IST-2002-507932

ECRYPT

European Network of Excellence in Cryptology

Network of Excellence

Information Society Technologies

D.VAM.6
Open Problems in Implementation and Application

Due date of deliverable: 31. July 2005
Actual submission date: 13. March 2006

Start date of project: 1. February 2004 Duration: 4.5 years

Lead contractor: Ruhr-University Bochum (RUB)

Revision 1.4

Project co-funded by the European Commission within the 6th Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	

Open Problems in Implementation and Application

Editor
Tanja Lange (DTU)

Contributors
Roberto Avanzi (RUB), Lejla Batina (KUL), Gerhard Frey (IEM),
Pierrick Gaudry (INRIA), Marc Joye (G+), Kerstin Lemke (RUB),
Elisabeth Oswald (IAIK), Christof Paar (RUB), Dan Page (BRIS),
Christine Priplata (EDIZONE) Nigel Smart (BRIS),
Colin Stahlke (EDIZONE), Ingrid Verbauwhede (KUL)

13. March 2006
Revision 1.4

The work described in this report has in part been supported by the Commission of the European Communities through the IST program under contract IST-2002-507932. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Contents

Executive summary

The VAMPIRE lab deals with implementations of cryptosystems in soft- and hardware. In this document we list open problems in this area.

The main objective for this deliverable was to find out whether VAMPIRE should open a new targeted working group. Our answer is that there are many open problems in this field but that they all fit well under one of the targeted working groups VAM1 (Software Implementations), VAM2 (Hardware Implementations), or VAM3 (Side-channel Attacks). So our recommendation is not to open a further working group. Going even further, we conclude that the three targeted working groups will be able to absorb open problems emerging in the future.

The problems we consider in this deliverable are classified according to the type of implementation platform, i.e. hardware or software, and whether side-channel attack resistance is considered. Since research is proceeding some of the more precise open problems we identified at the beginning of the project have been solved in the meantime or found at least partial solutions, so this document contains not only open problems but also problems on which we expect progress soon. For the solved problems we refer to the list of publications in the VAMPIRE report.

Chapter 1

Introduction

When a cryptosystem or protocol is implemented in hard- or software new problems arise during this realization. These might come from the fact that the device is restricted and thus some operations cannot be executed or are very inefficient; or from the desire to get the fastest implementation; or from the thread posed by side-channel attacks.

The Strategic Roadmap project STORK – Strategic Roadmap for Crypto (IST-2002-38273) issued a document on "Open Problems in Cryptology" [38] in which two sections are concerned with topics of the VAMPIRE lab (Virtual Applications and Implementations Research Lab). These problems have influenced our choice of working groups for VAMPIRE and we tried to tackle them over the past 24 months – most of them still remain open because they are pointing to general open areas rather than to specific and well defined tasks.

One of the objectives of the VAM4 working group on "Strategic Research" was to investigate whether new areas in implementations emerge that do not fit under the targeted working groups VAM1 "Software Implementation",VAM2 "Hardware Implementations" and VAM3 "Side-Channel Attacks and Security" and if so suggest the forming of a new targeted working group. Summing up the work of the past two years we can state that we could identify many open problems but all of them fit in at least one of the targeted working groups. As a consequence, this document not only suggests not to open a further targeted working group but even to close the working group on strategic research VAM4.

Not that there are not enough questions – there are many open problems but already in the past project months it was in the the targeted working groups that they were identified and then reported to VAM4 and it seems more natural to *state and solve* (if possible) problems where they occur.

Accordingly, in this deliverable we group the open problems into the categories software, hardware and side-channel attacks. We would like to stress that this separation has its pitfalls – the areas are strongly related and it might only be the try to obtain a more efficient implementation that renders the system prone to side-channel attacks or the other way round one may ask for the fastest implementation that is shielded against side-channel attacks – but we tried to keep the document slim and thus tried to place the problems in the most natural category.

We describe the big open problems in each category and then specify to smaller tasks that look solvable in the closer future – if possible.

For a project that is running since two years it goes without saying that some of the research problems we identified at the beginning have been solved in the meantime. The

publications in the area of VAMPIRE in the ECRYPT list of publications are examples of open problems that found a solution by the participants, sometimes in collaboration with visitors and in other cases groups outside the network were successful faster. For previously open and now solved problems we refer to the scientific publications.

There is also a huge overlap between the work of STVL and AZTEC on the one side and VAMPIRE on the other. Within the past month, STVL3 has produced an 88 pages long document on "Ongoing Research Areas in Symmetric Cryptography" [14] which also considers implementation issues like lightweight cryptography and side-channel attack resistant implementations.

Chapter 2

Open Problems in Software Implementation

In this chapter we consider two types of problems – the "real-life" implementation issues concerning portability, speed differences due to the use of low level programming (use of assembler language), and general crypto-aware programming (related to attacks on implementations) and on the other side more algorithmic problems.

We observe that the academic community is mainly concerned with the second type of problems – at least from what is visible in conference proceedings. However, when it comes to industry collaborations or real life implementations, the first type of problems usually consumes most time – and is rarely shared or reused.

2.1 Open problems in software engineering

In real life people are interested in portability, maintainability and ease of use by a non-expert as well as pure performance issues. A lot of academic crypto implementations are only interested in speed thus their merit for real-life is very limited. Very few have even heard about basic principles of software engineering.

