# Reverse Engineering Microprocessor Content Using Electromagnetic Radiation

Geir Olav Dyrkolbotn

Thesis submitted to Gjøvik University College
for the degree of Doctor of Philosophy in Information Security

2011

# Reverse Engineering Microprocessor Content Using Electromagnetic Radiation

Faculty of Computer Science and Media Technology
Gjøvik University College

Reverse Engineering Microprocessor Content Using Electromagnetic Radiation / Geir Olav Dyrkolbotn
Doctoral Dissertations at Gjøvik University College 2–2011
ISBN: 978-82-91313-73-3
ISSN: 1893-1227

*This thesis is dedicated to my wife, Susan and my two children, Guiliana and Bryan. Without your support and sacrifices this work would not have been possible.*

## Declaration of Authorship

I, Geir Olav Dyrkolbotn, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed:

(Geir Olav Dyrkolbotn)

Date:

# *Summary*

Moore's law has, for almost half a century, described a trend in which the number of transistors in integrated circuits have been doubled every year. Properties, such as processing speed, memory capacity and physical size of circuits, are strongly linked to Moore's prediction. Integrated circuits, such as microprocessors, therefore get smaller yet more and more powerful. The combination of smaller size and larger capacity allow more and more functionality to be included in small microprocessor devices, such as smart phones and smart cards. This includes security related functions, such as confidentiality, integrity, availability and non-repudiation. The use of microprocessor devices is said to make fraud more difficult, however, research has found them susceptible to side-channel attacks. Sensitive information can escape via side-channels such as power consumption or electromagnetic radiation (EMR). When a microprocessor executes its program, power consumption (or resulting EMR) can be used to reveal the content of program and/or data memory of the microprocessor. The correlation between power consumption and microprocessor activity has found many uses: to recover cryptographic keys, to reveal hidden hardware faults, to create a covert channel or to reverse engineer the code executed. This is concerning, considering the increasing demand for and dependability upon microprocessors in secure applications.

This thesis contributes by building a more realistic model of the arsenal available to an adversary engaged in reverse engineering microprocessor content through the electromagnetic side-channel. This includes; (i) presenting a new attack, resembling wireless skimming, (ii) a method for in-depth analysis of EMR and better understanding of what and how much EMR is necessary to launch an attack, (iii) a new power model that better explains the underlying phenomena and (iv) a non-invasive method for reverse engineering physical properties based on EMR.

The Wireless Covert Channel Attack (WCCA) contributes towards exploiting the electromagnetic side-channel in a new attack and attack scenario for microprocessor smart cards. The attack brings together knowledge from different fields; electromagnetic side-channels, covert channels and subversion. The scenario assumes that a highly skilled insider is able to hide a small program (subversive code) on a microprocessor smart card in an early stage of the products life cycle. During normal use of the smart card, the subversive code intentionally manipulates the electromagnetic side-channel, creating a covert channel that can potentially broadcast the cards internal secrets to a nearby receiver. The attack is launched without possession of the card and is, therefore, unlikely to be detected by the user. The feasibility of the attack has been demonstrated on modern, high-security cards with all available security features activated, which demonstrates that attacks resembling wireless skimming are feasible. This contribution highlights the importance of life-cycle security focus for products used in secure applications.

Challenges faced by WCCA and other side-channel attacks are: What and how much of the available EMR is necessary to launch an attack, and how do choices affect the efficiency of the attack? This thesis recognizes reverse engineering microprocessor content as a pattern recognition problem, and can therefore address these challenges as a feature selection problem. A comparison of several multi-class feature selection methods by their performance in a WCCA application is provided. Combining these results with the template attack provides a method for in-depth analysis of the electromagnetic side-channel. This method was applied to data transfer on the microprocessor's internal buses, which gave new insight as to the underlying phenomena and revealed that commonly used power models are not suitable to explain the level of detail achieved by Bayesian classification (e.g. template attack).

This thesis provides the hypothesis that the classification results can be explained by layout dependent phenomena (LDP) . LDP include; (i) inductance and capacitance of conductors, (ii) inductance and capacitance between conductors, (iii) wireless transmission characteristics (i.e. antenna properties) of conductors and other circuit elements and (iv) complex combinations of these phenomena. Simulations and experiments are provided that give new insight as to how capacitance between bus-wires (capacitive crosstalk) influence the energy dissipation and the resulting radiated electromagnetic field in any physical implementation of a digital circuit (e.g. microprocessor). A new power model, based on capacitive crosstalk, is proposed, which better explains the classification results achieved. This can improved side-channel exploitation capabilities.

The new power model shows that energy dissipation (i.e. EMR) is a function of internal physical structures of the microprocessor. It can therefore improve the performance of side-channel attacks that rely upon a good power model to be successful (e.g. power analysis attacks). A spinoff of this result is that if the microprocessor activity is known, it should be possible to reverse engineer physical structures of the microprocessor. This thesis provides a non-invasive method for determining the relative position of internal bus wires based on known transition pattern and the influence of capacitive crosstalk on EMR. By including other LDP it should be possible to reverse engineer other physical structures of the microprocessor. This is, to the best of our knowledge, a new application area for electromagnetic side-channel information and holds potential for future work.

# *Acknowledgments*

This work was carried out at the Norwegian Information Security laboratory (NISlab) at Gjøvik University College (GUC) in Norway. From 2003 until 2010 I was formally a PhD student at the University of Oslo. In 2010 I transferred to GUC.

First, I would like to express my thanks and appreciation to my advisor, Professor Einar Snekkenes. Thank you for all your support, contributions, guidance and especially for motivating me when I was struggling and for holding me back when I tried to do too much. A journey that was supposed to last 4 years has now endured for 8 years. Thank you for your patience and understanding.

During the completion of this thesis I have been employed by the Norwegian Armed Forces. I am much in debt to LtCol Roger Johnsen who saw the military's need for PhD work in information security and who, with the support of Brigadier Bakke, allowed me to take on the task. Thank you Roger for your friendship, unconditional support and encouragements. Without our understanding, ″work when I can, stay home when I need to″, this work would never have been completed. I wish to thank Ivar Kjærem and Bjarte Malmedal for continuing to honor this agreement and for your unconditional support. I would also like to thank all my colleagues in the Armed Forces for your support and patience in waiting for me to finish.

Even though you all finished before me, I want to thank my fellow PhD students; Nils Kalstad Svendsen, Kirsi Helkala, Davrondzhon Gafurov, Hanno Langweg, Knut Wold and Lasse Øverlier for discussions, feedbacks and many inspiring coffee breaks. In particular I would like to thank Nils and Kirsi for being my math experts and Knut for his help with simulations in PSpice as well as the article we wrote together.

I am grateful to Patrick Bours, Slobodan Petrovic and Katrin Franke for their guidance and support, particularly regarding pattern recognition. Thanks to Stephen Wolthusen for his motivating remarks as well as his help regarding LaTex and the formatting of this thesis. Thank you Erik Hjelmås for both administrative and technical help and all the extra trouble you spared me. I am also indebted to Arne Wold who always had the correct answer to any of my signal processing and Matlab questions and to Are Strandli for help with cumbersome statistics. I would also like to thank Jan Audestad for being a source of inspiration both inside and outside the classroom.

Unfortunately, my biggest inspiration for choosing to focus on side-channel attacks and smart cards, never got the chance to see the end result. I enjoyed many inspiring discussions with the late Torgeir Husum, Norway's Mr. Tempest.

Finally, I want to thank my family and friends, in particular my wife, Susan, for the unconditional support and encouragement during these years. Thank you for not giving up on me and for proof reading all my ″interesting″ papers. The completion of this work is just as much to your credit as to mine. I wish to thank my children, Guiliana and Bryan, for constantly reminding me of what is important in life.

# *Contents*

# *List of Figures*

# List of Tables

Chapter 1

# *Introduction*

Is it secret, is it safe?

GANDALF

## 1.1 Problem Description/Motivation

An increasing number of systems rely upon tamper resistant microprocessor devices, such as smart cards, for security related applications. It is well known that microprocessor devices leak information about their activity through side-channels [14]. Side-channel attacks exploit correlations between the internal sensitive information and unintentionally externally available information such as time [9] and power consumption [10], optical [12], acoustic [5] or electromagnetic radiation (EMR) [15]. These attacks can be invasive or non-invasive such that no traces are left behind.

Side-channel attacks are not new, military and government organizations have supposedly used them for a long time. In 1956, MI5s operation ENGULF used telephone taps to record the sound from Hagelin cipher machines. The sound was used to calculate the settings on the Hagelin machines [23]. The electromagnetic side-channel, when EMR is correlated to sensitive information about a system itself or data handled by the system, has even been given its own codeword; TEMPEST. Side-channel attacks were brought to the public interest in 1985 when Van Eck [22] showed how to eavesdrop on video display units from a considerable distance via EMR. In 1996, Anderson and Kuh published, *"Tamper Resistance: A Cautionary Note"* [3], which showed that trusting tamper resistant devices can be problematic. That same year, Kocher [9] published his work on exploiting differences in execution time (Timing Attacks). In 1999 Kocher et al. [10] published their ground-breaking power analysis attacks. Simple Power Analysis (SPA) and Differential Power Analysis (DPA) received some attention from, among others, the banking industry, and countermeasures were publicly announced. Power analysis has since then received a lot of attention, with many improved attacks and countermeasures. In 2007 S. Mangard et al. gave out the book, *Power Analysis Attack - Revealing the Secret of Smart Cards* [14] which is an excellent introduction to the topic, but also summarized development within the field.

In 2000, Quisquater and Samyde [15, 16] applied analysis techniques from SPA and DPA to electromagnetic side-channels, thus introducing electromagnetic analysis (EMA). Gandolfi et al. [8] showed, in 2001, that EMA was not only a theoretical possibility, as cryptographic keys from DES, RSA and the alleged comp128 were successfully retrieved using EMA. In recent years several papers have been published in an ongoing effort to systematically investigate electromagnetic side-channel attacks [1, 2, 6, 8, 11, 13, 15, 16, 17, 18]. The experiments have been extended to some distance from the target, implying that physical access to the target may not be necessary. It has been shown that EMA is at least as powerful as power analysis, and that EMA could circumvent power analysis countermeasures [1, 18]. In 2002, Chari et al. [6], presented the *Template Attacks*, which claims to be close to the theoretical limits of information extraction. There are also papers that further developed these ideas [1, 2, 4, 19]. At USENIX 2002 [17], Quisquater and Samyde described an automatic method to classify instructions, carried out by a simple CISC processor. A neural network (Kohonen's self organizing maps) was trained to automatically recognize, and thus reverse engineer, executed code based on stored electromagnetic and power signatures. In 2010, Rechberger et al. [7] presented a methodology to reverse engineer executed code on a microprocessor, based on side-channel information only.

Due to the continuing reduction in transistor size (Moore's law), microprocessor technology continues to shrink in size yet increase in power. This development has made possible devices such as smart phones and smart cards. However, as we become more dependent upon these devices and use them increasingly for sensitive information, protecting the device and the information it contains becomes ever more important. The importance of understanding side-channel attacks is therefore higher then ever, as no perfect protection exists. The number of problems are many and side-channel attacks are highly application dependent. Each practical case needs to be studied as the results obtained two years ago might be obsolete today. An ongoing effort to understand what is deducible from side-channel information is desirable as vulnerabilities, previously found infeasible to exploit, may be within the adversaries reach today.

## 1.2 Research Questions

Any acceleration of electric charges is accompanied by an electromagnetic field [20]. When a microprocessor executes its program, EMR is therefore generated as a consequence of accelerating electric charges associated with transistor transitions (i.e. the power consumption). This work is concerned with the vulnerabilities in which the correlation between EMR and sensitive information presents to a potential adversary. Better risk assessment and security measures can be achieved when the arsenal available to an adversary is better understood. This thesis is concerned with obtaining a better understanding of the origin of EMR, how to capture and represent EMR as well as how EMR can be exploited from a reverse engineering point of view. The overall goal can be stated as:

> **How can electromagnetic radiation be used to reverse engineer microprocessor content?**

During research the main research question was divided into the following problems, which are explained in further detail in chapter 3:

**Q1.1:** What is state-of-the-art regarding electromagnetic side-channel attacks?

**Q1.2:** Is it possible to demonstrate the correlations between microprocessor activity and electromagnetic radiation without a large investment of resources?

**Q2.1:** Is it possible to launch an attack in a normal scenario on advanced smart cards?

**Q3.1:** How can relevant electromagnetic radiation be selected?

**Q3.2:** What is the performance of a given choice?

**Q4.1:** Can very similar microprocessor activities be distinguished?

**Q4.2:** What model can explain the classification results achieved?

**Q5.1:** Is it possible to reverse engineer the internal physical structure of a microprocessor based on electromagnetic radiation?

## 1.3 Ethical and Legal Considerations

A non-disclosure agreement was signed to get access to modern smart cards with state-of-the-art security measures. Therefore, special consideration has been taken to ensure that business-confidential information and the identity of vendors are not revealed. In addition, a special focus has been devoted to making sure that classified TEMPEST information, available through the Norwegian Armed Forces and NATO, has not been revealed. All resources used for this thesis are unclassified research papers or textbooks.

Each time somebody publishes an article that reveals security vulnerabilities or introduces a new attack, the following question pops up: ″are you not giving the recipe on how to steal information to the wrong guys?″. The naive answer is yes, but in my opinion that would be security through obscurity. To pretend there are no vulnerabilities, thus claiming the system is secure is unethical. As an example, T. Tjøstheim wrote that [21]: *Norwegian online banks have supported secrecy, fearing that the discovery of vulnerabilities could have economic consequences or cause a loss of reputation* . The Norwegian online banks are not unique. In light of this a better question is: ″Are we now obligated to publish findings that bring vulnerabilities out in the open?″ This will facilitate, and in some cases force, more secure systems, and is far better than sticking ones head in the sand, hoping nobody discovers any security vulnerabilities. The chances are, the bad guys know about the vulnerabilities already.

## 1.4 Structure of the Thesis

The rest of the thesis is organized as follows: Chapter 2 presents necessary theory to understand the contributions of this thesis. This includes, basic knowledge of microprocessor technology, the origin and nature of EMR, practical laboratory knowledge and analysis techniques. Chapter 3 presents a summary of the work done for this thesis and shows the relationship between research questions and published papers. Future work is also found in chapter 3. Chapter 4 gives a summary of the contributions of this thesis. In chapter 5-11 the 7 research papers, constituting the main body of this thesis, are found. State-of-the-art is not included, as this is found in each contributing paper.

## 1.5 Bibliography

[1] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., AND ROHATGI, P. The em side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 29–45. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_4`. 1, 15, 17, 19, 23, 37, 50, 71, 85, 88, 113, 127

[2] AGRAWAL, D., RAO, J., AND ROHATGI, P. Multi-channel attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), C. Walter, e. Ko, and C. Paar, Eds., vol. 2779 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 2–16. Available from: `http://dx.doi.org/10.1007/978-3-540-45238-6_2`. 1, 17, 37, 50, 88

[3] ANDERSON, R., AND KUHN, M. Tamper resistance: A cautionary note. In *In Proceedings of the 2nd USENIX Workshop on Electronic Commerce (WOEC 96* (1996). 1, 50

[4] ARCHAMBEAU, C., PEETERS, E., STANDAERT, F. X., AND QUISQUATER. Template attacks in principal subspaces. In *Cryptographic Hardware and Embedded Systems - CHES* (2006), vol. 4249 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1–14. 1, 30, 39, 72, 91

[5] ASONOV, D., AND AGRAWAL, R. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy* (may 2004), pp. 3 – 11. 1

[6] CHARI, S., RAO, J., AND ROHATGI, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_3`. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[7] EISENBARTH, T., PAAR, C., AND WEGHENKEL, B. Building a side channel based disassembler. In *Transactions on Computational Science X* (2010), vol. 6340 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 78–99. Available from: `http://dx.doi.org/10.1007/978-3-642-17499-5_4`. 2

[8] GANDOLFI, K., MOURTEL, C., AND OLIVIER, F. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES* (2001), vol. 2162 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 251–261. Available from: `http://dx.doi.org/10.1007/3-540-44709-1_21`. 1, 7, 15, 17, 21, 26, 37, 85, 113, 127

[9] KOCHER, P. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology* (1996), vol. 1109 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 104–113. Available from: `http://dx.doi.org/10.1007/3-540-68697-5_9`. 1, 26, 50, 71, 88

[10] KOCHER, P., JAFFE, J., AND JUN, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: `http://dx.doi.org/10.1007/3-540-48405-1_25`. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[11] KUHN, M., AND ANDERSON, R. Soft tempest: Hidden data transmission using electromagnetic emanations. In *Information Hiding* (1998), vol. 1525 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 124–142. Available from: `http://dx.doi.org/10.1007/3-540-49380-8_10`. 1, 7, 37, 46, 51

[12] KUHN, M. G. Optical time-domain eavesdropping risks of crt displays. In *IEEE Symposium on Security and Privacy* (2002), pp. 3 – 18. 1

[13] KUHN, M. G. *Compromising Emanations: Eavesdropping Risks of Computer Displays*. Technical report, University of Cambridge, 2003. UCAM-CL-TR-577. 1, 25, 37

[14] MANGARD, S., OSWALD, E., AND POPP, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[15] QUISQUATER, J.-J., AND SAMYDE, D. A new tool for non-intrusive analysis of smart cards based on electromagnetic emissions:the sema and dema methods. *Eurocrypt rump session* (2000). 1, 37, 50, 88

[16] QUISQUATER, J.-J., AND SAMYDE, D. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security* (2001), vol. 2140 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 200–210. Available from: `http://dx.doi.org/10.1007/3-540-45418-7_17`. 1, 7, 15, 16, 17, 21, 26, 37, 50, 88

[17] QUISQUATER, J.-J., AND SAMYDE, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from:

http://portal.acm.org/citation.cfm?id=1250988.1250994. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[18] RAO, J., ROHATGI, AND PANKAJ. Empowering side-channel attacks. Tech. rep., IBM T.J. Watson Research Center, 2001. 1, 7, 16, 17, 37, 50, 88

[19] RECHBERGER, C., AND OSWALD, E. Practical template attacks. In *Information Security Applications* (2005), vol. 3325 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 440–456. Available from: http://dx.doi.org/10.1007/978-3-540-31815-6_35. 1, 30, 91, 93

[20] SKITEK, G., AND MARSHALL, S. *Electromagnetic Concepts and Applications*. Prentice Hall, 1987. 2, 14

[21] TJØSTHEIM, T. *Security analysis of electronic voting and online banking systems*. Ph.D. thesis, The University of Bergen, Department of Informatics, 2007. 3

[22] VAN ECK, W. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security 4*, 4 (1985), 269 – 286. Available from: http://www.sciencedirect.com/science/article/pii/016740488590046X. 1, 50

[23] WRIGHT, D. *Spycatcher*. Viking Penguin Inc, 1987. 1

Chapter 2

# Background

> If we knew what it was we were doing, it would not be called research, would it?
>
> ALBERT EINSTEIN

A microprocessor has a functional activity which is to transform a set of input bits to a set of output bits. It is well known that the functional activity also generates electromagnetic radiation (EMR) as a function of the power consumption of the microprocessor [13, 18, 29, 32]. This is concerning, from an information security point of view, if the EMR is correlated to sensitive information about, or handled by the microprocessor. Compromise of sensitive information is then at risk if an adversary is able to capture and analyze the EMR, known as the electromagnetic side-channel.

Electromagnetic side-channel attacks can be modeled as seen in Figure 2.1. The microprocessor executes it program using as set of basic operations (e.g. transfer, arithmetic, logic and shift), usually controlled by a ”square wave” clock. A sequence of basic operations necessary to execute a program is called an execution sequence, $o$. The majority of the power consumption, and associated EMR, during an execution sequence is related to the number of gates that change state [20]. This is determined by what bit values are processed and moved by the microprocessor. The EMR from each basic operation is, therefore, a function of the instruction (opcode and operand), the data, the address in memory and the prior state (upstate) of the microprocessor. This relationship can be exploited by an adversary to deduce the content of program or data memory, otherwise kept secret. The adversary may have knowledge of, or even be able to, choose part of the execution sequence in order to reveal specific details (e.g. run a known cryptographic algorithm for an unknown value of the cryptographic key [17] or alternate the execution of two different instructions [10]). The emitted electromagnetic signal, $s$, in Figure 2.1 will have a signal part and a noise part. The signal part, $s_{exp}$, is due to power consumption during gate transitions related to handling sensitive information. The noise part $s_{noise}$ is due to power consumption during gate transitions irrelevant to handling the sensitive information. Any electronic device will also be subject to electronic noise, $n_{el}$ [20].

Side-channel analysis requires interception of EMR subject to channel noise, $n_{ch}$, and measurement noise, $n_m$. Properties (features), $x$, of the intercepted signal are then extracted

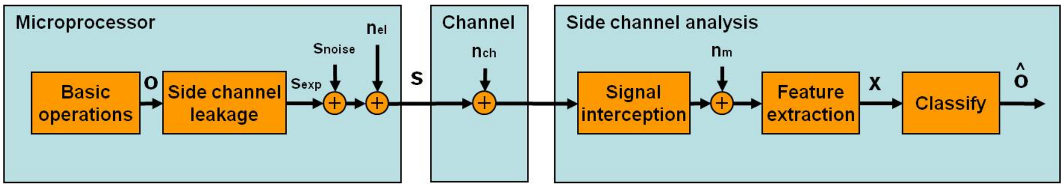

Figure 2.1: Electromagnetic side-channel leakage from a microprocessor.

and used to compute an estimate of the execution sequence, $\hat{o}$. The estimated execution sequence, together with known or chosen memory content, can then be used to reveal sensitive information, e.g. the content of program or data memory. This process can be viewed as a pattern classification problem [9]. Based on properties, $x$, of the intercepted EMR, predict the execution sequence which took place. Side-channel attacks have adapted a range of different classification approaches, such as visual inspection [17], difference of means or correlation coefficient [17], Bayesian classifier [8] or neural networks [30].

EMR only pose a security risk if sensitive information is correlated to EMR and the adversary can intercept and extract the information. In order to understand how EMR can reveal sensitive information, it is necessary to understand how basic microprocessor operations and execution sequences are correlated to the power consumption. This is the topic of section 2.1. Section 2.2 talks about how EMR is generated and how it is related to power consumption. The measurement setup necessary to intercept EMR is introduced in section 2.3 and section 2.4 introduce relevant analysis techniques.

## 2.1 Microprocessors

The purpose of this chapter is to get an understanding of how a microprocessor executes its program and how this is related to the power consumption. This will help understand how sensitive information handled by a microprocessor can be compromised through measurements of the power consumption.

It is difficult and resource-demanding for an adversary to capture and analyze the power consumption of the microprocessor every waking moment. In addition, it is likely to be unnecessary as correlations between sensitive information and power consumption probably are found at specific time instances. Detailed knowledge of the program and how it is executed by the microprocessor, i.e. the execution sequence, is then important to identify the time instance of relevant activity. The basic principles of a general microprocessor are covered first, followed by a detailed explanation of how a microprocessor smart card works. This introduction is based on the architecture of the Goldcard smart card, containing Microchips PIC 16F84A microcontroller, as it is easily available without a non-disclosure agreement and principles are easily generalized to more advanced cards used in security sensitive applications (e.g. mobile phones, travel documents, electronic wallets, credit cards and pay TV). Finally, an introduction to power consumption associated with bit-level microprocessor activity is given.

### 2.1.1 General microprocessor

A microprocessor typically includes memory (program memory, data memory), Central Processing Unit (CPU) , I/O and buses (data, program, control). The CPU is responsible for controlling the operation of the device by fetching, decoding and executing instructions, one by one. A set of instructions is called a program and is stored in program memory. Each instruction consist of opcode and operand . The opcode specifies the instruction type and the operands the operation of the instruction, e.g. memory location of data and where to store the result. The CPU use a set of basic operations such as: Transfer, arithmetic, logic and shift. Transfer is used to move data on buses between memory locations. Arithmetic, logic and shift operations are done by the Arithmetic Logical Unit (ALU) of the CPU. The ALU can perform arithmetic and boolean function between a working register and memory locations. The execution of basic operations (i.e. execution sequence) is usually controlled by a square wave clock. How fast the microprocessor can perform the tasks instructed by the instructions, depend on the clock and if parallel activities can take place. In traditional von Neumann architecture [23] data and program are fetched over the same bus. In Harvard architecture [23] separate buses are used, allowing data and instructions to be fetched simultaneously, thus increasing the speed of the microprocessor. This also makes

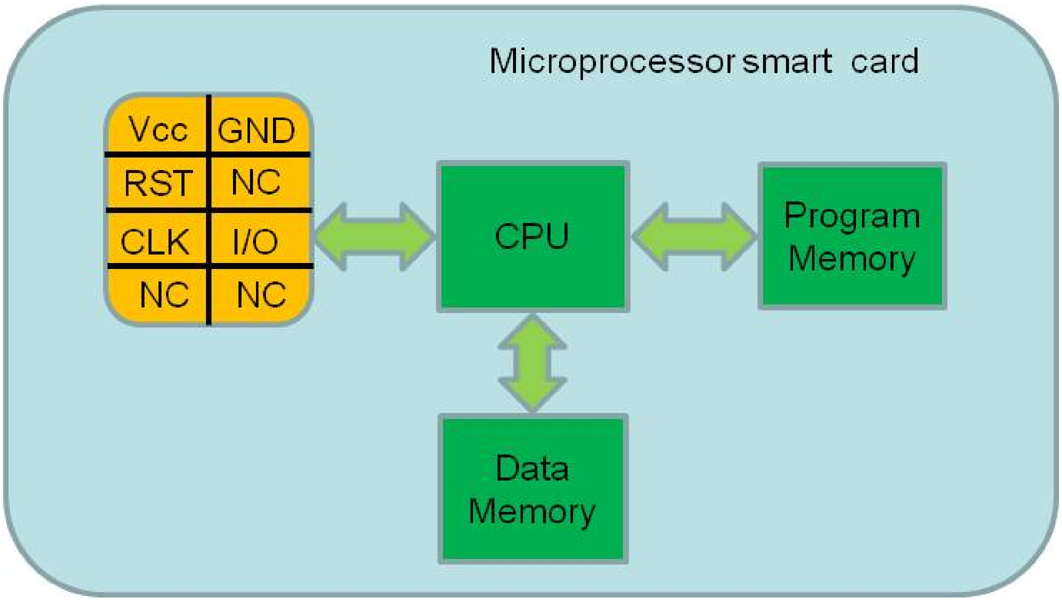

Figure 2.2: Smart card - a complete computer.

it possible to have data and instructions of different lengths, allowing for more efficient use of the program memory. This is important in resource limited devices, such as smart cards.

### 2.1.2 Smart Cards: Specific processor

Microprocessor smart cards  are complete computers embedded in a small chip (Figure 2.2), and typically contain CPU, program and data memory (RAM, ROM, EEPROM) , I/O interface and buses. They are typically used in security sensitive applications and sometimes have sensors, security logic and specialized coprocessors for handling cryptographic algorithms [31]. The electrical connections are also shown in Figure 2.2. A smart card usually have 8 contacts. ISO 7816-2 specifies the designation and function of the contacts [31]. Two connectors are reserved for auxiliary contacts and one has gone out of use (NC). The other connectors are $I/O$ for serial communication, $V_{cc}$ for power supply, $CLK$ for clock input, $GND$ for ground and $RST$ for reset input.

The Goldcard  is a microprocessor smart card which contains an 8 bit Microchip PIC 16F84A special-purpose microprocessor (i.e. microcontroller), which handles clock speeds up to 20 MHz. A simplified block diagram of PIC 16F84A is seen in Figure 2.3. The CPU consists of a Control Unit (CU) , ALU and a working register (w-register). The program memory can hold 1024 14bits instructions. 68 bytes of RAM and 64 bytes of EEPROM is also available. A Harvard architecture with separate 8 bit data and 14 bit program bus allows some parallel activities to take place. There are 35 instructions available grouped into three basic categories: byte-oriented operations, bit-oriented operations and literal and control operations. For details, see Microchips Reference Manual [22].

Microchip PIC 16F84A executes one instruction in four quadrature clock cycles (Q cycles) , which have a period equal to the clock provided through the $CLK$ connection (Figure 2.2). A two-stage pipeline is used to fetch the next instruction while the current instruction is executed (Figure 2.4). The microprocessor executes one instruction in each instruction cycle, except for program branches that require two cycles. Clock cycle 1 (Q1) is used to decode the current instruction in CU's instruction register. This may involve setting an address on the 7 bit address bus to RAM, setting an 8 bit literal to the ALU or setting up

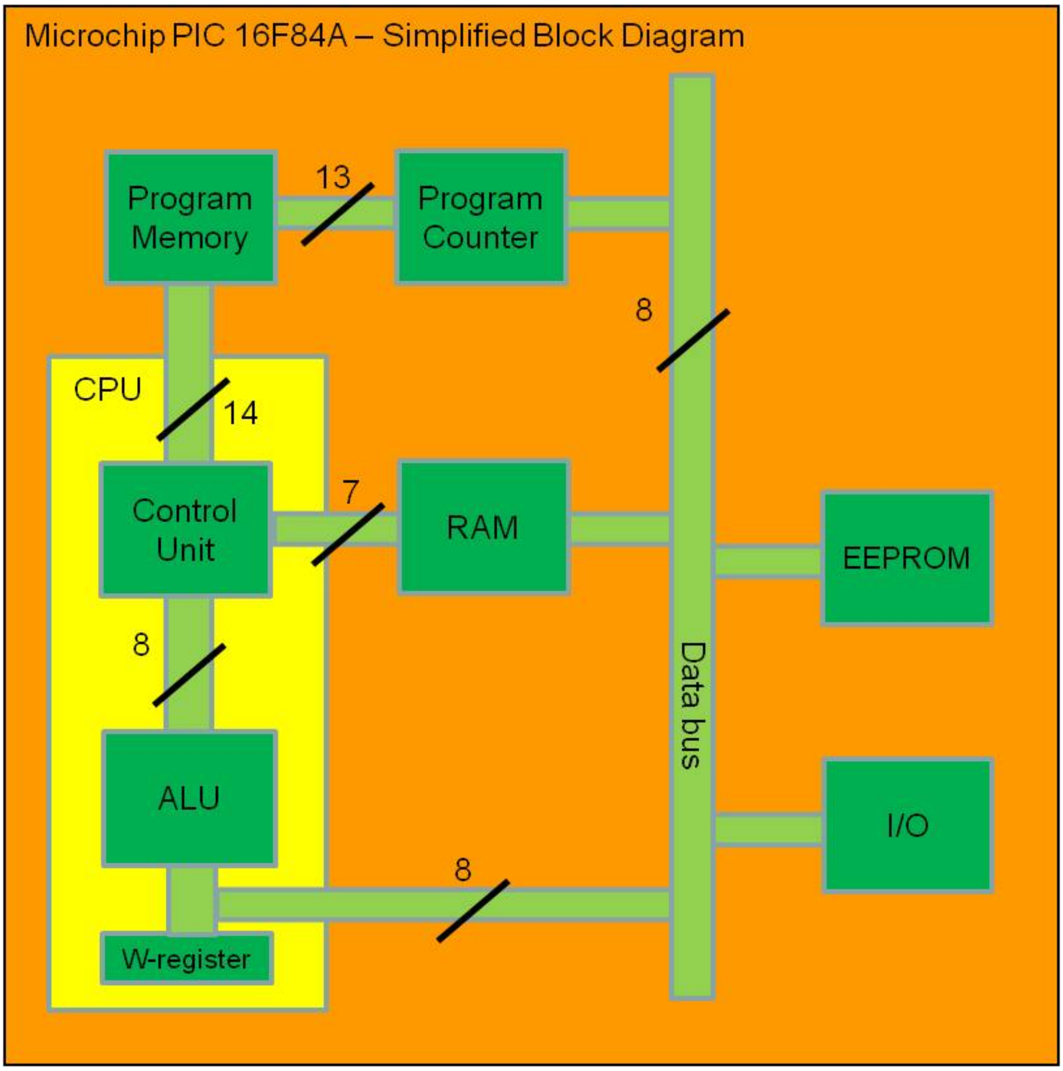

Figure 2.3: Simplified block diagram of Microchip PIC 16F84A.

control lines to ensure correct processing as specified by the opcode. Q1 is also used to increment the program counter, which points to the next instruction in program memory. Clock cycle 2 (Q2) is the operand read cycle where data memory is read, e.g. transfer data from RAM at the address provided by CU, to the ALU via the 8 bit data bus. Clock cycle 3 (Q3) is used to process data in the ALU as specified by the opcode (i.e. control signals set by CU). This could be addition, subtraction, shift or logic operations. Finally, clock cycle 4 (Q4) is used as the destination write cycle. Results of ALU operations in Q3 are either written to the working register or back to the memory location used in Q2. Q4 is also used to move the next instruction from program memory to the CU's instruction register. One instruction cycle is then complete and the next cycle can start.

Such detailed knowledge of the execution sequence can be used to identify time instances of basic microprocessor operations (e.g. transfer) of interest. An adversary interested in the opcode or memory location of instructions should focus on Q1. Correlation between power consumption and one particular byte of data, may be strongest during the read cycle (Q2) or the write cycle (Q4). Notice that in Q4 the parallel activities of moving

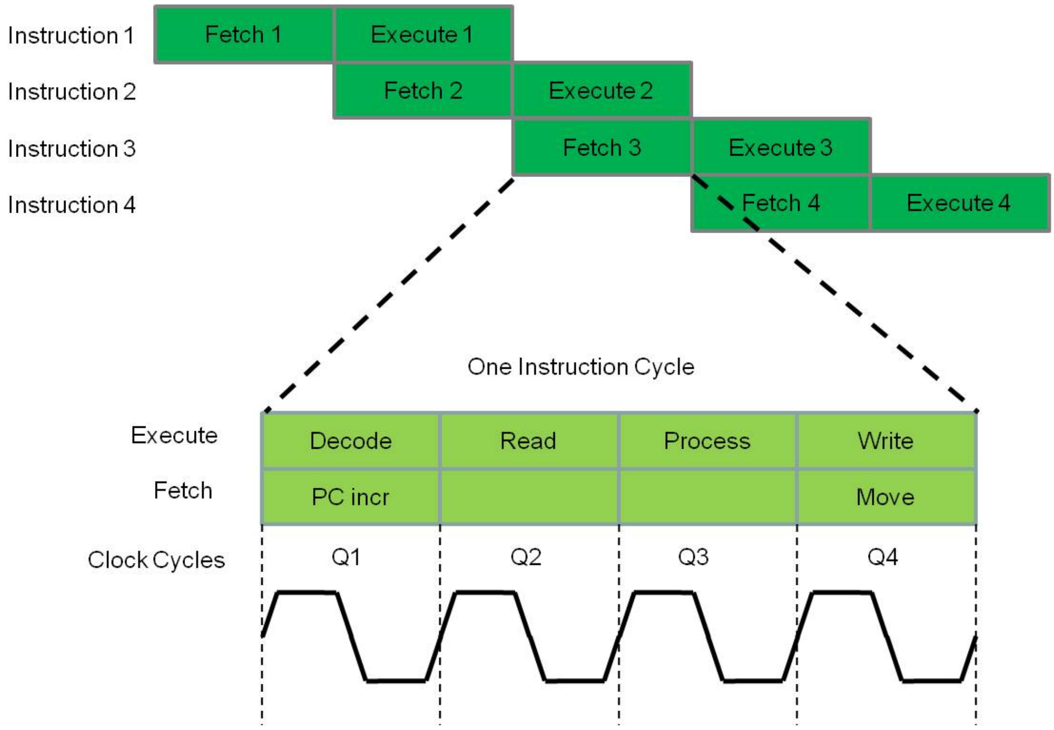

Figure 2.4: Execution sequence and pipelining of microprocessor PIC 16F84A.

the next instruction into the CU will add to the noise $s_{noise}$ (Figure 2.1) and make analysis harder. Correlations between power consumption and processing data (e.g. XOR of cryptographic key) in ALU is expected to be strongest in Q3. This information should only be used to initiate the analysis. There may well be correlations that do not follow these simple guidelines. For other microprocessor technologies, the execution cycle itself can provide information, e.g. a Motorola microcontroller has 3 or 4 clock cycles in the instructions cycle, depending on the opcode.

### 2.1.3 Power Consumption

The purpose of this chapter is to get an understanding of the power consumption associated with basic microprocessor operation, in particular, bit-level transfer operations on internal buses. This introduction is largely based on the book, Power Analysis Attacks [20] in which further detail can be found.

Digital circuits, such as ASIC's , FPGA's and microprocessors, are built using logic cells. Logic cells can be implementations of boolean function like inversion, NAND and XOR and are referred to as combinational cells as their outputs are logical combination of the inputs. Other types of logic cells are flip-flops and registers. These are called sequential cells, and the output depend on the input, preceding inputs or on their initial state. Combinational cells and sequential cells are used to build functionality of higher complexity, such as adders, counters and state machines. These building blocks are then used to build systems (e.g. microprocessor). All logic cells are implemented using transistors. There are many different types of transistors, however, the majority of cells use complementary metal-oxide semiconductor (CMOS) technology [27].

A precise simulation of the power consumption of digital circuits can be obtained using analog circuit simulators like PSpice [7] . This requires a netlist of all transistors, the con-

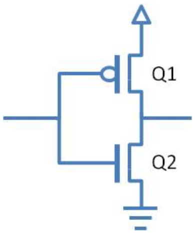

Figure 2.5: CMOS inverter.

nections between them and parasitic elements. The precision of the simulation depends on how accurate the parasitic elements are modeled. Simplifications are usually done to reduce the complexity. The lumped-C model is common, where all parasitic elements are lumped together as single capacitance at the output of a cell [20].

It is also possible to simulate the power consumption at a logic level. A netlist of logic cells and the connection between them allows simulation of the transitions taking place in the circuit. This can then be mapped to the power consumption by the circuit. Both these methods require detailed knowledge to make the netlits, that may not be available to an attacker. However, devices such as the microprocessor have components (buses, ALU, memory) that behave in a predictable way (e.g. execution sequences and basic operations). For an attacker it may be enough to map the power consumption to bit-level activities of the device, and the absolute power consumption may not be relevant.

The CMOS inverter is often used to explain the power consumption in CMOS logic cell. It is representative for other logic cells, because they are all based on the same principle of pull-up and pull-down networks. The CMOS inverter uses two transistors; one p-channel (Q1) and one n-channel (Q2) MOSFET [27] as seen in Figure 2.5.

There are 4 cases to consider. When the input is high (logic 1), Q1 is off and Q2 is on. This connects the output to ground. If the output was previously high a current pulse occurs as the output is discharged through Q2, resulting in a power consumption $P_{10}$. If the output was already low there is no discharge, but only a small static power consumption, $P_{00}$ . When the input is low, Q1 is on and Q2 is off. This connects the output to the power supply. If the output was previously low a current pulse occurs as the output charges up through Q1, resulting in a power consumption $P_{01}$. If the output was already high there is a small static power consumption, $P_{11}$. In the case of a transition on the output of the inverter, the power consumption will be much larger as a result of the added power associated with charging up or discharging the output. This is called dynamic power consumption [20]. The results are summarized in table 2.1.

Notice that the static power consumption is very small and usually neglected, such that $P_{00} = P_{11} \approx 0$. According to [20], the leakage current is in the range of 1pA. Since the majority of the power consumption occurs when there is a transition between logic one

Table 2.1: Relationship between power consumption and transitions for a CMOS inverter [20].

| Transition | Power Consumption | Type |
|---|---|---|
| $0 \rightarrow 0$ | $P_{00}$ | static |
| $0 \rightarrow 1$ | $P_{01}$ | static and dynamic |
| $1 \rightarrow 0$ | $P_{10}$ | static and dynamic |
| $1 \rightarrow 1$ | $P_{11}$ | static |

and zero, it can be argued that it is the number of changes in logic value that determines the power consumption. It is also common to assume the dynamic power for discharge and charging the output is equal, such that $P_{01} = P_{10}$. This is the background for two common power models: The Hamming Weight (HW) and Hamming Distance (HD) model.

HW is simply the number of bits set to one, and HD is the number of bits that are different. For two binary values $X$ and $Y$, both with length $l$, Hamming distance is given by $HD(X, Y) = \sum_{i=1}^{l} X_i \oplus Y_i$. The HD model can be used to describe the power consumption when consecutive data are known. This is often the case for buses and registers of a microprocessor. The HW model is used if no consecutive data is available and is therefore, in general, not very well suited for CMOS circuits. However, in [20] it is stated that ″..., in practice the HW of a data value is usually not completely unrelated to the power consumption that is caused by the processing of this value.″

Another explanation can be illustrated by considering a parallel bus (e.g. the data bus of a microprocessor), with 4 wires ($w1, \cdots, w_4$) transmitting data simultaneously. Current is drawn by each bus wire during the rising and falling edge of each logic ″1″. The dynamic power consumption will be proportional to how many wires conduct current, as illustrated in Figure 2.6. On the right side of Figure 2.6 the ″square wave″ signal has equal duration to the bit interval (non-return to zero - NRZ) . The power consumption is then proportional to the HD of the data. On the left side, the duration of the ″square wave″ signal is shorter than the bit interval (return to zero - RZ) . The power consumption is then proportional to the HW of the data. Pre-charge bus is one type of RZ.

To summarize: The power consumed by basic microprocessor (CMOS based) operation is proportional to bit values handled by the device, which is a function of the instruction (opcode and operand), the data, the address in memory and the prior state (upstate) of the microprocessor.

## 2.2 Electromagnetic Radiation

Electromagnetic radiation (EMR) is energy propagating through space. This energy has both a wave-like and a particle-like behavior that depends on the wavelength. EMR of different wavelengths are known by their more familiar names: radio waves, microwaves,

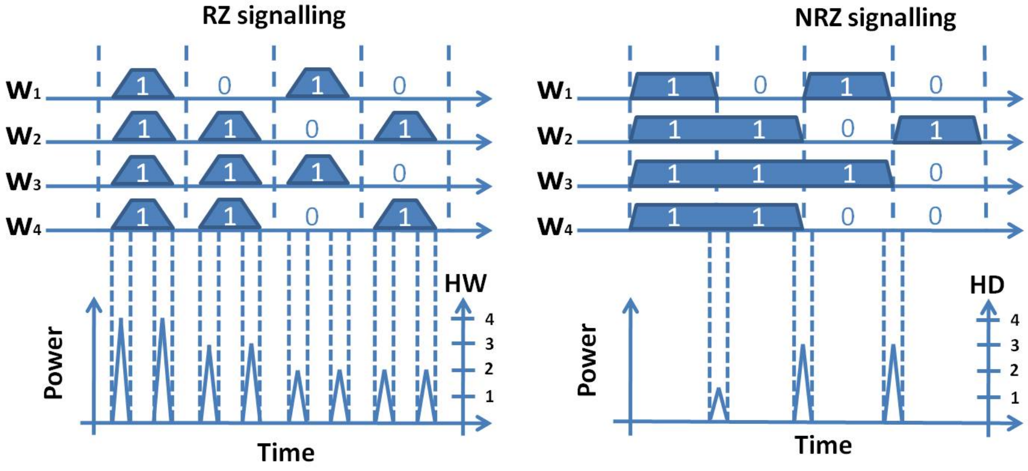

Figure 2.6: The power consumption of RZ signal types (left) is proportional to the HW of the data. The power consumption of NRZ signal types (right) is proportional to the HD of the data.

infrared, visible light, ultraviolet radiation, X-rays and gamma rays. For the purpose of this thesis the focus will be on EMR generated by electronic devices, e.g. microprocessor, in the radio wave band. This chapter provides an introduction to the origin and properties of EMR, in particular the expected radiation from digital circuits, such as the execution sequence of a microprocessor.

### 2.2.1 Origin and Field Strength of Electromagnetic Radiation

The law of physics explained by Maxwell's equations can be stated textually as follows [14]: *Accelerating electric charges give rise to electromagnetic waves.* According to Maxwell, an electric (E) field that changes in time, will produce a magnetic (H) field that changes in time and vice versa. The interacting E- and H-fields form an electromagnetic wave that can propagate through space at the speed of light ($c$). This wave is characterized by its wavelength ($\lambda$) or frequency ($f$) given by: $c = \lambda \cdot f$. Changing fields are generated by electric charges that undergo acceleration [14].

Accelerating charges are found in virtually any operating electronic circuitry. Any circuit element, conducting changing electric current, will act as an antenna and generate EMR into the surrounding environment. An antenna is a transducer converting electric current into EMR, characterized by properties such as: resonant frequency, gain, radiation pattern, impedance, efficiency, bandwidth and polarization. These properties depend on factors such as: amount of current, length/shape and material of the circuit element. In addition, EMR will be influenced by filtering, reflection and interference from surrounding material and circuit elements [35].

The field strength of the E- or H-field at a distance $r$ from the source can be determined by the current carried by the antenna, the shape of the antenna and the radiation impedance. By considering a very short wire (dipole) carrying a sinusoidal current $I$, it can be shown that the spatially radiated wave (i.e. E- and H-field) at a point P($r, \theta, \phi$) (polar coordinates) in space can be written as [35]:

$$\begin{aligned} H_\phi &= \frac{Idze^{-j\beta r}\sin\theta}{4\pi}(j\frac{\beta}{r}+\frac{1}{r^2}) && (\frac{A}{m}) \\ E_r &= \frac{Idze^{-j\beta r}\cos\theta}{2\pi\epsilon}(\frac{1}{Ur^2}+\frac{1}{j\omega r^3}) && (\frac{V}{m}) \\ E_\theta &= \frac{Idze^{-j\beta r}\sin\theta}{4\pi}\sqrt{\frac{\mu}{\epsilon}}(j\frac{\beta}{r}+\frac{1}{r^2}+\frac{1}{j\beta r^3}) && (\frac{V}{m}) \end{aligned} \tag{2.1}$$

Three basic terms can be extracted from 2.1:

**The radiation term** , representing the flow of energy away from the wire, proportional to $1/r$.

**The induction term** , representing the energy stored in the field close to the wire, proportional to $1/r^2$.

**The quasi stationary term** , also called the electrostatic term resulting from build up of charges at the end of the wire, proportional to $1/r^3$.

These terms are equal at distance $r = \lambda/2\pi$. It is common to refer to distances, $r < \lambda/2\pi$ as the *near field* and distances $r > \lambda/2\pi$ as the *far field* . It is also common to regard the induction and the quasi stationary term negligible in the far field such that the field strength is proportional to $1/r$. In the near field, the quasi stationary term is usually ignored such that the induction term dominates and the field strength is proportional to $1/r^2$. The rate of decay of EMR (i.e. E- and H-fields) therefore depends on both the wavelength and distance from the source. This relationship is shown in Table 2.2 for some harmonics of a clock speed

Table 2.2: Relationship between frequency ($f$), wavelength ($\lambda$) and near/far field border ($r_b = \lambda/2\pi$). The rate of decay is approximately $1/r$ for $r > r_b$ and $1/r^2$ for $r < r_b$.

| frequency (f) [MHz] | Wavelength ($\lambda$) [m] | Distance to near/far field border ($r_b$) [m] |
|---|---|---|
| 4 | 75 | 12 |
| 40 | 7,5 | 1,2 |
| 400 | 0,75 | 0,12 |
| 4000 | 0,075 | 0,012 |

of 4 MHz. Notice that when measuring up to a few hundred MHz, a probe positioned closer than 12 cm will be in the near field for the entire frequency range. According to Quisquater and Samyde, the E-field carries different information than the H-field, but fails for low frequencies (below 10 MHz) [29].

### 2.2.2 Electromagnetic Spectrum from a Microprocessor

D. Agrawal et al. divide EMR from microprocessors into two broad categories: direct emanation and unintentional emanation [1]. In the following the unintentional emanation is referred to as modulated emanation , as the term unintentional is misleading. Direct emanation is also unintentional in the sense that it is an unwanted secondary effect of intended microprocessor activity.

**Direct Emanation:** This is a result of intentional current flows within a microprocessor. The CMOS inverter (Figure 2.5), is often used to explain such radiation [1, 13]. The short burst of current pulses for transitions between logic ”1” and logic ”0”, have sharp rising and falling edges due to the speed of modern microprocessors. This leads to emanation over a wide frequency band.

**Modulated Emanation** This is a result of coupling effects between components in close proximity within the microprocessor devices. Components close to each other may interfere and result in compromising information modulated onto a carrier, e.g. harmonics of the clock signal. The modulation can be AM or PM depending upon how the coupling occurs, e.g. if the clock signal and circuitry handling sensitive information draw upon the same limited power supply, the clock signal may be modulated by the sensitive information.

Furthermore, EMR can be characterized by its frequency spectrum (amplitude and phase as a function of frequency). The frequency spectrum can be found using Fourier analysis [15]. Fourier analysis is a way to represent a time domain waveform as a sum of sinusoidal signals of different frequencies. The contribution of each sinusoidal is determined by its amplitude and phase. The magnitude of these sinusoidal plotted vs. frequency, is commonly known as the frequency domain or frequency representation of a signal. Fourier analysis and frequency spectrums are especially useful in signal processing, and the transform has different names, depending upon the nature of the signal to be transformed. Representing a continuous time signal, $x(t)$, such as EMR, on a computer requires sampling to create a discrete time waveform $x(nT)$, where $n$ is an integer and $T$ is the sampling interval. In practical experiments, $x(nT)$ has a finite length of $N$ samples and can be considered a discrete signal of period $N$. The Discrete Fourier Transform (DFT) is then appropriate to use. Even though there is a clear difference between the mathematical transformation DFT and its implementation on a computer (Fast Fourier Transform - FFT), these two terms are often used interchangeably [15].

The relationship between the current (i.e. power consumption) and the EMR can be expressed by a transfer function $h(t)$ [25]. Most physical systems are not linear by nature. Finding the transfer function $h(t)$ for a non-linear system is not trivial, if possible at all. However, assuming that the system is linear time-invariant (LTI) , the frequency content of

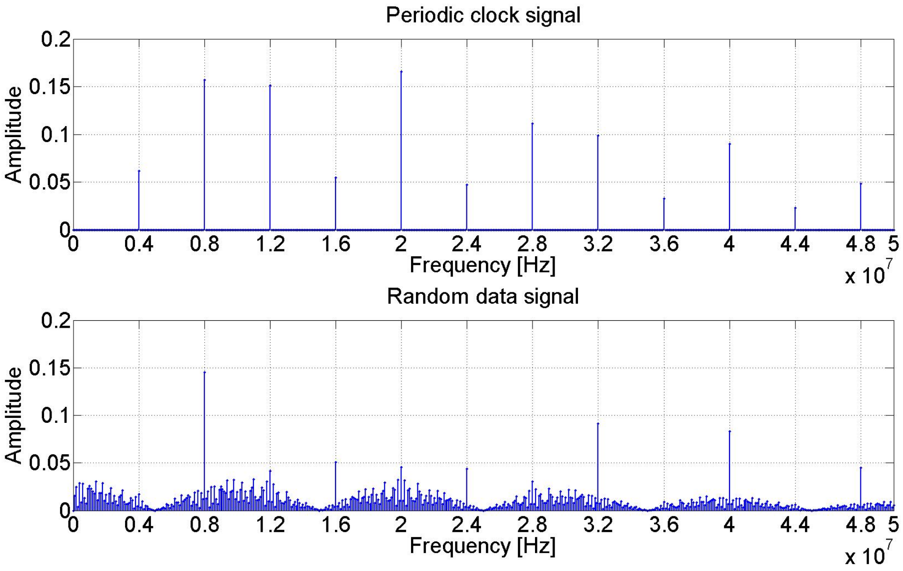

Figure 2.7: Estimated electromagnetic spectrum from microprocessor signals. Top: DFT of a periodic 4 MHz clock signal. Bottom: DFT of a random data signal synchronized by a 4 MHz clock. 40 periods used for both signals and triangular pulses used to estimate the current waveform.

the radiation will be the same as that of the current waveforms. This assumption can be used to estimate the electromagnetic spectrum of some relatively simple direct emanations. Intentional current flow within a microprocessor can be divided into two groups:

**Periodic clock/control signals** used to synchronize microprocessor activity.

**Asynchronous or random activity** , e.g. data signals.

According to [29] the periodic clock signal will result in narrow frequency bands at harmonics of the clock frequency and the random data signal will result in broadband emanation. This can be illustrated with a simple example. Consider two signals: (i) a periodic 4 MHz clock signal (typical in smart cards) and (ii) a random data signal synchronized by this clock. Since the majority of power consumption is associated with the transition between logic one and zero, the current waveform can be modeled as two triangular signals, one for the transition from $0 \rightarrow 1$ and another for the transition from $1 \rightarrow 0$. The duration of the triangle pulses is set equal to the rise and fall time specified in the data sheet for a 4 MHz smart card. Measurements of the current consumption associated with transitions in a CMOS inverter confirm this approximation [20]. The DFT of 40 periods of each of these two signals can then be calculated and the results are shown in Figure. 2.7.

The DFT of the periodic clock signal (top) show, as expected, frequency components at harmonics of the clock signal. The DFT of the random data signal, synchronized by the clock signal, still has strong narrow band components at harmonics of the clock, but broadband EMR is present as well.

### 2.2.3 Differences between Power and Electromagnetic Radiation

Sensitive information handled by a microprocessor will influence both the power consumption and the EMR. Investigation done by Rao and Rohatgi [32] shows that:

> ... although the EM side-channel superficially resembles the power side-channel in nature of information revealed, there are instances and situations where the EM side-channel can carry much more useful information.

Without invasive measures, power analysis can only be applied globally, by measuring the sum of all individual power consumptions in the microprocessor. This provides a 2-dimensional power trace, power vs. time. With access to the device and a sufficiently small probe, careful positioning of a single probe (using a stepping table) or multiple probes can in addition, provide spacial information [29, 28]. A 3-dimensional map of the radiation (x,y and EMR field strength) can then be built [29]. 4-dimensional information can be captured by building these 3D pictures over time [28]. This can be used to compare and analyze EMR from individual components within the microprocessor as done by Gandolfi et al. in [13]. Documentation of the use of all three spacial axis $(x, y, z)$ has not been found at this time.

Even though the most efficient method to capture radiated signals is to place a probe in the near field, as close as possible to the microprocessor [1], these signals can also be captured from a greater distance. In addition, modulated signals, such as harmonics of the clock signals, can potentially be picked up by an AM/PM receiver at considerable distances, 15 feet was reported in [1]. Using EMR, therefore opens up for remote measurements, without physical access to the circuit. It is then possible to launch an attack without the user being alerted.

In [32], Rao and Rohatgi show that EMR is at least as powerful as power analysis, and that in some cases, even more information is available. They report some "bad" instructions that leak much more information through EMR than power consumption. In [1], Agrawal et al. show that EMR consists of multiple signal, often leaking different information. Exploiting multiple side-channels (power and EMR) to improve side-channel attacks was suggested by Agrawal et al. in [3].

## 2.3 Capturing Electromagnetic Radiation: Measurement Setups

The purpose of this chapter is to provide a basic description of measurement setups used to capture EMR from microprocessor devices and the typical components involved. Results from an early feasibility study are used as practical examples.

### 2.3.1 General Overview

A typical measurement setup (Figure 2.8) consists of: source, antenna, analogue preprocessing, capturing device and digital post processing. The source, e.g. a microprocessor smart card, should execute the intended program in a desired environment at a specific time to generate EMR. Unwanted influence (i.e. noise) from internal or external sources should be limited. The antenna have to capture either the E- or H-field in a specific frequency range either remotely or from a precise location as close as possible to the device. Analogue preprocessing can enhance the signal through filtering, amplification, mixing or demodulation. Some analogue preprocessing is always found in oscilloscopes and spectrum analyzers. For weak signals it is recommended to use a sensitive radio receiver. Active antennas can also be used to amplify the signal. Simple analysis can be done directly on instruments, but analogue-to-digital (A/D) conversion by a capturing device makes it possible to perform off-line digital post processing on computers with programs such as MATLAB, Octave or LabVIEW. Notice that the signal can be sampled anywhere in the analogue preprocessing chain.

