Cryptology in smart cards: implementations, threats and counter-measures. Symmetrical & Asymetrical algs

David Gueguen

March 22, 2017

WARNING: .....Work in progress!

Principles

This is an interactive document: It shall be modified on demand.
Each time, for each suggestion: mistakes, typos, a missing topics, something not clear, etc...

E-mail ✉: davidgueguen@mdl29.net

- If you don't understand something this document is crap!
- Shannon [2001], *Link* : link to pdf via choucroutage.com
- Link: link to wikipedia

Objective

- The only reason for this document is to help in the understanding of an attack/implementation. Tell if useful or not, pleaze, if there are thing missing.
- This document is far away from being enough to understand many modern attacks, for this reason it is mainly focus on implementation.

As feasible as it is possible, was tried to make this document as precise and exact as possible on a mathematical point of view introducing systematically some required mathematical vocabulary.

To do :
* give -anonymous- benchmark
* give a precise estimation of the complexity, numerical examples
* make system of references reliable
* refine the indexes of frequency
* provide practical and theoretical ($\pm$) traces

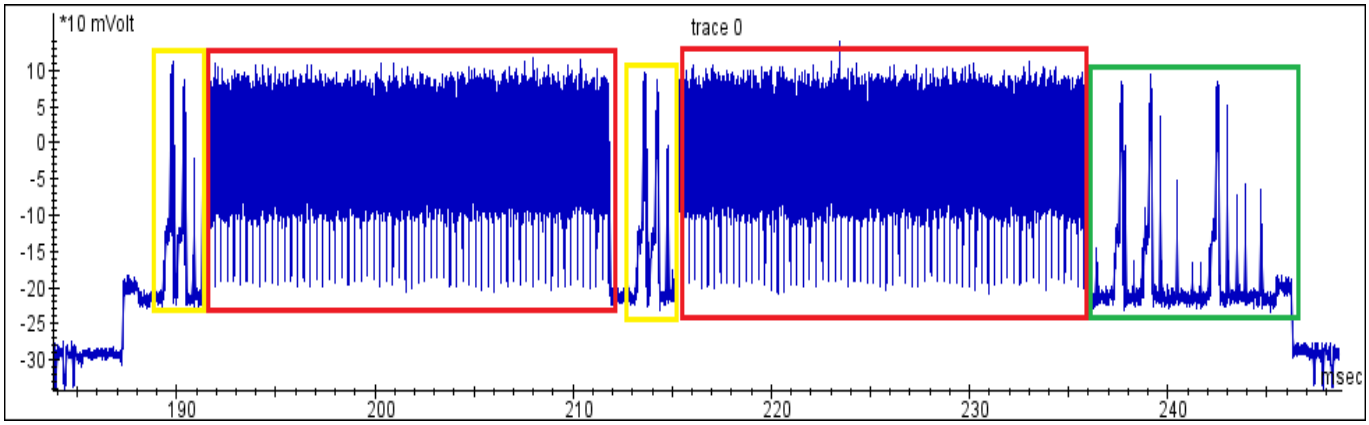
Thanx

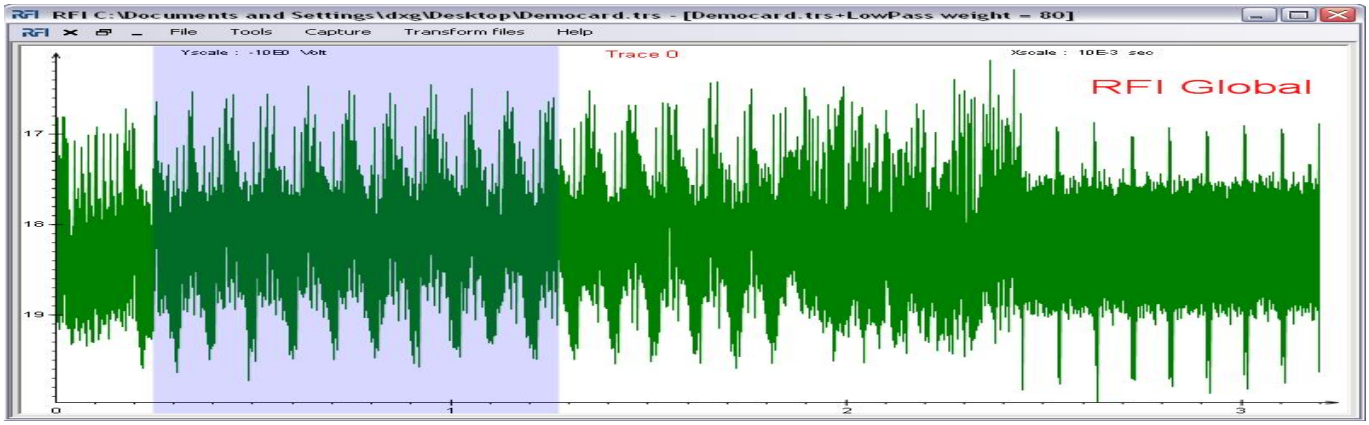

RFI
Global

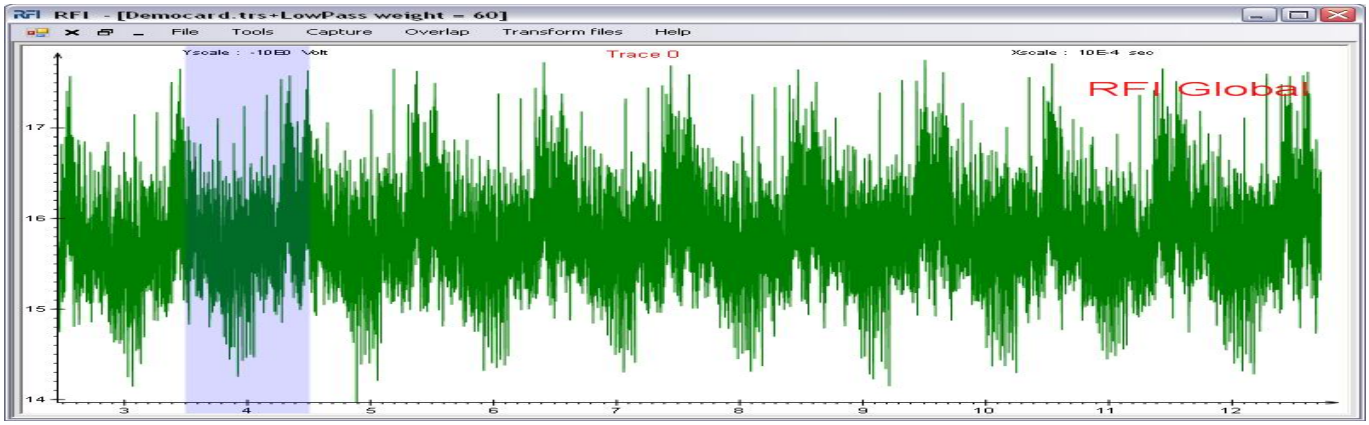

Underwriters
Laboratories

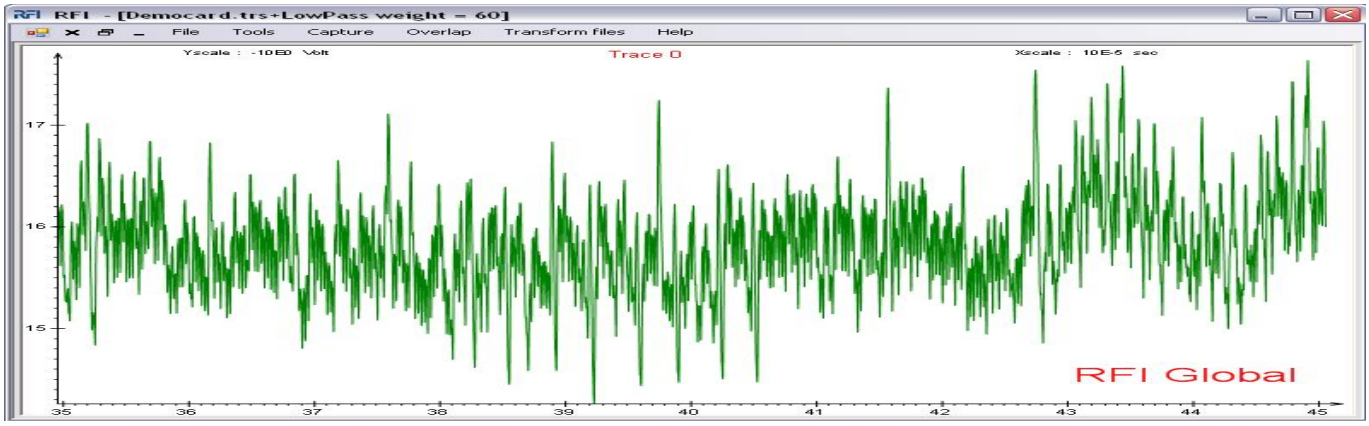

Appluche


Choucpoutage.com


*10 mVolt
trace 0
msec


RFI C:\Documents and Settings\dxg\Desktop\Democard.trs - [Democard.trs+LowPass weight = 80]
File Tools Capture Transform files Help
Trace 0
RFI Global


RFI - [Democard.trs+LowPass weight = 60]
File Tools Capture Overlap Transform files Help
Trace 0
RFI Global


RFI - [Democard.trs+LowPass weight = 60]
File Tools Capture Overlap Transform files Help
Trace 0
RFI Global

Contents

I Symmetric encryption **7**

I .1 Intro . . . 7

I .2 The DES specification . . . 7

I .3 Implement symmetrical algs in smart cards . . . 8

I .3.1 Prevent information leakage . . . 8

I .3.2 The Transforming Masking method . . . 10

I .3.2.1 General principle . . . 10

I .3.2.2 Notation . . . 12

I .3.2.3 A round with a secured DES . . . 15

I .3.2.4 Resistance against the first order DPA . . . 16

I .3.2.5 Practical implementation . . . 16

I .4 Attacks symmetrical algs in smart cards . . . 17

I .4.1 Differential Power Analysis . . . 17

I .4.2 About the power model . . . 21

I .4.3 Formalization . . . 23

I .4.3.1 P.Kocher's DPA mono-bit . . . 23

I .4.3.2 T-S.Messerge's DPA multi-bit . . . 23