A classic example is the case of MAC algorithm implementation. It is known that one needs to compute most MAC's in one go (that is why there are postprocessing phases on CBC-MAC). However, for software purposes (and some hardware tokens) the API usually works in the following way

1. MAC.Init
2. MAC.Update
3. MAC.Finalize

There are good programming reasons why this is done. However, they are insecure if used incorrectly, or exposed to the attacker. Solving this problem is a software engineering one which involves a little crypto knowledge, not a crypto problem which involves a bit of programming knowledge

For the crypto community this leads to the following problems and approaches.

Automatization One needs a way of automating a lot of the silly stuff which people are doing. Two new approaches here are either the approach by Bernstein on his new language *qhasm* to produce very fast code from simple mathematical descriptions [6], or the approach by Page et al. [4, 5, 31] which works at a higher level and aims to spot possible security weaknesses and optimizations at a higher level in the code. In summary - Bernstein aims to remove the need for a low level programmer and Page aims to remove the need for a crypto aware programmer. Together they both aim to reduce the total cost of code production.

Protocol and primitive aware implementations Much implementation work, and this is particularly true of pairing based systems, is done devoid from the people working on protocols. The protocol people also work without talking to the implementation people. This causes errors to be made, or blind alleys to be chased.

2.2 Open problems on algorithms

Most cryptosystems and ciphers are formulated in an algebraic or algorithmic way and if the generation of instances is an unsolved problem then this is more a topic for the STVL and AZTEC labs than for VAMPIRE. The questions we are facing in software implementations are usually how to implement a certain task most efficiently, where some initial way of implementing it is immediate from the question.

Clearly, we cannot come up with a proof in the mathematical sense that there is no faster way. For some smaller problems an exhaustive search might be possible but in general all we can get are implementations optimal to the best of our knowledge. The deliverable D.VAM.1 "Performance Benchmarks" [15] which was produced within the working group VAM1 on software implementation discusses this question on a theoretical level by listing operation counts for various public key cryptosystems. The results give good examples of formulas we assume to be optimal – but often enough the deliverable lists multiple solutions since different platforms favor different implementations. Only timings of actual implementations can give an answer to this question. This motivated us to even extend deliverable D.VAM.7 "Performance Benchmarks - Public Key Cryptography I" [18] to an open call for contributions of software implementations of public key cryptosystems and we will time the submissions on different machines to obtain comparisons.

A second example along these lines is eSTREAM the ECRYPT Stream Cipher Project [13]. Some comparison is possible based on operation count but only implementations can give advice which system to choose in which situation. In the context of stream ciphers it was not clear how to build good stream ciphers at all and so the the problem was not considered in VAMPIRE but STVL is running the eSTREAM project. Now that the systems are submitted, the main tasks are cryptanalysis and fast implementation. Clearly, the submitters also had the speed in mind when designing the system.

Naturally, the tasks are not restricted to implementations of cryptosystems. If an attack can actually be carried out then its optimized implementation is clearly also an appropriate problem in software implementations.

So the big open problem in this area is:

How to obtain the fastest implementation of a certain (rather specified) task on a certain platform and how fast is it?

In the following we give some examples of algorithmic problems in software implementation.

Computation of Gröbner bases Gröbner bases appear in many parts of cryptography, mainly cryptanalysis, e.g. in recent attacks on AES or stream ciphers as well as in the cryptanalysis of multivariate systems or polynomial based asymmetric systems like Poly Cracker. Buchberger's algorithm used to be the standard algorithm for computing Gröbner bases but the F4, F5 and F5/2 algorithms by Faugere provided improvements that e.g. broke HFE challenges (for an overview with links to the papers we refer to [37]). Any improvement on the speed of implementation has direct consequences for the security of some cryptosystems.

Currently a whole conference series is devoted to Gröbner bases (`http://www.ricam.oeaw.ac.at/srs/groeb/`) and we consider their study to be a very important open problem.

Efficient implementation of LLL Algorithms to find shortest vectors in lattices are often used in attacks on cryptosystems. Any advance in speed has influence on the choice of parameters of the cryptosystems, e.g. if the shortest vector problem can be solved faster then the parameter sizes of NTRU need to be adjusted.

Parallelized operations on large sparse matrices This problem is of interest for cryptanalysis. In e.g. index calculus attacks or after sieving in factorization one has to solve large sparse matrices. It would be interesting to obtain parallelized algorithms solving this task faster than on a single machine.

Inversion-free halving on elliptic curves This problem belongs to public key cryptography based on elliptic curves over binary fields. To compute scalar multiples of a point it was suggested in [24, 35] to use a half-and-add approach instead of the usual double-and-add method. The halving operation itself does not need any inversion provided that the input is given as (x, λ), where x is the x-coordinate of the point and λ is the slope of doubling this point. To get λ from an affine or projective point one inversion is needed; in return, the last halving step does not need an inversion when the output is represented as an affine point (x, y). So this approach is currently interesting for systems in which affine coordinates are used and where half-traces can be computed efficiently. The latter is an operation arising in solving quadratic equations. The question is now, whether is it possible to avoid the inversion and obtain halving formulas that could be faster than doubling, e.g. in fields represented with a normal basis. This question boils down to the problem of solving quadratic equation $T^2 + aT + b$ for $a \neq 0$ and for which one knows that 2 solutions exist without using an inversion. All tries so far introduced an enormous amount of squarings in return for the inversion so that the resulting algorithms were not at all competitive.