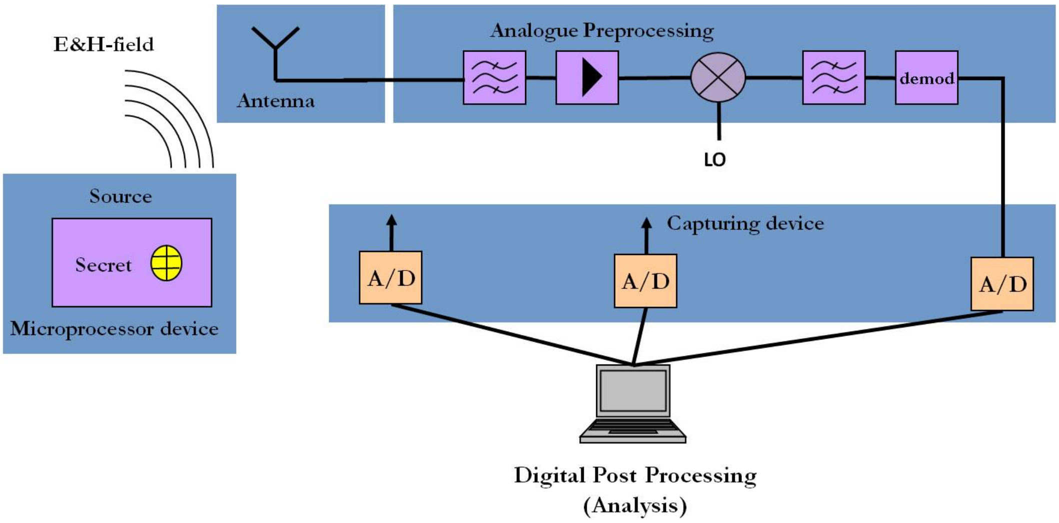

Figure 2.8: Schematics of the measurement setup.

### 2.3.2 The source

A number of test kits are available for different microprocessor architectures, such as Atmel's AVR [6] and Microchip's PIC [21]. These evaluation kits provide an easy start-up as they are relatively inexpensive and easily available. Only power is required, as the clock is generated on the circuit board. Communication to a PC is usually provided through an RS-232 or Ethernet connection. Programming and executing code on the device can be done through standard SW-packages provided by vendors on-line (e.g. MPLAB®IDE by Microchip [21]). Guidelines and help for beginners can also be found on-line. One drawback with evaluation boards is a lot of circuitry surrounding the chip that may cause unwanted disturbances.

Advanced microprocessor smart cards are harder to get hold of. They are commonly used in high-security applications and therefore, undergo stricter controls. Non-disclosure agreements are often necessary to get access to the latest technology, however, simple cards (e.g. gold card - PIC 16F84A) are available without such agreements. A smart card reader is necessary to program and use the card. These readers come with the same challenges as the microcontroller evaluation kits, concerning how to control unwanted disturbances from the circuitry. One solution, provided by this thesis, is to customize a reader . If the card is programmed in a traditional terminal, a customized reader only has to provide power and clock signal (Figure 2.9), greatly reducing the number of sources of unwanted disturbances.

When targeting real-world systems, measurement setup should focus on replicating the system, its operation and the environment as closely as possible to its actual use. However, when the objective is to provide better understanding of the relationship between microprocessor activity and EMR, it is more important to fully understand (and be able to control) every minute activity that takes place. It is desirable, however unlikely, to control the transition of individual transistor, but it is possible to control the microprocessor activity at a bit-level (e.g. execution sequences). One of the major challenges is to minimize influence from unwanted and irrelevant simultaneous activities, $s_{noise}$ in Figure 2.1. This can be addressed through careful assembly programming. The execution sequence can be manipulated to provide desired transition patterns (e.g. bus transfers) in most parts of the microprocessor. Limitations are dictated by the control and flexibility allowed by the instruction set.

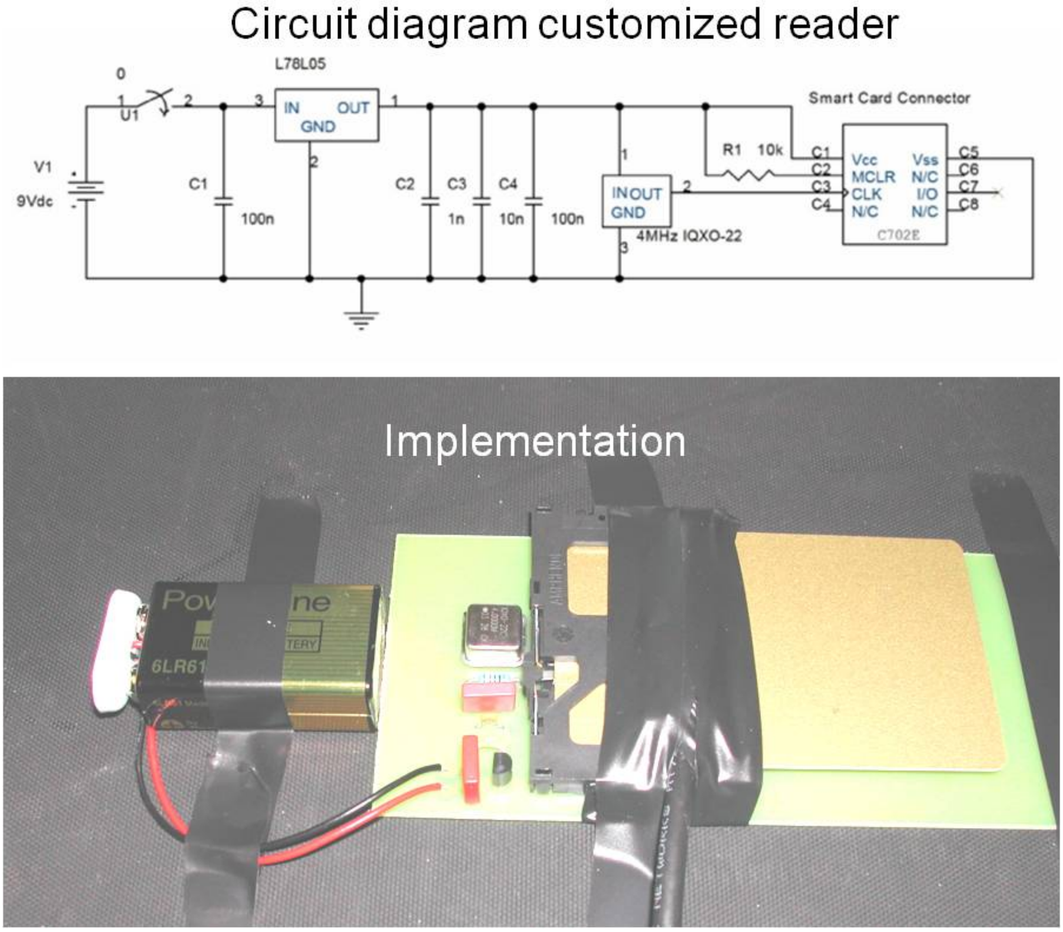

Figure 2.9: Customized smart card reader, circuit diagram and implementation.

All programs used in this thesis follow the same basic structure. The programs are written in assembly language off-line, using vendors development kits, and loaded to smart cards with a standard smart card terminal. When a programmed card is inserted into the customized reader, power and clock signal are provided and the microprocessor automatically executes from the beginning of the program. First, the I/O is toggled. This creates a relatively strong radiation compared to EMR from internal activity, and is used as a trigger point. The trigger point is essential for off-line alignment between captured EMR and the executed code. Next, appropriate instructions are used to generate the desired activity, e.g. transfer of a specific bit pattern on the data bus. The no operation (NOP) instruction is frequently used as a buffer between activities. Careful choice of instructions often allows the desired activity to take place in "quite clock cycles", Q2 and Q3 (Figure 2.4) in which parallel activities are at a minimum.

Finally, the program is repeated indefinitely. This makes it possible to trigger an oscilloscope on the I/O toggle, and fine tune settings manually, e.g. optimize the oscilloscope's resolution to a specific point in the execution sequence. Running the program indefinitely also facilitates capturing multiple observation, as long as the time between captures is kept larger than the time it takes to execute the entire program. Repetitive activity is also easy to study with an spectrum analyzer, which can be useful to identify carriers and potential useful EMR [1].

```
     Test Code
     ;Main program
     Start
     ;Trigger Turn I/O ON and OFF
1    movlw 80h        ; Turn I/O ON
2    movwf PORTB      ; by moving the value 80h onto port B
3    movlw 00h        ; Turn I/O OFF
4    movwf PORTB      ; by moving the value 00h onto port B
     ;10 NOP's to create buffer from I/O disturbances
5    nop
.
.
.
14   nop
     ; Transition: a:0000 0000 - b: 0001 1111
15   movlw 00h        ; a into W register
16   movwf DATA1      ; mov a from W to DATA1 register
17   movlw 1Fh        ; (b-a) into W register
18   addwf DATA1,1    ; Q2 read a, Q4 write b=(a+(b-a))
     ; Transition: a:0000 0000 - b: 1000 1111
19   movlw 00h        ; a into W register
20   movwf DATA1      ; mov a from W to DATA1 register
21   movlw 8Fh        ; (b-a) into W register
22   addwf DATA1,1    ; Q2 read a, Q4 write b=(a+(b-a))
     ; Continue for other transition patterns
.
.
.
     ; Back to the start of the program
23   goto Start
```

Table 2.3: Example of test code for PIC 16F84A.

The code in Table 2.3 was used in [11] and illustrates some of the challenges faced when designing test code to dictate microprocessor activity. The code is written for PIC 16F84A found in Goldcard smart cards. The objective of the code is to create a transition between value $a$ and value $b$ on the microprocessor's internal 8 bit data bus and minimizing irrelevant activity. It is essential that the code is designed such that there is no data bus activity taking place between value $a$ and $b$, to ensure validity of the result. It is also essential that the power consumption (i.e. EMR) is correlated with this bus activity and not dominated by noise (e.g. other irrelevant microprocessor activities due to pipelining).

Code lines $1-4$ toggles the smart cards I/O and provides a trigger-point for the oscilloscope. The following 10 NOP's create a buffer between electromagnetic disturbances caused by the relatively strong I/O toggle and the rest of the program. Code lines $15-18$ are used to create a transition from bit pattern 00000000 to 00011111 ($T_5^1$). Code lines $15-17$ are initialization, making sure value $a$ is available in DATA1 register and value $(b-a)$ is found in the working register. Transition between value $a$ and $b$ is then made possible by the ADDWF instruction of line 18. In clock cycle 2 (Q2) the value $a$ is read over the data bus, in clock cycle 3 (Q3), $a$ is added to $(b-a)$ found in the working register. The result, $b$, is written back over the databus in clock cycle 4 (Q4), creating the desired transition without unwanted data bus activity. The process can now be repeated for all other values of $a$ and $b$, as shown in code lines 19-22. Finally code line 23 repeats the program indefinitely. Notice that if the results of processing the add operation in Q3 also make the result valid on the databus in Q3, the transition will be found one clock cycle earlier. This is easily detectable by digital post processing.

### 2.3.3 The Antenna

An antenna (or probe) is a transducer that converts between electric currents and electromagnetic waves and vice versa. The antenna produces a signal (current or voltage) proportional to the field strength of either the magnetic or the electric component of the incoming wave. The choice of antenna will depend on factors such as; distance to the source, type of radiation (E- or H-field), frequency range and signal strength. Antennas are designed for a specific frequency range. They may work outside this range, but with significantly lower sensitivity. When the frequency range is large, this implies that it may be necessary to use more than one antenna to ensure best possible signal-to-noise ratio.

Published work on electromagnetic side-channel attacks [13, 29] often use small antennas that can be accurately placed. Sometimes the approach is semi-invasive, as they remove some capsulation to get the antenna as close as possible. The regions of the chip that radiate the most can then be isolated (CPU, data buses and power lines), allowing 4 dimensional (spatial and time) traces of the EMR to be made [13]. Many different types of antennas used for this purpose have been tested; such as hard disk heads, integrated inductors and magnetic loops. The design of the antenna will have an impact on sensitivity, frequency range, bandwidth and linearity. According to Gandolfi et al. [13], a simple, hand-made antenna (i.e. solenoids) can give adequate results (Figure 2.10). Such antennas are broadband, but frequency selectivity can be enhanced at the expense of sensitivity. For EMR where the spectrum is unknown, broadband antennas are preferable [13].

Commercial probes used in traditional EMC testing will also work well, but are usually expensive. An example is the Probe set HZ-11 from Rohde & Schwarz seen in Figure 2.11. This set provides 3 magnetic and 2 electric passive near field probes , covering from 100 kHz to 2 Ghz. The set also comes with an amplifier providing in excess of 30 dB gain. In the far field other antennas are suitable, such as Rohde & Schwarz broadband directional antenna (Ultralog HL562) or high sensitive active antenna (AM 524) seen in Figure 2.11 .

### 2.3.4 Analogue Preprocessing and Capturing Devices

The purpose of the capturing device is to take the analogue signal from the antenna, select the desired frequency range, apply preprocessing (e.g. amplification, filtering, demodulation/rectifying) and convert the signal into a digital representation that can be stored and

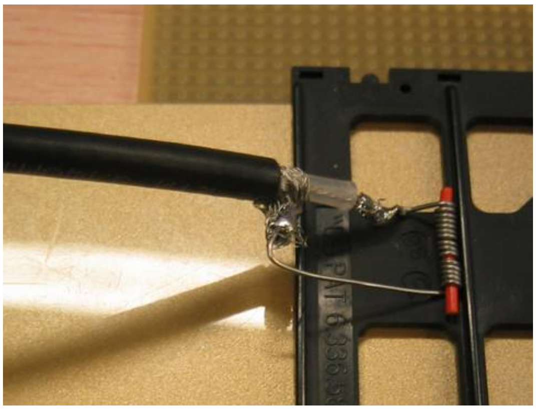

Figure 2.10: Solenoid probe.

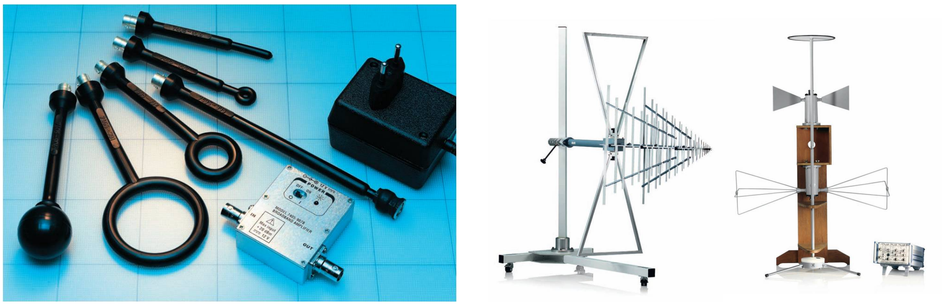

Figure 2.11: Rohde & Schwarz probe set for E and H near field emission (left). Rohde & Schwarz far field antennas, HL 561 and AM 524 (right). Pictures reproduced with permission from Rohde & Schwarz.

analyzed on a computer. The EMR from a microprocessor is found in a wide frequency range. It is therefore essential that the device can handle the frequency range of interest. Some other key features are:

- Input Bandwidth : The bandwidth determine the maximum frequency component that can be processed without distortion [20]. For modern oscilloscopes with bandwidths of several 100 MHz (GHz for high-end scopes) this is usually not a limiting factor.
- Sampling rate : This determines how many times per second the analogue signal is sampled and digitized. In order to avoid loss of information the sampling rate must be at least twice as high as the highest frequency component of interest (Nyquist sampling theorem [26]).
- Resolution : A fixed number of bits are used during quantization of each sample. This is often 8 bit. Each sample must then be assigned to one of 256 values. This introduces quantization noise, which can be a challenge if you are looking at minute differences on a relatively strong signal.
- Memory : The size of the devices memory will limit the length of the observation that can be captured. A trigger is therefore necessary, such that recording can be focused around interesting activities and not wasted on idle time.
- Interface : For large-scale testing, being able to automatically change setting on the instruments from a PC and automatically transfer data to a PC is essential, e.g. via an ethernet connection.
- Trigger : This is used to synchronize data capture with the relevant microprocessor activity. This can save valuable memory space, both in the instrument and on the PC, and is important during analysis when multiple observations have to be aligned.

Two different capturing devices were used for this thesis; spectrum analyzer and oscilloscope.

### 2.3.5 Frequency Domain Measurements with Spectrum Analyzer

A spectrum analyzer provides a frequency domain presentation of data (frequency vs. amplitude) by working as a tuned receiver. It will move through the specified frequency range in steps determined by the resolution bandwidth (RBW) . For each step it will measure received power in the specified bandwidth. A smaller RBW will give better frequency resolution, but will make the sweep time (the time it takes to sweep through the entire range once) longer. The result is a discrete power density spectrum (PDS) for a given frequency range. It is important to realize that all spectrum analyzers have a limited number of measurement points to represent the PDS. For example, Advantest 3641 has only 701 measurement points per frequency range. A way around this, when better resolution is needed, is to divide the frequency range into smaller sub-ranges and repeat the measurements.

Due to the sweeping operation, spectrum analyzers are best suited for repetitive signals and continuous EMR. Burst activity will not be detected unless the spectrum analyzer happens to measure in that particular frequency range when the burst occurs. The spectrum analyzer can give valuable information about the frequency spectrum of microprocessor activity, but is unable to pinpoint exactly what activity caused the various frequency components, as the timing information (when activities takes place) is not preserved. However, spectrum analyzers are useful for quickly detecting potential useful carriers and radiation [1].

The information available from a spectrum analyzer can be illustrated by designing a program with several periodic elements. Let a program executes one instruction (COM)

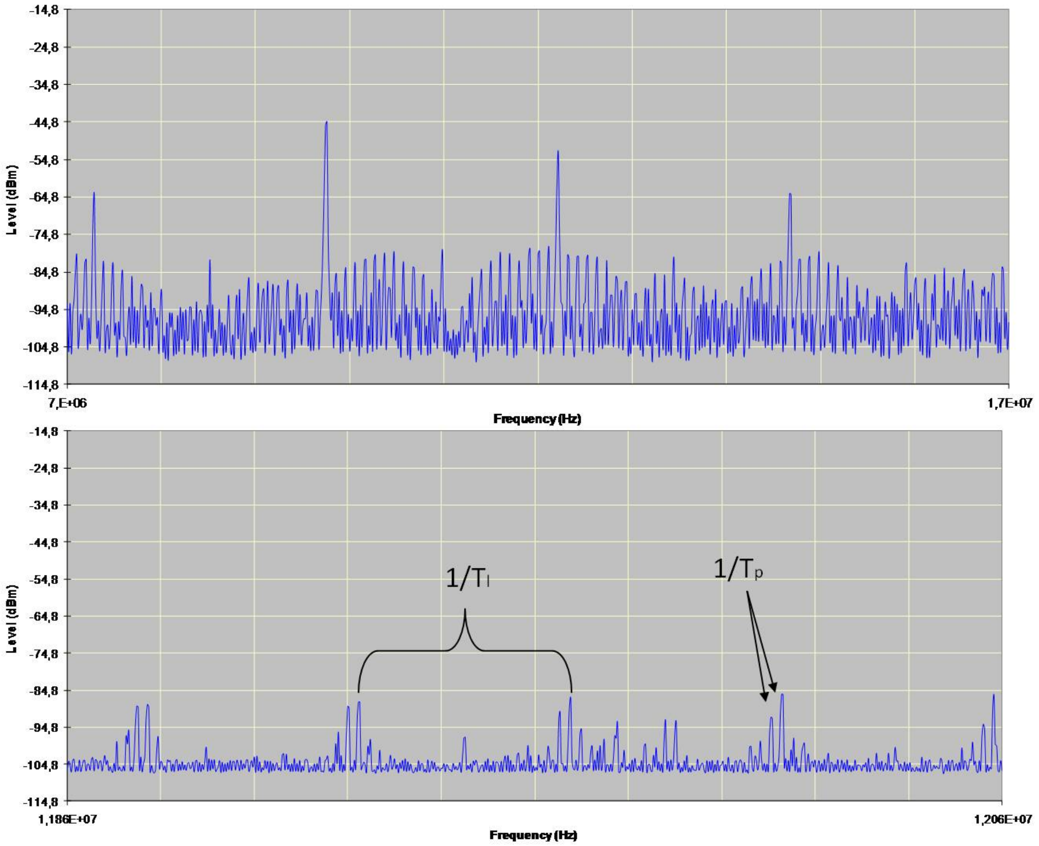

Figure 2.12: Top: EMR at harmonics of the clock as well as broadband radiation. Bottom: Identifying periodicity of period $T$ in the time domain as discretization with spacing $1/T$ in the frequency domain. $T_p$ is the time it takes to execute the entire program. $T_l$ is the time it takes to execute one instruction including the delay.

followed by 10 no operation (NOP) instructions as a delay. Then the same instruction is executed again, followed by another delay. This is repeated 20 times before a jump (JMP) instruction repeats the program indefinitely. This program has several periodic elements, such as the time it takes to execute the entire program ($T_p$) and the time it takes to execute one instruction including the delay ($T_l$). Two PDS's of this program running on a Motorola microcontroller test kit is shown in Figure 2.12 . Measurements were taken with the Advantest 3641 spectrum analyzer, with the customized solenoid probe (Figure 2.10) directly coupled. The top PDS show EMR at harmonics of the clock as well as broadband radiation. This is in accordance with theory from section 2.2, since the EMR is generated by a data signal (execution of a program). An important property of DFT is that: periodicity with period $T$ in time domain automatically implies discretization with spacing $1/T$ in the frequency domain [26]. This can be seen in the bottom PDS as frequency spacings equal to $1/T_p$ and $1/T_l$.

### 2.3.6 Time Domain Measurements with Oscilloscope

Oscilloscopes are by far the most common instrument to capture side-channel information. They provide a time domain representation (time vs. amplitude) of the EMR by performing A/D conversion [26]. When the signal is weak, the output from the antenna may be too low

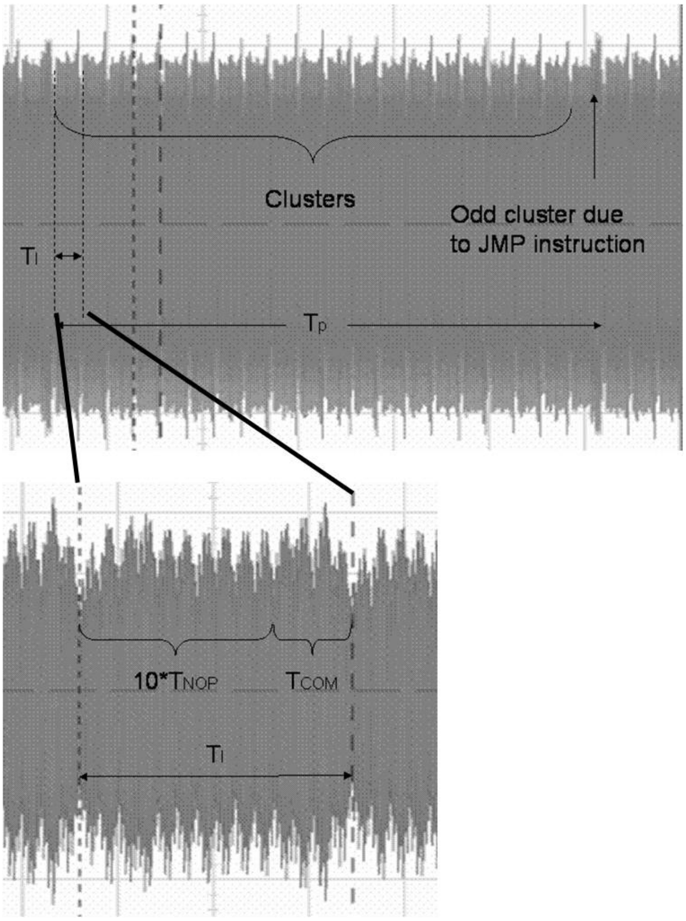

Figure 2.13: Time domain measurements of repetitive program: COM and 10 NOP's, JMP instruction every 20th execution.

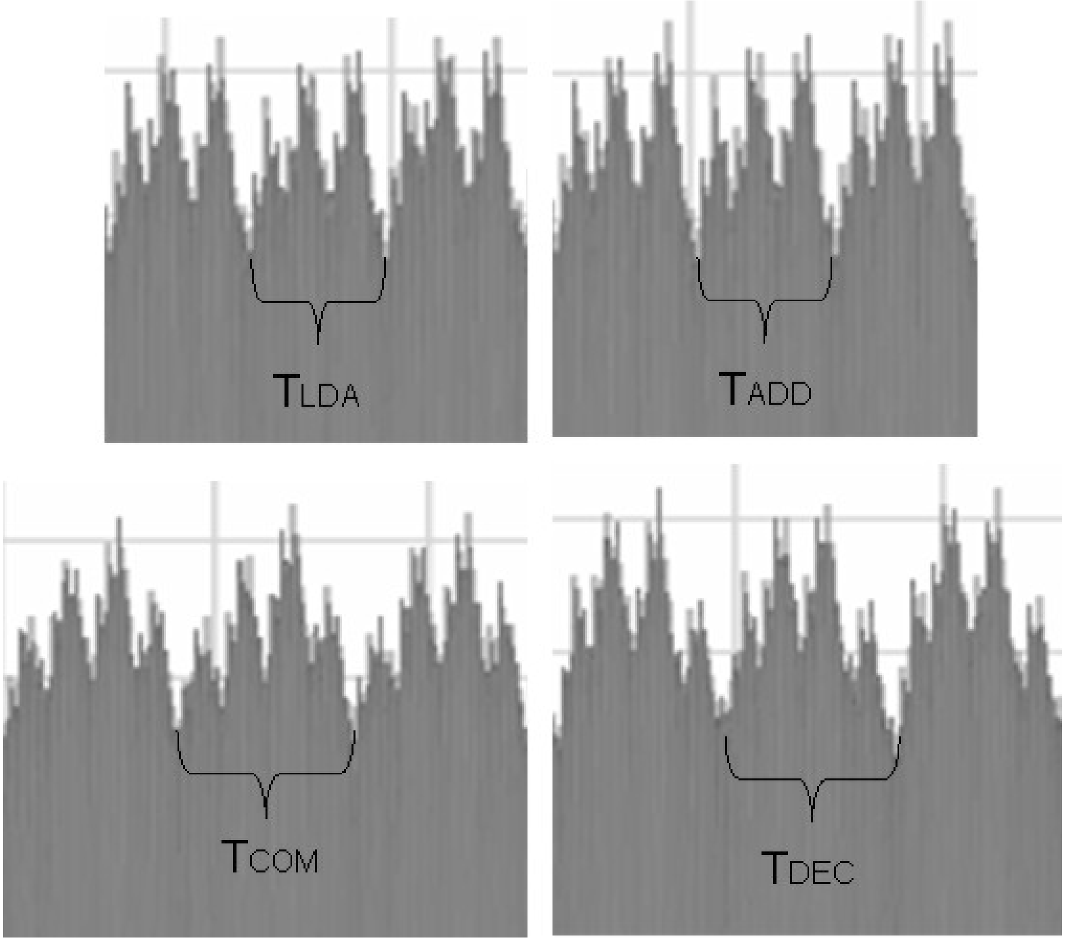

Figure 2.14: Comparison of LDA, DEC, ADD and COM instruction.

for the dynamic range for direct connection to an oscilloscope. In such cases, amplification is necessary, e.g. by using an active antenna or a sensitive receiver. A receiver was not used in this thesis, but an introduction to such devices can be found in [19].

Figure 2.13 shows the time domain measurement of the same program used with the spectrum analyzer. Most easily recognizable are clusters of peaks that periodically look slightly different (top). Each cluster represents the execution of one COM instruction and 10 NOP's (marked $T_l$). Every 20th cluster is different, as after $20$ repetitions, there is a JMP instruction to repeat the program indefinitely. The execution of the whole program, $T_p$, is also visible and marked. Zooming in on one cluster gives us additional information (Fig 2.13). Each cluster is built up of a number of peaks, each representing one internal clock cycle. Counting the number of peaks in one cluster reveals the 4 clock cycles for the COM instruction and the 10 clock cycles for the 10 NOP's requires to complete $T_l$.

It is also relatively easy to show that different instructions and even different operand generate different EMR. Figure 2.14 shows the PDS of execution of each of the four instructions LDA, ADD, COM and DEC. LDA and ADD can easily be distinguished from COM and DEC as they have different length execution cycles. It is more difficult to distinguish LDA from ADD based on visual comparison alone. Notice that in other microprocessor architectures (e.g. PIC) all instructions are executed using the same number of clock cycles. The number of clock cycles alone is, therefore, not a good classification feature. Using the

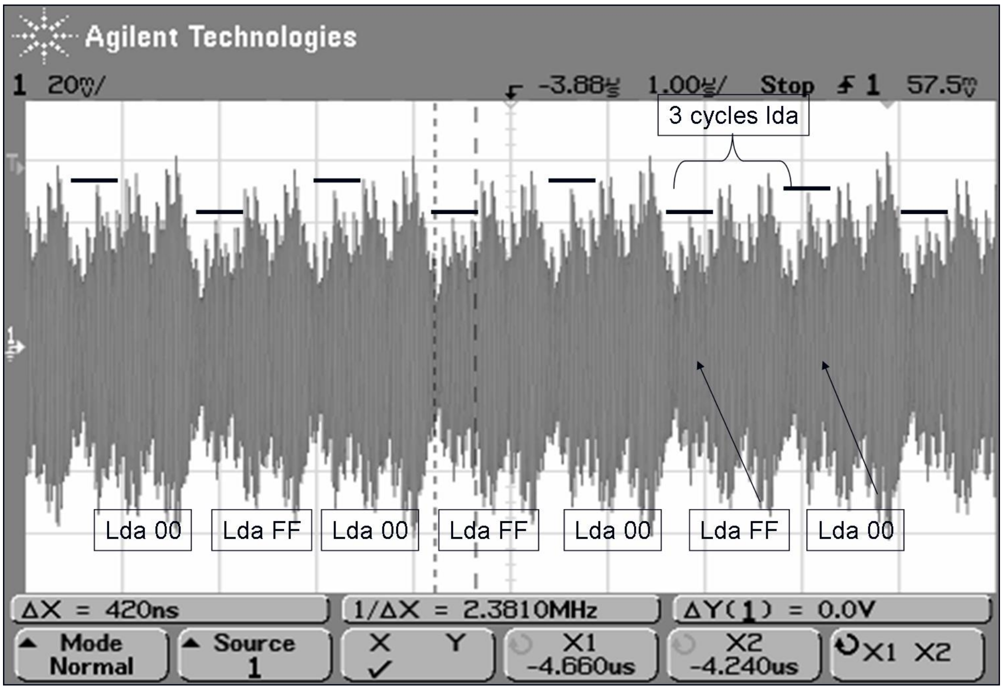

Figure 2.15: Each three-cycle instruction can easily be identified and the third peak (right to left ) clearly depends on the argument (LDA 00-high, LDA FF-low).

amplitude of each clock cycle can provide additional information.

To illustrate the potential of using amplitude as a classification feature, Figure 2.15 shows 8 executions of the instruction LDA with alternating argument (FF and 00). Each instruction is easily identified by three clock clock cycles. In addition, the third peak (from right to left) of each instruction is clearly correlated to the operand being FF or 00. LDA 00 gives a high third peak and LDA FF gives a low third peak.

These examples, using Motorola test kit, show that it is relatively easy to find correlations between EMR and the program running, instruction executed or even operand used, by visual inspection alone. These visible differences and the fact that statistical tools can extract much more information than the naked eye, form the foundation for some of the analysis techniques described next.

## 2.4 Analysis/Application - Side-Channel Attacks

According to [20], passive non-invasive attacks on cryptographic devices (e.g. smart cards) are referred to as *side-channel attacks* . Timing attacks [16] measuring execution time, power analysis attacks [17] measuring power consumption and electromagnetic attacks [13, 29] measuring EMR are the most important types of side-channel attacks. More generally, the process of reverse engineering microprocessor content based on features of the intercepted side-channel (e.g. EMR) can be viewed as pattern classification [9], in which a range of tools are available. The purpose of this chapter is to introduce some of the relevant analysis techniques in which the contribution of this thesis is based upon.

### 2.4.1 Power Analysis

Attacks exploiting correlations between power consumption and internal sensitive information handled by a device were first introduced by Kocher et al. in simple power analysis (SPA) and differential power analysis (DPA) [17]. The principle behind SPA and DPA are briefly outlined here, but for an extensive coverage see the textbook ”Power Analysis Attacks” by Mangard et al. [20] .

#### 2.4.1.1 Simple Power Analysis (SPA)

SPA [17] relies on differences observed directly from the captured power traces. Usually, these difference are visible for direct interpretation, however, simple statistics on single (or a few) traces can also be used to express the immediate differences. SPA can identify rounds (e.g. DES) and operations (e.g. RSA multiply and square) of cryptographic algorithms. Instruction sequences may also be revealed by a closer look at the trace. Cryptographic algorithms are vulnerable to SPA when the execution sequence depends on the secret key. SPA is particularly useful when only a few traces are available or as a starting point for other analysis techniques, by identify rounds (e.g. 16 rounds of DES [17]) or alignment etc [20].

#### 2.4.1.2 Differential Power Analysis (DPA)

If no apparent differences between traces are detectable, either by simple statistical properties or visually, there may still be statistical differences that can be detected by looking at a large number of traces. This is exactly what DPA proposed by Kocher et al. [17] exploits. They suggest using statistical properties of a large number of power traces to extract small power variations correlated to secret parameters handled by the device. The secret parameter can, for example, be intermediate values in a cryptographic algorithm. With detailed knowledge of the cryptographic algorithm it may be possible to deduce the secret key if a sufficient number of the intermediate values are revealed. Therefore, repeating the attack to reveal enough intermediate values can eventually expose the key. This attack has proved itself in practice. Kocher et al. had in 1999 already used power analysis to extract keys from almost 50 different products [17]. DPA can be divided into three phases:

**Power Prediction:** Predict the power consumption of a secret parameters activity performed by the microprocessor. This activity can e.g. be the output of the AES SubByte function or the output of an S-box in DES. The predicted power consumption is based on known plain text or known cipher text and all possible values of the secret parameter.

**Power Measurements:** Measure (i.e. sample) the ”real” power consumption of the secret parameter-dependent activity for a large number (usually more than 1000) of encryption operations (known plain text or cipher text). Synchronization between the microprocessor activity and measurements is vital. Without such alignment, averaging many traces would remove any dependencies.

**Comparison** Compare the measured power consumption with the predicted power consumption for all variations of the secret parameter. When the predicted power consumption is based on the correct secret parameter, it will be strongly correlated to the ”true” power consumption for all the measured power traces. Otherwise there will be no (i.e. very small due to noise) correlation between the predicted and measured power consumption.

### 2.4.2 Template Attack - Classification Using Bayesian Methods

The Template attack was first presented by Chari et al. at CHES 2002 [8]. Principles from signal detection and estimation, where extraction of a very weak signal is possible if the receiver has a good characterization of the signal and the ambient noise, were motivations for this attack. The template attack first builds fingerprints (templates) of relevant activity on a microprocessor. The fingerprints contain information about what signal to expect and noise characteristics for each activity. This information is based on captured side-channel information from the device. The activity can for example be execution of a cryptographic algorithm for different values of the secret key. When a trace from a device under attack is received, it is compared to all the fingerprints. The goal is to classify the unknown trace to one of the fingerprints and thus determine what activity took place. The Template Attack is Bayesian Classification [9], where the noise is assumed to follow a Gaussian distribution. Bayesian classification is used extensively in this thesis and is therefore explained in some detail.

Consider a finite set of classes, $\{\omega_1, \cdots, \omega_c\}$, e.g. execution of $c$ different instructions on a microprocessor. $P(\omega_i)$ is the *prior* probability of class $\omega_i$ occurring. Measurements of any class, result in observations of $d$ variables held in a d-dimensional vector $\vec{x}$, e.g. $d$ samples with an oscilloscope of the electric field strength associated with execution of a an instruction on a microprocessor. Let $p(\vec{x}|\omega_i)$ be the *a priori* class conditional density of $\vec{x}$ being observed given that class $\omega_i$ was measured. Faced with an observation $\vec{x}$ of unknown class, classification is interested in the probability of this observation belonging to the different classes, $\omega_i$. This is expressed by the *posteriori* probability $P(\omega_i|\vec{x})$ which can be calculated using Bayes rule :

$$P(\omega_i|\vec{x}) = \frac{p(\vec{x}|\omega_i)P(\omega_i)}{p(\vec{x})} \tag{2.2}$$

where $p(\vec{x})$ the total probability is given by $p(\vec{x}) = \sum_{i=1}^{c} p(\vec{x}|\omega_i)P(\omega_i)$. Given measurements $\vec{x}$ of unknown class, the question is: how to decide what class to assign $\vec{x}$ to? Generally a decision must be made based on the measurements $\vec{x}$ and a loss function , $\lambda_{ij}$. Let $\lambda_{ij}$ be the loss if the decision is $\omega_i$ when the true class is $\omega_j$. In the following, a zero-one loss function is used:

$$\lambda_{ij} = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \quad i, j = 1, \cdots, c \tag{2.3}$$

A correct decision gives no loss, and a wrong decision gives a unity loss. Bayes decision rule can now be stated as: decide in favor of the class that minimizes the probability of error. For the zero-one loss function this will be the class with the maximum a posteriori probability $P(\omega_i|\vec{x})$.

$$Choose \quad \omega_i \; if \; P(\omega_i|\vec{x}) \geq P(\omega_j|\vec{x}) \quad for \; all \; j \in 1, \cdots, c \tag{2.4}$$

A classifier is often represented by a discriminating function , $g_i(\vec{x}), i \in 1, \cdots, c$, where $g_i(\vec{x})$ is such that $\omega_i$ is chosen if

$$g_i(\vec{x}) \geq g_j(\vec{x}) \quad for \; all \; j \in 1, \cdots, c \tag{2.5}$$

The discriminating functions divide the feature space into decision regions, separated by decision boundaries [9]. It is often useful to realize that any monotonically increasing function can be applied to $g_i$ without affecting the classification result. For Bayes decision rule (equation 2.4) this means that any of the following discriminating functions yield the same result:

$$g_i(\vec{x}) = P(\omega_i|\vec{x}) \tag{2.6}$$

$$g_i(\vec{x}) = p(\vec{x}|\omega_i)P(\omega_i) \tag{2.7}$$

$$g_i(\vec{x}) = ln \; p(\vec{x}|\omega_i) + ln \; P(\omega_i) \tag{2.8}$$

As seen in equation 2.8 the discriminating function $g_i$ for Bayes classifier is determined by the *prior* probability $P(\omega_i)$ and the *prior* conditional density function $p(\vec{x}|\omega_i)$. $P(\omega_i)$ is usually known or easily calculated, however $p(\vec{x}|\omega_i)$ must be estimated based on available measurements of known class. This is called training the classifier. $p(\vec{x}|\omega_i)$ will have a signal part and a noise part. The Template Attack [8] assumes the noise to follow a multivariate Gaussian distributions , $p(\vec{x}|\omega_i) \sim N(M_i, \Sigma_i)$ for the noise, such that:

$$p(\vec{x}|\omega_i) = \frac{1}{\sqrt{2\pi}^d |\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\vec{x}-M_i)^T \Sigma_i^{-1}(\vec{x}-M_i)} \tag{2.9}$$

where $M_i$ is the mean vector, $\Sigma_i$ the covariance matrix and T denotes transposition. Using the discriminating function from 2.8 and the Gaussian multivariate assumption 2.9 gives:

$$g_i(\vec{x}) = -\frac{1}{2}(\vec{x} - M_i)^T \Sigma_i^{-1}(\vec{x} - M_i) - \frac{d}{2} ln 2\pi - \frac{1}{2} ln|\Sigma_i| + lnP(\omega_i) \tag{2.10}$$

Expressed as a quadratic function for the case of arbitrary covariance matrixes $\Sigma_i$ gives:

$$\begin{aligned} g_i(\vec{x}) &= \vec{x}^T A_i \vec{x} + \vec{x}^T B_i + C_i \quad where && (2.11) \\ A_i &= -\frac{1}{2}\Sigma_i^{-1} \\ B_i &= \Sigma_i^{-1} M_i \\ C_i &= -\frac{1}{2} M_i^T \Sigma_i^{-1} M_i - \frac{1}{2} ln|\Sigma_i| + lnP(\omega_i) \end{aligned}$$

where $x^T$ and $M^T$ is the transposed of $x$ and $M$ respectively. The coefficients of the discriminating function, $A_i, B_i$ and $C_i$ for the quadratic classifier can be pre-computed using a set of training data with known class. Training data is obtained using an identical device prior to the attack. In the Template Attack [8] maximum likelihood estimates (MLE) for $M_i$ and $\Sigma_i$ are used, given by:

$$\hat{M}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.12}$$

$$\hat{\Sigma}_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{M})(x_i - \hat{M})^T \tag{2.13}$$

The tuples ($\hat{M}_{MLE}$,$\hat{\Sigma}_{MLE}$) are the templates for each activity of interest. Upon receiving an observation of unknown activity, classification can be done according to equation 2.5. Once the coefficient of the discriminating function (equation 2.11) has been estimated, the real attack can be launched, but often the questions is: How good is the classifier? Usually a limited set of data is available for both training and testing. By reserving a portion of the data for testing (leave out k [9]), the accuracy can be found as the ratio of correct classification to the total number of classifications performed (based on a confusion matrix). It is important not to use the same data for training and testing. The accuracy will vary, depending on how the data is split. It is therefore common to randomly split data into training and testing sets several times, and use average accuracy as the performance for the classifier [9]. Classifying more than two classes is achieved in two ways, (i) pairwise manner or (ii) Bayes method [9]. Both produce the same result, since classification is based on maximizing the posteriori probability. Pairwise classification compares two classes and compares the third class with the result of the first comparison etc. Bayes method calculates the quadratic function for all classes and chooses the maximum result.

### 2.4.3 Practical Considerations and Feature Selection

Any practical attack based on pattern recognition, must overcome two challenges:

1. How to deal with a large number of classes.
2. How to deal with a large number of variables in each observation.

The number of classes depend upon the attack scenario. Reverse engineering the executed instruction requires one class for each possible instruction, in the range of 30-150 depending on the size of the microprocessor's instruction set. Recovering cryptographic keys requires one class for every variation of the key. It is not feasible to build that many templates. The "Extend-and-prune" technique as described in [8] is an iterative approach to this problem. Furthermore it is impractical to use all $d$ variables of each trace, as $d$ is often in the order of $10^5$. The size of the covariance matrix grows quadratic with $d$ and according to the Ugly Duckling Theorem it cannot be expected that more features results in better classification [9]. Therefore, the challenge is to find a lower dimensional representation that still contains the relevant information, through subsets or mapping of the original trace [12] .

Chari et al. [8] used pairwise difference between the mean observations to select variables with a large difference in amplitude. Rechberg and Oswald [33] suggests a "sum of differences" approach. Compressing the trace is described by Mangard et al. [20]. They suggest two methods:

**Maximum Extraction** The power from a device, during one clock cycle, is represented by the maximum peak within the clock cycle.

**Integration** The power is represented by integrating the power trace in a given time interval (window). The window size is usually chosen equal or smaller than one clock cycle. The integration in a time interval can be performed in different ways: (i) sum of all points in the window, (ii) sum of absolute values or (iii) sum of squares.

Simulations conducted [20] show that the peaks of the power traces are the most relevant points, therefore it is assumed that most of the side-channel information is preserved in these compression techniques. Preprocessing can also be done to enhance the classification accuracy. Rechberger and Oswald [33] show that using discrete Fourier transform (DFT) on all traces reduces the number of relevant points needed and performance of the classification is improved. The template attack assumes that time instances having a great variability are considered important to discriminate. Archambeau et al. [5] suggests using linear combinations of samples. Under the assumption that variability is important, principle component analysis (PCA) is the optimal linear transformation. In spite of these attempts, trial and error is still often the way to find the best number of points to include. Open issues are therefore still [33]: How to select the relevant samples (variables) and how many samples are needed to construct the attack?

### 2.4.4 Subversion

At CHES 2007, Pankaj Rohatgi from IBM T. J. Watson Research Center, talked about trustworthy hardware [34]. For devices with high levels of security requirements, he underlined the importance of considering the complete life cycle of the device. New challenges, requirements and attack scenarios will be found during design, building/testing, deployment, maintenance and decommissioning the device.

One of these challenges is subversion, which according to Anderson et al. [4], is a forgotten or ignored area of information security. Subversion is the attack preferred by professional attackers and common security techniques are shown to be useless [4]. Myers [24] describes subversion as :

> The subversion of computer systems is the covert and methodical undermining of internal and external controls over a systems lifetime to allow unauthorized and undetected access to system resources and/or information.

Myers [24] continues to characterize subversion as:

1. It can occur at any time during the life cycle of computer systems
2. It is under control of highly skilled individuals
3. It utilizes clandestine mechanisms called artifices deliberately constructed and inserted into a computer system to circumvent normal control or protection features

Anderson et al. [4] point out, ”...subversion threat touches the most fundamental aspect of the security problem: proving the absence of a malicious artifice.” In order for subversion to be successful, three factors must be present [4]:

**Means** - The adversary must possess the physical and intellectual means to develop subversion.

**Motive** - The adversary must have motivation for carrying out the attack.

**Opportunity** - The adversary must have access to the system at some point in its life cycle.

Several examples are mentioned in [4] that show that the subversion threat is real. With the increasing use of and dependency on microprocessor devices (e.g. bank cards, cell phones) it would be naive to think that subversion is not also relevant for small microprocessor devices, such as smart cards.

As an example, when integrated circuits (IC) are part of high-security systems, it is important to consider the entire life cycle, including outsourcing of production. Agrawal et al. [2] approaches the problem of how to detect insertion of trojans when outsourcing part of your manufacturing. They use side-channels to build signatures that can be used to detect the introduction of ”extra” circuitry in IC’s. The method is yet impractical as it requires that you pick a few of the produced IC to build fingerprints. The challenge then is to ensure that the few IC’s you pick are not infested. The idea, however, is very interesting as it takes side-channels that are normally considered a vulnerability and uses them as a protective measure.

## 2.5 Bibliography

[1] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., AND ROHATGI, P. The em side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 29–45. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_4`. 1, 15, 17, 19, 23, 37, 50, 71, 85, 88, 113, 127

[2] AGRAWAL, D., BAKTIR, S., KARAKOYUNLU, D., ROHATGI, P., AND SUNAR, B. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy* (may 2007), pp. 296 –310. 31, 46, 61, 71, 85, 99, 113, 127

[3] AGRAWAL, D., RAO, J., AND ROHATGI, P. Multi-channel attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), C. Walter, e. Ko, and C. Paar, Eds., vol. 2779 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 2–16. Available from: `http://dx.doi.org/10.1007/978-3-540-45238-6_2`. 1, 17, 37, 50, 88

[4] ANDERSON, E. A., IRVIN, C. E., AND SCHELL, R. R. Subversion as a threat in information warfare. *Journal of Information Warfare 3*, 2 (2004), 52–65. 30, 31

[5] Archambeau, C., Peeters, E., Standaert, F. X., and Quisquater. Template attacks in principal subspaces. In *Cryptographic Hardware and Embedded Systems - CHES* (2006), vol. 4249 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1–14. 1, 30, 39, 72, 91

[6] Atmel corporation. *Atmel AVR microcontrollers*. Available from: `http://www2.atmel.com/`. 18

[7] Cadence Design Systems. *PSpice*. Available from: `http://www.cadence.com/us/pages/default.aspx`. 11

[8] Chari, S., Rao, J., and Rohatgi, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_3`. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[9] Duda, R., Hart, P., and Stork, D. *Pattern Classification*. John Wiley and Sons, Inc, 2001. 8, 26, 28, 29, 30, 63, 64, 65, 75, 78, 90, 91, 94, 124, 137, 141

[10] Dyrkolbotn, G. O., and Snekkenes, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: `http://dx.doi.org/10.1007/11935308_18`. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[11] Dyrkolbotn, G. O., Wold, K., and Snekkenes, E. Layout dependent phenomena: A new side-channel power model. *Journal of Computers 7*, 4 (April 2012). 20, 40, 41, 42, 47, 103

[12] Fodor, I. K. A survey of dimension reduction techniques. Tech. Rep. UCRL-ID-148494, Lawrence Livermore National Laboratory, Technical Information Departments Digital Library, 2002. Available from: `http://www.llnl.gov/tid/Library.html`. 30, 72

[13] Gandolfi, K., Mourtel, C., and Olivier, F. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES* (2001), vol. 2162 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 251–261. Available from: `http://dx.doi.org/10.1007/3-540-44709-1_21`. 1, 7, 15, 17, 21, 26, 37, 85, 113, 127

[14] Giancoli, D. C. *Physics for Scientists and Engineers*. Prentice Hall, 1989. 14, 50, 88

[15] Haykin, S. *Communication Systems*, 2 ed. John Wiley and Sons, Inc, 1983. 15

[16] Kocher, P. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology* (1996), vol. 1109 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 104–113. Available from: `http://dx.doi.org/10.1007/3-540-68697-5_9`. 1, 26, 50, 71, 88

[17] Kocher, P., Jaffe, J., and Jun, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: `http://dx.doi.org/10.1007/3-540-48405-1_25`. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[18] Kuhn, M., and Anderson, R. Soft tempest: Hidden data transmission using electromagnetic emanations. In *Information Hiding* (1998), vol. 1525 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 124–142. Available from: `http://dx.doi.org/10.1007/3-540-49380-8_10`. 1, 7, 37, 46, 51

[19] Kuhn, M. G. *Compromising Emanations: Eavesdropping Risks of Computer Displays*. Technical report, University of Cambridge, 2003. UCAM-CL-TR-577. 1, 25, 37

[20] Mangard, S., Oswald, E., and Popp, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[21] Microchip Technology. *MPLAB Integrated Development Environment*. Available from: `http://www.microchip.com/`. 18

[22] Microchip Technology Inc. *PicMicro Mid-Range MCU Familiy Reference Manual*, 2007. Available from: `http://www.microchip.com/`. 9

[23] Morris, M., and Kime, C. R. *Logic and Computer Design Fundamentals*. Prentice Hall, 2004. 8

[24] Myers, P. Subversion: The neglected aspect of computer security. Master's thesis, Naval Postgraduate School, 1980. 30, 31, 51

[25] Peebles Jr, P. Z. *Probability, Random Variables, and Random Signal Principles*. McGraw-Hill Inc., 1993. 15, 141

[26] Proakis, J. G., and Manolakis, D. G. *Digital Signal Processing*, 2 ed. Macmillan Publishing Company, 1992. 22, 24, 77

[27] Pucknell, D. *"Basic VLSI Design, Systems and Circuits"*. Prentice Hall, 1988. 11, 12

[28] Quisquater, J.-J., and Koeune, F. "side channel attacks - state of the art". Tech. rep., Math RiZK, consulting, 2002. 17

[29] Quisquater, J.-J., and Samyde, D. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security* (2001), vol. 2140 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 200–210. Available from: `http://dx.doi.org/10.1007/3-540-45418-7_17`. 1, 7, 15, 16, 17, 21, 26, 37, 50, 88

[30] Quisquater, J.-J., and Samyde, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from: `http://portal.acm.org/citation.cfm?id=1250988.1250994`. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[31] Rankl, W., and Effing, W. *Smart Card Handbook*, third ed. Wiley Publishing, 2008. 9

[32] Rao, J., Rohatgi, and Pankaj. Empowering side-channel attacks. Tech. rep., IBM T.J. Watson Research Center, 2001. 1, 7, 16, 17, 37, 50, 88

[33] Rechberger, C., and Oswald, E. Practical template attacks. In *Information Security Applications* (2005), vol. 3325 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 440–456. Available from: `http://dx.doi.org/10.1007/978-3-540-31815-6_35`. 1, 30, 91, 93

[34] Rohatgi, P. Trustworthy hardware. In *Keynote speaker at Cryptographic Hardware and Embedded Systems - CHES* (2007), IBM T. J. Watson Research Center. 30, 46, 61

[35] Skitek, G., and Marshall, S. *Electromagnetic Concepts and Applications*. Prentice Hall, 1987. 2, 14

Chapter 3

# *Summary of Work*

> I love deadlines. I like the whooshing sound they make as they fly by.
>
> DOUGLAS ADAMS

This chapter presents a summary of the work done for this thesis. Figure 3.1 shows the relationship between research questions and published papers. The chapter follows the structure of Figure 3.1, starting with the main research question, followed by research questions related to the following topics: survey and feasibility study, a new attack, in-depth investigation and feature selection, understanding the phenomena and finally, a new application. For each topic, the work involved and papers submitted are introduced. The 7 research papers that constitute the main contribution of this thesis are:

1. G.O. Dyrkolbotn and E. Snekkenes. **A Wireless Covert Channel on Smart Cards (shortpaper)**. In *Proceedings of International Conference on Information and Communications Security (ICICS)*, Springer, Lecture Notes in Computer Science, pages 249-259, 2006, see chapter 5.

2. G.O. Dyrkolbotn. **Analysis of the Wireless Covert Channel Attack: Carrier Frequency Selection**. In *Proceedings of Annual Norwegian Computer Science Conference (NIK)*, Tapir, pages 153-163, 2007, see chapter 6.

3. G.O. Dyrkolbotn and E. Snekkenes. **Electromagnetic Side-Channel: A Comparison of Multi-Class Feature Selection Methods**. In *Proceedings of Signal Processing, Pattern Recognition and Application (SPPRA)*, ACTA Press, track 643-804, 2009, see chapter 7.

4. G.O. Dyrkolbotn and E. Snekkenes. **Modified Template Attack: Detecting Address Bus Signals of Equal Hamming Weight**. In *Proceeding of Annual Norwegian Information Security Conference (NISK)*, Tapir, pages 43-56, 2009, see chapter 8.

5. G.O. Dyrkolbotn, K. Wold and E. Snekkenes. **Security Implications of Crosstalk in Switching CMOS Gates** . In *Proceedings of Information Security Conference (ISC)*, Springer, Lecture Notes in Computer Science, pages 269-275, 2010, see chapter 9.

6. G.O. Dyrkolbotn. **Non-Invasive Reverse Engineering of the Relative Position of Bus Wires**, In *Proceeding of Annual Norwegian Information Security Conference (NISK)*, Tapir, pages 104-109, 2010, see chapter 10.

7. G.O. Dyrkolbotn, K. Wold and E. Snekkenes. **Layout Dependent Phenomena: A New Side-Channel Power Model**. In *Journal of Computers*. Academy Publisher, Volume 7, Number 4 of April 2012 (in press), see chapter 11.