I .4.3.3 Correlation Power Analysis . . . 24

I .4.3.4 Partitioning Power Analysis . . . 24

I .4.3.5 PPA generalize all previous DPA attacks . . . 25

I .4.3.6 Maximum of probability . . . 26

I .4.4 Compare those methods . . . 27

I .4.5 High Order DPA attacks . . . 28

I .4.5.1 A particular case: The superposition attack . . . 28

I .4.5.2 General HODPA . . . 30

II Asymmetric encryption **31**

II .1 RSA: description . . . 32

II .1.1 References . . . 32

II .1.2 Modular arithmetic . . . 32

II .1.3 Cryptographic problems . . . 36

II .2 Asymmetrical implementation in smart cards . . . 37

II .2.1 Number representations . . . 37

II .2.1.1 Mathematical notation . . . 38

II .2.1.2 Classical b-arry representations . . . 38

II .2.1.3 Some binary representations . . . 39

II .2.1.4 Basic in geometry of numbers . . . 41

II .2.1.5 Non Adjacent Forms . . . 43

II .2.1.6 Illustration of complex NAF: . . . 51

II .2.2 Group representations . . . 52

II .2.2.1 Chinese Theorem of Remainders: . . . 52

II .2.3 Multiplications . . . 52

II .2.4 Squarring . . . 54

II .2.5 Reduction . . . 55

II .2.6 Exponentiations . . . 59

II .2.6.1 Convention and Names ! . . . 59

II .2.6.2 Two Square & Multiply algorithms and the factor method . . . 60

II .2.6.3 Atomic Square & Multiply . . . 63

II .2.6.4 Scanning digits in base 2^k . . . 64

II .2.6.5 Sliding window algorithms . . . 67

II .2.6.6 The Montgomery Ladder . . . 69

II .2.6.7 Randomized algorithms . . . 70

II .2.6.8 NAF algorithms . . . 71

II .2.6.9 Square free algorithms . . . 74

II .2.7 what exponentiation is about . . . 75

II .2.7.1 Physical threat: side channel . . . 76

II .2.7.2 Physical threat: fault injection . . . 79

II .3 Physical threats and counter measures . . . 84

II .3.1 Physical threat 84
II .3.1.1 Counter measures to side channel 84
II .3.1.2 Counter measures side channel: slowing down the exponentiation . . 87
II .3.1.3 Counter measures to fault injection 88

IIIAbout template attacks **90**

III .1The original template attack 90
III .2Practical improvements 93
III .3Template based DPA attacks: 94
III .3.1Template attack with several traces 94
III .3.2DPA-Template attack 94
III .4Template attacks on symmetrical algorithms 94
III .5Template attacks on asymmetrical algorithms 97
III .6The stochastic attack 97
III .7The power consumption model & notations 99

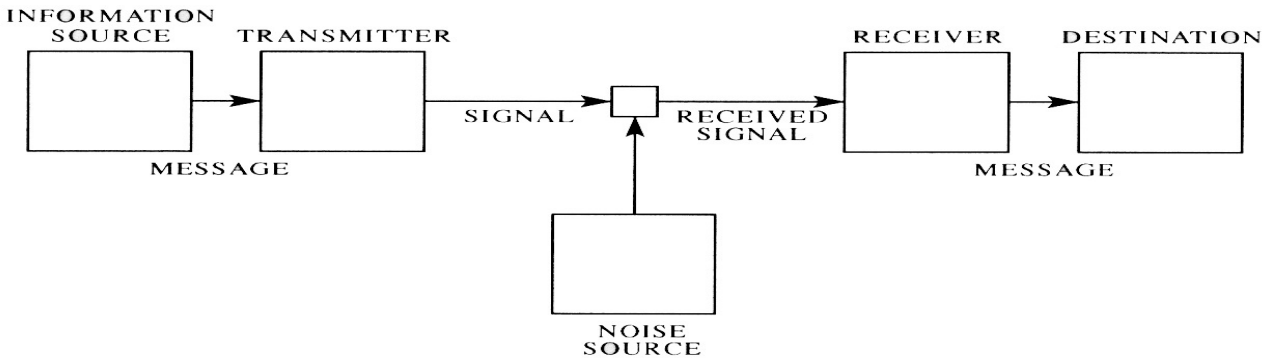
IVLatex definition & tricks **100**

Chapter I

Symmetric encryption

I .1 Intro

Symmetrical cryptography relies on Claude Shannon's 'theory of Information': quantify amount of information inside a theoretical channel.