More efficient halvings on genus 2 curves This problem is similar to the previous one. However for binary genus 2 curves it is still the case that the halving formulas are less efficient than affine doublings, so that the question is whether one can find faster formulas. The current state of the art is [23].

Inversion-free arithmetic on special binary genus 2 curves This problem belongs to public key cryptography based on hyperelliptic curves. In [27] a special family of binary genus 2 curves was identified on which doublings are particularly fast. Even though in binary fields $\mathbb{F}_{2^n}$ the I/M-ratio is comparably small on a PC implementation, we are interested in inversion-free formulas for e.g. lightweight cryptography where an inversion very slow e.g. it is performed by computing the $2^n - 2$-th power. In an FPGA implementation one will usually save the space needed for the inverter and thus would have such very slow inversions.

Some formulae were given in [28] but current research by Lange and Birkner goes into improvements and comparisons by implementation.

Improvements to double-base expansions of integers and algebraic integers for scalar multiplication Double base expansions, i.e. recodings of integers as linear combinations of products of powers of more than one base, have been already analyzed by Knuth. Usually the bases 2 and 3 can be used, and a straightforward modification of a double- -and-add scheme with two nested loops is used. However the doubly nested loop is often inefficient.

In [12] it is shown how to circumvent the doubly nested loop obstruction by imposing that the sequence of exponents for the two bases in the loops be both monotonous. In a different direction in [36] an analysis is shown with signed coefficients.

The problem is to provide more effectively useable double-base expansions, possibly with applications to pairing schemes. R. Avanzi and F. Sica have a preprint [3] (uploaded to the ECRYPT Vampire document servers) that described an essentially sublinear scalar multiplication method (in the number of additions) requiring no precomputations. This is part of work in progress, which includes the use of bases which correspond to "free" operations, such as the Frobenius endomorphism on subfield curves.

Montgomery ladder for Kummer surfaces Following ideas of Gaudry [21], the implementation of scalar multiplication on Kummer surfaces arising as quotient of the Jacobian of a hyperelliptic curve of genus 2 with two-torsion points in odd characteristic is much faster than the usual addition.

It is an open problem to extend this approach to even characteristic, or to skip conditions about two-torsion points – the formulas require the full 2-torsion group to be rational.

Since for some protocols it is necessary to work on the Jacobian as well and since there the operations are more expensive it would be interesting to rewrite protocols such that the burden to work on the Jacobian is distributed (trivial for key exchange).

Faster point-counting of genus 2 curves in large characteristic There are two alternatives for finding good curves for cryptographic use: either use curves with special structure (curves with Complex Multiplication, Koblitz curves) for which the number of points can be deduced more or less easily, or take random curves and count the points. For genus 2, the large characteristic case is the last case for which point counting is not easily feasible for cryptographic sizes. There exists a polynomial-time algorithm (Pila's generalization of Schoof's algorithm [32]), and a first milestone has been reached in [20] where a prime order group of 164 bits was found by point counting. However, the computation time was large (one week per tested curve) and this implementation will not allow the construction of genus

2 curve cryptosystems with a security of 2^{100} or higher. For these securities, the CM method is still the only alternative.

The problem is to find new ideas that would yield a practical speed-up in the implementation.

Rationality degree Compute the cover of a curve with minimal degree over the projective line; this problem is of relevance for index-calculus attacks since they do in fact depend on the degree of the model rather than solely on the genus of the curve.

Embedding degree Give a more systematic method to construct elliptic curves which are pairings-friendly. Solve the corresponding problem for genus two, and, most important, for genus 3 – which could lead to faster implementations.

Till now this is only done for very special cases, for which the complex multiplication with roots of unity is immediate. For genus 2 on has some curves of the form: $Y^2 = X^5 + 1$ with low embedding degree and for genus 3 one could do the same study and experiments for the corresponding curve with 7-th roots of unity.

Factoring Polynomials of low degree over finite fields Index calculus methods involve determining whether a polynomial of relatively low degree over a finite field splits into linear factors and, if that's the case, to find these factors. This is crucial for writing the group elements found during a random walk as sums of points, and for the later testing whether the original group element (divisor) is smooth.

The problem here is to find methods for polynomial factorization that in the context of fixed, small degree polynomials defined over binary fields of increasing size work faster than the currently known algorithms by Cantor-Zassenhaus, Berlekamp, Shoup and other authors.

Current work in progress by R. Avanzi and N. Theriault goes in this direction.

Improvements to the arithmetic of Koblitz Curves Koblitz curves are interesting because of their exceptional performance. In [1] it is shown how to further reduce the complexity of Koblitz curve scalar multiplication by inserting point doublings in the Frobenius-and-add loop. This method was originally applied to the context where no precomputations are allowed. An improvement has been presented in [2].

The open questions are: allowing larger digit sets and making better use of the halving. On the second problem there is work in progress by the authors of the latter paper.

Public key cryptography for heavily constrained environments Two emerging examples of PKC applications are radio frequency identification tags (RFIDs) and sensor networks. They put new requirements on implementations of PK algorithms with very tight constraints in number of gates, power, bandwidth etc. The arithmetic for PK cryptosystems should be revisited and further optimized especially in the case of hardware implementations. Hardware implementation allow for an extra degree of freedom with respect to parallelism of various processes which can lead to a more compact as well as more efficient solution.