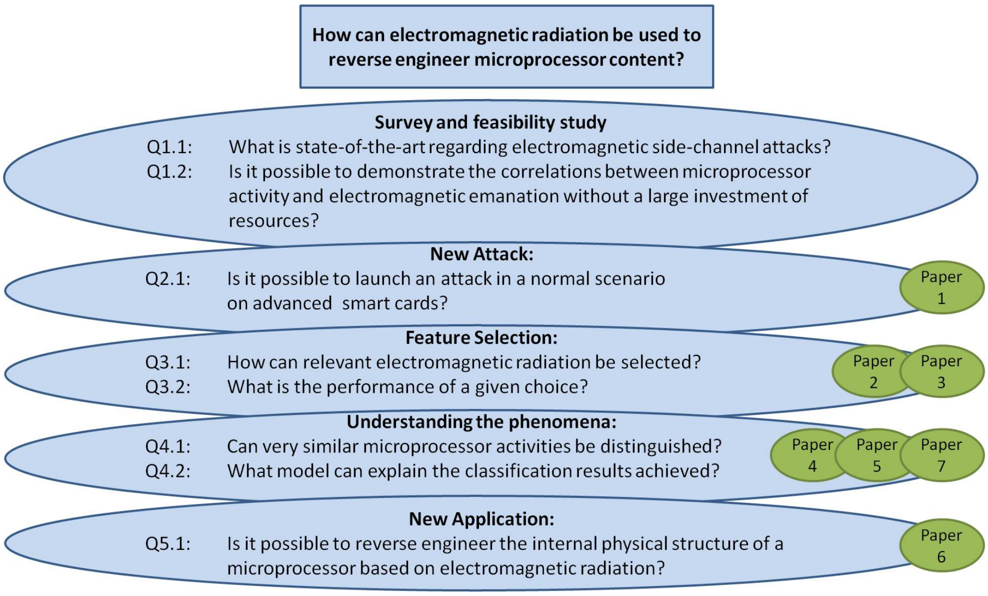
How can electromagnetic radiation be used to reverse engineer microprocessor content?
Survey and feasibility study
Q1.1: What is state-of-the-art regarding electromagnetic side-channel attacks?
Q1.2: Is it possible to demonstrate the correlations between microprocessor activity and electromagnetic emanation without a large investment of resources?
New Attack:
Q2.1: Is it possible to launch an attack in a normal scenario on advanced smart cards?
Paper 1
Feature Selection:
Q3.1: How can relevant electromagnetic radiation be selected?
Q3.2: What is the performance of a given choice?
Paper 2
Paper 3
Understanding the phenomena:
Q4.1: Can very similar microprocessor activities be distinguished?
Q4.2: What model can explain the classification results achieved?
Paper 4
Paper 5
Paper 7
New Application:
Q5.1: Is it possible to reverse engineer the internal physical structure of a microprocessor based on electromagnetic radiation?
Paper 6

Figure 3.1: Relationship between research questions and published papers.

## 3.1 Main Research Question

When a microprocessor executes its program, electromagnetic radiation (EMR) is generated as a consequence of accelerating electric charges associated with transistor transitions (i.e. the power consumption). From this arises the idea that all currents going through a microprocessor will leave a power or electromagnetic fingerprint, that can be used to identify microprocessor activity. Let an electromagnetic fingerprint be a representation of the EMR caused by a certain microprocessor activity. We are now concerned with how an adversary can exploit this fingerprint to reverse engineer microprocessor content. The intent of this thesis is to obtain a better understanding of the origin of EMR from microprocessors, how to capture and represent EMR and how EMR can reveal information about microprocessor content. This can be used to build a more realistic model of the arsenal available to an adversary and facilitate better risk assessment and defensive measures. The overall goal can be stated as:

> **How can electromagnetic radiation be used to reverse engineer microprocessor content?**

This problem had to be split into a number of smaller tasks. A description of each of the research questions and how they are related, follows.

## 3.2 Survey and Feasibility Study

Initially, a survey was conducted to answer the following research question:

**Q1.1:** What is state-of-the-art regarding electromagnetic side-channel attacks?

In the large amount of publications on side-channel attacks, relatively few papers make a clear distinction between the power side-channel and the electromagnetic side-channel, as the electromagnetic side-channel is often viewed only as an alternative method of measuring the power consumption. However, the survey did reveal papers [1, 2, 4, 14, 17, 18, 21, 22, 23, 24] published in an ongoing effort to systematically investigate electromagnetic side-channel attacks. They point out some fundamental differences between the power side-channel and the electromagnetic side-channel, i.e. spatial information and remote capture. Spatial information makes it possible to carefully move a small probe across the surface of a chip to map the radiation [22] or, several probes can be used to pinpoint the location of EMR. Remote capture makes it possible to launch the attack from a distance as shown in [1]. Capturing the electromagnetic side-channel can be done without invasive measures and all analysis techniques for power analysis are still applicable. It has also been claimed that for a stationary probe, electromagnetic analysis (EMA) is at least as powerful as the power analysis, and that EMA can circumvent power analysis countermeasures [1, 24].

The survey left the impression that working with side-channel attacks required knowledge of a broad variety of subjects such as, EMR, measurement techniques (e.g. concerning antennas and instruments), microprocessor devices (e.g. smart cards), programming (e.g. assembly), signal processing and statistics. It was therefore necessary, without a large investment of time and money, to address the following research question:

**Q1.2:** Is it possible to demonstrate the correlations between microprocessor activity and electromagnetic radiation without a large investment of resources?

A series of experiments on a microcontroller test kit and a simple smart card were carried out. Specially designed programs were executed on the devices. The resulting EMR was captured with a hand-made solenoid probe, spectrum analyzer and oscilloscope. Visual inspection of captured traces was enough to recognize program cycle, instructions and to some extent even operands. These experiments confirmed that it was possible to find correlations between EMR and microprocessor activity, at a considerable distance (up to 10 meters for some smart cards), without expensive equipment and with reasonable time investment. During experimentation to demonstrate side-channel leakage, we realized that through careful choice of executed code, the radiation could be manipulated to demonstrate the correlations more clearly. In addition, it was apparent that the main focus of current research was on EMR, from legitimate programs (e.g. specific implementations of cryptographic algorithms). Attack scenarios required a ″lost″ or ″stolen″ card and experiments were performed on simple cards of old technology. We came to the understanding that current attacks appeared impractical outside a lab environment. The current model of an adversaries arsenal was missing practical attacks on advanced smart cards launched in normal scenarios. Research questions Q1.1 and Q1.2 did not result in any publication, but were an important part of this thesis, as we needed a good platform from which to start.

## 3.3 A New Attack

An adversary looking to launch an attack on a system in use is faced with different challenges than in a ″lab scenario″, and may have to face state-of-the-art smart cards with several security measures against side-channel attacks. In order to investigate what it would take to launch an attack in a normal scenario, we asked the following research question:

**Q2.1:** Is it possible to launch an attack in a normal scenario on advanced smart cards?

Our approach is based upon deliberate manipulation of EMR, rather than exploiting unintentional EMR from legitimate activity. By combining knowledge from different fields; electromagnetic side-channels, covert channels and subversion, we proposed a new attack: The **Wireless Covert Channel Attack (WCCA)**, published in our first paper [9]. The scenario assumes that a highly skilled insider is able to hide a small program (subversive code) on a microprocessor smart card in an early stage of the products life cycle. During normal use of the smart card, the subversive code intentionally manipulates the electromagnetic side-channel, creating a covert channel that can potentially broadcast the cards internal secrets to a nearby receiver. The attack consists of three phases; (i) preparation, (ii) implementation and (iii) exploitation.

(i) The preparation phase is used to build a library of characteristic EMR from instructions executed by the microprocessor smart card. WCCA uses the average power density spectrum (PDS) obtained from a spectrum analyzer to characterize instructions. A symbol alphabet and a carrier frequency for the channel is obtained from analysis of the recorded PDS's. Notice that WCCA only requires manipulation of internal activity on the card. Manipulation of the I/O interface would create a stronger signal, however, this would require manipulation of both the terminal and the card, making WCCA harder to achieve and increasing the risk of detection.

(ii) The implementation phase is used to design and hide the subversive code needed to create the covert channel. The microprocessor executing the code is generally a M-ary bandpass digital system with multiple carrier frequencies, modulated with binary data. WCCA was demonstrated as a binary single-carrier system. A cheap narrow band AM receiver (ICOM IR-20) was used to intercept the covert channel. Three different subversive codes were designed to illustrate the potential. The first code demonstrates energy leakage from the microprocessor, by generating three audible tones detectable with the AM receiver. The second code demonstrates information leakage by broadcasting an SOS signal. The third code demonstrates the potential compromise of sensitive information, by broadcasting memory content. It was beyond the scope of this work to investigate how the subversive code could be hidden on microprocessor smart cards, however, third-party compilers and library files with more relaxed security policies were mentioned as a potential loophole.

(iii) The exploitation phase is used to intercept the broadcast channel with a receiver during normal operation of the card, thus getting access to sensitive information. The success will depend on factors such as: How many cards are effected (e.g. entire generations or only a few)? How well is the subversive code hidden (i.e. probability of revealing itself)? Properties of the covert channel (e.g. such as range, capacity, coding schemes, error correction). How long must the channel be active and what is the quality of the receiver (e.g. sensitivity and size)?

Experiments were carried out on modern smart cards[1] and confirmed the feasibility of the attack even with security features against side-channel attacks activated. The security features limited the range, but did not prevent the attack. WCCA showed that it is feasible to get access to sensitive information in an attack resembling wireless skimming , during normal use of advanced smart cards. The attack also underlined the necessity for considering security during the entire life cycle of the whole system. Testing WCCA in a normal operating scenario is left for future work, but the performance of the covert channel was investigated in the second paper [7]. Using theory from binary communication systems, implementation choices such as carrier frequency selection was compared in terms of probability of error (i.e. misinterpretation of symbol 0 and 1). During this work the idea for a customized receiver for WCCA emerged. This idea was not pursued, but is outlined in future work.

## 3.4 Feature Selection

Experimental results [9, 7] clearly showed that EMR is found in a wide frequency range, from DC up to 1 GHz. Obvious questions are: What is the best carrier frequency to use? How can more of the available energy (i.e. EMR) be exploited? This is expressed by the following research questions:

**Q3.1:** How can relevant electromagnetic radiation be selected?

**Q3.2:** What is the performance of a given choice?

[1] Identity withheld due to a non-disclosure agreement, see 1.3

Results from [7] showed that channel performance could easily be improved by increasing the symbol alphabet and using more than one carrier frequency. It was also found that expressing WCCA as a multi-class pattern recognition problem would be beneficial. The problem could then be stated as: How to detect and identify a set of classes (i.e. a set of activities on the microprocessor) based on a set of variables (i.e. the energy on selected carrier frequencies) of the intercepted EMR. Challenges are then typical for pattern recognition systems and are also found in other side-channel attacks (e.g. the template attack [4]), where open issues include [3]: How to select the relevant samples (variables) and how many samples are needed to construct the attack? These challenges are addressed in the third paper: "A Comparison of Multi-class Feature Selection Methods" [10], in which WCCA was used to perform an application specific comparison between different feature selection methods, in multi-class cases. Three groups of methods were compared:

**Variable ranking methods:** These methods use a score function to rank individual variables [15]. Due to their simplicity, this group of methods are a good starting point. Examples are single variable classifier, difference of means (DOM) and Mahalanobis distance. DOM was used in the original DPA attack [16]. A drawback with variable ranking methods is that correlations between variables are disregarded.

**Variable subset selection methods:** These methods try to take into consideration correlations between variables by selecting subsets of variables that work well together. Greedy forward selection is one such method.

**Feature construction methods:** These methods provide different mappings of the raw data. Examples are principal component analysis, Linear discriminant analysis, clustering and fourier transform.

The objective of feature selection can be threefold [15], (i) improving the prediction performance, (ii) faster and more cost-effective classifiers and (iii) better understanding of the underlying processes. The main focus in [10] was on improving the prediction performance. Alternative classifiers to Bayes were not studied. However, combining knowledge from WCCA with pattern recognition [10], also gave us new insight into underlying processes of side-channels, as well as a good foundation for future in-depth investigations of side-channel phenomena in general.

## 3.5 Understanding the Phenomena

An article by Quisquater and Samyde [23] demonstrates how to automatically reverse engineer software code, based on power and electromagnetic signatures. In that article, it is stated that data/addresses of equal Hamming Weight (HW), such as $55$ and $AA$ cannot be distinguished without using localization principles (i.e. positioning of tiny probes to identify the location of the radiation). Such signals are by HD/HW power models assumed to consume equal amounts of power and thus create identical EMR fingerprints. A recent methods, the Template Attack [4], has overcome this limitation. This encouraged us to apply our feature selection method in a template attack to investigate how minute a detail could be distinguished through classification. By focusing on similar/comparable activities, assuming that approximately equal power consumption results in similar/identical EMR fingerprints, the following research question was addressed:

**Q4.1:** Can very similar microprocessor activities be distinguished?

This research question is the focus of the forth paper [11]. A microprocessor was made to execute signals, that by HW model consume equal amount of power. The signals chosen were parallel address bus activity of the PIC 16F84A, the microprocessor found in Gold card smart cards. EMR was captured with a high-end oscilloscope (10 GS/s sampling rate) using a broadband E near field probe positioned as close to the microprocessor as possible, without decapsulation. The analysis method used was, in principle, Bayesian classifier with a Gaussian assumption on the noise distribution, as described in the template attack [4]. However, a lot of work was put into improving how appropriate features (interesting point) of the EMR traces are chosen. This was based on results from [10] and involved a score function for selecting the relevant time-domain window, Welch non-parametric estimation of PDS and finally greedy forward selection (GFS[2]). This reduced EMR traces with 30000 variables to 5 features in the final classification. Using this modified template attack, we were able to distinguish parallel address bus activities of equal HW [11]. During this work, a method for improving classification accuracy based on multiple observations (*Majority Voting*), was suggested, but was left for future work.

The modifications to the template attack provided us with a method for in-depth investigations of the electromagnetic side-channel. A variety of microprocessor activities, such as data transfer on the microprocessor's internal data bus and instruction bus were studied in addition to the address bus. The classification results that we achieved can not be explained by commonly used power models (HW/HD models [19]), therefore leading to the following research question:

**Q4.2:** What model can explain the classification results achieved?

In two papers [12, 13] we put forward the hypothesis that the classification results achieved can be explained by layout dependent phenomena (LDP). LDP are parasitic elements such as; inductance and capacitance of conductors, inductance and capacitance between conductors, wireless transmission characteristics (i.e. antenna properties) of conductors and other circuit elements and complex combinations of these phenomena. In particular, the coupling capacitance is well studied within sub-micron VLSI design [6, 20, 25]. When transistor technology gets smaller, capacitive crosstalk becomes such a major contributor to the power consumption and delays of on-chip buses, that it can no longer be ignored. These effects may be known to the security community [5, 19] but have received little attention.

A new power model, taking into consideration capacitive crosstalk, is suggested in [12] and [13] (preproceedings for [12] are included in appendix A). This model is generalized to $n$ wires and an expression to calculate the analytic power consumption is derived. PSPICE is used to compare analytic and simulated power consumption. Simulation results confirmed that the dissipated energy from CMOS switching gates depend not only on the HD, but also on the direction of switching activity on nearby data lines. This increases the number of possible energy levels and, therefore, allows us to explain why signals with the same number of transitions sometimes can be distinguished. Finally, a case study on EMR from smart cards is conducted in order to test the practicality of our theory and simulations. The results support that exploitation of microprocessor side-channels can be improved if one has access to details regarding the physical layout of the microprocessor, e.g. the capacitance between bus wires.

## 3.6 New Application

The new power model and the work of paper [12] and [13] show that EMR is a function of microprocessor activity, power consumption and LDP (i.e. internal physical structures). We have shown that with knowledge of microprocessor activity and LDP and captured EMR, we can predict the power consumption better than previous power models. However, it should also be possible to solve this equation for any of the other variables. Therefore,

[2] Notice that for static phenomena, in which the feature selection algorithm is executed once, GFS performs best. A dynamic phenomena, with repeated execution of the feature selection algorithm, would favor algorithms with short processing time [10]. Only static phenomena have been studied in this thesis.

given knowledge of microprocessor activity, power consumption and captured EMR, it should be possible to predict internal physical structures of the microprocessor. This spinoff of the work in [12] and [13] is, to the best of our knowledge, a new application area for side-channel information and led us to the following research question:

**Q5.1:** Is it possible to reverse engineer the internal physical structure of a microprocessor based on electromagnetic radiation?

As an example of the potential, in [8] we provided a noninvasive technique for establishing the relative position of microprocessor bus wires. This technique relies on the influence capacitive crosstalk has on the electromagnetic side-channel. This work can easily be extended by including more LDP. This may provide even more precise results when classifying microprocessor activity or have the potential to reveal additional physical structures of the microprocessor. This opens up for potential research in the future.

## 3.7 Future Work

In the following, topics for potential future work are described.

### 3.7.1 Customized WCCA Receiver

The feasibility of WCCA was demonstrated in [9] using a cheap narrow band AM receiver. The covert channel can be improved significantly by customizing a receiver for the purpose. In traditional communications system design, the carrier frequencies and waveforms are part of the design strategy. In WCCA, these parameters cannot be changed, as they are a product of unwanted leakage from microprocessors. A customized receiver for WCCA has to exploit whatever energy is available when the microprocessor executes a program. Two different approaches look interesting: (i) A harmonic receiver, in which a mixer with a harmonic signal (e.g. the clock signal of the microprocessor) could merge the energy of several harmonic carriers together. This is based on the assumption that merging several carriers together will give a stronger signal and preserve the information. (ii) Adapting the principle from multiple-input and multiple-output (MIMO) systems (e.g. IEEE 802.11n) could be used to select several carriers. Each carrier could be subject to different demodulation schemes.

Both receivers could use M-ary signalling and coding schemes. Implementing the receiver as an SW radio makes adaptive carrier selection possible, in which the receiver undergoes a training phase on the targeted microprocessor architecture prior to launching the attack. The universal software radio peripheral (USRP) may be a suitable platform to use.

Another unresolved issue with WCCA is the maximum range of the channel. From what distance is it possible to launch WCCA? Is it possible to design a subversive code that it detectable from orbiting satellites? Finally, no real-world testing has so far been carried out with WCCA.

### 3.7.2 Systematic In-depth Investigations of EMR

The results in [11] serve as a good platform for systematic, in-depth investigations of EMR from microprocessors. The classification method can be automated and used to compare EMR from a large number of microprocessors for a large frequency range. EMR from several microprocessors of the same and different architecture could be compared to check the reliability and validity of classifier training. It is likely that a classifier trained for PIC architecture cannot be used for AVR architecture. However, it is important to find out if a classifier trained for one PIC 16F84A can be used to attack any PIC 16F84A processor. Different choices of classifiers, other than Bayesian, could be compared. Extensive testing of bus transitions could be carried out, based on collecting thousand of observations of all transition patterns (65536 in total) on an 8 bit bus.

### 3.7.3 Majority Voting

The template attack is, in theory, capable of classifying microprocessor activity based on a single observation. In [11] we show that the classification error rate is, however, often large when a single observation is used. In many scenarios multiple observations of the same phenomena are available. A program may reuse some values over and over again (e.g. user name, password, PIN code or cryptographic key). A specific program (or part of program) may be used several times or maybe a subversive code or trojan can force repetitive execution. Given multiple observations of the same activity, different approaches can be used to reduce the probability of error. Two alternatives are evident: (i) data fusion first, classification second or (ii) classification first, data fusion second. Data fusion first has to combine all observations into a single observation, often by averaging, such that classification is done on a single observation. The other approach is to classify all individual observations first and then fuse the results into a single decision afterwards.

Some preliminary results with alternative two indicate that, given multiple observations, simply deciding in favor of the class with the majority of the classification decisions (votes), offers improved recognition performance over averaging before classification. This method, which we called *Majority Voting*, was never published, but a short introduction to the problem is given in appendix B as a motivation for future work.

### 3.7.4 Improved Power Model

In [12, 13] we present a new power model, taking into consideration capacitive crosstalk. Simulations and theoretical results in [12] support that LDP (e.g. capacitance) must be considered when analyzing security implications of electromagnetic side-channels. It remains to apply the new power model in actual side-channel attacks (e.g. DPA). Improving the model by including more LDP is also left for future work. Suggested phenomena to start with are: variations in coupling and load capacitance, inductance, effect of bends in circuit paths and multi layer capacitance (3-dimentional). The work can also be extended to study larger sized buses (e.g. 16 or 32 bit).

### 3.7.5 Reverse Engineering

The non-invasive method to reverse engineer the relative position of microprocessor bus wires in [8] has not been tested experimentally. We also believe that by considering other LDP it should be possible to reveal other physical structures of the microprocessor, such as memory, masking schemes and dual-rail logic.

## 3.8 Bibliography

[1] Agrawal, D., Archambeault, B., Rao, J., and Rohatgi, P. The em side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 29–45. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_4`. 1, 15, 17, 19, 23, 37, 50, 71, 85, 88, 113, 127

[2] Agrawal, D., Rao, J., and Rohatgi, P. Multi-channel attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), C. Walter, e. Ko, and C. Paar, Eds., vol. 2779 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 2–16. Available from: `http://dx.doi.org/10.1007/978-3-540-45238-6_2`. 1, 17, 37, 50, 88

[3] Archambeau, C., Peeters, E., Standaert, F. X., and Quisquater. Template attacks in principal subspaces. In *Cryptographic Hardware and Embedded Systems - CHES*

(2006), vol. 4249 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1–14. 1, 30, 39, 72, 91

[4] CHARI, S., RAO, J., AND ROHATGI, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_3`. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[5] CHEN, Z., HAIDER, S., AND SCHAUMONT, P. Side-channel leakage in masked circuits caused by higher-order circuit effects. In *Advances in Information Security and Assurance* (2009), vol. 5576 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 327–336. Available from: `http://dx.doi.org/10.1007/978-3-642-02617-1_34`. 40, 99, 108, 113, 127

[6] DUAN, C., CALLE, V., AND KHATRI, S. Efficient on-chip crosstalk avoidance codec design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 17*, 4 (april 2009), 551 –560. 40, 99, 100, 101, 102, 108, 114, 118, 121, 128, 132, 135

[7] DYRKOLBOTN, G. O. Analysis of the wireless covert channel attack: Carrier frequency selection). In *Norwegian Computer Science Conference - NIK* (2007), Tapir akademisk forlag, pp. 153–163. 38, 39, 45, 46, 71, 73

[8] DYRKOLBOTN, G. O. Non-invasive reverse engineering of the relative position of bus wires. In *Norwegian Information Security Conference - NISK* (2010), Tapir akademisk forlag, pp. 104–109. 41, 42, 47

[9] DYRKOLBOTN, G. O., AND SNEKKENES, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: `http://dx.doi.org/10.1007/11935308_18`. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[10] DYRKOLBOTN, G. O., AND SNEKKENES, E. Electromagnetic side channel: A comparison of multi-class feature selection methods. In *Signal Processing, Pattern Recognition and Application - SPPRA* (2009), vol. 643-804, ACTA Press. 39, 40, 46

[11] DYRKOLBOTN, G. O., AND SNEKKENES, E. Modified template attack: Detecting address bus signals of equal hamming weight. In *Annual Norwegian Information Security Conference - NISK* (2009), Tapir akademisk forlag, pp. 43–56. 39, 40, 41, 42, 46, 47, 99, 109, 113, 122, 124, 127, 135, 137, 142

[12] DYRKOLBOTN, G. O., WOLD, K., AND SNEKKENES, E. Security implications of crosstalk in switching cmos gates. In *Information Security Conference - ISC* (2010), vol. 6531 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 269–275. 40, 41, 42, 47, 107, 108, 111, 114, 118, 125

[13] DYRKOLBOTN, G. O., WOLD, K., AND SNEKKENES, E. Layout dependent phenomena: A new side-channel power model. *Journal of Computers 7*, 4 (April 2012). 20, 40, 41, 42, 47, 103

[14] GANDOLFI, K., MOURTEL, C., AND OLIVIER, F. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES* (2001), vol. 2162 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 251–261. Available from: `http://dx.doi.org/10.1007/3-540-44709-1_21`. 1, 7, 15, 17, 21, 26, 37, 85, 113, 127

[15] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (March 2003), 1157–1182. Available from: http://portal.acm.org/citation.cfm?id=944919.944968. 39, 72, 75, 76, 77, 78, 93

[16] Kommerling, O., and Kuhn, M. G. Design principles for tamper-resistant smartcard processors. In *USENIX Workshop on Smartcard Technology* (Berkeley, CA, USA, 1999), USENIX Association, pp. 9–20. Available from: http://portal.acm.org/citation.cfm?id=1267115.1267117. 39

[17] Kuhn, M., and Anderson, R. Soft tempest: Hidden data transmission using electromagnetic emanations. In *Information Hiding* (1998), vol. 1525 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 124–142. Available from: http://dx.doi.org/10.1007/3-540-49380-8_10. 1, 7, 37, 46, 51

[18] Kuhn, M. G. *Compromising Emanations: Eavesdropping Risks of Computer Displays*. Technical report, University of Cambridge, 2003. UCAM-CL-TR-577. 1, 25, 37

[19] Mangard, S., Oswald, E., and Popp, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[20] Moll, F., Roca, M., and Isern, E. Analysis of dissipation energy of switching digital cmos gates with coupled outputs. *Microelectronics Journal 34*, 9 (2003), 833 – 842. Available from: http://www.sciencedirect.com/science/article/pii/S0026269203001332. 40, 99, 100, 102, 108, 114, 116, 117, 118, 121, 128, 130, 131, 132, 134, 135

[21] Quisquater, J.-J., and Samyde, D. A new tool for non-intrusive analysis of smart cards based on electromagnetic emissions:the sema and dema methods. *Eurocrypt rump session* (2000). 1, 37, 50, 88

[22] Quisquater, J.-J., and Samyde, D. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security* (2001), vol. 2140 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 200–210. Available from: http://dx.doi.org/10.1007/3-540-45418-7_17. 1, 7, 15, 16, 17, 21, 26, 37, 50, 88

[23] Quisquater, J.-J., and Samyde, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from: http://portal.acm.org/citation.cfm?id=1250988.1250994. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[24] Rao, J., Rohatgi, and Pankaj. Empowering side-channel attacks. Tech. rep., IBM T.J. Watson Research Center, 2001. 1, 7, 16, 17, 37, 50, 88

[25] Sotiriadis, P., and Chandrakasan, A. Low power bus coding techniques considering inter-wire capacitances. In *Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000* (2000), pp. 507 –510. 40, 99, 108, 114, 118, 128, 132

Chapter 4

# *Summary of Contributions*

> The greatest enemy of knowledge is not ignorance, it is the illusion of knowledge
>
> STEPHEN HAWKING

This thesis contributes to build a more realistic model of the arsenal available to an adversary engaged in reverse engineering microprocessor content by the electromagnetic side-channel. Contributions in four areas can be identified:

- **Wireless Covert Channel Attack:** Contribution towards exploiting the electromagnetic side-channel in a new attack and attack scenario for microprocessor smart cards which highlights the importance of life-cycle security measures.

- **Feature Selection:** Contribution towards better understanding of what and how much of the available electromagnetic radiation is necessary to launch an attack. This results in a method for in-depth analysis of the electromagnetic side-channel.

- **A New Power Model:** Contribution towards better understanding of underlying phenomena by considering layout dependent phenomena. This can improved side-channel exploitation capabilities.

- **Reverse Engineering Physical Properties:** Contribution toward an alternative application of the electromagnetic side-channel as a consequence of the new power model.

## 4.1 Wireless Covert Channel Attack

The relationship between EMR and microprocessor activity can be exploited by an adversary to deduce content of program or data memory [3, 11, 14], otherwise kept secret. In side-channel attacks, the adversary may have knowledge of, or even be able to, choose part of the execution sequence in order to reveal specific details. By choosing inputs to legitimate programs (e.g. chosen plain text to a cryptographic algorithm) the adversary can use side-channel attacks to reveal sensitive information (e.g. cryptographic keys).

One of the contributions of this thesis is a new attack and attack scenario for microprocessor smart cards, *the Wireless Covert Channel Attack* (WCCA) [4, 6]. WCCA was a result of investigation as to what it would take to launch an attack on advanced smart cards, in a normal user scenario. The suggested attack resembles wireless skimming and has the potential to broadcast any sensitive information from the microprocessor to a remote receiver. Instead of relying on EMR from chosen input to legitimate programs, subversive code is used to manipulate the EMR. Subversive code is understood as malicious code hidden by a professional (i.e. highly skilled) adversary. The subversive code can search for sensitive information and broadcast it to a nearby receiver by manipulation of the electromagnetic side-channel.

The scenario is based on subversion, which, according to Anderson et al. [2], is a forgotten aspect of information security. The scenario assumes that a highly skilled insider can hide malicious code on smart cards prior to their distribution to customers. We propose that this can be done through the use of third-party compilers and library functions. Commercial interests in not revealing the source code of such programs and possibly less stringent security requirements than the developed SW, may be the loophole needed by the adversary. If successful, the security challenge for large-scale users, e.g. the banking industry and pay-tv, will be to prove the absence of malicious code. This is, by far, a trivial task [2]. A similar challenge has since been addressed by Agrawal et al. [1] in an attempt to use side-channel information to detect the introduction of ”extra” circuitry when outsourcing production of integrated circuits. In addition, Agrawal et al. [1] and P. Rohatgi [15], highlight the importance of considering security during the entire life-cycle of a high-security product. This is the same conclusion recommended in WCCA, two years earlier. Countermeasures should limit the opportunities to insert malicious code as well as limit the ability to place a receiver in the vicinity of a terminal during the entire life-cycle of a system.

The idea of deliberate manipulation of EMR from a device is not new and has, for example, been suggested by Kuhn and Aderson [12]. The application to microprocessor smart cards is, however, new and an important attribute is that no manipulation of the terminal is necessary. The attack is solely based on internal microprocessor activity of the card. Using I/O towards the terminal, would give higher S/N ratio, but at the cost of having to manipulate two units, making the attack more complicated and increasing the risk of detection. The attack is also tailored to the microprocessor architecture rather than the application.

The feasibility of the attack has been demonstrated on modern, high-security cards[1] with all available security features activated [6] [4]. It is worth mentioning that one of the security features actually activated a strong carrier frequency usable for the covert channel, with energy levels far above levels detected prior to activation. This should serve as a wake-up call to implementer of security features: addressing one problem may introduce a new weakness.

WCCA may be ahead of its time since cheaper and easier attacks exists. We believe this may change as the magnetic strip is phased out and necessary technology gets cheaper. The relevance of wireless skimming will most likely increase as new, on-line applications of the microprocessor smart card makes it easier to infect microprocessor cards with malicious code through traditional trojans. The popularity of smart-phones and tablet computers (e.g. iPad) may be areas of special concern, as they are already vulnerable to trojans and contain an increasing amount of sensitive information. These developments encourage further research on the topic.

## 4.2 Feature Selection

A challenge faced by WCCA and other side-channel attacks is: What and how much of the available EMR is necessary to launch an attack and how do choices affect the efficiency of the attack? These are also open issues in other side-channel attacks, such as the template attack [3].

This thesis recognizes side-channel attacks (i.e. reverse engineering microprocessor content) as a pattern recognition problem, and can therefore address these challenges as a feature selection problem. This thesis provides a comparison of several multi-class feature selection methods by their performance in a WCCA application [7]. The performance is based on: (i) the number of variables needed, (ii) the amount of time required and (iii) the error rate achieved. Combining these results with the Bayesian classification (e.g. template attack [3]) provided a method for in-depth analysis of side-channel phenomena. The method has been used to investigate the level of detail that can be distinguished by EMR from a microprocessor smart card [8]. The results revealed that commonly used power models are not suitable to explain the level of detail achieved by classification (e.g. template attack).

[1] Identity withheld due to a non-disclosure agreement, see 1.3

## 4.3 A New Power Model

Even though power simulations at analog level are mentioned as ”the most precise way to simulate power consumption on digital circuits” [13], they receive little attention in the security community. Simplifications are commonly applied, such as lumping together all intrinsic capacitances to a single capacitance to ground. These simplifications make commonly used power models (i.e. HD/HW model [13]) unable to explain the level of details obtained when reverse engineering microprocessor activity by classification [8].

This thesis contributes [9] towards better understanding of phenomena enabling side-channel attacks, by considering layout dependent phenomena (LDP), such as; inductance and capacitance of conductors, inductance and capacitance between conductors, wireless transmission characteristics (i.e. antenna properties) of conductors and other circuit elements and complex combinations of these phenomena.

An important contribution is a new power model that shows how capacitive crosstalk can explain why it sometimes is possible to distinguish transition patterns of an equal number of transitions [9]. Simulations and experiments in [9] and [10] show that capacitive crosstalk has a significant effect on gate energy dissipation and that LDP must be considered when analyzing security implications of electromagnetic side-channels. The DPA attack [11] rely upon a good power model to be successful. The new power model that we suggest could, therefore, improve the DPA attack.

## 4.4 Reverse Engineering Physical Properties

The new power model presented in [9] shows that energy dissipation (i.e. EMR) is a function of internal physical structures (i.e. LDP) of the microprocessor. With knowledge of internal physical structure, better power models can be built and reverse engineering microprocessor activity (e.g. breaking a cryptographic implementation) can be performed more precisely.

This thesis contributes towards an alternative area of application for the electromagnetic side-channel. The hypothesis is that if the microprocessor activity is known, it is possible to reverse engineer internal physical structures of the microprocessor by measuring the effect they have on the power consumption and thus the EMR. In particular, in [5] a non-invasive method to reverse engineer the relative position of microprocessor bus lines is presented. This method takes advantage of the new power model [9], i.e. the influence of capacitive crosstalk on the energy dissipation when the transition pattern is known.

We believe that by considering other LDP it should be possible to reverse engineer other physical structures of the microprocessor, such as memory, masking schemes or dual-rail logic.

## 4.5 Bibliography

[1] AGRAWAL, D., BAKTIR, S., KARAKOYUNLU, D., ROHATGI, P., AND SUNAR, B. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy* (may 2007), pp. 296 –310. 31, 46, 61, 71, 85, 99, 113, 127

[2] ANDERSON, R. J. *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2 ed. Wiley Publishing, 2008. 45, 46

[3] CHARI, S., RAO, J., AND ROHATGI, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_3`. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[4] Dyrkolbotn, G. O. Analysis of the wireless covert channel attack: Carrier frequency selection). In *Norwegian Computer Science Conference - NIK* (2007), Tapir akademisk forlag, pp. 153–163. 38, 39, 45, 46, 71, 73

[5] Dyrkolbotn, G. O. Non-invasive reverse engineering of the relative position of bus wires. In *Norwegian Information Security Conference - NISK* (2010), Tapir akademisk forlag, pp. 104–109. 41, 42, 47

[6] Dyrkolbotn, G. O., and Snekkenes, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: http://dx.doi.org/10.1007/11935308_18. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[7] Dyrkolbotn, G. O., and Snekkenes, E. Electromagnetic side channel: A comparison of multi-class feature selection methods. In *Signal Processing, Pattern Recognition and Application - SPPRA* (2009), vol. 643-804, ACTA Press. 39, 40, 46

[8] Dyrkolbotn, G. O., and Snekkenes, E. Modified template attack: Detecting address bus signals of equal hamming weight. In *Annual Norwegian Information Security Conference - NISK* (2009), Tapir akademisk forlag, pp. 43–56. 39, 40, 41, 42, 46, 47, 99, 109, 113, 122, 124, 127, 135, 137, 142

[9] Dyrkolbotn, G. O., Wold, K., and Snekkenes, E. Security implications of crosstalk in switching cmos gates. In *Information Security Conference - ISC* (2010), vol. 6531 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 269–275. 40, 41, 42, 47, 107, 108, 111, 114, 118, 125

[10] Dyrkolbotn, G. O., Wold, K., and Snekkenes, E. Layout dependent phenomena: A new side-channel power model. *Journal of Computers 7*, 4 (April 2012). 20, 40, 41, 42, 47, 103

[11] Kocher, P., Jaffe, J., and Jun, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: http://dx.doi.org/10.1007/3-540-48405-1_25. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[12] Kuhn, M., and Anderson, R. Soft tempest: Hidden data transmission using electromagnetic emanations. In *Information Hiding* (1998), vol. 1525 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 124–142. Available from: http://dx.doi.org/10.1007/3-540-49380-8_10. 1, 7, 37, 46, 51

[13] Mangard, S., Oswald, E., and Popp, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[14] Quisquater, J.-J., and Samyde, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from: http://portal.acm.org/citation.cfm?id=1250988.1250994. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[15] Rohatgi, P. Trustworthy hardware. In *Keynote speaker at Cryptographic Hardware and Embedded Systems - CHES* (2007), IBM T. J. Watson Research Center. 30, 46, 61

Chapter 5

# *A Wireless Covert Channel on Smart Cards*[1]

## Abstract

Microprocessor devices, such as smart cards, are used more and more to store and protect secret information. This development has its advantages, but microprocessor devices are susceptible to various attacks. Much attention has been devoted to side-channel attacks, exploiting unintentional correlation between internal secret information, such as cryptographic keys, and the various side-channels. We present a wireless covert channel attack (WCCA) that intentionally correlates secret information with the electromagnetic side-channel. WCCA exploits subversive code hidden on all cards during manufacture, to launch an attack, without physical access, when infected cards are used. Experiments on modern smart cards confirm that an insider with the opportunity to hide subversive code can potentially broadcast the card's internal secrets to a nearby receiver. Security features against side-channel attacks will limit the range but not prevent the attack.

## 5.1 Introduction

Since the birth of modern side-channel attack in the 90's there has been an explosion of proposed attacks exploiting side-channels to reveal secret information within a smart card. Current research focuses on exploiting unintended correlations between secret information (cryptographic key) and the side-channel, tailoring a specific implementation of a cryptographic algorithm. These attacks often require a "lost or stolen" card and experimental results are often obtained on simple cards of older technology, not on modern smart cards equipped with countermeasures.

By combining the efforts of different fields, electromagnetic side-channel attacks , covert channels and subversion, we propose a new attack: wireless covert channel attack (WCCA). We believe that hiding subversive code on cards during manufacture can manipulate the energy leakage from a smart card to create a covert broadcast channel. The channel is activated when the cards are used in a normal scenario and will give us access to secret information remotely (i.e. wireless), without the need for physical access to the target. The attack is tailored the microprocessor architecture rather than the actual cryptographic algorithm and experiments confirm that the attack will work on modern smart cards equipped with countermeasures against side-channel attacks.

This article will explain how to collect and analyze electromagnetic emanation from smart cards to build signatures of individual instruction executed by the microprocessor. These signatures will form a symbol alphabet for a covert communication channel. Subversive code hidden on the smart card will create the covert channel and use it to broadcast secret information to a nearby receiver. Practical result obtained on modern smart cards[2] equipped with counter measures will be shown.

[1] G.O. Dyrkolbotn and E. Snekkenes. In *Proceedings of International Conference on Information and Communications Security (ICICS)*, Springer, Lecture Notes in Computer Science, pages 249-259, 2006.

[2] Identity withheld due to a non-disclosure agreement, see 1.3

## 5.2 Previous Work

The basis for side-channel attacks has been available for a long time. It is possible to use the second law of thermodynamics to show that energy must escape from devices in one way or another(e.g. heat) [8]. The laws of physics explain that it is impossible for any operating device not to leak energy. The goal of side-channel attacks is to look for dependencies between this unavoidable energy leakage and the device's secret parameters.

Exploiting this leakage is not new. Military and government organizations have supposedly used them for a long time, with public interest beginning much later. In 1985 Van Eck [19] published the article on how to eavesdrop video display units via radiation from a considerable distance that attracted much attention. In 1996 Anderson and Kuhn published their work, *"Tamper Resistance - A Cautionary Note"* [5], which showed that trusting tamper resistant devices can be problematic. That same year Kocher [9] published his work on exploiting differences in execution time (Timing Attacks). This work was soon followed up and in 1999 Kocher et al. [10] introduced some powerful attacks through measurement of a device's power consumption. Simple Power Analysis (SPA) and Differential Power Analysis (DPA) received some attention from, among others, the banking industry, and countermeasures were publicly announced. In 2000, Quisquater and Samyde [14, 15] applied the analysis technique from SPA and DPA to electromagnetic side-channels, thus introducing electromagnetic analysis (EMA).

In recent years several papers have been published in an ongoing effort to systematically investigate electromagnetic side-channel attacks[6, 2, 3, 1, 4]. The experiments have been extended to some distance from the target, implying that physical access to the target may not be necessary. It has been shown that EMA is at least as powerful as power analysis, and that EMA could circumvent power analysis countermeasures [3, 17]. At USENIX 2002 [16], Quisquater and Samyde described an automatic method to classify instructions, carried out by a simple CISC processor. The power and electromagnetic signature of instructions were captured and then used to train a neural network. The neural network could automatically recognize, and thus reverse engineer, executed code based on stored electromagnetic and power signatures.

Common for previous attacks is that they exploit unintentional correlations between the side-channel and secret information, often a cryptographic key. Taking a more aggressive approach would be to manipulate the side-channel. It is not difficult to imagine a situation were the code on the smart card is manipulated to give specific results for the neural network of Quisquater and Samyde [16].

Covert communication was first introduced by Lampson in 1973 and was then defined as

> A communication channel is covert if it is neither designed nor intended to transfer information at all

An example can be found in an encrypted packet switched network. An adversary can monitor the packet flow, but can not read the encrypted content of each packet. A covert channel can be created if the following is decided beforehand.

- Packet sent from address A to B - interpret as logic 0
- Packet sent from address A to C - interpret as logic 1

This traffic will appear as regular packet switch traffic (at least to the untrained eye) and hopefully not raise any suspicions, e.g. it is covert.

Another example of a covert channel can be the running time of a program. This means that the timing attack of Kocher [9] can be seen as exploiting an unintentional covert channel. Unintentional in the way that the secret information was not intentionally correlated with the timing information. Similarly, other side-channel attacks can also be seen as exploiting unintentional covert channels. Side-channels also fit Lampsons definition from 1973 as stated above.

Kuhn and Andersson [11] talk about attacking a system with malicious code that will use a computer's RF emission to transmit stolen information. The possibility to plant a virus to infiltrate a bank or certificate authority and broadcast key material over an improvised radio channel is mentioned. Practical results are shown with hidden messages in a recovered video signal. This can also be categorized as a covert channel where the electromagnetic side-channel has been deliberately manipulated. This approach will be used in WCCA. We introduce the term **wireless covert channel** as *a hidden electromagnetic communication channel, detectable outside the system, as a result of intentional manipulation of valid system parameters.*

Creating a wireless covert channel can be viewed as subversion , described by Myers [12] as

> The subversion of computer systems is the covert and methodical undermining of internal and external controls over a systems lifetime to allow unauthorized and undetected access to system resources and/or information.

In the next chapter the wireless covert channel attack (WCCA) is presented.

## 5.3 Wireless Covert Channel Attack

The wireless Covert Channel Attack (WCCA) relies on a highly skilled insider to undermine the security mechanisms by hiding subversive code in the smart card's software (SW). This is done during an early stage (design or implementation) of its life cycle (figure 5.1) and will affect all cards produced. These infected cards may be used in the banking industry as credit cards, loaded with personal information (cryptographic key, PIN code, account number etc.) when issued to a customer. The adversary is interested in retrieving the secret, personal information and has an accomplice involved at the use stage of the life cycle. This will be somebody with access to a smart card terminal, a store owner or maintenance personnel. When a manipulated card is inserted into any terminal the subversive code exploits characteristic electromagnetic emanation (signatures) from the microprocessor, during execution of instructions, to broadcasting secret information over a wireless covert channel.

The success of this attack is ensured by the large number of cards infected. If a whole generation of smart cards to the banking industry is infected, there will be enough cards randomly used in the adversaries rigged locations to make the attack worth the effort.

WCCA can be divided into a preparation phase, an implementation phase and an exploitation phase.

### 5.3.1 Preparation Phase

The preparation phase is used to build a library of characteristic electromagnetic emanation from instructions executed by the microprocessor. WCCA uses the average power spectrum density (PSD) obtained from a spectrum analyzer to characterize an instruction.

For every instruction of interest, a smart card is prepared with a test code. The card inserted into a customized smart card reader will automatically start executing the test code. A homemade coil attached to a spectrum analyzer captures and records the resulting electromagnetic emanation. The average PSD is stored on a computer for further analysis. The test code is written in assembly language and executes one instruction in a loop. The instruction is repeated several times within the loop to reduce the effect of any unwanted instructions, such as "goto". A small homemade coil is placed on top of the smart card reader, as close as possible to the microprocessor, without any decapsulation. Using a spectrum analyzer, no synchronization between the executed code and the instrument is

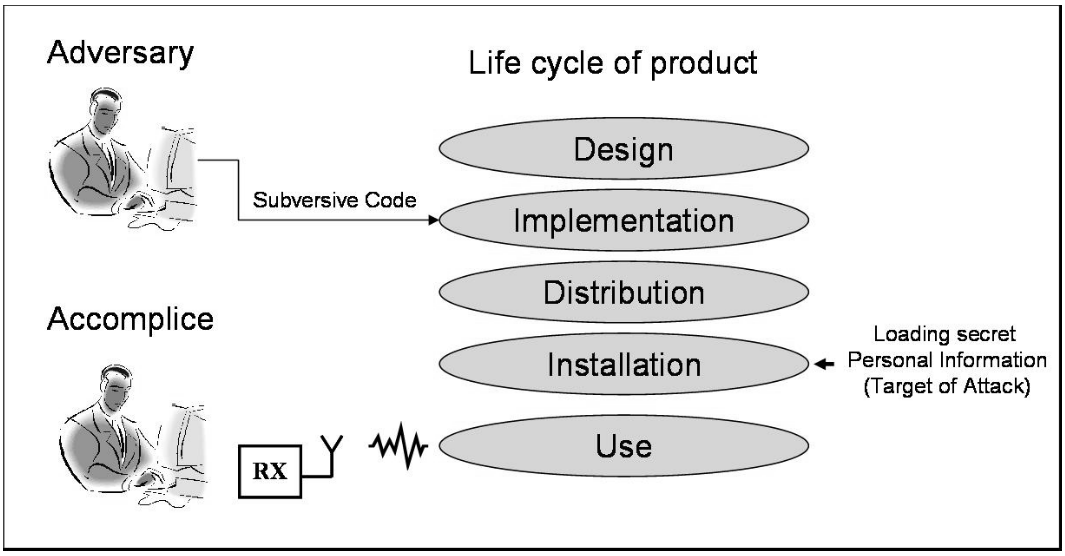

Figure 5.1: Scenario: The adversary hides subversive code during an early stage. Later, secret information is loaded to the card. When the card is used, the subversive code is activated and broadcasts secret information to the accomplice.

needed. Each instruction was measured several times in random order to enhance the reliability of the data. The signature of an instruction is the mean of all these repetitions.

### 5.3.2 Implementation Phase

The implementation phase is used to design and hide the subversive code needed to create the covert channel. A symbol alphabet for the channel, as well as a carrier frequency, is obtained from analysis of the recorded signatures. The smart card executing the subversive code can be considered a bandpass digital system [13], where a carrier is modulated with binary data. In this work we restrict the discussion to a binary system, and leave M-ary bandpass digital systems for future work.

For two possible messages, $m_1$ and $m_2$, two possible waveforms $s_1(t)$ or $s_2(t)$ are transmitted in a bit interval, $T_b$. The waveforms $s_1(t)$ and $s_2(t)$ cannot be chosen freely, but are a result of emitted energy when executing instructions on the smart card. Since the receiver makes the decision based on received energy, it is natural to look for frequencies where the emitted energy can be controlled by execution of different instructions.

In order to demonstrate the feasibility of the attack a low cost narrow band receiver was chosen. The approach chosen is therefore to look for one carrier frequency, $f_c$, with a large difference in emitted energy between execution of two instructions. Exploiting the energy emitted in a larger band is subject to work in progress.

Using the recorded signatures, a carrier frequency is easily found. Let diff(i,j) be the spectral difference between signature i and j. Diff(i,j) is calculated by taking the magnitude of the difference at each sample of signature i and j. Calculating diff(i,j) for all combinations of instructions and sorting all the individual sample differences , there will be two signatures i=A and j=B, where the diff(A,B) for one sample is largest. The frequency where this occurs is chosen as the carrier frequency, $f_c$.

The emitted energy level at carrier frequency $f_c$ can now be controlled by designing a subversive code that transmits secret information using

- Instruction A - logic 0 - small energy emission at $f_c$

- Instruction B - logic 1 - large energy emission at $f_c$

Once the subversive code is designed, the task is to make sure it is hidden on every card produced, undetectable. It is beyond the scope of this article to describe this in detail. However assuming that third party compilers and library functions have less stringent security requirements than the developed SW, together with commercial interest in not revealing the source code of compilers and library files, could be the loophole exploited by the adversary.

### 5.3.3 Exploitation Phase

The success of the attack relies on the adversary or his accomplice placing a receiver in the vicinity where infected cards are used. The subversive code will be executed during normal use and secret information broadcasted to the receiver.

The range of the covert channel will have a great impact on how difficult this will be. Given a range of several meters, the receiver can be placed somewhere in the room and maybe in an adjacent room. It may also be possible to carry a concealed receiver and stand nearby or be in the line behind the victim. If the range is in order of cm, the probe of the receiver must be placed close to the terminal. This may be possible if the accomplice is the store owner or maintenance personnel with access to the terminal.

The receiver can be optimized to cost, range, channel capacity, probability of error, size etc, but even a cheap commercial receiver used in this experiments gives promising results. Due to the relative long exposure time, when the card is used in a terminal (up to 30 sec) the bit rate does not need to be high. Assuming that the covert channel use only 1% of the processor time reduces the risk of detection, and still gives 0.3 sec for the attack. Sending 1024 bits in 0.3 sec requires a channel capacity of only 3.5 kbits/s.

## 5.4 Experiment

The experiments have been carried out on a modern smart card with several security features against side-channel attacks. The identity of the card and the details about the security features are withheld due to a Non Disclosure Agreement (NDA).

In the preparation phase, the electromagnetic signature of 25 instructions were collected. None of the instructions activated the I/O interface of the smart card. Signatures were collected with and without security features against side-channel attacks activated. Spectrum analyzer Advantest 3641 was used. Measurements were done from DC to 60 MHz, with 100 averages, providing signatures of 4206 samples. Typical signatures can be seen in figure 5.2.

The object of the implementation phase is to analyze the 25 signatures collected and to identify frequencies where the emitted energy can be controlled by toggling between two instructions. Therefore, the spectral difference diff(i,j) is of more importance than the shape of the signature, this is shown in figure 5.3. The maximum amplitude difference for all combinations of instructions, when security features are activated, has been plotted in figure 5.4.

These are potential carrier frequencies for the covert channel and the corresponding instructions are used to create the subversive code. As an example, using instruction A and B at frequency $f_c$=11.5 MHz provides an amplitude difference of 10.5 dB. It may be interesting to notice that the peaks around 53 to 57 MHz are introduced by one of the security features.

Three different subversive codes have been designed to test the covert channel and serves to illustrate the potential. It is assumed that the highly skilled insider will be able to create more sophisticated codes. The first code was designed to demonstrate that subversive code can manipulate the energy emitted and that the channel is detectable by a receiver. For this purpose two instructions are executed n times alternately, to create a

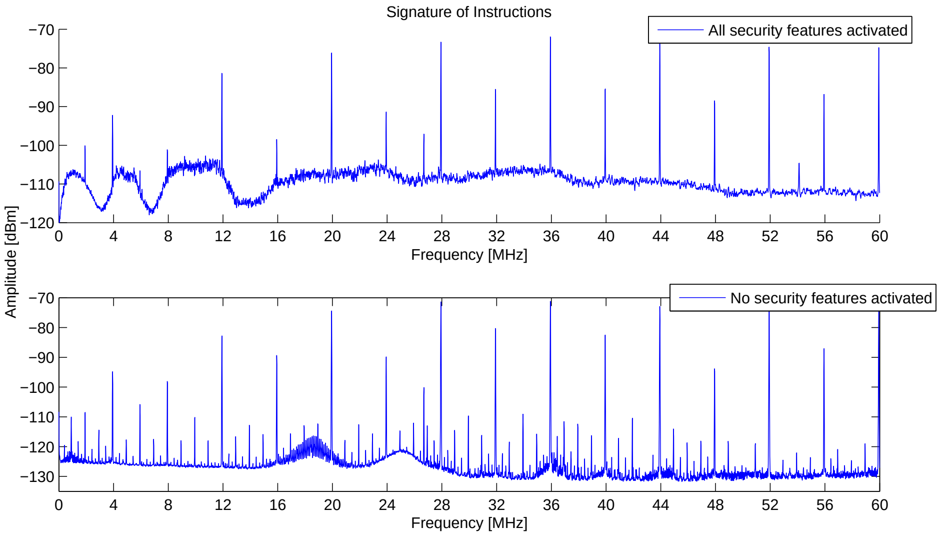

Figure 5.2: Average power spectrum density as signatures of instructions. Instructions executed on card with and without security features activated.

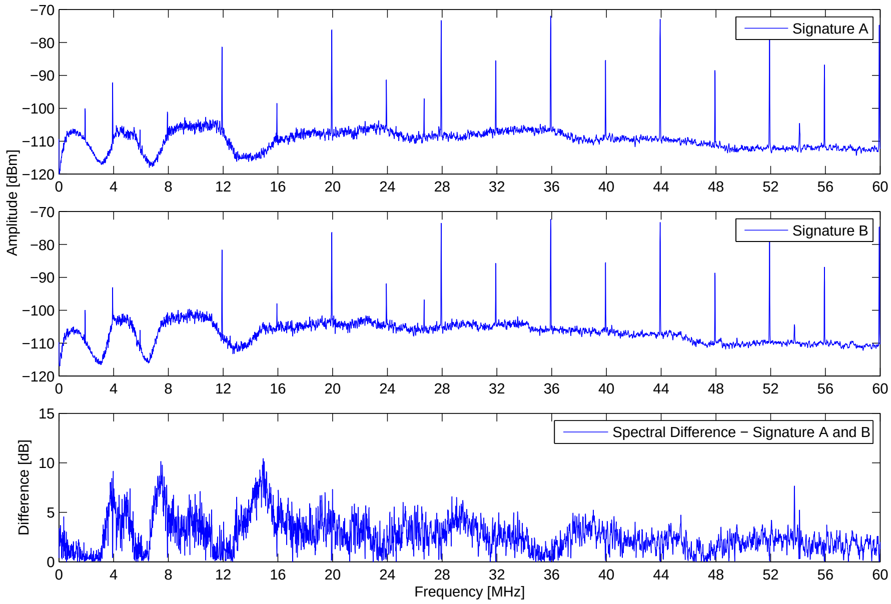

Figure 5.3: Individual signatures of two instruction is shown above. In a covert channel context the spectral difference between them, shown in the lower figure, illustrates our ability to manipulate the emitted energy for various frequencies.

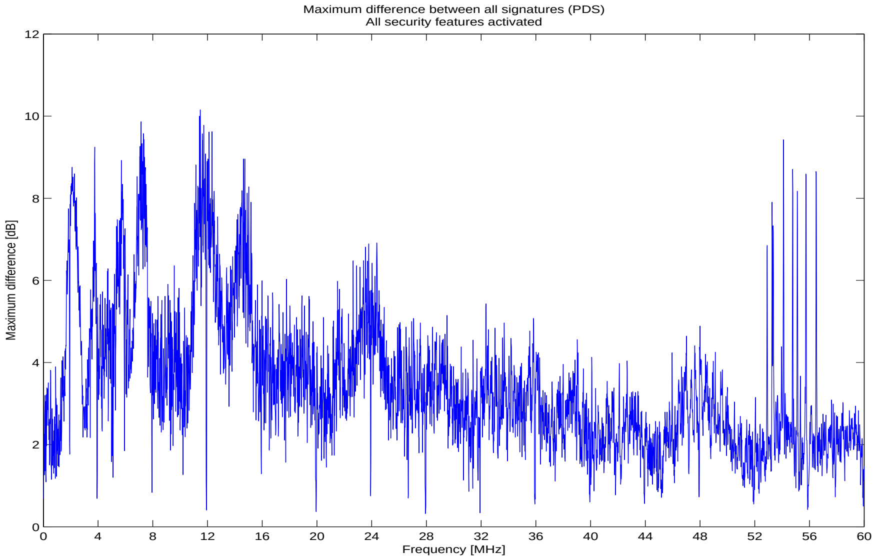

Figure 5.4: The largest spectral difference for all combinations of instructions, illustrates the overall covert channel potential.

pulse train with fundamental frequency dependent on $n$. With $1\mu s$ execution time of each instruction, choosing $n = 500,\ 250\ and\ 125$ results in a fundamental frequency of 1,2 and 4 kHz respectively. This is in the audible range and serves well for a demonstration with an AM receiver. The second code was designed to demonstrate that messages can be transmitted. A short or a long audible tone is used to send morse code (SOS) to the receiver. The third code broadcasts the memory contents of a smart card in an attempt to demonstrate how secret information can be compromised.