Classical concepts: entropy, noisy channel, channel capacity, coding theory, compression. See:

- Original paper Shannon [2001], *Link*
- The wiki page, Link

Theorem (Claude Shannon, 1948)**.** *For any given degree of noise contamination of a communication channel, it is possible to communicate discrete data (digital information) nearly error-free up to a computable maximum rate through the channel.*

I .2 The DES specification

For basics about Data Encyption Standard:

- FIPS specification: NIST [1977], *Link*
- DES wiki page
- Visual illustration of DES constants
- TripleDEs wiki page

Note that if main key length is 64-bits, the actual key strength is 56-bits because the DES key derivation algorithm begin by compressing the key.

Please:

- to distinguish two-key Triple DES from three-key Triple-DES
- what to think about $c = DES_{k_1}(DES_{k_2}(m))$ security ??
- concept of weak keys

I .3 Implement symmetrical algs in smart cards

I .3.1 Prevent information leakage

There are mainly five types of countermeasures against DPA :

- ***Secret data masking :***
  boolean masking : this method consist to Xor the secret data and with a random number generate for each algorithm execution, mostly used to dissimulate symmetrical secret.
  arithmetic masking : this method consist in the application of an addition between random value and sensible data, mostly used to dissimulate asymmetrical secret.

  Maghrebi et al. [a], *Link*

- ***Dual technology :*** This method roughlty consists in double all the bus, while on the first bus it traveling a value its complement is traveling on the other bus. Overall it permit to have the same consumption when we are manipulating whatever the manipulated bits.

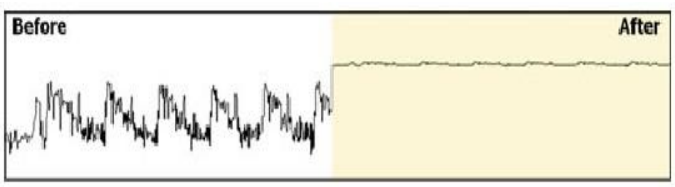
  Articles about masked dual-rail pre-charged logic technology (MDPL):

  Popp and Mangard [2005], *Link*
  Baddam and Zwolinski [2008], *Link*
  Rauzy et al. [2013], *Link*
  Cilio et al. [2013], *Link*

- ***Desynchronisation :*** This method consist in the addition of random loop, which create random waiting. It will be more difficult for the attacker to extract the information. Some like to distinguish these delay by their random delay: long delay -in ms- dummy cycle: few clock cycles -in μs- clock jitters: short delay -in ns-

- ***Filter:*** This method smooth the power consumption variation.

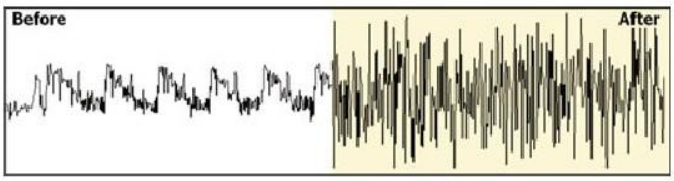
  LeGal et al. [2010], *Link*



  The previous pictures do not appear with Paul Kocher's kind authorization.

- ***Noise :*** this method consist in the 'random' augmentation the consumption of chip, like the previous method this method makes information more difficult to extract.

Brier et al. [2003a], *Link*



All this method can be used in parallel in order to limit the power of the attacker, moreover those method can be used in addition of a protected implementation of the algorithm to be protected.

I .3.2 The Transforming Masking method

Here is presented in detail for the DES algorithm an important counter-measure, called Transformed Masking Method presented [1] by M-L.Akkar and C.Giraud at the CHES'2001 Giraud [2007],

"what does prevent an programmer to effectively mask the intermediate values
by xoring the plaintext, M, with a random mask, R, without changing the cyphertext??"

the S-boxes !

I .3.2.1 General principle

Recall that in the NIST specification IP stands for the initial bijection of the DES, FP for the final one -inverse of the IP-, EP stands the compressive function (surjection) at the beginning of the f-function and P is the bijection of the f-function.

As a start, hereafter is the expression of the input of the S-boxes of the first round:

$$EP(M_{32-63}) \oplus K_1$$

In the case of a mask xored to the plain-text, the linearity of almost all DES operations -$\oplus$, IP, FP, E, P- play a central role, in the following lines, successively, we let the mask propagate itself till just before the S-box of the first round:

$$\begin{gathered} IP(M \oplus R)_{32-63} \\ EP(IP(M)_{32-63} \oplus IP(R)_{32-63}) \\ EP(IP(M_{32-63}) \oplus EP(IP(R)_{32-63}) \oplus K_1 \end{gathered}$$

Principles

To control the mask propagation become non realist in smart card: to perform the reverse operation is a complete non-sense, -S-boxes perform a non reversible operation-, to modify Sboxes two rounds by two rounds is also an option, but not in smart card it can only be done at an impressive memory cost. To anticipate the propagation of the mask, the only solution is to modify the S-box in order to remove the mask of its input:

$$SM(X) = S(X \oplus EP(IP(R)_{32-63})) \oplus somethingelse$$

This way, we have:

- The mask is not entering any S-boxes
- The output of any S-boxes shall is a function of the mask

Definition:
Are named respectively, for $i \in [1, 16]$ $DesLeft_i$, $DesRight_i$, $SecuredLeft_i$, $SecuredRight_i$ the left and right value at the beginning of the round i for each algorithms.

Proposition:

[1] Patterned and deprecated countermeasure !!

$$DesLeft_1 = IP(M)_{0-31} \qquad DesLeft_2 = IP(M)_{32-63}$$
$$DesRight_1 = IP(M)_{32-63} \qquad DesRight_2 = P(S(EP(M_{32-63}) \oplus K_1))) \oplus IP(M)_{32-63}$$

$$SecuredLeft_1 = IP(M)_{0-31} \oplus IP(R)_{0-31}$$
$$SecuredRight_1 = IP(M)_{32-63} \oplus IP(R)_{32-63}$$

$$SecuredLeft_2 = IP(M)_{0-32} \oplus IP(R)_{32-63}$$
$$SecuredRight_2 = P(S(EP(M_{32-63}) \oplus K_1))) \oplus P(somethingelse) \oplus IP(M)_{0-31} \oplus IP(R)_{0-31}$$

Then, an interesting relation can be viewed at the beginning of the first round:

$$DesLeft_1|DesRight_1 \oplus IP(R) = SecuredLeft_1|SecuredRight_1$$

or equivalently:

$$\begin{cases} DesLeft_1 \oplus IP(R)_{0-31} &= SecuredLeft_1 \\ DesRight_1 \oplus IP(R)_{32-64} &= SecuredRight_1 \end{cases}$$

Key idea If the previous relation hold for all round i, the masked algorithm is finished!!

To finish the construction changes have to be made so as that the previous to be extended to the second round. Using the proposition on previous page, let's take to a look at what happen at the end of the second round, and what should be modified so as the previous equations to be true for each round.

$$\begin{gathered} IP(M)_{0-31} \oplus IP(R)_{0-31} \\ = \\ IP(M)_{0-31} \oplus IP(R)_{32-63} \end{gathered}$$

$$\begin{gathered} P(S(EP(M_{32-63}) \oplus K_1))) \oplus IP(M)_{0-31} \oplus IP(R)_{32-63} \\ = \\ P(S(EP(M_{32-63}) \oplus K_1))) \oplus IP(M)_{0-31} \oplus P(somethingelse) \oplus IP(R)_{0-31} \end{gathered}$$

From these can be deduced

- The second equation impose to verify:

$$\begin{gathered} P(somethingelse) = IP(M)_{0-31} \oplus IP(M)_{32-63} \\ i.e. \\ somethingelse = P^{-1}(IP(M)_{0-31} \oplus IP(M)_{32-63}) \end{gathered}$$

- The Feistel scheme of the DES algorithm have to be modified a bit: the xor at the end of the rounds of the Feistel scheme is followed with a xor of the value $IP(M)_{0-31} \oplus IP(M)_{32-63}$

Let's discoverer the solution from the author, with their much more digest standards notations.