The first goal is to provide suitable embedded arithmetic to support the security in resource constrained devices. Embedded devices have limited energy supply, are battery operated, run on small embedded processors and have a limited amount of storage. What is needed for embedded systems is a systematic computational security-energy-memory trade-off. The

most challenging tasks are implementations of PKC. A promising example is the curve-based cryptography, more precisely Elliptic/Hyperelliptic Curve Cryptography (ECC and HECC).

Chapter 3

Open Problems in Hardware Implementation

In Hardware Implementations the questions can be very similar to those in the previous section – one is given a task and a target platform and should find the most efficient way of implementing it. The question gets essentially different if the target platform can be chosen – then the possibilities are usually limited by the budget but one may have the question how expensive is cryptography – or how much space does one need to reserve to secure communications with a device at a reasonable speed.

The big general problems can be phrased as:

- Develop performance-cost optimized implementation methods, i. e., designing chips and/or software techniques which which will achieve speed goal XY at the lowest possible area, using the least amount of memory (very relevant in applications involving embedded systems), or minimizing the chip's power dissipation.
- Develop optimized processor architectures for cryptography by customizing both the processor's instruction set and its micro-architecture.

Exiting extreme cases include the design of very lightweight cryptography for RFID tokens. The VAM2 working group is organizing a second workshop on this topic in July 2006. The contributions to the first workshop are available at `www.iaik.tu-graz.ac.at/research/krypto/events/index.php` and give good overview of the state of the art. So here the area and money are extremely restricted while the speed should be optimized.

On the other hand interesting research is going on in special purpose hardware for attacks. Questions are how expensive or how feasible is it to attack RSA with 1024 bits within a year. The SHARCS conferences `www.sharcs.org` deal with this type of question for both symmetric and asymmetric systems. Answers include the design of novel architectures or the clever use of massive computing. In some cases these machines are even physically built like in the case of the DES cracker and then demonstrate the weakness of a system; in most cases the construction costs are just to high. But very often the estimates end up being lower than one would expect knowing only the estimates for attacks by software implementations.

3.1 Open problems for light-weight cryptography

Low-power and low die-size implementations of public-key algorithms that are suitable for low-cost RFID tags State of the art low-cost RFID tags come, in the best cases, with custom made and therefore not standardized cryptographic algorithms and protocols. Due to the lack of review from the cryptographic community, these solutions are prone to be insecure. Embarrassing cryptanalytic attacks, such as shone by John's Hopkins University on RFID tags produced by Texas Instruments, are the consequence of this lack of using standardized algorithms. So far, only one implementation of the AES algorithm is available, that fits on a low cost RFID tag. However, symmetric key cryptography is mostly suited for closed systems, where key management is not a big problem. If low-cost RFID systems of different vendors should become interoperable, public-key systems are much better suited. Therefore, research into low-power and low die-size implementations of public-key algorithms are an important open research problem.

Architectures for embedded security Implementing public-key cryptography on platforms with limited resources, such as microprocessors, is a challenging task. Hardware/software co-design is often the only answer to implement the computationally intensive operations with limited memory and power at an acceptable speed. Good candidates for emerging applications of PKC in embedded devices appear to be Elliptic and Hyperelliptic curve cryptosystems. But all those requirements can only be achieved with the synergy of hardware and software. In all these examples we encounter strong demands for software-hardware co-design. It is a challenge to develop efficient, low-cost PKC implementations to achieve the best partition in a software-hardware co-design. Such an implementation strategy combines the best of both worlds, as a security protocol stays in the software and the computationally intensive tasks are moved to hardware. Such an investigation is of special interest as embedded devices are of vital importance for a broad area of pervasive computing such as sensor networks and wireless applications. Hardware/software co-design offers a new alternative for low-power and low-footprint devices. Therefore, one should explore other trade-offs between hardware and software in order to find the best partition. Additional architectural options can be made available by exploiting parallelism between ECC and HECC operations in the case of curve-based cryptography. Also the sequences of operations for PKC usually have to be rewritten in order to match the given architecture in the best way. Hence, these investigations should be done inseparably from the improvements that are related to the arithmetic.

Better utilization of vector and SIMD parallel processors These things exist more and more commonly but are often ignored in terms of performance evaluation. It would be nice to see an emphasis on developing specialized algorithms for arithmetic on these platforms.

Low-power power-analysis resistant designs With the advent of contactless technologies such as contactless smart cards or RFID tags, the issue of side-channel resistance has emerged also in these fields. However, many countermeasures that are popular and efficient have the drawback that they significantly increase the number of operations that have to be computed and therefore often both increase the size of an implementation and the power consumption of a device. This goes contrary to the needs of contactless technology. Consequently, looking into hardware implementations that are both small and secure is an open and challenging problem.

Secure instruction set extensions for cryptographic algorithms The subject of instruction set extensions for public and private key cryptography has received increasing interest over the past two years. Most research was however devoted to looking only at instruction set extensions that aim for speedup. Yet, security issues arise more and more for all kinds of embedded devices. Consequently, the issue of implementation attacks should be taken into account when an instruction set extension is being considered.

3.2 Open problems in special purpose hardware

The topic of special purpose hardware is studied in length in D.VAM.3 [16]. Here we state the most important problems in this area.

Alternative technologies Can analog technologies, e.g., analog micro electronics or optical systems, be applied to cryptanalysis? The twinkle design is based on such methods but the practicality is questionable. Can other designs make use of such approaches?

In general one needs to investigate, how realistic the technology assumptions made for the (few) existing factoring machines are in reality.