Using the ICOM IR-20 receiver with the extendable rod antenna, the audible tones and the morse code are easily detected. On a simple card the tone has been detected 10 meter from the terminal. The modern card is detectable within 1 meter even with security features activated. The covert channel is also detectable on the peaks introduced by one of the security features at 53-57 MHz. The same procedure as with morse code can be used to broadcast the memory content of a smart card to the AM receiver. This is a low rate transmission and work is in progress to improve the transmission rate.

## 5.5 Analysis

The feasibility of the attack has been confirmed through the use of a cheap AM receiver. An important issue left is the rate of information leakage from the smart card, the channel capacity of the covert channel.

The source rate, R, how fast the smart card (source) can transmit information, is given by [7]:

$$R = \frac{H}{T}\ bits/sec \tag{5.1}$$

where H is the average information (entropy) of the source , given by Shannon [18], and T is the time required to send each message. For a binary system with equal probability of sending zero and one, the entropy is H=1. Since each message, $m_1$ or $m_2$, is represented with the waveform resulting from execution of instructions, the execution speed of the microprocessor will set a lower limit on T. Assume a microprocessor architecture that requires multiple of 4 clock cycles per instruction. Choosing two, single cycle, instructions with clock frequency of 4 MHz gives $T = 1\mu s$.

The source rate can now be calculated using (5.1) to find $R = 1\ Mbit/s$. A more realistic value can be obtained by realizing that the subversive code must read the next value and decide if it is a zero or a one, before sending it. By analyzing the flow chart of the testcode, designed to read and broadcast the memory contents of a smart card, an average of 37 clock cycles ($T = 9.25\ \mu s$) is required to send each message. The source rate is then $R = 1/9.25\ \mu s \approx 108\ kbit/s$. This assumes that microprocessor is 100 % devoted to the covert channel during transmission.

The source rate , R, is important when designing a communication channel. Shannon [18] has shown that, for the case of signal plus white gaussian noise, it is theoretically possible to have the probability of error approach zero for a channel capacity of C bits/sec, as long as $R < C$. The equation for C is then

$$C = B \log_2(1 + \frac{S}{N}) \tag{5.2}$$

where B is the channel bandwidth in hertz (Hz) and S/N is the signal-to-noise power ratio (watts/watts) at the input to the receiver.

Using the recorded signature we can estimate B and S/N. B should be the lowest bandwidth in the communication chain. In our experiment this is the receiver with B=15 kHz. Using the amplitude difference in dB from diff(i,j) as the value for S/N can be justified since one of the instructions is close to the noise floor. The experiment suggested a pair of instructions with 10.5 dB difference. With a receiver bandwidth of 15 kHz this gives C=32 kb/s. This is an upper limit for error free communication that may be approached using efficient coding. Different coding schemes are subject to future work, but even a transmission rate of one tenth of C will be sufficient. A key of 1024 bits is transmitted in 32 ms at 3,2 kb/s. Assuming 1 % processor load for the covert channel requires the card to be turned on for 32 seconds, which is not unreasonable in a bank terminal.

The results above are believed to be moderate, as many improvements are possible. One obvious improvement would be to use a receiver with larger bandwidth. Work is in progress to calculate the potential channel capacity given a customized receiver.

## 5.6 Conclusion and Future Work

This article presents a new attack on smart cards. The wireless covert channel attack (WCCA) combines theory from subversion and covert channels with side-channel attack.

Experiments have shown that by manipulating the energy leakage from a smart card we can create a covert channel that will give us access to secret information when the card is used and that the attack will work on modern smart cards equipped with countermeasures against side-channel attacks. It has been estimated that transmitting secret information at a rate of 108 kb/s is possible from the tested card.

Work in progress include designing a receiver to match this transmission rate. This will include M-ary systems, coding schemes and exploiting energy differences in a larger frequency range.

### Acknowledgements

We would like to thank Arne Wold, at Gjovik University College, for helpful discussion and guidance related to the analysis of the results.

## 5.7 Bibliography

[1] AGRAWAL, D., ARCHAMBEAULT, B., CHARI, S., RAO, J., AND ROHATGI, P. "advances in side-channel cryptanalysis, electromagnetic analysis and template attacks". *CryptoBytes 6*, 1 (Spring 2003), 20–32. 50, 88

[2] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., , AND ROHATGI, P. "the em side-channel(s):attacks and assessment methodologies". In *CHES'03* (2003), Lecture Notes in Computer Science, Springer-Verlag. 50, 88

[3] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., AND ROHATGI, P. The em side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 29–45. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_4`. 1, 15, 17, 19, 23, 37, 50, 71, 85, 88, 113, 127

[4] AGRAWAL, D., RAO, J., AND ROHATGI, P. Multi-channel attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), C. Walter, e. Ko, and C. Paar, Eds., vol. 2779 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 2–16. Available from: `http://dx.doi.org/10.1007/978-3-540-45238-6_2`. 1, 17, 37, 50, 88

[5] ANDERSON, R., AND KUHN, M. Tamper resistance: A cautionary note. In *In Proceedings of the 2nd USENIX Workshop on Electronic Commerce (WOEC 96* (1996). 1, 50

[6] CHARI, S., RAO, J., AND ROHATGI, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_3`. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[7] COUCH II, L. W. *Digital and Analog Communication Systems*, 4 ed. Macmillan, 1993. 55

[8] GIANCOLI, D. C. *Physics for Scientists and Engineers*. Prentice Hall, 1989. 14, 50, 88

[9] KOCHER, P. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology* (1996), vol. 1109 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 104–113. Available from: `http://dx.doi.org/10.1007/3-540-68697-5_9`. 1, 26, 50, 71, 88

[10] KOCHER, P., JAFFE, J., AND JUN, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: `http://dx.doi.org/10.1007/3-540-48405-1_25`. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[11] KUHN, M., AND ANDERSON, R. Soft tempest: Hidden data transmission using electromagnetic emanations. In *Information Hiding* (1998), vol. 1525 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 124–142. Available from: `http://dx.doi.org/10.1007/3-540-49380-8_10`. 1, 7, 37, 46, 51

[12] MYERS, P. Subversion: The neglected aspect of computer security. Master's thesis, Naval Postgraduate School, 1980. 30, 31, 51

[13] PEEBLES JR, P. Z. *Digital Communication Systems*. Prentice Hall, 1987. 52, 61

[14] QUISQUATER, J.-J., AND SAMYDE, D. A new tool for non-intrusive analysis of smart cards based on electromagnetic emissions:the sema and dema methods. *Eurocrypt rump session* (2000). 1, 37, 50, 88

[15] Quisquater, J.-J., and Samyde, D. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security* (2001), vol. 2140 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 200–210. Available from: `http://dx.doi.org/10.1007/3-540-45418-7_17`. 1, 7, 15, 16, 17, 21, 26, 37, 50, 88

[16] Quisquater, J.-J., and Samyde, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from: `http://portal.acm.org/citation.cfm?id=1250988.1250994`. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[17] Rao, J., Rohatgi, and Pankaj. Empowering side-channel attacks. Tech. rep., IBM T.J. Watson Research Center, 2001. 1, 7, 16, 17, 37, 50, 88

[18] Shannon, C., and Weaver, W. *The Mathematical Theory of Communication*. University of Illinois Press, 1998. 55, 56

[19] van Eck, W. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security 4*, 4 (1985), 269 – 286. Available from: `http://www.sciencedirect.com/science/article/pii/016740488590046X`. 1, 50

Chapter 6

# *Analysis of the Wireless Covert Channel Attack: Carrier Frequency Selection*[1]

## Abstract

The wireless covert channel attack (WCCA) was suggested to see what it would take to make attacks on smart cards feasible outside a lab environment. Subversive code hidden on the card during manufacturing would manipulate the emitted electromagnetic energy during normal operation. Experiments on modern smart cards confirmed that the attack is feasible and that remote access of secret information resembling wireless skimming is possible. The performance of the channel however, has not previously been analyzed. In this article the performance of WCCA in terms of probability of error is presented. This allows a comparison of different choices of carrier frequency. The data collection necessary is also improved to allow pattern recognition techniques to be employed. This approach looks very promising when selecting multiple carrier frequencies to improve the performance of the covert channel.

## 6.1 Introduction

Combining the efforts of different fields, electromagnetic side-channel attacks, covert channels and subversion, the wireless covert channel attack (WCCA), by Dyrkolbotn and Snekkenes [3] has been suggested. WCCA exploits subversive code hidden on all cards during manufacture, to launch an attack, without physical access, when infected cards are used. Experiments on modern smart cards confirmed that an insider with the opportunity to hide subversive code could potentially broadcast the card's internal secrets to a nearby receiver. However, except for a few estimates of the channel capacity, the performance of the covert channel has not been investigated. Performance in terms of probability of error (i.e. misinterpretation of symbol 0 and 1) on the communication channel can be used to compare implementation choices such as what carrier frequency to choose. The measurements used in [3] is average power density spectrums (PDS) of the emitted power from smart cards, obtained with a spectrum analyzer. Information about the actual waveform, as a function of time, is not available. The maximum signal to noise ratio, within a receivers bandwidth, is used to calculate Shannon's channel capacity in [3]. The probability of error however, is decided by the actual waveform transmitted. Even though the waveform as a function of time is not available, using theory from binary communication systems, we will show how to estimate the probability of error for WCCA based on the average power density spectrums of the received waveform. The probability of error will then be used to evaluate the method of choosing the carrier frequency. We will also show the potential of using pattern recognition techniques to choose multiple carrier frequencies. This requires data to be captured as several individual PDS rather than one average PDS. Methods for choosing the optimal carrier frequencies (lowest possible probability of error) is work in progress.

[1] G.O. Dyrkolbotn. In *Proceedings of Annual Norwegian Computer Science Conference (NIK)*, Tapir, pages 153-163, 2007.

In section two, the wireless covert channel attack (WCCA) is introduced. WCCA is modeled as a binary system and necessary equations to calculate the probability of error is presented. In section three, pattern recognition is introduced along with the revised experiment allowing this approach. In section 4 the results are presented. Finally a conclusion and future work is found.

## 6.2 The Wireless Covert Channel Attack

The wireless Covert Channel Attack (WCCA) by Dyrkolbotn and Snekkenes [3] relies on a highly skilled insider to undermine the security mechanisms by hiding subversive code in the smart card's software (SW). This is done during an early stage (design or compile) of its life cycle (figure 6.1) and will affect all cards produced.

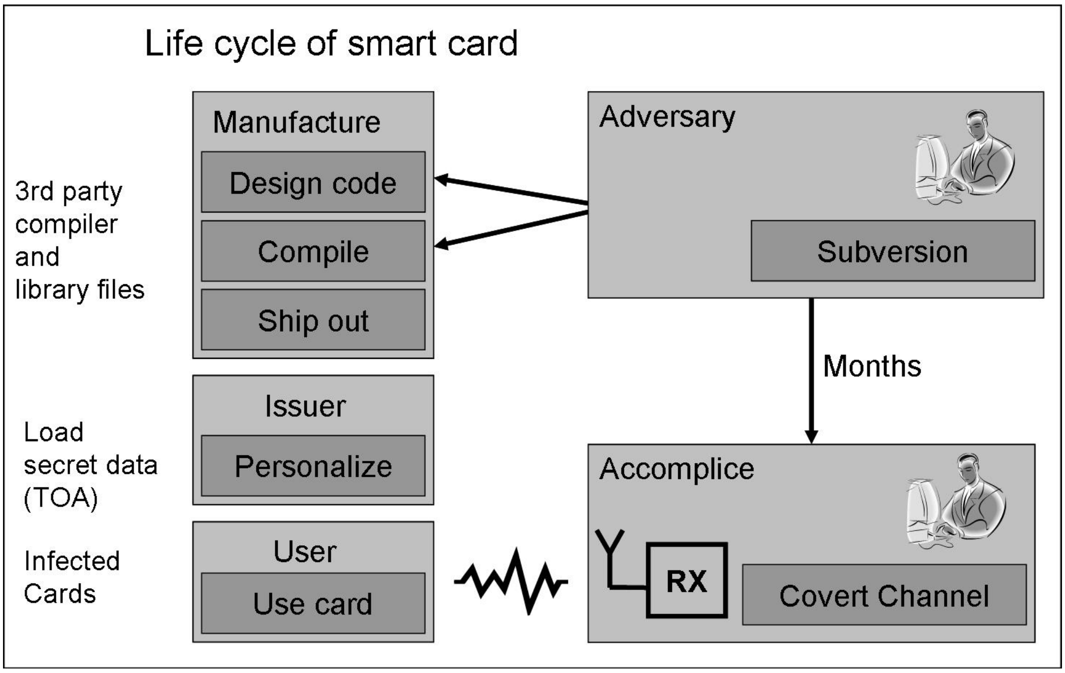

Figure 6.1: Scenario: The adversary hides subversive code on smart cards during an early stage the cards life cycle. Later, secret information is loaded to the card. When the infected card is used, the subversive code is activated and broadcasts secret information to the accomplice, over a covert communication channel.

These infected cards may be used e.g. in the banking industry as credit cards, loaded with personal information (cryptographic key, PIN code, account number etc.) when issued to a customer. The adversary is interested in retrieving the secret, personal information and has an accomplice involved at the use stage of the life cycle. The accomplice is typically somebody with access to a smart card terminal, e.g. a store owner or maintenance personnel. When a manipulated card is inserted into any terminal, by the owner, the subversive code exploits characteristic electromagnetic emanation (signatures) from the microprocessor, during execution of instructions, to broadcast secret information over a wireless covert channel. The success of this attack is ensured by the large number of cards infected. If a whole generation of smart cards to the banking industry is infected, there will be enough cards randomly used in the rigged locations to make the attack worth the effort. Experiments in [3] have shown that by manipulating the energy leakage from a smart card a covert channel can be created that will give access to secret information when the card is used and that the attack will work on modern smart cards equipped with countermeasures against side-channel attacks.

The importance of considering the complete life cycle of a trustworthy device, with high level of security requirements, was also brought to the attention by Pankaj Rohatgi, [6] in an invited talk at CHES 2007. Work has also been done on detecting tampering with a product during manufacturing. Agrawal et al. [1] approaches the problem of how to detect insertion of trojans when outsourcing part of your manufacturing. This work looks promising as a countermeasure to prevent the hiding of the subversive code necessary in WCCA.

### 6.2.1 Optimal Digital Receiver

WCCA can be modeled as a binary communication system. A basic binary system, as explained by Peebles [4], is shown figure 6.2. A message $m$ is sent in each symbol interval, $T_b$. This message $m$ can have of 2 possible values, $m = m_1$ if a binary $0$ is transmitted and $m = m_2$ if a binary $1$ is transmitted. The probability of sending each message is $P_1$ and $P_2$. The waveform actually transmitted is denoted $s(t)$ and $s(t) = s_1(t)$ if $m = m_1$ and $s(t) = s_2(t)$ if $m = m_2$. The input to the receiver, $r(t)$, is assumed to be the sum of $s(t)$ and an added noise $n(t)$. Based on measurements of $r(t)$ the receiver must make a decision on which message $m_1$ or $m_2$ that is received in each symbol interval.

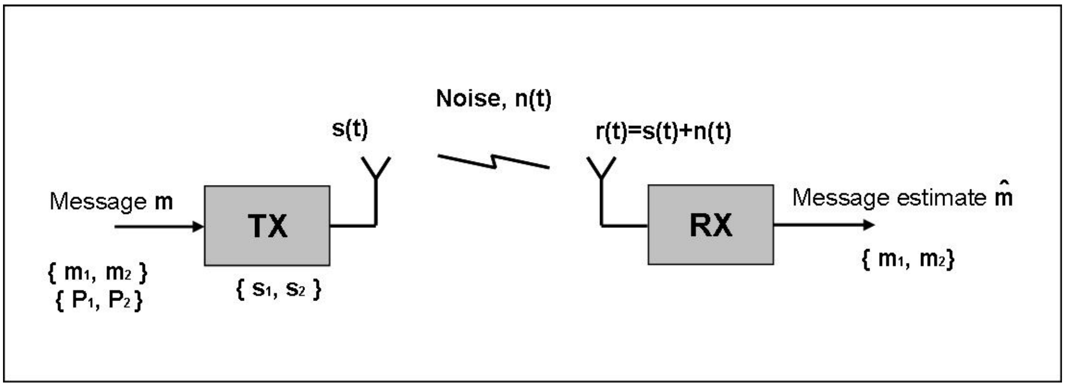

Figure 6.2: Basic Binary System

An optimal digital receiver, according to [4], is one that makes this decision based on maximizing the a posteriori probabilities and in this way minimized the probability of error. The decision rule can be written as:

$$if \quad \int_0^{T_b} r(t)[s_2(t) - s_1(t)]dt > V_T \quad choose\ m_2 \tag{6.1}$$

$$\tag{6.2}$$

otherwise choose $m_1$, where the threshold value $V_T$ is given by

$$V_T = \frac{E_2 + E_1}{2} + \frac{\mathcal{N}_0}{2}(\frac{P_1}{P_2}) \tag{6.3}$$

$\mathcal{N}_0$ is the white noise power density and $E_1$ and $E_2$ the energies of $s_1$ and $s_2$ at the receiver′s input, given by

$$E_i = \int_0^{T_b} s_i^2(t)dt \ , \ i = 1, 2 \tag{6.4}$$

It is assumed that the receiver knows the form of $s_1$ and $s_2$ (coherent). The correlation receiver or matched filter are two possible implementation of the optimum digital system.

### 6.2.2 Waveform and carrier frequency selection

When designing WCCA, the waveform $s(t)$, and a carrier frequency, $f_c$, must be wisely chosen. Usually these parameters are design choices and depends on the communication system implemented. In WCCA the only design freedom is which activity (i.e instruction) that is activated on the microprocessor. The challenge is therefore to select two instructions, that will result in suitable waveforms $s_1(t)$ and $s_2(t)$ and choose the best possible carrier frequency for the given waveforms.The choice made in [3] was to look for one carrier frequency where the energy emitted could be turned on and off, by executing two different instructions.

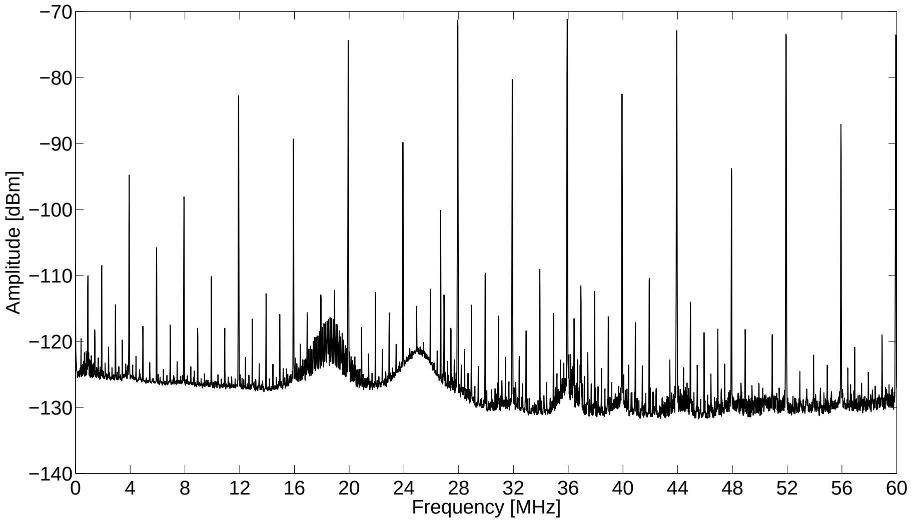

Figure 6.3: The electromagnetic signature of one instruction executed indefinite, represented as the average power density spectrum. 4206 sample points from DC to 60 MHz

Fundamental for choosing the instructions and the carrier frequency are recorded signatures of the electromagnetic energy emitted from the microprocessor. A signature, as defined in [3], is the average power spectrum densities (PDS) recorded from DC to 60 MHz when executing individual instructions, with fixed argument, in infinite loops, see figure 6.3. Averaging is done by the spectrum analyzer and variance of the individual amplitudes is therefore not available. Therefore, the carrier frequency of the covert channel in [3] is chosen as the frequency with maximum average amplitude difference between two signatures (distance of means). An exhaustive search is necessary to find which pair of instructions returns the largest distance of means. These two instructions form the symbol alphabet of the communication channel. For further detail refer to the WCCA article [3].

The obvious question know is: How to evaluate and compare the performance of the chosen symbol alphabet and carrier frequency? A common method in other communication system is to calculate the probability of error for the channel.

### 6.2.3 Performance - Probability of error

For the decision rule in equation 6.1, it can be shown that the probability of error for $P_1 = P_2 = 0.5$ can be written as:

$$P_e = \frac{1}{2} erfc\{\sqrt{\frac{E_1 + E_2 - 2\gamma\sqrt{E_1 E_2}}{4\mathcal{N}_0}}\} \tag{6.5}$$

where $\gamma$ is the correlation between $s_1$ and $s_2$ and is given by:

$$\gamma = \frac{1}{\sqrt{E_1 E_2}} \int_0^{T_b} s_1(t)s_2(t)dt \tag{6.6}$$

Calculating the probability of error using equation 6.5 requires that values for $E_1$, $E_2$, $\gamma$ and $\mathcal{N}_0$ can be obtained from the available PDS. Values for $E_i$ and $\mathcal{N}_0$ are found by taking the average of all amplitudes of the PDS within the receivers bandwidth. The waveforms are assumed to be uncorrelated such that $\gamma = 0$; The probability of error by this method is denoted $P_{eopt}$.

Further simplifications can be made by observing that choosing the carrier frequency according to the distance of mean method, usually finds a frequency where the PDS of $s_1(t)$ has a large peak and the PDS of $s_2(t)$ has a very low peak (close to noise). This basically turns the energy emitted at that frequency on and off. This is also the basic idea of Amplitude Shift Keying (ASK) , also known as on-off keying. The probability of error for ASK is given by:

$$P_e = \frac{1}{2}erfc(\sqrt{\frac{E_1}{2\mathcal{N}_0}}) = \frac{1}{2}erfc(\sqrt{\frac{\epsilon}{2}}) \tag{6.7}$$

where $\epsilon$ is the average energy per bit divided by 2 times the channel noise density. The probability of error by this method is denoted $P_{eask}$. Probability of error is usually plotted vs. $\varepsilon$ for better comparison between systems.

The average PDS in [3] was good enough to show the feasibility of the attack and to find a carrier frequency by calculating the difference of means. A less naive approach, will require to take the variance of the individual amplitudes into consideration. This calls for a revised experiment and opens up for pattern recognition techniques, as explained next.

## 6.3 Pattern Recognition Approach

Pattern recognition is described by Duda et al. [2] as:

> The act of taking in raw data and making an action based on the "category" of the pattern

The design cycle for pattern recognition presented in [2] is illustrated in Figure 6.4. Each step in the design cycle, for the revised experiment, is introduced next.

### 6.3.1 Collect data

Instead of measuring the average PDS as in WCCA [3], several single traces are measured. The amplitude of each sample will then vary due to added noise in the channel. Even though an accurate noise model may be important, the first approach assumes gaussian distributed noise and that the noise at different samples are independent. Each sample can then be considered as a random variable with gaussian distribution $X \sim N(\mu, \sigma^2)$. The signatures used in the WCCA attack are simply the mean, $\mu$, of all the single traces collected in this experiment. The activity considered is, as in WCCA, the execution of individual instructions, with fixed argument, in infinite loops. Terminology from pattern recognition has been adapted. The classes, $\omega_i$, correspond to different activities on the microprocessor that we would like to classify. For this experiment 5 different instructions, and noise were measured.

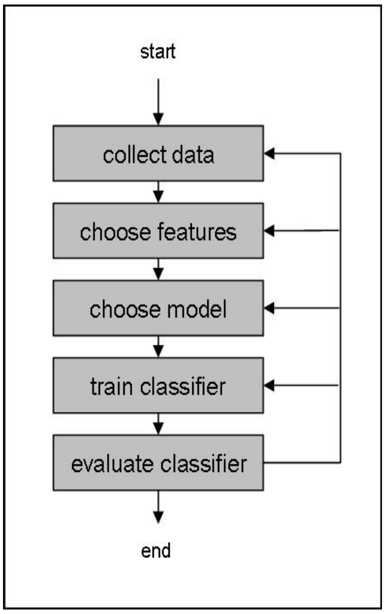

Figure 6.4: The design cycle of a pattern recognition system as described by Duda et al. [2]

| | | |
|---|---|---|
| $class\ \omega_1:$ | $bcf\ 03h, 5$ | $switch\ to\ bank\ 1$ |
| $class\ \omega_2:$ | $goto$ | $empty\ loop$ |
| $class\ \omega_3:$ | $movlw\ 0xaa$ | $move\ binary\ 10101010\ into\ w$ |
| $class\ \omega_4:$ | $nop$ | $no\ operation$ |
| $class\ \omega_5:$ | $sublw\ 0xaa$ | $subtract\ w\ from\ binary\ 10101010,\ store\ result\ in\ w$ |
| $class\ \omega_6:$ | $noise$ | $no\ activity\ (power\ off)$ |

Each sample (i.e. frequency) of the measured traces is considered as a feature, such that each class $\omega_i$ is represented by $d = 3006$ features in a column vector $X_i = \{x_{i1}, x_{i2}, ..., x_{id}\}$. A total of $n = 440$ measurements was collected of each class. Each measurement resulted in a trace of 3006 features from DC to 60 MHz, as can be seen in figure 6.5. Comparing this trace to a trace from the original WCCA (figure 6.3), notice that the noise floor is about 20 dB higher and only 3006 samples are available. This is a limitation of the spectrum analyzer used recently (Rhode and Schwarz FSL-6)

### 6.3.2 Choose Features

A feature is the carrier frequency for the covert channel. How to choose the carrier frequency depends on the optimization criteria. Maximizing the range should look for the best signal to noise ratio, but this does not necessarily give the lowest probability of error. A large number of distance measures exist, see [5]. For the one dimensional case (one carrier frequency), considered here, it is computational feasible to calculate the performance of all 3006 possibilities. First the carrier frequency will be chosen by the, naive difference of means, approach of WCCA. The result will be compared to results from an exhaustive search for the lowest probability of error.

For the pattern recognition case, it is straight forward to extend to more than two instructions (M-ary symbol alphabet) and more than one carrier frequency. This comes at a complexity cost. The curse of dimensionality quickly becomes a reality and smarter methods for feature selection must be designed. This is work in progress.

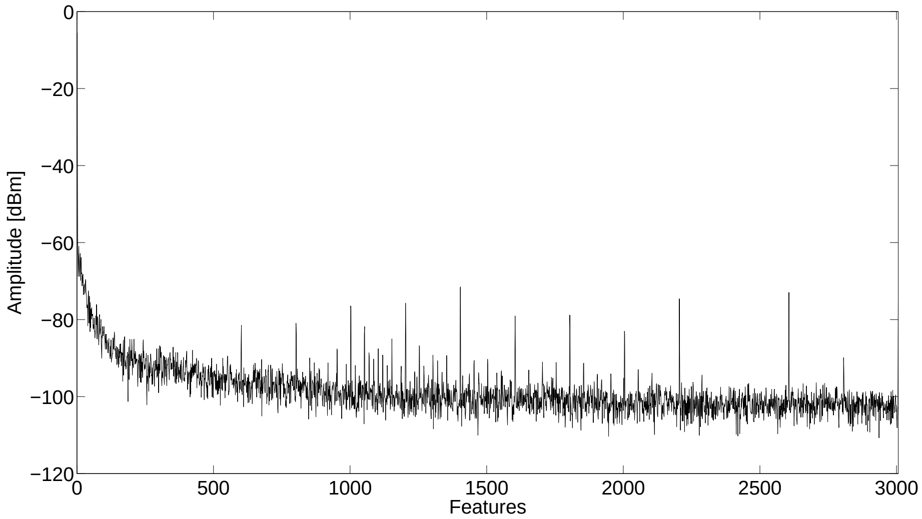

Figure 6.5: One trace of class 1 (bcf), represented by 3006 features. Each feature is the amplitude in dBm of frequencies from DC to 60 MHz

### 6.3.3 Choose Model

The model used is simply: Activity on a microprocessor can be classified based on energy emitted on a small finite set of frequencies. For the special case of WCCA this can be states as: Execution of two different instructions on a microprocessor can be classified based on the energy emitted on one frequency.

### 6.3.4 Train Classifier

In this approach Bayes classifier is used. Fundamental for the classification and the calculation of the decision boundaries are the a priori probabilities $P(\omega_i)$ and the conditional densities $p(x|\omega_i)$. The process of estimating $P(\omega_i)$ and $p(x|\omega_i)$, based on available data, is called training.

Bayes principle minimizes the probability of error by choosing the maximum a posteriori probability $P(\omega_i|x)$, related to a posteriori probability and density function by Bayes rule [2]. The case considered in this article is two classes and one feature. With 0/1 loss function (i.e wrong decision weighted zero, right decision weighted one) the decision rule can be written as:

$$if \quad p(x|\omega_1)P(\omega_1) > p(x|\omega_2)P(\omega_2) \quad choose\ \omega_1 \tag{6.8}$$

otherwise choose $\omega_2$. Assuming the distribution of observations X to be gaussian, $X \sim N(\mu, \sigma^2)$ and a priori probabilities to be equal, P($\omega_1$)=P($\omega_2$), the decision rule reduces to:

$$if \quad \frac{(x-\mu_1)^2}{2\sigma_1^2} - \frac{(x-\mu_2)^2}{2\sigma_2^2} < \log\frac{\sigma_2}{\sigma_1} \quad choose\ \omega_1 \tag{6.9}$$

otherwise choose $\omega_2$.

This is equivalent to, choosing $\omega_1$ if $h(x) < 0$ and $\omega_2$ otherwise, where $h(x)$ is given by:

$$\begin{array}{ll} h(x) = ax^2 + bx + c & where \\ a = \sigma_2^2 - \sigma_1^2 & second\ order\ (quadratic)\ term \\ b = 2(\mu_2\sigma_1^2 - \mu_1\sigma_2^2) & first\ order\ (linear)\ term \\ c = \sigma_2^2\mu_1^2 - \sigma_1^2\mu_2^2 - 2\sigma_1^2\sigma_2^2 \log\frac{\sigma_2}{\sigma_1} & constant \end{array} \tag{6.10}$$

The roots of $h(x) = 0$ defines our decision boundary between decision regions. Two roots means that the decision regions are not simply connected.

The goal of training is to take collected data of known classes and calculate a decision boundary that can be used on future measurements of unknown classes. Since only 440 traces are available, 220 traces are randomly taken out to be used for testing. Based on the remaining 220 data sets, the mean, $\mu$ and the variance $\sigma^2$ are estimated, using maximum likelihood estimators. Then the constants $a$, $b$ and $c$ of $h(x)$ are calculated according to (6.10) and stored. The decision boundary , and A posteriori densities $P(\omega_1|x)$ and $P(\omega_3|x)$ for class 1 and 3, and feature 1070 (i.e carrier frequency 21,3 MHz), using this method are shown in figure 6.6.

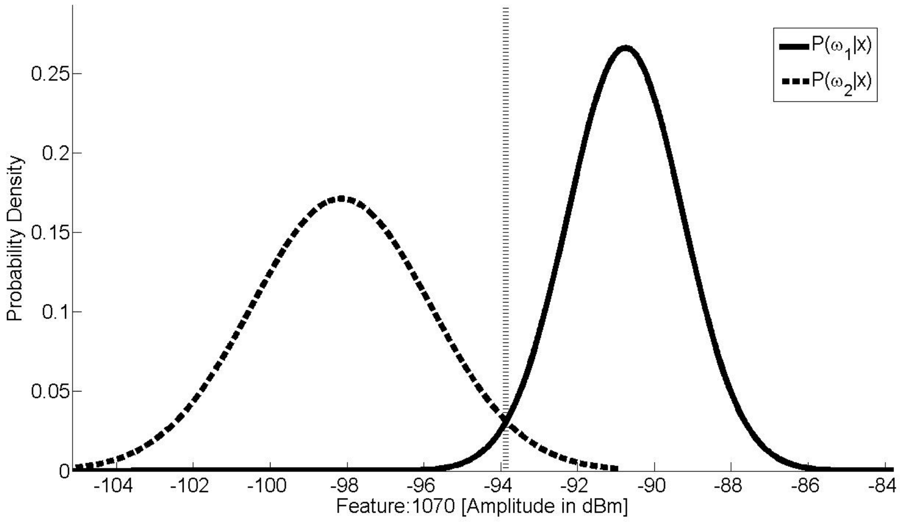

Figure 6.6: A posteriori probability densities for classes 1 and 3. Gaussian distribution assumed and maximum likelihood estimator used to estimate parameters. The decision line that minimizes the probability of error is located at the interception between the densities

### 6.3.5 Evaluate Classifier

For each trace reserved for testing, the set of amplitudes for the chosen feature (i.e carrier frequency) is used to calculate $h(x)$, using $a$, $b$ and $c$ from training. The trace is then classified according to $h(x) > 0$ or $h(x) < 0$. Since the correct class of every trace is known, the probability of error is simple the ratio of wrong classifications to the total number of test traces. The random split of available data does influence the result a little bit. The final probability of error, denoted $P_{eBayes}$ is therefore the mean of repeating the splits, training and testings $10$ times.

## 6.4 Results

### 6.4.1 WCCA Approach

First the carrier frequency is chosen according to the difference of means method suggested by [3]. The largest difference between any of the signatures was found between the goto instruction (class 2) and the nop instruction (class 4). The difference was 10,4 dB for carrier frequency $f_c = 23,3$ MHz. Since the resolution of the PDS's are about 19 kHz, a narrowband receiver with bandwidth of 15 kHz would receive one carrier frequency at the time. Under this assumption, using equation 6.5 and 6.7 gives the following results:

| $s_1(t)$ | $s_2(t)$ | $f_c$ [MHz] | $P_{eopt}$ | $P_{eask}$ |
|---|---|---|---|---|
| goto | nop | 23,3 | 0.0078 | 0.0103 |

The probability of error for 50 carrier frequencies is shown in figure 6.7. The two curves indicate what probability of error that can be expected. The horizontal line indicates the probability of error of the carrier frequency, $f_c = 23,3$ MHz, chosen by the maximum difference of means method. The result shows, not surprising, that the method to choose carrier frequency in WCCA is not optimal in terms of low probability of error. Several of the 49 carrier frequencies ranked after 23,3 MHz return a lower probability of error. The lowest probability of error in Figure 6.7 is $P_{eopt} = 10^{-9}$.

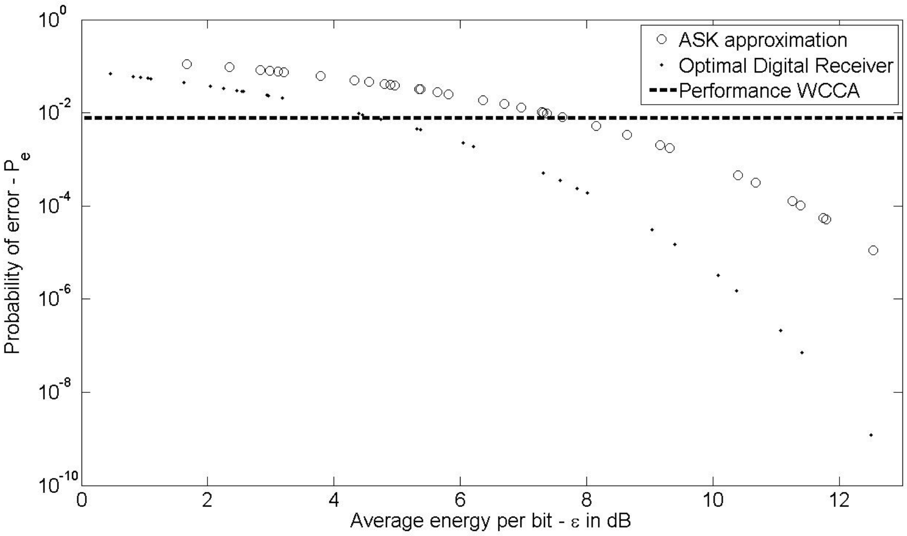

Figure 6.7: The probability of error calculated for the 50 carrier frequencies, using equation 6.7 for ASK approach, equation 6.5 for the optimal digital receiver approach. The horizontal line indicates the probability of error for the carrier frequency suggested by difference of means method

### 6.4.2 Pattern Recognition approach

The results in figure 6.7 clearly show that the approach in WCCA to choose the carrier frequency is not optimal in terms of probability of error. In the one dimensional case (i.e. one carrier frequency), it is computational feasible to run through all the possible carrier frequencies to choose the one returning the lowest probability of error. Doing this actually returns 9 frequencies with $P_{eBayes} = 0$ and 21 with $P_{eBayes} < 0.001$, all perform better than the frequency chosen by the WCCA method, which returned $P_{eBayes} = 0.0023$. The perfect classification by 9 features is probably due to a limited test set of 220 traces.

Feature 1404 (i.e. carrier frequency $f_c = 28,0$ MHz) is an excellent example of the difference between the two approaches. The amplitude difference between any classes for feature 1404 is less than 2 dB. The WCCA approach is looking at the amplitude difference of two classes in dB and will therefore dismiss feature 1404 as to low. It turn out though that there is a very small variance of the amplitude at feature 1404 such that for a limited set of 220 traces return $P_{eBayes} = 0$.

#### 6.4.2.1 Higher dimensions

An obvious improvement of the attack would be to exploit more of the available energy by taking advantage of more than one carrier frequency and use a symbol alphabet larger than 2 instructions.

WCCA only uses one out of 3006 carrier frequencies. It should be possible to achieve better performance by utilizing more of the available energy emitted from the card. It is therefore natural to look at what happens if more than one carrier frequency is used. Pattern classification easily extends to several features. Figure 6.8 shows how two dimensions can be used to reduce $P_e$. Classifying classes 2 and 3 based on feature 1170 alone gives $P_{eBayes} = 0.2$. Using feature 1170 together with 1019 gives $P_{eBayes} = 0.02$. However, unless careful choices are made, the classification can be made worse by increasing the dimensionality. The question is therefore: What is the best way of choosing features? There is no trivial answer to this question and the challenge is work in progress.

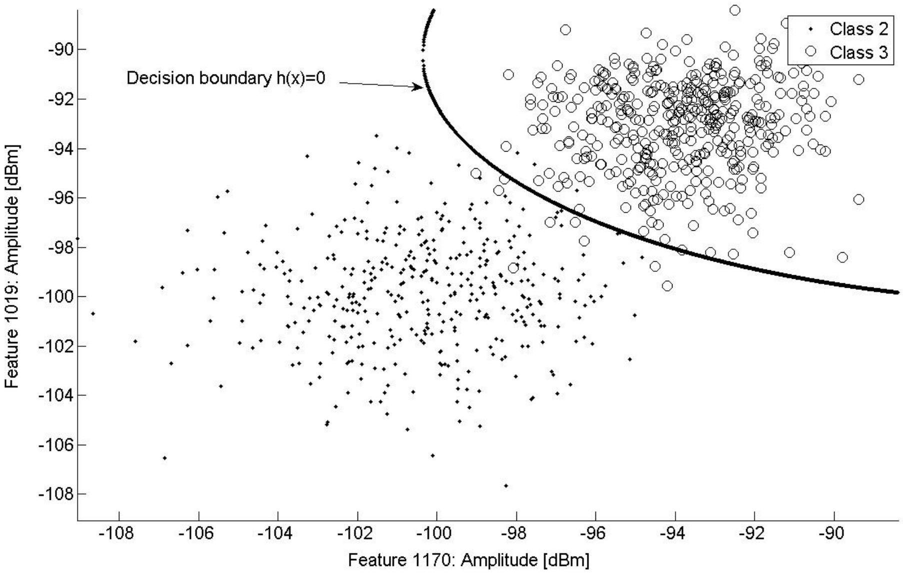

Figure 6.8: Two dimensional classification performs better if features are carefully chosen

Finally it is possible to extend the symbol alphabet. For a classifier this is straight forward. Preliminary results with Bayes classifier, using 5 instructions can be seen in the table below.

The results show that the probability of error can be made very small, by using enough carrier frequencies, if you select them correctly. As mentioned before, this is work in progress.

| 5 Instructions | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Feature | 1404 | 1804 | 1153 | 1136 | 1019 | 1170 | 1604 | 1203 |
| Number of features: d | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Performance: d features | 0,239 | 0,110 | 0,051 | 0,019 | 0,007 | 0,003 | 0,002 | 0,001 |

Table 6.1: Classifying 5 instructions based on an increasing number of features (i.e carrier frequencies)

## 6.5 Conclusion and Future Work

The wireless covert channel attack relies on proper choices of instructions and carrier frequency to implement a covert communication channel. The performance of the choices made, can be evaluated in terms of probability of error. Theory from a basic binary system has been used to calculate the probability of error, based on available average power density functions. The Results show that it is possible to achieve probability of error as low as $10^{-6}$. It is also obvious, if not surprising, that just looking at the distance of means, without taking the variance into consideration is far from optimal. That said, when only one carrier frequency is used, it is feasible to do an exhaustive search for the optimal frequency. The performance of the channel can easily be improved, at the cost of complexity, by increasing the symbol alphabet and using more than one carrier frequency. A pattern recognition approach, as illustrated by Bayes classifier in this article, looks like a promising tool to explore this improvement. Work is already in progress on how to optimally (i.e. minimizing probability of error) choose the number of instructions to use and the number of carrier frequencies to use.

## 6.6 Bibliography

[1] AGRAWAL, D., BAKTIR, S., KARAKOYUNLU, D., ROHATGI, P., AND SUNAR, B. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy* (may 2007), pp. 296 –310. 31, 46, 61, 71, 85, 99, 113, 127

[2] DUDA, R., HART, P., AND STORK, D. *Pattern Classification*. John Wiley and Sons, Inc, 2001. 8, 26, 28, 29, 30, 63, 64, 65, 75, 78, 90, 91, 94, 124, 137, 141

[3] DYRKOLBOTN, G. O., AND SNEKKENES, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: `http://dx.doi.org/10.1007/11935308_18`. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[4] PEEBLES JR, P. Z. *Digital Communication Systems*. Prentice Hall, 1987. 52, 61

[5] PERLIBAKAS, V. Distance measures for pca-based face recognition. *Pattern Recognition Letters 25*, 6 (2004), 711 – 724. Available from: `http://www.sciencedirect.com/science/article/pii/S0167865504000248`. 64

[6] ROHATGI, P. Trustworthy hardware. In *Keynote speaker at Cryptographic Hardware and Embedded Systems - CHES* (2007), IBM T. J. Watson Research Center. 30, 46, 61

Chapter 7

# *Electromagnetic Side-Channel: A Comparison of Multi-class Feature Selection Methods*[1]

## Abstract

When a microprocessor executes its program, electromagnetic signals are generated as a consequence of the power consumption associated with transistor transition. The intercepted electromagnetic signal can be used to reveal the contents of program and/or data memory of the microprocessor, exploited in a variety of attacks [12] [1] [2] [7]. Common for these approaches is the need to map an unknown intercepted signal (trace) to a known set of activities on the microprocessor, often based on thousand of available variables. It is unlikely that all variables contain relevant information and a large number of variables requires a lot of computational resources (storage and execution time). It is therefore of interest to reduce the number of variables by selecting lower dimensional subsets or mapping.

This article gives an introduction to feature selection methods as well as suggestions as to how to handle a multi-class case. We use the wireless covert channel attack [7] to perform an application specific comparison between a selection of these methods. The different methods are compared by classification error and computational cost. Our results suggest use of the Greedy Forward Selection method, if preprocessing time is of no concern. Otherwise, ranking based methods give fair results. Experimental results also show that, in a wireless covert channel application, it is possible to classify microprocessor activity (i.e instruction executed), with a very low probability of error, given the energy emitted on only a few selected carrier frequencies. Implementation of a M-ary, multi carrier covert channel with a reasonable error rate is therefore feasible.

## 7.1 Introduction

It is well known that microprocessors leak information about their activity through side-channels (timing, power, electromagnetic) and in cryptanalysis, this has been exploited in several attacks [12] [11] [4]. The Wireless Covert Channel Attack (WCCA) [7] has shown that in manipulating the electromagnetic emanation, an adversary can construct a covert broadcast channel. This channel can be used by the adversary to broadcast information (i.e secrets) from the microprocessor. The attack relies on the ability to detect chosen activity on a microprocessor by its corresponding electromagnetic signal. Analysis of the WCCA has shown that the energy received by a small number of narrow band receivers is enough to identify the activity on a microprocessor [6]. Expressed as a pattern recognition problem this can be stated as: The adversary wants to detect and identify a set of classes (i.e. a set of activities on the microprocessor) based on a set of variables (i.e. the energy on selected carrier frequencies) of the intercepted electromagnetic emanation. Challenges that arise are typical for pattern recognition systems: How do we select the variables (carrier frequencies)? How many variables do we included in the classification process? Other attacks,

[1] G.O. Dyrkolbotn and E. Snekkenes. In *Proceedings of Signal Processing, Pattern Recognition and Application (SPPRA)*, ACTA Press, track 643-804, 2009.

such as the template attack [4], are also based on pattern classification techniques, and face similar problems. Open issues include [3]: How to select the relevant samples (variables) and how many samples are needed to construct the attack? These challenges are the focus of feature selection techniques. According to [9], the objective of feature selection is three-fold:

> *Improving the prediction performance of a classifier, providing faster and more cost-effective classifiers, and providing a better understanding of the underlying process that generated the data.*

Assume that during microprocessor activity, the electromagnetic emanation is measured to form a p-dimensional trace, $\vec{x} = \{x_1, \cdots, x_p\}$, where $p$ is the number of variables measured. Some of the $p$ variables of $\vec{x}$ may not contain relevant information about the underlying phenomenon. Therefore, the challenge is to find a lower dimensional representation $\vec{y} = \{y_1, \cdots, y_q\}$ where $q < p$, that still contains the relevant information. The selected variables can, according to [8], be a subset or a mapping of the original $p$ variables of the trace. The selected variables are often referred to as features of the trace. Comparing the effectiveness of the different feature selection methods can be based on: (i) The number of variables needed, (ii) The amount of time required or (iii) The error rate achieved. Choice is highly application dependent.

As a further motivation for feature selection techniques, consider the following complexity considerations: The computational resources (e.g computational time) needed to choose $q \subset p$ variables depends on the method used. The simplest (smallest complexity) method is to choose $q$ variable by random from $p$. The computational cost of this method is constant, $\mathcal{O}(1)$. The most expensive (largest complexity) method is an exhaustive search of all combinations of $q$ out of $p$ variables. The number of combinations is given by the binomial coefficient: $\binom{p}{q} = \frac{p!}{(p-q)!q!}$. Since the number of variables, $q$, to include is unknown, all subset sizes $q = \{1 \cdots, p\}$ must be evaluated. The number of combinations is given by: $\sum_{q=1}^{p} \binom{p}{q} = 2^p$. Even a small number of (say 100) variables cumulates to approximately $1,3 \cdot 10^{30}$ combinations. With thousands (3006 in this paper) variables it become infeasible to perform an exhaustive search of $2^p$ combinations. All combinations up to 10 out of 3006 alone reaches approximately $1,6 \cdot 10^{28}$ combinations that must be evaluated. The computational cost of an exhaustive search is bound by $\mathcal{O}(2^p)$. It is therefore important to find methods with a reasonable tradeoff between computational cost and classification accuracy.

This article compares the multi-class classification error and computational cost of different feature selection methods, applied to the Wireless Covert Channel Attack [7]. The introduction to different feature selection methods and the evaluation method used are, however, general and could easily be applied elsewhere (i.e side-channel attacks).The phenomenon studied is electromagnetic emanation from microprocessors as they execute specific programs, and involves classification of several microprocessor activities based on up to 10 out of thousands of variables.

The following methods are compared: (i) Single Variable Classifier, (ii) Difference of Means, (iii) Variance, (iv) Mahalalobis Distance, (v) Greedy Forward Selection (GFS), (vi) GFS with preprocessing filters (bins, harmonics, hybrid), (vii) Principal Component Analysis and (viii) Fischer's Linear Discriminant.

Section 7.2 presents a model of the phenomenon and the experimental setup. Section 7.3 presents the different feature selection methods. In section 7.4, the results are compared and discussed. Finally, a conclusion is drawn and future work outlined in section 7.5.

## 7.2 The Phenomenon

A microprocessor typically includes; memory (program memory, data memory), CPU , I/O and busses. A microprocessor has a functional activity which is to transform a set of input

| Operation Sequence | Estimated Operation Seq. | |
|---|---|---|
| | with error | |
| $o$ | $\hat{o}$ | $H_d$ |
| add | add | 0 |
| jmp | sub | 1 |
| goto | nop | 1 |
| D: Hamming Distance, $H_d$ | | 2 |

Table 7.1: Hamming distance of true operation sequence $o$ and estimated operation sequence $\hat{o}$.

bits to a set of output bits. The functional activity is determined by its program, the contents of data memory and internal states. The CPU is responsible for fetching, decoding and executing the instructions, one by one. Each instruction is typically executed using a set of basic operations (e.g transfer, arithmetic, logic and shift). This is usually controlled by a "square wave" clock. The sequence of basic operations necessary to execute the program is called the execution sequence, $\vec{o} = \{o_1, o_2, \cdots, o_N\}$. It is well known that the majority of the power consumed during the execution sequence depends on the number of gates that change state. This is related to what bit values are processed and moved on various busses. The power consumed by each operation is, therefore, a function of the instruction (opcode and operand), the data, the address in memory and the prior state (upstate) of the microprocessor. This relationship can be exploited by an adversary to deduce the contents of program or data memory, otherwise kept secret. The adversary may have knowledge of, or even be able to, chose part of the operation sequence in order to reveal specific details (e.g run a known cryptographic algorithm for an unknown value of the cryptographic key or alternate the execution of two different instructions [7]). The intercepted signal (trace) will have a signal part and a noise part. Properties (features) of the trace, $x$, are then extracted and used to compute an estimate of the execution sequence, $\hat{o}$. The estimated operation sequence, together with known or chosen memory contents, can then be used to reveal the contents of program or data memory that the adversary is targeting. If $o$ is the actual but unknown execution sequence and $\hat{o}$ is the estimated execution sequence, this can be stated as a pattern recognition problem:

***Based on properties, x, of the intercepted electromagnetic emanation, compute $\hat{o}$ such that the distance $D(g, x, o, \hat{o})$ between $o$ and $\hat{o}$ is minimum for a given classifier $g$ and the properties $x$.***

Hamming distance ($H_d$) is used to express the distance $D$ between the true execution sequence $o$ and the estimated execution sequence $\hat{o}$. Hamming distance is an expression of the number of estimated steps (i.e instructions) in the sequence that are different, i.e. an expression of how many instructions that were estimated incorrectly. Let instructions estimated correctly be labeled $0$ and instructions estimated incorrectly labeled $1$. Table 7.1 shows an example of an operation sequence with three instructions.

The Wireless Covert Channel Attack (see [7] for details) manipulates the electromagnetic emanation from microprocessors by executing different instructions in loops. In its simplest form, two instructions (i.e classes) are used with a single carrier frequency (variable/feature). In [6] is it stated that in a pattern recognition context, it is straight forward to extend to M-ary symbol alphabet (multi class) and multiple carrier frequencies (features). Choosing multiple features comes at a complexity cost. The curse of dimensionality quickly becomes a reality and smarter methods for feature selection must be designed. This makes WCCA a good case for feature selection methods and is therefore used in this article.

This choice restricts the operation sequence to instructions with fixed argument. The objective for the classifier is therefore to recognize the instruction executed. This means that the classifier either estimated $o$ correctly ($H_d = 0$) or falsely ($H_d = 1$). When done for a large set of test data, the classifiers accuracy can be calculated as the ratio of correct classification to the total number of classification.

### 7.2.1 Experimental Setup

Data used to compare the different feature selection methods were collected the following way: A program was executed on a smart card microprocessor (Goldcard, PIC 16F84A), the program executed the same instruction with fixed argument $100$ times followed by a ”goto” instruction to repeat the program indefinitely, a large number of consecutive instructions (limited by the program memory) reduces the effect of the ”goto” instruction.

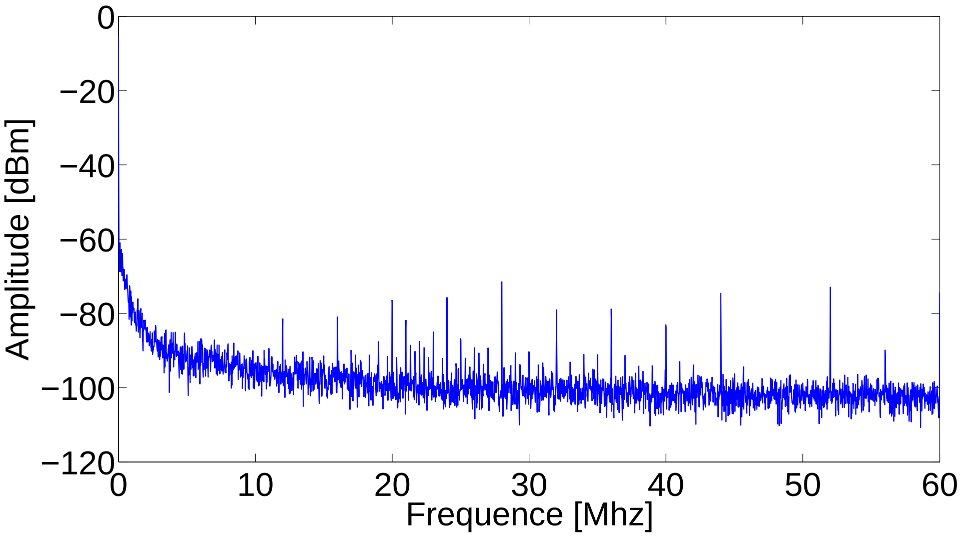

Figure 7.1: Unprocessed data: One measurement of instruction BCF. 3006 datapoints from DC to 60 Mhz, each representing the emitted power in dBm.

The resulting electromagnetic emanation was measured with a transducer and a spectrum analyzer (FSL-6) from DC to 60 MHz with 10 kHz resolution bandwidth (RBW). This results in a power density spectrum consisting of 3006 power measurements in dBm, as can be seen in figure 7.1. A sweep time on the spectrum analyzer of $0,6s$ means that during each of the 3006 power measurements, approximately 200 executions of each instruction is carried out. Harmonics of the 4 MHz clock used on the microprocessor are clearly visible. The fundamental frequency at 4 MHz is buried in the noise, probably due to low sensitivity of the transducer at lower frequencies. Several peaks are visible between these harmonics in the $20 - 30Mhz$ region. This is emanation related to the program executed.

