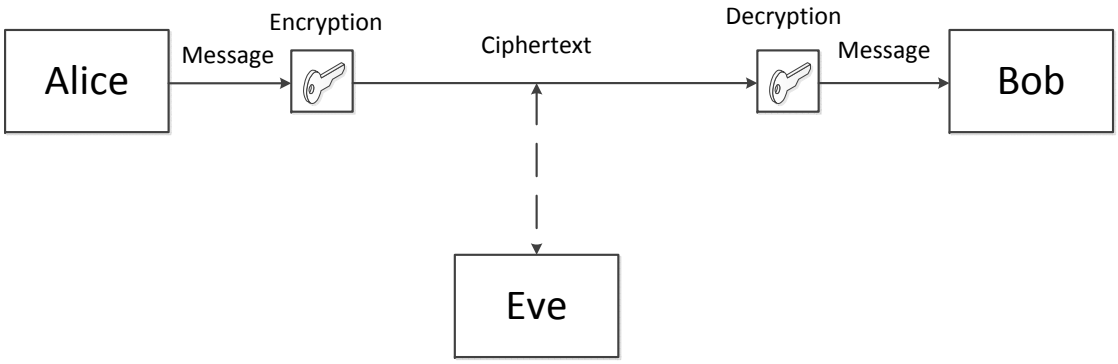


Figure 1.1: Encrypted communication of Alice and Bob [Kat14]

analyzed in this context and seem to be resistant against such attacks.

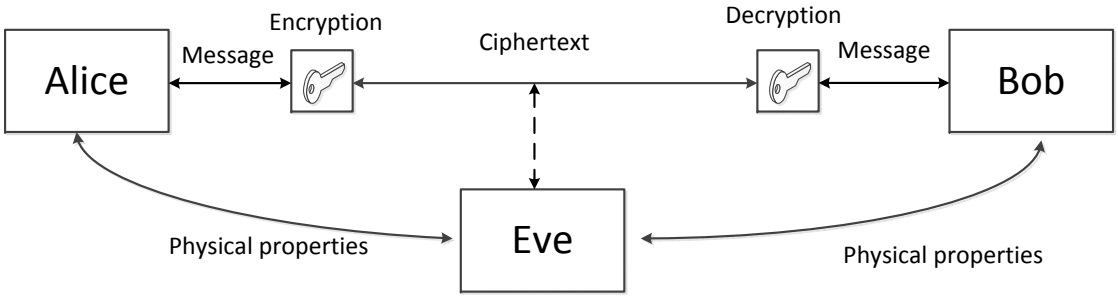


Figure 1.2: Model of a physical attack [Man07]

An even more powerful attack class is called physical attacks. Additionally to a logical attack, the attacker can observe or manipulate physical properties of the system, depicted in Figure 1.2.
In this thesis I focus on side channel attacks, which belong to physical attacks. In the following, I will briefly describe the most important properties of cryptographic algorithms first and physical attacks and side channel attacks afterwards.

1.1 Cryptographic Algorithms

There are mainly two kinds of cryptographic algorithms (ciphers), symmetric and asymmetric algorithms. Both mostly work as a function, which takes a

message (plaintext) and a key as input and returns the encrypted result (ciphertext). One requirement for a cipher is that any information about the message should be only revealed if the used key is known. This should be the case if an attacker has unlimited computing power and has access to some known plain-/ciphertext combinations. If a cipher holds this properties, it is "perfectly secure". The only known perfectly secret cipher is the one-time pad. The one-time pad is created by randomly generating bits with equal probability for zeros and ones, $\Pr(1) = \Pr(0) = 0.5$. This one time pad is XORed onto the plaintext. The practical problem of the one-time pad is the need of a large amount random numbers. Secondly, the key distribution is a practical problem, because the length of the key has to be equal to the message length. These two properties make it unusable in practice. [Kat14]

Thus, modern cryptographic algorithms try not to be perfectly secure, they try to reach the following two security-goals:

1. An attacker has to spend a significant amount of time, or computational complexity to break the encryption, e.g., brute force 112-bit (2^{112} combinations). [Bar15]

2. An attacker can potentially gain access to the encrypted information, however with a very small probability, e.g., $\leq 2^{-112}$

Both properties ensure a practical level of security. [Kat14]

1.1.1 Symmetric Cryptographic Algorithms

For symmetric ciphers the keys for en- and decryption are the same. Two groups of symmetric ciphers are so-called stream ciphers and block ciphers. Stream ciphers create a pseudo-random (indistinguishable from random in case of an unknown key) output sequence, which is combined with an XOR with the message to create the ciphertext, e.g., Salsa20 [Ber08] and Trivium [De 06]. In contrast, block ciphers encrypt a block with a fixed number of bits, e.g., 128 bits. One example for such a cipher is the Advanced Encryption Standard (AES) [NIS01]. Shannon [Sha49] defined two properties of cryptographic algorithms, which are confusion and diffusion. Confusion specifies a complex relationship between each key- and ciphertext-bit. Diffusion specifies a complex relationship between plaintext (and intermediate values) and ciphertext.

Typically block ciphers work based on so-called rounds, which consist out of one or more confusion and diffusion steps. One round is executed multiple times to increase confusion and diffusion to achieve that a change in one input-bit affects multiple (ideally 50 %) output bits. AES implements the confusion by a non-linear substitution function (S-box) and the diffusion by three linear functions. For the purposes in Chapter 4 of this thesis, the S-box function is of central importance and a common target for side channel attacks. Firstly because in the mean for one bit on the input, every second output-bit changes and secondly due to the computational complexity. [Kat14]

1.1.2 Asymmetric Cryptographic Algorithms

For asymmetric ciphers, keys for en- and decryption are different. Nowadays most popular candidates are Rivest, Shamir und Adleman (RSA) and Elliptic Curve Cryptography (ECC). Both are based on a one-way function to en- and decrypt messages. A one-way function states a mathematical problem, which is assumed to be easy to compute in one direction, but is "hard" to invert. "Hard" to invert means there is no known algorithm with polynomial runtime to solve this problem, only with exponential runtime, e.g., ECC is based on the so-called discrete logarithm problem. [Kat14] During the ECC operation, two big numbers have to be multiplied, e.g., 256 bit each. Therefore, special methods are applied, which calculate the result iteratively by multiplying one bit with the other multiplicand at the same time, e.g., double and add. [Men92, Kob87, Kna92] These multiplication algorithms are of central importance for Chapter 5.

1.2 Physical Attacks

Physical attacks belong to the most powerful attacks on embedded systems. Additional to logical attacks, it is possible to change and observe physical properties during the attack, e.g., modify the systems itself. Hence, additionally to the cipher itself, its implementation has to be secure. To be able to perform physical attacks, the attacker requires physical access to the device in most cases. This can be the case, e.g., if the attacker is the owner of the device.
Table 1.1 shows the classification and examples for physical attacks. Physical attacks can be split into passive and active attacks. Active attacks influence

	Active attacks	**Passive attacks**
Non-Invasive	Glitching, Temperature Change, Low Voltage, …	Side-Channel Attacks (Timing Analysis, Power Analysis, …)
Semi-Invasive	Light Attacks, Radiation Attacks	EM Attacks, Optical inspection (ROM, …)
Invasive	Forcing, Permanent circuit changes,…	Probing, …

Table 1.1: Overview over physical attacks [Man07]

physical properties of the device, instead passive attacks observe physical properties without manipulating such. The second criterion is the "invasiveness" of the attack. Non-invasive attacks leave the package unchanged and analyses have to be carried out, e.g., observing or interfering (glitching) external signals. Glitching attacks exceed the limits of the specification which (intendedly) cause misbehavior/faults of the chip. In general, the intended injection of a fault is called fault attack.
Semi-invasive attacks require to remove the chip package; however, do not alter the circuit itself. Examples for semi-invasive attacks are light attacks, which try to induce faults into electronic circuits by short light impulses, or localized EM side channel measurements, which observe the emanated magnetic field close to the die surface.
Invasive attacks manipulate the chip itself by removing the passivation layer or editing internal wiring, e.g., by a Focused Ion Beam (FIB) and lead to the most powerful attacks on chips. This enables active attacks where chip-internal buses can be changed (forcing) or values on these buses can be observed (probing) [Sko05, Man07].

In this thesis I apply non-invasive, passive power and semi-invasive, passive localized EM side channel analyses, which will be explained in the following.

1.3 Side Channel Attacks

Side channel attacks belong to passive attacks and observe a physical property of the device during an encryption, e.g., the consumed power or the emanated magnetic field, to reveal the used secret key. Side channel attacks do not attack the structure of the encryption algorithm, but its specific implementation. For Side channel attacks methods from cryptanalysis and knowledge from the physical attack are combined. Hence, both attack methods share similar threat models, compared to cryptanalysis: The attackers' access-capabilities and privileges increase from the ciphertext-only threat model to the related-key attack threat model.

Ciphertext-only: The attacker has access to the ciphertext only. For this scenario most published side channel attacks can be carried out [Koc99].

Known-Plaintext: The attacker has access to pairs of plain- and ciphertext.

Chosen Plain-/Ciphertext: The attacker can choose the plain- or ciphertext. This scenario is used (but not necessary) for most profiled attacks, which are explained below.

Adaptive Chosen Plain-/Ciphertext: The attacker can choose plain- or ciphertext and can adapt it after each side-channel observation. This can allow a faster key recovery by the attacker [Vey10], compared to the ciphertext-only scenario.

Related-Key Attack The attacker knows the relationship of keys, used for two (or multiple) encryptions with these keys. This scenario is rarely used for side channel attacks.

To extract the key from a cipher, three models, black-, grey- and white-box are defined, depending on the knowledge and capabilities of the attacker. In case of a black-box model plain-, ciphertext and algorithm are known to the attacker. Multiple encryption algorithms are assumed to be secure against such attacks, e.g., RSA, ECC and AES. A typical black-box attack scenario is, if an attacker can only use one (or more) communication interface(s), e.g., an internet accessible Internet Of Things (IOT) device. A white-box model contains

all knowledge from the black-box model plus all intermediate values (inside the black-box). Between the black- and white-box model exists the grey-box model, which includes all knowledge from the black-box model and partial information about intermediate values.

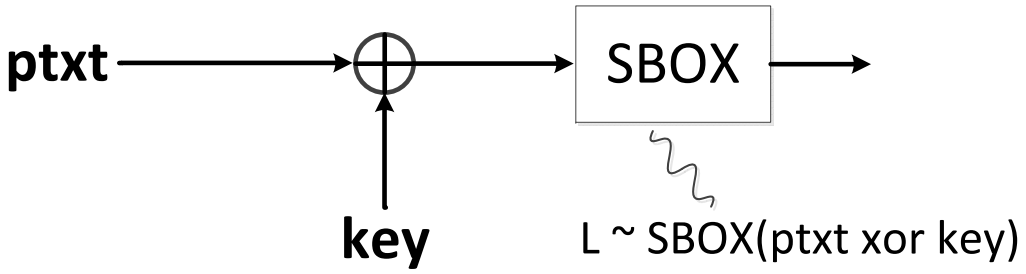


Figure 1.3: Examples for the combination of a public value with a secret value

Side channel attacks belong to the grey box model, because partial knowledge of the internal values is achieved, which depends in most cases on a combination of plaintext and key. In figure 1.3 I depict the combination of public (ptxt) and secret value (key) to the intermediate value "ptxt $\oplus$ key", which is processed by an S-box. [1] The measured leakage is depicted with L, which is the received side channel information of the intermediate value SBOX(ptxt $\oplus$ key). During a side channel attack, the goal of an attacker is to retrieve information about the intermediate value SBOX(ptxt $\oplus$ key) by measuring L. In most cases, a key-dependent value as early as possible of a cryptographic algorithm is attacked, because the dependency between parts of the input and key is strong, e.g., due to the bytewise combination of ptxt and key. ptxt and key become more and more uncorrelated with each round of the cipher, due to the above described properties confusion and diffusion. Side channel attacks can be split into two groups, Simple Power Analysis (SPA) and Differential Power Analysis (DPA).

1.3.1 Simple Power Analysis (SPA)

SPA uses just one (or a low number of) trace(s) to extract the secret. However, SPA commonly requires advanced knowledge about the exact implementation of the attacked algorithm and typically relies on optical inspection of traces,

[1] A similar combination is common for many published symmetric ciphers.

#	Targeted algorithm	Key recovery method	Reference
1	RSA	Optical Inspection	[Koc99]
2	RSA	Squared error	[Wal01]
3	RSA	Correlation	[Cla10]
4	ECC	Clustering	[Hey14]
5	ECC	Correlation	[Bau15]

Table 1.2: Overview of different SPA

e.g., [Man03]. Five examples for SPAs are listed in Table 1.2. Asymmetric cryptography is mostly the target of SPA due to a high computational complexity, long runtime and the processing of only few secret bits in parallel. The first attack in Table 1.2 uses key-dependent executed operations. During the attacked operation an exponentiation of large numbers has to be calculated. The chosen algorithm was the so-called "Square and Multiply" algorithm, which executes a square if the key-bit is equal to zero and a square and multiply if the key-bit is equal to one. These two cases can be (visually) distinguished by observing the consumed power of the device. This method is also called a horizontal attack, because it tries to identify similar operations at different points in time with one power trace. Another horizontal attack is Walters BigMac attack, which tries to identify similar operations to a preprocessing step. These operations are also key-dependent; thus the key can be recovered. Thereby the BigMac attack becomes more efficient with an increasing key size. Clavier et al. [Cla10] and Bauer et al. [Bau15] (numbers 3 and 5 in Table 1.2) carry out an attack called horizontal collision correlation analysis. Thereby they try to find similar operations by identifying one operation and correlate this to other possible occurrences of the same operations in time (horizontal attack). In case the same operation is executed (same key value), a high correlation appears, otherwise another operation is executed (different key value). Therefore it is possible to circumvent, e.g., the "square and multiply always" [2], which is not possible with the BigMac attack. Heyszl et al. [Hey14] published a similar attack, which uses location based leakage and unprofiled clustering to distinguish between two operations and thus recover the key.
In chapter 5 I show an improved version of this attack and present results for

[2]Square and multiply always is a classical square and multiply algorithm with a dummy multiply operation in case of a zero key-bit

#	Name	Targeted algorithm	Class	Reference
1	Original DPA	DES	unprofiled	[Koc99]
2	CPA	DES	unprofiled	[Bri04]
3	Original collision attack	DES	unprofiled	[Sch03]
4	Template attack	RC4	profiled	[Cha03]
5	Stochastic approach	AES	profiled	[Sch05]
6	Higher order attack	Block cipher	unprofiled	[Pee05]
7	Moments correlating DPA	unknown	(un)profiled	[Mor16b]

Table 1.3: Overview of different DPA

the combination of simultaneously captured leakage from multiple probes.

1.3.2 Differential Power Analysis (DPA)

DPA requires in contrast to SPA many (thousands or millions) traces to succeed. DPA uses statistical methods to (mostly) use data dependent leakage and recover the correct key. Therefore hypotheses tests for different keys, e.g., difference of means or correlation between a hypothetical power consumption and the measured power consumption are calculated. To recover the key, one hypothetical power value is calculated for every key-hypotheses and every trace. This results in a hypothetical power vector for every key-hypotheses. This vector is afterwards hypotheses tested against the measured traces. The hypotheses test should afterwards indicate the correct key. To calculate the hypothetical power value it is necessary to define a power model. This power model can be either profiled or unprofiled. Hence, DPA can be split into 2 groups, profiled and unprofiled. Profiled attacks require two phases, a profiling and an attack phase. During profiling phase it is possible to observe an identical device/training device, which is attacked during attack phase. For the observation during profiling the attacker knows or can choose the key. Therefore, the attacker can directly observe the power values of the targeted intermediate value; thus, profiled attacks are seen as the most powerful side channel attacks. One possibility is called the template attack, which creates a Gaussian template (a multidimensional Gaussian probability distribution) for every possible intermediate value [Cha03]. This attack is not restricted to Gaussian templates; however, these are the most common. A common reason is that Gaussian templates can be generalized to multidimensional Gaussian templates and noise is mostly

Gaussian distributed in side channel measurements [Man07]. Multidimensional templates allow, e.g., to attack multiple points in time or multiple side channels simultaneously, which is a so-called multivariate approach. Other approaches are called univariate and attack only one sample in time simultaneously, which can lower the exploitable leakage. Another profiled side channel attack is the stochastic approach [Sch05]. The attack consists out of a linear regression of the power consumption e.g., to a bitwise model of the intermediate value. Due to the linear regression, an attacker can also obtain the leakage of every single bit, which can give interesting inside to the hardware architecture.

In contrast to profiled attacks, unprofiled attacks do not need a training device to reveal the key. The original DPA [Koc99] carried out a statistical test to every single bit of the intermediate value to recover the key. A more powerful version was published, called Correlation-based differential Power Analysis (CPA) [Bri04], which does statistical tests on the Hamming weight of the intermediate value to recover the key. Another powerful class of unprofiled attacks are collision attacks [Sch03]. Collision attacks assume that hardware is reused, e.g., in case of AES, which operates on 16 byte in total but the S-box works bytewise, it is common to perform the S-box serially on each of the 16 bytes. Hence, the power consumption of 2 bytes, processed by the S-box is similar in case that the value of the 2 bytes is the same, which is called a collision. The moments correlating DPA [Mor16b] can be carried out as profiled and unprofiled attack. The unprofiled version works similar to a collision attack [Sch03] with the capability of attacking higher order statistical moments. Higher order attacks [Pee05] are necessary if specific countermeasures are implemented. The profiled moments correlating DPA, which I use is explained in chapter 3.1.3.
In chapter 4 I choose the profiled template attack [Cha03], because it is information theoretic the most powerful attack. The higher order profiled moments correlating DPA [Mor16b] is carried out to complement the analysis and to compare the results to the template attack. Both attacks will be explained in chapter 3.1.2 and 3.1.3.

1.4 EM Side Channel Attacks

To better understand EM side channel attacks, I briefly describe the behavior of the EM field first. In general, a change in electrical current causes an elec-

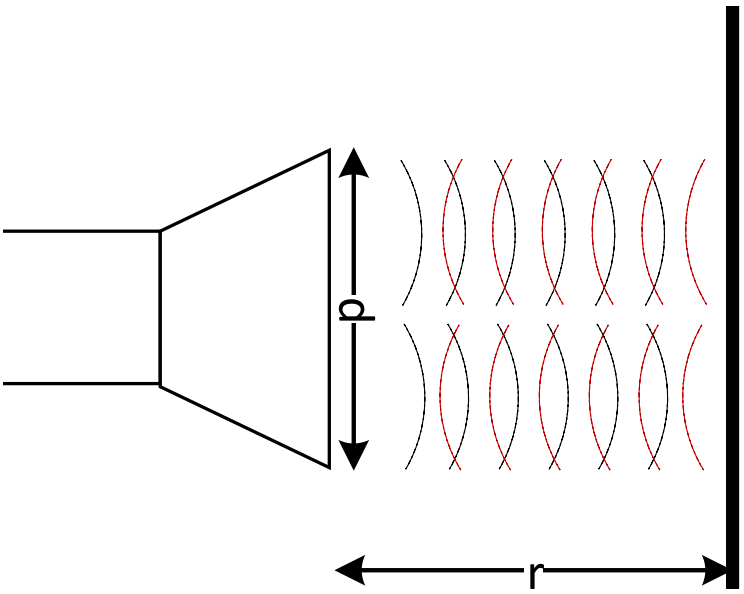


Figure 1.4: Reacive nearfield of an antenna

tromagnetic field, because the integrated circuit/cryptographic implementation acts as an antenna.
The literature distinguishes between reactive nearfield, radiating nearfield and farfield to describe antenna characteristics. The regions are not clearly separable; however, depending on targeted wavelength λ, the maximum antenna dimension d and the distance to the antenna r, there are "rule of thumbs" for distinct regions for nearfield, radiating nearfield and farfield. The wavelength is calculated by equation 1.1, where c is the speed of light in the medium and f is the transmitted frequency.

$$\lambda = {}^{c}/_{f} \tag{1.1}$$

Reactive nearfield "That portion of the near-field region immediately surrounding the antenna wherein the reactive field predominates." [Bal05] Absorbing (or reflecting) the field, e.g., by a receiver, does influence the load of the transmitter in this region [Rah95]. An example is depicted in figure 1.4, where the signal is reflected, influencing the load of the transmitter. "For a very short dipole, or equivalent radiator, the outer boundary is commonly taken to exist at a distance $\lambda/2\pi$ from the antenna surface." [Bal05]

Radiating nearfield "That region of the field of an antenna between the reactive near-field region and the far-field region [...]. If the antenna has a max-

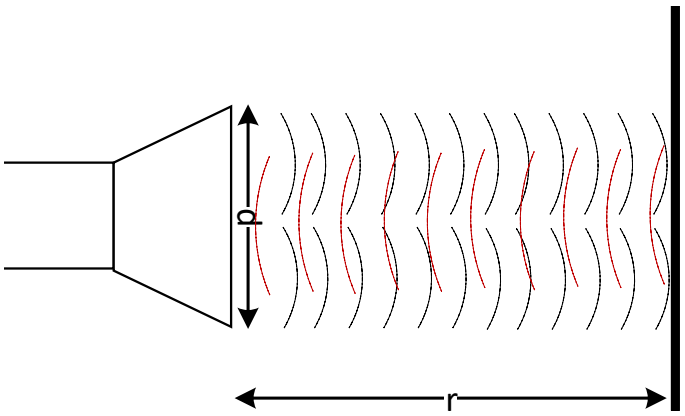


Figure 1.5: Radiating nearfield of an antenna

imum dimension that is not large compared to the wavelength, this region may not exist." [Bal05] In this region the absorption of the field does not necessarily influence the load of the transmitter. An example is depicted in figure 1.5, which shows a slight reflection. The influence to the load of the transmitter is not easily determinable. This region exists at a distance between $r \leq \sqrt{d^3/\lambda}$ and $r \geq 2d/\lambda$.

Farfield "That region of the field of an antenna where the angular field distribution is essentially independent of the distance from the antenna. If the antenna has a maximum overall dimension d, the far-field region is commonly taken to exist at distances greater than $2d^2/\lambda$ from the antenna, λ being the wavelength. " [Bal05]
An example is depicted in figure 1.6. In this region, the absorption of the field does not significantly influence the load of the transmitter.

In this thesis I measure frequencies up to 2.5 GHz, which results in a wavelength of $\sim 0.12\,\text{m}$ (see equation 1.1). Following the above defined "rule-of-thumb" I measure in the reactive nearfield region up to a distance of $0.019\,\text{m} = 1.9\,\text{cm}$. This is the case for all measurements in this thesis, where I measure at distances of a few mm or µm. This result is widely applicable to published side channel attacks, using the EM side channel.
In 2001 Gandolfi et al. [Gan01] presented practical results by successfully attacking a Data Encryption Standard (DES), comp128 and RSA by measuring the emanated EM field. This is possible, because an Integrated Circuit (IC) acts as an antenna due to a power consumption, e.g., caused by calculations of ci-

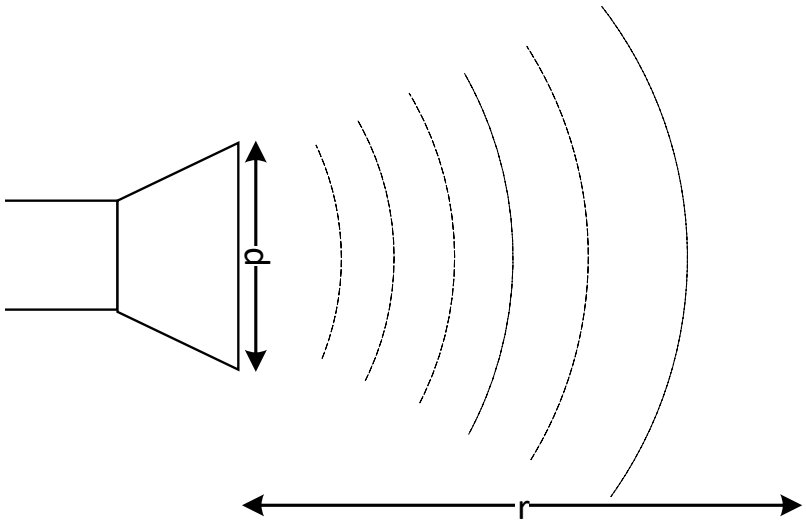


Figure 1.6: Radiating farfield of an antenna

phers. However, inside an IC, the single EM fields of each wire superpose and interfere, which results in a very complex radiation distribution. The first EM-based side channel attacks, mainly measured the EM emanation on the package of the chip. These are so-called off-chip measurements. Measurement results from off-Chip measurements indicate that vastly the leakage of the bonding wires (or a capacitor) is measured; hence the measured side channel leakage is comparable to the power side channel. [Spe14, Hey12a, Imm17] However, there are also more advanced measurements of the EM side channel, called on-chip measurements, which I will explain in the next section.

1.5 Localized EM Side Channel Attacks

During on-chip measurements a probe is directly placed on the surface of the die. This requires the depackaging of the chip. One method to carry out on-chip measurements is localized EM. Peeters et al. [Pee07] and Heyszl et al. [Hey12a] showed the power of localized EM side channel attacks. Localized EM measurements use magnetic near field probes with an inner diameter of a few 100 µm, placed directly above the die.
To explain the measurement principle of localized EM measurements I will describe the model of a modern IC. The model in figure 1.7 is simplified, but enables an abstract understanding of the measurement principle of localized EM and its application to ICs.

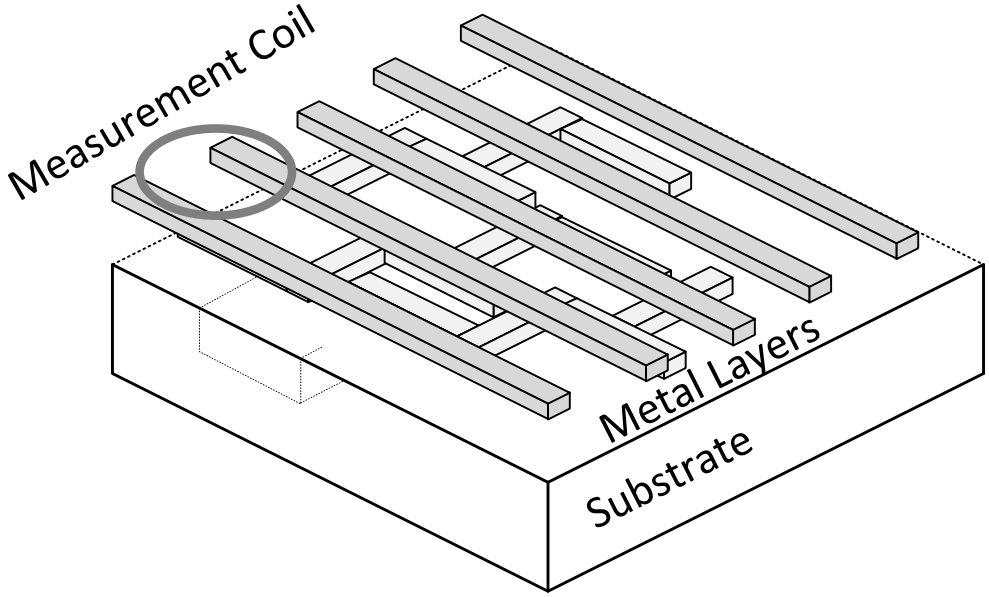


Figure 1.7: Typical structure of an integrated circuit [Hey13]

Classic production methods build ICs layerwise, with the silicon layer (substrate) in the bottom and multiple metal layers on top. The substrate implements the active region, which consists out of the transistors of the IC. The metal layers above the active area (lower layers) are commonly responsible for the transfer of signals between the transistors. To simplify production and development, it is common to minimize the wiring length between the transistors, especially at the lower layers. Therefore, it is likely that parts with a common purpose, e.g., crypto-cores or transceiver logic, are located in vicinity. Localized EM side channel measurements try to use this property by measuring with very small probes, e.g., an inner diameter of 150 µm directly above the die.

Localized EM probes are likely placed on top of the IC [Hey12b]. The top metal layers of the IC are commonly responsible for the power supply of the chip. Hence, the strongest measured signal is likely caused by the power supply wires, due to a close distance to the probe and only minor signal parts probably contain information of lower-layer signals.
The localized EM side channel has been shown to isolate logic parts, despite the measurement close to the power supply wires. This enables an attacker to neglect "non-relevant" logic parts and focus on relevant logic parts and massively decreasing the observation distance compared to off-chip measurements, which

results in a higher Signal to Noise Ratio (SNR) due to less noise and a higher signal level. Therefore, it is important to place the probe as close as possible to the signal source, because it is expected that the signal strength decreases with $1/r^2$, with r being the distance between probe and die [Spe14]. Additionally, the probe size is an important parameter of localized EM measurements. Due to the complex field behavior of an IC it is not determinable in advance, which probe size leads to the best SNR. Different probe sizes always lead to a trade-off between the observed IC area (bigger probe leads to a bigger observation area up to a global view on the chip) and the isolation properties of the probe (smaller probes lead to a better isolation property; however, have to be place more accurately). A probe size of 3 mm has shown to achieve similar results to power measurements in one publication [Imm17]. Additionally the localized view enables to exploit local non-uniformities, e.g., register placement [Hey12a] or routing imbalances [Imm17].

Furthermore, the small probe sizes lead to another advantage. It enables an attacker to place multiple probes above the die and to achieve more powerful attacks. However, it is unclear how and if multiple probes are able to improve the measurement result, in context to the above mentioned properties of one localized EM measurement probe.

1.6 Motivation

The previous sections have shown the power of side channel attacks and the importance to protect cryptographic implementations against such. To evaluate countermeasures and ensure an efficient protection, it is important to carry out strong attacks. Side channel attacks with the help of localized EM measurements belong to the most powerful EM based side channel attacks. Hence, this thesis has the following goals to increase the power of localized EM based attacks:

1. Attacks with multiple localized EM probes
2. Combining the simultaneous measurement results of multiple probes
3. Application of multiple probes to typical examples for symmetric and asymmetric ciphers

The two chosen experiments are especially interesting due to the different (mathematical) structure of the underlying cryptographic algorithms. The implemented countermeasures in both experiments and thus the side channel attacks, significantly differ. Commonly, symmetric ciphers operate multiple (thousands or millions) times with a constant key. This allows the collection of many traces for side channel analysis. In contrast, asymmetric ciphers can randomize the key dependent operation (at an algorithmic or protocol level) such that only one trace is usable for an attack. Hence, I use multiple localized EM probes in a DPA scenario for the symmetric cipher and an SPA scenario for the asymmetric cipher.

1.7 Contribution

In chapter 4 and 5 I carry out the above mentioned experiments. These experiments lead to the following contributions in this thesis:

Multiple Probes can increase the attack efficiency for profiled attacks In chapter 4 and 5 I show that attacks with multiple probes lead to a more powerful attack during a profiled attack scenario.

Investigating the combination of Multiple Probes by LDA and PCA Combining the signal of multiple probes can lead to significantly increased leakage. I show in detail the combination of multiple probes during an attack in chapter 4.7.3.

Improving State-of-the-Art Attacks for localized EM In chapter 5 I show that it is possible to massively improve an unprofiled single trace attack, which is based on the attack of Heyszl et al. [Hey14] by using a dimension reduction technique known from machine learning. In chapter 4 I extend a state-of-the-art template attack.

Multiple probes can significantly weaken a countermeasure for symmetric ciphers I show that multiprobe localized EM measurements can weaken a central security assumption for a symmetric DPA countermeasure. The implemented countermeasure splits the secret into multiple independent parts. It

can be shown that each of the placed probes can focus on one or more of these parts, which is assumed to be not possible.

1.8 Notation

In this thesis matrices are indicated by bold, capital letters, e.g., $\mathbf{T}$, vectors by bold, lower case letters, e.g., $\mathbf{t}$ and scalars by lower case letters, e.g., t and probability distributions by capital letters, e.g., P. For a specific side channel experiment the set $\mathbf{T}$ of traces is collected with n being the number of collected traces. $\mathbf{T}^v$ specifies the set of traces for the internal value v. One trace $\mathbf{t}$ of length γ is represented by its samples $\mathbf{t} = (t_0, ..., t_{\gamma-1})$ which have been acquired over time.