Special purpose designs for number field sieve and index calculus The first and second SHARCS workshop received submission to these topics and it is interesting to study designs which are optimal in the full-cost model. Often enough the algorithms are changed or at least the parameters chosen differently for the optimal cut-off.

For integer factorization (e.g. 1024-bit numbers) a good choice seems to be GNFS (General Number Field Sieve) supported by ECM (Elliptic Curve Method). It is still unclear how to choose optimal parameters with regard to the full-cost model, most of all how to balance the efforts between the sieving and the matrix step. Moreover it is important to study how much cofactorization should be done within the sieving step (e.g. balance between lattice sieving and ECM and choice of suitable parameters).

Special purpose designs for LLL Even if an algorithmic breakthrough is hard to obtain for LLL it might be possible to design special purpose hardware for LLL. So far only sieving and matrix operations have been considered.

Chapter 4

Open Problems in Side-Channel and Fault Attacks

This chapter deals with attacks on the implementations. The most prominent example are side-channel attacks are attacks which take advantage of measurements of the power consumption of the device, its timing behavior, and the electromagnetic radiation emitted by the device to discover secret and sensitive information such as secret keys. Closely related are fault attacks in which an active attacker induces a fault and then studies the reaction of the system. These attacks are a serious thread particularly for implementations on small devices but recent attacks based on cache misses or hyperthreading concern PCs.

One big but also very hard open problem is to find a theoretical model of side-channel attacks. Such a model, for example, would allow implementations to be provably secure given a set of assumptions, similarly to techniques used in the context of provable security of public-key schemes. Some initial work was carried out in the VAM3 working group; however, the basic assumptions are still very controversial.

It is also interesting to study the validity of such assumptions – some systems attain side-channel under very strong assumptions like that a squaring in a finite field is indistinguishable from a multiplication. The answer to the question whether this holds or not definitely depends on the library used for the field operations but even if that takes the same time it might be possible to notice that the inputs were identical. So a physical validation of assumptions is necessary.

4.1 Open problems in side-channel attacks and modelling

A very broad open problem in this area is to find a theoretical model for side-channel attacks. The ECRYPT deliverable D.VAM.12 [19] is concerned with this long term research project.

DPA resistance on circuits level Security is as strong as the weakest link, therefore protecting cryptographic systems should be done on all levels of abstraction, which are depicted in the security pyramid in Figure 4.1. Each abstraction layer represents specific modeling, design and implementation issues that must be covered for secure system operation [34].

Many countermeasures have been put forward, though no single solution or combination of solutions has been proven to be effective or practicable against all SCA. There is a lot of ongoing work done on a design methodology to combat DPA and a wide class of SCA's.

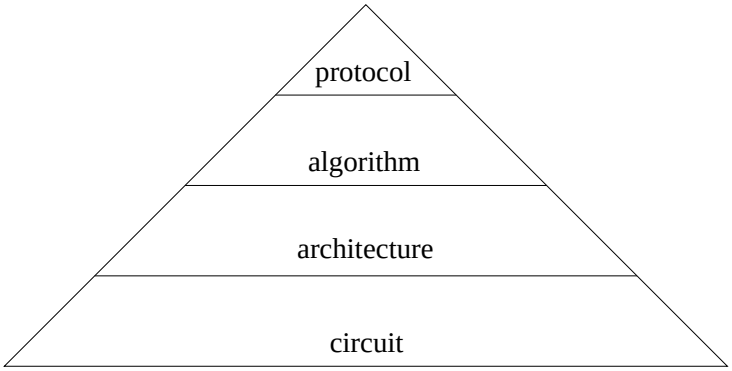

Figure 4.1: Security pyramid showing the different levels of abstraction.

It is known that the power consumption of a circuit depends on the output transitions of the gates. If these transitions depend on secret information, the security of the implementation can be compromised. Hence, to make a circuit resistant against power analysis attacks, the power consumption should be independent of the secret information. The circuit-level approaches to achieve this can be divided into two categories: custom logic styles and standard logic styles. Custom logic styles are only applicable to custom ASIC implementations. Standard logic styles combine standard cells from existing libraries into new standard cells. Hence, they can be used for FPGA implementations as well as standard cell ASIC implementations. Tiri *et al.* developed SABL, which is a custom logic style, and WDDL [40], which is a logic style consisting of standard cells. Some other solutions for special logic styles hindering DPA followed, such as [33]. Some types of side-channel attacks such as first-order DPA are already proven to be infeasible for the special logic styles. However, it remains to see how these special logic styles would resist some more advanced attacks, such as higher-order attacks, template attacks etc.

Side-channel attacks based on EMA The research on EMA is still an area with many open questions including the best shape of antennas, the amount of information available, attacks from a large distance etc. As EMA graphs can be considered in time-domain or frequency-domain, which includes techniques of Fourier analysis, there is many more aspects to explore than for power analysis. Some previous results show that these graphs contain more information than power graphs but useful data are more difficult to collect. Moreover, it is not easy to find the optimum frequencies for the measurements.

Higher Order Differential Power Analysis In a higher-order DPA attack, the attacker formulates an attack based on the joint statistical properties of multiple aspects of the signal. One simple example of a higher-order DPA attack is when an attacker collects signals from two sources. These attacks are the next threat once countermeasures to first-order DPA attacks are developed.