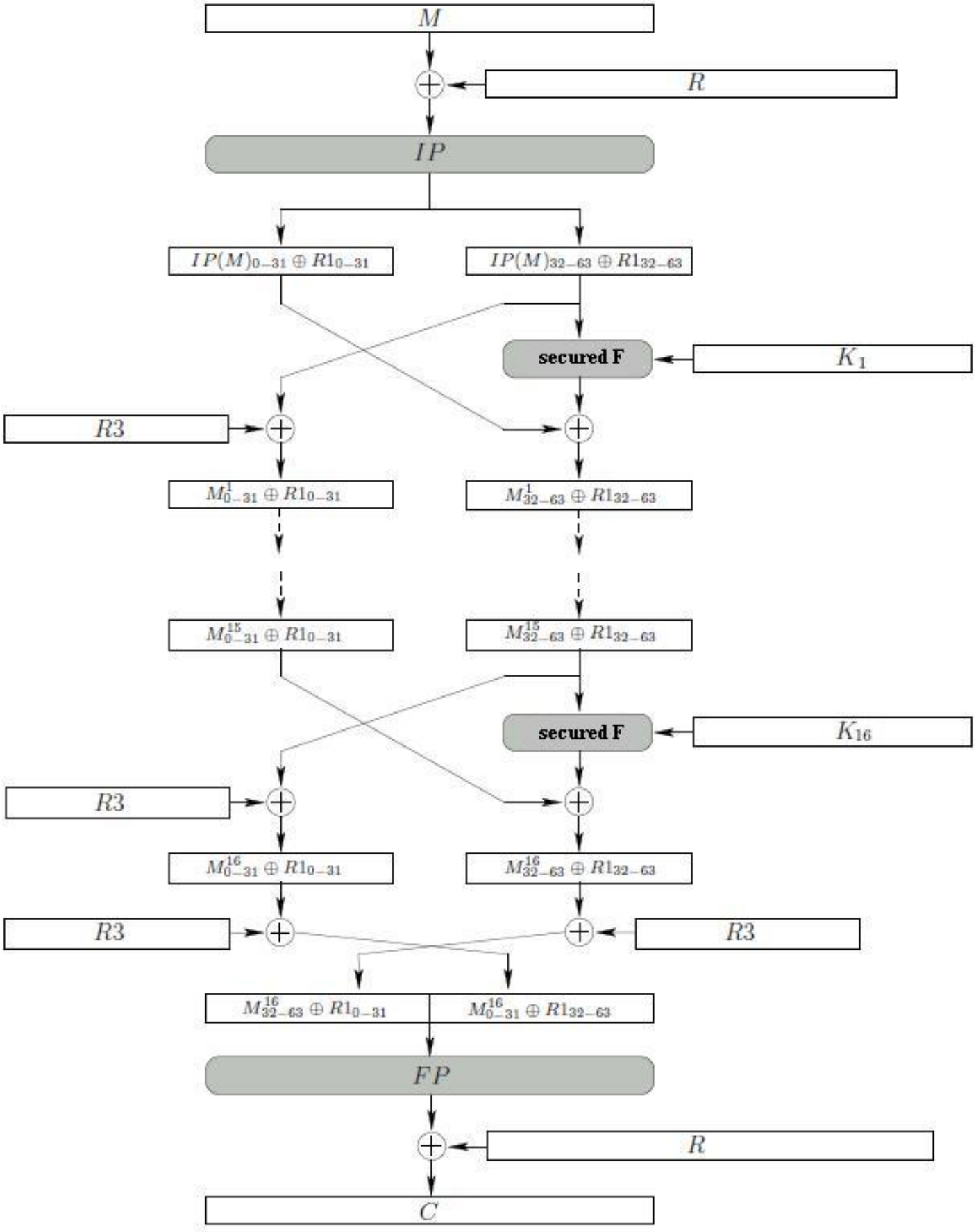
I .3.2.2 Notation

Notation: following NIST DES spec, R is for a 8 bytes mask, the following notation are required to simplify notation:

$$\begin{gathered} R1 = IP(R) \\ R2 = EP(R1_{32-63}) \\ R3 = R1_{0-31} \oplus R1_{32-63} \end{gathered}$$

The idea behind this algorithm is to mask all the intermediate values in such way that the ciphertext will be the same. The mask R, Xored with the main key, will be propagated during a whole round. The algorithm is build in such a manner that the mask is not allowed to enter in the -non linear- S-Box, then all the mask propagation is linear and finally easily controllable.

At the end of a round result is the same than with a normal DES execution but Xored with always the same value : $R1$. To anticipate the mask's propagation and be able to control them some evolution of the mask have to be pre-computed before the execution of the algorithm.

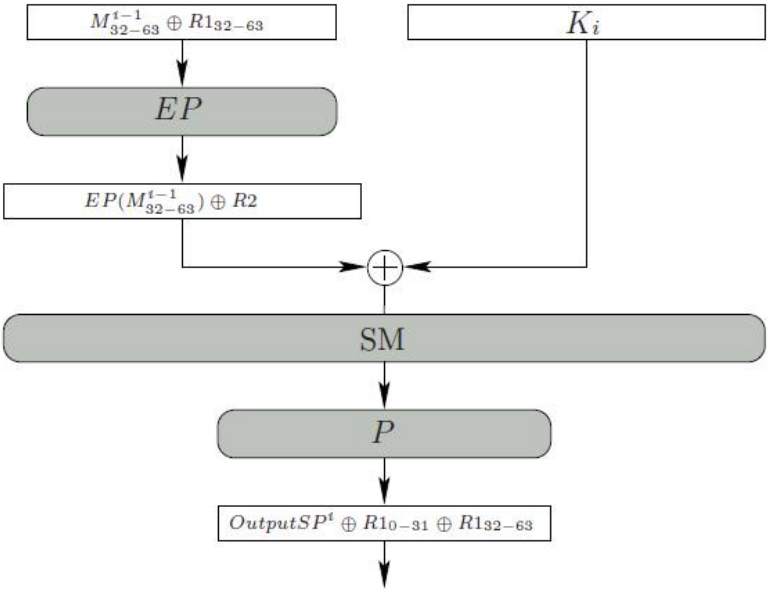


Secured DES general scheme

Note, that those pre-computation have to be computed in a very safe way. Indeed, if an attacker could know those values, then immediately she/he could anticipate the SM-boxes for each encryption, therefore they would just launch a DPA attack simply changing the S boxes for the SM-boxes at each encryption.

$$SM(X) = S(X \oplus R2) \oplus P^{-1}(R3)$$

The algorithm has to be modified so as to the mask no to enter in the Sbox. We obtain before the S-Box a intermediary value masked with $R2$. And clearly with this with new definition of the Sbox if the input is masked with R_2 then R_2 absolutely does not influence the output of those modified SMbox.



Secured F function

I .3.2.3 A round with a secured DES

Here will be presented a complete round with the secured DES, and we will see that that at the end of a round the difference between a normal DES and the secured DES is always constant and worth $IP(R)$.