1.9 Outline

This thesis is organized as follows. In chapter 2 I briefly describe the current state-of-the-art for the countermeasures and attacks, shown in chapter 4 and 5. Furthermore, I introduce state-of-the-art for combining multiple channels and probes. I describe selected theory to model measurements with multiple probes and the requirements for an optimal combination approach. In chapter 3 I explain the dimension reduction for combining multiple probes and analysis tools, which are used in chapter 4 and 5. Afterwards, I present results to the experiment for attacking a symmetric cipher with multiprobe side channel attacks in chapter 4. In chapter 5 I present results to the improvement of single trace attacks on asymmetric ciphers by single probe and multiprobe side channel attacks. In chapter 6 I draw conclusions.

Chapter 2

State-Of-The-Art

In this thesis I carry out two examples for a multi-probe localized EM attack. Hence, I split the state-of-the-art in three parts. Firstly, I describe the state-of-the-art for countermeasures and related attacks carried out in chapter 4 and 5. Afterwards, I describe the state-of-the-art for combining multiple probes or channels for side channel attacks. Then, I describe an information theoretic approach to model side channel analysis. This model shows that a combination of multiple probes corresponds to a dimensionality reduction. Afterward I explain the basic properties of a dimensionality reduction technique for side channel analysis and motivate the chosen machine learning techniques for dimensionality reduction.

Usually a combination of different countermeasures is implemented in most (hardened) cryptographic implementations to increase the resistance against side channel attacks. One class of countermeasures (hiding countermeasures) concentrate on the independency of power consumption and processed secrets, which is mostly implemented with dedicated logic styles [Her06, Man07] or the randomization of execution orders (shuffling). One other class of countermeasures randomizes the intermediate values. In general, this randomization belongs to the most powerful countermeasures and is called masking for symmetric ciphers. The goal is to randomize the intermediate value, such that it is independent from input data and key [Nik11, Osw05]. In contrast, asymmetric ciphers can implement a countermeasure called blinding. Thereby the key dependent operation can be randomized (at an algorithmic or protocol level), such that only the leakage of one trace is usable for an attack.

I will analyze with the help of multiprobe localized EM one masking scheme in chapter 4 and one implementation based on the randomization at the protocol level in chapter 5.

2.1 Related Work of Masking Schemes and Attacks

I describe the related work in two parts, firstly masking techniques as a side channel countermeasure and secondly, published attacks on such protected implementations. In chapter 4 I carry out an attack on a Threshold Implementation (TI), which is a special masking technique and is explained in section 4.1.

2.1.1 Masking as a Side Channel Countermeasure and Threshold Implementations

The idea of masking was introduced by Messerges et al. [Mes00a], which is known as first order masking scheme (it uses one mask for the internal value). This masking scheme is a Boolean masking scheme, which performs a XOR-operation between mask and intermediate value. However, dependent on the specific cryptographic algorithm or operation, other masking schemes can be advantageous, e.g., arithmetic, multiplicative and polynomial masking. Numerous publications showed in theory [Duc15] and practice [De 17] the vulnerability of imperfectly implemented masking schemes, e.g., in case of dependent leakage of mask and masked value, non-uniform distribution of masks [Man07] or glitches [Man05b, Man05a]. Nikova et al. [Nik06] suggested to secure a cryptographic algorithm by a Boolean masking scheme, the Threshold Implementation (TI) and theoretically proved that they show no leakage even in presence of glitches. The principle of threshold implementations is independent of a specific algorithm, but the structure itself has to be adapted strongly (similar to masking schemes) to specific operations, e.g., multiplications, or inversions in GF_{2^8} [Mor16a, De 16a, De 16b, Nik08].
However, masking schemes can be attacked with higher order DPAs [Pee05], which are explained in section 2.1.2. To further strengthen masking schemes, d^{th} order masking schemes (and also higher order TIs) were developed [Gro17b, Gro17a]. On the other side, d^{th} order secure masking schemes are costly in terms of size and timing [Riv09, Riv10, Cor07].

However, masking does not perfectly protect against DPA-approaches (as explained in the next section) and should be combined with other, e.g., hiding-based, countermeasures to further strengthen the security of an implemen-

tation. Hiding countermeasures try to solve the problem by avoiding data-dependencies in the power consumption, which can be realized on the algorithmic- or cell-level. The algorithmic approach mostly performs hiding in the time domain, e.g., dummy en- or decryptions, dummy rounds or randomizing the order of operations. Countermeasures at cell-level implement specific logic styles, which ideally remove the data-dependent power consumption and thereby equalize it. When considering the various proposals in this domain [Nas10, Lom09, Sau09, Bha10, Mor14, Yu,07, Kap10, He 11, He 12, Wil18], one identifies that most of them are based on Dual-Rail Precharge (DRP) logic or duplication schemes (DUP). Both no longer represent a bit as a single value but instead as complementary rails, such that regardless of the operation, each bit-flip is compensated by an inverse bit-flip.

2.1.2 Profiled Attacks Against Masked Symmetric Cipher Implementations

Many successful attacks on masking schemes are based on implementation flaws, e.g., the non-uniformity of masks or the biasing of mask values by fault attacks or an appropriate preprocessing of traces [Man07, Riv10]. Properly implementing a masked implementation is a challenging task. However, even when assuming a perfect implementation, implementations can be broken by higher order attacks [Mes00b, Pro09, Cha99] or the capability of an attacker to profile mask values. In general, a d^{th} order secure masking scheme (including threshold implementations), can be attacked by an at least $d^{th} + 1$ order DPA [Mes00b, Pro09, Cha99]. Higher order attacks need to observe $d + 1$ intermediate values to succeed. Afterwards, the intermediate value can be revealed, e.g., by estimating univariate higher order statistical moments [Man11] or a multivariate approach to combine data dependencies between multiple points in time [Osw07, Osw06]. [1]

Another approach to attack $d + 1$ intermediate values is to create profiles for each mask value; thereby, it is possible to recognize the currently used mask value and remove the mask from the intermediate value during the attack. Template attacks [Bar10, Cha03, Rec05, Man05b] are one example that appropriate template training can be used to recognize the current mask value and remove it. Lerman et al. [Ler15] extended the recognition of mask values by using

[1] The chosen attack heavily depends on the implementation

machine learning techniques and further improved the efficiency of such an attack. In chapter 4 I carry out a template attack on a second order secure implementation by recognizing and removing both mask values.

2.2 Related Work for Attacks on Asymmetric Algorithms

In chapter 5 I focus on a non-profiled attack; hence, I cover the related work for unprofiled attacks first. To evaluate the results of the unprofiled attack I compare these to a profiled approach. Thus, I briefly describe the related work for profiled attacks afterwards.

The main computation in public key cryptosystems is modular integer exponentiation with secret exponents (e.g. RSA, DSA) or elliptic curve scalar multiplication (e.g. ECDSA) with secret scalars. Due to a common structure of exponentiation and multiplication algorithms, I will use the generalized terms 'exponentiation algorithms' and 'secret exponents'.
The secret exponent is usually either ephemeral by the protocol design (e.g. ECDSA) or blinded through countermeasures (e.g. exponent blinding in RSA to prevent profiling). Therefore, it is different for every execution; and side-channel attacks may only exploit *single executions*. However, the calculation of asymmetric ciphers is complex, which leads to long calculation periods, compared to symmetric ciphers. These long calculations enable horizontal attacks, which operate on one (or a few) observations. E.g. the first *single-execution* attack on exponentiations was presented by Kocher [Koc99] who exploits key-dependent operation sequences. To avoid this, improved algorithms like the square-and-multiply-always, double-and-add-always or the Montgomery ladder were introduced, which have constant operation sequences (e.g. side-channel atomic routines) to avoid *simple side-channel attacks*. In all those algorithms, exponents are scanned bit- or digit-wise (depending on whether it is a binary, m-ary, or sliding window exponentiation) and the computation is performed in a loop iterating a constant sequence of operations. (I will continue to refer to the binary case.) Nonetheless, some side-channel leakage about the processed exponent remains in many cases which can be referred to as single-execution leakage.

2.2.1 Simple Power Analysis Non-Profiled Attacks Against Exponentiations

In the following I give a brief overview of existing non-profiled attack techniques. Two factors impact the practical threat of an attack, the required knowledge of the attacked implementation and the capability to create profiles. [2] The more knowledge an attacker requires, the more difficult is the attack to pursue. Clavier et al. [Cla10] use cross-correlation in non-profiled single-execution attacks on exponentiations and require full knowledge of all used algorithms, which makes the attack more challenging than the next one.

Walters BigMac attack requires less knowledge (only the used exponentiation algorithm and the method of the single exponentiation). It is a non-profiled approach, which uses data-dependent leakage from using pre-computed multiples in digit-wise multiplications [Wal01]. One example, which requires only to know the exponentiation algorithm was published by Goubin [Gou02], on an ECC implementation. This approach uses data-dependent leakage and chooses the input such that a specific intermediate value only occurs, e.g., in case the currently attacked target bit is 0.

In contrast, Perin et al. [Per14] described a two-stage approach and use address based leakage and require to know the exponentiation algorithm only. In the first step, Perin et al. extract the relevant points in time with a clustering algorithm. I pursue an approach, for *non-profiled* attacks based on another technique (Principal Component Analysis (PCA)) to extract relevant points in time, which only requires knowledge of the exponentiation algorithm. I explain PCA in chapter 3.2.2 and the attack in chapter 5. In 2016, Järvinen et al.[Jar16] show that a correlation based horizontal attack can perform better than clustering based approaches, especially in case of the computation of multiple key-bits in parallel; however this correlation based approach requires additional knowledge of the specific implementation, which is attacked. Thus increasing the overall effort for the attack.

[2] This is also true for symmetric algorithms; however, especially for attacks on asymmetric ciphers the details of the exact implementation allow more sophisticated attacks.

2.2.2 Simple Power Analysis Profiled Attacks Against Exponentiations

A more efficient exploitation of leakage, compared to unprofiled attacks, is possible if profiled attacks are carried out. Different types of profiled attacks are published. In most cases either a key-dependent operation or a key-dependent address can be exploited to recover the key. E.g. Bauer et al. [Bau12] showed that square and multiply operations can be distinguished with a single trace. Itoh et al. [Ito03] published an example for the leakage of key-dependent address-bits, which can be exploited to recover the key. A similar approach to the address-bits recovery is the location-dependent leakage from accessing different storage locations [Hey12a]. To estimate the performance of the suggested unprofiled attack in chapter 5 I compare the results to a profiled approach. Due to the similar unprofiled attack principle, a similar profiled approach of Heyszl et al. [Hey12a] is taken for comparison.

2.3 State of the Art in Multi-Probe Side Channel Measurements

The exploitable information leakage in side-channel measurements is generally limited. Using multiple side-channels concurrently, and combining them in an attack is an important way of increasing the exploitable leakage in side channel attacks. Agrawal et al. [Agr03] described the combination of current consumption and magnetic field measurements in a profiled template attack and an unprofiled DPA with a bit-hypothesis through concatenation of traces. The unprofiled DPA can only exploit leakage from multiple channels in case of a very similar leakage characteristic, because the described attack assumes that leakage occurs in the same point in time for both channels. [Agr03] Standaert and Archambeau [Sta08] report better results from magnetic field than current measurements and show an improvement from the combination of both channels. They compare a profiled PCA-based template attack and an LDA based template attack. The conditional entropy of both shows that LDA can be an alternative to PCA.
Afterwards Souissi et al. [Sou12] and Elaabid et al. [Ela11] presented results from combining two simultaneous measurements of the magnetic field. Souissi et al. measure the field close to two different supply capacitors of an FPGA.

They measure the supply of two different parts of the FPGA and extend the CPA to combine simultaneous measurements, using products [Ela11] or sums [Sou12] of correlation coefficients. However, the proposed methods require either that different channels leak information at the same cycles [Sou12] or that the leaking cycles for combination can be recovered in a profiling step [Ela11]. Heyszl et al. [Hey14] mention the combination of multiple high-resolution probes for non-profiled single-execution attacks with clustering algorithms. This approach enables an attacker to use leakage at different points in time with an unprofiled approach. However, Heyszl et al. [Hey14] did not perform actual simultaneous measurements. I extend the work and present results from an extensive practical study using three high-resolution micro-coil magnetic field probes to attack a secured implementation of the AES S-box and mount a non-profiled single-execution side channel attack against an asymmetric cipher.

2.4 SISO and MIMO Communication Channels

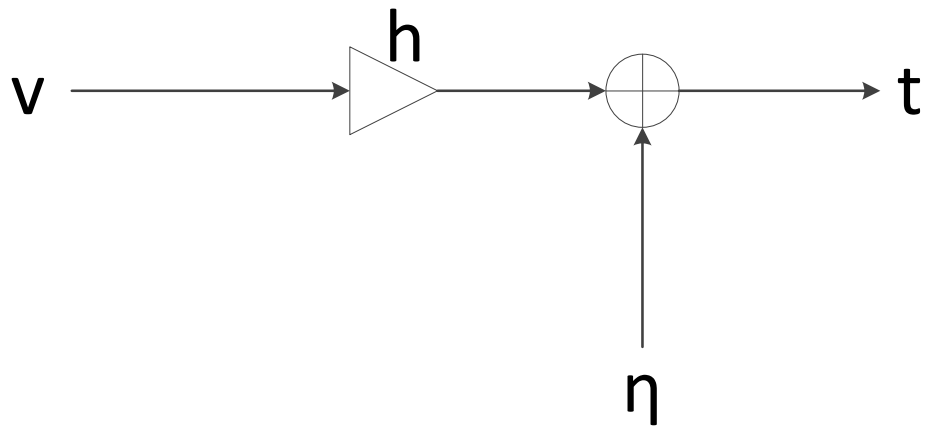


Figure 2.1: SISO Channel with additive noise

Communication channels are intensively researched in the area of wireless communication. With a communication channel it is possible to model the information transfer between a source and a sink. This is relevant for this thesis, because the side channel value L of Figure 1.3 can be modeled as a communication channel between die (source) and probe (sink). Furthermore, the use of multiple antennas for communication is similar to the use of multiple probes

for side channel analysis from an information theoretic point of view. Hence, I briefly describe the basics of communication channels and the application to multiprobe side channel setups.

The most popular and widely applicable channel model in communication theory is the additive white Gaussian noise channel, depicted in figure 2.1. Figure 2.1 shows the transmission of the data signal $v \in \mathbb{C}$ from the source, which is weighted with the channel coefficient $h \in \mathbb{C}$. h models the loss of the channel itself and is assumed to be deterministic, but unknown. The loss is commonly a signal attenuation, caused by objects in the transmission path or simply the transmission distance between sink and source. Afterwards the noise $\eta \sim N(0, \sigma)$ is added, which results in the received signal $t \in \mathbb{C}$ (sink). The noise η models all received signal parts of t, which do not contain information of v. In this model the origin of the noise, e.g., source amplifier, antenna, sink amplifier or communication channel, is not respected. We assume that the noise η is Gaussian distributed with a zero mean and standard deviation σ. The case in figure 2.1 is called the Single Input Single Output (SISO) channel, which is a typical use case for one sending antenna and one receiving antenna.

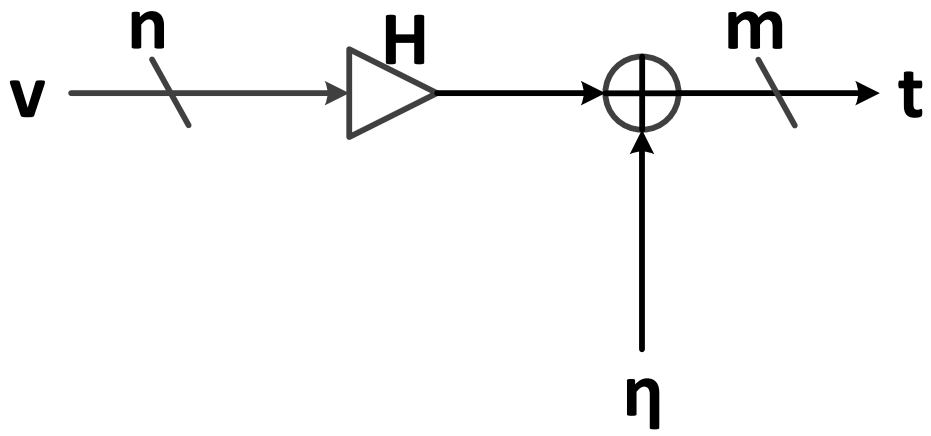


Figure 2.2: MIMO Channel with additive noise

The Multiple Input Multiple Output (MIMO) channel is depicted in Figure 2.2. This models e.g., a communication channel with multiple sending and multiple receiving antennas. Therefore, the data signal of the channel is $\mathbf{v} \in \mathbb{C}^n$, which is multiplied with the channel coefficient matrix $\mathbf{H} \in \mathbb{C}^{nxm}$. Each element of **H** describes the dependency between each element in **v** to each element in **t**.

The noise coefficient consists out of an m-element vector $\boldsymbol{\eta} \sim N(0, \sigma)$ and the output corresponds to $\mathbf{t} \in \mathbb{C}^m$.

2.5 Application of the MIMO Channel Theory to Side Channels

Unterluggauer et al. [Unt17] showed the application of Multiple Input Multiple Output (MIMO) communication channel theory to side channel analysis. In the following we use the Gaussian noise assumption, which is common for side channel attacks and supported by practical measurements [Man07]. The side channel is modeled as an n to m communication channel, where **v** models the secret state bitwise and **t** models the acquired side channel information, which can be calculated by equation 2.1. Thereby, **t** can model multiple sampling points in time or multiple probes. By estimating **H** it is possible to analyze the influence of each element of **v**, to each element of **t**, e.g., each bit to each sampling point.

$$\mathbf{t} = \mathbf{H}\mathbf{v} + \boldsymbol{\eta} \tag{2.1}$$

The theoretical bound for leakage/information of **v** measurable by **t**, can be modeled by the channel capacity c, defined by the maximum average Mutual Information (MI) between the probability distribution V of **v** and the probability distribution T of **t** in equation 2.2, where p(v) denotes the marginal distribution.

$$c = \max_{p(v)} MI(V, T) \tag{2.2}$$

Assuming Gaussian noise for $\boldsymbol{\eta}$ the channel capacity can be estimated by equation 2.3. $\boldsymbol{\Sigma}_{\boldsymbol{\eta}} \in \mathbb{C}^{mxm}$ denotes the covariance matrix of the noise and $\boldsymbol{\Sigma}_{\mathbf{y}} \in \mathbb{C}^{mxm}$ the covariance matrix of $\mathbf{y} = \mathbf{H}\mathbf{v}$.

$$c = \frac{1}{2}\log_2(\det(\mathbf{I}_m + \boldsymbol{\Sigma}_{\boldsymbol{\eta}}{}^{-1}\boldsymbol{\Sigma}_{\mathbf{y}})) \tag{2.3}$$

Unterluggauer et al.[Unt17] showed that $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}{}^{-1}\boldsymbol{\Sigma}_{\mathbf{y}}$ corresponds to the multidimensional SNR including all correlations between all side channel signals **y**.

3

The vector **t** can model multiple samples in time and/or samples from multiple probes. An exemplary matrix **H** for three probes can be modeled as seen in equation 2.4, where $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3$ denotes the **H** of the first to the third probe and **0** denotes a zero matrix with the same dimensions as one of the $\mathbf{H}_1$ [Unt17]. Please note that the zero matrix in equation 2.4 assumes the independency of all three probes, which might not be necessarily true.

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_3 \end{bmatrix} \tag{2.4}$$

Hence, $\mathbf{H}_1$ models the influence of **v** to the according dimensions of **t** for the first probe, $\mathbf{H}_2$ for the second probe, and $\mathbf{H}_3$ for the third probe. In this thesis I use multiple samples in time and multiple probes in chapter 4 and 5 for side channel analysis. I denote the total trace length of all probes with γ and the number of traces with n, which results in the trace set $\mathbf{T}^{n x \gamma}$.

Combining the Measurement Data of Multiple Probes Combining the measurement data of multiple probes is of central importance to maximize the exploitable information of all probes. The information theoretic approach of equ. 2.1 shows that the combination of the measurement data of multiple probes can be interpreted as the reduction in number of dimensions of the vector **t**.
Dimensionality reduction is a well known problem in machine learning. Thus, I will focus on machine learning techniques for dimensionality reduction in this thesis. In the following I will describe the requirements for an optimal dimension reduction strategy in side channel analysis.

2.6 Optimal Dimensionality Reduction in Side Channel Analysis

For side channel analysis, commonly traces with hundreds or thousands of points in time are measured. Each trace can be seen as element in a high dimensional space. Hence, side channel attacks can be seen as separating the elements

[3]The derived channel capacity is similar to the representation for MIMO communication channels

according to their v in this high dimensional space. However, analysis in high dimensional spaces can cause several problems, including numerical instabilities, long runtimes or large memory consumption. Only very few dimensions (e.g., ~ 0.2 % in chapter 5) are relevant for a successful attack. Thus, the number of dimensions can be reduced in most cases and problems caused by high dimensionality analysis can be avoided, while retaining the relevant information. [4]

The field of machine learning suffers from very similar problems, e.g., image recognition [BBHK97]. A lot of tools to reduce the number of dimensions and to retain the relevant information were published. The process to reduce the number of dimensions is split into two steps, feature extraction and dimension reduction. During feature extraction the dimensions of the original space are weighted. During dimension reduction, the dimensions are combined (according to their weights) and transformed to a space with reduced dimensions. Inspired by machine learning, side channel analysis uses similar or the same dimension reduction techniques. In 2015, Nicolas et al. [Nic15] showed that an optimal dimension reduction strategy fulfills the following properties in side channel analysis:

1. "The optimal attack on the multivariate traces [...] is equivalent to the optimal attack on the monovariate traces" [Nic15]

2. "The optimal dimensionality reduction is made by a linear combination of the samples" [Nic15]

3. "After optimal dimensionality reduction, the signal-noise-ratio is given by $\alpha^T \mathbf{\Sigma}_{\eta}{}^{-1} \alpha$, where $\alpha \in \mathbb{R}^{1xm}$ corresponds to the weight for each dimension" [Nic15] [5]

According to Nicolas et al. [Nic15] LDA can hold these properties. Hence, we use LDA as preprocessing technique in chapter 4.

However, LDA belongs to a profiled preprocessing technique. In chapter 5 I mount an unprofiled attack; thus LDA is not suited. An unsupervised dimension reduction technique, which is wide-spread and well known for dimension reduction in machine learning is PCA. Furthermore, PCA has been applied to

[4] A reduction in number of dimensions can not increase the contained information [Nic15, Cha03]

[5] In contrast to equation 2.3, $\mathbf{t}$ is assumed to have normalized variance; Hence $\mathbf{\Sigma}_{\mathbf{y}}$ is equal to the identity matrix

side-channel analysis for data reduction in several contributions [Boh03, Arc06, Sta08, Bat12, Mav12] and for different attacks, where Archambeau et al. [Arc06] were the first to describe the use of PCA in the context of template attacks. Additionally, Chang et al. stated that PCA is a highly promising candidate for a statistical problem, similar to our application in chapter 5. Thus, PCA is used in chapter 5.

Beside LDA and PCA, other dimension reduction techniques are common in side channel analysis. Most are usually justified by electrical properties for side channel analysis and use simple functions for combinations, e.g., computing the sum, sum-of-squares, multiplying the sample values, or extracting peak values for given dimensions/samples. [Man07]
Additionally, more complex non-linear trace compression techniques have been published, e.g., sine-based functions [Osw07]. However, these non-linear functions require significant effort to adapt to a specific use-case. Furthermore, testing and comparing all published preprocessing techniques would exceed the focus of this thesis.
Especially in terms of the usability, LDA and PCA are superior to other simple dimension reduction techniques, because relevant information is tried to be extracted automatically during computation out of determined dimensions and only very limited "fine-tuning" is required in contrast to simple preprocessing techniques, which require a lot more experimenting with their parameters. Hence, PCA and LDA are the preferred dimension reduction techniques in this thesis.
In this thesis, I carry out attacks with one simple trace compression technique, the summation of all single probe signals and compare the results to LDA in chapter 4 and to PCA in chapter 5.

Chapter 3

Applied Evaluation Methods for Side Channel Analysis

In the last chapter I covered the state-of-the-art for selected countermeasures and attacks. Furthermore, I explained an information theoretic modeling approach for side channel analysis, which shows combining the measurement data from multiple probes can be performed with dimensionality reduction techniques. The criteria for an optimal dimensionality reduction (listed in chapter 2.6) can be fulfilled by LDA and PCA. Hence, I explain these two machine learning methods in this chapter. For the application of machine learning methods it is important to reduce the number of points in time beforehand. Points in time with key dependent behavior of the measurements are so-called Points of Interest (POI). The chosen Points of Interest (POI)s are denoted as $\mathbf{q}$ and are selected from the points in time of the trace $z \in [0, ..., \gamma - 1]$. Furthermore, localized EM probes can be positioned at multiple places above the die. A position with key dependent information is called Location of Interest (LOI). In this chapter, I will firstly explain the tools to determine POI and LOI in chapter 4. In this chapter, I will firstly explain the used evaluation methods for chapter 4 and the tools to determine POI and LOI in chapter 3.1. Afterwards, I will introduce the attack principle and evaluation methods for chapter 5 in chapter 3.2.

3.1 Evaluation Methods used in Chapter 4

In the following I explain the used dimensionality reduction and analysis tools of chapter 4.

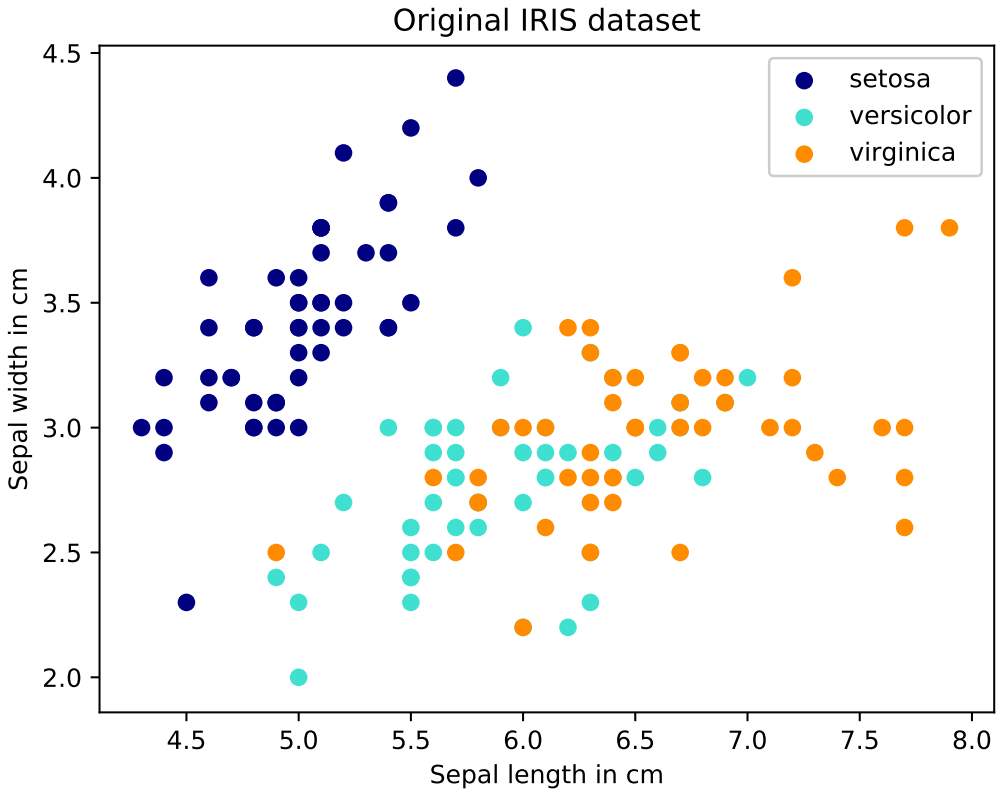


Figure 3.1: First two dimensions of the IRIS dataset

3.1.1 Linear Discriminant Analysis (LDA) for Dimensionality Reduction and Feature Selection

The goal is to construct a sub-space that maintains relevant information in less dimensions. In practice this goal is achieved if classes of elements are separable in the sub-space. Hence, LDA requires a supervised profiling phase to calculate a transformation matrix which is then multiplied with individual traces to transform them into a sub-space with lower dimensionality. Hereby, every trace is treated as an element with dimensionality γ. During the profiling phase, LDA computes a linear transformation, which minimizes the variance of the elements within the same class (within class scatter) and maximizes the variance of elements between differing classes (between-class scatter). More precisely, LDA maximizes the ratio of between-class to within-class scatter.

I explain the steps of LDA with the help of the IRIS dataset [Fis36]. The IRIS

dataset contains 3 kinds of flowers (Setosa, Versicolour and Virginica), which results in the class set $\mathbf{V} = [0, 1, 2]$, described with 4 attributes (4 dimensional space, $\gamma = 4$). The four measured attributes of the flowers are the following:

1. Sepal length
2. Sepal width
3. Petal length
4. Petal width

I depict the first over the second dimension of this dataset in figure 3.1. In figure 3.1 especially the classes Versicolour (light blue) and Virginica (orange) are hard to separate.

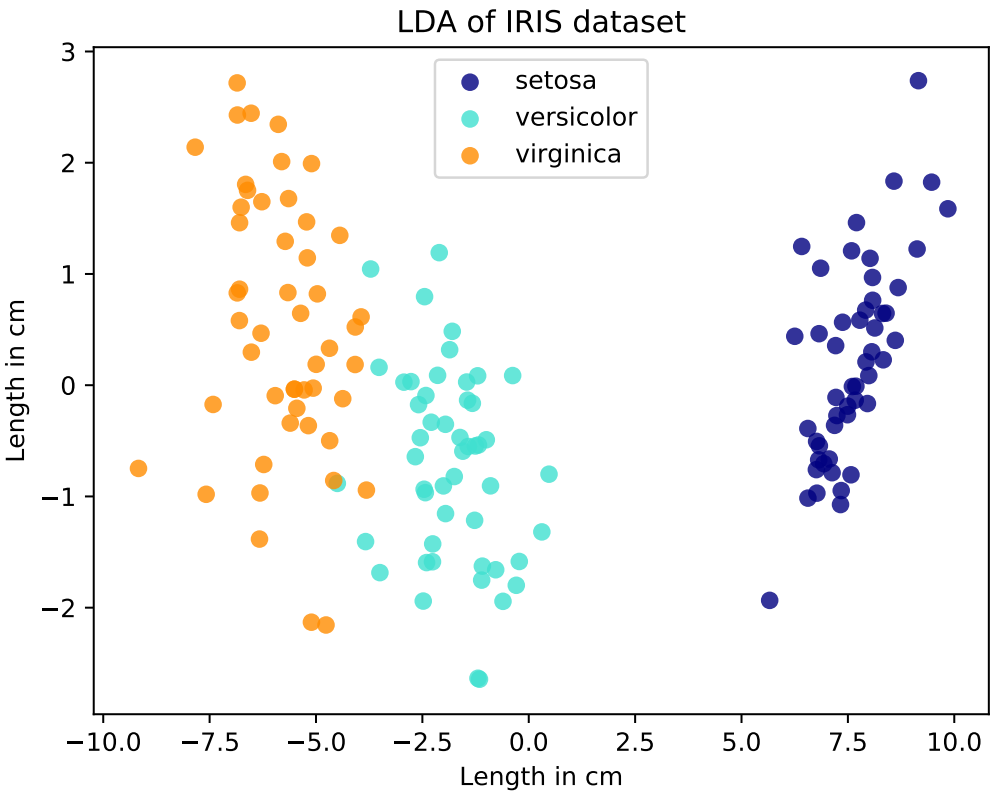


Figure 3.2: IRIS dataset transformed with LDA

LDA tries to find a subspace, where the classes are better separable. LDA works as follows:

1. The elements are grouped in the sets $\mathbf{T}^v \in \mathbb{R}^{n_v x \gamma}$ according to their groups with n_v elements of class v. E.g. $\mathbf{T}^0$ contains all blue elements of figure 3.1.