Five different instructions were used, as shown in table 7.2, each measured $440$ times. In some application domains, e.g side-channel attacks, one could argue that this number should be larger. In a covert channel context, however, 5 classes are sufficient to evaluate the relative performance of the feature selection methods and confirm that a symbol alphabet greater than two (multi-class) is feasible, while keeping a low error rate.

The available data (i.e traces) can be expressed as: $X_{ijk}$, where $i = \{1, \cdots, 5\}$ is the instruction (i.e $c = 5$ classes), $j = \{1, \cdots, 3006\}$ is the frequency $f$, also called the variable, and $k = \{1, \cdots, 440\}$ is the measurement number. The frequency is given by $f = (j -$

| | |
|---|---|
| Instruction 1 ($\omega_1$): | bcf 03h,5 |
| Instruction 2 ($\omega_2$): | goto |
| Instruction 3 ($\omega_3$): | movlw 0xaa |
| Instruction 4 ($\omega_4$): | nop |
| Instruction 5 ($\omega_5$): | sublw 0xaa |

Table 7.2: A sample of the instruction executed by the microprocessor

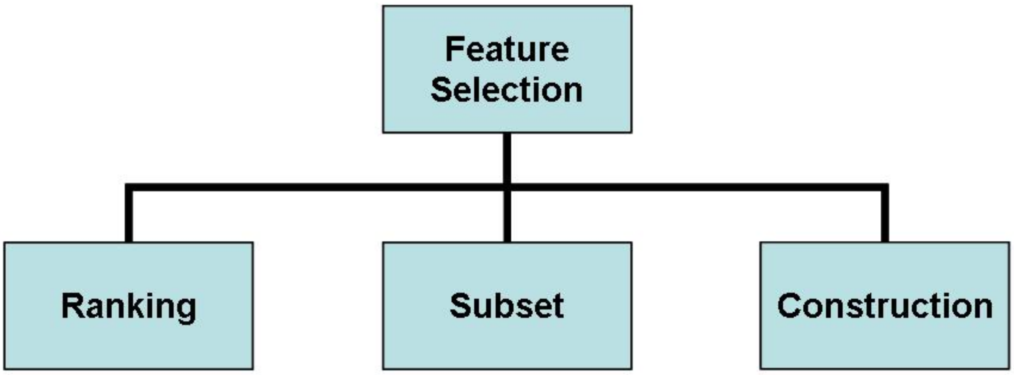

Figure 7.2: Feature Selection techniques can be divided into three groups: variable ranking, subset selection or feature construction.

1) $* \frac{60 \cdot 10^6}{3005}$. The data, $X_{ijk}$ therefore consist of $440 \cdot 5 = 2200$, p-dimensional ($p = 3006$) individual observation traces $\vec{x}$, as seen in figure 7.1.

### 7.2.2 Classification

All the feature selection methods presented in this article are tested with Bayesian quadratic classifier [5]. The average classification accuracy, $\bar{acc}$, of 100 random split of the data set with $k = 0.5$ (50% for training and 50% for testing) is used. Probability of error , $P_e = 1 - \bar{acc}$ is used for plotting and comparison.

## 7.3 Feature Selection Methods

Feature selection techniques can, according to [9], be organized as illustrated in figure 7.2. An explanation of each group and the methods used in this paper follows.

### 7.3.1 Variable Ranking Methods

Variable ranking methods use a score function $S$ (e.g distance measure) to rank individual variables [9]. This is computational efficient, since $p$ variables only requires $p$ computations. The individual scores for each variable are sorted and $q < p$ extremities are selected. The drawback is that correlations between variables are disregarded. The method may therefore ignore low scoring variable that work excellent together or include high scoring variables, with negligible contribution, due to high correlation with already selected variables. The simplicity of ranking based methods will, however, make them a good starting point.

#### 7.3.1.1 Single Variable Classifier:

Single Variable Classifier (SVC) ranks individual variables by their performance (i.e. error rate, $P_e$) with a given classifier (e.g. Bayes). The score function for variable $j$, given a classifier with discriminating function $g$, can be written as:

$$S(j) = P_e(g, j) \tag{7.1}$$

The $q$ variables with the lowest probability of error, $P_e(g, j)$, are selected.

#### 7.3.1.2 Difference of Mean:

Difference of Mean (DOM) is a very simple method also used in early DPA attacks [12] and WCCA [7]. The mean of all observations $k$, for class $i$ and variable $j$ is given by: $\mu_{ij} = \frac{1}{n}\sum_{k=1}^{n} X_{ijk}$. The score function for variable $j$ between class $m$ and $n$ can be written as:

$$S_{mn}(j) = abs(\mu_{mj} - \mu_{nj}) \tag{7.2}$$

For c=2 classes, there is only one score, $S_{12}$, for each variable. A simple sort will reveal the highest scoring variables. For $c > 2$ there are $c(c-1)/2$ scores for each variable that must be combined into a single score. Two approaches have been used in this paper: **Mean Difference of Means** that calculates the mean of the c(c-1)/2 distance scores, and **MaxMin Difference of Means**, that finds the minimum of the c(c-1)/2 scores and selects the $q$ variable with the largest minimum distance. This ensures a minimum distance between any classes.

#### 7.3.1.3 Variance:

Assume that variables with a small variance within each class are less likely to overlap than those variables with a larger variance. Under this assumption, selecting variables with a small variance could serve as a ranking method. The variance of variable $j$ for class $i$ is given by $\sigma_{ij}^2 = \frac{1}{n-1}\sum_{k=1}^{n}(X_{ijk} - \mu_{ij})^2$. The score function for variable $j$ is chosen as the mean of the variance $\sigma_{ij}^2$ for all $c$ classes, given by:

$$S(j) = \frac{1}{c}\sum_{i=1}^{c} \sigma_{ij}^2 \tag{7.3}$$

The $q$ variables with the lowest mean variance, $\min S(j)$ are selected.

#### 7.3.1.4 Mahalanobis:

Intuitively, it makes sense to take into consideration both the between-class distance and the within-class distance (spread). This is referred to as one-dimensional Mahalanobis distance, or Fischer's criteria [9]. The score function, $S_{mn}(j)$, for variable $j$ between class $m$ and $n$ can be written as:

$$S_{mn}(j) = \frac{(\mu_{mj} - \mu_{nj})^2}{\sigma_{mj}^2 + \sigma_{nj}^2} \tag{7.4}$$

As with difference of means this gives $c(c-1)/2$ scores for each variable. Two multi-class approaches, as described with difference of mean, will be used, **Mean Mahalanobis Distance** and **MaxMin Mahalanobis Distance**.

### 7.3.2 Variable Subset Selection Methods

Another group of methods is variable subset selection [9]. It can easily be shown that variables which are useless by themselves, can be useful when used together with other variables [9]. Therefore, the idea is to select subsets of variables that work well together, rather than looking at their individual performance. The methods can be divided into wrappers, filters or embedded methods [9]. Wrappers selects subsets of variables and uses a classifier to evaluate the performance of each subset selection. Filters are similar, but are used as a pre-processing step independent of the choice of classifier. Embedded methods selects the subsets as part of the training process. An exhaustive search is generally necessary to guaranty optimal selection, but is usually considered infeasible. In this article, one wrapper (Greedy Forward Selection, GFS) will be used. In addition, knowledge of the phenomenon will be used to suggest a few filters. These filters will be used as a preprocessing step to GFS and other methods explained later.

#### 7.3.2.1 Greedy Forward Selection:

Greedy Forward Selection (GFS) is a semi-exhaustive search for the variable subset that gives the highest classification accuracy [9]. The method is optimal if local minimum equal global minimum. The classification accuracy is first calculated for all variables individually as in SVC. The variable with the highest accuracy is selected. In each iteration thereafter, the variable that, used together with previously selected variables, most increases the classification accuracy is selected. The method is greedy, as it never looks back and reevaluates variables already selected. In each iteration one more variable is added. For p variables, this method requires p calculations in the first round, p-1 in the second round, etc. The total number of calculation when selecting $q$ variables is: $\frac{p+(p-q+1)}{2} \cdot q$. For $p = 3006$ and e.g $q = 10$ variables, this requires 30015 iterations. Notice that GFS is not able to distinguish between variables after zero probability of error has been reached, because adding any variable will not improve the score.

#### 7.3.2.2 Preprocessing filter - Bins:

One observation or trace, $\vec{x}$ consists of $p$ variables. This can be thought of as $p$ non-overlapping intervals (bins) with one variable in each bin. Reducing the number of variables can be achieved by reducing the number of bins to $q < p$, while increasing the size at the same time. Each bin will then contain more than one variable. E.g. with $p = 3006$, using 167 bins will result in 18 variables in each bin, equivalent to approximately 320 kHz bandwidth. Each bin can then be represented by the integrated power of all variables within the bin, thus reducing the number of variables to $q$. This method has been evaluate as a preprocessing step to the GFS method.

#### 7.3.2.3 Harmonics Preprocessing Filter:

Assume that the waveform generated by the power consumption of a microprocessor goes through a Linear Time-Invariant (LTI) system before appearing as electromagnetic emanation outside the microprocessor device. A LTI system may affect the amplitude and the phase of the waveform, while the frequency contents will remain unchanged [13].

Due to the aperiodic nature of the signals found in a microprocessor, it is well known that the expected frequency spectrum will be continuous [13]. However, observations show that the majority of the energy is concentrated in the harmonics of the fundamental frequency of these signals. Since a clock signal is used to synchronize the activity of a microprocessor device, the fundamental frequency of the clock will also be the fundamental frequency of these signals as well. Most of the power will therefore be found at the harmonic frequencies of the microprocessors clock signal. Therefore, under the LTI assumption, we expect to find the majority of the emitted power in the harmonics of the clock frequency as well. Another dominant frequency component is the instruction execution cycle. Therefore, it is expected to find power concentration around the harmonics of this cycle as well.

A simple preprocessing filter could therefore be to select only the samples at the harmonics of either the clock frequency or the instruction execution cycle. In this experiment, the clock frequency is $4$ MHz. One instruction is executed in 4 clock cycles, therefore, harmonics of the instruction execution cycle are found every $1$ MHz. Both $4$ MHz and $1$ MHz harmonics have been evaluated as a preprocessing step to the GFS method.

#### 7.3.2.4 Hybrid:

The bins and the harmonics methods both have limitations. The bin method risks getting the harmonics on the boarder of two bins, splitting relevant power into two different bins. The harmonics method ignores the possibility of relevant power emission to be found in a window around the harmonics, rather than exactly at one sample. Both these problems can be addressed by selecting non-overlapping bins centered at the harmonics. This method is called the hybrid method. In this paper, a bin size of 7 variables is used. The power of the harmonics and $+/-3$ variables (approximate 120 kHz bandwidth) are integrated to form the new variables. This has been done for both the $4$ MHz and the $1$ MHz harmonics. A reduction from $3006$ variables to $14$ and $59$ variables respectively, is achieved by the hybrid method. The hybrid filter can be implemented as a comb-filter. The performance of both hybrids were evaluated as a preprocessing step to the GFS method.

### 7.3.3 Feature Construction Methods

Finally, if reducing the dimensionality of the data is not necessary for storing or processing purposes, different mappings of the raw data can be used. Dimension reduction techniques, also called feature construction [9], can be divided into two major types: linear and non-linear. Examples of such techniques include: clustering, principal component analysis, linear discriminant analysis, fourier transforms and wavelets. In this paper the two classical linear transformations, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) have been tried. A description of PCA and LDA can be found in numerous textbooks, such as [10] and [5].

## 7.4 Comparison and Discussion

Recall the three objectives of feature selection as stated by [9]: (i) Improve the prediction performance of a classifier (ii) Faster/more cost effective classifiers (iii) Better understanding of the phenomenon. In this paper, 6 variable ranking methods and 6 variations (with and without preprocessing filters) of Greedy Forward Selection subset method have been used to select 10 out of 3006 variables. In addition, two construction techniques (PCA and LDA) have been tested. Classification of 5 classes where achieved using Bayes quadratic classifier with features selected by the various methods. The classification error (from confusion matrix) using features selected by the ranking methods are shown in figure 7.3 and the subset methods are shown in figure 7.4. The best ranking and subset methods and the construction methods are compared in figure 7.5. The probability of error is plotted as a function of the number of features selected. The frequency corresponding to each feature is found in table 7.3. This would be the carrier frequencies used by the covert channel. The execution time, as a function of the number of features selected, is shown in figure 7.6 for ranking based methods, and in figure 7.7 for subset based methods. All ranking based methods depend on the number of variables, $p$, but not on how many variables are selected $q$. The execution time of the subset methods depends on both the number of variables $p$ and the number of variables selected $q$. Execution times were obtained using Matlab on a $2,33$ GHz, 2 GB RAM, windows PC.

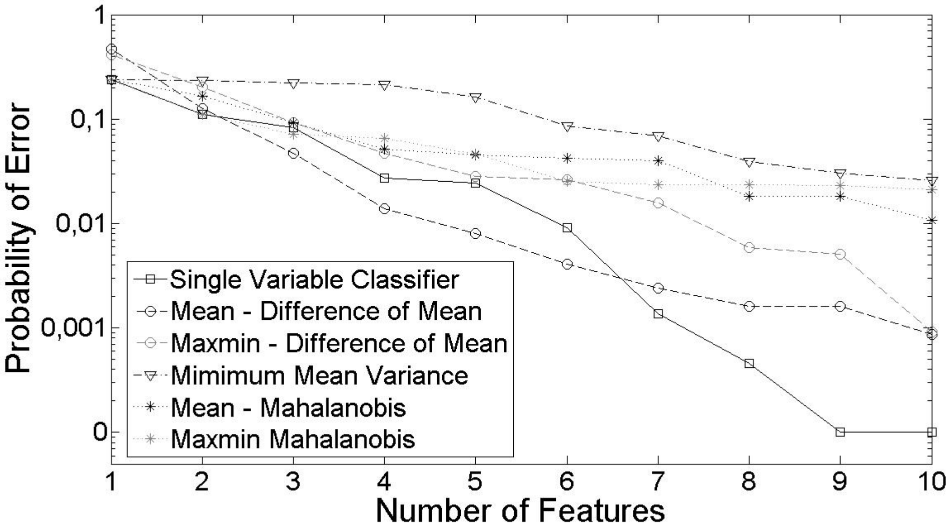

Figure 7.3: Error performance of ranking based feature selection methods

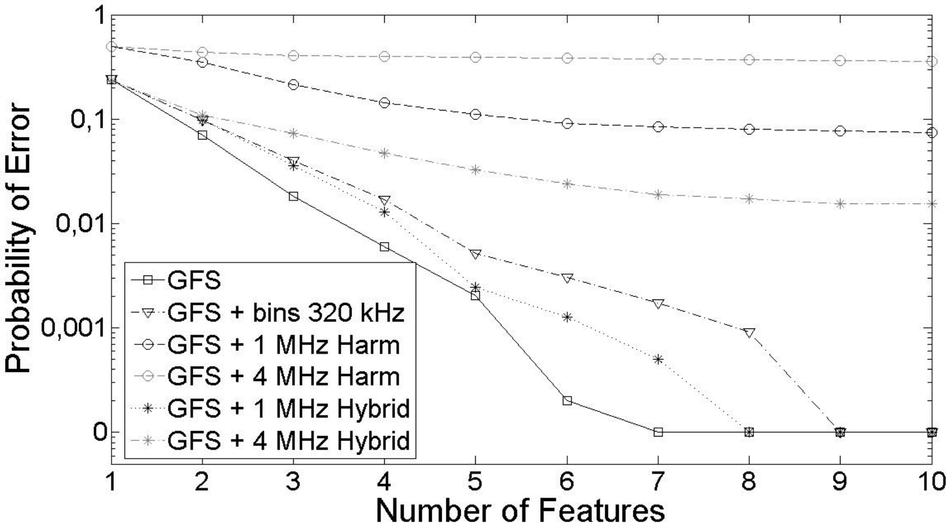

Figure 7.4: Error performance of subset based feature selection methods

### 7.4.1 Faster/more cost effective classifiers

As seen in figure 7.3, of the ranking based methods, Single Variables Classifier (SVC) gives the best performance, except when using 3-6 variables where it is outperformed slightly by Mean Difference of Means (DOM). Using 9 variables selected by SVC achieves zero probability of error ($P_e = 0$). SVC is, however, more than 1000 times slower than the other ranking methods including DOM, which is the fastest ranking method used (figure 7.6 and 7.7). Looking at the probability of error (figure 7.5), ranking methods are generally outperformed by subset selection methods, since the latter also looks for correlations between variables.

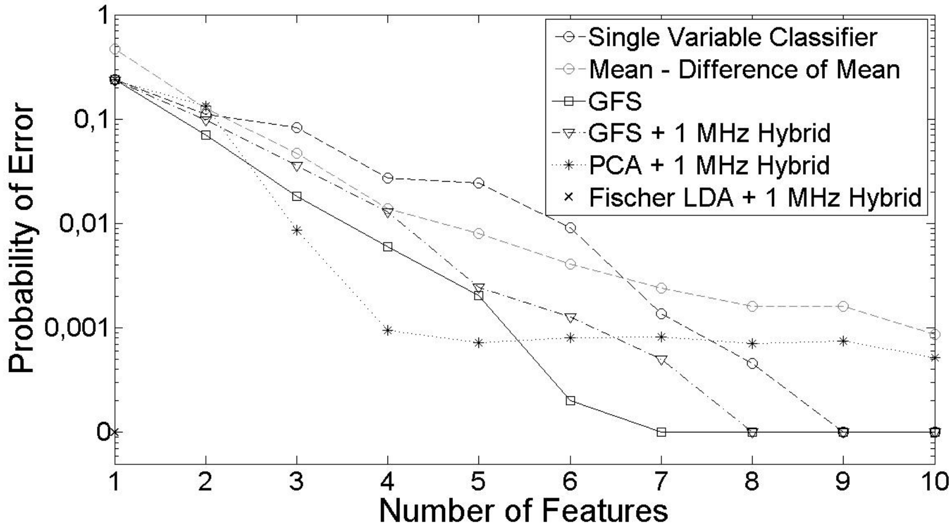

Figure 7.5: Comparing the error performance of ranking based, subset based and construction based feature selection methods

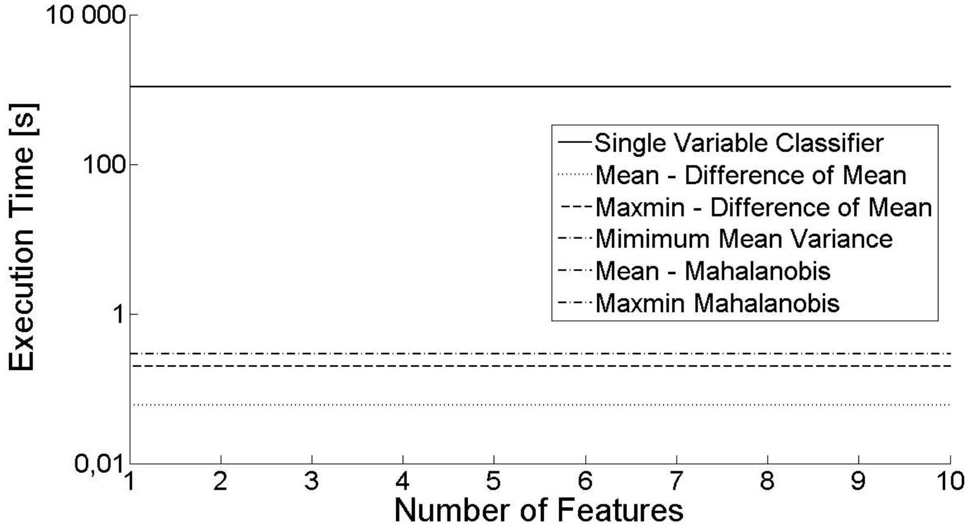

Figure 7.6: Execution time of ranking based feature selection methods

The GFS subset method gives the lowest probability of error of any ranking and subset method, regardless of the number of variables, as it should under the assumption that local minimum is global minimum. $P_e = 0$ can be achieved using seven variables (q=7) (figure 7.4). This comes at an increase in execution time from 18 minutes (SVC) to almost 5 hours (GFS), the highest of any method. The execution time with GFS can be greatly improved by using a preprocessing filter. The best filter tested is the 1 MHz hybrid filter (i.e. 1 MHz comb filter, BW 120 kHz). GFS on the output of this filter achieves $P_e = 0$ using 8 variables, but cuts the execution time from 5 hours down to about 4 minutes.

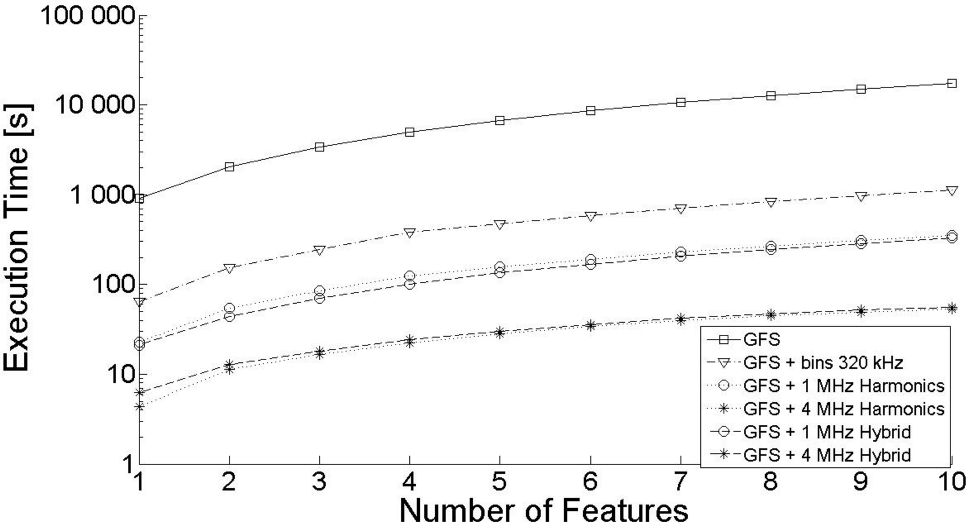

Figure 7.7: Execution time of subset based feature selection methods

| Method | Feature | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Single Variable Classifier | 28,0 | 36,0 | 32,0 | 23,0 | 24,0 | 22,7 | 20,3 | 23,3 | 25,3 | 22,3 |
| Mean - Difference of Means | 23,7 | 22,7 | 20,3 | 25,3 | 23,3 | 22,3 | 21,3 | 26,7 | 24,3 | 21,7 |
| Maxmin - Difference of Means | 20,3 | 22,7 | 23,3 | 25,3 | 21,3 | 29,3 | 28,7 | 24,7 | 20,7 | 23,7 |
| Minimum - Mean Variance | 28,0 | 52,0 | 44,0 | 60,0 | 24,0 | 32,0 | 20,0 | 36,0 | 40,0 | 16,0 |
| Mean Mahalanobis | 28,0 | 24,0 | 32,0 | 36,0 | 20,0 | 52,0 | 44,0 | 23,0 | 16,0 | 21,0 |
| Maxmin Mahalanobis | 28,0 | 36,0 | 24,0 | 16,0 | 21,0 | 23,0 | 52,0 | 44,0 | 60,0 | 19,0 |
| Greedy Forward Selection (GFS) | 28,0 | 25,3 | 23,0 | 24,3 | 36,0 | 20,7 | 19,3 | NA | NA | NA |
| GFS preprocessed - bins (320 kHz) | 24,8 | 22,2 | 19,7 | 20,3 | 28,6 | 18,7 | 26,7 | 20,6 | 23,2 | 2,7 |
| GFS preprocessed - 1MHz harmonics | 20,0 | 22,0 | 21,0 | 19,0 | 30,0 | 32,0 | 23,0 | 10,0 | 18,0 | 31,0 |
| GFS preprocessed - 4MHz harmonics | 20,0 | 32,0 | 12,0 | 24,0 | 44,0 | 52,0 | 8,0 | 16,0 | 48,0 | 36,0 |
| GFS preprocessed - 1 MHz Hybrid | 28,0 | 25,0 | 23,0 | 22,0 | 32,0 | 24,0 | 40,0 | 37,0 | 8,0 | 2,0 |
| GFS preprocessed - 4 MHz Hybrid | 28,0 | 36,0 | 24,0 | 32,0 | 40,0 | 20,0 | 16,0 | 56,0 | 8,0 | 44,0 |

Table 7.3: s

elected features correspond to.]The table shows what frequency [MHz] selected features corresponds too.

### 7.4.2 Better understanding of the phenomenon

The bin filter, with 320 kHz bandwidth, also performs well, but the 4 MHz filters and 1 MHz harmonics filter all have poor performances. This performance difference between the different filters has indicated a few thing about the underlying phenomenon. Harmonics of the clock frequency and the instruction execution cycle are often selected. The harmonics of the clock frequency alone is not enough for accurate classification. A single variable at the harmonics (i.e. smallest possible bins size, determined by the measurement setup) is too small, a certain bandwidth around the harmonics is necessary. This may be explained by measurement error due to clock jitter or that relevant information is spread over larger frequency ranges (i.e not associate with single narrow band carriers). In either case a bandwidth or bin size that includes relevant information, is essential.

Counter-measures exists that aim to eliminate information carried by the clock harmonics. Therefore, it is worth mentioning that the DOM method does not select a single harmonics but still gives fair results, with a classification error about $P_e = 1 \cdot 10^{-3}$ with 10 variables. DOM is also one of the quickest methods tested.

### 7.4.3 Improve the prediction performance of the classifier

Finally, the two construction techniques were tested. Results from using PCA and LDA show that the classification accuracy can be significantly improved by using linear combinations of variables. Fischer LDA, used with the 1 MHz hybrid filter, achieves $P_e = 0$. PCA has excellent performance for $q < 5$, but does not improve further for $q > 5$. The experiment used in this article is not designed well enough to draw any conclusion about the best way to improve the prediction performance. With the available data, it is possible to achieve ”perfect” classification. This can be achieved using 9 variables selected by the SVC method, 7 variables using GFS method or by using Fischer's LDA. The perfect classification accuracy could be due to limited data set and an improved experiment should therefore be carried out.

## 7.5 Conclusion and Future Work

In this paper we have introduced feature selection methods and compared a selection of them by their performance in a wireless covert channel application.

Experimental results show that, in a wireless covert channel application, it is possible to classify several microprocessor activities (i.e instruction executed), with a very low probability of error, given the energy emitted on only a few selected carrier frequencies. Implementation of a M-ary, multi carrier covert channel with a reasonable error rate is therefore feasible.

During the design phase of a classifier, 5 hours of processing time to select which features to use, is often not a problem. In that case, the Greedy Forward Selection Method would be the best choice. If time is of the essence, ranking based methods such as Single Variable Classifier or even Mean Distance of Means, give fair results. Using domain knowledge to build a good preprocessing filter, such as the suggested 1 MHz comb filter, significantly reduces the execution time with a small reduction in performance, compensated by using a few more variables. The error performance achieved with construction techniques, such as PCA and LDA, is encouraging for future work. The results also suggest that harmonics of the clock and instruction cycle are important for classification, but that classification can be done without harmonics as well.

Finally, we believe that the results in this article serve as a good platform for future, in-depth investigations of the phenomenon. Further work should include an experiment looking at real time date (time domain data) collected from several microprocessors with a larger number of microprocessor activities over a larger frequency range.

## 7.6 Acknowledgement

Thanks to Slobodan Petrovic, Katrin Franke and Patrick Bours for an introduction to Pattern Recognition. Thanks to Line Eikvil for discussions on feature selection and to to Anne Solberg for discussions on feature selection and the Greedy Forward Selection method. Thanks to Nils Karlstad Svendsen and Are Strandli for discussion concerning maths and statistics. Thanks to John Oommen for help with the theory of Bayesian classifiers. Thanks to Erik Hjelmaas for help with the theory of Principal Component Analysis and Linear Discriminant Analysis.

## 7.7 Bibliography

[1] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., AND ROHATGI, P. The em side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 29–45. Available

from: http://dx.doi.org/10.1007/3-540-36400-5_4. 1, 15, 17, 19, 23, 37, 50, 71, 85, 88, 113, 127

[2] AGRAWAL, D., BAKTIR, S., KARAKOYUNLU, D., ROHATGI, P., AND SUNAR, B. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy* (may 2007), pp. 296 –310. 31, 46, 61, 71, 85, 99, 113, 127

[3] ARCHAMBEAU, C., PEETERS, E., STANDAERT, F. X., AND QUISQUATER. Template attacks in principal subspaces. In *Cryptographic Hardware and Embedded Systems - CHES* (2006), vol. 4249 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1–14. 1, 30, 39, 72, 91

[4] CHARI, S., RAO, J., AND ROHATGI, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: http://dx.doi.org/10.1007/3-540-36400-5_3. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[5] DUDA, R., HART, P., AND STORK, D. *Pattern Classification*. John Wiley and Sons, Inc, 2001. 8, 26, 28, 29, 30, 63, 64, 65, 75, 78, 90, 91, 94, 124, 137, 141

[6] DYRKOLBOTN, G. O. Analysis of the wireless covert channel attack: Carrier frequency selection). In *Norwegian Computer Science Conference - NIK* (2007), Tapir akademisk forlag, pp. 153–163. 38, 39, 45, 46, 71, 73

[7] DYRKOLBOTN, G. O., AND SNEKKENES, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: http://dx.doi.org/10.1007/11935308_18. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[8] FODOR, I. K. A survey of dimension reduction techniques. Tech. Rep. UCRL-ID-148494, Lawrence Livermore National Laboratory, Technical Information Departments Digital Library, 2002. Available from: http://www.llnl.gov/tid/Library.html. 30, 72

[9] GUYON, I., AND ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (March 2003), 1157–1182. Available from: http://portal.acm.org/citation.cfm?id=944919.944968. 39, 72, 75, 76, 77, 78, 93

[10] JOHNSON, R. A., AND WICHERN, D. W. *Applied Multivariate Statistical Analysis*. Prentice-Hall, 1992. 78

[11] KOCHER, P. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology* (1996), vol. 1109 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 104–113. Available from: http://dx.doi.org/10.1007/3-540-68697-5_9. 1, 26, 50, 71, 88

[12] KOCHER, P., JAFFE, J., AND JUN, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: http://dx.doi.org/10.1007/3-540-48405-1_25. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[13] PROAKIS, J. G., AND MANOLAKIS, D. G. *Digital Signal Processing*, 2 ed. Macmillan Publishing Company, 1992. 22, 24, 77

Chapter 8

# *Modified Template Attack: Detecting Address Bus Signals of Equal Hamming Weight*[1]

## Abstract

Side-channel attacks usually target implementations of cryptographic algorithms on smart cards or FPGA's, but can also be used to recognize other microprocessor activities, such as program, instruction or argument executed. One particular article by Quisquater and Samyde [18] demonstrates how to automatically reverse engineer software code, based on power and electromagnetic signatures. In their article, it is stated that data/addresses of equal Hamming Weight (HW), such as 55 and AA cannot be distinguished without using localization principles (i.e positioning of tiny probes to identify the location of the emanation). Recent methods, such as the Template Attack [7] has overcome this limitation. This is achieved by building precise noise models using identical devices, prior to the attack. Key bytes or intermediate values of cryptographic algorithms, i.e data transfers , can be distinguished, using a Bayesian classifier, even for byte values of equal HW [7].

In this article we present experimental results, validating that the Template Attack can distinguish very similar micro processor activities. Using a modified Template Attack on electromagnetic emanation from smart cards, we are able to distinguish parallel address bus activities of equal HW. We also present an alternative method to select the application specific features necessary (i.e points of interest).

## 8.1 Introduction

When a microprocessor executes its program, electromagnetic signals are generated as a consequence of the power consumption associated with transistor transition. The intercepted electromagnetic signal can be used to reveal the contents of program and/or data memory of the microprocessor. The relationship between electromagnetic emanation (or power) and microprocessor activity is used in side-channel attacks to break cryptographic algorithms [14, 3, 10, 15]. The side-channel information has also been used to reveal hidden hardware faults (trojans) on integrated circuits [4], to control the emanation through subversive software in the Wireless Covert Channel Attack [9] and to to reverse engineer the code executed by microprocessors [18].

The model used in these attacks are often based on the Hamming Weight (HW) or the Hamming Distance (HD) of the code or data handled [15]. HW is simply the number of bits set to one, and HD is the number of bits that are different. For two binary values $X$ and $Y$, both with length $l$, hamming distance is given by $\mathrm{HD}(X,Y) = \sum_{i=1}^{l} X_i \oplus Y_i$. Consider a parallel bus (e.g the address, data or control bus of a microprocessor), with n wires $(w_1, \cdots, w_n)$ transmitting data simultaneously. There will be emanation associate with each rising and falling edge on each bus wire, as seen in figure 8.1. If the "square wave" has equal duration to the bit interval, it is called non-return to zero (NRZ) and the emanation is related to the hamming distance of the data. If the duration of the square

[1] G.O. Dyrkolbotn and E. Snekkenes. In *Proceeding of Annual Norwegian Information Security Conference (NISK)*, Tapir, pages 43-56, 2009.

wave is smaller than the bit interval, it is called return to zero (RZ) and the emanation is related to the Hamming Weight (HW) of the data. Pre-charge bus is one type of RZ.

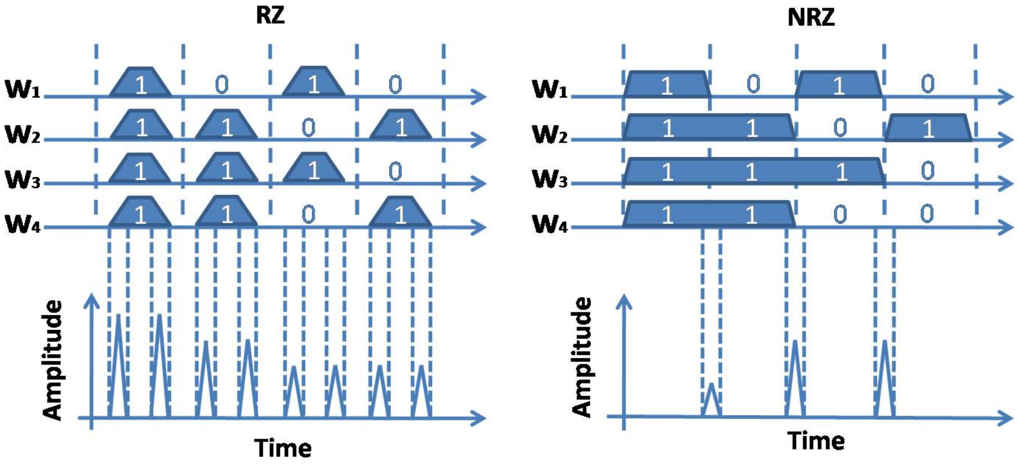

Figure 8.1: Expected emanation from a parallel bus. The emanation from RZ signal types (left) is proportional to the HW of the data. The emanation from NRZ signal types (right) is proportional to the HD of the data.

This figure illustrates the simplest of all models. All transition lines are synchronized and the waveforms on all bus lines are identical. It is easy to see that, given this model, the emanation is proportional to the number of transitions. This makes it straight forward to identify data with different HW (RZ case) or HD (NRZ case). It is also clear that data of equal HW (RZ case) or equal HD (NRZ case) give the same emanation. This is in accordance with the statements in [18] that, ”Two consecutive addresses with the same HW generate two identical power peaks”. The data/address, such as b10101010 and b01010101, should therefore be impossible to distinguish, without using localization principles with extremely small probes.

Results claimed by the Template Attack contradicts this. S. Chari et.al [7] claim that they are able to distinguish signals of equal HW. This means that there are phenomenon, not covered by the simple HW-model, that influence the electromagnetic emanation and make it possible to distinguish very similar microprocessor activities, such as data of equal HW (RZ) or HD (NRZ). It is beyond the scope of this article to study such phenomenon. Using a more general models not constrained to synchronous transition lines and identical waveforms is work in progress.

It looks like the results in [7] are obtained using the data bus or even loading from EEPROM. We are interested in applying a Template Attack to see if we can obtain similar results on activities expected to have a smaller signature, e.g the address bus. We have, therefore, conducted an experiment to validate that the Template Attack can distinguish very similar microprocessor activities. The experiment looks at electromagnetic emanation from address bus activity (as opposed to data bus activity usually targeted) on a microprocessor in a smart card. The results, presented in this article, support that data of equal HW (i.e address bus values) can be classified. The probability of error using a single observation is, however, large. In certain scenarios this can be improved by considering more than one electromagnetic trace. This is future work. This paper is organized as follows: Section 8.2 presents necessary theories concerning the workings of microprocessor, side-channel attack and the Template Attack in particular. Section 8.3 presents the experimental setup and data collection. Section 8.4 presents a method for analyzing the data. Finally a conclusion is drawn and future work outlined in section 8.5.

## 8.2 Microprocessors

A microprocessor typically includes memory (program memory, data memory), CPU , I/O and busses. A microprocessor has a functional activity which is to transform a set of input traces to a set of output traces. The functional activity is determined by the program, a set of instructions stored in program memory, and the contents of data memory. The CPU is responsible for fetching, decoding and executing the instructions, one by one. This is usually controlled by a ”square wave” clock. E.g. the PIC 16F84A microprocessor executes instructions during 4 clock cycles as shown in figure 8.2 (a few exceptions exists). Pipelining is used to fetch the next instruction while the current instruction is executed. In clock cycle 1 (Q1) the current instruction is decoded at the same time as the program counter is incremented. In Q2, data is read if applicable. In Q3, data is processed according to the current instruction. Finally, in Q4, any results are stored at the same time as the next instruction is moved into an instruction register.

It is well known [15] that the majority of the power consumed during the execution sequence depends on the number of gates that change state. This is related to the current state, which bit values are processed and which bit values are moved on various busses. The power consumed by each operation is therefore a function of the instruction (opcode and operand), the data, the address in memory and the prior state (upstate) of the micro processor. This relationship has long been exploited in side-channel attacks.

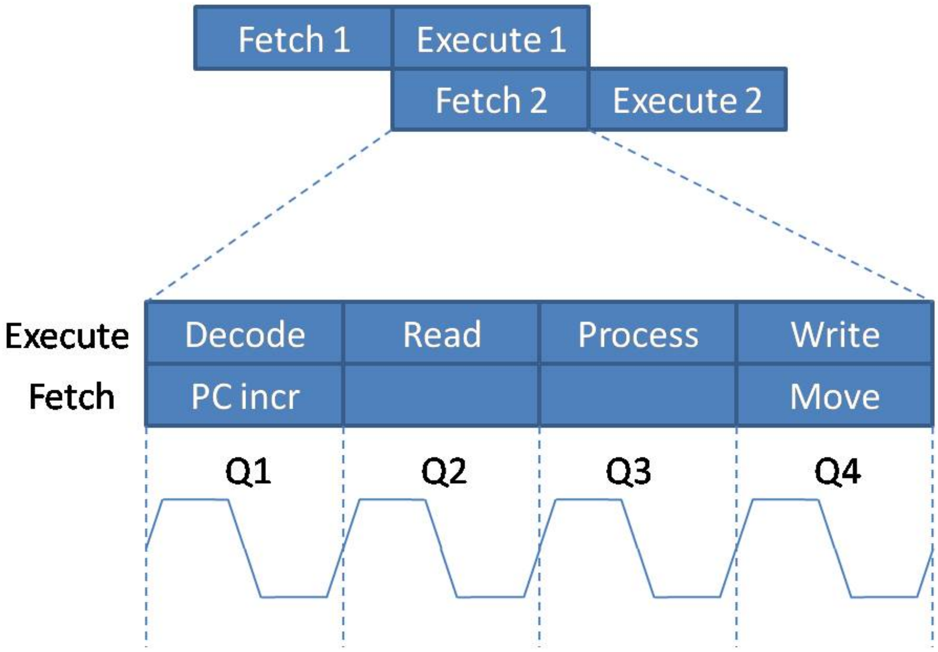

Figure 8.2: Execution cycles and pipelining of microprocessor PIC 16F84A.

### 8.2.1 Side-Channel Attacks

It is possible to use the second law of thermodynamics to show that energy must escape from devices in one way or another (e.g. heat) [11]. The laws of physics explain that it is impossible for any operating device not to leak energy. The goal of side-channel attacks is to look for dependencies between this unavoidable energy leakage and the device's secret parameters.

Exploiting this leakage is not new. Military and government organizations have supposedly used them for a long time, with public interest beginning much later. In 1996 Kocher [13] published his work on exploiting differences in execution time (Timing Attacks). This work was soon followed up and in 1999 Kocher et al. [14] introduced some powerful attacks through measurement of a device's power consumption. Simple Power Analysis (SPA) and Differential Power Analysis (DPA) received some attention from, among others, the banking industry, and countermeasures were publicly announced. In 2000, Quisquater and Samyde [16, 17] applied the analysis technique from SPA and DPA to electromagnetic side-channels, thus introducing electromagnetic analysis (EMA).

In recent years several papers have been published in an ongoing effort to systematically investigate electromagnetic side-channel attacks [3, 7, 2, 1, 5]. The experiments have been extended to some distance from the target, implying that physical access to the target may not be necessary. It has been shown that EMA is at least as powerful as power analysis, and that EMA could circumvent power analysis countermeasures [3, 19].

At USENIX 2002 [18], Quisquater and Samyde described an automatic method to classify instructions, carried out by a simple CISC processor. The power and electromagnetic signature of instructions were captured and then used to train a neural network (Kohonen's self organizing maps). The neural network could automatically recognize, and thus reverse engineer, executed code based on stored electromagnetic and power signatures in 93 % of the cases. The signatures were shown to be a function of the address in memory, data handled and address where the result is stored. A limitation of the approach was that data/address of the same HW were impossible to distinguish. E.g. in [18] it is shown that the same instruction executed from address 55 and AA will give identical signatures. The Template Attack by S.Chari et.al [7] overcomes this limitation by using Bayesian classification methods. This is described next.

### 8.2.2 Template Attack - Classification Using Bayesian Methods

The Template Attack is nothing but Bayesian Classification, where the noise is assumed to follow a Gaussian distribution. Therefore a thorough description of Bayesian classification is given. As with all classifiers, feature selection is important. This is explained later during the analysis of the captured data.

Consider a finite set of classes, $\{\omega_1, \cdots, \omega_c\}$, e.g. execution of $c$ different instructions on a microprocessor. Measurements of any class, results in observations of $d$ variables held in a d-dimensional vector $\vec{x}$. $P(\omega_i)$ is the prior probability of class $\omega_i$ being observed. Let $p(\vec{x}|\omega_i)$ be the a priori class conditional density of $\vec{x}$ being observed given that class $\omega_i$ was measured. Faced with an observation $\vec{x}$ of unknown class, classification is interested in the probability of this observation belonging to the different classes, $\omega_i$. This is expressed by the posteriori probability $P(\omega_i|\vec{x})$ which can be calculated using Bayes rule (equation 8.1):

$$P(\omega_i|\vec{x}) = \frac{p(\vec{x}|\omega_i)P(\omega_i)}{p(\vec{x})} \tag{8.1}$$

where $p(\vec{x})$ the total probability is given by $p(\vec{x}) = \sum_{i=1}^{c} p(\vec{x}|\omega_i)P(\omega_i)$.

Given features $\vec{x}$ of unknown class, the question is how to decide what class to assign $\vec{x}$ to. Generally a decision must be made based on the features $\vec{x}$ and a loss function, $\lambda_{ij}$. Let $\lambda_{ij}$ be the loss if the decision is $\omega_i$ when the true class is $\omega_j$. In the following, a zero-one loss function will be used, where a correct decision gives no loss, and a wrong decision gives a unity loss, equation 8.2.

$$\lambda_{ij} = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \quad i, j = 1, \cdots, c \tag{8.2}$$

Bayes decision rule can be stated as: decide in favor of the class that minimize the probability of error. For the zero-one loss function this will be the class with the maximum a posteriori probability $P(\omega_i|\vec{x})$.

$$Choose \quad \omega_i \; if \; P(\omega_i|\vec{x}) \geq P(\omega_j|\vec{x}) \quad for \; all \; j \in 1, \cdots, c \tag{8.3}$$

A classifier is often represented by a discriminating function, $g_i(\vec{x}), i \in 1, \cdots, c$. $g_i(\vec{x})$ is such that we choose $\omega_i$ if

$$g_i(\vec{x}) \geq g_j(\vec{x}) \quad for \; all \; j \in 1, \cdots, c \tag{8.4}$$

The discriminating functions divide the feature space into decision regions, separated by decision boundaries. It is often useful to realize that any monotonically increasing function $f$ can be applied to $g_i$ without affecting the classification result. For Bayes decision rule (equation 8.3) this means that any of the following discriminating function yield the same result:

$$\begin{aligned} g_i(\vec{x}) &= P(\omega_i|\vec{x}) && (8.5) \\ g_i(\vec{x}) &= p(\vec{x}|\omega_i)P(\omega_i) && (8.6) \\ g_i(\vec{x}) &= ln \, p(\vec{x}|\omega_i) + ln \; P(\omega_i) && (8.7) \end{aligned}$$

As seen in equation 8.7 the discriminating function $g_i$ for Bayes classifier is determined by the prior probability $P(\omega_i)$ and the prior conditional density function $p(\vec{x}|\omega_i)$. $P(\omega_i)$ is usually known or easily calculated, however $p(\vec{x}|\omega_i)$ must be estimated based on available measurements of known class (training the classifier). If a multivariate Gaussian distribution can be assumed, as the Template Attack [7] does, then, $p(\vec{x}|\omega_i) \sim N(M_i, \Sigma_i)$:

$$p(\vec{x}|\omega_i) = \frac{1}{\sqrt{2\pi}^d |\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\vec{x}-M_i)^T \Sigma_i^{-1} (\vec{x}-M_i)} \tag{8.8}$$

where $M_i$ is the mean vector, $\Sigma_i$ the covariance matrix and T denotes transposition. Using the discriminating function from 8.7 and the Gaussian multivariate assumption 8.8 we get:

$$g_i(\vec{x}) = -\frac{1}{2}(\vec{x} - M_i)^T \Sigma_i^{-1} (\vec{x} - M_i) - \frac{d}{2} ln 2\pi - \frac{1}{2} ln|\Sigma_i| + lnP(\omega_i) \tag{8.9}$$

Expressed as a quadratic function for the case of arbitrary covariance matrixes $\Sigma_i$ we get:

$$\begin{aligned} g_i(\vec{x}) &= \vec{x}^T A_i \vec{x} + \vec{x}^T B_i + C_i \quad where && (8.10) \\ A_i &= -\frac{1}{2}\Sigma_i^{-1} \\ B_i &= \Sigma_i^{-1} M_i \\ C_i &= -\frac{1}{2} M_i^T \Sigma_i^{-1} M_i - \frac{1}{2} ln|\Sigma_i| + lnP(\omega_i) \end{aligned}$$

where $x^T$ and $M^T$ is the transposed of $x$ and $M$ respectively. The coefficients of the discriminating function, $A_i, B_i$ and $C_i$ for the quadratic classifier can be pre-computed using a set of training data with known class. Training data is obtained using an identical device prior to the attack. In the Template Attack [7] maximum likelihood estimates (mle) for $M_i$ and $\Sigma_i$ are used, given by:

$$\hat{M}_{mle} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{8.11}$$

$$\hat{\Sigma}_{mle} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{M})(x_i - \hat{M})^T \tag{8.12}$$

The tuples ($\hat{M}_{mle}$,$\hat{\Sigma}_{mle}$) are the templates for each activity of interest. Upon receiving an observation of unknown activity, classification can be done according to equation 8.4.

Once the coefficient of the discriminating function (equation 8.10) has been estimated, the real attack can be lauched, but often the questions is: How good is the classifier? Usually a limited set of data is available for both training and testing. By reserving a portion of the data for testing (leave out k), the accuracy can be found as the ratio of correct classification to the total number of classifications performed (based on a confusion matrix). It is important not to use the same data for training and testing. The accuracy will vary, depending on how the data was split. It is therefore common to randomly split the data into training and testing sets several times, and use the average accuracy as the performance for the classifier. Classifying more than two classes can be done in two ways, Pairwise manner or Bayes method [8]. Both produces the same result, since classification is based on maximizing the posteriori probability. Pairwise classification compares two classes and compares the third class with the result of the first comparison etc. Bayes method calculates the quadratic function for all classes and chooses the maximum result.

## 8.3 The Experiment

The objective of the experiment is to validate that the Template Attack can detect small changes on a parallel bus, such as signals of equal HW. In order to do so, a large amount of data is necessary for training and testing of the classifier.

A smart card (i.e PIC 16F84A microprocessor, 4 MHz clock) executing a test code was place in a screened enclosure. The emanation was captured using an broadband E near-field probe positioned as close to the microprocessor as possible, without decapsulation. Expecting small changes in rise/fall time to manifest itself at high frequencies, a digital oscilloscope was used at 10 Gs/s sampling rate.

The main challenge in this experiment was to design the test code to isolate activity on a parallel bus, with limited interference from other activities. Looking at the pipelining in figure 8.2, two options are evident, a read operation exiting the data bus in clock cycle Q2 or a move operation exciting the address bus in Q4, as explained in 2.1. Since the main objective of the experiment is to investigate if we can detect and classify very similar microprocessor activities, it makes sense to choose activity expected to give the smallest signature, such as the address bus, rather than the data bus.

Designing the test code presents a couple of challenges. First, how do we reduce emanation from other components? The address bus is used during the move operation in Q4, simultaneously with the write operation. The write operation can be eliminated by using the NOP instruction, which has no activity during this operation. The move operation moves the next instruction into the instruction register from a given memory location, activating both the address bus and the program bus. If the instruction is the same each time (e.g. only NOP's) there is no change on the program bus. The emanation in the move operation is therefore only associated with the memory location (i.e value on the address bus) of the instruction. Secondly, how do we create a test pattern for the address bus? Deciding the value of the address bus is much less flexible than on the data bus. On the data bus a simple move instruction can decide the value directly, but on the address bus the value is a function of where in memory the next instruction is found. The test code designed for this experiment executes 128 NOP's in a loop. The values on the 13 bit address bus (disregarding the 5 MSB, as they never change) will increment from b0000 1010 to b1000 1001, giving sufficient combinations to evaluate our hypothesis. A "trigger" toggles the I/O pin on the smart card. This is used to synchronize the oscilloscope and the microprocessor, enabling alignment of the executed code with the measured data. Figure 8.3 shows three consecutive

NOP's, each 1 $\mu$s, separated by vertical lines. Emanation from 8 edges are clearly visible in each NOP, 2 for each of the 4 clock cycles.

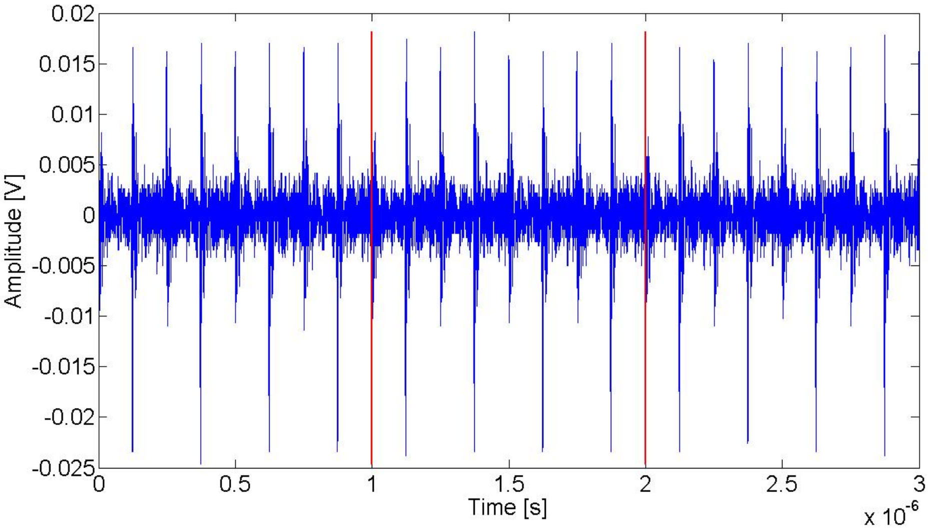

Figure 8.3: Three consecutive NOP's, each 1 $\mu$s. The 8 edges of 4 clock cycles in each NOP are clearly visible

A total of 1000 traces (observations), each recording one execution of the loop, were collected. The next section presents a method for analyzing the data. During this, when the word *instructions* is used, it refers to the address where the instruction is found (i.e value on the address bus).

## 8.4 Analysis

In order to perform the Template Attack on the recorded data, it is necessary to select which variables (i.e data points) to use. Each observation of one executed NOP contains $10000$ variables, such that the trace in figure 8.3 consists of $30000$ variables. The relevant variables for classification are sometimes referred to as points of interest [7] [20] and other times features when pattern classification terms [8] are used. The features will be a subset and/or a transformation of the available variables in each trace. The selected features are then used to train (calculate the the templates) the classifier and evaluate its expected performance in a real attack.

### 8.4.1 Feature Selection

It is unlikely that all variables (i.e. data points) contain relevant information, and a large number of variables requires a lot of computational resources (storage and execution time). It is therefore of interest to reduce the number of variables used by a classifier. In the original attack [7] selection was based on calculating the difference of mean traces. Later other suggestions have been made, such as ranking by sum of mean differences [20] or Principal Component Analysis (PCA) [6]. There has also been attempts using a fourier transform prior to selection appropriate variables [20]. These methods are usually "best practice" as the optimal algorithm is highly application dependent. Our suggestion tries to eliminate some of the choices by basing the selection on calculated scores.

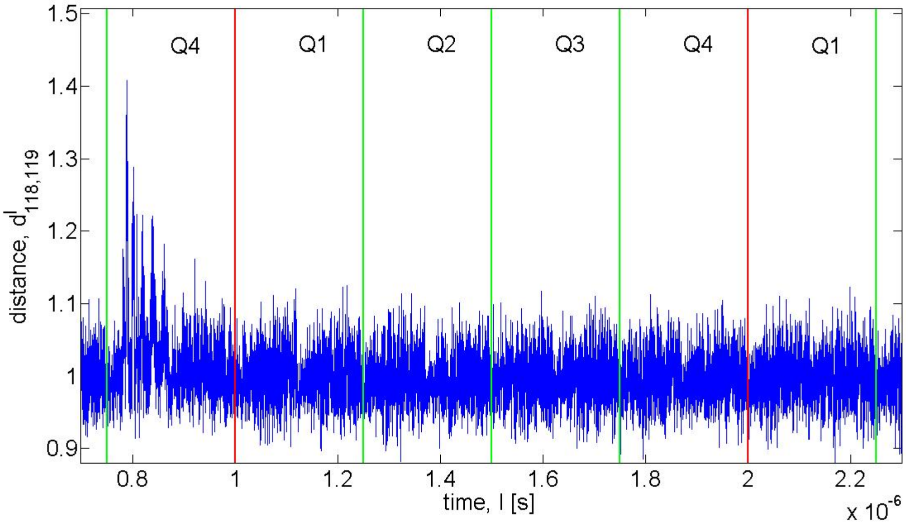

Figure 8.4: The distance, $d^l_{118,119}$, $l = 0.7-2.3\cdot 10^{-6}$ sec., between NOP number 118 (HW=7) and NOP number 119 (HW=1)

#### 8.4.1.1 Select a Time Domain Window

The objective is to identify a time domain window that contains relevant emanation, suitable to classify instructions (i.e values on the address bus) and remove emanation that contribute negative to the classification. A distance measure is used on ″easy″ instructions (i.e one with low HW and one with high HW, expected to have strong emanation), to evaluate the different window selections. Windows that look promising (i.e high score) in the ″easy″ case are used for all other cases. Available data within a window $l$ can be expressed as: $X^l_{ij}$,where $i = \{1, \cdots, 128\}$ is the instruction (i.e NOP at different memory locations) and $j = \{1, \cdots, 1000\}$ is the observation number. The objective of a score function $S$ is to find samples in the time domain window where the distance between observations of the same instruction is short and the distance between observation of different instructions is long. There are many score functions that have this property. The following simple score function between instruction $a$ and $b$ is used:

$$S^l_{a,b,i,j,k} = \frac{|X^l_{ai} - X^l_{bj}|}{|X^l_{ai} - X^l_{ak}|} \tag{8.13}$$

This score function is used to calculate the distance between instruction $a$ and $b$ expressed by:

$$d^l_{a,b} = \frac{1}{n} \sum_{i,j,k=V} S^l_{a,b,i,j,k} \tag{8.14}$$

where $V$ is a list of $n$ random triplets (i.e permutations) drawn from the available observations, $i$, $j$ and $k = \{1, \cdots, 1000\}$ and $i \neq k$. We used $n = 1000$ in this experiment. In our case the emanation is expected to be found in clock cycle Q4 of the fetch cycle. The execution of each NOP is defined to start at the peak of the rising edge of clock cycle Q1 and continue for 1 $\mu$s (i.e 10000 samples). The distance, $d^l_{118,119}$, between NOP number 118 (address $b$01111111, HW=7) and 119 ($b$10000000 ,HW=1) is shown in figure 8.4. With HD=8 between the two addresses this is considered one of the ″easy″ cases.