Here the letter M is redefined for M^i with the following convention

- M^0 stands fir the plain-text.
- $\forall \in [1, 16],\ M^i$ stand for the left—right value at the beginning of the i^{th} round.
- M^{17} stands fir the plain-text.

First of all, let's recall what is the output of this round DES algorithm, if we concatenate the two part, we have :

$$DesLeft|DesRight = M^i_{32-63}|P(S(EP(M^i_{32-63}) \oplus K_i))) \oplus M^i_{0-31}$$

Starting [2] at the beginning of a round, noted i, let's first calculate the result at the end of this round for the left side:

[2] In fact this lines are the induction step of a recurrence proof

$SecuredLeft_i = M^i_{0-31} \oplus R1_{0-31}$
$SecuredLeft^{'} = M^i_{32-63} \oplus R1_{32-63} \oplus R3$
$SecuredLeft_{i+1} = M^i_{32-63} \oplus R1_{0-31}$
Secondly the same computation will be done on the right side:
$SecuredRight_i = M^i_{32-63} \oplus R1_{32-63}$
$SecuredRight^{'} = EP(M^i_{32-63}) \oplus EP(R1_{32-63})$
$SecuredRight^{'} = EP(M^i_{32-63}) \oplus EP(R1_{32-63}) \oplus R2$
$SecuredRight^{'} = EP(M^i_{32-63}) \oplus K_i$
$SecuredRight^{'} = S(EP(M^i_{32-63}) \oplus K_i)) \oplus P^{-1}(R1_{0-31} \oplus R1_{32-63})$
$SecuredRight^{'} = P(S(EP(M^i_{32-63}) \oplus K_i))) \oplus R1_{0-31} \oplus R1_{32-63}$
$SecuredRight_{i+1} = P(S(EP(M^i_{32-63}) \oplus K_i))) \oplus M^i_{0-31} \oplus R1_{32-63}$

$$SecuredLeft_i|SecuredRight_i = M^i_{32-63} \oplus R1_{0-31}|P(S(EP(M^i_{32-63}) \oplus K_i))) \oplus M^i_{0-31} \oplus R1_{32-63}$$

As a result we can see that the result, which explain that the variable obtained at the end of a round a secured DES is the same than the one obtained by a normal DES Xored by $R1$:

$$SecuredLeft_i|SecuredRight_i = DesLeft_i|DesRight_i \oplus R1$$

I .3.2.4 Resistance against the first order DPA

As we could see all the sensible variables of the secured DES are masked, moreover this boolean random mask has a uniform distribution.

With the following lemma :
Let $\alpha \in {\mathbb{F}_2}^n$ ***an independent variable and*** β ***another variable distributed uniformly on*** ${\mathbb{F}_2}^n$ ***and independent with*** α***. The variable*** $\alpha \oplus \beta$ ***is distributed uniformly and is independent with*** α***.***

Consequently we can deduct that all the variable which are manipulated during the execution of the algorithm are uniformly distributed and independent with the input and the key round. Thus this method resist against the attacks of the first order DPA.

I .3.2.5 Practical implementation

In fact if the transforming masking method clearly prevent DPA attacks, -see before- it is not immune against side channel attacks -see I .4.5.1-. Therefore, this implementation can not be considered as secure anymore. But in fact the "The superposition attack" can be prevented quick easily in fact: we just have to forbid the superposition of the curve of 1st and 16th round. This is why this Protected DES can be used at the condition to use two set of SM-box (one for round 1 to 8 and another for round 8 to 16). In this configuration it is a quite popular implementation.

I .4 Attacks symmetrical algs in smart cards

I .4.1 Differential Power Analysis

Differential Power Analysis was originally proposed by P.Kocher, J.Jaffe and B.June Kocher et al. [1999] in 1998. As the power consumption is dependant from value manipulated by the algorithm -see section **??**- this dependency can leads to an attack when a manipulated value depends only on few bits of the secret key.

The principle of the Differential Power Analysis is to make hypothesis about those a part of the key, to simulate the power consumption under this hypothesis and then to perform statistical -see section **??**- comparisons with real curves.

This attack require a large number of power trace, comparing to Simple Power Analysis, but it can reveal the secret key of a device even if the recorded power traces are extremely noisy. This dependency can be exploited when a variable is manipulated and only depend on few bits of the secret key.
To achieve such an analysis several hypothesis have to be made:

- about the model of power consumption, this is very sensitive.
- about the way that the algorithm is designed.
- about the manipulated variable.
- about a part of the key, noted K_j, on which depend the targeted variable.

Taking the example of the DES algorithm, the aim of this attack is to reveal round key, for that it calculate 8 part of the round key, in order to get all the key with it. Once the attack is achieved, it can be launched again to calculate another part of the round key, and finally get the whole round key.

Step 1: Choosing an Intermediate Result.

The first thing to do is to choose an intermediate result of the algorithm that is executed by the attacked device, by example on the DES algorithm we can choose the output of a S-box.

So to attack an algorithm with this method you need an exemplary of this one, please note that we are only interested in the value of this intermediate result so the type of implementation of the target algorithm does not matter, only its design does(*e.g.* where we take the intermediate value don't depend od the Des implementation).

This intermediate result have to be a function $f(d, k)$, called *selection function*, where k is a small part of the key and, most of the time, d is either the plain-text or the cypher-text.To avoid false peaks the result of the f function must be uniformly distributed :

$$i.e.\ \forall k\ \mathbb{P}(f(M_j, K_k) = 1) = \tfrac{1}{2}$$

Step 2: Measuring the Power Consumption.

Here we will simulate a run of the smart card. To do so we'll need to have several text that is to be a plaintext or a cypher text, noted $d_i \in \{d_1, \ldots, d_D\}$ and to encrypt -or decrypt- all of this values. For each of these encryptions -or decryptions- runs the attacker will know this value involved in the calculation of the intermediate result chosen at step 1 and he will records the corresponding power trace.

Each of this trace, $\mathcal{C}_i$, will be noticed as following $(t_{i,j})_j = (t_{i,1}, \ldots, t_{i,T})$ for the i^{th} consumption trace corresponding to d_i and where T is denotes the length of the trace. The attacker measure the trace for each data block, and hence, the traces can be written as matrix $\mathcal{T}$ of size $D \times T$.

It is important for DPA attacks that the measured trace are correctly aligned, because this means that the power consumption value of each column of $\mathcal{T}$ need to be caused by the same operation. In order to do this two different means: to record traces on a oscilloscope with trigger signal, or to align curve with mathematical algorithms.

Step 3: Calculating Hypothetical Intermediate Values.

The next step of the attack is to calculate hypothetical intermediate value for every possible choice of k_j. Let's write $k_j \in \{k_1, \ldots, k_K\}$ where K denotes the total number of possible choice for the round key. K will be 64 bits because the length of one round key is 6 bytes and for each byte there are only two possibilities 1 or 0. In the context of DPA attacks, we usually refer to those elements as *key hypothesis*. Now an attacker can easily calculate $f(d_i, k_j)$. This construction results in a matrix $\mathcal{V}$ of size $D \times K$ defined by the following relation:

$$\forall(i,\ j) \text{ such as } 1 \leq i \leq D \text{ and } 1 \leq j \leq K : v_{i,j} = f(d_i, k_j)$$

The j^{th} column of $\mathcal{V}$ contains the intermediate results that have been calculated on the key hypothesis $k = k_j$ and as we are trying all possibility for k the value used in the device is one of them. The goal of DPA is to find this correspondence.