2. The mean of each class set $\mathbf{T}^v$ (e.g., class 0, blue) is estimated by equation 3.1:

$$\boldsymbol{\mu}_v = \frac{1}{n_v} \sum_{n=0}^{n_v-1} \mathbf{t}_n^v \tag{3.1}$$

3. The overall mean is estimated by equation 3.2: [1]

$$\boldsymbol{\mu} = \frac{1}{|\mathbf{V}|} \sum_{v=0}^{|\mathbf{V}|-1} \boldsymbol{\mu}_v \tag{3.2}$$

4. LDA calculates the within-class scatter matrix $\mathbf{S_w}$ according to equation 3.3, which is the covariance matrix of all elements within the same class, e.g., the elements in $\mathbf{T}^0$.

$$\mathbf{S_w} = \sum_{v=0}^{|\mathbf{V}|} \sum_{n=0}^{n_v-1} (\mathbf{t}_n^v - \boldsymbol{\mu}_v)(\mathbf{t}_n^v - \boldsymbol{\mu}_v)^T \tag{3.3}$$

5. LDA calculates the between class scatter matrix $\mathbf{S_b}$ according to equation 3.4, which is the covariance matrix between each class. E.g. the covariance matrix between $\mathbf{T}^0$, $\mathbf{T}^1$ and $\mathbf{T}^2$.

$$\mathbf{S_b} = \sum_{v=0}^{|\mathbf{V}|-1} n_v (\boldsymbol{\mu}_v - \boldsymbol{\mu})(\boldsymbol{\mu}_v - \boldsymbol{\mu})^T \tag{3.4}$$

6. The ratio $J(\mathbf{W})$ is maximized according to equation 3.5 and 3.6, with $\mathbf{W} \in \mathbb{R}^{\gamma x \beta}$ and β the number of output dimensions:

$$J(\mathbf{W}) = \frac{\mathbf{W}^T \mathbf{S_b} \mathbf{W}}{\mathbf{W}^T \mathbf{S_w} \mathbf{W}} \tag{3.5}$$

[1] Assuming that v is approximately uniformly distributed

Output dimension	**Values**			
1	0.21	0.40	-0.56	-0.70
2	-0.2	-0.60	0.64	0.44

Table 3.1: Matrix **W** (resulting weights) for the first two LDA dimensions of the IRIS dataset. Please note that each output dimensions corresponds to one matrix column.

$$\mathbf{S_w^{-1}S_bW} = \lambda\mathbf{W} \tag{3.6}$$

J(**W**) in equation 3.5 can be transformed into a constrained optimization problem, which results in a (generalized) eigenvalue problem, shown in equation 3.6. Hence, the transformation matrix **W** consists out of eigenvectors, which are normalized. In the example of the IRIS dataset the matrix **W** contains entries described in table 3.1 for two dimensions. Multiplying these weights **W** to the IRIS dataset **TW** results in figure 3.2, with the trace set $\mathbf{T} \in$ nxγ.
Comparing figure 3.1 and 3.2 shows that the classes are better separable. Already along the first dimension after LDA, the classes are well separable.

3.1.2 Template Attack

Template attacks belong to profiled attacks and represent an information theoretic optimal attack [Cha03]. Template attacks estimate a Probability Density Function (PDF) P(**t**|v) for each v. A probability for specific measurement values $\mathbf{t} \in \mathbb{C}^{1x|\mathbf{q}|}$ given an intermediate value v can be determined. **q** denotes the chosen POIs, which are selected points in time between 0 and $\gamma - 1$. Please note that **q** can also be modified by the preprocessing and denotes the input length for the template attack. [Osw07]
It can be shown that by estimating P(**t**|v), P(v|**t**) can be derived. Thus, it is possible to calculate the probability for each v given the measurement value **t**. When assuming that $v = p \oplus k$, combined with a known p, I can calculate the probability for each key-candidate for each **t**. Due to a high noise and low SNR for side channel measurements, the attack usually requires multiple measurements to determine the correct key.
The chosen probability density distribution heavily relies on the Device under Test (DUT). Commonly chosen PDFs for side channel analyses are e.g., multi-

variate Gaussian distributions. I choose a Gaussian distribution in this chapter, because the goal is to mainly exploit first order leakage. Hence, one Gaussian PDF $\sim (\mu_v, \Sigma_v)$ is estimated for each v. Commonly, the noise is assumed to be Gaussian side channel attacks, which is supported by practical measurements [Man07]. $\sim (\mu_v, \Sigma_v)$ is determined by equation 3.7 and 3.8, where n_v denotes the number of traces in the trace set $\mathbf{T}^v$ for one specific v:

$$\mu_v(\mathbf{t}|v) = \frac{1}{n_v} \sum_{n=0}^{n_v-1} \mathbf{t}_n^v \tag{3.7}$$

$$\Sigma_v(\mathbf{t}|v) = \frac{1}{n_v} \sum_{n=0}^{n_v-1} (\mathbf{t}_n^v - \mu_v)(\mathbf{t}_n^v - \mu_v)^T \tag{3.8}$$

The score for each trace can be calculated by the following equation, given the Gaussian profile $\sim (\mu_v, \Sigma_v)$ and the current trace $\mathbf{t}$:

$$\text{score}(v|\mathbf{t}) = -\frac{1}{2}(\mathbf{t} - \mu_v)^T \Sigma_v^{-1} (\mathbf{t} - \mu_v) \tag{3.9}$$

This score is accumulated for each v and each trace. In chapter 4 the score for each key candidate $\hat{k}$ can be assigned by calculating $\hat{k} = v \oplus p$. The $\hat{k}$ with the highest score is the most probable.

3.1.3 Moments Correlating DPA

I use the moments correlating DPA as leakage test to determine LOI and POIs in chapter 4 and as a 3rd order attack on the implementation.
Moments correlating DPA is an advanced DPA proposed by Moradi et al. [Mor16b], which allows to isolate the leakage of higher order moments. Thereby, it is based (exactly like the classical DPA) on the Pearson correlation coefficient, which estimates the linear relationship between two variables. I focus on the profiled version of the moments correlating DPA, called the Moments Correlating Profiled DPA (MCP-DPA) which is a profiled attack method. Hence, the attack is split into two steps, firstly the profiling step, secondly the attack step.

The profiling step is depicted in figure 3.3. For illustration purposes, the profiling step is depicted for one point in time. Please note that a chosen plain- or

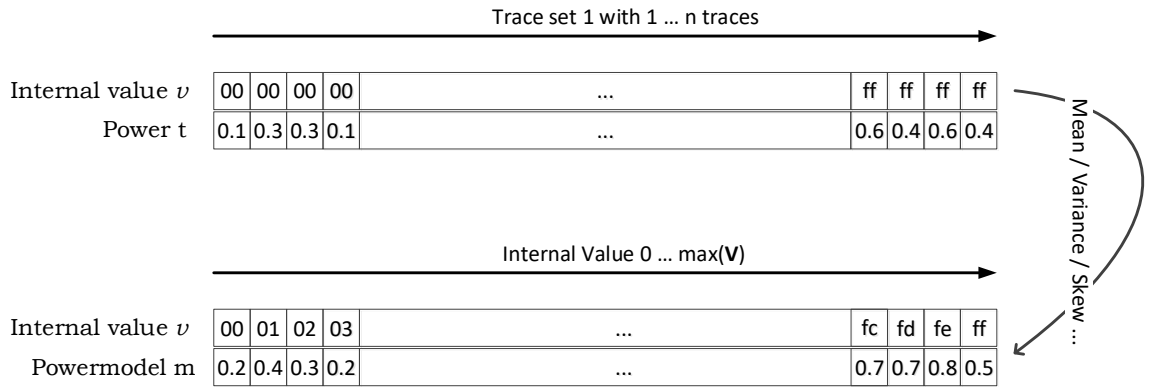


Figure 3.3: Creation of the power model for the moments correlating DPA for one point in time

ciphertext scenario is not strictly necessary; however, in most cases it is possible for the training device. For building the power model $\mathbf{M}$, I collect the trace set $\mathbf{T}_1$ and create one power model $m_{v,z} \in \mathbf{M}$ for each intermediate value $v \in \mathbf{V}$ and each point in time $z \in [0, \gamma]$. As an example, the power model $\mathbf{m}_v \in \mathbb{R}^{1x\gamma}$ is calculated by the mean value for each v, which is described in equation 3.10. $\mathbf{T}_1^v$ denotes all traces of $\mathbf{T}_1$ with the intermediate value v and n_v denotes the number of traces in the trace set $\mathbf{T}_1^v$.

$$\mathbf{m}_v = \frac{1}{n_v} \sum_{n=0}^{n_v-1} \mathbf{t}_n \tag{3.10}$$

The internal value v can be mostly determined by a function with the inputs of the plaintext p and the key k, e.g., the AES-S-box $v = \mathrm{SBOX}(p \oplus k)$. Hence, a power model can be created for each element of $\mathbf{V}$ or a different power model, e.g., the hamming weight or hamming distance.

Figure 3.4 depicts the attack phase for one point in time and one key hypothesis with the attacking trace set $\mathbf{T}_2$.

As step one in figure 3.4, vector $\mathbf{m}_{z,\hat{k}}$ is computed for each point in time $z \in [0, \gamma]$ and each key hypothesis $\hat{k}$, following equation 3.11.

$$\mathbf{m}_{z,\hat{k}} = (m_{v(t_0),z}, \ldots, m_{v(t_n),z}) \tag{3.11}$$

Where $v(\mathbf{t}_n)$ determines the v individually for each trace and key hypothesis $\hat{k}$. $m_{v,z}$ denotes the element of m_v for the intermediate value v and point in time z. Thus, each $\mathbf{m}_{z,\hat{k}}$ has length n.

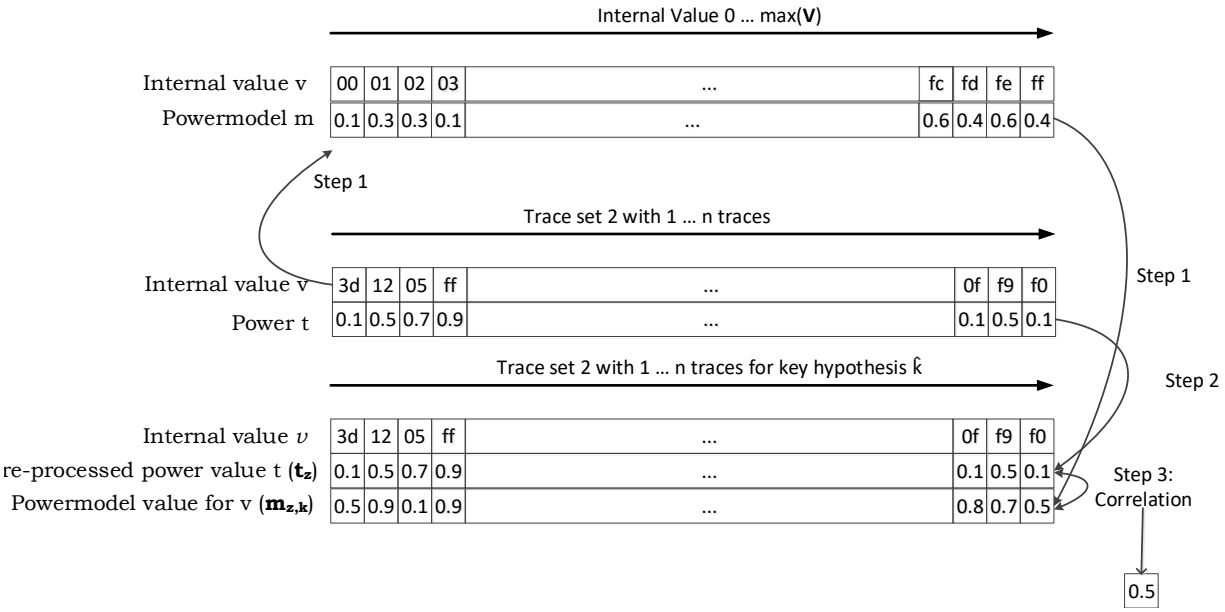


Figure 3.4: Attack phase of moments correlating DPA for one point in time and one key hyportheses

Step 2 in figure 3.4 corresponds to, assembling the vector $\mathbf{t}_z$, which denotes the samples of all traces at the point in time z.

$$\mathrm{corr}_{z,\hat{k}} = \varrho(\mathbf{m}_{z,\hat{k}}, \mathbf{t}_z) \tag{3.12}$$

Step 3 in figure 3.4 depicts the computation of the correlation vector $\mathrm{corr}_{z,\hat{k}}$, which can be determined for each key candidate and each point in time.

When attacking higher order statistical moments, the power measurement samples and the power model have to be modified. For the power model instead of the mean for every v, the variance for the second order or the skew for the third order is calculated. Depending on the targeted statistical moment, the measurement values in $\mathbf{T}_2$ have to be pre-processed, e.g., according to Equation 3.13 or 3.14 for the second or third statistical moment.

$$\hat{\mathbf{T}}_v = \mathbf{T}_v - \mathrm{mean}(\mathbf{T}_v) \tag{3.13}$$

$$\hat{\mathbf{T}}_v = \frac{\mathbf{T}_v - \mathrm{mean}(\mathbf{T}_v)}{\mathrm{variance}(\mathbf{T}_v)} \tag{3.14}$$

MCP-DPA allows us to perform an attack, using the third statistical order on the threshold implementations as a reference to the performed template attacks. Furthermore it is suited as a correlation based leakage test to determine LOIs and POIs.

MCP-DPA as a Correlation Based Leakage Test The goal of leakage tests is to show the data dependency of the intermediate value v and the measured traces $\mathbf{T}$. Durvaux et al. [Dur16] proposed a correlation based leakage test. This test is very similar to the MCP-DPA. The only difference is the use of cross validation sets. The use of cross validation sets enables a more accurate estimation of the leakage in case of a low number of available traces. During the measurement the trace set $\mathbf{T}_{all}$ is collected. When using, e.g., 10 cross-validation sets, $\mathbf{T}_{all}$ gets split into 10 sets. $\mathbf{T}_{all,0}, ..., \mathbf{T}_{all,9}$. During the first run $\mathbf{T}_{all,0}, ..., \mathbf{T}_{all,8}$ is used for as profiling trace set ($\mathbf{T}_1$ in the MCP-DPA case). $\mathbf{T}_{all,9}$ is used as attack trace with the correct key only ($\mathbf{T}_2$ in the MCP-DPA case). During the second run $\mathbf{T}_{all,0}, ..., \mathbf{T}_{all,7}, \mathbf{T}_{all,9}$ is used for as profiling trace set ($\mathbf{T}_1$ in the MCP-DPA case). $\mathbf{T}_{all,8}$ is used as attack trace with the correct key only ($\mathbf{T}_2$ in the MCP-DPA case). This is carried out, until each cross validation set is attacked once. The resulting correlation trace is calculated by the mean of the correlation traces of all sets.

3.2 Evaluation Methods used in Chapter 5

3.2.1 Attack Concept against Exponentiations

A common structure of an exponentiation algorithm is shown in Algorithm 1. The secret d with n bits is processed bitwise. Depending on the bit-value d_o, the algorithm executes a write and a read operation on register a in case $d_o = 1$ and register b in case $d_o = 0$. If an attack can distinguish the write and read accesses to register a or b, the key can be revealed. The segmentation borders of the loop iterations must be known a priori, or can often be derived from visual inspection or comparison of shifted trace parts. With known segment borders, each loop iteration can be split into one trace-segment, which corresponds to one bit of the key in this case (such attacks are referred to as horizontal attacks). The trace for measuring n exponent bits consists of n trace segments $\mathbf{t_d} = (t_{1+(d-1)\cdot\gamma}, ..., t_{d\cdot\gamma})$ with $d \in [1, n]$, each of length γ (time-samples), which is

Algorithm 1 Typical structure of an exponentiation algorithm, taken from [Hey12a]

Input: Secret $d = d_n d_{n-1} ... d_2 d_1$ with $d_o \in \{0, 1\}$

```
1: for o = n downto 1 do
2:     if d_o = 1 then
3:         c ← c^2 + a
4:         a ← c
5:     else
6:         c ← c^2 + b
7:         b ← c
8:     end if
9: end for
```

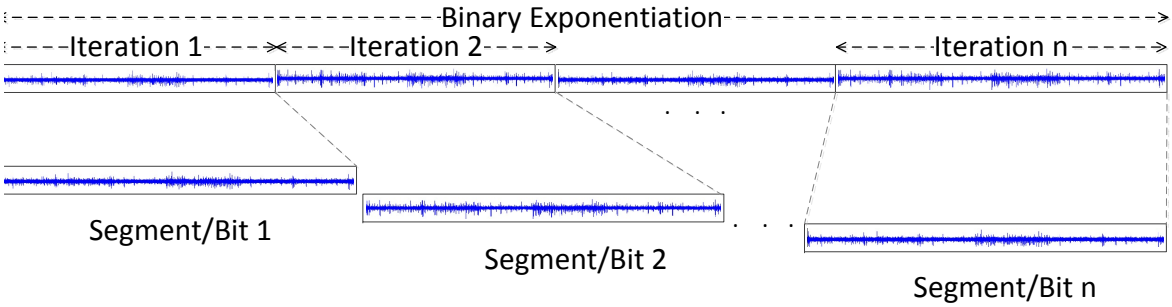


Figure 3.5: Creating the trace segments from the collected trace

referred to as its dimensionality (of features). This procedure is shown in figure 3.5. For analyzing and attacking the measurement data, the $n \times \gamma$ matrix $\mathbf{T}$ is constructed by placing each segment in one row. This result is depicted in figure 3.6. The attack is successful if labeling the rows/segments, which each corresponds to one key bit, recovers the true secret.

Clustering based attack of Heyszl et al. [Hey14] The attack in chapter 5 is based on the algorithmic approach to clustering-based non-profiled attacks on exponentiations of [Hey14]. Thus, I will briefly describe this approach. Hence, the $\mathbf{T}$ is created in the same way as described above. Heyszl et al. use as preprocessing a simple trace compression technique, the sum of squares. Each sample is squared and afterwards the samples, belonging to the same clock

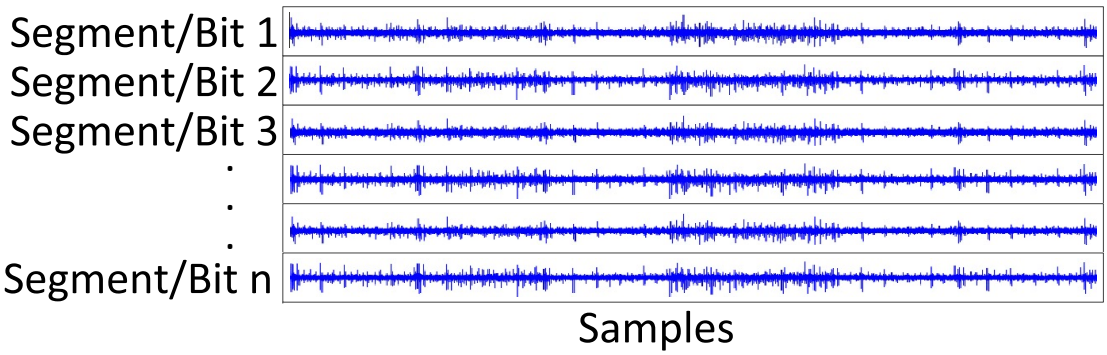


Figure 3.6: Resulting trace matrix **T**

cycle are summed to one sample. After preprocessing, the key is tried to be revealed with the help of the k-means clustering algorithm. However, the use of such simple trace compression techniques is shown to be not optimal [Hey12b]. Thus, I improve the algorithmic approach by using PCA as preprocessing and the "expectation-maximization" clustering algorithm.

3.2.2 Principal Component Analysis (PCA) for Dimensionality Reduction and Feature Selection

PCA maximizes the variance between all samples, but does not take into account the class-labels and class-dependent variances. In other words, PCA transforms the original space into a subspace, where the variance between all elements is maximized, e.g., all elements (independent of the color) of figure 3.1. The maximization variances supports a basic assumption in side channel analysis: A power trace contains data/key-dependent differences and thus variance. The PCA can be calculated using Singular Value Decomposition (SVD). The SVD of $\mathbf{T} \in \mathbb{R}^{nxy}$ is denoted in equation 3.15.

$$\mathbf{T} = \mathbf{U} * \boldsymbol{\Sigma} * \mathbf{V}^* \tag{3.15}$$

$$\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{x} = \lambda\mathbf{x} \tag{3.16}$$

$$\mathbf{T}^{\mathrm{T}}\mathbf{T}\mathbf{x} = \lambda\mathbf{x} \tag{3.17}$$

U and **V** are unitary matrices containing the left- and right-singular vectors and $\boldsymbol{\Sigma}$ is a diagonal matrix containing the singular values, where * denotes the Hermitian transpose of a matrix.
The SVD can be calculated by solving two eigenvalue problems. The matrix **U** can be determined by the eigenvectors of equation 3.16 and the matrix **V** can be determined by the eigenvectors of equation 3.17. Finally, the entries on the diagonal of $\boldsymbol{\Sigma}$ are the square roots of the eigenvalues determined by equation 3.16 or equation 3.17. [Ban14]

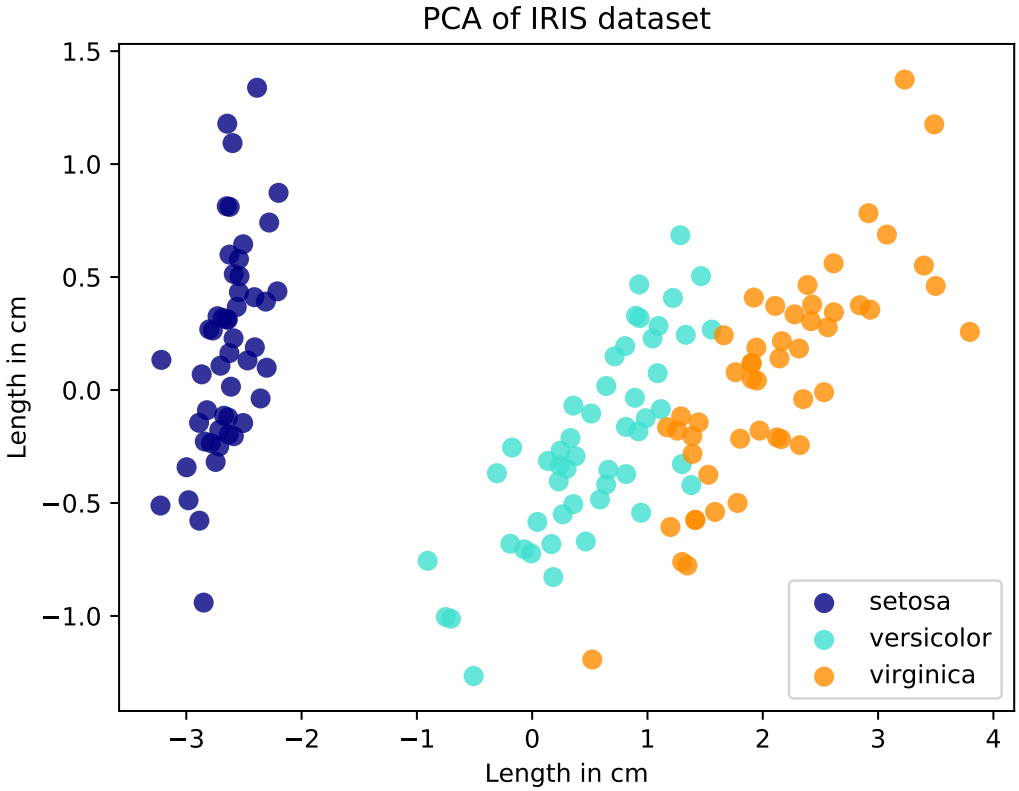


Figure 3.7: IRIS dataset transformed with PCA

The original data can be completely restored by calculating $\mathbf{U} * \boldsymbol{\Sigma} * \mathbf{V}^*$ (no information is lost). The transformation into the subspace consists of $\mathbf{T}_T = \mathbf{U} * \boldsymbol{\Sigma}$. The matrix $\mathbf{U} * \boldsymbol{\Sigma}$ consists of column vectors $(\mathbf{PC}_1, ..., \mathbf{PC}_r)$ with r being the number of row-vectors, where $\mathbf{PC}_j$ being a column-vector of shape $n \times 1$, which is called a principal component. The maximum number of components

is equal to $r = \max|PC| = \min(n, \gamma)$. In case of the application in chapter 5 the length γ is usually much larger than n; hence the number of principal components is usually n. After PCA, the components are ordered according their variance, which can be found in the diagonal matrix Σ. In our experiments in chapter 5, I normalize the variances of the principal components to one, i.e. I directly use $\mathbf{T}_{\mathrm{PCA}} = \mathbf{U}$ as transformation instead of $\mathbf{U} * \Sigma$.
Figure 3.7 shows the first two principal components of the IRIS dataset (shown in figure 3.1) and shows a good separation of the classes.

In this example two output dimensions were used from PCA. However, for practical use-cases the optimal number of dimensions used for the subsequent analysis is difficult to determine. Thus, a "rule-of-thumb" is commonly applied. The matrix $\boldsymbol{\Sigma}$ contains the variances sorted by their value for each principal component. Afterwards, the largest entries of $\boldsymbol{\Sigma}$ are selected, until the sum of the selection reaches 90 % of the sum of all values. Then, the eigenvectors of the selected eigenvalues are taken for transformation; note that the number of selected eigenvectors determines the number of output dimensions. This rule is called the 90 % explained variance rule [Jol02]. [2] I apply the same rule to estimate the number of components for LDA.

3.2.3 Used Clustering Algorithms

Clustering algorithms can be split into supervised, semi-supervised and unsupervised algorithms. The focus on *non-profiled* attacks in chapter 5 restricts the choice to *unsupervised* algorithms. Clustering algorithms are used to label data, e.g., the flower-types of figure 3.1 and can be used in a non-profiled attack to partition n trace-segments into classes according to their secret exponent values.
Unsupervised clustering algorithms use basic assumptions for the data to be labeled, e.g., the Density Based Spatial Clustering of Applications with Noise (DBSCAN) [Est96], the hierarchical based DIvisive ANAlysis Clustering (DIANA) [Kau90] or the PDF-based k-means and expectation maximization algorithm. Due to the above described principle in chapter 3.2.1, the search for two (multidimensional, overlapping) Gaussian PDFs is assumed. The Gaussian PDFs are expected, due to a Gaussian shaped noise, which can have many

[2] Note that this rule cannot be applied to the PCA in chapter 5, because the highest rank components mainly contain high-variance noise

sources, e.g., electrical noise from resistors and amplifiers, quantization noise and noise from surrounding magnetic fields. Hence, the K-Means and expectation maximization clustering algorithms are used in chapter 5, which try to find a given number of Gaussian PDFs in the data.

3.2.3.1 K-Means Clustering Algorithm

K-means tries to find a definable number e of clusters in data $\mathbf{T} \in \mathbb{R}^{nx\gamma}$. In chapter 5 $e = 2$, due to 2 possible bit values. Each row **t** in **T** is treated as one sample with dimensionality γ. In general this labeling-problem is NP-hard; however k-means uses an efficient heuristic to find local optima for clustering. K-means initializes e random samples, assigns class labels to these and assumes that these points are the center of the e clusters. Now the iterative part of the clustering algorithm begins:

1. Each **t** is assigned to the cluster of the closest cluster center. In chapter 5 the Euclidean distance is used, due to the Gaussian noise assumption.

2. After each **t** was assigned to a cluster, a new cluster center is calculated by calculating the mean over all assigned **t** to the cluster

3. The above steps are repeated until a maximum number of loops is reached, or the cluster means do not change anymore. A maximum of 10 000 iterations is used in chapter 5.

In order to find a global optimum, instead of a local optimum, the k-means clustering is carried out multiple times with new random start samples as class centers. In chapter 5 this is performed 200 times and the execution, which produces the smallest summed distance between the cluster center and the according **t** is taken.

3.2.3.2 Expectation Maximization Clustering Algorithm

The goal is again to label all **t** of **T**. The expectation maximization tries to estimate the free parameters of the classes' assumed PDFs. The choice of free parameters depends on the assumed probability distribution model, hence shape of the clusters. *Expectation-maximization clustering* provides more free parameters than k-means clustering, which leads to a generally improved approximation of the cluster distributions, which usually leads to better classification results.

For initialization of the expectation maximization parameters, k-means is used in our case. Afterwards, the algorithm is based on repeated expectation and maximization steps. During these iterations the maximum likelihood means and covariances for the Gaussian distribution are derived (instead of the means only by k-means). The result is a classification and a class-membership probability for each $\mathbf{t}$, which indicates the reliability of correct classification for each segment (resp. secret exponent bit).
The number of free parameters in the clustering algorithm can be chosen. I assume that the cluster shapes are mainly defined by Gaussian distributed noise. Additionally, I assume the noise being independent of the processed bit value. Hence, I chose to estimate two means and one joint full covariance matrix. By applying PCA the dimensions of the data are reduced, which also reduces the number of mean and covariance elements, which need to be estimated.

3.2.4 Classification Errors and Required Brute-Force Complexity

If the recovered exponent is incorrect, faulty bits need to be identified, which is usually hard. As described by Heyszl et al. [Hey14], an attacker can use the bits' probabilities of correctness to judge which need to be trialed for correctness and follow a simple strategy to enumerate possible keys. This strategy leads to an estimated remaining Brute Force Complexity (BFC), which I use to assess practical attack outcomes. Better, even optimal, key enumeration strategies [Vey13a, Vey13b] will result in a lower amount of required brute-force effort if the attacker applies them. However, the typically large key sizes in asymmetric cryptography, e.g., RSA-4096, make the enumeration of the key rank challenging for attackers as well as evaluators. Thus, the required brute-force complexity is often estimated. The said BFC which is used instead can be seen as an upper bound for the rank of the correct key as derived from an optimal enumeration.
I chose to use the silhouette index score [Rou87] to determine the bits' error probability. It is based on the cumulative distance of each trace-segment to other trace-segments of each cluster. The silhouette index is calculated for every given transformed trace by PCA $\mathbf{T}_{\mathrm{PCA}}$, which corresponds to one row of $\mathbf{T}_{\mathrm{PCA,pc:pc+i}}$, with $\mathcal{C}_1$ being the set of trace segments $\mathbf{t_d}$ of the same cluster like $\mathbf{T}_{\mathrm{PCA}}$ (determined by the expectation maximization algorithm) and $\mathcal{C}_2$ being the

set of trace segments belonging to other clusters. With the distance function dist$(\mathbf{a}, \mathbf{b})$, where the distance between $\mathbf{b}$ and $\mathbf{a}$ is calculated. I use Euclidean distance due to the Gaussian noise assumption.
The silhouette index s is computed as:

$$s(\mathbf{t_d}, \mathcal{C}_1, \mathcal{C}_2) = \frac{f(\mathbf{t_d}, \mathcal{C}_1) - f(\mathbf{t_d}, \mathcal{C}_2)}{\max(f(\mathbf{t_d}, \mathcal{C}_1), f(\mathbf{t_d}, \mathcal{C}_2))} \tag{3.18}$$

$$f(\mathbf{t_d}, \mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{x} \in \mathcal{C}} \text{dist}(\mathbf{t_d}, \mathbf{x}) \tag{3.19}$$

After calculating the score for all n segments $\mathbf{t_d}$, the ones with the lowest s are brute-forced first, while including an increasing number of bits [Hey14]. Let q be the last bit which is trialed until the correct exponent is found, then $2^{(n+1+1)}$ different exponents have to be tested at maximum which can be referred to as remaining *brute-force complexity* after the attack [Hey14]. One additional bit is included for both possibilities to assign labels to the two classes. It equals $2^{(n+1+1)}$ at maximum and 2^1 at minimum.
An alternative to estimate the reliability of exponent bit classifications is using a Support Vector Machine (SVM) to get a hyperplane, which separates two clusters. However, I could not achieve a significant improvement using this and it significantly increases the computational complexity. Another way is to use the discriminant score[Cho13], which measures the reliability by calculating the two probabilities of each sample to be part of a cluster similar to equation 3.9. I derived similar results as for the silhouette index, however, with significantly increased computational complexity.