If masking or secret splitting schemes are applied, this leads to a reduction of information available at side channels: an adversary has to consider joint probability densities of intermediate states of an algorithm instead of directly predicting results. The security gain in terms of reduced leakage in the presence of masking should be elaborated in more detail.

Relationship between hardware and software countermeasures To prevent side channel based attacks both hardware based countermeasures and software (basically algorithmic) countermeasures have been proposed. Hardware countermeasure include random processor interrupts [10] or special logic styles such as SABL [39]. Algorithmic countermeasures make use of masking or secret splitting schemes. Their relative effectiveness, also in terms of costs such as area, throughput, power and energy of a hardware-software co-design, is worth to be studied.

4.2 Algorithmic problems related to side-channel attacks

Design-time evaluation of vulnerability For the time being cryptosystems are proposed without considering their side-channel attack resistance. Once they are to be implemented fixes to render them SCA secure are applied which often reduce the speed significantly – in the worst case to the extend that the previously identified merits of the new systems are outweigh.

Design-time evaluation of side-channel vulnerability in software and hardware seems a challenge. Clearly, having a formal formal models for side-channels would significantly ease the study which makes this problem related to the the problem of modelling considered in the previous section.

Design of light-weight systems resistant to SCA Smart cards are known to be particularly vulnerable to these type of attacks, but nevertheless side-channel analysis is also a concern in the area of light-weight implementations such as RFIDs and sensor networks. This means that previously known countermeasures should be revisited for this purpose. Hence, side-channel resistance is also an issue, but in this case at a very low cost.

Uniform group operations on hyperelliptic curves This problem is a rather theoretical one but its motivation comes from side-channel attacks. For elliptic curves [9] change the description of the group operation on elliptic curves in such a manner that additions and doublings can be carried out with the same formulas. It is theoretically impossible to describe the group operation with only one set of equations [25] but the advantage of the presented formulas is that the special case does not occur regularly like the doubling does in the double-and-add method. In addition they show how to handle the special case.

For hyperelliptic curves it would be very interesting to have such formulae as well but we are not aware of any success in this direction.

4.3 Open problems in masking

There is ongoing research in developing masking schemes to protect cryptographic implementations from first-order side channel attacks. Masking leads to a randomization of intermediate states of an algorithm so that intermediate states can no longer be predicted by an adversary. If this predictability is lost, first order differential side channel attacks are likely to fail. However, higher order differential attacks can still be applied by considering joint probability distributions at two or multiple intermediate results, e.g., [30].

Masking AES with Galois Field Arithmetic in Circuitry At CHES 2005, it was demonstrated by Mangard, Pramstaller and Oswald [29] that circuit implementations of AES masking schemes can be successfully attacked by using first-order techniques. This was feasible by considering glitching activities at the logic gates of the circuit that can be predicted from simulations. It is an open question whether an effective and compact AES masking scheme can be designed at the algorithmic level in hardware, depart from randomizing an S-Box table look-up.

Converting from Arithmetic to Boolean masking Masking cryptographic algorithms that combine boolean and arithmetic operations such as hash function based MAC constructions in an efficient way is still a hard problem. While switching from boolean masking to arithmetic masking is fast thanks to the conversion algorithm proposed by Goubin [22] masking the opposite direction is very costly. [22] proposed a method that requires $5k + 5$ operations for a k-bit register. By using precomputed tables Coron and Tchulkine [11] presented a conversion algorithm that is more efficient for higher values of k, e.g. it was shown that 32-bit conversions on a 8-bit microprocessor can be implemented with 140 operations whereas Goubin's method requires 660 operations per conversion. For practical purposes these numbers are still not acceptable.

4.4 Open problems in fault analysis

Whereas side channel attacks act in a passive way, fault analysis attacks aim at inducing an erroneous behavior of the implementation that can result in a vulnerability of a cryptographic implementation or even a total break. We expect that fault analysis continues to be of at least the same importance than side channel based attacks, especially if cryptographic devices are handed over to an end-user who might be a potential adversary.

Modelling Tampering Attacks Generally (e.g., [7]), fault attacks assume that a transient fault can be invoked within the implementation at a certain intermediate state of the algorithm. For practical purposes it is an open question how to determine the probability that a precise transient fault can be successfully injected. A first approach towards understanding the manifold nature of faults was undertaken in [17].

Counteracting Multi-Fault Attacks Until now, fault analysis has concentrated on studying single-fault based attacks. Countermeasures proposed for single fault attacks might not be effective against multi-fault attacks. What kind of security can ever be guaranteed if each cell of data memory can be tampered with in a probabilistic sense and if the adversary is able to induce faults at any internal state and computation of the physical device? This is an open research area.

Countermeasures for both SCA and Fault Analysis Cryptographic implementations that are used in the end user environment have strong needs to be secured both against side channel analysis and tampering attacks. Are there any countermeasures suitable to prevent both SCA and Fault Analysis?

Fault attacks on RSA As for side-channel attacks, fault attacks should be taken into consideration when designing hardware or software implementations of cryptosystems. We describe an open problem related to fault attacks and the widely deployed RSA-PSS (now part of the PKCS #1 standard) signature scheme.

Let $N = pq$ denote an RSA modulus. Let (e, d) be the corresponding verification/signing exponents, satisfying $ed \equiv 1 \pmod{\phi(N)}$. An RSA signature on a message m is obtained by applying a padding function μ on m and then by raising the resulting value to the d modulo N: $S = \mu(m)^d \bmod N$. The validity of signature S is verified by checking that $S^e \equiv \mu(m) \pmod{N}$. Padding function μ can be probabilistic.