Step 4: Mapping Intermediate Values to Power Consumption Values.

For this purpose, the attacker uses a technique simulation. The quality of this simulation strongly depends on the knowledge of the attacker about the device. **The better this simulation of the attacker matches the actual power consumption characteristics of the device, the more effective is the DPA attack.** We will see next how to do it. This mapping applied to the matrix $\mathcal{V}$ give a matrix, noticed $\mathcal{H}$ of the same size: $D \times K$.

Step 5: Comparing Hypothetical to Recorded Consumption.

Now the final step of the DPA attack can be performed: each column of the matrix $\mathcal{H}$ is compared with each column of the matrix $\mathcal{T}$ with some statistical method. This means that the attacker compares the Hypothetical power consumption values of each key hypothesis with the recorded trace at every position. The result of this comparison is a matrix $\mathcal{R}$ of size $K \times T$, where each element $r_{i,j}$ contains the result of the comparison between the columns h_i and t_j. The comparison is done based on algorithms I will discuss later. **But in every algorithm, one thing is similar : the value $r_{i,j}$ is the higher, the better the columns h_i and t_j match.** The key of the attacked device can hence be revealed based on the following observation.

The power traces correspond to the power consumption of the device while it executes a cryptographic algorithm using different data inputs. The intermediate result that has been chosen in **step 1** is a part of this algorithm. That's why the device needs to calculate the intermediate values of the matrix $\mathcal{V}$ during the execution of the algorithm. Consequently, also the recorded traces depend on these intermediate values at some position.

The hypothetical power consumption values in $\mathcal{H}$ have been simulated by the attacker based on the values of the matrix $\mathcal{V}$. In fact, the column of $\mathcal{H}$ and $\mathcal{T}$ are strongly related and lead to the highest values in the matrix $\mathcal{R}$. All other values of $\mathcal{R}$ are low because the other columns of $\mathcal{H}$ and $\mathcal{T}$ are not strongly related.

The index of the line in which is the highest values of the matrix $\mathcal{R}$ is reveals which of the key hypothesis is the more probable.

problem 1 In practice the matrix $\mathcal{R}$ have sometimes the same coefficient. In this case, the attacker has usually not measured enough power traces to estimate the relationship between the columns of $\mathcal{H}$ and $\mathcal{T}$. Indeed the more traces an attacker measures, the more elements are in the columns $\mathcal{H}$ and $\mathcal{T}$, and more precisely the attacker can determine the relationship between the columns.

problem 2 This also implies that the more measurements are made, the smaller relationships between the columns can be determined. So I had to get a good number of measurements. On a DES implementation with no counter measure 5000 traces are enough to find a round key, see section **??**.

The following diagram is illustrating the steps 3 to 5 of a DPA attack.

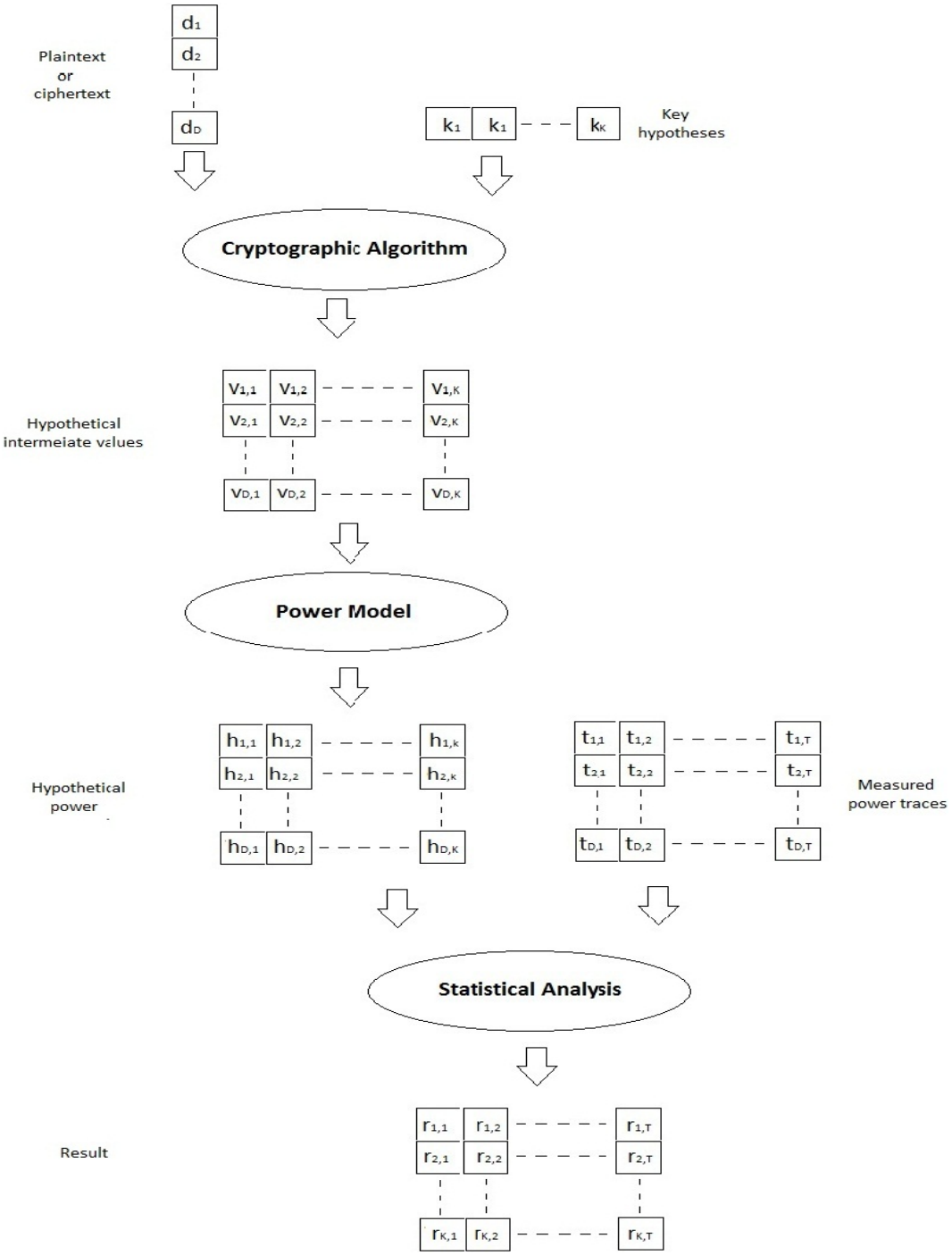


Figure 16. Schema of the DPA attack

I .4.2 About the power model

Once the $\mathcal{V}$ matrix is filled, it have to be transformed to $\mathcal{H}$ matrix thanks to power consumption model, this model is crucial because it will depend the success of the attack.

Linear models This model is simple and efficient because it permit to separate the consumption of the target electrical component (register, RAM cell, bus) from the rest of the circuit.

For every component and at any time we have :

$$C = C_{all\ the\ circuit\ but\ the\ component} + C_{component}$$

Nowadays, technology CMOS is most widespread to implement the integrated circuits. With this kind of technology, as said in Messerges [2000], the component mainly consumes current when there is a change of state of the logical parts between two blows of clock: the static consumption is negligible.