3.2.5 Carried Out Template Attack

I apply the template attack in chapter 5 to compare the performance of the non-profiled attack to a profiled attack. The last chapter showed that LDA is efficient in exploiting leakage for a profiled attack. However, for a fair comparison to the unprofiled case I also apply PCA. The same preprocessing is especially important to estimate the capability of our unprofiled approach to exploit leakage. The template attack can explain if PCA is able to extract useful features from the original trace and if the expectation maximization clustering algorithm can efficiently exploit that leakage.

I collect one trace for profiling with a known key. This trace is again cut into the trace-segments, resulting in the matrix $\mathbf{T}$. Applying PCA, results in the matrix $\mathbf{T}_{PCA}$. To train the template attack, I create one multidimensional Gaussian PDF $\sim (\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$ for each possible value of the key-bits (zero and one). To lower the estimation error of the covariance matrix, I calculate one common covariance matrix for both key-bit values (the pooled covariance matrix).

For evaluation I collect one $\mathbf{T}$ with an unknown key and apply PCA (with the transformation matrix of the profiling phase). Afterwards, I calculate for each trace segment $\mathbf{t_d}$ the score for both templates (values zero and one, see equation 3.9), and choose the bit value with the highest score/probability for each trace-segment. [Cho13] Note that in this case the score is determined by equation 3.9 and not the silhouette coefficient. Using these scores, it is possible to compute the brute force complexity as described in section 3.2.4, by brute forcing the bits with the lowest probability of corresponding classes and compare the results directly to the unprofiled case.

Chapter 4

Multiprobe Attacks on Symmetric Ciphers

In the last chapters I introduced the dimension reduction technique LDA, the used analysis tools in this chapter and the related work to masking schemes. In this chapter I show how multiprobe attacks can significantly impact the security of an implemented masking scheme by practical measurements.

I first briefly explain selected theory to TIs in section 4.1 and the attack concept in section 4.2, which is applicable to a majority of published masking schemes. Afterwards, I describe the chosen implementation in section 4.3. Then, I introduce the measurement setup in section 4.4. The profiling phase of the template attack and LDA is described in section 4.5. Afterwards, I present practical results for the attack carried out with multiple localized EM probes in section 4.7. I firstly evaluate the EM probe measurements separately and secondly combined. Finally, I analyze in detail how LDA combines multiple probes. Then, I compare the results with the power side channel. I summarize the contribution and findings in Sect. 4.8.
Results of this chapter have been published at HOST conference in 2018 in the following publication: [Spe18] The presented results are originated in the collaboration with the coauthors.

4.1 Selected Theory of Threshold Implementations

A Threshold Implementation (TI) for hardware design is highly related to multiparty computation and threshold cryptography [Nik06]. The idea is to split a secret intermediate value into multiple independent parts, called shares, and distribute the computation such that, a malicious party participating in a multi-

party computation cannot reveal the secret, since the other shares are unknown. One possibility to create the shares is to split the secret intermediate value with the help of one (or multiple) individual random numbers. If r shares can be known by an adversary, without revealing the secret, an implementation is called r-th order secure and requires at least $r + 1$ shares. The shares are modified by a set of functions $f_0, ..., f_r$. The security of a TI is provable, if the functions fulfill three properties, *correctness*, *non-completeness*, and *uniformity*. After applying the functions to the shares, the correct result can be obtained by combining the shares. This property is termed *correctness*, defined in Equation 4.1.

$$v = \sum_{n=1}^{r} v_n = \sum_{n=1}^{r} f_n(...) \tag{4.1}$$

Non-completeness, defined in Equation 4.2, reflects the property that each function operates with at most $i - 1$ shares. While this property is easy to fulfill for linear operations, it is difficult to achieve for non-linear operations.

$$\begin{aligned} v_1 &= f_0(x_2, ..., x_r) \\ v_2 &= f_1(x_1, x_3, ..., x_r) \\ &... \\ v_r &= f_r(x_1, ..., x_{r-1}) \end{aligned} \tag{4.2}$$

The third property is named *uniformity* and is known to be the hardest property to achieve. A function f is called *uniform* if $v = f(x)$ is uniform given that the input x is uniform. Sometimes the output of a f is used for another f. Thus, all output values of each f must be uniformly distributed. [Bil14]

For the most basic operations of ciphers, e.g., inverting in GF_{2^8}, a uniform splitting exists, with the drawback of usually requiring more shares than $i + 1$. To reduce the number of shares and thus the implementation size, remasking ensures uniformly distributed inputs for each function. Most publications "circumvent" the uniformity requirement by remasking, to achieve a small implementation. This has the drawback that more random numbers have to be available for the encryption. Hence, the amount of random numbers per encryption is required as an additional rating factor, over size and order of security for masked implementations [Mor11].

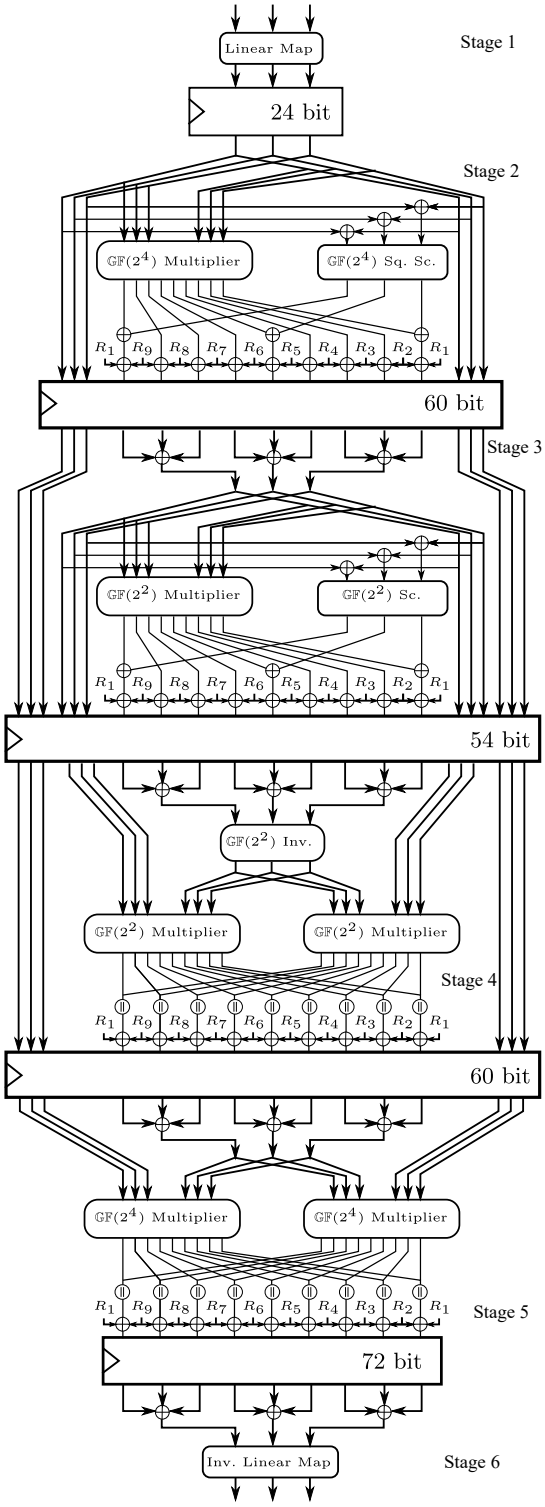


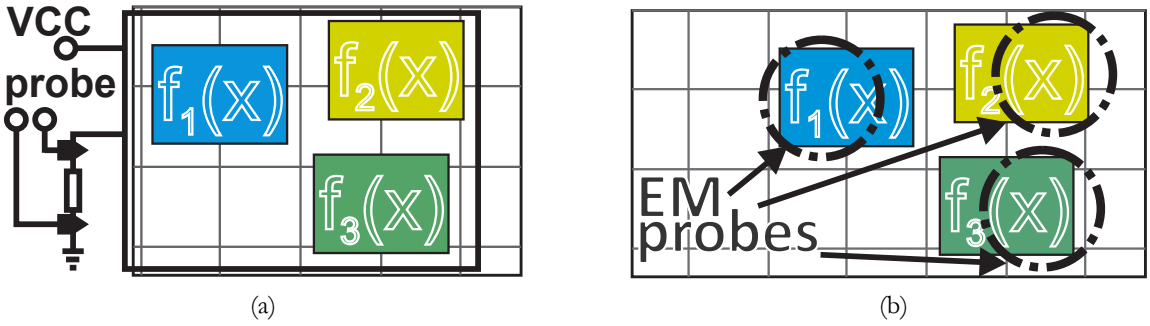
Figure 4.1: Implemented TI [De 16b]

4.2 Attack Concept on Threshold Implementations

To illustrate the advantages of multiprobe localized EM measurements, let us consider the case of a power-based side-channel attack first. As depicted in Figure 4.2a, using this approach results in a global view on the Device-Under-Test (DUT). Hence, all shares (the functions f_1,f_2,f_3) must be attacked with one common set of traces. To overcome this limitation, localized multi-probe EM measurements eventually allows to spatially separate and observe the shares independently, thereby enabling a detailed view on the shares.
Figure 4.2b illustrates the case of using multiple probes to attack TIs. By positioning the probes at different locations above the die, we benefit (in theory) from an optimal position for each share to measure them individually. This allows to directly exploit the leakage of each share which exceeds the scope of previously known attacks. In an attack on both masks (each processed by one share) and the thereby protected sensitive value (third share), we try to first recognize the masks and subsequently use them to unmask the sensitive value. Figure 4.2b shows a clear theoretical benefit, which has to be evaluated by practical experiments.
The original TI paper of [Nik06] states: "*Our proposal […] makes it more difficult to implement the attack, because the parallel computation of the n shares lowers the signal-to-noise ratio. […] Since we assume that all shares are uncorrelated, the number of shares n effectively multiplies the bit-width. As a consequence the number of samples needed in the profiling step is greatly increased.*" Hence, the security assumption of threshold schemes is that observing the leakage of individual shares is difficult [Nik06]. In Chapter 4.7 this underlying assumption is thereby practically challenged by using multiple localized EM probes and carrying out a template attack.
The attack is divided in the following steps: All steps are explained in detail in section 4.5

1. Detection of Points of Interest (POI) and Location of Interest (LOI)
   This step defines the LOIs and the POIs with the help of the moments correlating DPA as a leakage test, defined in chapter 3.1.3

2. Training of Linear Discriminant Analysis (LDA)
   All selected POIs **q** are input into the LDA. In combination with the known v the transformation matrix **W** is calculated, which is described



(a) (b)

Figure 4.2: Measurement principle for power (cf. Figure 4.2a) and multi-probe, localized EM measurements (cf. Figure 4.2b).

in chapter 3.1.1. From this point on, all traces are transformed with the help of **W**.

3. Training of the template attack
   The transformed output is used for the training of the template attack. In combination with the known v the Gaussian PDFs are calculated for all v, shown in chapter 3.1.2.

4. Applying LDA and carrying out the template attack
   During the attack phase the trace is transformed with **W**. Afterwards, the probability for each possible v is derived with the help of the PDFs trained in the last step and equ. 3.9. Then, the probability is assigned to each key candidate by combining the p of the attacked byte and each possible v, where the key candidate is derived by $\hat{k} = v \oplus p$.

4.3 Implementation of the Threshold Scheme

In this section I present implementation details to the analyzed threshold scheme. The implemented S-box is related to the Canright S-box [Can05] and uses the same mathematical blocks, e.g., linear map, multiplications, squaring and square scaling (Sq.Sc.) in GF_{2^4} and GF_{2^2}. In general, the AES S-box performs an inversion over $GF(2^8)$. The Canright S-box splits this inversion over $GF(2^8)$ into multiple multiplications in the subfield $GF(2^4)$. A simplification can be achieved with splitting the $GF(2^4)$ operations into $GF(2^2)$, e.g., in $GF(2^2)$ a squaring operation is the same as an inverting operation for a polynomial ba-

sis. [Can05]

The implemented second-order secure AES S-box threshold implementation of De Cnudde et al. [De 16b] was implemented with three shares on a Xilinx Spartan 6 FPGA, which is depicted in figure 4.1. The S-box is split into six stages, with $\mathbf{a_i} = (a_i^1, ..., a_i^8)$ being the input byte for share i at the first stage:

"**First Stage.** The first operation occurring in the decomposed S-box performs a change of basis through a linear map. Its masking requires instantiating this linear map once for each share i. This mapping is implemented in combinational logic and it maps the 8-bit input $(a_i^1, ..., a_i^8)$ to the 8-bit output $(y_i^1, ..., y_i^8)$ for each share i " [De 16b]
Now the 8-bit $\mathbf{y_i}$ gets split into two 4-bit values $\mathbf{b_i} = \mathbf{y_i^1}, ..., \mathbf{y_i^4}$ and $\mathbf{c_i} = \mathbf{y_i^5}, ..., \mathbf{y_i^8}$.

"**Second Stage.** [...] The resulting equations are given by:
$\mathbf{d_1} = \mathbf{b_1} \otimes \mathbf{c_1} \oplus \mathrm{SqSc}(\mathbf{b_1} \oplus \mathbf{c_1})$
$\mathbf{d_2} = \mathbf{b_1} \otimes \mathbf{c_1}$
$\mathbf{d_3} = \mathbf{b_1} \otimes \mathbf{c_3}$
$\mathbf{d_4} = \mathbf{b_2} \otimes \mathbf{c_1}$
$\mathbf{d_5} = \mathbf{b_2} \otimes \mathbf{c_2} \oplus \mathrm{SqSc}(\mathbf{b_2} \oplus \mathbf{c_2})$
$\mathbf{d_6} = \mathbf{b_2} \otimes \mathbf{c_3}$
$\mathbf{d_7} = \mathbf{b_3} \otimes \mathbf{c_1}$
$\mathbf{d_8} = \mathbf{b_3} \otimes \mathbf{c_2}$
$\mathbf{d_9} = \mathbf{b_3} \otimes \mathbf{c_3} \oplus \mathrm{SqSc}(\mathbf{b_3} \oplus \mathbf{c_3})$ " [De 16b]

$\otimes$ denotes a multiplication and $\oplus$ denoted an addition/XOR-operation. The non-completeness property can be implemented "such that combinations of up to $[r - 1]$ component functions are independent of at least one input share of each variable" [De 16b]. For the implementation, the functions are split according to the non-completeness property into shares: share1: d1 to d3; share2: d4 to d6; share3: d7 to d9. I apply the separation into three shares in the same manner for all six stages. $R_1, ..., R_9$ denote the masks used to ensure an uniform input for the next stage. The concept is explained in chapter 4.2.

"**Third Stage.** This stage is similar to the second stage. Here, the received nibbles are split in 2-bit couples for further operation. The Scaling operation (Sc) replaces the similar affine Square Scaling and is executed alongside the mul-

tiplication in GF(2^2). By combining both operations, we can share the total function by taking again the non-completeness into account. " [De 16b]

"**Fourth Stage.** The fourth stage is composed of an inversion and two parallel multiplications in GF(2^2). The inversion in GF(2^2) is linear and is implemented by swapping the bits using wires and comes at no additional cost. The outputs [...], to form 4- of the multiplications are concatenated, denoted by ⓘ bit values in GF(2^4) " [De 16b]

"**Fifth Stage.** Stage 5 is similar to Stage 4. The difference of the two stages lies in the absence of the inversion operation and the multiplications being performed in GF(2^4) instead of GF(2^2). " [De 16b]

"**Sixth Stage.** In the final stage of the S-box, the inverse linear map is performed. By using a register between Stage 5 and Stage 6, we can remask the shares and perform a compression before the inverse linear map is performed resulting in only three instead of nine instances of inverse linear maps. As with the linear map, no uniform sharing of its inputs is required for security. However, in the full AES, this output will at some point reappear at the input of the S-box, where it undergoes nonlinear operations again. This is why we insert the remasking." [De 16b]

Note that in between of all stages, the data is remasked with 9 masks $R_1, ..., R_9$ to ensure a uniform data distribution for the next stage; thus, ensuring uniformity. Each of the masks is applied with the same width of the data path. Hence, requiring as many random bits as data bits each clock cycle.

A TI-typical initial split of the share-inputs was chosen: Share 1: Sboxin ⊕ mask1 ⊕ mask2, share 2: mask1, share 3: mask2 [De 17]. Beside the initial split and implementation, the placement of the shares is important to prevent first order leakage due to coupling. To ensure that the single shares do not couple, I controlled the placement with the help of so-called Pblocks in Xilinx ISE, as depicted in figure 4.3. These blocks restrict entities to defined locations, thereby reducing the risk that shares are coupled which I additionally confirmed by manually checking the implementation output and performing practical measurements to verify the behavior. Beside the placement, undesired "optimizations" by the synthesis have to be avoided. Thus, I used several con-

straints provided by Xilinx ISE, e.g., the keep hierarchy constraint and manually instantiated LUTs. To still be able to reduce the size of the S-box to a minimum I chose a byte-serial, single S-box implementation. Hence, I expect very similar leakage for all bytes of an AES round and evaluate all results for byte 0 of the state only.

4.4 Measurement Setups

In the following, I briefly describe the measurement setups and the used equipment for practical evaluations.

4.4.1 High Resolution EM Measurement Setup

For localized EM measurements to work, the DUT has to be decapsulated. The decapsulation of the DUT was realized by chemical etching of the DUT. As DUT I used a Spartan 6 xc6slx9 FPGA.
I placed the high-resolution near-field coils at a distance of about 30 µm above the die. I use the following probes:

1. Langer ICR HH 150-6:
   Horizontal magnetic field probe with an inner diameter of 150 µm, 6 Windings and 2.5 MHz to 6 GHz frequency span

2. Langer ICR HH 150-27:
   Horizontal magnetic field probe with an inner diameter of 150 µm, 27 Windings and 1.5 MHz to 6 GHz frequency span

3. Langer ICR HH 100-6:
   Horizontal magnetic field probe with an inner diameter of 100 µm, 6 Windings and 2.5 MHz to 6 GHz frequency span

Please note that, using different probes does not significantly affect the result, because during preprocessing and profiling phase of the template attack the characteristics of every probe are implicitly respected. The maximum bandwidth of each probe is 6 GHz with a built-in 30 dB preamplifier. Additionally, a 30 dB Langer PA303 amplifier was attached to each probe, such that each resulting signal is amplified by 60 dB in total. This amplification avoids quantization noise caused by the oscilloscope. The used oscilloscope is a LeCroy

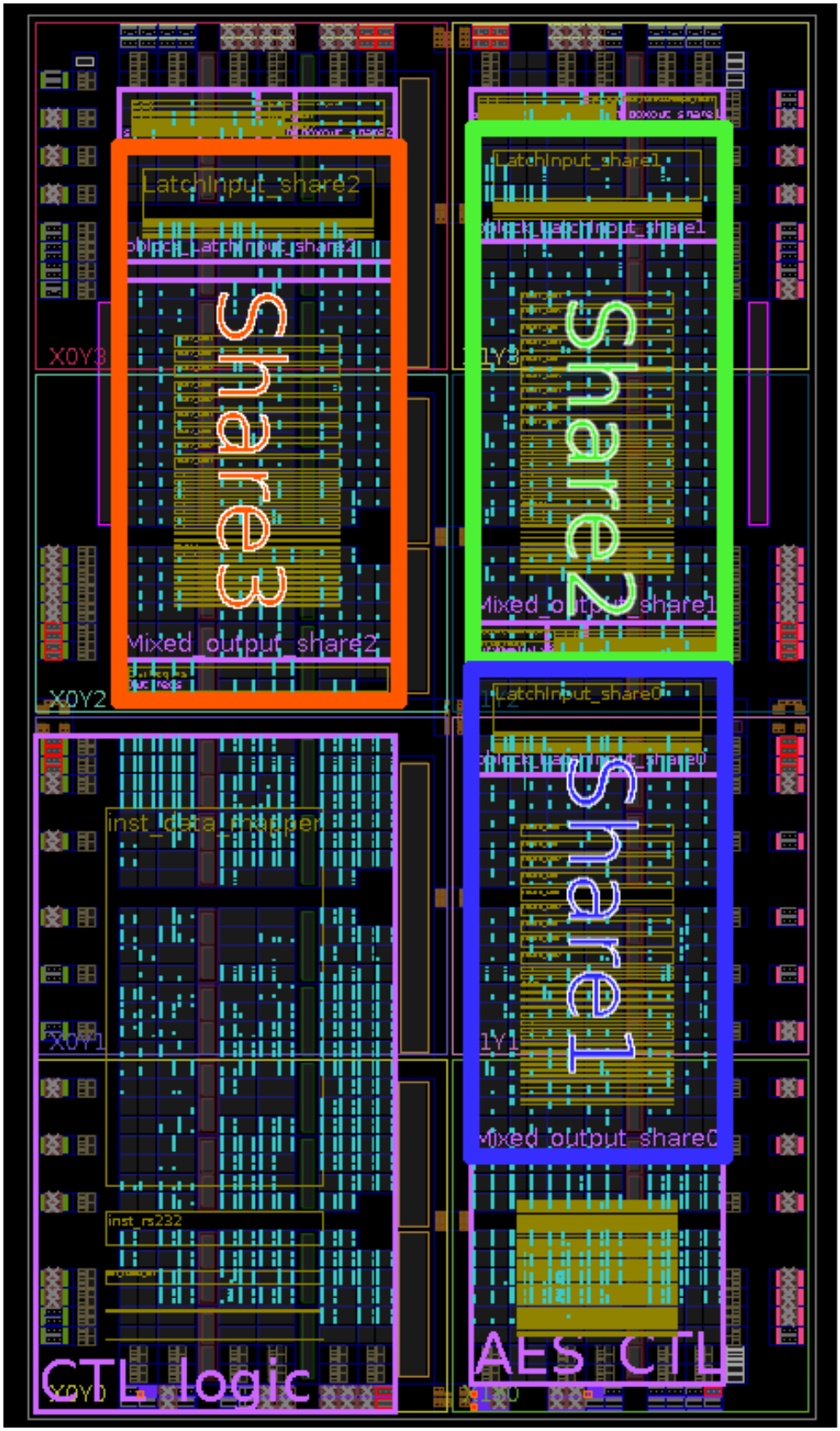


Figure 4.3: Floorplan of TI with 3 shares on Spartan 6 FPGA.

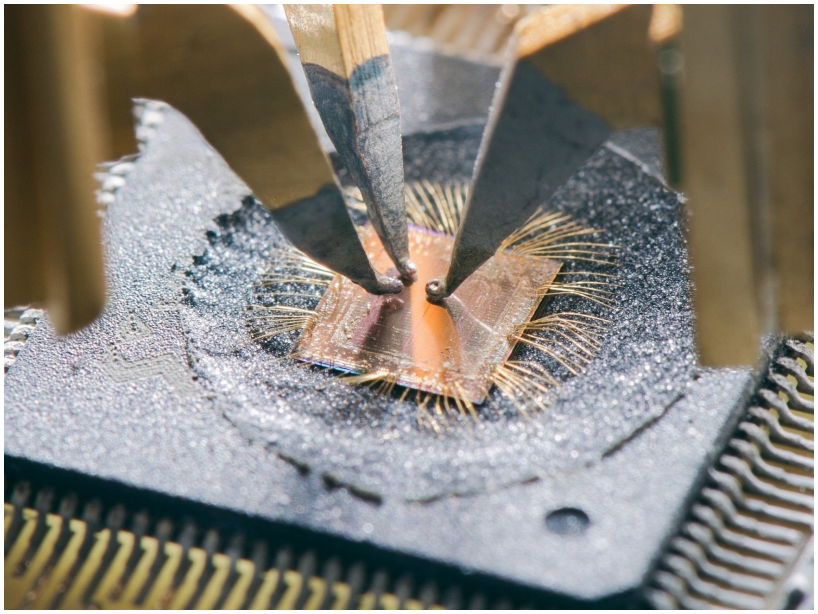


Figure 4.4: High resolution EM measurement probes on FPGA die surface

WavePro 725 Zi with 2.5 GHz bandwidth and a samplingrate of 5 GS/s, which was synchronized to our DUT. All channels have an adjusted offset and a resolution, such that no clipping occurs.

To move and place the measurement coils accurately above the die, the DUT was mounted on an X-Y-Table. For the measurements with one measurement coil, 225 (15 x 15 grid, equally spaced) measurements positions were used in an area of 2730 µm x 2500 µm with the Langer ICR HH 150-6. When using three probes simultaneously, I placed them by hand and therefore very likely did not achieve optimal placement.

4.4.2 Power Measurement Setup

Most of the setup is kept the same for the power-based measurement, i.e., the same FPGA using the same design. Instead of the H-field probes and amplifiers, a differential probe (LeCroy AP033) measures the voltage drop across a 10 Ω shunt resistor.

4.5 Profiling Steps for the Template Attack

In this section I explain and evaluate the first three steps of the attack concept, explained in section 4.2, which contain the profiling phase of the attack. I firstly determine the LOIs and POI and afterwards the training of LDA and the template attack. The access to masks, key, and plaintext during profiling is assumed.

4.5.1 Detecting Location and Points of Interest

For determining the LOIs and POIs, I use the first order MCP-DPA as a leakage test, which is explained in section 3.1. [Dur16] LOIs and POIs are selected where this test returns high values.

To carry out the leakage test in the spatial- and time-domain, the targeted intermediate value has to be known for equ. 3.10 to perform the profiling. For a straight-forward AES S-box implementation, the intermediate value is defined by the AES S-box Sboxin $= (k_{i,n} \oplus p_{i,n})$ for the subkey and plaintext of target byte $i \in [0, 15]$ and trace number $n \in [1, ..., n]$. However, due to multiple shares in this implementation, I carry out one test for the value of each share, i.e., share 1: Sboxin $\oplus$ mask1 $\oplus$ mask2, share 2: mask1, share 3: mask2. Hence, one model **m** is created for each test by equation 3.10. Using the calculated model **m** for equation 3.11 and only evaluating the correct key for equation 3.12 gives the data dependency between the measured traces **T** and the targeted intermediate value v.

Detection of LOIs As the first step of the profiling, the LOIs are identified. I collected 150 000 Traces at every position and scanned the die with one probe. Afterwards, I evaluated the leakage test for every position and each share. Figures 4.5a, 4.5b and 4.5c depict the scan of the die surface and the corresponding leakage for each share.

The figures show that all shares are spatially separable and a practical optimal measurement position is chosen for each probe of the multi-probe setup. The chosen positions are denoted in Table 4.1. Note that the origin (0, 0) is located in top left corner in figure 4.5. For power measurements, this step is not necessary due to the position invariant nature of the side channel.

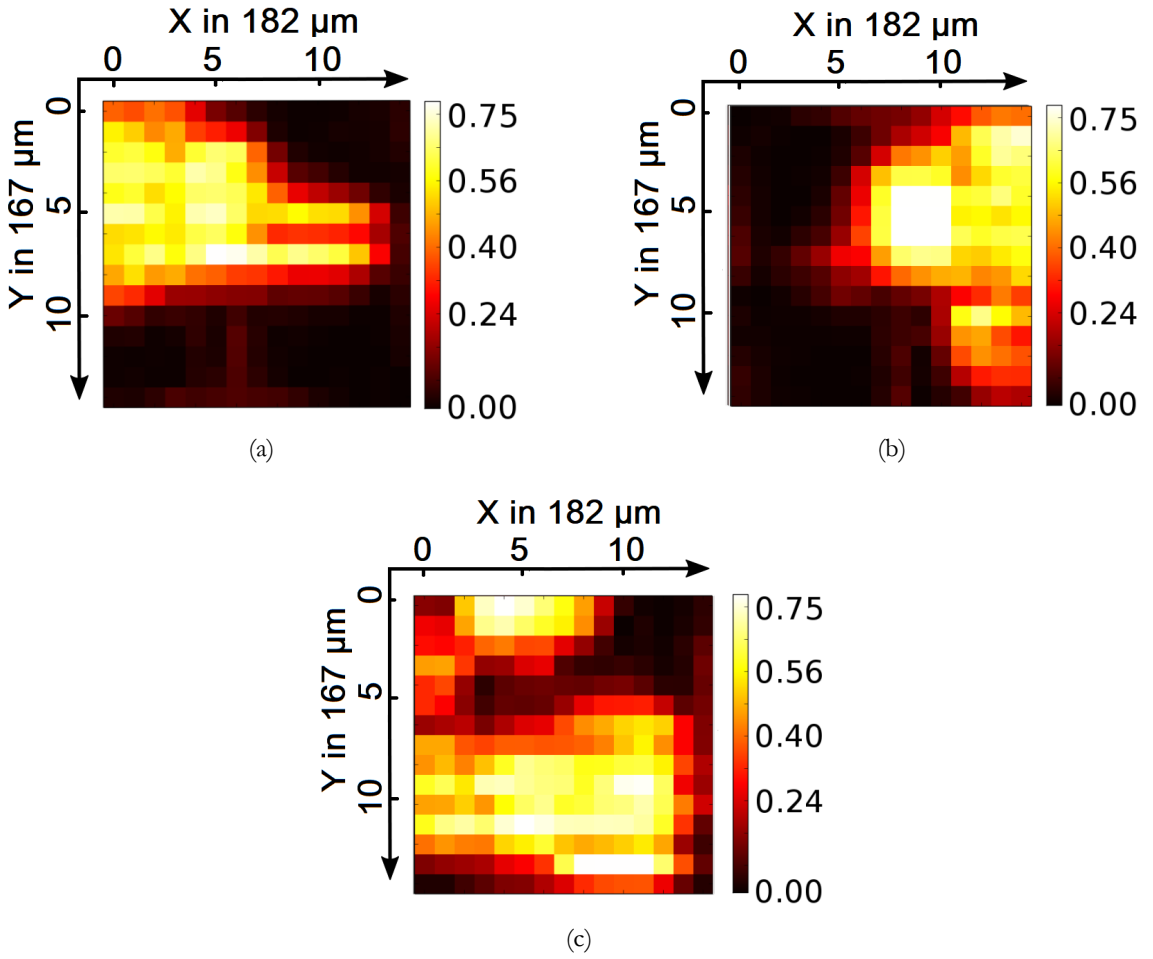


Figure 4.5: Heatmaps of CPOI for all shares. Figure 4.5a Share1. Figure 4.5b Share2. Figure 4.5c Share3.

# **Share**	**X-position in** 182 µm	**Y-position in** 167 µm
1	5	7
2	9	5
3	4	1

Table 4.1: Measurement positions for multiprobe measurements

Detection of POIs For the detection of the POIs I place the three probes at the positions denoted in table 4.1 and collected 150 000 Traces. Since the implementation operates on 3 shares, 3 probes are designated to give the best results. Again, the MCP-DPA based leakage test is evaluated according to the intermediate values of the shares for each probe.
Figure 4.6 shows the leakage for all three shares of the best probe. Some probes did also collect a signal from multiple shares. However, for simplicity reasons I only plot the best probe for each share. In chapter 4.7.3.2 I discuss the results of all probes in detail.

Leakage mainly occurs between sample **1 490 and 1 520** in figure 4.6, which are 30 samples in total. For the power side channel, the leakage test was performed, based on the same trace set for each share. Note that the correlation based leakage test operates on a univariate (one point in time) basis. The below carried out template attack is a multivariate attack; hence, the exploitable leakage of the template attack and the leakage test may differ.