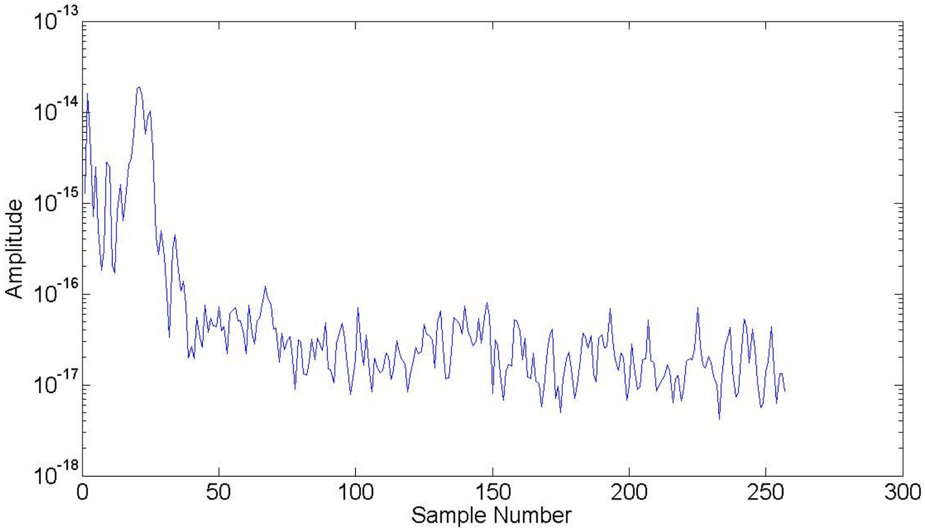

Figure 8.5: Frequency domain representation of Q4 (i.e. first half), using Welch spectral estimation method.

It is clear from figure 8.4 that the highest distance is obtained in the first half of Q4 (i.e rising edge of the clock) during the fetch operation of the instruction. This is the clock cycle where the instruction is transferred to the instruction register. This corresponds to what we expect and these differences are therefore likely to be caused by the differences in data on the address bus. The window identified (first half of Q4) will be used for further analysis.

#### 8.4.1.2 Calculate a Frequency Domain Representation

While the selection of a time domain window is done purely to save computational and storage resources, it has been shown [20] that a transformation into the frequency domain can affect the classification accuracy. In this work a frequency domain representation is obtained using Welch non-parametric estimation method. Welch is the average of overlapping periodograms. The time domain window previously chosen is 1024 samples wide. Welch was used with a 512 sample window with 75 % overlap. The frequency representation therefore consists of 257 samples (i.e DC to $f_s/2$), where $f_s = 10Gs/s$, as seen in figure 8.5. A subset of the variables in the frequency representation is used as features necessary to train and test a classifier. The features are selected using Greedy Forward Selection (GFS) explained next.

#### 8.4.1.3 Greedy Forward Selection

Greedy Forward Selection (GFS) is a semi-exhaustive search for the variable subset that gives the highest classification accuracy [12], where variables in this case are the samples in the frequency domain representation (figure 8.5). GFS is optimal if local minimum equal global minimum. The classification accuracy is first calculated for all variables individually and the variable with the highest accuracy is selected. In each iteration thereafter, the variable that, together with previously selected variables, most increases the classification accuracy is added. The method is greedy as it never looks back and reevaluates variables already selected. In each iteration one more variable is added. For p variables, this method requires p calculations in the first round, p-1 in the second round, etc. The total number of calculation when selecting $q$ variables is: $\frac{p+(p-q+1)}{2} \cdot q$.

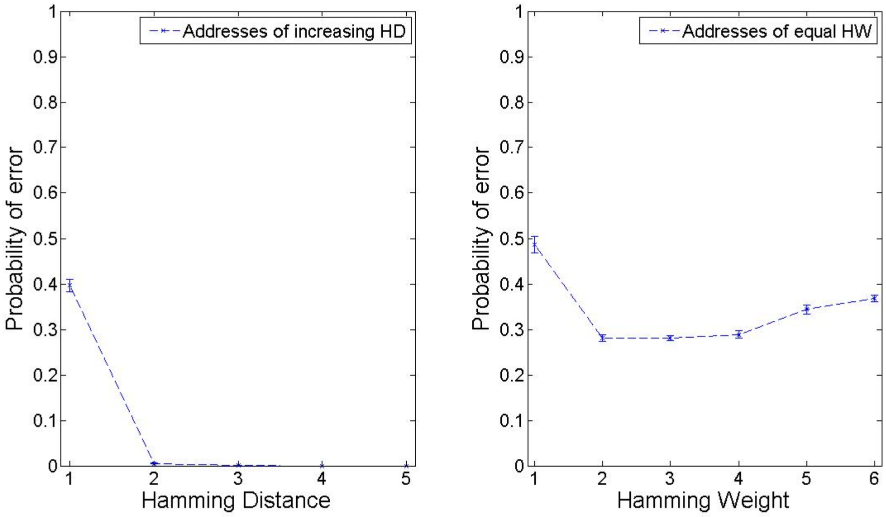

Figure 8.6: Probability of error when classifying address bus values of increasing HD (left) and equal HW (right).

### 8.4.2 Train and Evaluate the Classifier

Given 1000 observations of each address bus value, 200 observation will be used to train Bayes quadratic classifier [8] and the remaining 800 will be used for testing the classifier. The probability of error, $P_e$ is calculated from the confusion matrix [8]. Since the classification accuracy depends on how the observations are split, the average of 100 random permutations of 200 training observations and 800 test observations is used.

A number of different address bus values were compared to evaluate the difficulty in distinguishing them. This includes both addresses of equal HW and addresses of increasing HD. The results, surprisingly, show that the probability of error can be made as low as desired, given a large enough number of features included (over 100). This is probably a result of ”over fitting” the classifier [8], but some work remains to investigate this. The conservative result using only 5 features is presented here (see figure 8.6). One standard deviation is plotted as a confidence interval.

The results confirmed that difficulty in distinguishing activity is linked to the HD of the addresses in question. In the left figure it is clear that for $HD > 1$ it is easy to classify the addresses, with $P_e = 0$. One exception is $HD = 1$ with $P_e = 0,40$. The relationship between HD and difficulty in distinguishing activity becomes even clearer when we calculate the distance $d^l_{a,b}$ (equation 8.14), based on the frequency representation. Figure 8.7 clearly shown that the distance, $d^l_{a,b}$ is proportional to the HD between the two addresses $a$ and $b$.

The right of figure 8.6 shows that classifying addresses of equal HW have a probability of error varying roughly between $0.25 < P_e < 0.5$. The importance of this result is that there is a probabilistic, however small, favor in selecting the correct address.

This confirms that addresses of equal HW can be distinguished as $P_e < 0.5$ and therefore invalidates the statement of Quisquater and Samyde [18] and validates the result of the Template Attack [7]. It may not be accurate to classify the correct address using only one observation, however, under certain circumstances, this can be compensated for by using more than one observation. A classification algorithm using majority voting for this purpose is work in progress.

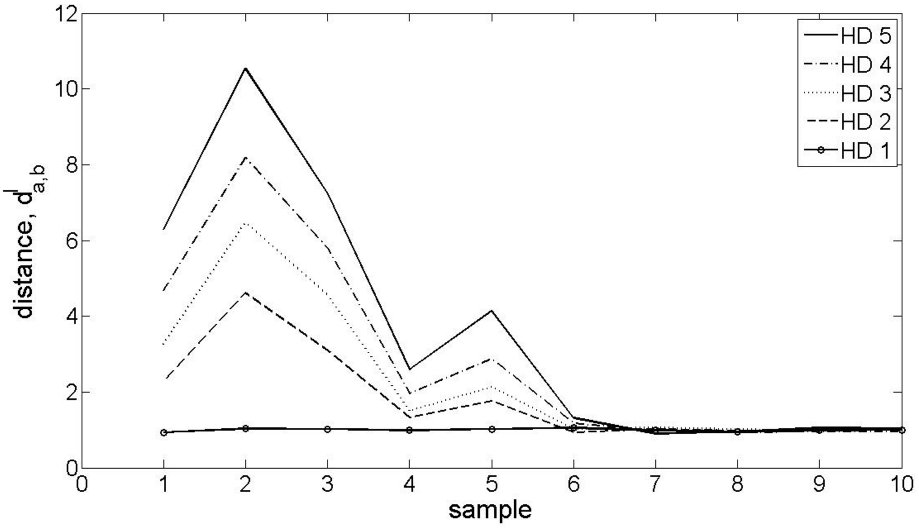

Figure 8.7: Distance, $d^l_{a,b}$ based on frequency domain representation is clearly proportional to the HD.

### 8.4.3 Discussions

The template attack [7], as confirmed in this paper, has the potential to identify any microprocessor activity, even very similar bus values (i.e equal HW). Even though these attacks currently are not practical outside a lab environment, the potential is concerning considering the increasing demand and dependability on microprocessors in secure applications (e.g. smart cards in bank/store terminals). As an example, look at how the use of smart cards is encouraged to prevent skimming of credit cards. However, solving the skimming problem by using microprocessors, opens up for other more sophisticated attacks, such as template attacks. The electromagnetic emanation from the microprocessors in smart cards is detectable at some distance (1-2 meters, possibly more, in our experience). Wireless skimming is therefore a possibility, as physical access is no longer necessary.

The implication of this is that the vicinity of a smart card terminal, not only the terminal itself, should be tamper resistant. It should not be possible to place antennas and receivers, necessary for wireless skimming, in the vicinity of the terminal. We therefore suggest that security measures, such as tamper resistance, shielding and restrictive installation-, maintenance-, operating-procedures, are considered. These measures should be considered during the entire life cycle of a system (e.g. terminal and smart cards) [9], even for smaller facilities in local stores. We have recently seen examples that the store owner facilitates tampering of terminals to allow skimming. The proposed countermeasures should therefore be strong enough to discourage such behavior.

The classification errors show that with a single observation it is difficult to identify the correct activity (sometimes close to guessing). It is therefore essential that security measures strive to refuse the attacker multiple observations.

In our work and the original template attack [7] the noise is assumed to follow a gaussian distribution. The noise consists of channel noise, measurement noise and noise due to transistor transitions not related to the phenomena of interest. In a real world application, without the ability to influence the code executed, the noise caused by unrelated transition may be difficult to predict and may not necessarily follow a gaussian distribution as assumed.

Selecting the relevant features and how many features to use is still a major challenge.

The implication of poor choices has both a complexity and an accuracy implications. Finally, further studies should be done to evaluate the reliability of our method and results. This could be done by repeating our experiment using many identical microprocessors and a set of microprocessors of different architecture. Several variations of the the test program should also be used to investigate the influence of unrelated microprocessor activity.

## 8.5 Conclusion and Future work

Quisquater and Samyde stated in [18] that data/addresses of equal Hamming Weight (HW) cannot be distinguished, without using minute physical probes pointed at individual circuit elements. We have confirmed that using the Template Attack [7] this statement is not correct. Using the electromagnetic side-channel from a smart card, we are able to classify parallel address bus activities of equal HW. Using only a single observation, this can be done with probability of error, in certain cases as low as $P_e = 0.3$. One of the main challenges with the Template Attack is how to choose which features (interesting points) to use. We have suggested selecting feature in the frequency domain, using Greedy Forward Selection method. Future work include improving the classification accuracy by considering more than one observation of each activity. We are already working on a more general model that is able to explain why signals of equal HW can be distinguished.

## 8.6 Bibliography

[1] AGRAWAL, D., ARCHAMBEAULT, B., CHARI, S., RAO, J., AND ROHATGI, P. ”advances in side-channel cryptanalysis, electromagnetic analysis and template attacks”. *CryptoBytes 6*, 1 (Spring 2003), 20–32. 50, 88

[2] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., , AND ROHATGI, P. ”the em side-channel(s):attacks and assessment methodologies”. In *CHES’03* (2003), Lecture Notes in Computer Science, Springer-Verlag. 50, 88

[3] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., AND ROHATGI, P. The em side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 29–45. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_4`. 1, 15, 17, 19, 23, 37, 50, 71, 85, 88, 113, 127

[4] AGRAWAL, D., BAKTIR, S., KARAKOYUNLU, D., ROHATGI, P., AND SUNAR, B. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy* (may 2007), pp. 296 –310. 31, 46, 61, 71, 85, 99, 113, 127

[5] AGRAWAL, D., RAO, J., AND ROHATGI, P. Multi-channel attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), C. Walter, e. Ko, and C. Paar, Eds., vol. 2779 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 2–16. Available from: `http://dx.doi.org/10.1007/978-3-540-45238-6_2`. 1, 17, 37, 50, 88

[6] ARCHAMBEAU, C., PEETERS, E., STANDAERT, F. X., AND QUISQUATER. Template attacks in principal subspaces. In *Cryptographic Hardware and Embedded Systems - CHES* (2006), vol. 4249 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1–14. 1, 30, 39, 72, 91

[7] CHARI, S., RAO, J., AND ROHATGI, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_3`. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[8] Duda, R., Hart, P., and Stork, D. *Pattern Classification*. John Wiley and Sons, Inc, 2001. 8, 26, 28, 29, 30, 63, 64, 65, 75, 78, 90, 91, 94, 124, 137, 141

[9] Dyrkolbotn, G. O., and Snekkenes, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: `http://dx.doi.org/10.1007/11935308_18`. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[10] Gandolfi, K., Mourtel, C., and Olivier, F. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES* (2001), vol. 2162 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 251–261. Available from: `http://dx.doi.org/10.1007/3-540-44709-1_21`. 1, 7, 15, 17, 21, 26, 37, 85, 113, 127

[11] Giancoli, D. C. *Physics for Scientists and Engineers*. Prentice Hall, 1989. 14, 50, 88

[12] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (March 2003), 1157–1182. Available from: `http://portal.acm.org/citation.cfm?id=944919.944968`. 39, 72, 75, 76, 77, 78, 93

[13] Kocher, P. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology* (1996), vol. 1109 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 104–113. Available from: `http://dx.doi.org/10.1007/3-540-68697-5_9`. 1, 26, 50, 71, 88

[14] Kocher, P., Jaffe, J., and Jun, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: `http://dx.doi.org/10.1007/3-540-48405-1_25`. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[15] Mangard, S., Oswald, E., and Popp, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[16] Quisquater, J.-J., and Samyde, D. A new tool for non-intrusive analysis of smart cards based on electromagnetic emissions:the sema and dema methods. *Eurocrypt rump session* (2000). 1, 37, 50, 88

[17] Quisquater, J.-J., and Samyde, D. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security* (2001), vol. 2140 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 200–210. Available from: `http://dx.doi.org/10.1007/3-540-45418-7_17`. 1, 7, 15, 16, 17, 21, 26, 37, 50, 88

[18] Quisquater, J.-J., and Samyde, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from: `http://portal.acm.org/citation.cfm?id=1250988.1250994`. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[19] Rao, J., Rohatgi, and Pankaj. Empowering side-channel attacks. Tech. rep., IBM T.J. Watson Research Center, 2001. 1, 7, 16, 17, 37, 50, 88

[20] Rechberger, C., and Oswald, E. Practical template attacks. In *Information Security Applications* (2005), vol. 3325 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 440–456. Available from: `http://dx.doi.org/10.1007/978-3-540-31815-6_35`. 1, 30, 91, 93

Chapter 9

# *Security Implications of Crosstalk in Switching CMOS Gates*[1]

## Abstract

The energy dissipation associated with switching in CMOS logic gates can be used to classify the microprocessors activity. It is relatively easy to classify activity by the number of transitions, i.e. Hamming Distance (HD). In this paper we consider layout dependent phenomena, such as capacitive crosstalk to derive a more precise power model. We show that for an 8 bit data bus, crosstalk may improve detection performance from 2.5 bits (HD based detector) to theoretical 5.7 bits and simulated 5.0 bits (crosstalk based detector) of information per sample. Thus we have shown that a layout specific phenomenon (capacitance) must be considered when analyzing security implications of power and electromagnetic side-channels. A small case study has also been carried out that support our simulations/theoretical results.

## 9.1 Introduction

When a microprocessor executes its program, power consumption (or resulting electromagnetic emanation) can be used to reveal the contents of program and/or data memory of the microprocessor. The correlation between power consumption and microprocessor activity is well known and has found many uses [1, 2, 5, 8, 11].

In a side-channel attack a common power model, used to simulate the power consumption, is the Hamming Distance (HD) model, as it is simple and generic [9]. The model assumes the power consumption to be proportional to the number of transitions taking place. If this assumption was correct, signals transmitted on a parallel bus (e.g. intermediate values of the cryptographic algorithm) with the same HD should have equal power consumption and therefore be indistinguishable. This is not always the case, as demonstrated by the template attack [2, 6]. The phenomena behind this may be known in the security community, but has received little attention. One paper by Chen et al. [3] investigates the effect of the coupling capacitance on masking schemes without a detailed examination of the phenomena. In their book "Power Analysis Attacks", Mangard et.al. [9] mention power simulation at analog level as "the most precise way to simulate the power consumption of digital circuits...". Parasitic elements, such as parasitic capacitances between the wires and unwanted capacitances in the transistors are mentioned. However, it is also stated that it is very common to make simplifications by lumping together extrinsic and intrinsic capacitances into a single capacitance to ground. This will in fact make the model incapable of explaining the results we are addressing in this paper.

Parasitic couplings, and the coupling capacitance in particular, is however a great concern within sub-micron VLSI design [4, 10, 12]. Parasitic couplings between interconnects, such as on-chip buses, influence both the power consumption and maximum obtainable speed [4]. Moll et al. [10] did a detailed analysis of the energy dissipation from two metal

[1] G.O. Dyrkolbotn, K. Wold and E. Snekkenes. In *Proceedings of Information Security Conference (ISC)*, Springer, Lecture Notes in Computer Science, pages 269-275, 2010.

lines running close together and show that ″coupling capacitance is very different from the capacitance to ground because it depends on the switching activity... ″. Duan et al. [4] show that for 3 adjacent lines, transition patterns can be divided into 5 crosstalk classes based on the influence of the coupling capacitances. The focus in VLSI design, such as [10, 4], is on power consumption and delays, not on security.

In this paper we look at the security implication of capacitive crosstalk in switching CMOS gates. We put forward the hypothesis that layout dependent phenomena, such as parasitic coupling between wires, can explain why it sometimes is possible to distinguish transition patterns with the same HD. We present a new power model that takes into account capacitive coupling between parallel wires. Theory and PSPICE simulations are used to evaluate how the new power model effects our ability to classify activity in a microprocessor. We use the ability to extract entropy as our classifier performance indicator and show that a detector capable of detecting energy levels due to crosstalk can extract more information than a detector based on HD only.

This paper is organized as follows: Section 9.2 presents the hypothesis of layout dependent phenomena. Section 9.3 presents our model and necessary theory to calculate the energy dissipation. Section 9.4 is an analytic analysis of security implications. Section 9.5 presents simulation results. Finally a conclusion is drawn in section 9.6.

## 9.2 Layout Dependent Phenomena

In a physical implementation of any circuit (e.g. CMOS based microprocessor) a number of phenomena will influence the energy dissipation and the resulting radiated electromagnetic field. These phenomena includes inductance and capacitance of conductors, inductance and capacitance between conductors, wireless transmission characteristics (i.e antenna properties) of conductors and other circuit elements and complex combinations of these phenomena. These phenomena apply to any transistors and wires in a circuit, but we choose to look at a portion of wires running in parallel as we expect them to be relatively good antennas and therefor a good source for side-channel information. We believe, complex combinations of layout dependent phenomena may be the key to identify minute differences in microprocessor activity. In the following we will assume that the coupling capacitance is the dominating factor and show how this can explain why some signals with the same HD can be distinguished. This will show the potential effect of layout dependent phenomena on classifying microprocessor activity. Our work can easily be extended by including other layout dependent phenomena if a more precise result is needed.

## 9.3 Theoretical Considerations

A model of a parallel bus driven by CMOS inverters with only coupling and load capacitances is shown in Fig. 9.1. This is a generalization of the model for two lines used by Moll et al. [10] and includes a model of the CMOS inverter.

It can be shown that the total energy dissipation, $E_T$ from an $n$ wire bus, can be written as:

$$E_T = \sum_{j=1}^{n} V_{DD} \int i_{pj} dt - \sum_{j=1}^{n} \int V_j (i_{pj} - i_{nj}) dt \tag{9.1}$$

For PSPICE simulation the following assumptions are used:

- The load capacitances for data bus lines are identical ($C_j = C_L$ for $j = \{1, 2, \cdots, n\}$)
- Coupling capacitances are only found between adjacent line and are identical ($C_{j,j+1} = C_C$ for $j = \{1, 2, \cdots, n-1\}$)

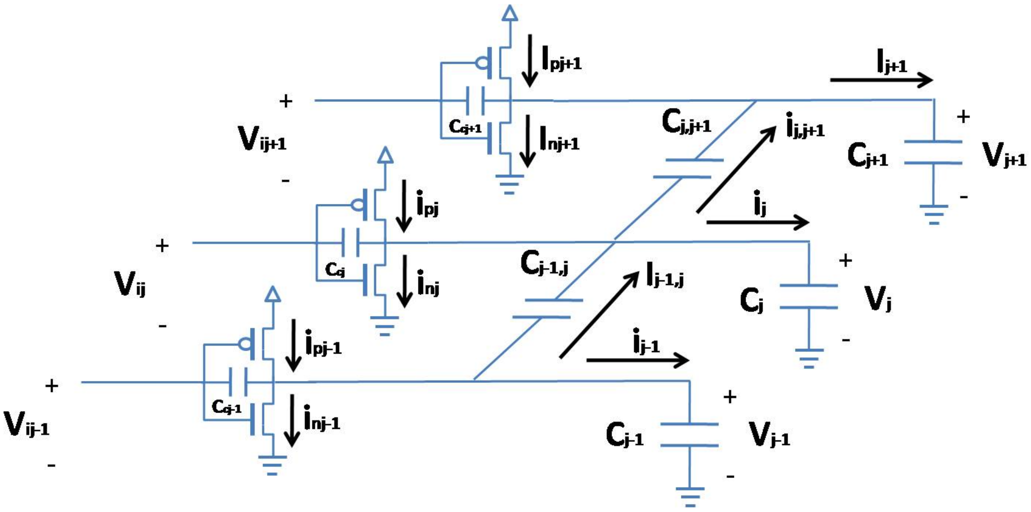

Figure 9.1: Simplified model, assuming load and coupling capacitances to be dominant

In order to compare the simulated energy dissipation ($\hat{E}_T$) with analytic values ($E_T$), a different expression than (9.1) is needed. If the contributions from the load ($C_L$) and coupling capacitance ($C_C$) are dominant to the dissipated energy, then $E_T$ can be expressed in the following power model:

$$E_T = \frac{1}{2} C_L V_{DD}^2 (k + \alpha\lambda) = E_0(k + \alpha\lambda) \tag{9.2}$$

where $E_0 = \frac{1}{2} C_L V_{DD}^2$, $V_{DD}$ is the power supply voltage, $k$ is the number of transitions on the data bus ($k = 0, 1, 2, \ldots$), $\lambda = C_C/C_L$ and $\alpha$ is the crosstalk index indicating the coupling capacitance induced crosstalk, similar to the crosstalk classes in [4]. For a $n$ line bus the crosstalk index $\alpha$ is the sum of the crosstalk influence of each line:

$$\alpha = \sum_{j=1}^{n} \alpha_j \tag{9.3}$$

Let $\delta_j \in \{0, \pm 1\}$ be the normalized voltage change on line $j$, then $\delta_{j,k} = \delta_j - \delta_k$, and

$$\alpha_j = \begin{cases} 0 & no\ transition\ line\ j \\ |\delta_{j,j-1} + \delta_{j,j+1}| & otherwise \end{cases} \tag{9.4}$$

It can be shown that $\alpha_j = \{0, 1, 2\}$ for lines with only one adjacent line (edges), and $\alpha_j = \{0, 1, 2, 3, 4\}$ for lines with two adjacent lines. In the next section we will use (9.2) and (9.3) to analyze which transition patterns that can be distinguished.

## 9.4 Security Implications

How will the new power model (9.2) effect our ability to classify activity in a microprocessor, such as data transfer on a parallel bus?

Let $A$ be the set of possible transitions on an $n$ bit parallel bus. A model of the energy dissipation should ideally be able to distinguish all $|A| = 4^n$ transitions. This may not be possible, either because of simplifications of the model or physical properties such that multiple transitions patterns indeed uses the same amount of energy. A model can only distinguish transition patterns by the distinct energy levels explained by the model. A model that assumes energy dissipation proportional to the number of transition, can therefor only distinguish transition pattern into subsets $A_k$, $k = \{0, \cdots, n\}$ being subsets of $A$ that has $k$ transitions. The number of transition patterns in each subset is given by: $|A_k| = 2^n \binom{n}{k}$. Using the new power model (9.2), each subset $A_k$ can be split into a number of new energy levels, giving a number of smaller subsets $A_k^\alpha$, where $\alpha$ is the crosstalk index of (9.3). For an 8 bit bus, taking into consideration the coupling capacitance increases the number of energy levels from $9$ in the HD model to $93$ in the crosstalk model, e.g. the $14336$ transitions patterns with 5 transitions, previously indistinguishable, can now be split into 18 energy levels.

Finally, we have only shown how to split the subset $A_k$ into smaller subsets $A_k^\alpha$ by considering the effect of the coupling capacitance (i.e $\alpha$). This idea can easily be generalized, such that $A_k$ is split into subsets $A_k^\beta$, where $|A_k| > |A_k^\beta|$, and $\beta$ is the influence of other layout dependent phenomena, such as variations in coupling and load capacitance and inductance.

### 9.4.1 Classification Performance

For the purpose of comparing alternative detectors we will assume uniform random transition, thus for a 8 bit bus we would like the detector to extract 16 bits. We will use the ability to extract entropy as our classifier performance indicator. The entropy (i.e bits of information) extracted by a detector, when there are $r$ energy levels, can be calculated using:

$$H(x) = -\sum_{i=1}^{r} p(x_i) log\ p(x_i) \tag{9.5}$$

In the following we have assumed a 8 bit bus width, thus there are $4^8 = 65536$ possible transitions. Call the detector that can extract 16 bits of information a level detector. If we assume that one only has bus activity when initial and final state are different, and that $0 \rightarrow 1$ and $1 \rightarrow 0$ can be distinguished, an observation will give us the following entropy: $-(1/2log1/2 + 1/4log1/4 + 1/4log1/4) = 3/2$ bits as we cannot distinguish $0 \rightarrow 0$ from $1 \rightarrow 1$. The theoretical optimum for an 8 bit bus with a 'transition detector' would be $8 \cdot 3/2\ bits = 12\ bits$. The entropy extracted by a HD detector is found using (9.5) with $r = 9$ and $p(x_i) = |A_{i-1}|/65536$ giving an entropy of 2.5 bits. The entropy extracted by a crosstalk detector is found using (9.5) with $r = 93$ and $p(x_i) = |A_{i-1}^\alpha|/65536$ giving an entropy of 5.7 bits. The difference between the ideal value of a level detector and the entropy extracted by other detectors, represent the amount of guessing needed when classifying an observation. By considering the coupling capacitance and not only HD, we extract more information out of each observation, therefore reducing the amount of "guessing" needed for classification. In the next section we present simulations validating the effect of the coupling capacitance.

## 9.5 Simulations

The simulations are performed in PSPICE with $C_L = 400fF$, $C_C = 250fF$, $V_{dd} = 3V$ and a rise- and fall-time of $200ps$ of the input voltages (same as [10]). The inverter drivers are equal and balanced. Equation (9.1) is used in PSPICE to find the simulated energy dissipation $\hat{E}_T$. Having first validated our simulations against the results of [4, 10] simulations for all possible subsets $A_k^\alpha$ were carried out. The results for $k = 5$ can be seen i Table 9.1.

The simulated energy levels $\hat{E}_T$ are similar to the analytic values $E_T$ (9.2). The results confirm that energy consumption is proportional to the number of transitions and the crosstalk index, $\alpha$. The crosstalk index depends on switching activity on adjacent lines and position, edge (one adjacent wire) or middle (two adjacent wires). A theoretical crosstalk detector capable of separating all 93 energy levels can extract $5.7$ bits of information. It is

Table 9.1: Analytic ($E_T$) and simulated ($\hat{E}_T$) dissipated energy when considering crosstalk for bus with 8 lines. k=5 is the number of transitions (Hamming Distance) and $\alpha$ is the crosstalk index

| **Transition pattern** | **k** | $\alpha$ | $E_T$ **[pJ]** | $\hat{E}_T$ **[pJ]** |
|---|---|---|---|---|
| 0000 0000 → 0001 1111 | 5 | 1 | 10,1 | 10.4 |
| 0000 0000 → 1000 1111 | 5 | 2 | 11,3 | 11.5 |
| 0000 0000 → 0010 1111 | 5 | 3 | 12,4 | 12.6 |
| 0000 0000 → 1001 0111 | 5 | 4 | 13,5 | 13.8 |
| 0000 0000 → 0101 0111 | 5 | 5 | 14,6 | 14.9 |
| 0000 0000 → 1010 1011 | 5 | 6 | 15,8 | 16.0 |
| 0000 0010 → 0111 0001 | 5 | 7 | 16,9 | 16.8 |
| 0000 0010 → 1011 0001 | 5 | 8 | 18,0 | 17.9 |
| 0000 0010 → 0101 1001 | 5 | 9 | 19,1 | 19.3 |
| 0000 0010 → 1010 1001 | 5 | 10 | 20,2 | 20.4 |
| 0000 0010 → 0110 0101 | 5 | 11 | 21,4 | 21.3 |
| 0000 0010 → 1010 0101 | 5 | 12 | 22,5 | 22.5 |
| 0000 0010 → 0101 0101 | 5 | 13 | 23,6 | 23.5 |
| 0000 1010 → 1000 0101 | 5 | 14 | 24,8 | 24.5 |
| 0000 1010 → 0100 0101 | 5 | 15 | 25,9 | 25.6 |
| 0001 0100 → 0100 1010 | 5 | 16 | 27,0 | 27.0 |
| 0000 1010 → 0001 0101 | 5 | 17 | 28,1 | 27.9 |
| 0001 0100 → 0010 1010 | 5 | 18 | 29,3 | 29.1 |

Table 9.2: Comparing the ability to extract information of different detectors for an 8 wire bus

| type of detector | Entropy (information) |
|---|---|
| | [bits] |
| Level detector | 16,0 |
| Optimum transition detector | 12,0 |
| Crosstalk detector (theoretical) | 5,7 |
| Crosstalk detector (simulated) | 5,0 |
| HD detector | 2,5 |

expected that a practical crosstalk detector will extract less information, due to some subsets having almost equal energy levels. A random loss of 20% of the subsets will still on average have an entropy of 5.0. This still gives an information gain of 2.5 bits compared to the HD detector. The performance of the detectors are summarized in Table 9.2. We also have experimental data, suggesting that the average classification error gets reduced as $\alpha$ distance ($\Delta\alpha = |\alpha_i - \alpha_j|$) increases. This supports our simulation/theoretical results, but details are omitted due to limited space[2].

## 9.6 Conclusion

It is known that one can distinguish bus activity generated from signal transitions having different HD. In this paper we put forward the hypothesis that layout dependent phenomena, such as inductance and capacitance in and between conductors and radiation properties of circuit elements, can explain why it sometimes is possible to distinguish transition patterns with the same HD. Our simulations show that capacitive crosstalk has a significant effect on gate energy dissipation. Our results confirm that the dissipated energy from CMOS switching gates depend not only on the HD, but also on the direction of switching

[2] Details are published in [7], see subsection 11.5.3

activity on nearby data lines. Where as a HD based detector can provide about 2.5 bits of information per sample, a crosstalk based detector will yield about 5.7 bits (theoretical) or 5.0 bits (simulated) of information per sample - in all cases for an 8 bit bus. Thus we have shown that a layout specific phenomenon (capacitance) must be considered when analyzing security implications of electromagnetic side-channels.

## 9.7 Bibliography

[1] Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., and Sunar, B. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy* (may 2007), pp. 296 –310. 31, 46, 61, 71, 85, 99, 113, 127

[2] Chari, S., Rao, J., and Rohatgi, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: http://dx.doi.org/10.1007/3-540-36400-5_3. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[3] Chen, Z., Haider, S., and Schaumont, P. Side-channel leakage in masked circuits caused by higher-order circuit effects. In *Advances in Information Security and Assurance* (2009), vol. 5576 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 327–336. Available from: http://dx.doi.org/10.1007/978-3-642-02617-1_34. 40, 99, 108, 113, 127

[4] Duan, C., Calle, V., and Khatri, S. Efficient on-chip crosstalk avoidance codec design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 17*, 4 (april 2009), 551 –560. 40, 99, 100, 101, 102, 108, 114, 118, 121, 128, 132, 135

[5] Dyrkolbotn, G. O., and Snekkenes, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: http://dx.doi.org/10.1007/11935308_18. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[6] Dyrkolbotn, G. O., and Snekkenes, E. Modified template attack: Detecting address bus signals of equal hamming weight. In *Annual Norwegian Information Security Conference - NISK* (2009), Tapir akademisk forlag, pp. 43–56. 39, 40, 41, 42, 46, 47, 99, 109, 113, 122, 124, 127, 135, 137, 142

[7] Dyrkolbotn, G. O., Wold, K., and Snekkenes, E. Layout dependent phenomena: A new side-channel power model. *Journal of Computers 7*, 4 (April 2012). 20, 40, 41, 42, 47, 103

[8] Kocher, P., Jaffe, J., and Jun, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: http://dx.doi.org/10.1007/3-540-48405-1_25. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[9] Mangard, S., Oswald, E., and Popp, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[10] Moll, F., Roca, M., and Isern, E. Analysis of dissipation energy of switching digital cmos gates with coupled outputs. *Microelectronics Journal 34*, 9 (2003), 833 – 842. Available from: http://www.sciencedirect.com/science/article/pii/S0026269203001332. 40, 99, 100, 102, 108, 114, 116, 117, 118, 121, 128, 130, 131, 132, 134, 135

[11] QUISQUATER, J.-J., AND SAMYDE, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from: `http://portal.acm.org/citation.cfm?id=1250988.1250994`. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[12] SOTIRIADIS, P., AND CHANDRAKASAN, A. Low power bus coding techniques considering inter-wire capacitances. In *Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000* (2000), pp. 507 –510. 40, 99, 108, 114, 118, 128, 132

Chapter 10

# *Non-Invasive Reverse Engineering of the Relative Position of Bus Wires*[1]

## Abstract

Exploitation of microprocessor side-channels can be improved if one has access to details regarding the physical layout of the microprocessor. It has been shown that the capacitance between bus wires influence the signals at the electromagnetic side-channel. This paper provides a non-invasive technique for establishing the relative position of microprocessor bus wires. Thus, used in combination with signal analysis techniques, our results may improve side-channel exploitation capabilities.

## 10.1 Introduction

An increasing number of systems rely upon tamper resistant devices, such as smart cards. It is important to realize that these cards are not tamper proof, but provide speed bumps for the dedicated adversary. Given enough resources, these cards can be reverse engineered using state of the art semi-conductor test equipment [10]. Due to the high cost of such equipment, a number of low cost techniques have been suggested [1]: Differential fault analysis, chip rewriting attacks, memory remanence attacks and protocol failure. Finally there is a large group of attacks based on side-channel information, such as power and electromagnetic emission analysis, that are powerful and surprisingly low cost. Many of these attacks (e.g DPA attacks [6]) rely upon good power models to be successful. Detailed information about the internal structure of a microprocessor allows for better power models and therefore more efficient attacks [7].

In any physical implementation of a circuit (e.g CMOS based microprocessor) many phenomena will influences the energy dissipation and the resulting radiated electromagnetic field. These phenomena includes inductance and capacitance of conductors, inductance and capacitance between conductors, wireless transmission characteristics (i.e antenna properties) of conductors and other circuit elements and complex combinations of these phenomena [5]. In [5] we have shown through simulations and measurement that capacitive crosstalk has a significant effect on the energy dissipation. When capacitive crosstalk is part of a power model, it is possible to extract more information (in terms of entropy) about microprocessor activity than when using simpler power models. The proposed power model in [5] assumes that the relative position of bus wires are known.

In this paper we provides a non-intrusive technique for establishing the relative position of microprocessor bus wires. With this information, the crosstalk power model, combined with signal analysis techniques, may improve the efficiency of side-channel attacks. For future work, it would be interesting to see if layout dependent phenomena [5] has the potential to reverse engineer other physical structures of a micro processor, such as memory, masking schemes and dual-rail logic.

[1] G.O. Dyrkolbotn. In *Proceeding of Annual Norwegian Information Security Conference (NISK)*, Tapir, pages 104-109, 2010.

This paper is organized as follows: Section 10.2 presents necessary theory about crosstalk in switching CMOS gates. Section 10.3 presents our non-invasive black box reverse engineering method. Section 10.4 contains conclusion and future work.

## 10.2 Crosstalk in Switching CMOS gates

Crosstalk can be defined as the coupling of energy between two conductors. Inductive coupling is caused by mutual inductance, $L$, (i.e magnetic field) and capacitive coupling is caused by mutual capacitance, $C$, (i.e electric field) between two wires. Especially the coupling capacitance is a great concern in sub-micron VLSI design [3, 8, 9]. As CMOS technology gets smaller the speed and power consumption is increasingly influenced by parasitic effects, such as capacitive crosstalk [3]. These phenomena may also be known in the security community, but has received little attention. Other than our own article [5], we have found one article by Chen et al. [2] that studies the effect of the coupling capacitance on masking schemes and the book ”Power Analysis Attacks”, Mangard et.al. [7], mentions power simulation at analog level as ”the most precise way to simulate the power consumption of digital circuits...”. However, a very common simplification, lumping together coupling capacitance and capacitance to ground into a single capacitance to ground removes any crosstalk influence. In [5] we present theory and simulations on some security implications of capacitive crosstalk in CMOS driven data busses.

Assume that the load ($C_L$) and coupling capacitance ($C_C$) are dominant to the dissipated energy ($E_T$) from a $n$ bit bus. Then it is shown in [5] that the total dissipated energy can be expressed by:

$$E_T = \frac{1}{2} C_L V_{DD}^2 (k + \alpha\lambda) = E_0(k + \alpha\lambda) \tag{10.1}$$

where $E_0 = \frac{1}{2}C_L V_{DD}^2$, $V_{DD}$ is the power supply voltage, $k$ is the number of transitions on the data bus ($k = 0, 1, 2, \ldots$), $\lambda = C_C/C_L$ and $\alpha$ is the crosstalk index indicating the coupling capacitance induced crosstalk. In the proposed method we will use transition patterns with known crosstalk indexes to establish the relative position of microprocessor bus wires. For further detail and how to calculate the crosstalk index $\alpha$ we refer to [5].

## 10.3 Non-Invasive Black Box Reverse Engineering Method

The goal is to reverse engineer/determine the relative position of physical wires of a bus by analyzing electromagnetic emanation, without any invasive actions. Consider a bus of $n$ wires. Let $\omega_1\omega_2, \cdots, \omega_{n-1}\omega_n$ be the physical layout of the bus and $b_1 b_2, \cdots, b_{n-1}b_n$ be the logic bit values used by the micro processor during execution of a program (e.g data transfer a bus).

The objective of the following method is to find a mapping $b \rightarrow \omega$ between the logic bit value $b_i$ to the physical wire $\omega_k$ that it invokes. The mapping may be straight forward, such that $\omega_1 = b_1, \omega_2 = b_2, \cdots, \omega_n = b_n$, however there are cases where the mapping is different. The physical layout of the bus could be altered due to design (i.e routing) choices. The logic value could also be masked before driving the bus wires, such as CODEC’s, scramblers or encryption. In this paper we will consider the case were the physical layout is different from the logic order of the bits.

Our method exploits the effect of coupling capacitance on the energy dissipation during bit transitions. Notice from (10.1) that the total energy dissipation from the bus, $E_T$, is proportional to $\alpha$ (for the same number of transitions). The smaller $\alpha$ is the less energy is dissipated. If all neighboring wires up to a point $k$ ($\omega 1 \omega 2, \cdots, \omega_k$) have the same transition and all other wires have no transitions, it can be shown [5] that the crosstalk index is always one ($\alpha = 1$). It turns out that, given an equal number of transitions, these transition patterns represent the minimum possible dissipated energy, and all other transitions patterns will

Table 10.1: For a given number of transitions, these transition patterns uses the least amount of energy ($\alpha = 1$), compared to any other transition patterns. U denotes a transition from $0 \rightarrow 1$, 0 no transition

| $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ | $\omega_7$ | $\omega_8$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ | $\omega_7$ | $\omega_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | U |
| U | U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | U | U |
| U | U | U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | U | U | U |
| U | U | U | U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | U | U | U | U |
| U | U | U | U | U | 0 | 0 | 0 | 0 | 0 | 0 | U | U | U | U | U |
| U | U | U | U | U | U | 0 | 0 | 0 | 0 | U | U | U | U | U | U |
| U | U | U | U | U | U | U | 0 | 0 | U | U | U | U | U | U | U |

have $\alpha > 1$ and thus dissipate more energy. Two exceptions exist: No transitions and transitions on all wires, in which $\alpha$ can be zero. For an $n = 8$ bit bus, the minimum energy transition patterns ($\alpha = 1$) are shown in table 10.1 ($U$ is a positive transition ($0 \rightarrow 1$) and 0 is no transition). If the positive transition was substituted for a negative transition ($D$: $1 \rightarrow 0$) the result would be same. Also notice that it is not possible to distinguish between mirror images of the layout, as e.g. transition pattern $00UUUUUU$ and $UUUUUU00$ both have $\alpha = 1$. Therefore $\omega_1\omega_2, \cdots, \omega_{n-1}\omega_n$ and $\omega_n\omega_{n-1}, \cdots, \omega_2\omega_1$ will be indistinguishable.

### 10.3.1 The Method

The proposed method takes advantage of the patterns of table 10.1, such that finding physical neighbors reduces to finding the test pattern that dissipates the least amount of energy. The method requires $n - 1$ steps to determine the physical layout of an $n$ bit bus. The first step finds the two edges $\omega_1$ and $\omega_n$. The next steps successively finds one more neighbor, from one side, until all wires are covered. In the following these steps are explained for an $n = 8$ bit bus, but the method applies to any sized bus $n$. In each step it is necessary to measure the microprocessors energy dissipation as each transition pattern is transferred on the bus. Making sure the microprocessor carries out the correct activity, that the appropriate emanation is captures and how to calculate the dissipated energy is not a topic in this article. Some help on these issues can be found in [4].

#### 10.3.1.1 Step 1: Find the Edge Wires, $\omega_1$ and $\omega_8$:

The transition pattern necessary to detect the edges, $\omega_1$ and $\omega_8$ can be seen in table 10.2. The transition patterns are designed to produce a single transition ($0 \rightarrow 1$) on each wire. The two test patterns that cause a transition in either $\omega_1$ or $\omega_8$ will have $\alpha = 1$ (10.1). The other 6 patterns will cause a transition on wires with two neighbors and therefore have $\alpha = 2$ (10.1). A mapping for the two edge wires, $\omega_1$ and $\omega_8$, is therefore found by detecting the two transition patterns that dissipate the least amount of energy. As an example: assume that transition on $b_2$ (test pattern 7) and $b_5$ (test pattern 4) were found to dissipate the least amount of energy, then the two edges would be mapped by: $\omega_1 = b_5$ and $\omega_8 = b_2$.

#### 10.3.1.2 Step 2: Find wire $\omega_2$

We choose to detect wires from $\omega_1$ and up, but is it also possible to detect wires from $\omega_8$ and down. The transition pattern necessary to detect the next wire, $\omega_2$, given $\omega_1 = b_5$ and $\omega_8 = b_2$ can be seen in table 10.3. The transition patterns are designed to produce two transition. One transition is always kept on the edge wire, $\omega_1 = b_5$ found in step 1. The other transition is on each of the remaining unknown wires. The other edge wire ($\omega_8$) is therefore excluded. Both transitions must be in the same direction, e.g. $0 \rightarrow 1$. According to the result in table 10.1, the transition pattern that creates a transition next to $\omega_1$ will have

Table 10.2: Test patterns necessary to determine edge wires $\omega_1$ and $\omega_8$. The two test patterns that cause a transition in either $\omega_1$ or $\omega_8$ will have $\alpha = 1$ and thus dissipate less energy than the other 6 patterns

| Pattern | $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ | $b_7$ | $b_8$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | U |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | U | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | U | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | U | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | U | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | U | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | U | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | U | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 10.3: Test patterns necessary to determine wire $\omega_2$. The test patterns that cause a transition in $\omega_1$ and $\omega_2$ will have $\alpha = 1$ (see table 10.1) and thus dissipate less energy than the other 5 patterns

| Pattern | $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ | $b_7$ | $b_8$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | U | 0 | 0 | U |
| 2 | 0 | 0 | 0 | 0 | U | 0 | U | 0 |
| 3 | 0 | 0 | 0 | 0 | U | U | 0 | 0 |
| 4 | 0 | 0 | 0 | U | U | 0 | 0 | 0 |
| 5 | 0 | 0 | U | 0 | U | 0 | 0 | 0 |
| 6 | U | 0 | 0 | 0 | U | 0 | 0 | 0 |

Table 10.4: Mapping between physical layout $\omega$ and logic value $b$ for our example. 7 steps necessary to map all 8 wires

| | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ | $\omega_7$ | $\omega_8$ |
|---|---|---|---|---|---|---|---|---|
| start | ? | ? | ? | ? | ? | ? | ? | ? |
| step1 | $b_5$ | ? | ? | ? | ? | ? | ? | $b_2$ |
| step2 | $b_5$ | $b_4$ | ? | ? | ? | ? | ? | $b_2$ |
| step3 | $b_5$ | $b_4$ | $b_3$ | ? | ? | ? | ? | $b_2$ |
| step4 | $b_5$ | $b_4$ | $b_3$ | $b_1$ | ? | ? | ? | $b_2$ |
| step5 | $b_5$ | $b_4$ | $b_3$ | $b_1$ | $b_8$ | ? | ? | $b_2$ |
| step6 | $b_5$ | $b_4$ | $b_3$ | $b_1$ | $b_8$ | $b_7$ | ? | $b_2$ |
| step7 | $b_5$ | $b_4$ | $b_3$ | $b_1$ | $b_8$ | $b_7$ | $b_6$ | $b_2$ |

$\alpha = 1$. All other test patterns will have a larger energy dissipation ($\alpha > 1$). Locating $\omega_2$ is again done by detecting the transition pattern with the lowest energy dissipation. Continuing the example: if test pattern 4 is found to dissipate the least amount of energy, then the next wire is mapped by: $\omega_2 = b_4$.

#### 10.3.1.3 Step j: Find wire $\omega_j$

The transition pattern necessary to detect the next wires are found much the same way until all wires are located. Transition patterns for wire $\omega_j$ must have transitions an all known wires $\omega_1, \cdots, \omega_{j-1}$, but not $\omega_8$. All transitions must be in the same direction. The lowest energy dissipation corresponding to $\alpha = 1$ will again reveal the mapping of wire $\omega_j$.

Table 10.4 shows all steps when continuing our example. E.g step 3, using 3 transitions reveals $\omega_3 = b_3$, step 4, using 4 transitions reveals $\omega_4 = b_1$, etc. Finally after $n - 1 = 7$ steps the mapping is complete and the relative position of the physical bus wires are known.

## 10.4 Conclusion and Future Work

In this article we have presented a non-invasive method to reverse engineer the relative position of microprocessor bus wires. The method takes advantage of the influence of capacitive crosstalk on energy dissipation from bus wires. In each step of the method, a set of transition patterns are design, such that one by one wire is mapped from logic value to physical location. In each step the transition pattern dissipation the least amount of energy reveal the next wire on the bus. This method is non-invasive assuming that the energy dissipation can be found by capturing the electromagnetic emanation from the device. Knowing the relative position of bus wires is a prerequisite for our crosstalk power model of [5], that combined with with signal analysis techniques may improve side-channel exploitation capabilities. We believe that by considering other layout dependent phenomena its should be possible to reveal other physical structures of the microprocessor, such as memory, masking schemes and dual-rail logic. This is future work.

## 10.5 Acknowledgement

First I would like to thank my supervisor Einar Snekkenes for helpful discussion and guidance. I would also like to thank Knut Wold for letting me explain my ideas.

## 10.6 Bibliography

[1] ANDERSON, R., AND KUHN, M. Low cost attacks on tamper resistant devices. In *Security Protocols*, B. Christianson, B. Crispo, M. Lomas, and M. Roe, Eds., vol. 1361 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1998, pp. 125–136. Available from: `http://dx.doi.org/10.1007/BFb0028165`. 107

[2] CHEN, Z., HAIDER, S., AND SCHAUMONT, P. Side-channel leakage in masked circuits caused by higher-order circuit effects. In *Advances in Information Security and Assurance* (2009), vol. 5576 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 327–336. Available from: `http://dx.doi.org/10.1007/978-3-642-02617-1_34`. 40, 99, 108, 113, 127

[3] DUAN, C., CALLE, V., AND KHATRI, S. Efficient on-chip crosstalk avoidance codec design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 17*, 4 (april 2009), 551 –560. 40, 99, 100, 101, 102, 108, 114, 118, 121, 128, 132, 135

[4] DYRKOLBOTN, G. O., AND SNEKKENES, E. Modified template attack: Detecting address bus signals of equal hamming weight. In *Annual Norwegian Information Security Conference - NISK* (2009), Tapir akademisk forlag, pp. 43–56. 39, 40, 41, 42, 46, 47, 99, 109, 113, 122, 124, 127, 135, 137, 142

[5] DYRKOLBOTN, G. O., WOLD, K., AND SNEKKENES, E. Security implications of crosstalk in switching cmos gates. In *Information Security Conference - ISC* (2010), vol. 6531 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 269–275. 40, 41, 42, 47, 107, 108, 111, 114, 118, 125

[6] KOCHER, P., JAFFE, J., AND JUN, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: `http://dx.doi.org/10.1007/3-540-48405-1_25`. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[7] MANGARD, S., OSWALD, E., AND POPP, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[8] MOLL, F., ROCA, M., AND ISERN, E. Analysis of dissipation energy of switching digital cmos gates with coupled outputs. *Microelectronics Journal 34*, 9 (2003), 833 – 842. Available from: `http://www.sciencedirect.com/science/article/pii/S0026269203001332`. 40, 99, 100, 102, 108, 114, 116, 117, 118, 121, 128, 130, 131, 132, 134, 135

[9] SOTIRIADIS, P., AND CHANDRAKASAN, A. Low power bus coding techniques considering inter-wire capacitances. In *Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000* (2000), pp. 507 –510. 40, 99, 108, 114, 118, 128, 132

[10] TORRANCE, R., AND JAMES, D. Reverse engineering in the semiconductor industry. In *Custom Integrated Circuits Conference - CICC* (sept. 2007), pp. 429 –436. 107

Chapter 11

# *Layout Dependent Phenomena: A New Side-channel Power Model*[1]

## Abstract

The energy dissipation associated with switching in CMOS logic gates can be used to classify the microprocessors activity. In VLSI design layout dependent phenomena, such as capacitive crosstalk, becomes a major contributor to the power consumption and delays of on-chip busses, as transistor technology get smaller. These effects may be known to the security community but have received little attention.

In a recent paper we have presented a new power model, taking into consideration capacitive crosstalk. We have shown that capacitive crosstalk has a significant effect on gate energy dissipation. Our results confirm that the dissipated energy from CMOS switching gates depend not only on the hamming distance (HD), but also on the direction of switching activity on nearby data lines. We show that for an 8 bit data bus, crosstalk may improve detection performance from 2.5 bits (HD based detector) to theoretical 5.7 bits and simulated 5.0 bits (crosstalk based detector) of information per sample.

In this paper we elaborate on the theory and simulations of layout dependent phenomena and how they must be considered when analyzing security implications of power and electromagnetic side-channels. We have also added a small case study, i.e the electromagnetic side-channel of a smart card, that support our simulations/theoretical results.

## 11.1 Introduction

When a microprocessor executes its program, power consumption (or resulting electromagnetic emanation) can be used to reveal the contents of program and/or data memory of the microprocessor. The correlation between power consumption and microprocessor activity has found many uses: to recover cryptographic keys [11, 3, 12, 1, 10], to reveal hidden hardware faults (trojans) on integrated circuits [2], to control the emanation through subversive software in the Wireless Covert Channel Attack [7] and to reverse engineer the code executed by microprocessors [14].

In side-channel attacks, a common power model used to simulate the power consumption is the Hamming Distance (HD) model, as it is simple and generic [12]. The model assumes the power consumption to be proportional to the number of transitions taking place. If this assumption was appropriate, signals transmitted on a parallel bus (e.g. intermediate values of the cryptographic algorithm) with the same HD should have equal power consumption and therefore be indistinguishable. This is not always the case, e.g. if Bayes classifier is used, as suggested by the template attack [3]. It has also been demonstrated in [8] that signals with the same number of transitions can be classified using a modified template attack.

The phenomena behind this may be known in the security community, but has received little attention. One paper by Z. Chen, S. Haider and P. Schaumont [4], investigates the effect of the coupling capacitance on masking schemes without a detailed examination of

[1] G.O. Dyrkolbotn, K. Wold and E. Snekkenes. Submitted to *Journal of Computers*

the phenomena. In their book ”Power Analysis Attacks”, S. Mangard, E. Oswald and T. Popp [12] mention power simulation at analog level as ”the most precise way to simulate the power consumption of digital circuits...”. Parasitic elements, such as capacitances between the wires and unwanted capacitances in the transistors are mentioned. However, it is also stated that it is very common to make simplifications by lumping together extrinsic and intrinsic capacitances into a single capacitance to ground. This will, in fact, make the model incapable of explaining the results we are addressing in this paper.