Asymmetric model Let $B = (\beta_m, ..., \beta_1)$ the value that will be saved in the targeted component, and $A = (\alpha_m, ..., \alpha_1)$ the value previously stored in this component.
Let also c^i_{01} and c^i_{10} be respectively the amount of currents needed to change a bit from 0 to 1 and from 1 to 0 for the i^{th} bit. If after a clock the value of the register is set to B, the consumption after the shot clock is:

$$C(t) = \theta + \sum_{i=1}^{m} (1 - \alpha_i)\beta_i c_{01} + \alpha_i(1 - \beta_i)c_{10}$$

where θ is the electrical consumption due to others part of the circuit, and c_{01}, c_{10} and θ are function of time.

Symmetric models As proposed by Brier Brier et al. [2003b] we assume that $c_{01} = c_{10}$, the model of power consumption, can be simplified using $d_{\mathcal{H}}$ for the Hamming distance operator to:

$$C(t) = \theta(t) + d_{\mathcal{H}}(A, B) \times c(t)$$

where $d_{\mathcal{H}}$ is the Hamming distance operator.
This model can be simplified again if assumed that the reference state A is always zero, this model has been proposed by T.Messerges, where $\omega_{\mathcal{H}}$ is the Hamming weight operator.

$$C(t) = \theta(t) + \omega_{\mathcal{H}}(B) \times c(t)$$

Hamming weight Finally the most common, for the -realistic- case that θ and c are not considered as function of time:

$$C(t) = \omega_{\mathcal{H}}(B)$$

Other model exist, like the one from S.Aumonier Aumonier [2007] based on the concept of mutual information. This model is much more general and precise that previous one, but on the other hand this model is much more slower.

Conclusions

- There are a many power model more or less close to the reality. As always, the simpler it is, the quicker it is and of course the less precise it is.
- Note that on Kocher's original article Kocher et al. [1999]: "mono-bit DPA" the intermediate value is only constituted of one bit and the present section who studied the problem of the power consumption is not relevant in this case.

I .4.3 Formalization

I .4.3.1 P.Kocher's DPA mono-bit

Hereafter is presented the original Differential Power Analysis, also called mono-bit DPA , which was published on P.Kocher, J.Jaffe & B.June, Kocher et al. [1999], *Link* . This attack is called mono-bit because the f function only returns a single bit, with such a statistical model there is no need of power model *i.e.*:

$$f \longrightarrow \{0,\ 1\}$$
$$\mathcal{H} = \mathcal{V}$$

The principle of this statistical test is to separate traces in two classes or packet the trace that have been collected from oscilloscope, and then to compute the means of each class. The way used to separate those trace in two class, noted G_0 and G_1, lies in the value returned by the function f:

$$G_{0,k} := \{\mathcal{C}_i \| f(M_i, K_k) = 0\}$$
$$G_{1,k} := \{\mathcal{C}_i \| f(M_i, K_k) = 1\}$$

Then is computed the *signal of decision the the sub key* K_k:

$$\Delta_{K_j} := \overline{G_{0,k}} - \overline{G_{1,k}}$$

If the hypothesis made is not the good one then the distribution between the two group G_0 and G_1 are random and after subtraction the curve of decision is narrow to zero. If this hypothesis is good then appears a pic at the moment when the target bit is manipulated, so, the aim of the attacker is to find the curve with the biggest peak.

advantage: Robust ans simple

inconvenient: false pike might occurs!
To avoid this several hypothesis have to be verified:

- the result f function must be uniformly distributed : $\forall k\ \mathbb{P}(f(M_j, K_k) = 1) = \frac{1}{2}$
- The selection function must have all its output bits independent
- The power model have to be relevant

I .4.3.2 T-S.Messerge's DPA multi-bit

Is presented now, an improvement of the previous statistical tools, and called All-or-Nothing multi-bit DPA, introduced by T-S.Messerges Messerges [2000], *Link* . The main idea of this attack is to increase the distance between means with a f function returning m bits instead of only one.

$$f \longrightarrow \{0,\ 1\}^m$$

To achieve this objective curves will be partitioned in three class, here are the two class that will be used for the computation of the signal of decision:

$$G_{0,k} := \{\mathcal{C}_i \| f(M_i, K_k) = \overbrace{0 \dots 0}^{m\ bits}\}$$
$$G_{1,k} := \{\mathcal{C}_i \| f(M_i, K_k) = \overbrace{1 \dots 1}^{m\ bits}\}$$

And:

$$G_{01,k} = \{\mathcal{C}_i \| \mathcal{C}_i \notin \{G_{0,k} \cup G_{1,k}\}\}$$

The attacker can compute the curve of the distance between the means of each class:

$$\Delta_{K_j} := \overline{G_{0,k}} - \overline{G_{1,k}}$$

advantage:

With this method the goods pike will be m times much higher than with mono bit DPA
inconvenient:

All curves of $G_{01,k}$ are lost for this attack, which clearly represent most of them.

In order that multi-bit DPA to be still effective a compromise in the definition of the partition have to be found, nowadays the most accepted one is:

$$G_{0,k} := \{\mathcal{C}_i \| \omega_{\mathcal{H}}(f(M_i, K_k)) < m/2\}$$
$$G_{1,k} := \{\mathcal{C}_i \| \omega_{\mathcal{H}}(f(M_i, K_k)) > m/2\}$$

advantage: far much relevant than All-or-nothing DPA.
inconvenient: If m is odd, some curve won't be used.

I .4.3.3 Correlation Power Analysis

The main problem with the multi-bit attack previously presented is that all the curve are not always used when m is odd which append all the time because all kind of architecture have an odd number of bits. This point have encourage researchers to find another method which would not use a partition of the curves in different classes.

Few years after the publication of T.Messerges Messerges [2000] Brier et al. [2004], *Link* published a new method transforming the problem of an attack changing the problem of detection of a difference of means for a problem of detection of a correlation between a set of measures and a power consumption model.

In DPA attacks, the correlation coefficient is used to determine the linear relationship between the i^{th} column of $\mathcal{H}$ and the j^{th} column of $\mathcal{T}$ for $i \in \{1, \dots, K\}$ and $j \in \{1, \dots, T\}$. This results in a matrix $\mathcal{R}$ of estimated correlation coefficients. We estimate each value $r_{i,j}$ based on the D elements of the columns h_i and t_j. So we can rewrite with the same notation the following formula :

$$r_{i,j} = \frac{\sum_{d=1}^{D}(h_{d,i} - \bar{h_i}).(t_{d,j} - \bar{t_j})}{\sqrt{\sum_{d=1}^{D}(h_{d,i} - \bar{h_i})^2}.\sqrt{\sum_{d=1}^{D}(t_{d,j} - \bar{t_j})^2}}$$

advantage: All the curve are systematically used.

I .4.3.4 Partitioning Power Analysis

In 2006 a group of researcher composed by T-H.Le, J.Clédière, C.Canovas, B.Robisson, C.Sérvière and J-L Le et al. [2006], *Link* presented a new method that was generalizing all DPA attacks in the right lineage of Kocher et al. [1999]. In order to generalize the multi-bit DPA method, they purpose this method defining a large number of packets. The concept of multi-partitioning has been suggested by M.L-Akkar and Al, but these authors did not formalize the concept.