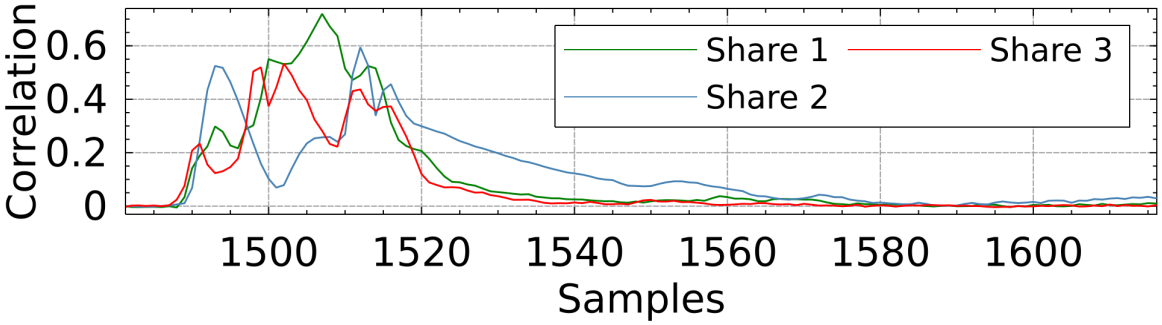


Figure 4.6: CPOI for EM

4.5.2 Profiling steps of Linear Discriminant Analysis (LDA) as Preprocessing and the Template Attack

Firstly, I profile LDA. Afterwards, I transform the traces to a lower dimensional space for the profiling phase of the template attack. This is necessary to improve the template attack in terms of computation time and numerical stability.
LDA and templates are trained for the value of each share (both masks and the masked S-box-input) with 500 k traces (using the same traces for both trainings).

Please note, more profiling traces did not improve the performance of the attack.
For each share the profiling was performed independently by taking the corresponding v of the share and the corresponding sets of traces from the probe above the share. The profiling is performed for the following intermediate values: share 1: Sboxin $\oplus$ mask1 $\oplus$ mask2, share 2: mask1, share 3: mask2

Profiling LDA From the traces $\mathbf{T}$ the chosen POIs $\mathbf{q} = [1490, ..., 1520]$ are used as input for equation 3.1, 3.2, 3.3 and 3.4 to calculate the mean, the overall mean, the within class scatter matrix and the between class scatter matrix. With these means and scatter matrices $\mathbf{W}$ is estimated by solving the eigenvalue problem in equation 3.6.
For power measurements I replaced the different traces of every probe with one power trace.
The number of dimension is reduced from 30 POIs $\mathbf{q}$ to 15 dimensions. This choice of output dimensions is oriented at the 90 % explained variance rule for the (combined) EM measurements explained in chapter 3.2.2.

Profiling the Template Attack The template attack is described in sect. 3.1.2. For the profiling (and attack) phase the traces are transformed by LDA. Hence, the template attack receives 15 dimensions for each trace. Again, the profiling is performed for each share independently by taking the corresponding v. With the help of the 500 000 training traces, a Gaussian template is created for each share and intermediate value, resulting in 256 Gaussian PDFs $\sim (\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$ for each share, which are estimated with equation 3.7 and 3.8. Note that these PDFs are used in equation 3.9 to determine the probabilities of all v for a given trace.
For power measurements I replaced the different traces of every probe with one power trace. These templates are used in the next section during the attack phase.

4.6 The Carried out Template Attack

The input of the template attack is a trace (with unknown key), transformed with the above trained $\mathbf{W}$. Hence, the number of input dimensions is 15 for the attack. The complete attack is depicted in figure 4.7. It is possible to attack

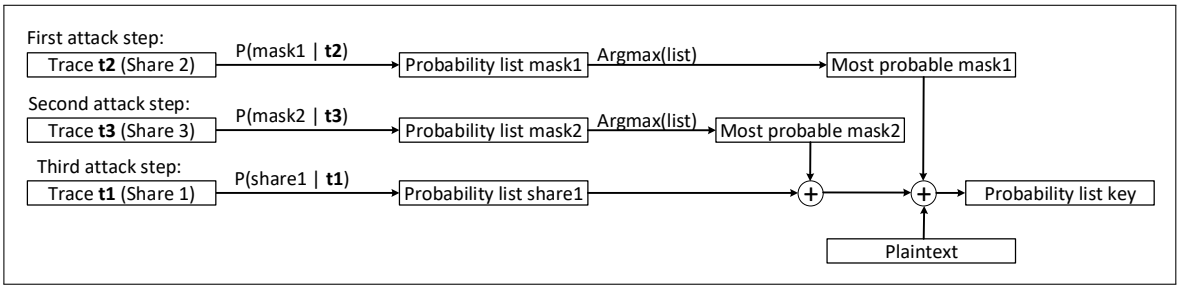


Figure 4.7: Attack principle of multi-probe, localized EM template attacks based on first-order statistical moments.

each share independently, denoted by attack steps one, two and three in figure 4.7, due to the independent profiling for each share before. Hence, I carry out one template attack for each share. Firstly, I use the Gaussian templates of two probes to recognize the mask values mask1 and mask2 and one probe to recognize the masked S-box input Sboxin $\oplus$ mask1 $\oplus$ mask2, by applying equation 3.9.

Each template attack results in a probability list for each intermediate value of each share, denoted by probability list mask1, mask2, and share1 in figure 4.7. Afterwards, the most probable value for each mask is extracted and removed from the masked S-box input (share1) as shown in figure 4.7. The key guesses $\hat{k}$ can be determined by equation 4.3 for all possible v and the used plaintext p.

$$\hat{k} = \text{mostprobablemask1} \oplus \text{mostprobablemask2} \oplus v \oplus p \tag{4.3}$$

The key probability list can be calculated by equation 4.4 for the key guesses $\hat{k}$, where Sc(.) denotes the score of the given element determined by equation 3.9:

$$Sc(\hat{k}) = Sc(\text{mostprobablemask1}) + Sc(\text{mostprobablemask2}) + Sc(v \oplus p) \tag{4.4}$$

Thus, the recognition of the correct masks is an important step, because a wrong mask value leads to the wrong assignment of the key probabilities. Note that I multiply the calculated score from equation 3.9 of both recovered masks and the current key value, to penalize traces with low scores for the mask values.

The described calculations are repeated for each trace and the resulting list of

key-probabilities is accumulated. The position of the correct key in the list is called key rank. The template attack is successful if the correct key is the most probable one, which corresponds to key rank 1. Thus, it is important after how many traces the correct key is at key rank 1, which is called the Measurements To Disclosure (MTD).
For power measurements I replaced trace **t1**, **t2**, and **t3** (Figure 4.7) with one power trace for the attack and only determined the POIs in time, as this measurement method is position invariant.

Mounting the Attack with More than the Most Probable Mask Due to the low recognition rates of the correct mask values, the idea of removing the most probable mask was enhanced by calculating the scores for less probable masks, too. This ideally includes the correct mask values; thus, improving the results, but increasing the computational complexity of the attack. Each tried mask value results in a full list of key probabilities, which has to be accumulated. Thus considering the 10 most probable masks for both masks results in $10 \cdot 10 = 100$ accumulated probabilities for each key candidate for each trace. The evaluation was limited to the best 10 masks due to the quadratic increase in computation time.

4.7 Practical Attack Results

In this section I evaluate the template attack. Firstly, I present the results to the power side channel. Afterwards, I present results to separately and combined evaluated probes.

4.7.1 Power Measurement Results

To ensure correct functionality of the threshold implementation, I carried out the correlation-based leakage test with $p_0 \oplus k_0$ as hypotheses, the (non-masked) S-box input of the first byte. There is no first-order leakage after 2 000 000 traces, which confirms the proper behavior of the implementation.
Performing a higher order attack on a TI protected S-box leads to figure 4.8, which shows the results of the template attack (green line). The key is revealed after 1 025 400 traces (entry "P2" of Table 4.2). This shows the effectiveness of a template attack and proves that despite the simple measurement setup

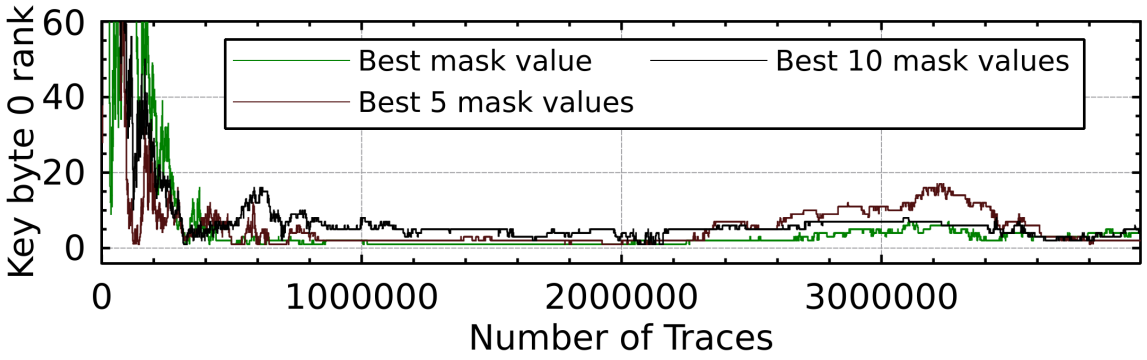


Figure 4.8: Key rank of Byte 0 over number of traces for power template attack.

and a lot of traces, the attack succeeds. As mentioned above, the recongnition of both masks is important; thus, the mask recovery rates strongly influence the Measurements To Disclosure (MTD) and have to be significantly above guessing probability $1/256 = 0.39\,\%$. I successfully recover the masks in 0.97 % for mask1 and 1.03 % for mask2 of cases, see Table 4.3 entry "P2". In the mean the correct masks got rank 97 (mask1) and 95 (mask2) of 256 values. Notable is that even in presence of such low ranks and recovery values, the attack still succeeds. Please note, recovering the mask is done only with one encryption since each encryption uses a different mask.
Considering less probable masks during the attack phase shows no improvement for the power side-channel, as shown in Figure 4.8. In cases of mask ranks of 97 and 95 in the mean, the chance that the correct mask is in the 10 most probable masks is small. Thus, consideration of the 10 most probable mask for the power side channel is not enough.
To complement the analysis, a 3rd-order MCP-DPA was carried out, which succeeds after approximately 600 000 traces (entry "P3" of Table 4.2). Hence, it is favorable for the power side-channel to mount a 3rd-order MCP-DPA [Mor16b] instead of the above described 3rd-order template attack.

4.7.2 Analyzing Probes Separately

Carrying out the attack with the first positioned probe above share 1 failed, even after 2 Mio traces (see entry "S2" of Table I). Therefore, two more probes were added to acquire signals from each individual TI share.

I analyze the probes separately; hence I try to recover share 1 with probe1 only,

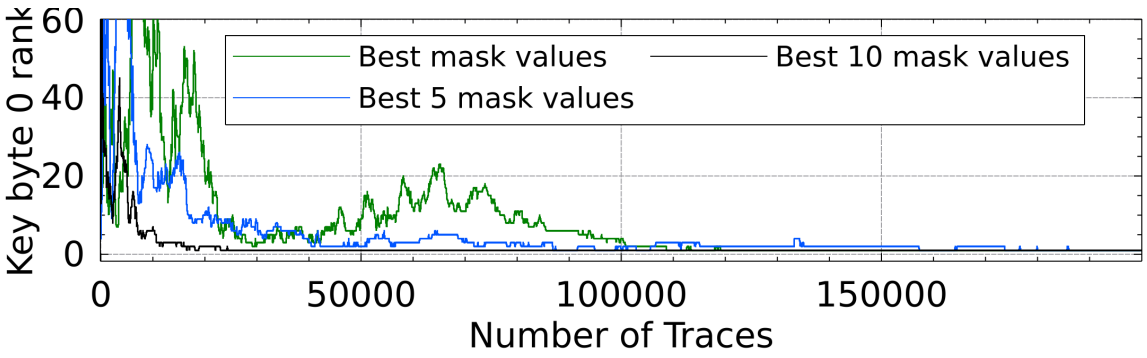


Figure 4.9: Key rank of byte 0 over number of traces with a template attack of separately evaluated EM probes

share 2 with probe 2 only and share 3 with probe 3 only as described in chapter 4.6.

With separately evaluated probes I recover the correct key with 119,300 attack traces (entry "M1" of Table 4.2).

I succeed to recover the masks in 4.2 % (mask1) and 8.3 % (mask2) of cases. In the mean the correct masks got rank 46 (mask1) and 37 (mask2) of 256 possible values. The results show that localized EM is capable of recognizing the correct masks with a much higher success rate, compared to the power side-channel, which leads to a lower MTD.

In the next step I take the 10 most probable masks for each trace into account; thus, I calculate 10^2 key probability lists for each trace. I improve the attack by factor 4.8, compared to the consideration of the most probable mask value only, and reveal the correct key after 24,600 traces (table entry "M2"). Primarily, this is due to the correct masks being among the 10 best candidates in 25 % of the cases for mask 1 and 36 % of the cases for mask 2.

A 3rd-order MCP-DPA succeeds after about 1,300 000 Traces (table entry "S3") with one probe. The use of one probe is usually preferred for the analysis. However, in this case, (unsurprisingly) the pick-up of a single probe is too narrow and the leakage of all 3 shares is not captured using only one probe, which is supported by results from [Imm17].

#	Type	#Best Masks	#Traces
Power Measurement			
P1	unprotected	–	2
P2	TI,LDA	1/10	1,025,400
P3	TI,MCP-DPA*	-	600 000
EM Measurement			
	with 1 probe		
S1	unprotected	–	2
S2	TI,LDA**	–	> 2 M
S3	TI,MCP-DPA*	–	1,300 000
	with 3 probes		
M1	TI,LDA,separate	1	119,300
M2	TI,LDA,separate	10	24,600
M3	TI,LDA,combined	1	18,200
M4	TI,LDA,combined	10	**4,300**
M5	TI,LDA,combined,acc	1	> 1 M

* : 3rd-order profiled MC-DPA [Mor16b]

** : First probe

Table 4.2: Measurement results, TI implementation follows [De 16a].

4.7.3 Combining Multiple Probes

Here I present results for the combination of multiple probes. Firstly I combine probes by concatenation, because it is optimal from an information theoretic point of view. In a second step I combine multiple probes by summing the raw signal of every probe, to evaluate a simple combination strategy with reduced computational effort. After each evaluation, I discuss the results by analyzing the mechanisms of LDA and back these results by leakage tests for the combination of multiple probes.

4.7.3.1 Combining Multiple Probes by Concatenation

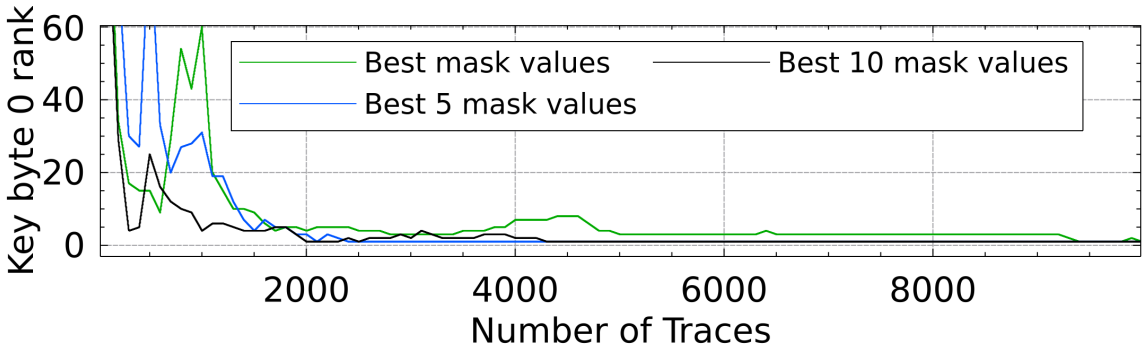


Figure 4.10: Key rank of byte 0 over number of traces with combined EM probe template attack

The combination of multiple probes by concatenation is analyzed in this section, i.e., the trace sets $\mathbf{T}$ of all three probes is combined to one trace set by tripling the length of the original traces. This results in the same trace set for all shares; however with the information of all probes. I use the same 30 POIs, derived above in chapter 4.5.1 for all three trace sets. Leading to $3 * 30 = 90$ input dimensions for LDA. The number of output dimensions of LDA stayed the same at 15 in total, to be comparable to other results. Afterwards the concatenated traces are used for the profiling and attack steps, described in chapter 4.5 and 4.6. The advantage of concatenation is that LDA can treat each sample point of each probe independently; thus, can extract the leakage ideally. The extracted leakage by LDA is the key-element in the attack chain, because the template attack has constant number of 15 input dimensions, independent if the probes are evaluated separately or combined.

#	Type	RR* mask1	RR* mask2
Power Measurement			
P1	unprotected	–	-
P2	TI,LDA	0.97	1.03
EM Measurement			
	with 1 probe		
S1	unprotected	–	-
	with 3 probes		
M1	TI,LDA,separate	4.2	8.3
M3	TI,LDA,combined	10.6	8.6

* : Recovery Rate

Table 4.3: Measurement results, TI implementation follows [De 16a].

The result of combining the signal of multiple probes in table 4.2 shows key rank 1 with 18,200 (table entry "M3") traces, which is factor 6.5 less, than for the non-combined case with 119,300 traces (table entry "M1"). Figure 4.10 depicts that the key can be revealed with a total of 4,300 Traces (table entry "M4") when considering the 10 most probable masks, which is factor 5.7 less traces, compared to the independent evaluation of each probe (table entry "M2"). Again, considering more than just the most probable mask leads to significant improvements.

To emphasize the power of the combined probes, I compare the ranks and recovery rates to the separate evaluation. The main reason for the lower MTD is that I succeed to recover the masks in 8.6% (mask1) and 10.6 % (mask2) of cases (entry "M3" of table 4.3), instead of 8.3 % (mask1) and 4.2 % (mask2) for the individual evaluation (entry "M1" of table 4.3). The correct masks are within the most 10 candidates in 44 % of cases for mask 1 and 37 % of cases for mask 2. In the mean the correct masks got rank 28 (mask1) and 36 (mask2) of 256 possible values.

These results highlight the power of combining the information of multiple probes.

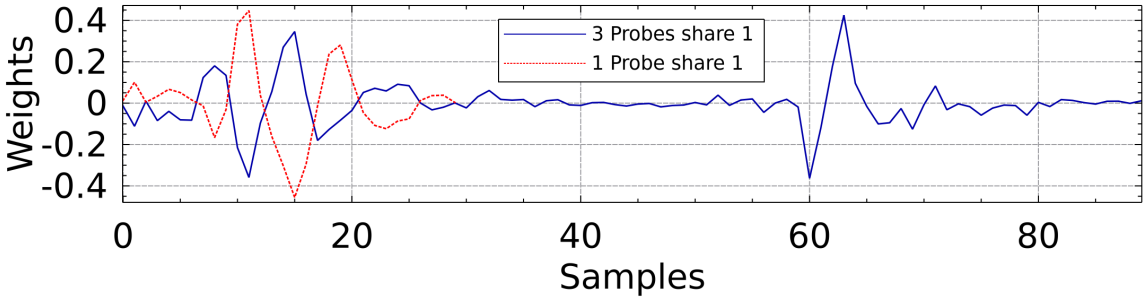


Figure 4.11: LDA weights for combined and single evaluated probes for share 1

4.7.3.2 Illustration of the Combination of Multiple Probes by Concatenation

In the following, I analyze the specific cause for the improvement when combining the information of multiple probes (table entry "M4"), compared to the separate evaluation (table entry "M2"). I firstly show, which probe contributes to the exploited leakage for each share in the combined case. Hence, I analyze for each share the contributing LDA weights of each probe. A high weight corresponds to high leakage. Secondly I want to compare the LDA between the separated evaluated case and the combined evaluation. Please note that the number of input dimensions and hence, the number of dimensions of the weight vectors $\mathbf{w}$, for LDA differ for the separated (30 points) and combined case (90 points). To back the results I compare the LDA weights to results of the correlation based leakage test for every probe. Note that I depict the weights for one LDA-dimension only, because the mentioned effects are observable for other dimensions as well.

Analyzing LDA Weights Figure 4.11 depicts the LDA weights over all 3 probes of share 1 in the combined case and over 1 probe in the separated case. To show the measured leakage of each probe for this share, I firstly analyze the LDA weights for the combined (blue) case. Hence, sample 0 to 29 correspond to probe 1, sample 30 to 59 to probe 2 and sample 60 to 89 to probe 3. For the first probe (samples 0 to 29) there is significant leakage exploitable.
For the second probe (samples 30 to 59), there is no clearly exploitable leakage, probably due to a spatial location of the probe, where no leakage is observable. However, for the third probe there is a similar amount of leakage as for the

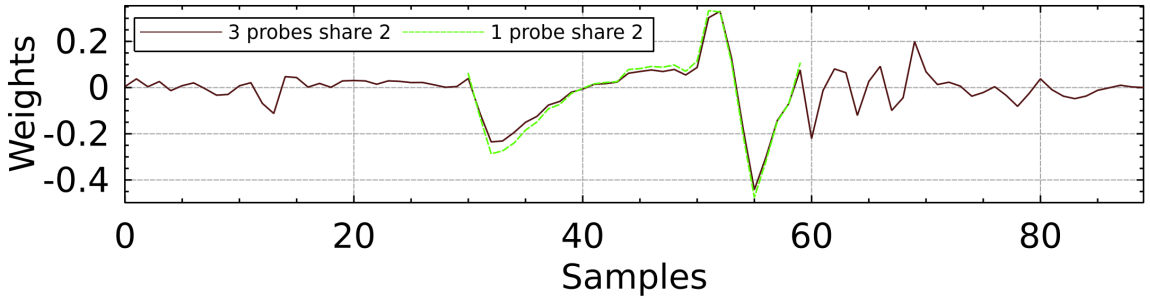


Figure 4.12: LDA weights for combined and single evaluated probes for share 2

first probe. Thus, mainly probe 1 and 3 contribute to the recovery of share 1 in the combined case. Comparing the combined evaluation of 3 probes (blue) with the separate evaluation of probe 1 (red) shows that the shape and the most leaking points are similar (mirrored at the x-axis).

Figure 4.12 depicts the weights for one output dimension of share 2. Beside probe 2, probe 1 and 3 contribute to the measured leakage, however with much lower weights. Thus, all 3 probes contribute to the recovery of share 2 in the combined case. This is backed by comparing the entries "M1" to "M3" from Table 4.3. The combination of multiple probes significantly increases the recovery rate, e.g., from 4.2 % to 10.6 % for share 2, in case that multiple probe measure leakage. Again, comparing the combined evaluation of 3 probes (brown) with the separate evaluation of probe 1 (green) shows that the shape and the most leaking points are similar.

Figure 4.13 depicts the weights for one output dimension of share 3. Again, the shape and leaking points from LDA with one probe are similar to the one with three probes. Probe 1 contributes marginally and probe 2 does not contribute to the measured leakage. In this case mainly probe 3 contributes to the recovery of share 3 in the combined case. This is backed by comparing the entries "M1" to "M3" from Table 4.3. The recovery rate for share 3 is constant in case that mainly one probe measures leakage, e.g., 8.3 % and 8.6 %. Again, comparing the combined evaluation of 3 probes (purple) with the separate evaluation of probe 1 (blue) shows that the shape and the most leaking points are similar.

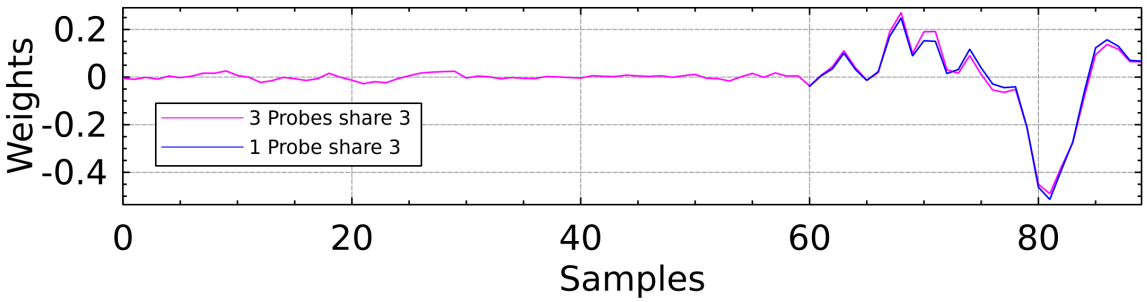


Figure 4.13: LDA weights for combined and single evaluated probes for share 3

The above results show that combining multiple probes leads to a massive improvement of measurement results (if multiple probes measure leakage), without significantly increasing the computational complexity of the attack. I attribute the improvement to the following two reasons: Firstly, the leakage of multiple probes is accumulated, which results in more exploitable leakage. Secondly, the effects caused e.g., by electrical noise, thermal noise, etc. can be easier eliminated by measuring the S-box-input with multiple independent observations.

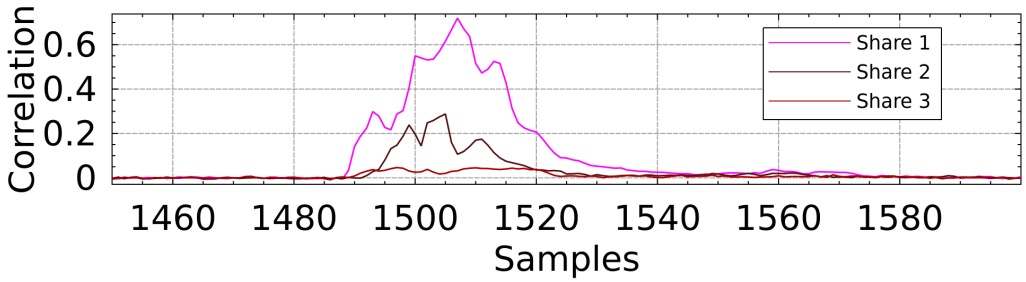


Figure 4.14: Correlation based leakage test for all shares and probe 1

Analyzing Correlation Based Leakage Test Results To backup the results from the previous section, I carried out the correlation based leakage test for each probe and each share. In theory the LDA weights and the leakage test should show leakage for the same points in time and shares for each probe. Figures 4.14, 4.15 and 4.16 show the leakage of every probe for each share. As input for LDA, samples 1490 to 1520 are taken from the figures, which covers

most of the leaking points.

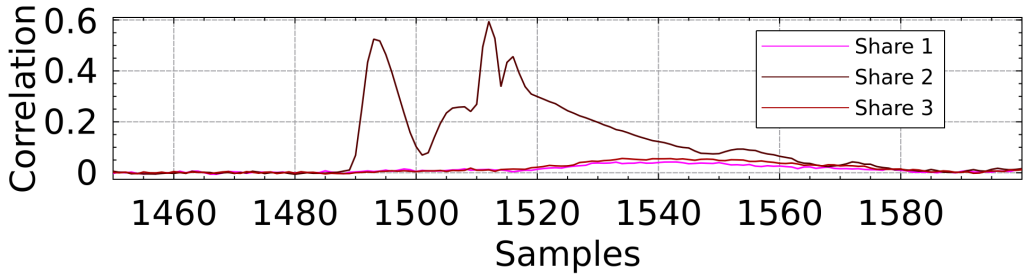


Figure 4.15: Correlation based leakage test for all shares and probe 2

Figure 4.14 and probe 1 show leakage for share 1 and share 2. This corresponds to the observed weights of Figure 4.11 and 4.12 in samples 0 to 29. The lower correlation for share 2 results in a lower weight factor.
Probe 2 depicted in Figure 4.15 mainly shows leakage for share 2, which is supported by the high amplitude in sample 30 to 59 in Figure 4.12 and close-to 0 weights in Figures 4.11 and 4.13.
Leakage for probe 3 is depicted in Figure 4.16 and shows significant leakage for all shares. This results in weights higher than zero in Figures 4.11, 4.12 and 4.13 in sample 60 to 89. Similar to the behavior of probe 1, lower leakage results in lower weights.

Comparing the weights determined by LDA to the detected leakage by the correlation based leakage test shows that time and amplitude of the leakage match each other.

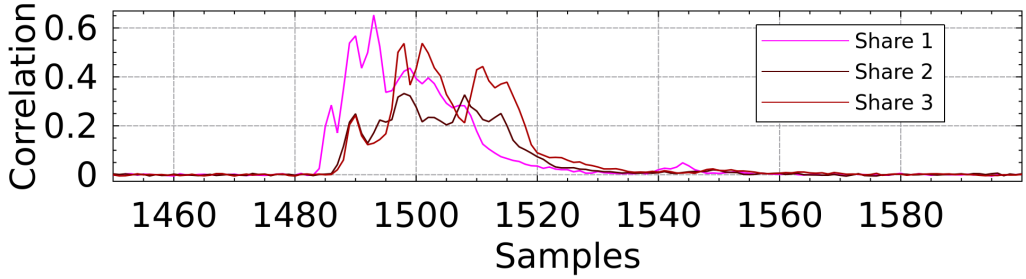


Figure 4.16: Correlation based leakage test for all shares and probe 3

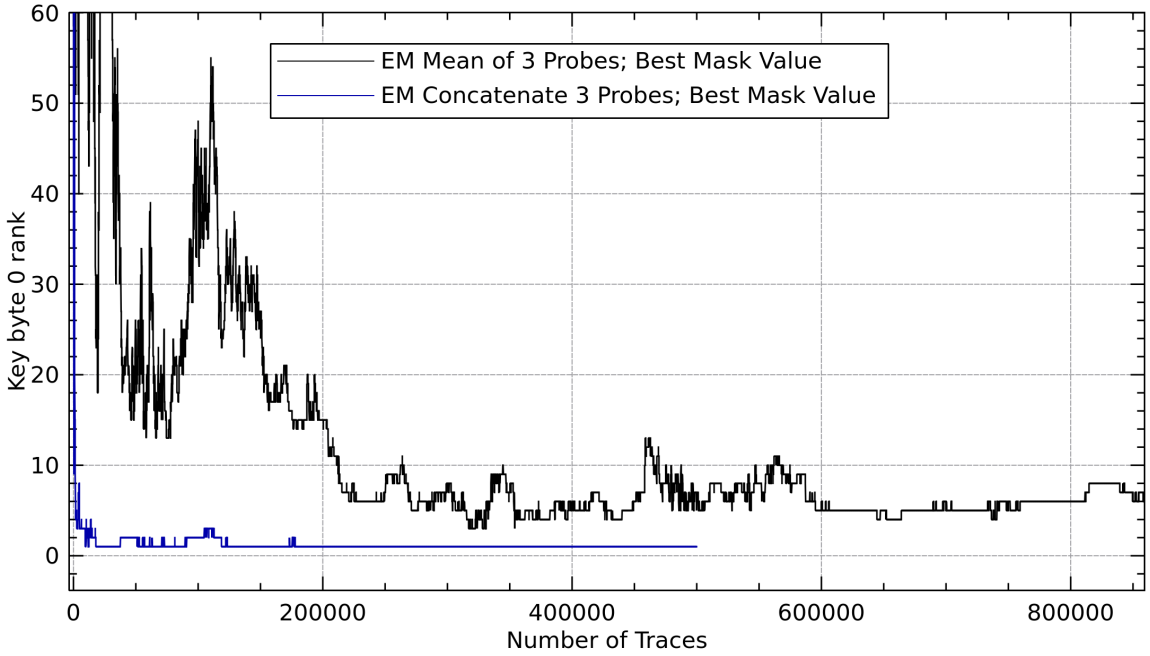


Figure 4.17: Key rank of byte 0 over number of traces with combined EM probe template attack by summation

In this section, I examined the combination of multiple probes by the concatenation and showed that thereby combined measurements lead to more exploitable leakage. In the following I will investigate the combination of multiple probes by summing the signals of every single probe.