Parasitic couplings, and the coupling capacitance in particular are, however, a great concern within sub-micron VLSI design [5, 13, 15]. CMOS technology is currently being pushed into deep sub-micron range. As the number of transistors increase, the need for on-chip wiring increases as well and must be scaled accordingly. Parasitic couplings between interconnects, such as on-chip buses, must be taken seriously as they influence both the power consumption and maximum obtainable speed [5]. F. Moll, M. Roca and E. Isern [13] did a detailed analysis of the energy dissipation from two metal lines running close together. The lines were driven by CMOS inverters and transitions in one or two wires were studied. The effect of coupling capacitance between the two lines on the power consumption was shown analytically and simulated in HSPICE. The main result was that if two bus lines have transitions in the same or opposite direction at the same time, the total energy is either lower or higher than if the two transitions are treated independently. This is due to the coupling capacitance. C. Duan, V.H.C. Calle and S.P. Khatri [5] focus on crosstalk avoidance codes that aim to reduce the effect of the coupling capacitances by avoiding specific data transition patterns. Their model considers coupling capacitance, $C_C$, between three adjacent lines. They show that 3 bit transition patterns can be divided into 5 crosstalk classes based on the influence of the coupling capacitances, $C_C$. The energy consumption therefore depends on which crosstalk class the transition pattern belong to, as seen in Table 11.1 reprinted from [5].

Table 11.1: Classes of crosstalk from [5]. $C_{eff}$ is the efficient capacitance, $C_L$ the load capacitance and $\lambda = C_C/C_L$

| Class | $C_{eff}$ | Transition pattern |
|---|---|---|
| 0C | $C_L$ | $000 \rightarrow 111$ |
| 1C | $C_L(1+\lambda)$ | $011 \rightarrow 000$ |
| 2C | $C_L(1+2\lambda)$ | $010 \rightarrow 000$ |
| 3C | $C_L(1+3\lambda)$ | $010 \rightarrow 100$ |
| 4C | $C_L(1+4\lambda)$ | $010 \rightarrow 101$ |

The focus within VLSI design, such as [5] and [13] is on power consumption and delays caused by the coupling capacitance. They do not considered security implications, such as the ability to use the variation in energy consumption to classify transition patterns. However, correlations between data and energy consumption are exactly what side-channel attacks, such as DPA and Template attack, rely upon.

In this paper we elaborate on the hypothesis put forward in [9] that layout dependent phenomena, such as capacitive coupling between wires, can explain why it sometimes is possible to distinguish transition patterns with the same HD. We extend the theory and simulations of how the new power model, that takes into account capacitive crosstalk, affect our ability to classify activity in a microprocessor.

We look at the total dissipated energy from a parallel data bus driven by CMOS inverters. Our model is a generalization of [13], with inverters consisting of two MOSFET transistors, a load capacitance $C_L$ connected to each inverter output and a coupling capacitance $C_C$ connected between each bus line. Our model is generalized to $n$ lines and simulations in PSPICE are done with eight bus lines.

The purpose of our simulation is to show that when the dissipated energy depends on

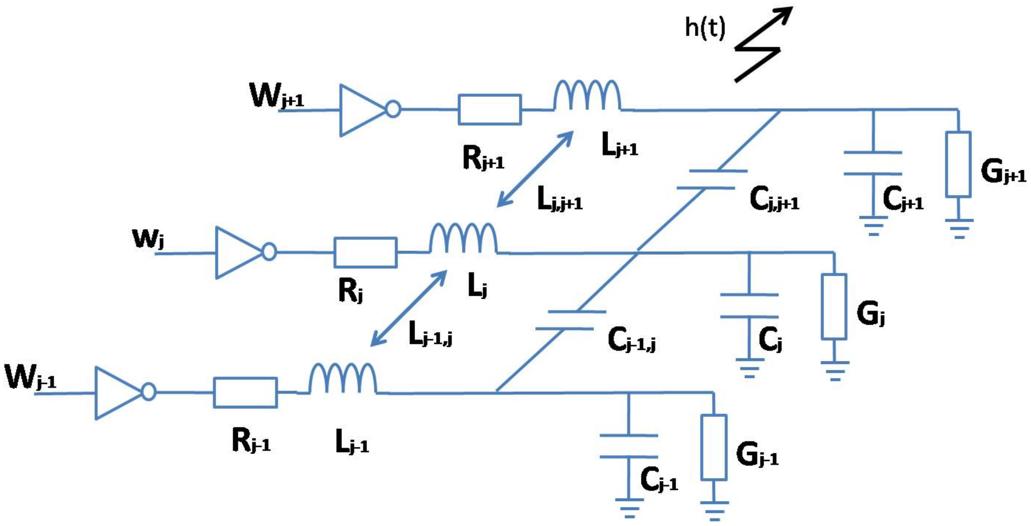

Figure 11.1: Model of layout dependent phenomena

the direction of change of nearby data lines, and not only the number of transitions taking place, the number of possible energy levels dissipating from the bus will increase, thus allowing classification of a larger number of transition patterns. Our hypothesis is that this can be used to explain why some signal with the same HD can be distinguished. Our model can easily take into consideration other layout dependent phenomena, potentially offering an explanation to classification of an even larger set of transition patterns. We will use the ability to extract entropy as our classifier performance indicator and show that a detector capable of detecting energy levels due to crosstalk can extract more information than a detector based on HD only.

Finally, in order to probe the practicality of our theory and simulations, we have included a small case study, in which the objective is to see if analysis of electromagnetic side-channel information also supports the division into crosstalk energy levels.

This paper is organized as follows: Section 11.2 presents the hypothesis of layout dependent phenomena. Section 11.3 presents our model and necessary theory to calculate the energy dissipation. Section 11.4 is an analytic analysis of security implications. Section 11.5 presents simulation results and the case study. Finally, a conclusion is drawn in Section 11.6.

## 11.2 Layout Dependent Phenomena

In a physical implementation of any circuit (e.g. CMOS based microprocessor) a number of phenomena will influence the energy dissipation and the resulting radiated electromagnetic field. These phenomena include inductance and capacitance of conductors, inductance and capacitance between conductors, wireless transmission characteristics (i.e antenna properties) of conductors and other circuit elements and complex combinations of these phenomena. These phenomena apply to any transistors and wires in a circuit, but we choose to look at a portion of wires running parallel, as we expect them to be relatively good antennas and therefore a good source for side-channel information. This is illustrated in the model of a parallel bus, driven by CMOS inverters, seen in Fig. 11.1.

### 11.2.1 Inductance and Capacitance of Conductors

Any conductor, $W_j$, carrying an electric current will have an associated distributed resistance $R_j$, inductance $L_j$, conductance $G_j$ and capacitance $C_j$, expressed as a characteristic impedance, $Z_{0j}$. The characteristic impedance is often modeled as an infinite series of lumped components. The inductance $L_j$ and capacitance $C_j$ will both block high frequency signals and act as a low pass filter. Small variations in the length and width of conductors result in small variations in the inductance. Small variations in the area and distance to ground plane result in small variations in the capacitance. There will therefore be small variations in how signals on different conductors (e.g. bus lines) are filtered.

### 11.2.2 Inductance and Capacitance between Conductors

Crosstalk can be defined as the coupling of energy between two conductors. Inductive coupling is caused by mutual inductance, $L_{j,j+1}$, (i.e magnetic field) and capacitive coupling is caused by mutual capacitance, $C_{j,j+1}$, (i.e electric field) between wire $j$ and $j+1$. These couplings occur along the entire length of the conductor, but are also modeled as lumped components (Fig. 11.1). The interaction of magnetic and electric fields will effectively change the characteristic impedance, $Z_{0j}$, associated with the conductor. This interaction is layout dependent (e.g. distance and length of wires) and will effect both delays and energy dissipation. An important property of crosstalk is its dependency on the activity on the wires. F. Moll, M. Roca and E. Isern [13] state that, "coupling capacitance is very different from the capacitance to ground because it depends on the switching activity... ". If two lines are low and rise at the same time, the mutual capacitance coupling, $C_{j,j+1}$, does not have to be charged. However, if one line remains low and the other rises, $C_{j,j+1}$ must be charged, resulting in increased rise time and power consumption.

### 11.2.3 Wireless Transmission Characteristics

Any circuit element in the microprocessor, conducting electric current, can be considered an antenna. An antenna is a transducer converting electric current into electromagnetic waves, characterized by properties such as: resonant frequency, gain, radiation pattern, impedance, efficiency, bandwidth and polarization. These properties depend on factors such as: amount of current, length/shape and material of the circuit element. In addition, the electromagnetic waves will be influenced by filtering, reflection and interference from surrounding material and circuit elements. The relationship between the current (i.e power consumption) and the electromagnetic wave can be expressed by a transfer function $h(t)$ (Fig. 11.1). Predicting $h(t)$ is not trivial, if possible at all, as most physical systems are not linear by nature. This is left for future work, but it is a fair assumption that relatively long bus lines are good antennas.

### 11.2.4 Complex Combinations of Factors

Finally, complex combinations of layout dependent phenomena may be the key to identify minute differences in microprocessor activity, e.g. the radiation efficiency of bus lines combined with data and layout dependencies of the line characteristics due to crosstalk suggest that the emanation detected will have data and layout dependent variations in power consumption and delay. In the following, we will assume that the coupling capacitance is the dominating factor, and show how this can explain why some signals with the same HD can be distinguished. This will show the potential effect of layout dependent phenomena on classifying microprocessor activity. Our work can easily be extended by including other layout dependent phenomena if a more precise model is needed.

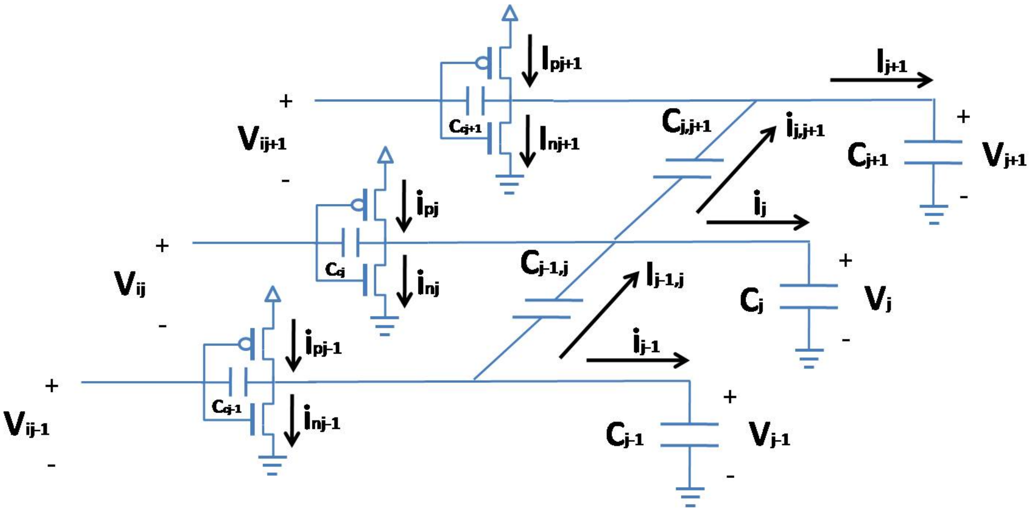

Figure 11.2: Simplified model, assuming load and coupling capacitances to be dominant

## 11.3 Theoretical Considerations

By limiting the model to only coupling and load capacitances, the model in Fig. 11.1 can be simplified as seen in Fig. 11.2. This is a generalization of the model for two lines used in [13] and includes a model of the CMOS inverter.

In order to run simulations in PSPICE, we need an expression for the total energy dissipation, $E_T$. The energy dissipation for wire $j$ in the $p$ and $n$ type transistor can be expressed as:

$$E_{pj} = \int (V_{DD} - V_j) i_{pj} dt \tag{11.1}$$

$$E_{nj} = \int V_j i_{nj} dt \tag{11.2}$$

The overall energy dissipation for an $n$ wire bus is then given by:

$$E_T = \sum_{j=1}^{n} (E_{pj} + E_{nj}) \tag{11.3}$$

Combining and rearranging (11.1), (11.2) and (11.3) the overall energy dissipation can be written as:

$$E_T = \sum_{j=1}^{n} V_{DD} \int i_{pj} dt - \sum_{j=1}^{n} \int V_j (i_{pj} - i_{nj}) dt \tag{11.4}$$

Using Kirchhoff's circuit laws and the current voltage relationship $i(t) = C\frac{dV(t)}{dt}$, the terms $(i_{pj} - i_{nj})$ can be written as:

$$\begin{aligned} i_{pj} - i_{nj} &= (C_j + C_{j,j+1} + C_{j-1,j} + C_{cj}) \frac{dV_j}{dt} \\ &- C_{cj} \frac{dV_{ij}}{dt} - C_{j,j+1} \frac{dV_{j+1}}{dt} - C_{j-1,j} \frac{dV_{j-1}}{dt} \end{aligned} \tag{11.5}$$

Notice that the results in [13] are easily found from (11.4) and (11.5) by setting $n = 2$ (two adjacent lines). Equation (11.4) is used in PSPICE to simulate the total energy dissipation, $\hat{E}_T$, with the following assumptions:

- The transitions on the data bus are concurrent in time. It has been shown [13] that the effect of the coupling capacitance is maximum when transitions occur simultaneously on all bus lines.
- The load capacitances for data bus lines are identical ($C_j = C_L$ for $j = \{1, 2, \cdots, n\}$)
- Coupling capacitances are only found between adjacent line and are identical ($C_{j,j+1} = C_C$ for $j = \{1, 2, \cdots, n-1\}$)

These assumptions are not unrealistic in real bus architecture on a device. If, however, the transitions are shifted in time with more than the rise time of signal, the effect of the coupling capacitance is reduced and the transitions can be regarded as single transitions [13].

In order to compare the simulated energy dissipation ($\hat{E}_T$) with analytic values ($E_T$), different expressions than (11.4) and (11.5) are needed.

It is only when the individual line has a transition, that it is subject to capacitive crosstalk. Quantifying this crosstalk influence has to take into consideration voltage changes on the line itself and one (edges) or two adjacent lines. Let $\delta_j \in \{0, \pm 1\}$ be the normalized voltage change on line $j$, then the voltage change between two lines $j$ and $k$ is $\delta_{j,k} = \delta_j - \delta_k$. The crosstalk influence $\alpha_j$ on line j can then be defined as:

$$\alpha_j = \begin{cases} 0 & no\ transition\ line\ j \\ |\delta_{j,j-1} + \delta_{j,j+1}| & otherwise \end{cases} \tag{11.6}$$

It can be shown that $\alpha_j = \{0, 1, 2\}$ for lines with only one adjacent line (edges), and $\alpha_j = \{0, 1, 2, 3, 4\}$ for lines with two adjacent lines. Let the total crosstalk influence for an $n$ line bus be called a crosstalk index $\alpha$, defined as the sum of the crosstalk influence of each line:

$$\alpha = \sum_{j=1}^{n} \alpha_j \tag{11.7}$$

$$\tag{11.8}$$

If the contributions from the load ($C_L$) and coupling capacitance ($C_C$) are dominant to the dissipated energy, then $E_T$ can be expressed by the following power model [9]:

$$E_T = \frac{1}{2} C_L V_{DD}^2 (k + \alpha\lambda) = E_0(k + \alpha\lambda) \tag{11.9}$$

where $E_0 = \frac{1}{2} C_L V_{DD}^2$, $V_{DD}$ is the power supply voltage, $k$ is the number of transitions on the data bus, $\lambda = C_C/C_L$ and $\alpha$ is the crosstalk index of (11.7) indicating the coupling capacitance induced crosstalk, similar to the crosstalk classes in [5].

In the next section we will use (11.9) to analyze which transition patterns that can be distinguished.

## 11.4 Security Implications

The relationship between energy dissipation, number of transitions, crosstalk index, load capacitance and coupling capacitance in (11.9) can be used to analyze delays and energy dissipation of sub-micron VLSI design [5, 13, 15]. However, we are interested in the security implications of layout dependent phenomena, and in this paper the coupling capacitance in particular. How will a power model (11.9) that includes coupling capacitance effect our ability to predict the energy dissipation of activity in a microprocessor, such as data transfer on a parallel bus?

Let $A$ be the set of possible transitions on an $n$ bit parallel bus. Since "no transition" can be both $0 \to 0$ and $1 \to 1$ there are $|A| = 4^n$ possible transition patterns for an $n$-bit bus. Assuming that each transition pattern's energy dissipation is unique, a model should ideally predict a total of $|A| = 4^n$ energy levels. This may not be possible for at least two reasons: (1) Such model is too complicated. (2) The assumption is incorrect, physical properties are such that multiple transition patterns indeed use the same amount of energy.

Classification by energy dissipation can only distinguish transition patterns by the distinct energy levels explained by the model. A model that assumes energy dissipation proportional to the number of transition, can therefor only distinguish transition pattern into subsets $A_k$, $k = \{0, \cdots, n\}$ being subsets of $A$ that has $k$ transitions. The number of transition patterns in each subset is given by: $|A_k| = 2^n \binom{n}{k}$. The total number of possible transitions on an 8 wire bus ($|A| = 65536$) can be divided into 9 subsets, $A_0, A_1, \cdots, A_8$ based on the number of transitions, $k$. The energy dissipation, $E_T$ (using (11.9) with $\alpha = 0$), associated with each subset $|A_k|$ can be seen in Table 11.2. A model that assumes energy dissipation proportional to the number of transition, can only classify transition pattern by the energy level of these 9 subsets. In Table 11.2 there are e.g. 14336 transition patterns with energy level $3E_0$ that are indistinguishable by the number of transitions alone.

Using the new power model (11.9), taking into consideration the coupling capacitor, each subset $A_k$ can be split into a number of new energy levels. This gives a number of smaller subsets $A_k^\alpha$, $|A_k| > |A_k^\alpha|$ and $\sum_{\forall \alpha \in q_k} |A_k^\alpha| = |A_k|$, where $\alpha$ is the crosstalk index of (11.7) and $q_k$ is the set of possible values of $\alpha$ for $k$ transitions.

Computing $|A_k^\alpha|$ for a fixed number of bus lines $n$ can be done by constructing a table of $(2^k)^2$ elements corresponding to all possible transition patterns. For each of these, first compute the crosstalk index $\alpha$ (11.7), then the energy dissipation $E_T$ (11.9). $|A_k^\alpha|$ can then be computed by counting the table entries for each tuple $\{k, \alpha\}$. Notice that for a finite $n$, there are restrictions on the sets $q_k$ of possible values of $\alpha$. As the number of transitions increase, all energy levels are not possible. This applies to 6,7 and 8 transitions for an 8 bit bus.

The results for an 8 bit bus can be seen in Table 11.2. The results show that taking into consideration the coupling capacitance increases the number of energy levels from 9 in the HD model to 93 in the crosstalk model, e.g. the 14336 transition patterns with 3 transitions previously indistinguishable can now be split into 10 energy levels. The largest increase in energy levels is found for 6 transitions with 21 new energy levels. Note that energy level $\alpha = 20$ does not exist.

Also notice that given an ideal classifier, there is no confusion between subsets of the same $k$ as they all have unique energy levels. There may, however, be confusion between subsets of different $k$. The extent of this confusion is architecture dependent, expressed by $\lambda$, e.g. subset $A_2^6$ has the same energy level as $A_3^2$ if $\lambda = 1/4$, in case they should be treated as one subset. It is easy to show that confusion between transition $A$ (energy $E_{TA}$, $k_A$ transitions and crosstalk index $\alpha_A$) and $B$ (energy $E_{TB}$, $k_B$ transitions and crosstalk index $\alpha_B$) happens when:

$$\lambda_{AB} = \frac{k_B - k_A}{\alpha_A - \alpha_B} \tag{11.10}$$

$\lambda_{AB}$ values that are close to the real $\lambda = C_C/C_L$ indicate subsets that will be difficult to distinguish.

Finally, we have only shown how to split the subset $A_k$ into smaller subsets $A_k^\alpha$ by considering the effect of the coupling capacitance (i.e $\alpha$). This idea can easily be generalized, such that $A_k$ is split into subsets $A_k^\beta$, where $|A_k| > |A_k^\beta|$, and $\beta$ is the influence of other layout dependent phenomena. Examples of phenomena for future work include: variations in coupling and load capacitance, coupling capacitance between line $j$ and $j+2$, inductance, effect of bends in circuit paths and multi layer capacitance (3-dimentional). We believe that the key to identify minute differences in microprocessor activity is to combine several layout dependent phenomena, $\beta_1, \cdots, \beta_m$, such that:

$$|A_k| > |A_k^{\beta_1}| > |A_k^{\beta_1+\beta_2}| > \cdots > |A_k^{\sum_{i=1}^m \beta_i}| \tag{11.11}$$

Table 11.2: The Table shows the number of transition patterns, without ($|A_k|$) and with ($|A_k^\alpha|$) crosstalk influence, belonging to a certain energy level, $E_T$. $k$ is the number of transitions (Hamming Distance) and $\alpha$ is the crosstalk index

| k | $E_T$ [pJ] | $\lvert A_k\rvert$ | $\alpha$ | $E_T$ [pJ] | $\lvert A_k^\alpha\rvert$ | k | $E_T$ [pJ] | $\lvert A_k\rvert$ | $\alpha$ | $E_T$ [pJ] | $\lvert A_k^\alpha\rvert$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 256 | 0 | 0 | 256 | 6 | $6E_0$ | 7168 | 1 | $E_0(6+\lambda)$ | 16 |
| 1 | $E_0$ | 2048 | 1 | $E_0(1+\lambda)$ | 512 | | | | 2 | $E_0(6+2\lambda)$ | 88 |
| | | | 2 | $E_0(1+2\lambda)$ | 1536 | | | | 3 | $E_0(6+3\lambda)$ | 160 |
| 2 | $2E_0$ | 7168 | 1 | $E_0(2+1\lambda)$ | 256 | | | | 4 | $E_0(6+4\lambda)$ | 320 |
| | | | 2 | $E_0(2+2\lambda)$ | 896 | | | | 5 | $E_0(6+5\lambda)$ | 80 |
| | | | 3 | $E_0(2+3\lambda)$ | 2560 | | | | 6 | $E_0(6+6\lambda)$ | 360 |
| | | | 4 | $E_0(2+4\lambda)$ | 2560 | | | | 7 | $E_0(6+7\lambda)$ | 640 |
| | | | 5 | $E_0(2+5\lambda)$ | 256 | | | | 8 | $E_0(6+8\lambda)$ | 960 |
| | | | 6 | $E_0(2+6\lambda)$ | 640 | | | | 9 | $E_0(6+9\lambda)$ | 160 |
| 3 | $3E_0$ | 14336 | 1 | $E_0(3+1\lambda)$ | 128 | | | | 10 | $E_0(6+10\lambda)$ | 560 |
| | | | 2 | $E_0(3+2\lambda)$ | 512 | | | | 11 | $E_0(6+11\lambda)$ | 960 |
| | | | 3 | $E_0(3+3\lambda)$ | 2048 | | | | 12 | $E_0(6+12\lambda)$ | 960 |
| | | | 4 | $E_0(3+4\lambda)$ | 2560 | | | | 13 | $E_0(6+13\lambda)$ | 160 |
| | | | 5 | $E_0(3+5\lambda)$ | 3328 | | | | 14 | $E_0(6+14\lambda)$ | 400 |
| | | | 6 | $E_0(3+6\lambda)$ | 1792 | | | | 15 | $E_0(6+15\lambda)$ | 640 |
| | | | 7 | $E_0(3+7\lambda)$ | 2048 | | | | 16 | $E_0(6+16\lambda)$ | 320 |
| | | | 8 | $E_0(3+8\lambda)$ | 1536 | | | | 17 | $E_0(6+17\lambda)$ | 80 |
| | | | 9 | $E_0(3+9\lambda)$ | 128 | | | | 18 | $E_0(6+18\lambda)$ | 120 |
| | | | 10 | $E_0(3+10\lambda)$ | 256 | | | | 19 | $E_0(6+19\lambda)$ | 160 |
| 4 | $4E_0$ | 17920 | 1 | $E_0(4+\lambda)$ | 64 | | | | 21 | $E_0(6+21\lambda)$ | 16 |
| | | | 2 | $E_0(4+2\lambda)$ | 288 | | | | 22 | $E_0(6+22\lambda)$ | 8 |
| | | | 3 | $E_0(4+3\lambda)$ | 1152 | 7 | $7E_0$ | 2048 | 1 | $E_0(7+\lambda)$ | 8 |
| | | | 4 | $E_0(4+4\lambda)$ | 1728 | | | | 2 | $E_0(7+2\lambda)$ | 48 |
| | | | 5 | $E_0(4+5\lambda)$ | 2496 | | | | 5 | $E_0(7+5\lambda)$ | 48 |
| | | | 6 | $E_0(4+6\lambda)$ | 1824 | | | | 6 | $E_0(7+6\lambda)$ | 240 |
| | | | 7 | $E_0(4+7\lambda)$ | 2816 | | | | 9 | $E_0(7+9\lambda)$ | 120 |
| | | | 8 | $E_0(4+8\lambda)$ | 2304 | | | | 10 | $E_0(7+10\lambda)$ | 480 |
| | | | 9 | $E_0(4+9\lambda)$ | 2496 | | | | 13 | $E_0(7+13\lambda)$ | 160 |
| | | | 10 | $E_0(4+10\lambda)$ | 864 | | | | 14 | $E_0(7+14\lambda)$ | 480 |
| | | | 11 | $E_0(4+11\lambda)$ | 1152 | | | | 17 | $E_0(7+17\lambda)$ | 120 |
| | | | 12 | $E_0(4+12\lambda)$ | 576 | | | | 18 | $E_0(7+18\lambda)$ | 240 |
| | | | 13 | $E_0(4+13\lambda)$ | 64 | | | | 21 | $E_0(7+21\lambda)$ | 48 |
| | | | 14 | $E_0(4+14\lambda)$ | 96 | | | | 22 | $E_0(7+22\lambda)$ | 48 |
| 5 | $5E_0$ | 14336 | 1 | $E_0(5+\lambda)$ | 32 | | | | 25 | $E_0(7+25\lambda)$ | 8 |
| | | | 2 | $E_0(5+2\lambda)$ | 160 | 8 | $8E_0$ | 256 | 0 | $E_0(8)$ | 2 |
| | | | 3 | $E_0(5+3\lambda)$ | 512 | | | | 4 | $E_0(8+4\lambda)$ | 14 |
| | | | 4 | $E_0(5+4\lambda)$ | 896 | | | | 8 | $E_0(8+8\lambda)$ | 42 |
| | | | 5 | $E_0(5+5\lambda)$ | 896 | | | | 12 | $E_0(8+12\lambda)$ | 70 |
| | | | 6 | $E_0(5+6\lambda)$ | 1024 | | | | 16 | $E_0(8+16\lambda)$ | 70 |
| | | | 7 | $E_0(5+7\lambda)$ | 1536 | | | | 20 | $E_0(8+20\lambda)$ | 42 |
| | | | 8 | $E_0(5+8\lambda)$ | 1920 | | | | 24 | $E_0(8+24\lambda)$ | 14 |
| | | | 9 | $E_0(5+9\lambda)$ | 1728 | | | | 28 | $E_0(8+28\lambda)$ | 2 |
| | | | 10 | $E_0(5+10\lambda)$ | 1088 | | | | | | |
| | | | 11 | $E_0(5+11\lambda)$ | 1536 | | | | | | |
| | | | 12 | $E_0(5+12\lambda)$ | 1152 | | | | | | |
| | | | 13 | $E_0(5+13\lambda)$ | 896 | | | | | | |
| | | | 14 | $E_0(5+14\lambda)$ | 256 | | | | | | |
| | | | 15 | $E_0(5+15\lambda)$ | 512 | | | | | | |
| | | | 16 | $E_0(5+16\lambda)$ | 128 | | | | | | |
| | | | 17 | $E_0(5+17\lambda)$ | 32 | | | | | | |
| | | | 18 | $E_0(5+18\lambda)$ | 32 | | | | | | |

### 11.4.1 Classification Performance

Table 11.2 shows that, taking into consideration the coupling capacitance, we are able to increase the number of subsets (or energy levels) $A_k$ to $A_k^\alpha$. For the purpose of comparing alternative detectors we will assume uniform random transition. Thus for an 8 bit bus we would like the detector to extract 16 bits of information, i.e. high or low (2 bits of information) for each of the 8 wires. We will use the ability to extract entropy as our classifier performance indicator. The entropy (i.e bits of information) for a detector, when there are $r$ energy levels, can be calculated using:

$$H(x) = -\sum_{i=1}^{r} p(x_i) log\ p(x_i) \tag{11.12}$$

In the following, we have assumed an 8 bit bus width, thus there are $4^8 = 65536$ possible transitions. Call the detector that can extract 16 bits of information a level detector. If we assume that one only has bus activity when initial and final state are different, and that $0 \rightarrow 1$ and $1 \rightarrow 0$ can be distinguished, an observation will give us the following entropy: $-(1/2log1/2 + 1/4log1/4 + 1/4log1/4) = 3/2$ bits as we cannot distinguish $0 \rightarrow 0$ from $1 \rightarrow 1$, but $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$ can be distinguished. Thus, each observation will give us $3/2$ bits per line. The theoretical optimum for an 8 bit bus with a 'transition detector' would be $8 \cdot 3/2\ bits = 12\ bits$, assuming all observable transitions are distinguishable. In other words, by observing transitions rather than levels, we loose 4 bits (1/2 bit per line) compared to the setting where we would observe the states.

Using the results of Table 11.2, we can now calculate the entropy extracted by a detector that can distinguish HD only ($A_k$) and a detector that can distinguish energy levels due to crosstalk ($A_k^\alpha$).

The entropy extracted by a HD detector is found using (11.12) with 9 energy levels ($r = 9$) and $p(x_i) = |A_{i-1}|/65536$ ($|A_{i-1}|$ from column 3 and 9 Table 11.2) giving an entropy of 2.5 bits. The entropy extracted by a crosstalk detector is found using (11.12) with 93 energy levels ($r = 93$) and $p(x_i) = |A_{i-1}^\alpha|/65536$ ($|A_{i-1}^\alpha|$ from column 6 and 12 Table 11.2) giving an entropy of 5.7 bits.

The difference between the ideal value of a level detector and the entropy extracted by other detectors, represent the amount of guessing needed for classifying an observation. By considering the coupling capacitance and not only HD, we extract more information out of each observation, therefore reducing the amount of "guessing" needed for classification. In the next section we present simulations validating the effect of the coupling capacitance.

## 11.5 Simulations

The simulations are performed in PSPICE with $C_L = 400fF$, $C_C = 250fF$, $V_{dd} = 3V$ and a rise- and fall-time of $200ps$ of the input voltages (same as [13]). The inverter drivers are equal and balanced. Equation (11.4) is used in PSPICE to find the simulated energy dissipation $\hat{E}_T$.

### 11.5.1 Model Validation

Simulations were initially carried out and compared with the results of [5, 13] as a model validation. The results are shown in Table 11.3 and 11.4. Transition pattern refers to transitions in the output voltage $V_j$ (Fig. 11.2) and also shows the number of bus lines used. Column 2 is the number of transitions $k$ followed by the crosstalk index $\alpha$. Theoretical energy, $E_T$, is calculated from (11.9) and simulated energy, $\hat{E}_T$ is from PSPICE simulations.

The simulations for two lines are consistent with [13]. For two wires, as seen in Table 11.3, it is clear that the energy dissipation for two simultaneous transitions is either lower or higher than if treated as two single transitions, depending on the direction of the transitions, as expected. This means that introducing the coupling capacitance it is possible to explain a difference in the energy dissipation for transition patters $00 \leftrightarrow 11$ from $01 \leftrightarrow 10$. Without this difference in energy dissipation the two transition patterns should not be distinguishable.

Simulations of three lines confirms the difference in energy dissipation of the 5 crosstalk classes (Table 11.1) introduced in [5]. Notice that only the transition pattern with the same number of transitions (first and last, second and fourth) can be used to evaluate the effect of the coupling capacitance.

The small differences between analytic and simulated energy dissipation can be explained by simplifications in deriving (11.9) (e.g. omitting leakage currents, such as short-circuit and sub-threshold currents). Having validated our model, all the following simulations are done on an 8 bit bus.

Table 11.3: Dissipated energy when considering crosstalk for 2 adjacent wires

| **Transition pattern** | **Transitions k** | **Crosstalk** $\alpha$ | **Theoretical** $E_T$ **[pJ]** | **Simulated** $\hat{E}_T$ **[pJ]** |
|---|---|---|---|---|
| 00 → 01 | 1 | 1 | 2.9 | 2.7 |
| 00 → 10 | 1 | 1 | 2.9 | 2.7 |
| 00 → 11 | 2 | 0 | 3.6 | 3.5 |
| 01 → 10 | 2 | 4 | 8.1 | 8.0 |

Table 11.4: Dissipated energy when considering crosstalk for bus with 3 lines

| **Transition pattern** | **Transitions k** | **Crosstalk** $\alpha$ | **Theoretical** $E_T$ **[pJ]** | **Simulated** $\hat{E}_T$ **[pJ]** |
|---|---|---|---|---|
| 000 → 111 | 3 | 0 | 5,4 | 5.4 |
| 000 → 011 | 2 | 1 | 4,7 | 4.4 |
| 000 → 010 | 1 | 2 | 4,1 | 3.9 |
| 010 → 100 | 2 | 5 | 9,2 | 9.5 |
| 010 → 101 | 3 | 8 | 14,4 | 14.2 |

### 11.5.2 Results and Discussion

Simulation results for $8$ lines are shown in Table 11.5. The table is not exhaustive, but includes results for all possible subsets $A_k^\alpha$.

The simulated energy levels $\hat{E}_T$ are similar to the analytic values $E_T$. The results confirm that energy consumption is proportional to the number of transitions and the crosstalk index, $\alpha$. The crosstalk index depends on switching activity on adjacent lines and position, edge (one adjacent wire) or middle (two adjacent wires). As seen in Table 11.5, the results also confirm that there is no confusion between energy levels for subsets of an equal number of transitions. However, there may be some confusion between some of the 93 subgroups, e.g. the energy dissipation of subset $A_2^6$ and $A_3^4$ are almost equal. This is expected as $\lambda_{AB} = 0,5$ is close to $\lambda = 0,63$ used in this experiment. Other examples can be found and this reduces the number of subsets depending on how accurate our detector is. A theoretical crosstalk detector capable of separating all 93 energy levels can extract $5.7$ bits of information. It is therefore expected that a practical crosstalk detector will extract less information, due to some subset having almost equal energy levels. Which of the simulated energy levels that should be considered indistinguishable will depend on the accuracy of the detector and the number of observations available. A random loss of $20\%$ of the subsets will still, on average, have an entropy of 5.0. Even with this loss due to similar energy levels, the information gain is still $2.5$ bits compared to the HD detector. The performance of the detectors are summarized in Table 11.6:

### 11.5.3 Case Study and Future Work

In order to probe the practicality of our theory and simulations, we have collected a small set of experimental data. The objective was to see if analysis of electromagnetic side-channel information also supports the division into crosstalk energy levels of Table 11.2 and 11.5.

When classifying two transition patterns by their energy dissipation, we expect a lower probability of error ($P_e$) when the difference in energy level is large and lower $P_e$ as the difference in energy levels decreases. When the energy dissipation of two transition patterns are equal, we don't expect to be able to do any better than flipping a coin. Transition patterns with different number of transitions have a relatively large difference in energy dissipation and are therefore fairly easy to distinguish, as shown in [8].

Table 11.5: Analytic ($E_T$) and simulated ($\hat{E}_T$) dissipated energy when considering crosstalk for bus with 8 lines. k is the number of transitions (Hamming Distance) and $\alpha$ is the crosstalk index

| **Transition pattern** | **k** | $\alpha$ | $E_T$ **[pJ]** | $\hat{E}_T$ **[pJ]** |
|---|---|---|---|---|
| 0000 0000 → 0000 0001 | 1 | 1 | 2,9 | 2.9 |
| 0000 0000 → 0000 0010 | 1 | 2 | 4,1 | 4.1 |
| 0000 0000 → 0000 0011 | 2 | 1 | 4,7 | 4.8 |
| 0000 0000 → 1000 0001 | 2 | 2 | 5,9 | 5.9 |
| 0000 0000 → 0000 0101 | 2 | 3 | 7,0 | 7.0 |
| 0000 0000 → 0000 1010 | 2 | 4 | 8,1 | 8.1 |
| 0000 0010 → 0000 0001 | 2 | 5 | 9,2 | 9.3 |
| 0000 0100 → 0000 0010 | 2 | 6 | 10,4 | 10.1 |
| 0000 0000 → 0000 0111 | 3 | 1 | 6,5 | 6.7 |
| 0000 0000 → 1000 0011 | 3 | 2 | 7,7 | 7.8 |
| 0000 0000 → 0000 1011 | 3 | 3 | 8,8 | 8.9 |
| 0000 0000 → 1000 1001 | 3 | 4 | 9,9 | 10.0 |
| 0000 0000 → 0001 0101 | 3 | 5 | 11,0 | 11.1 |
| 0000 0000 → 0010 1010 | 3 | 6 | 12,2 | 12.3 |
| 0000 0010 → 0000 1001 | 3 | 7 | 13,3 | 13.3 |
| 0000 0100 → 0001 0010 | 3 | 8 | 14,4 | 14.4 |
| 0000 0010 → 0000 0101 | 3 | 9 | 15,5 | 15.4 |
| 0000 0100 → 0000 1010 | 3 | 10 | 16,7 | 16.6 |
| 0000 0000 → 0000 1111 | 4 | 1 | 8,3 | 8.6 |
| 0000 0000 → 1000 0111 | 4 | 2 | 9,5 | 9.6 |
| 0000 0000 → 0001 0111 | 4 | 3 | 10,6 | 10.8 |
| 0000 0000 → 1000 1011 | 4 | 4 | 11,7 | 11.9 |
| 0000 0000 → 0010 1011 | 4 | 5 | 12,8 | 13.0 |
| 0000 0000 → 1001 0101 | 4 | 6 | 14,0 | 14.2 |
| 0000 0000 → 0101 0101 | 4 | 7 | 15,1 | 15.3 |
| 0000 0010 → 1001 0001 | 4 | 8 | 16,2 | 16.0 |
| 0000 0100 → 1001 0010 | 4 | 9 | 17,3 | 17.5 |
| 0000 0010 → 1000 0101 | 4 | 10 | 18,5 | 18.3 |
| 0000 0010 → 0100 0101 | 4 | 11 | 19,6 | 19.4 |
| 0000 0100 → 0100 1010 | 4 | 12 | 20,7 | 20.7 |
| 0000 1010 → 0000 0101 | 4 | 13 | 21,8 | 21.5 |
| 0001 0100 → 0000 1010 | 4 | 14 | 23,0 | 22.7 |
| 0000 0000 → 0001 1111 | 5 | 1 | 10,1 | 10.4 |
| 0000 0000 → 1000 1111 | 5 | 2 | 11,3 | 11.5 |
| 0000 0000 → 0010 1111 | 5 | 3 | 12,4 | 12.6 |
| 0000 0000 → 1001 0111 | 5 | 4 | 13,5 | 13.8 |
| 0000 0000 → 0101 0111 | 5 | 5 | 14,6 | 14.9 |
| 0000 0000 → 1010 1011 | 5 | 6 | 15,8 | 16.0 |
| 0000 0010 → 0111 0001 | 5 | 7 | 16,9 | 16.8 |
| 0000 0010 → 1011 0001 | 5 | 8 | 18,0 | 17.9 |
| 0000 0010 → 0101 1001 | 5 | 9 | 19,1 | 19.3 |
| 0000 0010 → 1010 1001 | 5 | 10 | 20,2 | 20.4 |
| 0000 0010 → 0110 0101 | 5 | 11 | 21,4 | 21.3 |
| 0000 0010 → 1010 0101 | 5 | 12 | 22,5 | 22.5 |
| 0000 0010 → 0101 0101 | 5 | 13 | 23,6 | 23.5 |
| 0000 1010 → 1000 0101 | 5 | 14 | 24,8 | 24.5 |
| 0000 1010 → 0100 0101 | 5 | 15 | 25,9 | 25.6 |
| 0001 0100 → 0100 1010 | 5 | 16 | 27,0 | 27.0 |
| 0000 1010 → 0001 0101 | 5 | 17 | 28,1 | 27.9 |
| 0001 0100 → 0010 1010 | 5 | 18 | 29,3 | 29.1 |
| 0000 0000 → 0011 1111 | 6 | 1 | 11,9 | 12.3 |
| 0000 0000 → 1001 1111 | 6 | 2 | 13,1 | 13.4 |
| 0000 0000 → 0101 1111 | 6 | 3 | 14,2 | 14.5 |
| 0000 0000 → 1010 1111 | 6 | 4 | 15,3 | 15.7 |
| 0000 0001 → 0011 1110 | 6 | 5 | 16,4 | 16.7 |
| 0000 0001 → 1001 1110 | 6 | 6 | 17,6 | 17.9 |
| 0000 0001 → 0101 1110 | 6 | 7 | 18,7 | 19.0 |
| 0000 0001 → 1010 1110 | 6 | 8 | 19,8 | 20.0 |
| 0000 0010 → 0011 1101 | 6 | 9 | 20,9 | 20.8 |
| 0000 0010 → 1001 1101 | 6 | 10 | 22,1 | 21.9 |
| 0000 0010 → 0101 1101 | 6 | 11 | 23,2 | 23.0 |
| 0000 0010 → 1010 1101 | 6 | 12 | 24,3 | 24.1 |
| 0000 0101 → 0011 1010 | 6 | 13 | 25,4 | 25.5 |
| 0000 0101 → 1001 1010 | 6 | 14 | 26,6 | 26.5 |
| 0000 0101 → 0101 1010 | 6 | 15 | 27,7 | 27.7 |
| 0000 0101 → 1010 1010 | 6 | 16 | 28,8 | 28.9 |
| 0000 1010 → 0011 0101 | 6 | 17 | 29,9 | 29.7 |
| 0000 1010 → 1001 0101 | 6 | 18 | 31,1 | 30.9 |
| 0000 1010 → 0101 0101 | 6 | 19 | 32,2 | 32.0 |
| 0010 1010 → 0001 0101 | 6 | 21 | 34,4 | 34.0 |
| 0101 0100 → 0010 1010 | 6 | 22 | 35,6 | 35.2 |
| 0000 0000 → 0111 1111 | 7 | 1 | 13,7 | 14.2 |
| 0000 0000 → 1011 1111 | 7 | 2 | 14,9 | 15.3 |
| 0000 0001 → 0111 1110 | 7 | 5 | 18,2 | 18.5 |
| 0000 0001 → 1011 1110 | 7 | 6 | 19,4 | 19.7 |
| 0000 0010 → 0111 1101 | 7 | 9 | 22,7 | 22.7 |
| 0000 0010 → 1011 1101 | 7 | 10 | 23,9 | 23.8 |
| 0000 0101 → 0111 1010 | 7 | 13 | 27,2 | 27.4 |
| 0000 0101 → 1011 1010 | 7 | 14 | 28,4 | 28.5 |
| 0000 1010 → 0111 0101 | 7 | 17 | 31,7 | 31.6 |
| 0000 1010 → 1011 0101 | 7 | 18 | 32,9 | 32.6 |
| 0001 0101 → 0110 1010 | 7 | 21 | 36,2 | 36.2 |
| 0001 0101 → 1010 1010 | 7 | 22 | 37,4 | 37.3 |
| 0010 1010 → 0101 0101 | 7 | 25 | 40,7 | 40.5 |
| 0000 0000 → 1111 1111 | 8 | 0 | 14,4 | 14.9 |
| 0000 0001 → 1111 1110 | 8 | 4 | 18,9 | 19.3 |
| 0000 0010 → 1111 1101 | 8 | 8 | 23,4 | 23.4 |
| 0000 0101 → 1111 1010 | 8 | 12 | 27,9 | 28.1 |
| 0000 1010 → 1111 0101 | 8 | 16 | 32,4 | 32.3 |
| 0001 0101 → 1110 1010 | 8 | 20 | 36,9 | 36.9 |
| 0010 1010 → 1101 0101 | 8 | 24 | 41,4 | 41.1 |
| 0101 0101 → 1010 1010 | 8 | 28 | 45,9 | 45.6 |

Table 11.6: Comparing the ability to extract information of different detectors for an 8 wire bus

| Type of detector | Entropy (information) [bits] |
|---|---|
| Level detector | 16,0 |
| Optimum transition detector | 12,0 |
| Crosstalk detector (theoretical) | 5,7 |
| Crosstalk detector (simulated) | 5,0 |
| HD detector | 2,5 |

In the following consider two transition patterns $A$ (crosstalk index $\alpha^A$) and $B$ (crosstalk index $\alpha^B$) of an equal number of transitions. Let $\alpha$-distance, $\Delta\alpha = |\alpha^A - \alpha^B|$, be the difference in crosstalk index between transition patterns $A$ and $B$. According to our model

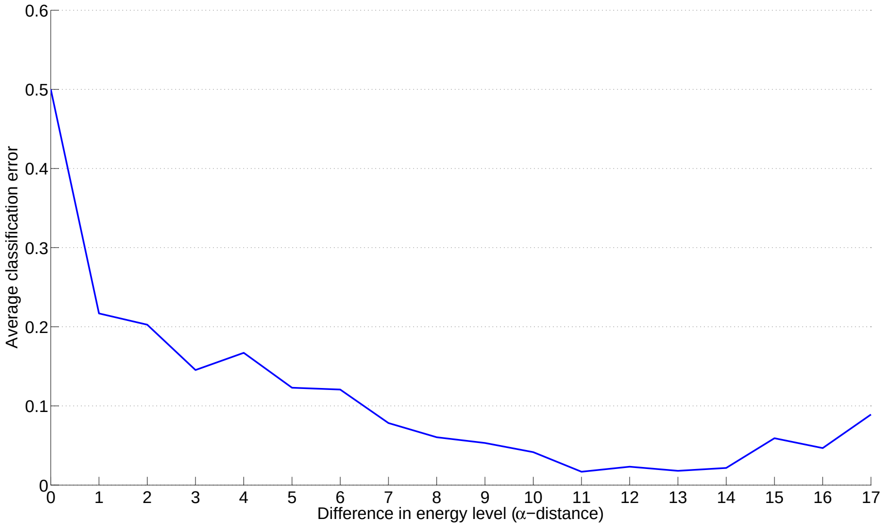

Figure 11.3: Average classification error as a function of difference in energy level (expressed as $\alpha$ distance, $\Delta\alpha$)

(11.9), patterns with $\Delta\alpha = 0$ dissipate the same amount of energy and are therefore assumed to be indistinguishable with an expected classification error, $P_e = 0,5$ (guessing), e.g. $00000000 \rightarrow 00011111$ and $00000000 \rightarrow 11111000$ both belonging to $A_5^1$ (Table 11.2). Patterns with $\Delta\alpha > 0$ are assumed to be distinguishable with $P_e < 0,5$ and $P_e$ is expected to decrease as $\Delta\alpha$ increases, e.g it is expected to be easier (lower $P_e$) to classify $00000000 \rightarrow 00011111$ from $00010100 \rightarrow 00101010$ ($\Delta\alpha = 17$), than $00000000 \rightarrow 00011111$ from $00000000 \rightarrow 01010111$ ($\Delta\alpha = 4$) (Table 11.5), simply because $\Delta\alpha = 17$ indicate a larger difference in energy levels than $\Delta\alpha = 4$.

The following experiment was designed to validate the expected relationship between difference in energy levels, $\Delta\alpha$, and classification error, $P_e$. The experiment was designed to look at transition patterns with 5 transitions (8 bit bus), giving us 18 possible crosstalk indexes (Table 11.2). A program was designed to execute on a smart card (i.e PIC 16F84A microprocessor), such that transition patterns for each of the 18 crosstalk indexes was transferred on the internal data bus. A total of 1000 traces (observations) of the electromagnetic emanation, for each of the 18 transition patterns, were collected. A 10 Gs/s oscilloscope with a broadband E near-field probe was used. The probe was positioned as close to the microprocessor as possible, without any decapsulation.

Classification between all pairs of transition patterns was done according to the Modified Template Attack [8]. This includes feature selection, training and evaluating the performance of a quadratic Bayes classifier (for details refer to [8]). The probability of error, $P_e$, was found from the confusion matrix [6]. Since the classification accuracy depends on how the observations are split, the average of 100 random permutations of 200 training observations and 800 test observations was used. Finally, the average classification error as a function of $\alpha$ distance ($\Delta\alpha$) was calculated and plotted in Fig. 11.3.

The results (Fig. 11.3) show that transition patterns belonging to equal energy levels ($\Delta\alpha = 0$) have $P_e = 0,5$. This is equal to guessing as expected. When the difference in energy level increase (larger $\Delta\alpha$) the results suggest that the average classification error decrease. This supports our simulation/theoretical results. We hypothesize that the discrepancy between our simulation/theoretical results for alpha distance 4 is a consequence of statistical uncertainty/noise in the experimental data. Currently, we cannot offer any explanation for why classification error seems to increase from alpha distance 11/13.

## 11.6 Conclusion

It is known that one can distinguish bus activity generated from signal transitions having different HD. In this paper we elaborate on the theory and simulations on the hypothesis from [9] that layout dependent phenomena, such as inductance and capacitance in and between conductors and radiation properties of circuit elements, can explain why it sometimes is possible to distinguish transition patterns with the same HD. Our simulations show that capacitive crosstalk has a significant effect on gate energy dissipation, and confirm that the dissipated energy from CMOS switching gates depend not only on the HD, but also on the direction of switching activity on nearby data lines. For an 8 bit bus, this increases the number of possible energy levels from 9 (HD) to 93 (crosstalk), and therefore allows us to explain why signals with the same HD sometimes can be distinguished. Where as an HD based detector can provide about 2.5 bits of information per sample, a crosstalk based detector will yield about 5.7 bits (theoretical) or 5.0 bits (simulated) of information per sample - in all cases for an 8 bit bus.

In this paper we have also shown that experimental data, i.e the electromagnetic side-channel of a smart card, suggest that the average classification error gets reduced as $\alpha$ distance (i.e difference in energy level due to capacitive crosstalk) increases. This support our simulations/theoretical results that layout specific phenomena (e.g. capacitance) must be considered when analyzing security implications of electromagnetic side-channels.

## 11.7 Bibliography

[1] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., AND ROHATGI, P. The em side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 29–45. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_4`. 1, 15, 17, 19, 23, 37, 50, 71, 85, 88, 113, 127

[2] AGRAWAL, D., BAKTIR, S., KARAKOYUNLU, D., ROHATGI, P., AND SUNAR, B. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy* (may 2007), pp. 296 –310. 31, 46, 61, 71, 85, 99, 113, 127

[3] CHARI, S., RAO, J., AND ROHATGI, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: `http://dx.doi.org/10.1007/3-540-36400-5_3`. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[4] CHEN, Z., HAIDER, S., AND SCHAUMONT, P. Side-channel leakage in masked circuits caused by higher-order circuit effects. In *Advances in Information Security and Assurance* (2009), vol. 5576 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 327–336. Available from: `http://dx.doi.org/10.1007/978-3-642-02617-1_34`. 40, 99, 108, 113, 127

[5] DUAN, C., CALLE, V., AND KHATRI, S. Efficient on-chip crosstalk avoidance codec design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 17*, 4 (april 2009), 551 –560. 40, 99, 100, 101, 102, 108, 114, 118, 121, 128, 132, 135

[6] DUDA, R., HART, P., AND STORK, D. *Pattern Classification*. John Wiley and Sons, Inc, 2001. 8, 26, 28, 29, 30, 63, 64, 65, 75, 78, 90, 91, 94, 124, 137, 141

[7] Dyrkolbotn, G. O., and Snekkenes, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: `http://dx.doi.org/10.1007/11935308_18`. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[8] Dyrkolbotn, G. O., and Snekkenes, E. Modified template attack: Detecting address bus signals of equal hamming weight. In *Annual Norwegian Information Security Conference - NISK* (2009), Tapir akademisk forlag, pp. 43–56. 39, 40, 41, 42, 46, 47, 99, 109, 113, 122, 124, 127, 135, 137, 142

[9] Dyrkolbotn, G. O., Wold, K., and Snekkenes, E. Security implications of crosstalk in switching cmos gates. In *Information Security Conference - ISC* (2010), vol. 6531 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 269–275. 40, 41, 42, 47, 107, 108, 111, 114, 118, 125

[10] Gandolfi, K., Mourtel, C., and Olivier, F. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES* (2001), vol. 2162 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 251–261. Available from: `http://dx.doi.org/10.1007/3-540-44709-1_21`. 1, 7, 15, 17, 21, 26, 37, 85, 113, 127

[11] Kocher, P., Jaffe, J., and Jun, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: `http://dx.doi.org/10.1007/3-540-48405-1_25`. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[12] Mangard, S., Oswald, E., and Popp, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[13] Moll, F., Roca, M., and Isern, E. Analysis of dissipation energy of switching digital cmos gates with coupled outputs. *Microelectronics Journal 34*, 9 (2003), 833 – 842. Available from: `http://www.sciencedirect.com/science/article/pii/S0026269203001332`. 40, 99, 100, 102, 108, 114, 116, 117, 118, 121, 128, 130, 131, 132, 134, 135

[14] Quisquater, J.-J., and Samyde, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from: `http://portal.acm.org/citation.cfm?id=1250988.1250994`. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[15] Sotiriadis, P., and Chandrakasan, A. Low power bus coding techniques considering inter-wire capacitances. In *Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000* (2000), pp. 507 –510. 40, 99, 108, 114, 118, 128, 132

Appendix A

# *Preproceedings: Security Implications of Crosstalk in Switching CMOS Gates*[1]

## Abstract

The energy dissipation associated with switching in CMOS logic gates can be used to classify the microprocessors activity. It is relatively easy to classify activity by the number of transitions, i.e. Hamming Distance (HD). In this article we consider layout dependent phenomena, such as capacitive crosstalk to derive a more precise power model. We show that for an 8 bit data bus, crosstalk may improve detection performance from 2.5 bits (HD based detector) to theoretical 5.7 bits and simulated 5.0 bits (crosstalk based detector) of information per sample. Thus we have shown that a layout specific phenomenon (capacitance) must be considered when analyzing security implications of power and electromagnetic side-channels. A small case study is also included that support our simulations/theoretical results.

## A.1 Introduction

When a microprocessor executes its program, power consumption (or resulting electromagnetic emanation) can be used to reveal the contents of program and/or data memory of the microprocessor. The correlation between power consumption and microprocessor activity has found many uses: to recover cryptographic keys [10, 3, 11, 1, 9], to reveal hidden hardware faults (trojans) on integrated circuits [2], to control the emanation through subversive software in the Wireless Covert Channel Attack [7] and to reverse engineer the code executed by microprocessors [13].

In side-channel attacks, a common power model used to simulate the power consumption is the Hamming Distance (HD) model, as it is simple and generic [11]. The model assumes the power consumption to be proportional to the number of transitions taking place. If this assumption was correct, signals transmitted on a parallel bus (e.g. intermediate values of the cryptographic algorithm) with the same HD should have equal power consumption and therefore be indistinguishable. This is not always the case, e.g. if Bayes classifier is used, as suggested by the template attack [3]. It has also been demonstrated in [8] that signals with the same number of transitions can be classified using a modified template attack.

The phenomena behind this may be known in the security community, but has received little attention. One article by Chen et al. [4], studies the effect of the coupling capacitance on masking schemes without a detailed examination of the phenomena. In their book ”Power Analysis Attacks”, Mangard et.al. [11] mention power simulation at analog level as ”the most precise way to simulate the power consumption of digital circuits...”. Parasitic elements, such as parasitic capacitances between the wires and unwanted capacitances in the transistors are mentioned. However, it is also stated that it is very common to make simplifications by lumping together extrinsic and intrinsic capacitances into a single

[1] G.O. Dyrkolbotn, K. Wold and E. Snekkenes. Preproceedings presented at *Information Security Conference (ISC), 2010.*

capacitance to ground. This will, in fact, make the model incapable of explaining the results we are addressing in this article.