Several fault attacks have been reported against RSA signatures. We review below some of them.

GCD attack The more devastating attack applies when the signature is computed using the Chinese Remainder Theorem (CRT). In this case, an RSA signature S is evaluated as $S = S_p + p\big(i_p(S_q - S_p) \bmod q\big)$ where $S_p = \mu(m)^{d_p} \bmod p$, $S_q = \mu(m)^{d_q} \bmod q$, $d_p = d \bmod (p-1)$, $d_q = d \bmod (q-1)$, and $i_p = p^{-1} \bmod q$. If the computation of S_p is faulty (but the computation of S_q is not) then, letting S' the resulting (faulty) signature, we have

$$\gcd\big(S'^e - \mu(m) \pmod{N}, N\big) = q \ .$$

An analogous attack applies if the value of i_p is faulty.

Flipping-bit attack This attack assumes that the attacker is able to flip the value of a bit in (the representation of) secret exponent d. So an RSA signature is computed using (faulty) exponent $d' = d + 2^k(1 - 2\,d_k)$, where d_k denotes the flipped bit. Hence, from the resulting (faulty) signature, $S' = \mu(m)^{d'} \bmod N$, the flipped bit is recovered by checking if

$$\frac{S'^e}{\mu(m)} \equiv S'^{\pm e\,2^k} \pmod{N}$$

for $k = 0, 1, 2, \ldots$

Sticking-bit attack This attack is like the previous one but assumes a much stronger attacker. The attacker should be now able to *fix* the value of a *chosen* bit. For example, if the value of bit number k (in the representation of) of d is fixed to 0 (we let d'_k the corresponding bit; i.e., $d'_k = 0$) then it is easy to deduce the value of d_k by checking whether or not the resulting signature is valid (in which case $d'_k = d_k$ and so is equal to 0; $d_k = 1$ otherwise).

For RSA-PSS, padding function μ is *probabilistic* and is defined as:

$$\mu(m, r) = 0 \,\|\, w \,\|\, (g_1(w) \oplus r) \,\|\, g_2(w) \quad \text{with } w = h(m\|r)$$

where h, g_1, g_2 are hash functions.

The GCD attack and the flipping-bit attack do not apply against RSA-PSS as the attacker does not know $\mu(m, r)$. Only the sticking-bit attack applies. A problem, still open since the seminal paper by Boneh, DeMillo and Lipton [8], is to mount a successful fault attack (apart the sticking-bit attack) against RSA-PSS.

Bibliography

[1] R. Avanzi, M. Ciet, and F. Sica. Faster scalar multiplication on Koblitz curves combining point halving with the Frobenius endomorphism. In: Public Key Cryptography (PKC 2004). *Lecture Notes in Comput. Sci.* 2947, pages 28–40. Springer-Verlag, 2004.

[2] R. Avanzi, C. Heuberger, and H. Prodinger. Optimality of the τ-NAF for Koblitz Curves and of its Combination with Point Halving. In: Proceedings of SAC 2005, *Lecture Notes in Comput. Sci.* 3897, pages 332–344. Springer Verlag 2006.

[3] R. Avanzi, and F. Sica. Scalar Multiplication on Koblitz Curves using Double Bases. . Submitted.

[4] M. Barbosa, R. Noad, D. Page, and N. P. Smart. First Steps Toward a Cryptography-Aware Language and Compiler. *ePrint Archive, Report 2005/160.*

[5] M. Barbosa and D. Page. On the Automatic Construction of Indistinguishable Operations. *ePrint Archive, Report 2005/174.*

[6] D. J. Bernstein. *Writing high-speed software.* `http://cr.yp.to/qhasm.html`.

[7] E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In *Advances in Cryptology – CRYPTO 1997*, volume 1294 of *Lecture Notes in Comput. Sci.*, pages 513–525. Springer-Verlag, Berlin, 1997.

[8] D. Boneh, R. DeMillo, and R. Lipton. On the importance of checking cryptographic protocols faults. *Advances in Cryptology – Eurocrypt 1997*, volume 1233 of *Lecture Notes in Comput. Sci.*. Springer-Verlag, Berlin, 1997, 37–51.

[9] É. Brier, I. Déchène, and M. Joye. Unified point addition formulæ for elliptic curve cryptosystems. *Embedded Cryptographic Hardware: Methodologies & Architectures.* Nova Science Publishers, 2004.

[10] C. Clavier, J.-S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *Lecture Notes in Comput. Sci.*, pages 252–263. Springer-Verlag, Berlin, 2000.

[11] J.-S. Coron and A. Tchulkine. A New Algorithm for Switching from Arithmetic to Boolean Masking. In *Cryptographic Hardware and Embedded Systems – CHES 2003*, volume 2779 of *Lecture Notes in Comput. Sci.*, pages 89–97. Springer-Verlag, Berlin, 2003.

[12] V. Dimitrov, L. Imbert, and P. Mishra. Efficient and Secure Elliptic Curve Point Multiplication Using Double-Base Chains In: ASIACRYPT 2005, LNCS 3788, pp. 59–78, IACR 2005.

[13] ECRYPT STVL. *eSTREAM - ECRYPT Stream Cipher Project.* `http://www.ecrypt.eu.org/stream`.