4.7.3.3 Combining Multiple Probes by Summation

The summation of raw signals is a simple and "low effort" method to combine multiple probes. Thus, I evaluate in the following this combination technique The sum of the three probe raw signals is calculated, resulting in one common trace set for all shares (similar to power measurements). Thus, the number of input dimensions stays constant at 30, compared to the separately evaluated case. I present the results for this kind of combination and explain the results with the help of analyzing LDA weights and correlation based leakage test results. Due to the summation, one trace results in 30 points in time for 3 probes. To be comparable to the other evaluation methods, I chose the number of LDA-output dimensions equal 15. For the summed case, the attack does not succeed after 1 Mio. traces.
To achieve a better understanding of the mechanisms of the combination by

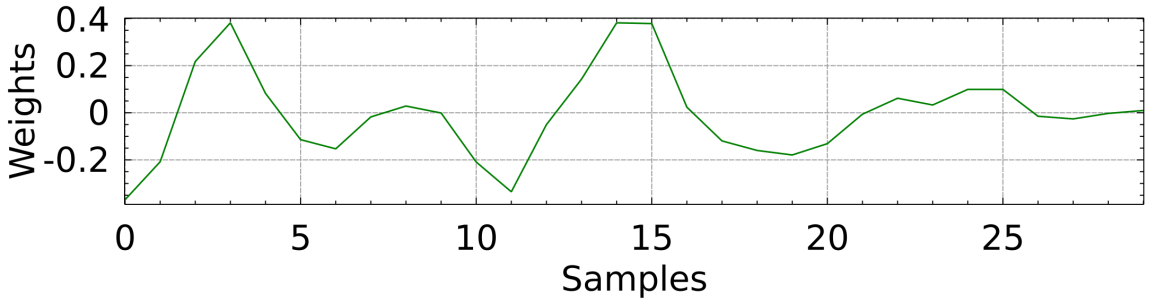


Figure 4.18: LDA weights for accumulated probes for share 1

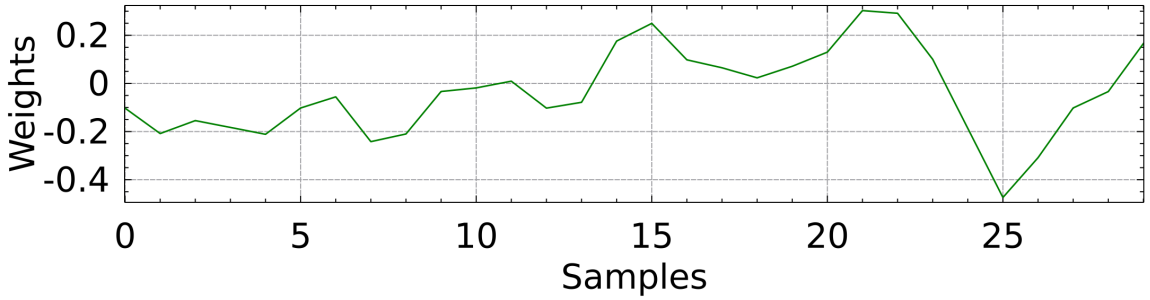


Figure 4.19: LDA weights for accumulated probes for share 2

summation multiple probes I firstly analyze the LDA weight factors for every share.

Illustration of the Combination of Multiple Probes by Summation Summing the signals decreases the performance of the attack significantly. Figure 4.18, 4.19 and 4.20 show the LDA weights for the summed combination of share 1, 2 and 3. The leakage for share 1 and 3 are mainly extracted from the first 20 samples. This shows that LDA extracts the leakage for 2 different shares out of the same points in time/dimensions, based on the same trace, which can lead to an increased algorithmic noise between shares. Note that the amplitudes of different combination strategies cannot be easily compared, due to the normalization during LDA computation, which is explained in chapter 3.1.1.

Analyzing Correlation Based Leakage Test Results To back the results from the analysis of the LDA weights, I carried out the correlation based leakage test for each share. In the accumulated case this consists of only one plot,

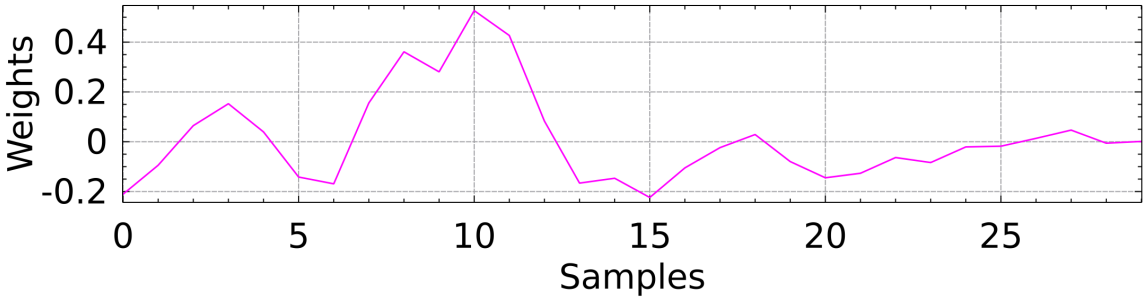


Figure 4.20: LDA weights for accumulated probes for share 3

due to the summed probe signals. Figure 4.21 shows a maximum correlation of 0.56 for share 1, 0.48 for share 2 and 0.32 for share 3. Compared to the separate evaluation, where the correlation reaches 0.72 for share1, 0.59 for share2 and 0.53 for share3, this is significantly less. The reduced correlation can be caused, e.g., by a higher (algorithmic) noise level.

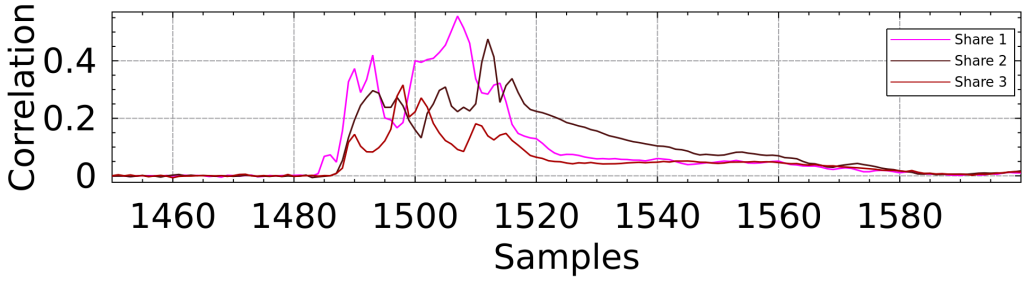


Figure 4.21: Correlation based leakage test for accumulated probes for all shares

4.7.4 Coupling between Multiple Probes

Due to the close distance between all three probes coupling is a threat. Coupling between probes would make an independent measurement of the three shares impossible, which would result in a decreased attack efficiency. Therefore, coupling is relevant in case that a significant amount of leakage is caused. Analyzing figures 4.14 to 4.16 shows that all probes show different leakage characteristics for all shares. Thus, the coupling between the three probes is neglectable in this case.

4.7.5 Comparing Localized EM and Power Measurements

In this section I compare the results of power and multiprobe localized EM measurements, which are both using the identical FPGA configuration (bit file). For the evaluation, I used 500k profiling-traces for both setups and the same number of input and output LDA-dimensions to make the results comparable. To rule out a performance gain due to the different measurement approaches, I perform a standard template attack for localized EM and power against an unprotected Canright S-box. The attack shows that the MTD is similar for both measurements. Substantial differences can therefore only be caused by the individual measurement approaches. With the help of localized EM measurements the attack is stronger, compared to power measurements, because each of the three probes has a focused view on one share. In contrast the power side-channel is measuring all shares at the same time and has to recover the values of all shares out of the same trace. On a conceptual level this would mean at least a higher level of algorithmic noise and a lower SNR.

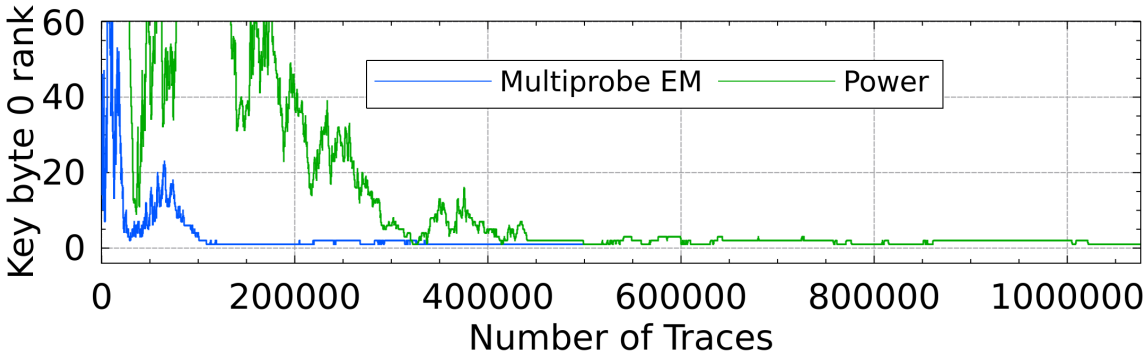


Figure 4.22: Key rank of byte 0 over number of traces for power and EM (best masks for both).

The measurements show that I am able to reduce the MTD by a factor of $1,025,400/4,300 \approx 238$, comparing combined multi-probe localized EM attacks (table entry "M4") with the power side channel (table entry "P2"). The lower security gain is probably caused by higher ranks of the correct masks, when compared to the power side channel. This strongly supports the argument that in case of considering the ten most probable masks, multi-probe localized EM measurements can further reduce the amount of required traces. Whereas power measurements do not benefit in the same way when consider-

ing the ten most probable masks. Power measurements would probably require to consider much more less probable masks than ten, unfortunately this was not possible due to a quadratic increase in runtime.
Nonetheless, the power measurements show that an attack is still possible with much lower ranks of the correct masks. However, multi-probe localized EM can significantly reduce the security gain; thus leading to a more powerful attack.

4.8 Conclusion

In this chapter I showed a successful multiprobe attack on a second-order secure TI. The measurements are based on simultaneously acquired traces using three, high-resolution EM probes. I show that the shares can be spatially separated by localized EM measurements on an FPGA implementation. Thereby, undermining a central security assumption in the original paper. As a result, more powerful attacks can be carried out by placing one probe above each share. Thus, one probe can focus one share, leading to a significantly decreased MTD, compared to the power side channel, which enables a global view on all shares. However, beside the most significant shares, also other shares are observed by most probes. Thus, combining the information of all probes for all shares by concatenation further decreases the MTD by factor 5.7, compared to the individual evaluation. In total, the attack requires only 4, 300 traces to break the scheme in the combined case.
As a central element I use LDA to combine the leakage of multiple probes. Hence, I firstly describe the impact of each probe to the exploited leakage by analyzing the LDA weights. I show that in our case, LDA combines the available leakage of multiple probes for each share. The improvement due to the combination of probes can be observed for the mask recovery rates (share 2 and 3), comparing the separated evaluation case and the combined. The LDA weights show leakage for all 3 probes for share 2, which results in a mask recovery rate of 10.6 % in the combined case instead of 4.2 % in the separate evaluation. In contrast share 3 is mainly observed by one probe. Thus, the combination does not significantly increase the mask recovery rate.
Additionally, I show that considering less probable masks can improve the attack. In my case, the attack reduced the MTD by factor 4.2 when considering the 10 most probable masks in the combined (by concatenation) case. To complement the analysis, I carry out a 3rd-order MCP-DPA [Mor16b]. This

attack succeeds for the power side channel after 600 000 traces and for the EM side channel for 1,300 000 traces, which shows that it is the better choice for the power side channel; however, multiprobe EM attacks still outperform the power side channel.

Chapter 5

Single Trace Multiprobe Side Channel Attacks Against an Asymmetric Cipher

In the first chapters I motivated and briefly explained the background of multiprobe measurements. In the last chapter, I showed the weakening of a security assumption for a masking scheme by using multiprobe measurements. In this chapter, I carry out a single trace multiprobe attack against an exponentiation of asymmetric ciphers as a second experiment with multiprobe side channel attacks.
To efficiently combine the leakage of multiple probes, it is of central importance to maximize the exploited leakage for each individual probe. Firstly, I improve the algorithmic approach of Heyszl et al. [Hey14] by using PCA for pre-processing and the expectation maximization algorithm for one probe. Afterwards, I combine the measurements of 3 probes and compare the results for an unprofiled and profiled attack to the single probe approach.

In section 5.1 I firstly introduce the improved algorithmic approach for one probe. I back the suggested algorithmic improvements by practical experiments. I introduce the measurement setup in section 5.2 and present practical results for the algorithmic improvement of one probe in section 5.3. Afterwards, I combine multiple probes by concatenation and summation for an unprofiled and profiled attack scenario in section 5.4. Later I discuss the use of PCA, the different ways of combining multiple probes and the coupling between the probes in section 5.5, 5.6 and 5.7. I summarize the contribution and findings in section 5.8.
The background and related work of single trace non-profiled and profiled attacks against exponentiations is described in section 2.2. The attack concept and the base of the chosen approach developed by Heyszl et al. [Hey14] is ex-

plained in section 3.2.1. Parts of this chapter have been published on COSADE conference in 2015 in the following publication: [Spe15] The presented results are originated in the collaboration with the coauthors.

5.1 Improved Unprofiled Attack Against Exponentiations

In this section, I describe the algorithmic approach to clustering-based *non-profiled* attacks on exponentiations, which improves previous work [Hey14].

5.1.1 PCA for Dimensionality Reduction and Feature Selection

Side-channel measurements usually lead to big amounts of data, especially when high sampling rates for magnetic field measurements are required. This increases required computational power and memory consumption during subsequent data analysis. Only a small part of the data will contain exploitable leakage information. Hence, *feature selection* to discard non exploitable trace parts is desirable.
Simple trace compression [Man07] is commonly used and usually justified by electrical properties. However, simple trace preprocessing techniques have been shown to have negative effects on results [Hey12b]. Thus, more advanced trace compression techniques are desirable. Hence, PCA is used before clustering.
Standaert and Archambeau [Sta08] compare PCA and LDA in the context of template attacks for trace compression and confirm that LDA leads to superior results. I disregard LDA in the following, because the requirement of training data. Furthermore, PCA has been applied to side-channel analysis for data reduction in several contributions [Boh03, Arc06, Sta08, Bat12, Mav12] for different attacks of which Archambeau et al. [Arc06] were the first to describe the use of PCA in the context of template attacks. I concentrate on non-profiled, unsupervised methods, specifically, on PCA in this section.
As described in Sect. 3.2.1, the recorded side-channel trace $\mathbf{t}$ is cut into trace-segments corresponding to n exponent bits (see chapter 3.2.1). This leads to the real matrix $\mathbf{T}$ of measurement data, with the shape $n \times \gamma$ for every probe

(see chapter 3.2.1). Before applying PCA, I removed the mean of every trace-segment as a standard measure. Then I apply PCA to $\mathbf{T}$, which results in the matrix of principle components $\mathbf{T}_{\mathrm{PCA}}$ (see chapter 3.2.2). In these experiments, I normalize the variances of the principal components to one, i.e., I directly use $\mathbf{T}_{\mathrm{PCA}} = \mathrm{U}$ instead of $\mathrm{U} * \Sigma$ (see chapter 3.2.2). This measure results in a clustering independent of the original variance, which improves our results.
Ideally, a transformation into a reduced subspace should maintain the 'useful" information (key-dependent) while neglecting 'not useful' (not key-dependent) information, which is difficult without supervision, because the leakage characteristic for the device is unknown. PCA combines correlating input dimensions into single principal components. Thus, I assume that leakage is located in one or a few principal components; hence, following this idea, the other components contain only non-leaking information (noise). Thus, to reduce the noise, principal components have to be selected. [1] Archambeau et al. [Arc06] propose to only retain the first-ranked components assuming that the leakage is contained there, while discarding the remaining low-variance ones, assuming only noise is contained. This is only true in case the key-dependent leakage contains the biggest variance, e.g., after calculating the mean of multiple traces with the same secret. Batina et al. [Bat12] found in their practical experiments, that results of correlation-based DPA improved when removing first-ranked components. There are several reasons for high variances (which are located in the first-ranked components) of the trace segments, e.g., data-dependent signal influences, the clock signal and noise, which are irrelevant to the desired classification. I suspect that relevant and irrelevant signal parts will aggregate within separate components. Also, from practical experience, the 'interesting' leakage signal parts are rather low-variance in the case of single-execution attacks.
Hence, I propose a selection strategy which discards several highest-ranked as well as many, low-ranked components, because they either contain noise or information which is not useful. I either select *single* principal components or a number of *consecutive* components. In my opinion, this strategy should nonetheless apply to many attacks against implementations of exponentiations. Reduced trace-segments $\mathbf{T}_{\mathrm{PCA,pc:pc+i}} = (\mathbf{PC_{pc}}, ..., \mathbf{PC_{pc+i}})$ are derived, where the principle components are located in the columns of the matrix, with pc the first selected component and $\mathrm{i} \geq 1$ the number of consecutive components retained. The values of $\mathrm{pc} \in [1, 20]$ and $\mathrm{i} \in \{1, 2, 4, 6, 9\}$ were tested

[1] The effect of neglecting unwanted signal parts, by principal component selection, is illustrated in chapter 5.5

during practical experiments. This selection strategy reflects the approach of an attacker who is unable to perform profiling. An optimal selection of components can certainly *not* be determined *a priori* because it is highly device- and application-specific (general issue in machine learning [Wol97]). Hence, without a priori-knowledge, an attacker has to trial different values for pc and i.
The reduced trace segments $\mathbf{T}_{\mathrm{PCA,pc:pc+i}}$ are reduced in number of dimensions, compared to the original trace $\mathbf{T}$. The subsequent task is to extract information about the key out of $\mathbf{T}_{\mathrm{PCA,pc:pc+i}}$, which corresponds to a labeling of the data. While Heyszl et al. use k-means clustering, I improve this by using the expectation maximization algorithm while keeping the Gaussian distribution assumption, which both algorithms are based on. The k-means and expectation maximization algorithm, introduced in chapter 3.2.3 and 3.2.3.2. To evaluate the result, the BFC is calculated, which is explained in chapter 3.2.4. The BFC roughly corresponds to the number of bits an attacker has to brute-force in case the key is brute-forced according to the probabilities determined by the silhouette index in equ. 3.18.

5.2 Measurement Setups

5.2.1 High Resolution EM Measurement Setup

For measurements in this chapter there are the following modifications compared to the setup in chapter 4:

- We use an area of 1700 µm × 1700 µm on the surface of the die
- We arrange three probes in a fixed formation (see figure 5.1), and place them on 400 (20 × 20) different positions within this area to able to evaluate 400 data sets by our analysis.
- The distance of the probes to the die surface is approximately 100 µm.
- We used three probes with coil diameters of 250 µm, 150 µm and 100 µm
- The signal is sampled synchronously to the device's clock at 2.5 GS/s.
- Instead of a 10 Ω resistor a 1 Ω resistor is used.

Figure 5.1 depicts the geometric arrangement of the probes from the side and from the top.

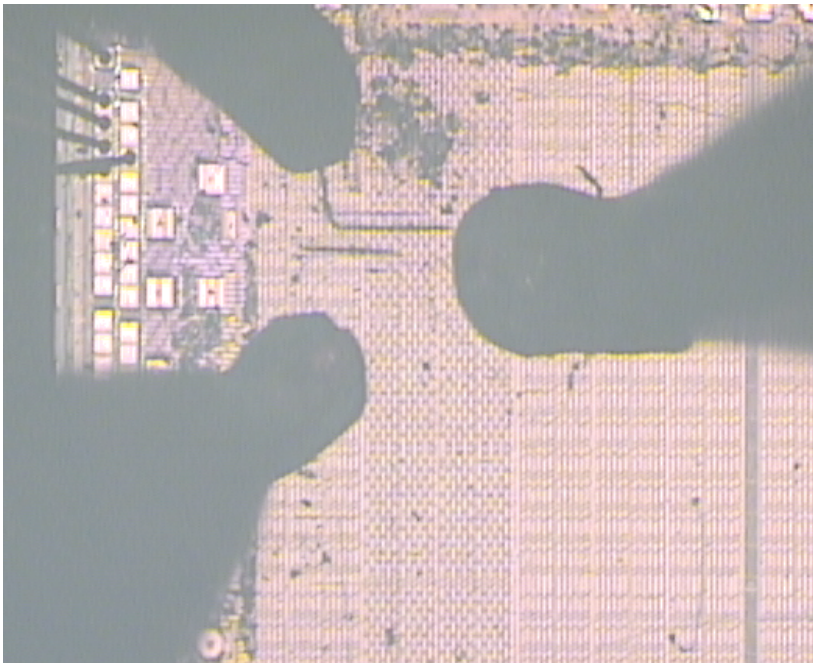


Figure 5.1: Geometric arrangement of measurement-probes on FPGA die surface

5.2.2 Device-under-Test

As a device under test, I use a decapsulated Xilinx Spartan 3A FPGA chip which is configured with an ECC design and performs an 163 bit elliptic curve scalar multiplication using a Montgomery ladder. This algorithm is a classical candidate for attacks against exponentiation algorithms since it processes the secret exponent bit-wise in n constant time segments. Furthermore the Montgomery ladder is performing the same operations in every iteration, independent of the key value, which minimizes the leakage. The key dependent part of the algorithm is the access to different registers, similar to the algorithm 1, described in chapter 3.2.1.

5.3 Separate Evaluation of Probes

In this section I present practical results for the improved approach of Heyszl et al. [Hey14]. I firstly investigate the behavior of differing PCA components and window sizes. Afterwards, I show practical results for the clustering based and profiled attack for one probe and compare the unprofiled attack to the approach

of Heyszl et al. [Hey14]. For the practical evaluation I collect measurements at 400 measurement positions (with all 3 probes concurrently) to gain conclusive insights from a high number of tests.

5.3.1 Quality of Principal Components

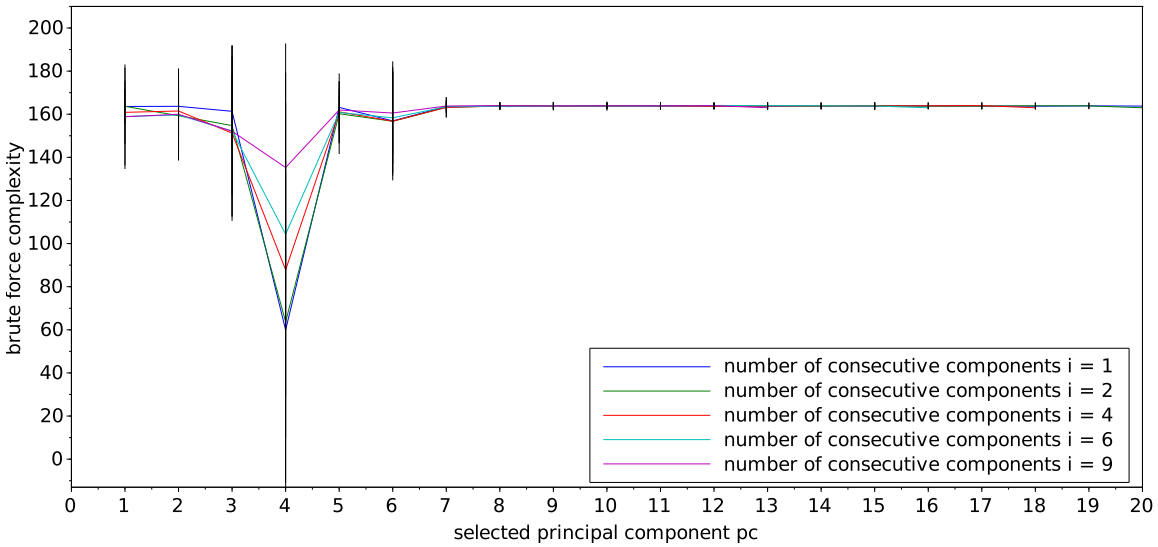


Figure 5.2: Mean brute-force complexity for different selected principal components (pc and i) over all measurement positions *including standard deviation as bars* of the unprofiled analysis

The algorithmic approach of chapter 5.1 includes the selection of principal components after PCA as a first step before clustering. The selection can be described by two parameters, pc the first selected component, and, $i \geq 1$ the number of consecutively selected components after the pc-th one as described in chapter 5.1.1. In this section I investigate the quality of different parameter choices. I executed the clustering-based attack on *every* single measurement from all 3 probes and 400 positions with choices of $pc \in [1, 20]$ and $i \in \{1, 2, 4, 6, 9\}$ and assess the quality using the remaining brute-force complexity explained in chapter 3.2.4.

I show the means over 400 results (measurement positions) for the resulting brute-force complexities for each combination of parameters pc and i in fig-

ure 5.2 for the second probe. I am able to show some fundamental properties of the measurements. High *mean* brute-force complexities of > 100 bits are certainly not within the range of realistic computing capabilities. They result from including many low-scoring results. The standard deviations are shown as vertical bars and indicate that there are multiple results with significantly lower brute-force complexities (the diagram does not include $+1$ bits for assigning labels to classes). As an important observation, low-ranked components ($pc < 10$) seem preferable overall and *first-ranked principal components do not contain exploitable leakage* (see curve with $i = 1$ or $i = 2$ in figure 5.2). This confirms the assumptions from chapter 5.1.1 as well as similar observations from Batina et al. [Bat12]. Thus, I *discard first-ranked as well as low-ranked principal components* before further analysis and achieve significantly improved brute-force complexities.
This is especially important for distribution-based clustering algorithms, because high-variance noise will lead such algorithms in wrong directions. Note that even if 'good quality' components are included in addition to higher-ranked 'non quality' ones (e.g. figure 5.2, curve for $pc = 1$ and $i \in \{4, 6, 9\}$), the results are unsatisfactory (brute-force complexity > 100 bits).
The component number $pc = 4$ seems to contain the most leakage on average, reaching the lowest mean brute-force complexities. It seems that PCA *concentrates most of the exploitable leakage information into a single principal component*. This means that a choice of $i = 1$ for the number of selected consecutive principal components led to the best results in circumstances for the joint analysis of 3 probes and the second probe. [2] I used this choice in the practical evaluation in the next Sect. 5.3.2. As another observation, curves with $i > 2$ lead to low complexities as soon as component 4 is included in the consecutively selected components. For illustrative purposes, I show the resulting principal components after PCA transformation of an example trace in chapter 5.5.

Profiled Evaluation I compare the improved *non-profiled* attack with a *profiled* template attack, for details see chapter 3.2.5, to compare their performance. Figure 5.3 shows the mean for different window sizes and the vertical bars show the standard deviation. When carrying out the template attack with $i = 1$, the BFC shows a value lower than 100 only for $pc = 4$. Most of the (usable) key-

[2] Probe 1 and 3 show a different optimal choice of i. However the mean for both probes is still above 120 bits and also results from Figure 5.4d suggest that probe 2 mainly contributes to the measured leakage, thus i should be chosen in a way that it is favorable for the best probe.

dependent information is located in pc = 4 (and marginally the 2 succeeding principal components). This corresponds to the observation of the unprofiled attack. In contrast to the unprofiled case, the profiled case can profit from a bigger window size, where window size i = 9 shows the best results with a BFC of 50 bits in the mean (see entry 'SP2' in Table 5.1, page 96), compared to the unprofiled case for i = 1 of 60 bits (see entry 'SU2' in Table 5.1, page 96). Due to profiling, the profiled attack can use (and combine) information in different principal components. In contrast the unprofiled attack is not capable of using and combining the information in different principal components. This demonstrates that the profiled attack (like expected) outperforms the non-profiled attack and that profiled attacks can profit from the information contained in dimensions with a BFC-mean of 100 and higher.

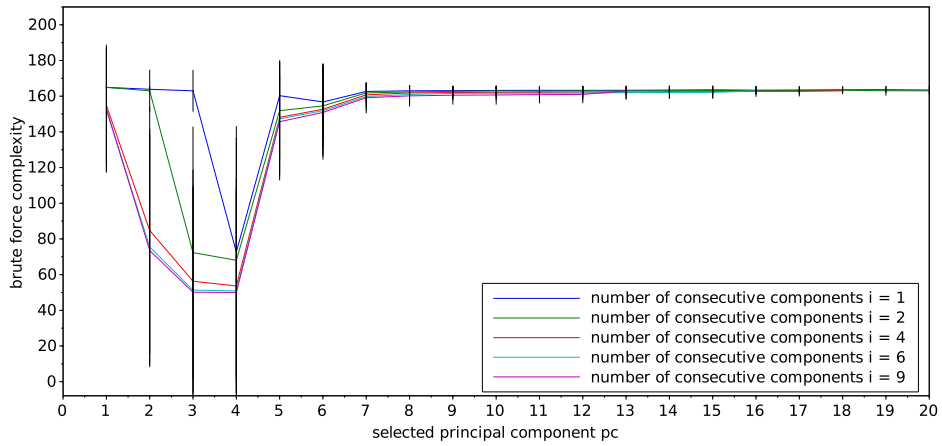


Figure 5.3: Mean brute-force complexity for different selected principal components (pc and i) over all measurement positions *including standard deviation as bars* for the profiled evaluation of the profiled analysis

5.3.2 Analyzing Probes Separately With an Unprofiled and Profiled Attack

For every probe, there are 400 measurements from different positions. I analyze the data from the three available channels separately: Firstly I perform pre-processing by applying PCA, secondly I perform clustering using the expectation maximization algorithm and thirdly I compute the remaining brute-

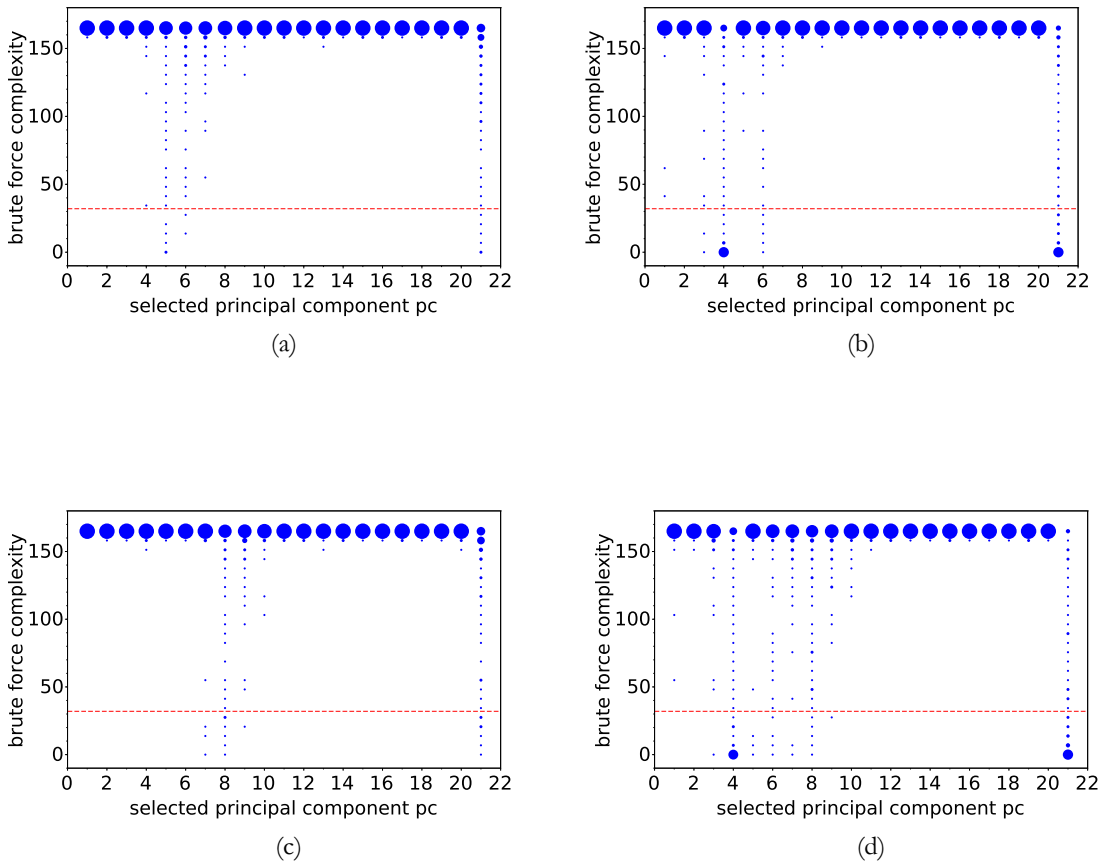


Figure 5.4: Brute force complexity occurrences over different principal components
Single probe 1 (250 µm Ø): Fig. 5.4a
Single probe 2 (150 µm Ø): Fig. 5.4b
Single probe 3 (100 µm Ø): Fig. 5.4c
Combined probes: Fig. 5.4d

force complexity. For every probe separately, and for every selection of principal components (for every pc $\in [1, 20]$ while $i = 1$), I summarize the results from 400 tests in figures 5.4a, 5.4b, and 5.4c. Figure 5.4a shows results for the 250 µm probe, figure 5.4b for the 150 µm probe and figure 5.4c for the 100 µm probe. The figures show, *how many of the* 400 *measurements of each probe*, and for every selection of pc, lead to which brute-force complexities. Note that the results, plotted for pc $= 21$ consists out of the best component pc for every position, which becomes relevant in the next section. The occurrence rate is

visually indicated by the size of the respective dots. Bigger dots mean that the corresponding brute-force complexity has occurred more often. For example, in figure 5.4a, almost all of the 400 measurements lead to a maximum brute-force complexity of 163 for $pc < 5$ and $pc > 10$. For $pc = 5$, however, many measurements lead to lower resulting brute-force complexities, some even of the minimum. The red dashed line highlights the 32 bit complexity level up to which all outcomes are easily manageable for attackers through computation.
As an important finding, it can be observed, that the 150 µm probe depicted in Fig. 5.4b leads to the best results by far. For the principal component $pc = 4$, an astonishing percentage of 59 % out of the 400 measurements led to a remaining brute-force complexity $\leq$ 32 bit (summing up all outcomes equal or lower the red dashed line). This high number was unexpected and means that *with the improved algorithmic approach, more than half of all measurement positions exhibited sufficient leakage* for a break. The 100 µm probe depicted in Fig. 5.4a leads to only 6 % $\leq$ 32 bit for $pc = 5$ and the 250 µm probe depicted in Fig. 5.4c only leads to 5 % $\leq$ 32 bit for $pc = 8$. Hence, the 150 µm probe seems to work best for this measurement setup and DUT. Please note that we adjusted the single probes by hand, hence probes 1 and 3 perform probably worse due to a lager probe to die distance, which results in less leakage. Since finding suitable measurement positions is rather easy (in our case, a probability of 50 % using the best probe), attackers should test different measurement positions instead of employing extensive computational brute-force.
Without knowing $pc = 4$ and $i = 1$ a priori, attackers could make minimal heuristic assumptions e.g., $pc \in [3, 10]$ and $i \in [1, 4, 9]$ which could fit similar circumstances. This would result in an additional brute-force complexity of +4 bits which is not included in Fig. 5.4 and justified by significantly improved results.