Parasitic couplings, and the coupling capacitance in particular are, however, a great concern within sub-micron VLSI design [5, 12, 14]. CMOS technology is currently being pushed into deep sub-micron range. As the number of transistors increase, the need for on-chip wiring increases as well and must be scaled accordingly. Parasitic couplings between interconnects, such as on-chip buses, must be taken seriously as they influence both the power consumption and maximum obtainable speed [5]. In [12], Moll et al. did a detailed analysis of the energy dissipation from two metal lines running close together. The lines were driven by CMOS inverters and transitions in one or two wires were studied. The effect of coupling capacitance between the two lines on the power consumption was shown analytically and simulated in HSPICE. The main result was that if two bus lines have transitions in the same or opposite direction at the same time, the total energy is either lower or higher than if the two transitions are treated independently. This is due to the coupling capacitance. Duan et al. [5] focus on crosstalk avoidance codes that aim to reduce the effect of the coupling capacitances by avoiding specific data transition patterns. Their model considers coupling capacitance, $C_C$, between three adjacent lines. They show that 3 bit transition patterns can be divided into 5 crosstalk classes based on the influence of the coupling capacitances, $C_C$. The energy consumption therefore depends on which crosstalk class the transition pattern belong to. The focus of Moll et al. [12] and Duan et al. [5] are both on power consumption and delays caused by the coupling capacitance. They have not considered security implications, such as the ability to use the variation in energy consumption to classify transition patterns. Correlations between data and energy consumption are exactly what side-channel attacks, such as DPA and Template attack, rely upon.

We put forward the hypothesis that layout dependent phenomena, such as parasitic coupling between wires, can explain why it sometimes is possible to distinguish transition patterns with the same HD. In this article we present theory and simulations on some security implications of capacitive crosstalk in CMOS driven data busses. How will the coupling capacitance affect our ability to classify activity in a microprocessor, such as data transfer on a parallel bus? We look at the total dissipated energy from a parallel data bus driven by CMOS inverters. Our model is a generalization of Moll et al's [12], with inverters consisting of two MOSFET transistors, a load capacitance $C_L$ connected to each inverter output and a coupling capacitance $C_C$ connected between each bus line. Our model is generalized to $n$ lines and simulations in PSPICE are done with eight bus lines.

The purpose of our simulation is to show that when the dissipated energy depends on the direction of change of nearby data lines, and not only the number of transitions taking place, the number of possible energy levels dissipating from the bus will increase, thus allowing classification of a larger number of transition patterns. Our hypothesis is that this can be used to explain why some signal with the same HD can be distinguished. Our model can easily take into consideration other layout dependent phenomena, potentially offering an explanation to classification of an even larger set of transition patterns. We will use the ability to extract entropy as our classifier performance indicator and show that a detector capable of detecting energy levels due to crosstalk can extract more information than a detector based on HD only.

This paper is organized as follows: Section A.2 presents the hypothesis of layout dependent phenomena. Section A.3 presents our model and necessary theory to calculate the energy dissipation. Section A.4 is an analytic analysis of security implications. Section A.5 presents simulation results. Finally, a conclusion is drawn in section A.6.

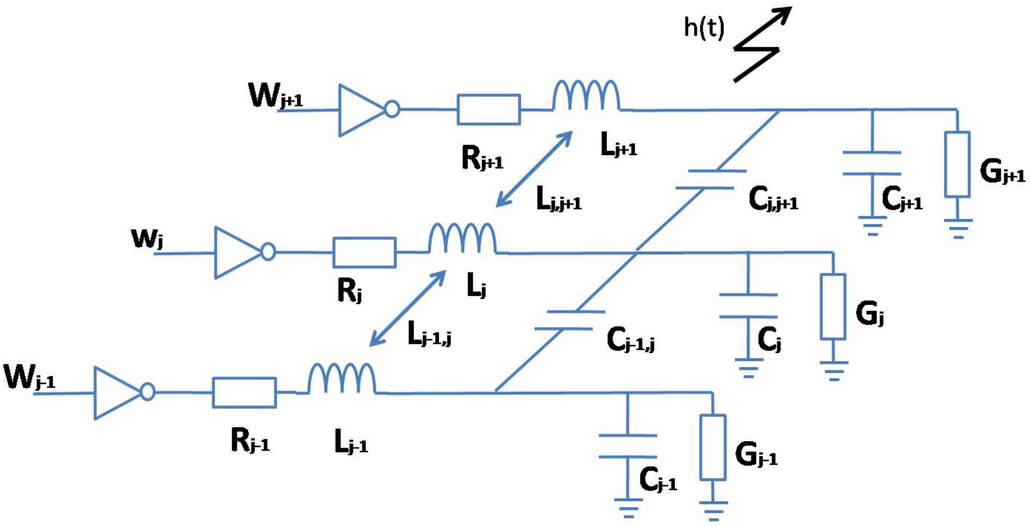

Figure A.1: Model of layout dependent phenomena

## A.2 Layout Dependent Phenomena

In a physical implementation of any circuit (e.g. CMOS based microprocessor) a number of phenomena will influence the energy dissipation and the resulting radiated electromagnetic field. These phenomena include inductance and capacitance of conductors, inductance and capacitance between conductors, wireless transmission characteristics (i.e antenna properties) of conductors and other circuit elements and complex combinations of these phenomena. These phenomena apply to any transistors and wires in a circuit, but we choose to look at a portion of wires running parallel, as we expect them to be relatively good antennas and therefore a good source for side-channel information. This is illustrated in the model of a parallel bus, driven by CMOS inverters, seen in Fig. A.1.

### A.2.1 Inductance and Capacitance of Conductors

Any conductor, $W_j$, carrying an electric current will have an associated distributed resistance $R_j$, inductance $L_j$, conductance $G_j$ and capacitance $C_j$, expressed as a characteristic impedance, $Z_{0j}$. The characteristic impedance is often modeled as an infinite series of lumped components. The inductance $L_j$ and capacitance $C_j$ will both block high frequency signals and act as a low pass filter. Small variations in the length and width of conductors result in small variations in the inductance. Small variations in the area and distance to ground plane result in small variations in the capacitance. There will therefore be small variations in how signals on different conductors (e.g. bus lines) are filtered.

### A.2.2 Inductance and Capacitance between Conductors

Crosstalk can be defined as the coupling of energy between two conductors. Inductive coupling is caused by mutual inductance, $L_{j,j+1}$, (i.e magnetic field) and capacitive coupling is caused by mutual capacitance, $C_{j,j+1}$, (i.e electric field) between wire $j$ and $j+1$. These couplings occur along the entire length of the conductor, but are also modeled as lumped components (Fig. A.1). The interaction of magnetic and electric fields will effectively change the characteristic impedance, $Z_{0j}$, associated with the conductor. This interaction is layout dependent (e.g. distance and length of wires) and will effect both delays and energy dissipation. An important property of crosstalk is its dependency on the activity on the wires. Moll et al. [12] state that, "coupling capacitance is very different from the capacitance to ground because it depends on the switching activity... ". If two lines are low and rise at the same time, the mutual capacitance coupling, $C_{j,j+1}$, does not have to be charged. However, if one line remains low and the other rises, $C_{j,j+1}$ must be charged, resulting in increased rise time and power consumption.

### A.2.3 Wireless Transmission Characteristics

Any circuit element in the microprocessor, conducting electric current, can be considered an antenna. An antenna is a transducer converting electric current into electromagnetic waves, characterized by properties such as: resonant frequency, gain, radiation pattern, impedance, efficiency, bandwidth and polarization. These properties depend on factors such as: amount of current, length/shape and material of the circuit element. In addition, the electromagnetic waves will be influenced by filtering, reflection and interference from surrounding material and circuit elements. The relationship between the current (i.e power consumption) and the electromagnetic wave can be expressed by a transfer function h(t) (see Fig. A.1). Predicting $h(t)$ is not trivial, if possible at all, as most physical systems are not linear by nature. This is left for future work, but it is a fair assumption that relatively long bus lines are good antennas.

### A.2.4 Complex Combinations of Factors

Finally, complex combinations of layout dependent phenomena may be the key to identify minute differences in microprocessor activity, e.g. the radiation efficiency of bus lines combined with data and layout dependencies of the line characteristics due to crosstalk suggest that the emanation detected will have data and layout dependent variations in power consumption and delay. In the following, we will assume that the coupling capacitance is the dominating factor, and show how this can explain why some signals with the same HD can be distinguished. This will show the potential effect of layout dependent phenomena on classifying microprocessor activity. Our work can easily be extended by including more layout dependent phenomena if a more precise result is needed.

## A.3 Theoretical Considerations

By limiting the model to only coupling and load capacitances, the model in Fig. A.1 can be simplified as seen in Fig. A.2. This is a generalization of the model for two lines used by Moll et al. [12] and includes a model of the CMOS inverter.

In order to run simulations in PSPICE, we need an expression for the total energy dissipation, $E_T$. The energy dissipation for wire $j$ in the $p$ and $n$ type transistor can be expressed as:

$$E_{pj} = \int (V_{DD} - V_j) i_{pj} dt \tag{A.1}$$

$$E_{nj} = \int V_j i_{nj} dt \tag{A.2}$$

The overall energy dissipation for an $n$ wire bus is then given by:

$$E_T = \sum_{j=1}^{n} (E_{pj} + E_{nj}) \tag{A.3}$$

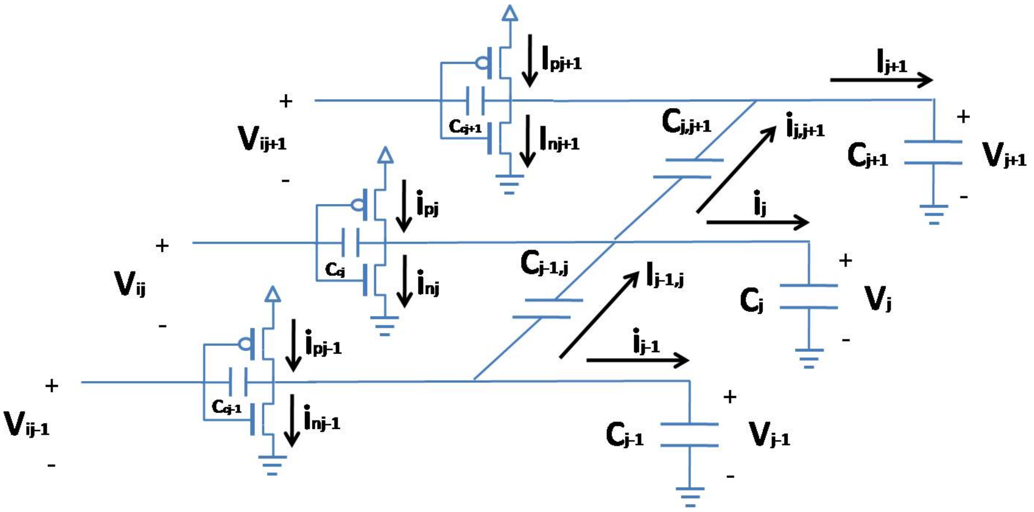

Figure A.2: Simplified model, assuming load and coupling capacitances to be dominant

Combining and rearranging (A.1), (A.2) and (A.3) the overall energy dissipation can be written as:

$$E_T = \sum_{j=1}^{n} V_{DD} \int i_{pj} dt - \sum_{j=1}^{n} \int V_j (i_{pj} - i_{nj}) dt \qquad \text{(A.4)}$$

Using Kirchhoff's circuit laws and the current voltage relationship $i(t) = C\frac{dV(t)}{dt}$, the terms $(i_{pj} - i_{nj})$ can be written as:

$$i_{pj} - i_{nj} = (C_j + C_{j,j+1} + C_{j-1,j} + C_{cj})\frac{dV_j}{dt} - C_{cj}\frac{dV_{ij}}{dt} - C_{j,j+1}\frac{dV_{j+1}}{dt} - C_{j-1,j}\frac{dV_{j-1}}{dt} \text{(A.5)}$$

Notice that the results in [12] are easily found from (A.5) and (A.4) by setting $n = 2$ (two adjacent lines). Equation (A.4) is used in PSPICE to simulate the total energy dissipation, $\hat{E}_T$, with the following assumptions:

- The transitions on the data bus are concurrent in time. Moll et al [12] showed that the effect of the coupling capacitance is maximum when all transitions are synchronized.
- The load capacitances for data bus lines are identical ($C_j = C_L$ for $j = \{1, 2, \cdots, n\}$)
- Coupling capacitances are only found between adjacent line and are identical ($C_{j,j+1} = C_C$ for $j = \{1, 2, \cdots, n-1\}$)

These assumptions are not unrealistic in real bus architecture on a device. If, however, the transitions are shifted in time with more than the rise time of signal, the effect of the coupling capacitance is reduced and the transitions can be regarded as single transitions.

In order to compare the simulated energy dissipation ($\hat{E}_T$) with analytic values ($E_T$), simpler expressions than (A.4) and (A.5) are needed. If the contributions from the load ($C_L$) and coupling capacitance ($C_C$) are dominant to the dissipated energy, then $E_T$ can be expressed by the more intuitive equations:

$$E_T = \frac{1}{2} C_L V_{DD}^2 (k + \alpha\lambda) = E_0 (k + \alpha\lambda) \qquad \text{(A.6)}$$

where $E_0 = \frac{1}{2} C_L V_{DD}^2$, $V_{DD}$ is the power supply voltage, $k$ is the number of transitions on the data bus ($k = 0, 1, 2, \ldots$), $\lambda = C_C / C_L$ and $\alpha$ is the crosstalk index indicating the coupling capacitance induced crosstalk, similar to the crosstalk classes in [5]. For an $n$ line bus the crosstalk index $\alpha$ is the sum of the crosstalk influence of each line:

$$\alpha = \Sigma_{j=1}^{n} \alpha_j \tag{A.7}$$

$$\tag{A.8}$$

Let $\delta_j \in \{0, \pm 1\}$ be the normalized voltage change on line $j$, then $\delta_{j,k} = \delta_j - \delta_k$, and

$$\alpha_j = \begin{cases} 0 & no\ transition\ line\ j \\ |\delta_{j,j-1} + \delta_{j,j+1}| & otherwise \end{cases} \tag{A.9}$$

It can be shown that $\alpha_j = \{0, 1, 2\}$ for lines with only one adjacent line (edges), and $\alpha_j = \{0, 1, 2, 3, 4\}$ for lines with two adjacent lines. In the next section we will use (A.6) and (A.7) to analyze which transition patterns that can be distinguished.

## A.4 Security Implications

The relationship between energy dissipation, number of transitions, the crosstalk index, load capacitance and coupling capacitance in (A.6) can be used to analyze delays and energy dissipation of sub-micron VLSI design [5, 12, 14]. However, we are interested in the security implications of layout dependent phenomena, and in this article the coupling capacitance in particular. How will the coupling capacitance effect our ability to classify activity in a microprocessor, such as data transfer on a parallel bus?

Let $A$ be the set of possible transitions on an $n$ bit parallel bus. Let $U$ be the number of positive transitions and $D$ the number of negative transitions. Since "no transition" can be both $0 \rightarrow 0$ and $1 \rightarrow 1$ there are $|A| = 4^n$ possible transition patterns for an $n$-bit bus. Given an unknown observation of the energy dissipation, the objective is to assign the observation to one of the possible transition patterns (classes), e.g. by using the template attack [3]. A model of the energy dissipation should ideally be able to explain all $|A| = 4^n$ classes. This may not be possible, either because of simplifications to the model or physical properties such that multiple transition patterns indeed use the same amount of energy.

Let $A_k$, $k = \{0, \cdots, n\}$ be the subset of $A$ that has $k = U + D$ transitions. The number of transition patterns in each subset given by:

$$|A_k| = 2^n \binom{n}{k} \tag{A.10}$$

The total number of possible transitions on an 8 wire bus ($|A| = 65536$) can be divided into 9 subsets, $A_0, A_1, \cdots, A_8$ based on the number of transitions, $k$. The energy dissipation, $E_T$ (using (A.6) with $\alpha = 0$), associated with each subset without crosstalk influence and $|A_k|$ can be seen in Table A.1. A model that assumes energy dissipation proportional to the number of transition, can only classify transition pattern by the energy level of these 9 subsets. In Table A.1 there are e.g. 14336 transition patterns with energy level $3E_0$ that are indistinguishable by the number of transitions alone.

Taking into consideration the coupling capacitor of the model in Fig. A.2 and using (A.6), each subset $A_k$ can be split into a number of new energy levels, depending on the crosstalk influence. This gives a number of smaller subsets $A_k^\alpha$, $|A_k| > |A_k^\alpha|$ and $\sum_{all\alpha} |A_k^\alpha| = |A_k|$, where $\alpha$ is the crosstalk index of (A.7). Computing $|A_k^\alpha|$ for a fixed number of bus lines $n$ can be done by constructing a table of $(2^k)^2$ elements corresponding to all possible transition patterns. For each of these, first compute the crosstalk index $\alpha$ (A.7), then the energy dissipation $E_T$ (A.6). $|A_k^\alpha|$ can then be computed by counting the table entries for each tuple $\{k, \alpha\}$. This has been done for an 8 bit bus in Table A.1. The results show that taking into consideration the coupling capacitance increases the number of energy levels from 9 in the HD model to 93 in the crosstalk model, e.g. the 14336 transition patterns

Table A.1: The Table shows the number of transition patterns, without ($|A_k|$) and with ($|A_k^\alpha|$) crosstalk influence, belonging to a certain energy level, $E_T$. $k$ is the number of transitions (Hamming Distance) and $\alpha$ is the crosstalk index

| k | $E_T$ [pJ] | $\|A_k\|$ | $\alpha$ | $E_T$ [pJ] | $\|A_k^\alpha\|$ |
|---|---|---|---|---|---|
| 0 | 0 | 256 | 0 | 0 | 256 |
| 1 | $E_0$ | 2048 | 1 | $E_0(1+\lambda)$ | 512 |
| | | | 2 | $E_0(1+2\lambda)$ | 1536 |
| 2 | $2E_0$ | 7168 | 1 | $E_0(2+1\lambda)$ | 256 |
| | | | 2 | $E_0(2+2\lambda)$ | 896 |
| | | | 3 | $E_0(2+3\lambda)$ | 2560 |
| | | | 4 | $E_0(2+4\lambda)$ | 2560 |
| | | | 5 | $E_0(2+5\lambda)$ | 256 |
| | | | 6 | $E_0(2+6\lambda)$ | 640 |
| 3 | $3E_0$ | 14336 | 1 | $E_0(3+1\lambda)$ | 128 |
| | | | 2 | $E_0(3+2\lambda)$ | 512 |
| | | | 3 | $E_0(3+3\lambda)$ | 2048 |
| | | | 4 | $E_0(3+4\lambda)$ | 2560 |
| | | | 5 | $E_0(3+5\lambda)$ | 3328 |
| | | | 6 | $E_0(3+6\lambda)$ | 1792 |
| | | | 7 | $E_0(3+7\lambda)$ | 2048 |
| | | | 8 | $E_0(3+8\lambda)$ | 1536 |
| | | | 9 | $E_0(3+9\lambda)$ | 128 |
| | | | 10 | $E_0(3+10\lambda)$ | 256 |
| 4 | $4E_0$ | 17920 | 1 | $E_0(4+\lambda)$ | 64 |
| | | | 2 | $E_0(4+2\lambda)$ | 288 |
| | | | 3 | $E_0(4+3\lambda)$ | 1152 |
| | | | 4 | $E_0(4+4\lambda)$ | 1728 |
| | | | 5 | $E_0(4+5\lambda)$ | 2496 |
| | | | 6 | $E_0(4+6\lambda)$ | 1824 |
| | | | 7 | $E_0(4+7\lambda)$ | 2816 |
| | | | 8 | $E_0(4+8\lambda)$ | 2304 |
| | | | 9 | $E_0(4+9\lambda)$ | 2496 |
| | | | 10 | $E_0(4+10\lambda)$ | 864 |
| | | | 11 | $E_0(4+11\lambda)$ | 1152 |
| | | | 12 | $E_0(4+12\lambda)$ | 576 |
| | | | 13 | $E_0(4+13\lambda)$ | 64 |
| | | | 14 | $E_0(4+14\lambda)$ | 96 |
| 5 | $5E_0$ | 14336 | 1 | $E_0(5+\lambda)$ | 32 |
| | | | 2 | $E_0(5+2\lambda)$ | 160 |
| | | | 3 | $E_0(5+3\lambda)$ | 512 |
| | | | 4 | $E_0(5+4\lambda)$ | 896 |
| | | | 5 | $E_0(5+5\lambda)$ | 896 |
| | | | 6 | $E_0(5+6\lambda)$ | 1024 |
| | | | 7 | $E_0(5+7\lambda)$ | 1536 |
| | | | 8 | $E_0(5+8\lambda)$ | 1920 |
| | | | 9 | $E_0(5+9\lambda)$ | 1728 |
| | | | 10 | $E_0(5+10\lambda)$ | 1088 |
| | | | 11 | $E_0(5+11\lambda)$ | 1536 |
| | | | 12 | $E_0(5+12\lambda)$ | 1152 |
| | | | 13 | $E_0(5+13\lambda)$ | 896 |
| | | | 14 | $E_0(5+14\lambda)$ | 256 |
| | | | 15 | $E_0(5+15\lambda)$ | 512 |
| | | | 16 | $E_0(5+16\lambda)$ | 128 |
| | | | 17 | $E_0(5+17\lambda)$ | 32 |
| | | | 18 | $E_0(5+18\lambda)$ | 32 |
| 6 | $6E_0$ | 7168 | 1 | $E_0(6+\lambda)$ | 16 |
| | | | 2 | $E_0(6+2\lambda)$ | 88 |
| | | | 3 | $E_0(6+3\lambda)$ | 160 |
| | | | 4 | $E_0(6+4\lambda)$ | 320 |
| | | | 5 | $E_0(6+5\lambda)$ | 80 |
| | | | 6 | $E_0(6+6\lambda)$ | 360 |
| | | | 7 | $E_0(6+7\lambda)$ | 640 |
| | | | 8 | $E_0(6+8\lambda)$ | 960 |
| | | | 9 | $E_0(6+9\lambda)$ | 160 |
| | | | 10 | $E_0(6+10\lambda)$ | 560 |
| | | | 11 | $E_0(6+11\lambda)$ | 960 |
| | | | 12 | $E_0(6+12\lambda)$ | 960 |
| | | | 13 | $E_0(6+13\lambda)$ | 160 |
| | | | 14 | $E_0(6+14\lambda)$ | 400 |
| | | | 15 | $E_0(6+15\lambda)$ | 640 |
| | | | 16 | $E_0(6+16\lambda)$ | 320 |
| | | | 17 | $E_0(6+17\lambda)$ | 80 |
| | | | 18 | $E_0(6+18\lambda)$ | 120 |
| | | | 19 | $E_0(6+19\lambda)$ | 160 |
| | | | 21 | $E_0(6+21\lambda)$ | 16 |
| | | | 22 | $E_0(6+22\lambda)$ | 8 |
| 7 | $7E_0$ | 2048 | 1 | $E_0(7+\lambda)$ | 8 |
| | | | 2 | $E_0(7+2\lambda)$ | 48 |
| | | | 5 | $E_0(7+5\lambda)$ | 48 |
| | | | 6 | $E_0(7+6\lambda)$ | 240 |
| | | | 9 | $E_0(7+9\lambda)$ | 120 |
| | | | 10 | $E_0(7+10\lambda)$ | 480 |
| | | | 13 | $E_0(7+13\lambda)$ | 160 |
| | | | 14 | $E_0(7+14\lambda)$ | 480 |
| | | | 17 | $E_0(7+17\lambda)$ | 120 |
| | | | 18 | $E_0(7+18\lambda)$ | 240 |
| | | | 21 | $E_0(7+21\lambda)$ | 48 |
| | | | 22 | $E_0(7+22\lambda)$ | 48 |
| | | | 25 | $E_0(7+25\lambda)$ | 8 |
| 8 | $8E_0$ | 256 | 0 | $E_0(8)$ | 2 |
| | | | 4 | $E_0(8+4\lambda)$ | 14 |
| | | | 8 | $E_0(8+8\lambda)$ | 42 |
| | | | 12 | $E_0(8+12\lambda)$ | 70 |
| | | | 16 | $E_0(8+16\lambda)$ | 70 |
| | | | 20 | $E_0(8+20\lambda)$ | 42 |
| | | | 24 | $E_0(8+24\lambda)$ | 14 |
| | | | 28 | $E_0(8+28\lambda)$ | 2 |

with 3 transitions previously indistinguishable can now be split into 10 energy levels. The largest increase in energy levels is found for 6 transitions with 21 new energy levels. Notice that for a finite $n$, there are restrictions on the values of $\alpha$ as the number of transitions increase, i.e all energy levels are not possible. This applies to 6,7 and 8 transitions for an 8 bit bus.

Also notice that given an ideal classifier, there is no confusion between subsets of the same $k$ as they all have unique energy levels. There may, however, be confusion between subsets of different $k$. The extent of this confusion is architecture dependent, expressed by $\lambda$, e.g. subset $A_2^6$ have the same energy level as $A_3^2$ if $\lambda = 1/4$, in case they should be treated as one subset. It is easy to show that confusion between transition $A$ (energy $E_{TA}$, $k_A$ transitions and crosstalk index $\alpha_A$) and $B$ (energy $E_{TB}$, $k_B$ transitions and crosstalk index $\alpha_B$) happens when:

$$\lambda_{AB} = \frac{k_B - k_A}{\alpha_A - \alpha_B} \tag{A.11}$$

$\lambda_{AB}$ values that are close to the real $\lambda = C_C/C_L$ indicate subsets that will be difficult to distinguish.

Finally, we have only shown how to split the subset $A_k$ into smaller subsets $A_k^\alpha$ by considering the effect of the coupling capacitance (i.e $\alpha$). This idea can easily be generalized, such that $A_k$ is split into subsets $A_k^\beta$, where $|A_k| > |A_k^\beta|$, and $\beta$ is the influence of other layout dependent phenomena. Examples of phenomena for future work include: variations in coupling and load capacitance, coupling capacitance between line $j$ and $j+2$, inductance, effect of bends in circuit paths and multi layer capacitance (3-dimentional).

### A.4.1 Classification Performance

Table A.1 shows that, taking into consideration the coupling capacitance, we are able to increase the number of subsets (or energy levels) $A_k$ to $A_k^\alpha$. For the purpose of comparing alternative detectors we will assume uniform random transition. Thus for an 8 bit bus we would like the detector to extract 16 bits. We will use the ability to extract entropy as our classifier performance indicator. The entropy (i.e bits of information) for a detector, when there are $r$ energy levels, can be calculated using:

$$H(x) = -\sum_{i=1}^{r} p(x_i) log\ p(x_i) \tag{A.12}$$

In the following, we have assumed an 8 bit bus width, thus there are $4^8 = 65536$ possible transitions. Call the detector that can extract 16 bits of information a level detector. If we assume that one only has bus activity when initial and final state are different, and that $0 \rightarrow 1$ and $1 \rightarrow 0$ can be distinguished, an observation will give us the following entropy: $-(1/2log1/2 + 1/4log1/4 + 1/4log1/4) = 3/2$ bits as we cannot distinguish $0 \rightarrow 0$ from $1 \rightarrow 1$, but $0 \rightarrow 0$,$0 \rightarrow 1$, $1 \rightarrow 0$ can be distinguished. Thus, each observation will give us 3/2 bits per line. The theoretical optimum for an 8 bit bus with a 'transition detector' would be $8 \cdot 3/2\ bits = 12\ bits$, assuming all observable transitions are distinguishable. In other words, by observing transitions rather than levels, we loose 4 bits (1/2 bit per line) compared to the setting where we would observe the states.

Using the results of Table A.1, we can now calculate the entropy of a detector that can distinguish HD only ($A_k$) and a detector that can distinguish energy levels due to crosstalk ($A_k^\alpha$).

The entropy of an HD detector is found using (A.12) with $r = 9$ and $p(x_i) = |A_{i-1}|/65536$ giving an entropy of 2.5 bits. The entropy of a crosstalk detector is found using (A.12) with $r = 93$ and $p(x_i) = |A_{i-1}^\alpha|/65536$ giving an entropy of 5.7 bits.

The difference between the ideal value of a level detector and the entropy of other detectors, represent the amount of guessing needed for classifying an observation. By considering the coupling capacitance and not only HD, we extract more information out of each observation, therefore reducing the amount of "guessing" needed for classification. In the next section we present simulations validating the effect of the coupling capacitance.

## A.5 Simulations

The simulations are performed in PSPICE with $C_L = 400fF$, $C_C = 250fF$, $V_{dd} = 3V$ and a rise- and fall-time of $200ps$ of the input voltages (same as [12]). The inverter drivers are equal and balanced. Equation (A.4) and (A.5) are used in PSPICE to find the simulated energy dissipation $\hat{E}_T$.

Table A.2: Dissipated energy when considering crosstalk for 2 adjacent wires

| **Transition pattern** | **Transitions k** | **Crosstalk $\alpha$** | **Theoretical $E_T$ [pJ]** | **Simulated $\hat{E}_T$ [pJ]** |
|---|---|---|---|---|
| 00 → 01 | 1 | 1 | 2.9 | 2.7 |
| 00 → 10 | 1 | 1 | 2.9 | 2.7 |
| 00 → 11 | 2 | 0 | 3.6 | 3.5 |
| 01 → 10 | 2 | 4 | 8.1 | 8.0 |

### A.5.1 Model Validation

Simulations were initially carried out and compared with the results of [5, 12] as a model validation. For two wires, as seen in Table A.2, it is clear that the energy dissipation for two simultaneous transitions is either lower or higher than if treated as two single transitions, depending on the direction of the transitions, as expected. The small differences between analytic and simulated energy dissipation can be explained by simplifications in deriving (A.6) (e.g. omitting leakage currents, such as short-circuit and sub-threshold currents). Having validated our model, all the following simulations are done on an 8 bit bus.

### A.5.2 Results and Discussion

Simulation results for $8$ lines are shown in Table A.3. The table is not exhaustive, but includes results for all possible subsets $A_k^\alpha$.

The simulated energy levels $\hat{E}_T$ are similar to the analytic values. The results confirm that energy consumption is proportional to the number of transitions and the crosstalk index, $\alpha$. The crosstalk index depends on switching activity on adjacent lines and position, edge (one adjacent wire) or middle (two adjacent wires). As seen in Table A.3, the results also confirm that there is no confusion between energy levels for subsets of an equal number of transitions. However, there may be some confusion between some of the 93 subgroups, e.g. the energy dissipation of subset $A_2^6$ and $A_3^4$ are almost equal. This is expected as $\lambda_{AB} = 0,5$ is close to $\lambda = 0,63$ used in this experiment. Other examples can be found and this reduces the number of subsets depending on how accurate our detector is. As seen earlier, a theoretical crosstalk detector capable of separating all 93 energy levels will have an entropy of $5.7$. It is therefore expected that a practical crosstalk detector will have a smaller entropy. It is difficult to decide which of the simulated energy levels should be merged together, as this will depend on the accuracy of the detector and the number of observations available. A random loss of $20\%$ of the subsets will still, on average, have an entropy of 5.0. Even with this loss due to similar energy levels, the information gain is still $2.5$ bits compared to the HD detector. The performance of the detectors are summarized in Table A.4:

### A.5.3 Case Study and Future Work

In order to probe the practicality of our theory and simulations, we have collected a small set of experimental data. The objective was to see if analysis of electromagnetic side-channel information also supports the division into crosstalk energy levels of Table A.1 and A.3.

A total of 1000 traces (observations) of each of the 18 crosstalk indexes for transitions pattern with Hamming Distance 5 where collected. The target was a smart card (i.e PIC 16F84A microprocessor). The electromagnetic emanation was captured using a broadband E near-field probe positioned as close to the microprocessor as possible, without any decapsulation.

Analysis was done according to the Modified Template Attack [8], and includes feature selection, training and evaluating the performance of a quadratic Bayes classifier (for de-

Table A.3: Analytic ($E_T$) and simulated ($\hat{E}_T$) dissipated energy when considering crosstalk for bus with 8 lines. k is the number of transitions (Hamming Distance) and $\alpha$ is the crosstalk index

| Transition pattern | k | $\alpha$ | $E_T$ [pJ] | $\hat{E}_T$ [pJ] |
|---|---|---|---|---|
| 0000 0000 → 0000 0001 | 1 | 1 | 2,9 | 2.9 |
| 0000 0000 → 0000 0010 | 1 | 2 | 4,1 | 4.1 |
| 0000 0000 → 0000 0011 | 2 | 1 | 4,7 | 4.8 |
| 0000 0000 → 1000 0001 | 2 | 2 | 5,9 | 5.9 |
| 0000 0000 → 0000 0101 | 2 | 3 | 7,0 | 7.0 |
| 0000 0000 → 0000 1010 | 2 | 4 | 8,1 | 8.1 |
| 0000 0010 → 0000 0001 | 2 | 5 | 9,2 | 9.3 |
| 0000 0100 → 0000 0010 | 2 | 6 | 10,4 | 10.1 |
| 0000 0000 → 0000 0111 | 3 | 1 | 6,5 | 6.7 |
| 0000 0000 → 1000 0011 | 3 | 2 | 7,7 | 7.8 |
| 0000 0000 → 0000 1011 | 3 | 3 | 8,8 | 8.9 |
| 0000 0000 → 1000 1001 | 3 | 4 | 9,9 | 10.0 |
| 0000 0000 → 0001 0101 | 3 | 5 | 11,0 | 11.1 |
| 0000 0000 → 0010 1010 | 3 | 6 | 12,2 | 12.3 |
| 0000 0010 → 0000 1001 | 3 | 7 | 13,3 | 13.3 |
| 0000 0100 → 0001 0010 | 3 | 8 | 14,4 | 14.4 |
| 0000 0010 → 0000 0101 | 3 | 9 | 15,5 | 15.4 |
| 0000 0100 → 0000 1010 | 3 | 10 | 16,7 | 16.6 |
| 0000 0000 → 0000 1111 | 4 | 1 | 8,3 | 8.6 |
| 0000 0000 → 1000 0111 | 4 | 2 | 9,5 | 9.6 |
| 0000 0000 → 0001 0111 | 4 | 3 | 10,6 | 10.8 |
| 0000 0000 → 1000 1011 | 4 | 4 | 11,7 | 11.9 |
| 0000 0000 → 0010 1011 | 4 | 5 | 12,8 | 13.0 |
| 0000 0000 → 1001 0101 | 4 | 6 | 14,0 | 14.2 |
| 0000 0000 → 0101 0101 | 4 | 7 | 15,1 | 15.3 |
| 0000 0010 → 1001 0001 | 4 | 8 | 16,2 | 16.0 |
| 0000 0100 → 1001 0010 | 4 | 9 | 17,3 | 17.5 |
| 0000 0010 → 1000 0101 | 4 | 10 | 18,5 | 18.3 |
| 0000 0010 → 0100 0101 | 4 | 11 | 19,6 | 19.4 |
| 0000 0100 → 0100 1010 | 4 | 12 | 20,7 | 20.7 |
| 0000 1010 → 0000 0101 | 4 | 13 | 21,8 | 21.5 |
| 0001 0100 → 0000 1010 | 4 | 14 | 23,0 | 22.7 |
| 0000 0000 → 0001 1111 | 5 | 1 | 10,1 | 10.4 |
| 0000 0000 → 1000 1111 | 5 | 2 | 11,3 | 11.5 |
| 0000 0000 → 0010 1111 | 5 | 3 | 12,4 | 12.6 |
| 0000 0000 → 1001 0111 | 5 | 4 | 13,5 | 13.8 |
| 0000 0000 → 0101 0111 | 5 | 5 | 14,6 | 14.9 |
| 0000 0000 → 1010 1011 | 5 | 6 | 15,8 | 16.0 |
| 0000 0010 → 0111 0001 | 5 | 7 | 16,9 | 16.8 |
| 0000 0010 → 1011 0001 | 5 | 8 | 18,0 | 17.9 |
| 0000 0010 → 0101 1001 | 5 | 9 | 19,1 | 19.3 |
| 0000 0010 → 1010 1001 | 5 | 10 | 20,2 | 20.4 |
| 0000 0010 → 0110 0101 | 5 | 11 | 21,4 | 21.3 |
| 0000 0010 → 1010 0101 | 5 | 12 | 22,5 | 22.5 |
| 0000 0010 → 0101 0101 | 5 | 13 | 23,6 | 23.5 |
| 0000 1010 → 1000 0101 | 5 | 14 | 24,8 | 24.5 |
| 0000 1010 → 0100 0101 | 5 | 15 | 25,9 | 25.6 |
| 0001 0100 → 0100 1010 | 5 | 16 | 27,0 | 27.0 |
| 0000 1010 → 0001 0101 | 5 | 17 | 28,1 | 27.9 |
| 0001 0100 → 0010 1010 | 5 | 18 | 29,3 | 29.1 |
| 0000 0000 → 0011 1111 | 6 | 1 | 11,9 | 12.3 |
| 0000 0000 → 1001 1111 | 6 | 2 | 13,1 | 13.4 |
| 0000 0000 → 0101 1111 | 6 | 3 | 14,2 | 14.5 |
| 0000 0000 → 1010 1111 | 6 | 4 | 15,3 | 15.7 |
| 0000 0001 → 0011 1110 | 6 | 5 | 16,4 | 16.7 |
| 0000 0001 → 1001 1110 | 6 | 6 | 17,6 | 17.9 |
| 0000 0001 → 0101 1110 | 6 | 7 | 18,7 | 19.0 |
| 0000 0001 → 1010 1110 | 6 | 8 | 19,8 | 20.0 |
| 0000 0010 → 0011 1101 | 6 | 9 | 20,9 | 20.8 |
| 0000 0010 → 1001 1101 | 6 | 10 | 22,1 | 21.9 |
| 0000 0010 → 0101 1101 | 6 | 11 | 23,2 | 23.0 |
| 0000 0010 → 1010 1101 | 6 | 12 | 24,3 | 24.1 |
| 0000 0101 → 0011 1010 | 6 | 13 | 25,4 | 25.5 |
| 0000 0101 → 1001 1010 | 6 | 14 | 26,6 | 26.5 |
| 0000 0101 → 0101 1010 | 6 | 15 | 27,7 | 27.7 |
| 0000 0101 → 1010 1010 | 6 | 16 | 28,8 | 28.9 |
| 0000 1010 → 0011 0101 | 6 | 17 | 29,9 | 29.7 |
| 0000 1010 → 1001 0101 | 6 | 18 | 31,1 | 30.9 |
| 0000 1010 → 0101 0101 | 6 | 19 | 32,2 | 32.0 |
| 0010 1010 → 0001 0101 | 6 | 21 | 34,4 | 34.0 |
| 0101 0100 → 0010 1010 | 6 | 22 | 35,6 | 35.2 |
| 0000 0000 → 0111 1111 | 7 | 1 | 13,7 | 14.2 |
| 0000 0000 → 1011 1111 | 7 | 2 | 14,9 | 15.3 |
| 0000 0001 → 0111 1110 | 7 | 5 | 18,2 | 18.5 |
| 0000 0001 → 1011 1110 | 7 | 6 | 19,4 | 19.7 |
| 0000 0010 → 0111 1101 | 7 | 9 | 22,7 | 22.7 |
| 0000 0010 → 1011 1101 | 7 | 10 | 23,9 | 23.8 |
| 0000 0101 → 0111 1010 | 7 | 13 | 27,2 | 27.4 |
| 0000 0101 → 1011 1010 | 7 | 14 | 28,4 | 28.5 |
| 0000 1010 → 0111 0101 | 7 | 17 | 31,7 | 31.6 |
| 0000 1010 → 1011 0101 | 7 | 18 | 32,9 | 32.6 |
| 0001 0101 → 0110 1010 | 7 | 21 | 36,2 | 36.2 |
| 0001 0101 → 1010 1010 | 7 | 22 | 37,4 | 37.3 |
| 0010 1010 → 0101 0101 | 7 | 25 | 40,7 | 40.5 |
| 0000 0000 → 1111 1111 | 8 | 0 | 14,4 | 14.9 |
| 0000 0001 → 1111 1110 | 8 | 4 | 18,9 | 19.3 |
| 0000 0010 → 1111 1101 | 8 | 8 | 23,4 | 23.4 |
| 0000 0101 → 1111 1010 | 8 | 12 | 27,9 | 28.1 |
| 0000 1010 → 1111 0101 | 8 | 16 | 32,4 | 32.3 |
| 0001 0101 → 1110 1010 | 8 | 20 | 36,9 | 36.9 |
| 0010 1010 → 1101 0101 | 8 | 24 | 41,4 | 41.1 |
| 0101 0101 → 1010 1010 | 8 | 28 | 45,9 | 45.6 |

Table A.4: Comparing the performance of different detectors for an 8 wire bus

| type of detector | Entropy (information) [bits] |
|---|---|
| Level detector | 16,0 |
| Optimum transition detector | 12,0 |
| Crosstalk detector (theoretical) | 5,7 |
| Crosstalk detector (simulated) | 5,0 |
| HD detector | 2,5 |

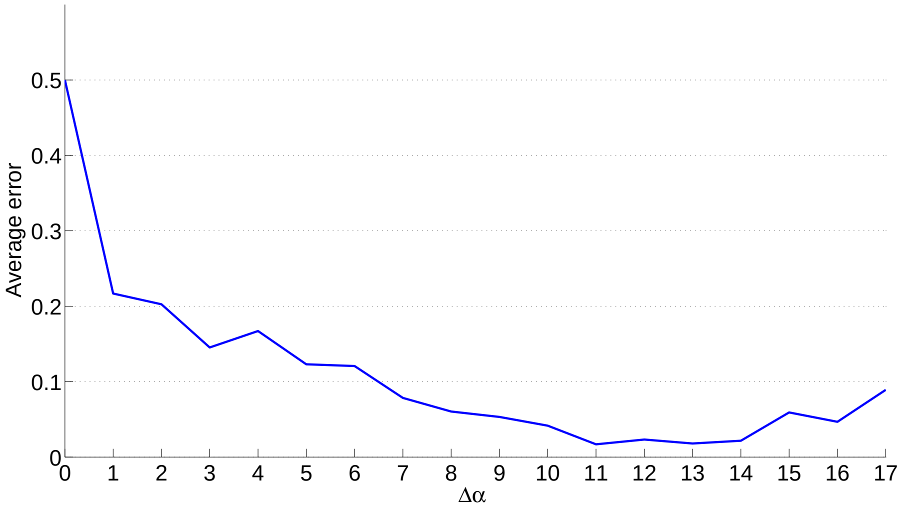

Figure A.3: Average classification error as a function of $\alpha$ distance, $\Delta\alpha$

tails refer to [8]). The probability of error, $P_e$, was found from the confusion matrix [6]. The classification accuracy depends on how the observations are split, therefore, the average of 100 random permutations of 200 training observations and 800 test observations was used. Finally, the average classification error as a function of $\alpha$ distance ($\Delta\alpha = |\alpha_i - \alpha_j|$) was calculated.

Figure A.3 is a plot of our experimental data, suggesting that the average classification error gets reduced as alpha distance increases. This supports our simulation/theoretical results. We hypothesize that the discrepancy between our simulation/theoretical results for alpha distance 4 is a consequence of statistical uncertainty/noise in the experimental data. Currently, we cannot offer any explanation for why classification error seems to increase from alpha distance 11/13.

## A.6 Conclusion

It is known that one can distinguish bus activity generated from signal transitions having different HD. In this article we put forward the hypothesis that layout dependent phenomena, such as inductance and capacitance in and between conductors and radiation properties of circuit elements, can explain why it sometimes is possible to distinguish transition patterns with the same HD. In this article we provide a general model for layout dependent phenomena and study some security implications of the capacitive crosstalk between parallel wires. Our simulations show that capacitive crosstalk has a significant effect on gate energy dissipation. Our results confirm that the dissipated energy from CMOS switching gates depend not only on the HD, but also on the direction of switching activity on nearby data lines. For an 8 bit bus, this increases the number of possible energy levels from 9 (HD) to 93 (crosstalk), and therefore allows us to explain why signals with the same HD sometimes can be distinguished. Where as an HD based detector can provide about 2.5 bits of information per sample, a crosstalk based detector will yield about 5.7 bits (theoretical) or 5.0 bits (simulated) of information per sample - in all cases for an 8 bit bus. Thus we have shown that a layout specific phenomenon (capacitance) must be considered when analyzing security implications of electromagnetic side-channels.

## A.7 Bibliography

[1] AGRAWAL, D., ARCHAMBEAULT, B., RAO, J., AND ROHATGI, P. The em side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 29–45. Available from: http://dx.doi.org/10.1007/3-540-36400-5_4. 1, 15, 17, 19, 23, 37, 50, 71, 85, 88, 113, 127

[2] AGRAWAL, D., BAKTIR, S., KARAKOYUNLU, D., ROHATGI, P., AND SUNAR, B. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy* (may 2007), pp. 296 –310. 31, 46, 61, 71, 85, 99, 113, 127

[3] CHARI, S., RAO, J., AND ROHATGI, P. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES* (2003), vol. 2523 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 51–62. Available from: http://dx.doi.org/10.1007/3-540-36400-5_3. 1, 8, 28, 29, 30, 37, 39, 40, 45, 46, 50, 71, 72, 85, 86, 88, 89, 91, 94, 95, 96, 99, 113, 127, 132

[4] CHEN, Z., HAIDER, S., AND SCHAUMONT, P. Side-channel leakage in masked circuits caused by higher-order circuit effects. In *Advances in Information Security and Assurance* (2009), vol. 5576 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 327–336. Available from: http://dx.doi.org/10.1007/978-3-642-02617-1_34. 40, 99, 108, 113, 127

[5] DUAN, C., CALLE, V., AND KHATRI, S. Efficient on-chip crosstalk avoidance codec design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 17*, 4 (april 2009), 551 –560. 40, 99, 100, 101, 102, 108, 114, 118, 121, 128, 132, 135

[6] DUDA, R., HART, P., AND STORK, D. *Pattern Classification*. John Wiley and Sons, Inc, 2001. 8, 26, 28, 29, 30, 63, 64, 65, 75, 78, 90, 91, 94, 124, 137, 141

[7] DYRKOLBOTN, G. O., AND SNEKKENES, E. A wireless covert channel on smart cards (short paper). In *Information and Communications Security - ICICS* (2006), vol. 4307 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 249–259. Available from: http://dx.doi.org/10.1007/11935308_18. 7, 37, 38, 41, 45, 46, 59, 60, 62, 63, 67, 71, 72, 73, 76, 85, 95, 99, 113, 127

[8] DYRKOLBOTN, G. O., AND SNEKKENES, E. Modified template attack: Detecting address bus signals of equal hamming weight. In *Annual Norwegian Information Security Conference - NISK* (2009), Tapir akademisk forlag, pp. 43–56. 39, 40, 41, 42, 46, 47, 99, 109, 113, 122, 124, 127, 135, 137, 142

[9] GANDOLFI, K., MOURTEL, C., AND OLIVIER, F. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES* (2001), vol. 2162 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 251–261. Available from: http://dx.doi.org/10.1007/3-540-44709-1_21. 1, 7, 15, 17, 21, 26, 37, 85, 113, 127

[10] KOCHER, P., JAFFE, J., AND JUN, B. Differential power analysis. In *Advances in Cryptology* (1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 388–397. Available from: http://dx.doi.org/10.1007/3-540-48405-1_25. 1, 7, 8, 26, 27, 45, 47, 50, 71, 76, 85, 88, 99, 107, 113, 127

[11] MANGARD, S., OSWALD, E., AND POPP, T. *Power Analysis Attack - Revealing the Secret of Smart Cards*. Springer, 2007. 1, 7, 11, 12, 13, 16, 22, 26, 27, 30, 40, 47, 85, 87, 99, 107, 108, 113, 114, 127

[12] MOLL, F., ROCA, M., AND ISERN, E. Analysis of dissipation energy of switching digital cmos gates with coupled outputs. *Microelectronics Journal 34*, 9 (2003), 833 – 842. Available from: `http://www.sciencedirect.com/science/article/pii/S0026269203001332`. 40, 99, 100, 102, 108, 114, 116, 117, 118, 121, 128, 130, 131, 132, 134, 135

[13] QUISQUATER, J.-J., AND SAMYDE, D. Automatic code recognition for smart cards using a kohonen neural network. In *Conference on Smart Card Research and Advanced Application Conference* (Berkeley, CA, USA, 2002), USENIX Association. Available from: `http://portal.acm.org/citation.cfm?id=1250988.1250994`. 1, 8, 37, 39, 45, 50, 85, 86, 88, 94, 96, 99, 113, 127

[14] SOTIRIADIS, P., AND CHANDRAKASAN, A. Low power bus coding techniques considering inter-wire capacitances. In *Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000* (2000), pp. 507 –510. 40, 99, 108, 114, 118, 128, 132

Appendix B

# *Majority Voting*

In some scenarios, it may be possible to observe and measure the same activity more than once. A program may reuse some values over and over again (e.g. User name, password, PIN code or cryptographic key). A specific program (or part of program) may be used several times or maybe a subversive code or trojan can force repetitive execution. If more than one observation of the same activity is indeed available, several approaches can be used to reduce the probability of error. Two alternatives are evident: data fusion first or classification first. Data fusion first has to combine all observations into a single observation. A simple and commonly used method is simply to calculate the average observation. Regardless of the fusion method, classification is done using a single observation.

The other approach is to classify all individual observations first and then fuse the data into a single decision afterwards. We suggest Majority Voting, that simply decides in favor of the class with the majority of the classification decisions (votes). Majority Voting consist of the following three steps.

1. Collect n observations of the same phenomenon
2. Classify each observation individually
3. Decide in favor of the class with the majority of the classification decisions (votes)

This algorithm can be considered a binomial experiment (also known as Bernouilli Process) [3], and the probability of error can be reduced as much as desired, given enough observations. The properties of a binomial experiment are:

- The experiment consists of a fixed number of trials, n
- Each trial has two possible outcomes (success or failure)
- The probability of success, p, and failure, 1-p, is constant
- Each trial is independent of other trials

The probability of r successes in n trials is then given by:

$$P[r|n,p] = \binom{n}{r} p^r (1-p)^{n-r} \tag{B.1}$$

Assuming that $n$ independent observations of an activity are classified into two classes with a probability of error $P_e = 1 - p$, the properties of a binomial experiment holds. If more than two classes are to be classified, the method can easily extend to $c$ classes using pairwise classifications [1]. The classification error for majority voting is given by the cumulative probability $P_{e,majority} = P(r < n/2|n,p)$. An interesting questions is then: How many observations are necessary to get the classification error, $P_{e,majority}$ , below a given threshold, $t$? For a fixed probability of success $p$, this means finding the smallest number of experiment that satisfy, $P_{e,majority} < t$. The minimum number of observations $n$ needed, as a function of the probability of error $P_e$, is shown in figure B.1.

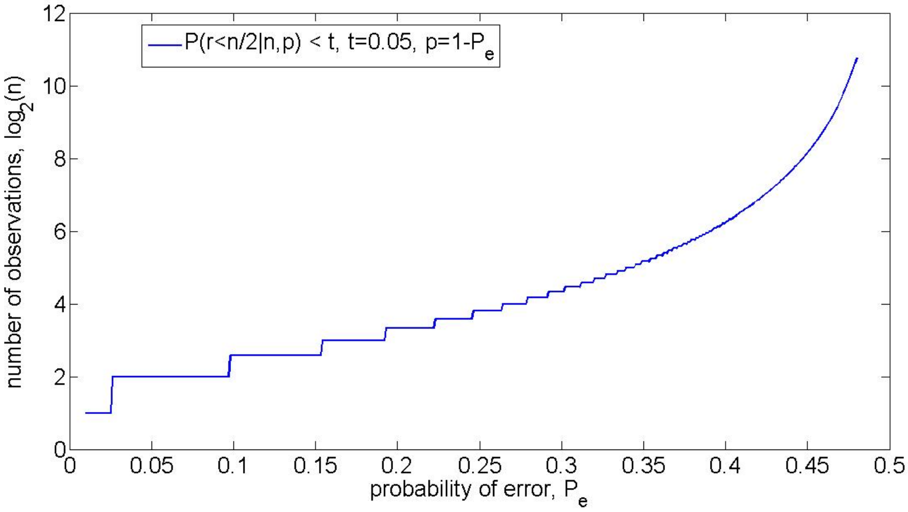

Figure B.1: The smallest number of observations, $n$, as a function of the probability of error $P_e$ that satisfy: $P[r < n/2|n, p] < t$ for $t = 0, 05$.

Majority voting applied on the experimental data confirms the relationship in figure B.1. As an example of the improvement, consider classifying two instructions, both with HW=4. Based on a single observation the probability of error is, $P_e = 0.29$ [2]. Using majority voting with 20 observations reduces the probability of error to, $P_{e,majority} = 0.01$. Finally a comparison with classifying the mean of $n$ observation indicate that majority voting achieves the same error rate with significantly less observations (less than half in some cases). This is crucial when a limited number of observations are available. Quantification of this advantage is subject to future work.

## B.1 Bibliography

[1] DUDA, R., HART, P., AND STORK, D. *Pattern Classification*. John Wiley and Sons, Inc, 2001. 8, 26, 28, 29, 30, 63, 64, 65, 75, 78, 90, 91, 94, 124, 137, 141

[2] DYRKOLBOTN, G. O., AND SNEKKENES, E. Modified template attack: Detecting address bus signals of equal hamming weight. In *Annual Norwegian Information Security Conference - NISK* (2009), Tapir akademisk forlag, pp. 43–56. 39, 40, 41, 42, 46, 47, 99, 109, 113, 122, 124, 127, 135, 137, 142

[3] PEEBLES JR, P. Z. *Probability, Random Variables, and Random Signal Principles*. McGraw-Hill Inc., 1993. 15, 141

# Nomenclature

| | |
|---|---|
| A/D | Analogue-to-Digital |
| ALU | Arithmetic Logical Unit |
| ASIC | Application-Specific Integrated Circuit |
| ASK | Amplitude Shift Keying |
| CMOS | Complementary Metal-Oxide Semiconductor |
| CPU | Central Processing Unit |
| CU | Control Unit |
| DFT | Discrete Fourier Transform |
| DOM | Difference Of Mean |
| DPA | Differential Power Analysis |
| E-field | Electric field |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| EMA | Electromagnetic Analysis |
| EMR | Electromagnetic Radiation |
| FFT | Fast Fourier Transform |
| FPGA | Field-Programmable Gate Array |
| GFS | Greedy Forward Selection |
| H-field | Magnetic field |
| HD | Hamming Distance |
| HW | Hamming Weight |
| IC | Integrated Circuit |
| LDA | Linear Discriminant Analysis |
| LDP | Layout Dependent Phenomena |
| LTI | Linear Time-Invariant |
| MIMO | Multiple-Input and Multiple-Output |
| MLE | Maximum Likelihood Estimates |
| MOSFET | Metal-Oxide-Semiconductor Field-Effect Transistor |

| | |
|---|---|
| NISlab | Norwegian Information Security laboratory |
| NOP | NO Operation |
| NRZ | Non-Return to Zero |
| PCA | Principal Component Analysis |
| PDS | Power Density Spectrum |
| Q cycles | Quadrature clock cycles |
| RAM | Random Access Memory |
| RBW | Resolution BandWidth |
| ROM | Read-Only Memory |
| RZ | Return to Zero |
| SPA | Simple Power Analysis |
| SVC | Single Variable Classifier |
| USRP | Universal Software Radio Peripheral |
| WCCA | Wireless Covert Channel Attack |

# Index