[14] ECRYPT STVL, working group STVL3. *Ongoing Research Areas in Symmetric Cryptography.* D.STVL.4 of the ECRYPT project.
available at `http://www.ecrypt.eu.org/documents.html`, 2006.

[15] ECRYPT VAMPIRE, working group VAM1. *Performance Benchmarks.* D.VAM.1 of the ECRYPT project.
available at `http://www.ecrypt.eu.org/documents.html`,2005.

[16] ECRYPT VAMPIRE, working group VAM2. *Hardware Crackers.* D.VAM.3 of the ECRYPT project.
available at `http://www.ecrypt.eu.org/documents.html`,2005.

[17] ECRYPT VAMPIRE, working group VAM3. *Electromagnetic Analysis and Fault Attacks: State of the Art.* D.VAM.4 of the ECRYPT project.
available at `http://www.ecrypt.eu.org/documents.html`,2005.

[18] ECRYPT VAMPIRE, working group VAM1. *Performance Benchmarks - Public Key Cryptography I.* D.VAM.7 of the ECRYPT project.
available at `http://www.ecrypt.eu.org/documents.html`,2006.

[19] ECRYPT VAMPIRE, working group VAM3. *Theoretical Model for Side-Channel Attacks (outline).* D.VAM.12 of the ECRYPT project.
available at `http://www.ecrypt.eu.org/documents.html`,2006.

[20] P. Gaudry and É. Schost. Construction of secure random curves of genus 2 over prime fields. In *Eurocrypt 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 239–256. Springer-Verlag, 2004.

[21] P. Gaudry. Fast genus 2 arithmetic based on Theta functions. *ePrint Archive, Report 2005/314.*

[22] L. Goubin. A Sound Method for Switching between Boolean and Arithmetic Masking. In *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Comput. Sci.*, pages 3–15. Springer-Verlag, Berlin, 2001.

[23] I. Kitamura, M. Katagi, T. Takagi. A Complete Divisor Class Halving Algorithm for Hyperelliptic Curve Cryptosystems of Genus Two. *ACISP*, volume 3574 of *Lecture Notes in Comput. Sci.*, pages 146–157. Springer-Verlag, Berlin, 2005.

[24] E. W. Knudsen. Elliptic Scalar Multiplication Using Point Halving. *Advances in Cryptology – Asiacrypt 1999*, Lecture Notes in Comput. Sci., vol. 1716, Springer-Verlag, Berlin, 1999, 135–149.

[25] H. Lange and W. Ruppert. Complete systems of addition laws on abelian varieties. *Invent. Math.* **79**, 1985, 603–610.

[26] T. Lange. Formulae for arithmetic on genus 2 hyperelliptic curves. *Appl. Algebra Engrg. Comm. Comput.*, 15(5):295–328, 2005.

[27] T. Lange and M. Stevens. Efficient doubling for genus two curves over binary fields. In *Selected Areas in Cryptography – SAC 2004*, volume 3357 of *Lecture Notes in Comput. Sci.*, pages 170–181. Springer-Verlag, Berlin, 2005.

[28] T. Lange. Arithmetic on Binary Genus 2 Curves Suitable for Small Devices. *Workshop on RFID and Lightweight Crypto, Graz 13.-15.07.2005*, 2005.

[29] S. Mangard, N. Pramstaller, and E. Oswald. Successfully Attacking AES Hardware Implementations. In *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *Lecture Notes in Comput. Sci.*, pages 157–171. Springer-Verlag, Berlin, 2005.

[30] T.S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *Lecture Notes in Comput. Sci.*, pages 238–251. Springer-Verlag, Berlin, 2000.

[31] D. Page. *CAO : A Cryptography Aware Language and Compiler.* `http://www.cs.bris.ac.uk/ page/research/cao.html`.

[32] J. Pila. Frobenius maps of abelian varieties and finding roots of unity in finite fields. *Math. Comp.* **55**, 1990, 745–763.

[33] T. Popp and S. Mangard. Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints. In *Proceedings of 7th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 3659 of *Lecture Notes in Comput. Sci.*, pages 172–186. Springer-Verlag, Berlin, 2005.

[34] P. Schaumont and I. Verbauwhede. Domain specific codesign for embedded security. In *IEEE Computer Magazine*, 2003, Vol. 36, nr.4, pages 68–74.

[35] R. Schroeppel. *Elliptic curves: Twice as fast!* Presentation at the Crypto 2000 Rump Session, 2000.

[36] F. Sica and M. Ciet An Analysis of Double Base Number Systems and a Sublinear Scalar Multiplication Algorithm. To appear in: proceedings of Mycrypt 2005, Springer Verlag.

[37] A. Steel. Allan Steel's Gröbner Basis Timings Page. `http://magma.maths.usyd.edu.au/users/allan/gb/`.

[38] STORK WG 3. *Open Problems in Cryptology.* D6 of the STORK Project. available at `http://www.stork.eu.org/documents.html`, 2003.

[39] K. Tiri and I. Verbauwhede. Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology. In *Cryptographic Hardware and Embedded Systems – CHES 2003*, volume 2779 of *Lecture Notes in Comput. Sci.*, pages 125–137. Springer-Verlag, Berlin, 2003.

[40] K. Tiri and I. Verbauwhede. Design Method for Constant Power Consumption of Differential Logic Circuits. In *Proceedings of Design, Automation and Test in Europe Conference (DATE)*, pages 628–633, 2005.