I compared the performance of the k-means versus the expectation maximization clustering algorithm in the context of single channels. Since I only select single components ($i = 1$) after PCA, channels only consist of single dimensions and there is not much benefit from more free parameters in the clustering algorithm. This is confirmed by the fact that expectation maximization and pc-means clustering *lead to almost equal results*. This means that the reported *improvement is mainly due to the PCA transformation and the selection of components*. In the multi-channel case, however, more dimensions aggregate from separate channels making expectation maximization more eligible.

Comparison of this algorithmic approach to the attack of Heyszl et al. [Hey14] To demonstrate the improvement of the proposal, I carried out the original attack of Heyszl et al. [Hey14] on the same measurements. I firstly compressed the trace of each probe according to the "sum-of-squares" method of each clock cycle and applied k-means clustering afterwards. This results in a remaining brute-force complexity of $\leq$ 32 bit in *none* (0 %) of the 400 measurement cases using the 150 µm probe. Compared to 59 % from the improved attack, this means that an astonishingly improved result from applying PCA and expectation maximization clustering is achieved. (Only the 250 µm probe led to marginally better results using the previous method, i.e., 8 % instead of 3 % of the cases $\leq$ 32 bit, however, this does not invalidate the previous statement in my opinion.)

Profiled Evaluation To compare the performance of the unprofiled algorithmic approach, I mounted a profiled attack. I firstly show the resulting BFCs for the profiled attack and compare the results to the unprofiled case afterwards.

Figures 5.5a, 5.5b, 5.5c, show the BFCs for $i = 9$ over the different principal components for probe 1, 2 and 3 for the profiled case. 57 % of positions (59 % from the non-profiled attack) lead to remaining brute-force complexities $\leq$ 32 bit for the 150 µm probe, with $i = 9$ and $pc = 4$. Using this specific score, the profiled attack does not outperform the non-profiled attack. However, considering the mean values of the brute-force complexity, the profiled template attack outperforms the unprofiled attacks with 50 bits in the mean, instead of 60 bits in the unprofiled case.

Expanding the analysis the occurrences of the BFC between 32 and 160 bits, the profiled analysis shows more occurrences than the unprofiled attack in this region. For the profiled measurements 34 % of measurement positions lead to a BFC between 32 and 150 bits; however in 13 % only, in the unprofiled case. This clearly shows an "all or nothing" tendency for the unprofiled evaluation. Thus, the unprofiled attack is more likely to recover the whole key or no key information at all.

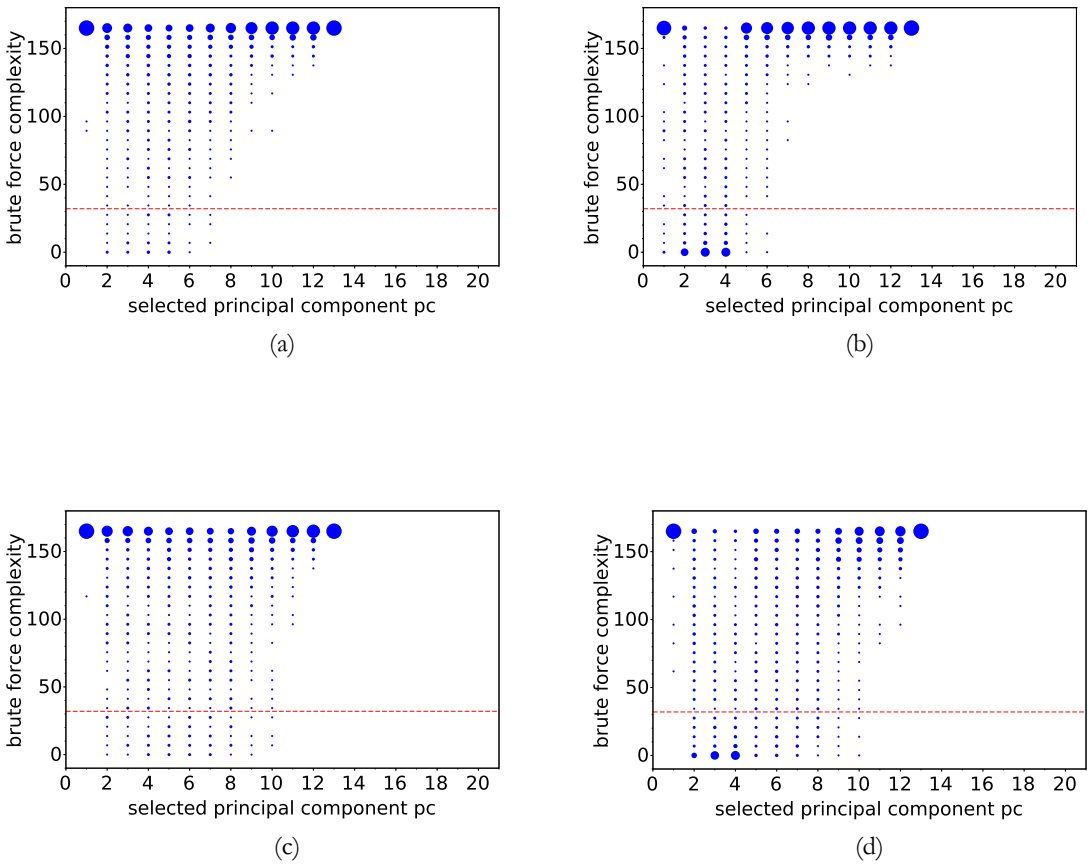


Figure 5.5: Brute force complexity occurrences over different principal components for the profiled evaluation
Single probe 1 (250 µm Ø): Fig. 5.5a
Single probe 2 (150 µm Ø): Fig. 5.5b
Single probe 3 (100 µm Ø): Fig. 5.5c
Combined probes: Fig. 5.5d

5.4 Combining Multiple Channels

In this section I combine the measurement of multiple probes. At first, I will examine the results for the side channel measurements using a concatenation of probe signals.

Multiple simultaneous measurements channels are combined by concatenating the trace-segments from different channels, which correspond to the same exponent bits [Agr03, Hey14]. PCA is applied to all side-channel measurement

channels separately before concatenation. For example, segments $\mathbf{T}^1_{\mathrm{PCA,pc:pc+i}}$ from measurement channel 1 are combined with segments $\mathbf{T}^2_{\mathrm{PCA,pc:pc+i}}$ from measurement 2 leading to combined segments $\mathbf{T}^{\mathrm{combined}}_{\mathrm{PCA,pc:pc+i}} = (\mathbf{T}^1_{\mathrm{PCA,pc:pc+i}}, \mathbf{T}^2_{\mathrm{PCA,pc:pc+i}})$. I chose this approach due to better results, compared to the case of concatenating the traces first and applying PCA afterwards. This procedure leads to a BFC of 67 bits in the mean. This is different to the procedure of the last chapter, where I concatenated first and applied the dimension reduction afterwards. However, first concatenating all 3 probes and afterwards applying PCA leads to a BFC of 92 bits for all 3 probes combined. This shows the limits of our chosen unprofiled method PCA and the difficulty to extract relevant information from high dimensional spaces. Please note that the approach of chapter 4.7.2 is not comparable, because in chapter 4.7.2 no information of multiple probes is combined for the same share.

Hence, I firstly apply PCA to each trace and afterwards concatenate the principal components. Furthermore I show in the following that the selection strategy of the combined components is of central importance to the success of the attack and compare the unprofiled attack to the profiled case. Finally, I will investigate the combination by summing the collected signals of each probe.

Please note that an attacker would rather use the *same values for* pc *and* i *in all channels* because it significantly increases the attack complexity to test different pc-s and i-s for every channel *without profiling* (e.g. repeat the clustering process for all combinations of 20 principal components, with 5 window sizes and 3 probes leads to $(20 * 5)^3$ combinations).
The combination of multiple probes is especially important in the case that an attacker can only observe one execution with a constant key, because it may be one way to increase the observed leakage as explained in chapter 2.2.
For the measurements I chose a fixed geometric arrangement of the probes, close to the surface of an FPGA die and performed 400 measurements at different positions to gain conclusive insights from a high number of tests.

5.4.1 Combining Multiple Probes by Concatenation

After the individual analysis of the three measurement channels, I combined the channels for analysis. Firstly, I perform pre-processing by applying PCA

to each probe individually, secondly I perform clustering using the expectation maximization algorithm and thirdly I compute the remaining brute-force complexity. The motivation for attackers to combine channels is to increase the exploitable leakage to improve attack outcomes, e.g. instead of trying to find better measurement positions.

A visual comparison of the combined results in Fig. 5.4d to the individual results in Figures 5.4a, 5.4b, and 5.4c gives the impression, that the overall result is comparable to Fig. 5.4b. However, expressed quantitatively as before, the combined channels lead to a remaining brute-force complexity of ≤ 32 bit in only 54 % of the cases and a mean of 67 bits for $pc = 4$ (see entry 'MU1' in Table 5.1). Hence, as an important result, instead of an improvement, I observe a slight degradation compared to the best individual case which led to 59 % of cases ≤ 32 bit and a mean of 60 bits (see entry 'SU2" in Table 5.1), which is probably caused by the low amount of leakage from probe 1 and 3 and the leakage in different components. This means that the described clustering-based *non-profiled* attack is *unable to benefit* from a combination of channels (with this measurement setup and DUT).

Selection Strategy of Principal Components for the Unprofiled Combination The degradation in case of (unprofiled) combined probes may be caused by the chosen selection strategy of principal components, which is depicted in Figure 5.6. I perform PCA for every probe individually and select equal values for pc and i and concatenate the selected vectors.
This strategy minimizes the required computational complexity for the selection. Increasing the number of selected components i to lower the number of possible combinations would include more noise, in these circumstances. This would degrade classification results significantly (Fig. 5.2 shows curves with $i > 1$, which result in higher mean-values). To be able to estimate the impact of the chosen selection strategy, I combine the best principal component of every probe for every position, which assumes that either an attacker has a close-to optimal unprofiled selection strategy or that the attacker can test all combinations of principal components for combination. However, testing all combinations of pc's of every channel, increases the complexity, e.g., 20 principal components and 3 probes would result in 20^3 times clustering. Note that the selection of the best components is no realistic attack scenario for todays state-of-the art attacks, because it selects the best components based on a

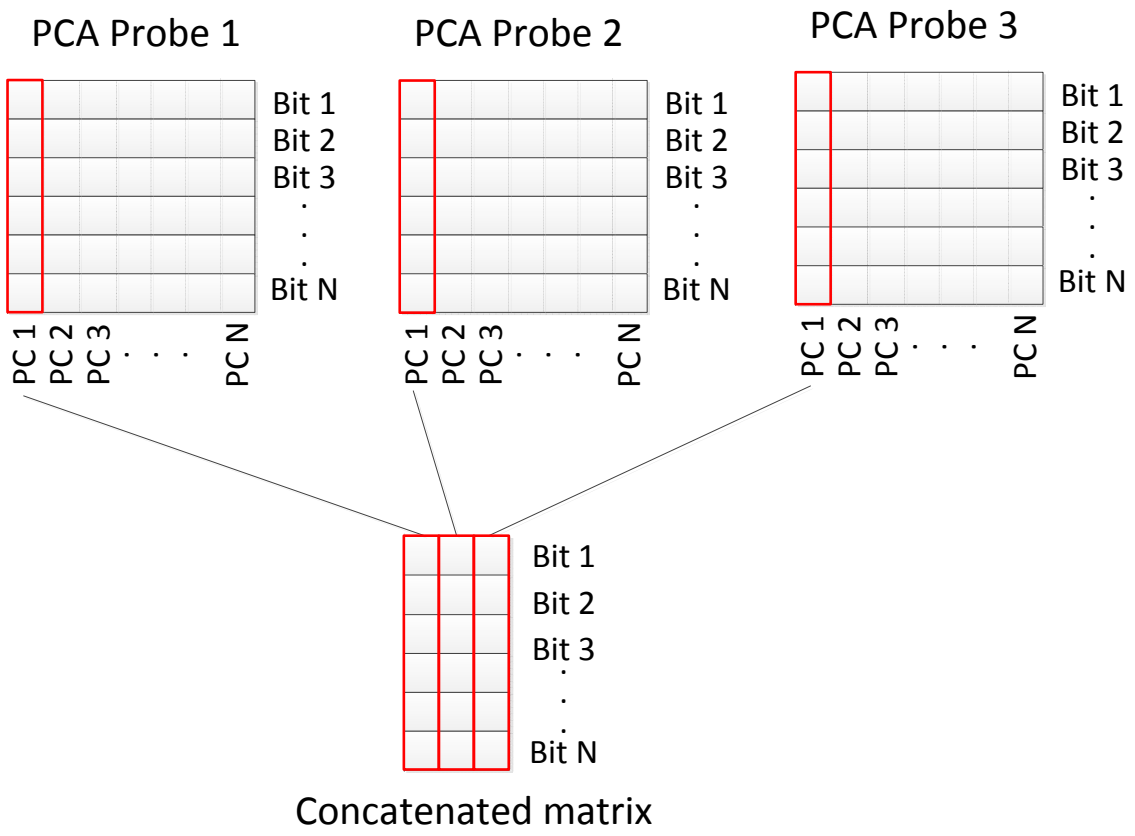


Figure 5.6: Selection strategy for the combination of multiple probes

known key.

I depict the case of this "close-to optimal" selection strategy in Figure 5.4d for pc = 21. It improves the results, compared to the simple selection strategy for combination, where 75 % of positions lead to a brute-force complexity of ≤ 32 bit and a mean of 37 bits. This test shows that the selection strategy of components massively impacts the success of an unprofiled multiprobe attack.

Combination of Simulated Simultaneous Measurements with the Second Probe To show that in general the above presented unprofiled approach is able to improve the attack result by the combination of multiple probes, I combine the second probe at three different positions. Thus, I simulate a simultaneous measurement approach with the exact same probe. The two additional positions are taken from $x+1, y+1$ and $x+2, y+2$, with x, y being the original position, which results in an 18x18 measurement grid. The three traces from different positions are again, transformed separately by PCA and afterwards

#	Type	#Probe	#BFC < 32 Bit	BFC Mean	i
Separate Evaluation					
SU1	unprofiled	1	24	148	1
SU2	unprofiled	2	235	60	1
SU3	unprofiled	3	19	147	1
SP1	profiled	1	41	122	9
SP2	profiled	2	226	50	9
SP3	profiled	3	42	121	9
Combined Evaluation					
MU1	unprofiled	1+2+3	215	67	1
MU2	unprofiled; k = 21	1+2+3	280	37	1
MP3	profiled	1+2+3	242	44	9
MP4	profiled,acc	1+2+3	7	151	9
MU5	unprofiled; three x 150	1+2+3	324 *	30	1

* : extrapolated result

Table 5.1: Measurement results, ECC implementation

concatenated with i = 1, in the same way as explained above. These concatenated traces are again evaluated by the expectation maximization algorithm and the BFC is calculated. To be comparable with the other measurements, it is necessary to extrapolate the number of positions, where the BFC is below 32 Bit, because due to the smaller measurement grid of 324 positions, instead of 400 measurement positions. Extrapolated to 400 measurement positions results in 324 measurement positions below 32 Bit (see entry 'MU1' in Table 5.1) of BFC, which is a major improvement compared to 235 measurement positions below 32 Bit in the individual case. Hence, the above presented unprofiled attack approach is able to combine the leakage of multiple probes. This is probably possible due to the same probe characteristics at each measurement point.

Profiled Evaluation In this section I show the results for the profiled combination by concatenation and compare it to the unprofiled case.

The combination of channels leads to an improved 61 % of the cases with a BFC $\leq$ 32 bit and a mean of 44 bits (see entry 'MP3' in Table 5.1), with

i = 9 and pc = 4, compared to the best individual probe with 57 % brute-force complexity of ≤ 32 bit and a mean of 50 bits (see entry 'SP2' in Table 5.1). This illustrates that the profiled attack sill benefits from measurements with BFCs higher than 100 bits in the mean. For the combination of multiple probes, the profiled evaluation outperforms the unprofiled evaluation in BFC-mean and number of positions with BFC ≤ 32 bit. Comparing Figure 5.5d to Figure 5.4d shows that in the profiled case, the combination can benefit from every single probe, e.g., for pc = 6 to pc = 8 mainly probe 3 shows measurements with a BFC less than 32 bits, which is also the case for the combined evaluation.

5.4.2 Combining Multiple Probes by Accumulation

In this section I combine traces by summing the signals of the individual probes and apply PCA afterwards. The evaluation is only shown for the profiled template attack, which is compared to the profiled template attack for the concatenated combining of multiple probes.

Figure 5.7 shows the mean of BFCs for different i over the selected principal component pc. It reaches a mean of 151 bits and 2 % of the position with a BFC ≤ 32 bit for i = 9. In comparison to the concatenated case, which reaches a mean of 44 bits and 61 % of measurement positions with a BFC ≤ 32 bit, this is significantly worse.

Figure 5.8a and 5.8b show different BFCs over principal components for the combination of probe 1+2 and 1+2+3. Combining probe 1+2 shows again that the second probe is capable to exploit most leakage, which results in a mean of 89 bits and 23 % of measurements positions with a BFC ≤ 32 bit (see entry 'MP4" in Table 5.1). However, adding the third probe diminishes the leakage captured with the second probe, which results in a mean of 151 bits and 2 % of measurement positions with a BFC ≤ 32 bit. In contrast to the concatenated case, the template attack cannot benefit from a probe with an individual BFC mean of 121 bits, in fact it significantly lowers the exploitable leakage.

The results show that combining multiple probes by summing is outperformed by the combination by concatenating the signals.

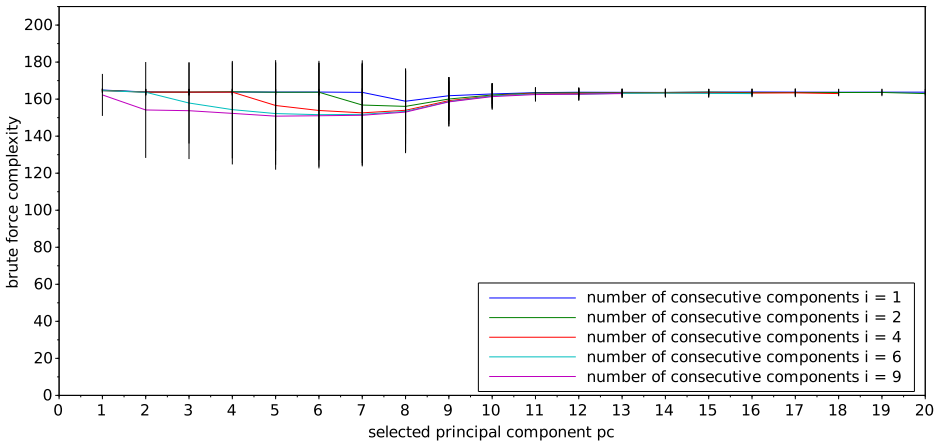
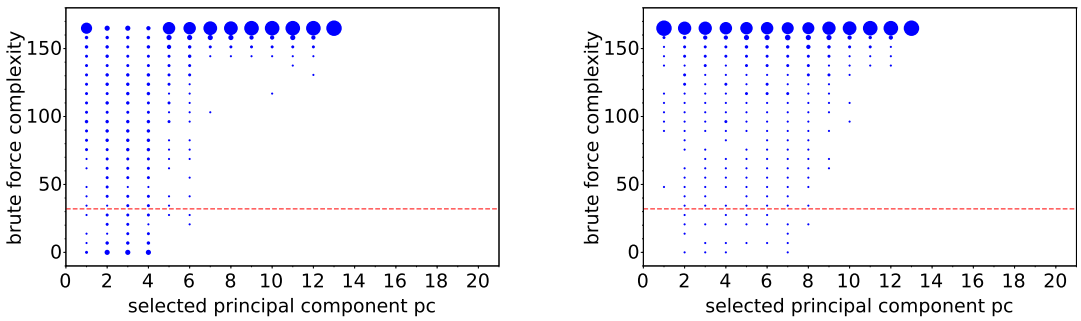


Figure 5.7: Mean brute-force complexity for different selected principal components (pc and i) over all measurement positions *including standard deviation as bars* for the combined profiled evaluation by summing the signals of every probe



(a) (b)

Figure 5.8: Brute force complexity occurrences over different principal components for the profiled evaluation for summing the probe-signals
Combined probes 1+2: Fig. 5.8a
Combined probes 1+2+3: Fig. 5.8b

5.5 Discussions of Principal Component Analysis

In this section I want to discuss in detail, how PCA enables an unprofiled attack to reduce the required brute-force complexity.

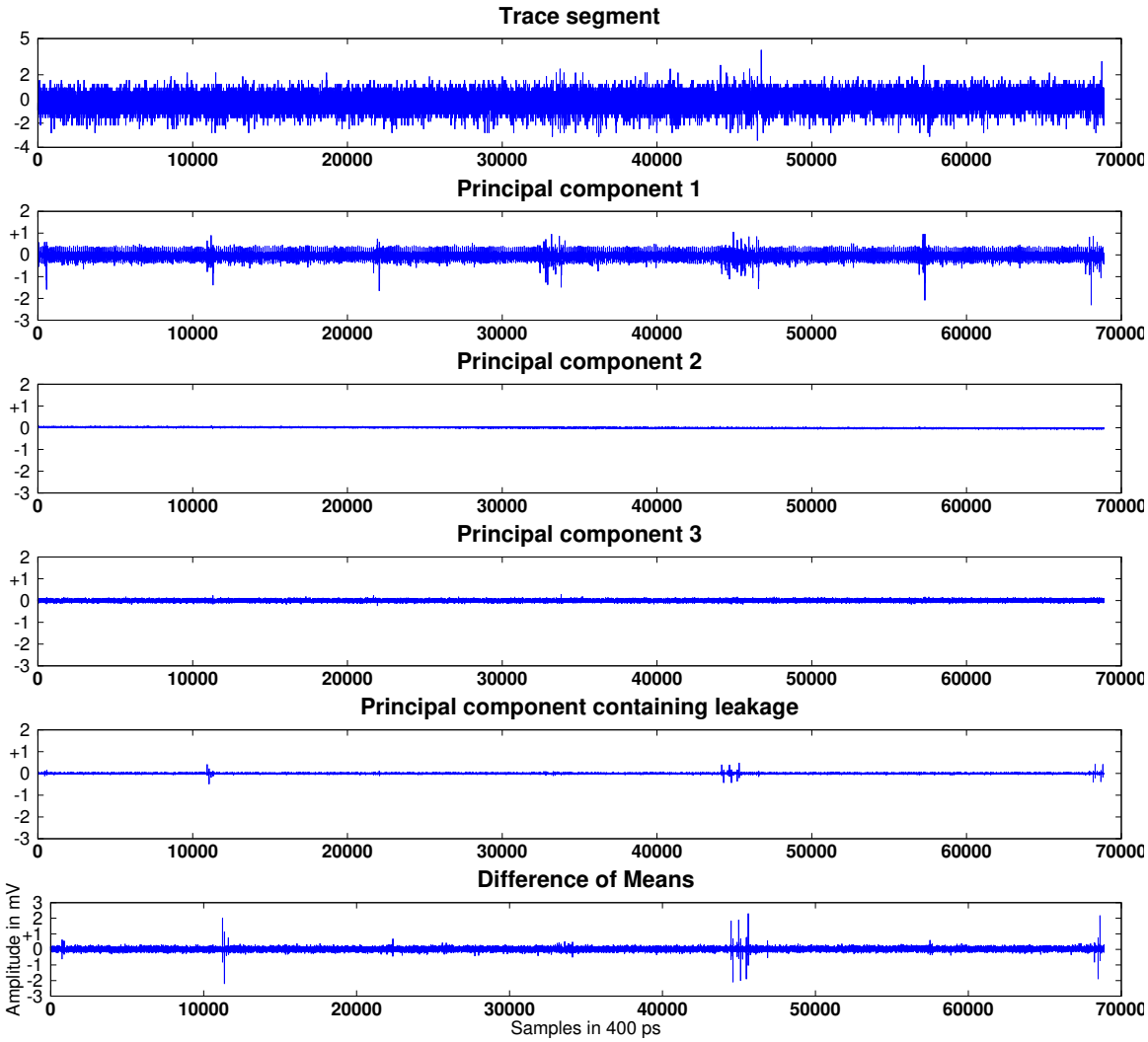


Figure 5.9: Example of an original trace-segment (topmost) and its high-ranked principal components below. The 4-th component contains signal leakage. The bottom trace depicts the profiled Difference of Means (DOM).

To illustrate the power of PCA, figure 5.9 depicts principal components after PCA transformation for illustrative purposes. I used an example measurement

where the side-channel leakage is sufficient for the attack to succeed without false classifications when selecting the pc = 4-th component and i = 1 for expectation maximization clustering. The topmost diagram depicts one trace-segment in its original form. Below, the four highest-ranked principal components are depicted. For figure 5.9, the principal components, which would be represented as a single sample, are transformed back to be comparable to the original trace. The (back-)transformation is calculated by $\mathbf{T}_b = U * \Sigma * V^*$ and setting all singular values in Σ to zero, except the one I want to plot. The first backtransformed principal component shows a regular pattern, with a peak every 125 samples, which corresponds to the clock frequency. This indicates (together with above measurement results) that no key-dependent information is contained in the first principal component. Components 2 and 3 mainly contain no visible structure. However, the fourth principal component (the principal component containing leakage) shows a structure again. The lowest diagram depicts the *profiled* DOM, hence the ideal weights for each point in time. When comparing the DOM with the pc = 4-th component, they show similar POI and weights. This shows that PCA is able to extract the leakage of the traces and can transform the trace segments into key-dependent samples with a low number of principal components. A comparison to the other components in Fig. 5.9 clearly shows that the *leakage is small compared to the remaining signal parts*.

5.6 Discussions of the Accumulated Combination of Probes

In this section I discuss the differences of the presented combination methods, concatenation and summing and their impact on the results. I analyze one trace segment, the corresponding principal components and the DOMs in the summed case, which is depicted in Figure 5.10. In the concatenated case, it looks similar to Figure 5.9 (with triple length).
The topmost signal shows one trace segment. When comparing the trace segment of the summed case to the one from Figure 5.9, the summed case shows a higher signal value of about 10 mV. However, the summed case shows a higher noise level of about +/- 5 mV compared to +/- 2 mV in the concatenated case, e.g., between sample 20000 to 30000. This higher noise level is also shown in the first principal component, which (again) mainly shows peaks corresponding

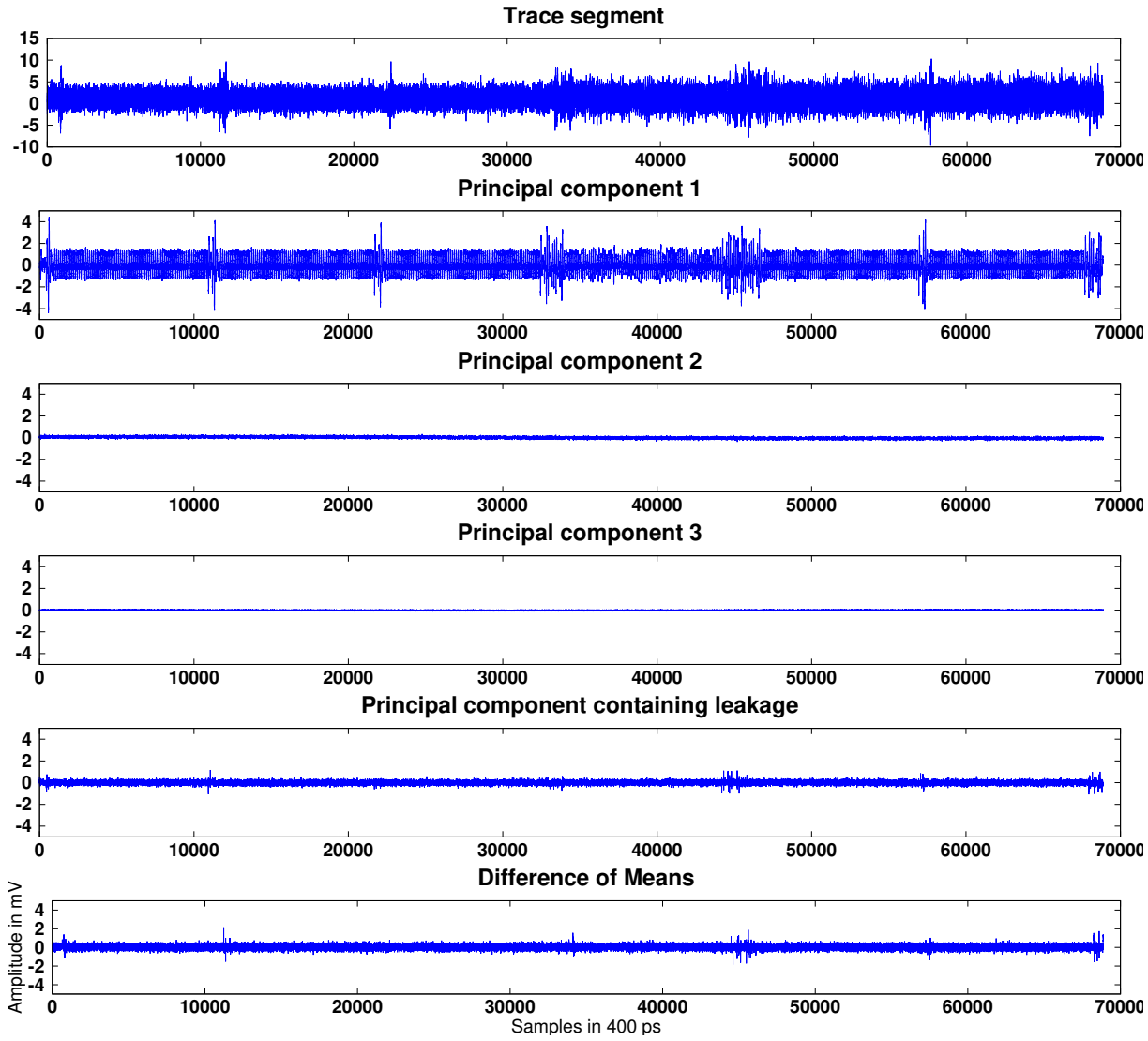


Figure 5.10: Example of an original trace-segment (topmost) and its high-ranked principal components below. The 7-th component contains signal leakage. The bottom trace depicts the profiled DOM.

to the FPGA clock frequency. Again, component 2 and 3 show no meaningful structure. In Figure 5.10 the principal component containing leakage is the 7th (instead of 4 in the above case). Hence, I plot the 7th principal component for comparison with the DOM. Already the component's position (7th) indicates a lower level of leakage (less variance), compared to the trace segment. However, comparing the principal component with leakage of Figure 5.9 to Figure 5.10 shows a similar amplitude and shape. Comparing the 7th principal

component to the DOM demonstrates that PCA is able to extract most of the leakage, even it is much smaller compared to the trace segment. Comparing the DOM of Figure 5.9 to Figure 5.10 shows roughly the same amplitude of 2 mV. Thus, the DOM does not significantly increase when combining the probes by summing the signals. In contrast, the DOM shows a higher noise level, in case no peaks are visible, which corresponds to a lower SNR which leads to less exploitable leakage.
This data supports that in the case of a combination by concatenation, a profiled attack is able to exploit significantly more leakage than in the case of summed signals.

5.7 Discussion of the Coupling Between Probes

In this chapter, the probes were placed in vicinity to each other. Thus, coupling effects are a danger. However, coupling would impact the efficiency of the attack if it lead to similar leakage of the three probes. Comparing the results in figure 5.4 shows that all three probes show different number of positions, where the BFC is smaller than 32 bit. Thus, the common leakage seems to be low and the coupling effects between the three probes are most probably neglectable. Evaluating the coupling on a signal level is challenging, because all three probes collect similar (key independent) signal parts, e.g., clock.

5.8 Conclusion

In this chapter I carried out a profiled and non-profiled single trace side channel attack on an asymmetric cipher by using three probes for measurement. Firstly, I evaluated the multiprobe measurements separately. I showed that I could significantly improve the state-of-the art single trace side channel analysis for one probe, which resulted in a BFC of lower than 32 bits in 59 % of cases and results in the mean in a BFC of 60 bits for the best probe. This improvement is mainly caused by using PCA as preprocessing step for the attack and selecting principal components for the attack. PCA is capable of extracting leakage into one principal component. Like expected, the profiled evaluation method outperforms the non-profiled with a BFC of 50 bits in the mean for the best probe.
In case of a combination of all measurements probes, the unprofiled approach

slightly degrades the results. This is probably caused, by a non-optimal principal component selection strategy in combination with different leaking characteristics of the three probes for the combination of probes. I showed that by combining the probes with a "close-to-optimal" selection strategy improves the results to 280 measurement positions below 32 bit. Furthermore, simulating simultaneous measurements by evaluating the best probe at three positions, shows that in case of a high leakage and similar leaking characteristics, the unprofiled approach is able to improve the results by combining multiple probes. In contrast, the profiled scenario massively benefits from the combination with 3 actual simfultaneous measurements. Even in case that 2 probes did not measure enough leakage to reduce the BFC below 150 bits if evaluated separately. Combining these 2 probes with a third one improved the results of a BFC of 44 bits in the mean, comparing it to the best single probe with a BFC mean of 50 bits. However, these results are only valid for a combination by concatenating the signals. In case of summing the probe signals the combined BFC mean was 151 bits, which is significantly worse compared to the concatenated case. In contrast to the profiled attack, the non-profiled can not benefit from the combination of multiple probes. The combination of 3 probes during a non-profiled attack leads to a BFC mean of 67 bits, compared to 60 bits for the best individual probe. Hence, in a profiled scenario it is beneficial to combine multiple probes.

Chapter 6

Conclusion

In this thesis it has been investigated if multiple probes enable a more powerful attack by combination. This has been tested for one implementation of a symmetric and one implementation of an asymmetric cipher. Countermeasures and attacks differ significantly between symmetric and asymmetric ciphers. Symmetric ciphers, commonly allow thousands or millions observations with the same key. However, implemented countermeasures for asymmetric ciphers can allow only one single side channel observation.[1] I can show that by combining signals of three probes I can significantly improve the performance of the attack against the symmetric cipher. Measuring with multiple localized EM probes allows to isolate and simultaneously observe multiple parts of an IC. Thereby, bypassing assumptions of countermeasures, e.g., in case of TIs that the single shares are not separable. I compare the combination of multiple probes to the separate evaluation and the power side channel. The combination of multiple probes reduces the MTD by factor 5.7, compared to individually processing them and by factor 238 compared to the power side-channel.
Furthermore, I analyze in detail the combination of multiple probes by LDA and the thereby achieved reduction in MTD in chapter 4. I show that LDA is capable of combining measured leakage of multiple probes efficiently in the concatenated case. Furthermore, it weights the signal of all probes according to the leakage (in time and amplitude) detected by a correlation based leakage test.
In chapter 5 a single trace attack is carried out with multiple probes against an asymmetric cipher implementation. Thereby, I first significantly improve the unprofiled algorithmic approach for one probe, which is based on the approach published by Heyszl et al. [Hey14]. I use PCA as preprocessing and expectation-maximization clustering for identifying the key. The improved version of the attack succeeds to lower the BFC in 59 % of cases below 32 bit for one probe,

[1] Also keys can be changed for symmetric ciphers after each execution; however this countermeasure is not common for symmetric ciphers.

where the non-improved version does fail to lower the BFC below 32 bit at all. When combining multiple probes, the achieved BFC improves of a single trace attack from 50 to 44 bits in the mean, compared to the best single probe. Thus, a profiled attack can even benefit from probes, which do not measure strong leakage, e.g., 2 probes in chapter 5 lead to a BFC of larger than 150 bits in the mean. The high BFC of probe 1 and 3 is most probably caused by a higher distance between probes and die than probe 2.
In case of the *clustering-based, non profiled* attack, the results from the combination are only comparable to the best individual one. Thus, the combination of channels only improves the attack results, if a profiled attack is carried out in this work.

However, more experiments are needed to validate if an improvement for combining multiple channels is possible in general. Furthermore, more effort can be spend for a non-profiled principal selection strategy to improve the results of the unprofiled combination of multiple probes. Also the profiled template attack of chapter 4 gets even more powerful in case that no access to the masks is required during profiling phase.

Bibliography

[Agr03] Agrawal, Dakshi and Rao, Josyula and Rohatgi, Pankaj. Multi-channel Attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 2–16. Springer Berlin / Heidelberg, 2003.

[Arc06] Archambeau, Cedric and Peeters, Eric and Standaert ,François-Xavier and Quisquater, Jean-Jacques . Template Attacks in Principal Subspaces. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, pages 1–14, 2006.

[Bal05] Balanis, Constantine A. *Antenna theory: analysis and design*. Wiley-Interscience, 2005. ISBN 9780471667827.

[Ban14] Banerjee, S. and Roy, A. *Linear Algebra and Matrix Analysis for Statistics*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2014.

[Bar10] Bar, Martin and Drexler, Hermann and Pulkus, Jürgen. Improved template attacks. In *Constructive Side-Channel Analysis and Secure Design, COSADE 2010*, 2010.

[Bar15] Barker, Elaine and Roginsky, Allen. Recommendation for the Transitioning of Cryptographic Algorithms and Key Lengths. In *NIST Special Publication*, 2015.

[Bat12] Batina, Lejla and Hogenboom, Jip and Woudenberg, JasperG.J. Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis. In *Topics in Cryptology - CT-RSA 2012*, volume 7178 of *Lecture Notes in Computer Science*, pages 383–397. Springer Berlin Heidelberg, 2012.

[Bau12] Bauer, Sven. Attacking Exponent Blinding in RSA without CRT. In *Constructive Side-Channel Analysis and Secure Design*, volume 7275 of *Lecture Notes in Computer Science*, pages 82–88. Springer Berlin / Heidelberg, 2012.

[Bau15] Bauer, Aurélie and Jaulmes, Eliane and Prouff, Emmanuel and Reinhard, Jean-René and Wild, Justine. Horizontal collision correlation attack on elliptic curves. In *Selected Areas in Cryptography – SAC 2013*, Berlin, Heidelberg, 2015.

[BBHK97] Belhumeur, Peter and Hespanha, Joao and Kriegman, David, P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 711–720. IEEE, 07 1997.

[Ber08] Bernstein, Daniel J. The Salsa20 family of stream ciphers. In *New Stream Cipher Designs: The eSTREAM Finalists*, pages 84–97, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Bha10] Bhasin, Shivam and Guilley, Sylvain and Flament, Florent and Selmane, Nidhal and Danger, Jean-Luc. Countering Early Evaluation: An Approach Towards Robust Dual-Rail Precharge Logic. In *Proceedings of the 5th Workshop on Embedded Systems Security*, WESS '10, page 6, New York, NY, USA, 2010. ACM.

[Bih00] Biham, Eli and Dunkelman, Orr. Cryptanalysis of the A5/1 GSM Stream Cipher. In *Progress in Cryptology —INDOCRYPT 2000*, pages 43–51, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[Bil14] Bilgin, Begül and Gierlichs, Benedikt and Nikova, Svetla and Nikov, Ventzislav and Rijmen, Vincent. A More Efficient AES Threshold Implementation. In *Progress in Cryptology – AFRICACRYPT 2014*, pages 267–284, Cham, 2014. Springer International Publishing.

[Boh03] Bohy, Lilian and Neve, Michael and Samyde, David and Quisquater, Jean-jacques. Principal and Independent Component Analysis for Crypto-systems with Hardware Unmasked Units. In *In Proceedings of e-Smart 2003*, 2003.

[Bri04] Brier, Eric and Clavier, Christophe and Olivier, Francis. Correlation Power Analysis with a Leakage Model. In *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 135–152. Springer Berlin / Heidelberg, 2004.

[Can05] Canright, David. A Very Compact S-Box for AES. In *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *LNCS*, pages 441–455, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Cha99] Chari, Suresh and Jutla, Charanjit and Rao, Josyula and Rohatgi, Pankaj. Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology CRYPTO 99*, pages 791–791. Springer, 1999.

[Cha03] Charim, Suresh and Rao, Josyula R. and Rohatgi, Pankaj. Template Attacks. In *Cryptographic Hardware and Embedded Systems – CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2003.

[Cho13] Choudary, Omar and Kuhn, Markus G. Efficient Template Attacks. In *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, pages 253–270, 2013.

[Cla10] Clavier, Christophe and Feix, Benoit and Gagnerot, Georges and Roussellet, Myne and Verneuil, Vincent. Horizontal Correlation Analysis on Exponentiation. In *Information and Communications Security*, volume 6476 of *Lecture Notes in Computer Science*, pages 46–61. Springer Berlin Heidelberg, 2010.

[Cor07] Coron, Jean-Sébastien and Prouff, Emmanuel and Rivain, Matthieu. Side channel cryptanalysis of a higher order masking scheme. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 28–44, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[De 06] De Canniere, Christophe and Preneel, Bart. TRIVIUM Specifications. In *eSTREAM, ECRYPT Stream Cipher Project*, 2006. http:

//www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf, last accessed: 10.01.2019.

[De 16a] De Cnudde, Thomas and Bilgin, Begül and Reparaz, Oscar and Nikov, Ventzislav and Nikova, Svetla. Higher-Order Threshold Implementation of the AES S-Box. In *Revised Selected Papers of the 14th International Conference on Smart Card Research and Advanced Applications - Volume 9514*, CARDIS 2015, pages 259–272, New York, NY, USA, 2016. Springer-Verlag New York, Inc.

[De 16b] De Cnudde, Thomas and Reparaz, Oscar and Bilgin, Begül and Nikova, Svetla and Nikov, Ventzislav and Rijmen, Vincent. Masking AES with d+ 1 shares in hardware. In *International Conference on Cryptographic Hardware and Embedded Systems - CHES 2016*, pages 194–212. Springer, 2016.

[De 17] De Cnudde, Thomas and Bilgin , Begül and Benedikt and Ventzislav Nikov and Svetla Nikova and Vincent Rijmen. Does Coupling Affect the Security of Masked Implementations? In *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017*, pages 1–18, 07 2017.

[Duc15] Duc, Alexandre and Faust, Sebastian and Standaert, François-Xavier. Making masking security proofs concrete-or how to evaluate the security of any leaking device. In *34th Annual International Conference on the Theory and Applications of Cryptographic Techniques,*, volume 9056, pages 401–429. Springer, 2015.

[Dur16] Durvaux, François and Standaert, François-Xavier. From Improved Leakage Detection to the Detection of Points of Interests in Leakage Traces. In *Advances in Cryptology – EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 240–262, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[Ela11] Elaabid, M. and Meynard, Olivier and Guilley, Sylvain and Danger, Jean-Luc. Combined Side-Channel Attacks. In *Information Security Applications*, volume 6513 of *Lecture Notes in Computer Science*, pages 175–190. Springer Berlin / Heidelberg, 2011.

[Est96] Ester, Martin and Kriegel, Hans-Peter and Sander, Jörg and Xu, Xiaowei. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[Fis36] Fisher, R. A. The Use of Multiple Measurements in Taxonomic Problems. In *Annals of Eugenics*, volume 7, pages 179–188, 1936.

[Gan01] Gandolfi, Karine and Mourtel, Christophe and Olivier, Francis. Electromagnetic Analysis: Concrete Results. In *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer Berlin / Heidelberg, 2001.

[Gou02] Goubin, Louis. A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. In *Public Key Cryptography - PKC 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 199–211. Springer Berlin / Heidelberg, 2002.

[Gro17a] Gross, Hannes and Mangard, Stefan. Reconciling d+1 Masking in Hardware and Software. In *Cryptographic Hardware and Embedded Systems – CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 115–136, Cham, 2017. Springer International Publishing.

[Gro17b] Gross, Hannes and Mangard, Stefan and Korak, Thomas. An efficient side-channel protected aes implementation with arbitrary protection order. In *Cryptographers Track at RSA conference*, pages 95–112. Springer, 2017.

[He 11] He ,Wei and de la Torre, Eduardo and Riesgo, Teresa. A Precharge-Absorbed DPL Logic for Reducing Early Propagation Effects on FPGA Implementations. In *ReConFig 2011*. IEEE Computer Society, 2011.

[He 12] He ,Wei and Oteron, Andrés and de la Torre, Eduardo and Riesgo, Teresa . Automatic Generation of Identical Routing Pairs for FPGA Implemented DPL Logic. In *ReConFig 2012*. IEEE, 2012.

[Her06] Herbst, Christoph and Oswald, Elisabeth and Mangard, Stefan. An AES Smart Card Implementation Resistant to Power Analysis Attacks. In *ACNS 2006*, volume 3989 of *LNCS*, pages 239–252. Springer, 2006.

[Hey12a] Heyszl, Johann and Mangard, Stefan and Heinz, Benedikt and Stumpf, Frederic and Sigl, Georg. Localized Electromagnetic Analysis of Cryptographic Implementations. In *Topics in Cryptology - CT-RSA 2012*, volume 7178 of *Lecture Notes in Computer Science*, pages 231–244. Springer Berlin / Heidelberg, 2012.

[Hey12b] Heyszl, Johann and Merli, Dominik and Heinz, Benedikt and De Santis, Fabrizio and Sigl, Georg. Strengths and Limitations of High-Resolution Electromagnetic Field Measurements for Side-Channel Analysis. In *Smart Card Research and Advanced Applications - 11th International Conference, CARDIS*, pages 248–262, 2012.

[Hey13] Heyszl, Johann. *Impact of localized electromagnetic field measurements on implementations of asymmetric cryptography.* PhD thesis, Technical University Munich, 2013.

[Hey14] Heyszl, Johann and Ibing, Andreas and Mangard, Stefan and De Santis, Fabrizio and Sigl, Georg. Clustering Algorithms for Non-profiled Single-Execution Attacks on Exponentiations. In *Smart Card Research and Advanced Applications*, Lecture Notes in Computer Science, pages 79–93. Springer International Publishing, 2014.

[Imm17] Immler, Vincent and Specht, Robert and Unterstein, Florian. Your Rails Cannot Hide from Localized EM: How Dual-Rail Logic Fails on FPGAs. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 403–424, 2017.

[Ito03] Itoh, Kouichi and Izu, Tetsuya and Takenaka, Masahiko. Address-Bit Differential Power Analysis of Cryptographic Schemes OK-ECDH and OK-ECDSA. In *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 399–412. Springer Berlin / Heidelberg, 2003.

[Jar16] Jarvinen, Kimmo and Balasch, Josep. Single-Trace Side-Channel Attacks on Scalar Multiplications with Precomputations. In *Smart Card Research and Advanced Applications - 15th International Conference, CARDIS 2016, Cannes, France, November 7-9, 2016, Revised Selected Papers*, pages 137–155, 2016.

[Jol02] Jolliffe, I. T. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, 2002. ISBN 9781420095388.

[Kap10] Kaps, Jens-Peter and Velegalati, Rajesh . DPA Resistant AES on FPGA Using Partial DDL. In *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, pages 273–280. IEEE Computer Society, 2010.

[Kat14] Katz, Jonathan and Lindell, Yehuda. *Introduction to modern cryptography*. CRC press, 2014. ISBN 9781466570269.

[Kau90] Kaufman, L. and Rousseeuw, P.J. *Finding Groups in Data: an introduction to cluster analysis*. Wiley, 1990. ISBN 9780471878766.

[Ker83] Kerckhoffs, Auguste. La cryptographie militaire. volume IX, pages 5–83, January 1883.

[Kna92] Knapp, A. W. *Elliptic Curves*. Mathematical Notes Series. Prentice Hall, 1992. ISBN 9780691085593.

[Kob87] Koblitz, Neal. Elliptic Curve Cryptosystems. In *Mathematics of Computation*, volume 48, pages 203–209, 1987. DOI https://doi.org/10.1090/S0025-5718-1987-0866109-5, last accessed: 10.01.2019.

[Koc99] Kocher, Paul C. and Jaffe, Joshua and Jun, Benjamin. Differential Power Analysis. In *Advances in Cryptology — CRYPTO' 99*, volume 1666 of *LNCS*, pages 388–397, Berlin, Heidelberg, 1999. Springer.

[Ler15] Lerman, Liran and Bontempi, Gianluca and Markowitch, Olivier. A machine learning approach against a masked AES. In *Journal of Cryptographic Engineering*, volume 5, pages 123–139. Springer, Jun 2015.

[Lom09] Lomné, Victor and Maurine, Philippe and Torres, Lionel and Robert, Michel and Soares, Rafael and Calazans, Ney . Evaluation on FPGA

of triple rail logic robustness against DPA and DEMA. In *DATE 009*, pages 634–639. IEEE, 2009.

[Man03] Mangard, Stefan. A Simple Power-analysis (SPA) Attack on Implementations of the AES Key Expansion. In *Proceedings of the 5th International Conference on Information Security and Cryptology*, ICISC'02, pages 343–358, Berlin, Heidelberg, 2003. Springer-Verlag.

[Man05a] Mangard, Stefan and Popp, Thomas and Gammel, Berndt M. Side-Channel Leakage of Masked CMOS Gates. In *Topics in Cryptology – CT-RSA 2005*, volume 3376, pages 351–365, Berlin, Heidelberg, 2005. Springer, Springer Berlin Heidelberg.

[Man05b] Mangard, Stefan and Pramstaller, Norbert and Oswald, Elisabeth. Successfully Attacking Masked AES Hardware Implementations. In *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *LNCS*, pages 157–171, Berlin, Heidelberg, 2005. Springer.

[Man07] Mangard, Stefan and Oswald, Elisabeth and Popp, Thomas. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007. ISBN 9780387381626.

[Man11] Mangard, Stefan and Oswald, Elisabeth and Standaert, Francois-Xavier. One for All - All for One: Unifying Standard DPA Attacks. In *IET Information Security*, volume 5, pages 100–110, 2011.

[Mav12] Mavroeidis, Dimitrios and Batina, Lejla and Van Laarhoven, Twan and Marchiori, Elena. PCA, eigenvector localization and clustering for side-channel attacks on cryptographic hardware devices. In *Machine Learning and Knowledge Discovery in Databases*, pages 253–268, Berlin, Heidelberg, 2012. Springer.

[Men92] Menezes, Alfred. Elliptic curve cryptosystems. University of Waterloo, 1992. ISBN 9781461531982.

[Mes00a] Messerges, Thomas S. Securing the AES finalists against power analysis attacks. In *International Workshop on Fast Software Encryption*, pages 150–164, Berlin, Heidelberg, 2000. Springer, Springer Berlin Heidelberg.

[Mes00b] Messerges, Thomas S. Using second-order power analysis to attack DPA resistant software. In *Cryptographic Hardware and Embedded Systems — CHES 2000*, pages 238–251, Berlin, Heidelberg, 2000. Springer, Springer Berlin Heidelberg.

[Mor11] Moradi, Amir and Poschmann, Axel and Ling, San and Paar, Christof and Wang, Huaxiong. Pushing the Limits: A Very Compact and a Threshold Implementation of AES. In *Advances in Cryptology – EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 69–88, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Mor14] Moradi, Amir and Immler, Vincent. Early Propagation and Imbalanced Routing, How to Diminish in FPGAs. In *Cryptographic Hardware and Embedded Systems – CHES 2014: 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, 2014.

[Mor16a] Moradi, Amir and Schneider, Tobias. Side-Channel Analysis Protection and Low-Latency in Action: –Case Study of PRINCE and Midori–. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, pages 517–547. Springer, 2016.

[Mor16b] Moradi, Amir and Standaert, Franancois-Xavier. Moments-Correlating DPA. In *Proceedings of the 2016 ACM Workshop on Theory of Implementation Security*, TIS '16, pages 5–15. ACM, 2016.

[Nas10] Nassar , Maxime and Bhasin, Shivam and Danger, Jean-Luc and Duc, Guillaume and Guilley, Sylvain. BCDL: A high speed balanced DPL for FPGA with global precharge and no early evaluation. In *DATE 2010*, pages 849–854. IEEE, 2010.

[Nic15] Nicolas, Bruneau and Guilley, Sylvain and Heuser , Annelie and Damien, Marion and Rioul, Olivier. Less is More - Dimensionality Reduction from a Theoretical Perspective. In *Cryptographic Hardware and Embedded Systems – CHES 2015*, Saint-Malo, France, September 2015.

[Nik06] Nikova, Svetla and Rechberger, Christian and Rijmen, Vincent. Threshold Implementations Against Side-Channel Attacks and Glitches. In *Information and Communications Security: 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006. Proceedings*, 2006.

[Nik08] Nikova, Svetla and Rijmen, Vincent and Schläffer, Martin. Secure Hardware Implementation of Non-linear Functions in the Presence of Glitches, Information Security and Cryptology—ICISC 2008: 11th International Conference, Seoul. pages 3–5, 2008.

[Nik11] Martin Nikova, Svetla and Rijmen, Vincent and Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. In *Journal of Cryptology*, volume 24, pages 292–321, 2011.

[NIS01] NIST. *Advanced Encryption Standard (AES) (FIPS PUB 197)*. U.S. National Institute of Standards and Technology, November 2001.

[Noh08] Nohl, Karsten and Evans, David and Starbug, Henryk Pltz. Reverse-Engineering a Cryptographic RFID Tag. In *Proceedings of the 17th Conference on Security Symposium*, SS'08, pages 185–193, Berkeley, CA, USA, 2008. USENIX Association.

[Osw05] Oswald, Elisabeth and Mangard, Stefan and Pramstaller, Norbert and Rijmen, Vincent. A Side-Channel Analysis Resistant Description of the AES S-Box. In *FSE 2005*, volume 3557 of *LNCS*, pages 413–423. Springer, 2005.

[Osw06] Oswald, Elisabeth and Mangard, Stefan and Herbst, Christoph and Tillich, Stefan. Practical second-order DPA attacks for masked smart card implementations of block ciphers. In *Topics in Cryptology – CT-RSA 2006*, pages 192–207, Berlin, Heidelberg, 2006. Springer, Springer Berlin Heidelberg.

[Osw07] Oswald, Elisabeth and Mangard, Stefan. Template attacks on masking-resistance is futile. In *Topics in Cryptology – CT-RSA 2007*, volume 4377, pages 243–256, Berlin, Heidelberg, 2007. Springer, Springer Berlin Heidelberg.

[Pee05] Peeters, Eric and Standaert, François-Xavier and Donckers, Nicolas and Quisquater, Jean-Jacques. Improved higher-order side-channel attacks with FPGA experiments. In *Proceedings of the 7th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'05, pages 309–323, Berlin, Heidelberg, 2005. Springer, Springer-Verlag.

[Pee07] Peeters, Eric and Standaert,Francois-Xavier and Quisquater, Jean-Jacques. Power and Electromagnetic Analysis: Improved Model, Consequences and Comparisons. In *Integration, the VLSI Journal*, volume 40, pages 52–60, Amsterdam, The Netherlands, The Netherlands, January 2007. Elsevier Science Publishers B. V.

[Per14] Perin, Guilherme and Imbert, Laurent and Torres, Lionel and Maurine, Philippe. Attacking Randomized Exponentiations Using Unsupervised Learning. In *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, pages 144–160, 2014.

[Pro09] Prouff, Emmanuel and Rivain, Matthieu and Bevan, Régis. Statistical analysis of second order differential power analysis. In *IEEE Transactions on computers*, volume 58, pages 799–811. IEEE, 2009.

[Rah95] Rahmat-Samii , Y. and Williams, L. I. and Yaccarino, R. G. The UCLA bi-polar planar-near-field antenna-measurement and diagnostics range. volume 37, pages 16–35, Dec 1995.

[Rec05] Rechberger, Christian and Oswald, Elisabeth. Practical Template Attacks. In *Information Security Applications*, volume 3325, pages 440–456, Berlin, Heidelberg, 2005. Springer, Springer Berlin Heidelberg.

[Riv09] Rivain, Matthieu and Prouff, Emmanuel and Doget, Julien. Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747, pages 171–188, Berlin, Heidelberg, 2009. Springer, Springer Berlin Heidelberg.

[Riv10] Rivain, Matthieu and Prouff, Emmanuel. Provably secure higher-order masking of AES. In *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 413–427, Berlin, Heidelberg, 2010. Springer.

[Rou87] Rousseeuw, Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. In *Journal of computational and applied mathematics*, volume 20, pages 53–65, Amsterdam, The Netherlands, The Netherlands, November 1987. Elsevier.

[Sau09] Sauvage, Laurent and Guilley, Sylvain and Danger, Jean-Luc and Mathieu, Yves and Nassar, Maxime. Successful Attack on an FPGA-based WDDL DES Cryptoprocessor Without Place and Route Constraints. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '09, 2009.

[Sch03] Schramm, Kai and Wollinger, Thomas and Paar, Christof. A New Class of Collision Attacks and Its Application to DES. In *Fast Software Encryption*, volume 2887 of *Lecture Notes in Computer Science*, pages 206–222. Springer Berlin / Heidelberg, 2003.

[Sch05] Schindler, Werner and Lemke, Kerstin and Paar, Christof. A Stochastic Model for Differential Side Channel Cryptanalysis. In *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer Berlin / Heidelberg, 2005.

[Sha49] Shannon, C.E. Communication theory of secrecy systems. In *The Bell System Technical Journal*, volume 28, pages 656–715, October 1949.

[Sko05] Skorobogatov, Sergei P. *Semi-invasive attacks - A new approach to hardware security analysis*. Number 630 in ISSN 1476-2986. University of Cambridge, 2005.

[Sou12] Souissi, Youssef and Bhasin, Shivam and Guilley, Sylvain and Nassar, Maxime and Danger, Jean-Luc. Towards Different Flavors of Combined Side Channel Attacks. In *Topics in Cryptology - CT-RSA 2012*, volume 7178 of *Lecture Notes in Computer Science*, pages 245–259. Springer Berlin / Heidelberg, 2012.

[Spe14] Specht, Robert and Heyszl, Johann and Sigl, Georg. Investigating Measurement Methods for High-Resolution Electromagentic Field Side-Channel Analysis. In *2014 Internation Symposium on Integrated Circuits*

(ISIC), 2014. This work was already published as part of my masters thesis.

[Spe15] Specht, Robert and Heyszl, Johann and Kleinsteuber, Martin and Sigl, Georg. Improving Non-profiled Attacks on Exponentiations Based on Clustering and Extracting Leakage from Multi-channel High-Resolution EM Measurements. In *Constructive Side-Channel Analysis and Secure Design: 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers*, pages 3–19, Cham, 2015. Springer International Publishing.

[Spe18] Specht, Robert and Immler, Vincent and Unterstein, Florian and Heyszl, Johann and Sigl, Georg. Dividing the Threshold: Multi-Probe Localized EM Analysis on Threshold Implementations. In *2018 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2018, McLean, VA, USA, April 30 - May 4, 2018*, 2018.

[Sta08] Standaert, Francois-Xavier and Archambeau, Cedric. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *Cryptographic Hardware and Embedded Systems - CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer Berlin / Heidelberg, 2008.

[Unt17] Unterluggauer, Thomas and Korak, Thomas and Mangard, Stefan and Schilling, Robert and Benini, Luca and Gürkaynak, Frank K and Muehlberghuber, Michael. Leakage Bounds for Gaussian Side Channels. In *Smart Card Research and Advanced Applications*, pages 88–104, Cham, 2017. Springer International Publishing.

[Ver12] Verdult, Roel and Garcia, Flavio D. and Balasch, Josep. Gone in 360 Seconds: Hijacking with Hitag2. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pages 237–252, Bellevue, WA, 2012. USENIX.

[Vey10] Veyrat-Charvillon, Nicolas and Standaert, François-Xavier. Adaptive Chosen-Message Side-Channel Attacks. In *Applied Cryptography and Network Security*, pages 186–199, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Vey13a] Veyrat-Charvillon, Nicolas and Gérard, Benoît and Renauld, Mathieu and Standaert, François-Xavier. An optimal key enumeration algorithm and its application to side-channel attacks. In *Selected Areas in Cryptography*, pages 390–406. Springer, 2013.

[Vey13b] Veyrat-Charvillon, Nicolas and Gérard, Benoît and Standaert, François-Xavier. Security Evaluations beyond Computing Power. In *EUROCRYPT*, volume 7881, pages 126–141, Berlin, Heidelberg, 2013. Springer, Springer Berlin Heidelberg.

[Wal01] Walter, C. Sliding Windows Succumbs to Big Mac Attack. In *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 286–299. Springer Berlin / Heidelberg, 2001.

[Wil18] Wild, A. and Moradi, A. and Gyneysu, T. . GliFreD: Glitch-Free Duplication Towards Power-Equalized Circuits on FPGAs. In *IEEE Transactions on Computers*, volume 67, pages 375–387, 2018.

[Wol97] Wolpert, David H and Macready, William G. No free lunch theorems for optimization. In *IEEE Transactions on Evolutionary Computation*, volume 1, pages 67–82. IEEE, 1997.

[Yu,07] Yu, Pengyuan and Schaumont, Patrick. Secure FPGA circuits using controlled placement and routing. In *Proceedings of the 5th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, CODES+ISSS '07, pages 45–50, New York, NY, USA, 2007. ACM.

Acronyms

AES: Advanced Encryption Standard

BFC: Brute Force Complexity

CPA: Correlation-based differential Power Analysis

DBSCAN: Density Based Spatial Clustering of Applications with Noise

DES: Data Encryption Standard

DIANA: DIvisive ANAlysis Clustering

DOM: Difference of Means

DPA: Differential Power Analysis

DUT: Device under Test

ECC: Elliptic Curve Cryptography

EM: Electro-Magnetic

FIB: Focused Ion Beam

IC: Integrated Circuit

IOT: Internet Of Things

LDA: Linear Discriminant Analysis

LOI: Location of Interest

MCP-DPA: Moments Correlating Profiled DPA

MIMO: Multiple Input Multiple Output

MTD: Measurements To Disclosure

PCA: Principal Component Analysis

PDF: Probability Density Function

POI: Points of Interest

RSA: Rivest, Shamir und Adleman

SISO: Single Input Single Output

SNR: Signal to Noise Ratio

SPA: Simple Power Analysis

SVD: Singular Value Decomposition

SVM: Support Vector Machine

TI: Threshold Implementation

List of Figures

List of Tables