For this method the hypothesis about power consumption have to be done, so we will begin to explain this attack from the H matrix that has been introduced previously. The goal, when attacking a DES algorithm, is still to find a part of a round key associated to a fixed S-Box.
This method can be explain by the following steps :

- Define packets : our selection function f is returning m bits, in this case we will defined $m+1$ packets defined with the previous notation :

$$\forall\ 0 \leq j \leq m\text{: } G_{j,k} := \{\mathcal{C}_i, i = 1..N \| \omega_{\mathcal{H}}(f(M_i, K_k)) = j\}$$

- Created average : to each packets will be associated the average of the curve belonging to it, that will be noted $\overline{G_{j,k}}$.
- Computation of the decision signal :

$$\Delta_{K_j} := \sum_{j=0}^{m} a_{j,k}\ \ \overline{G_{j,k}}$$

  where $(a_{j,k})_{0 \leq j \leq m}$ are weight that will strongly impacts the performance of this attack. Those weight can be dependent or not of the key assumption, in our case we have $a_{j,k} = a_j$.

Briefly, the attack works similarly as the DPA, the exception is after the step 5 of our DPA description. Indeed, we put the traces in m packets, and then we calculate Δ_{K_j}. As usually with such a signal decision the curve on which is maximum of this value will possibly indicate the correct hypothetical key. It's done for one S-box, so we have to do the same with all of them.

Advantages :
Less curves are necessary to recover a key.
As the partition is more precise, the results are much more aches-2006-lee
Strong reduction of the false and secondary peaks. The next figure illustrate it.

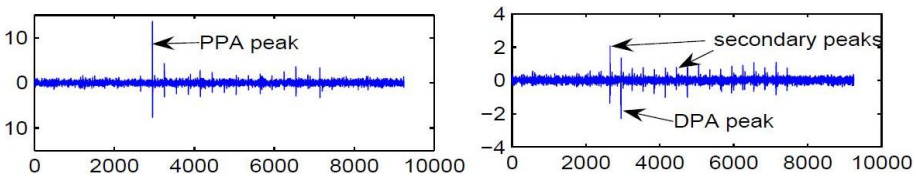


Figure 24. Comparison between PPA and DPA for a good key hypothesis

inconvenient:
In the case of $m = 2$, all curve of G_2 will not be used for the computation of Δ_{K_j}.

I .4.3.5 PPA generalize all previous DPA attacks

1) For specific values of the weight we are in fact defining well known attacks :
• if the values are : $a_0 = -1$ and $a_4 = 1$ and other set to 0 it's the P.Kocker mono-bit DPA.
• if the values are :

$$a_j = \begin{cases} -1 \text{ if } 0 \leq j < m/2 \\ 1 \text{ if } m/2 \leq j \leq d \end{cases}$$

it's the P.Kocher mono-bit DPA.
it's the all T.Messerges multi-bit DPA and OAN-DPA.
2) The reader can also read the demonstration which explain that the CPA is also a particular case of PPA for some specific values of a_j.
3) Optimal values for weights are $(a_j)_{0 \leq j \leq m} = \{-1, -2, 0, 2, 1\}$, this set of weights is maximizing the Signal Noise Ratio for the linear model of power consumption. See **?**, *Link* .

I .4.3.6 Maximum of probability

This method was invented by R.Bevan and E.Knudsen [1]. This method takes into account what appends when the amount of current to pass a bit from 0 to 1 is different from that needed to pass a bit from 1 to 0. With the model of consumption we have :

$$\mathcal{C}_i = \sum_{k=1}^{p} \lambda_{ki}\theta_k + w_i,$$

which θ_k is the parameters of the consumption model and p the number of parameters.
For N different consumption measures, we have the matrix relation :

$$\mathcal{C} = \begin{pmatrix} \mathcal{C}_1 \\ \vdots \\ \mathcal{C}_N \end{pmatrix} = \begin{pmatrix} \lambda_{1,1} & \lambda_{2,1} & \dots & \lambda_{p,1} \\ \vdots & \vdots & & \vdots \\ \lambda_{1,N} & \lambda_{2,N} & \dots & \lambda_{p,N} \end{pmatrix} \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_N \end{pmatrix} + \begin{pmatrix} w_1 \\ \vdots \\ w_N \end{pmatrix}$$

with this notation the attacker will keep the key which maximize the following relation :

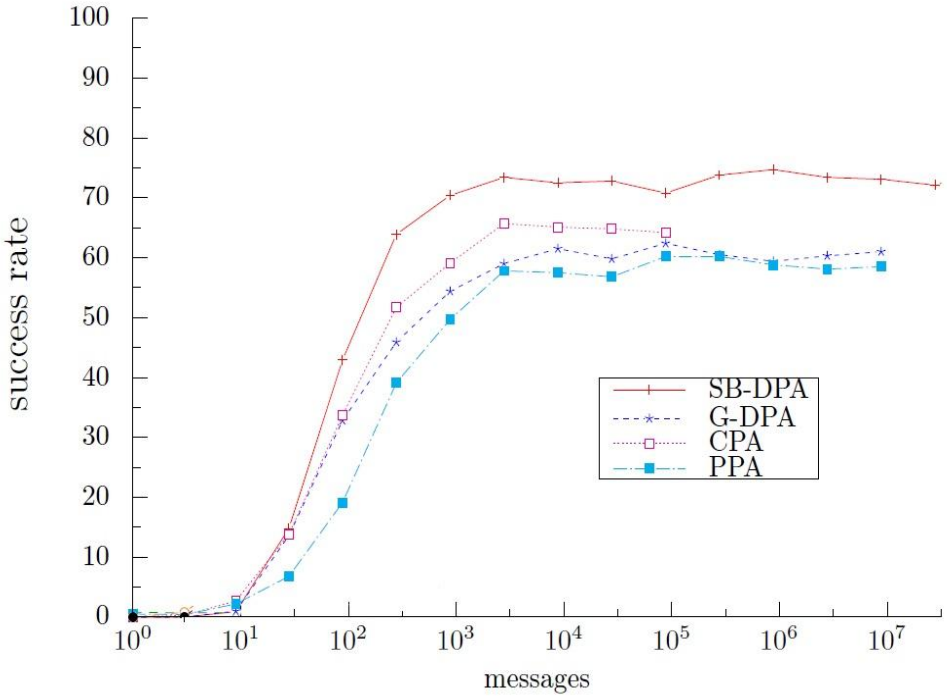
$$L_{K_j} = \frac{N}{2}(ln({}^t\mathcal{C}.\mathcal{C} - \frac{1}{N}(\sum_{i=1}^{N} \mathcal{C}_i)^2) - ln({}^t\mathcal{C}.\mathcal{C} - ln({}^t\mathcal{C}.\lambda_{K_j}.({}^t\lambda_{K_j}.\lambda_{K_j})^{-1}.{}^t\lambda_{K_j}.\mathcal{C}))$$

I .4.4 Compare those methods

Reference The two references could be :

- In 2004 Regis Bevan PhD in Bevan [2004], *Link* : are compared mono-bit, 'All or Nothing multi bit DPA', multi bit DPA and CPA.
- In 2011 Julien Doget's article Doget et al. [2011], *Link* : much more comparison, ultimately clear notations.

Hereafter picture beeing taken from the excellent article of Julien Doget, a comparison in term of efficiency between many statistical methods. Note that mono bit DPA return bit of key whereas the other return 6.



I .4.5 High Order DPA attacks

See Giraud [2007], his PhD study in detail this attack on AES, and see also the original article Akkar and Goubin [2003] and Akkar and Giraud [2001].

We saw that testing hypothesis about a manipulated variable depending on a part of the key, part small enough so that a brute force search could be undertaken, could lead to reveal the used secret key.

We saw afterwards, that this kind of threat can be prevented, simply by breaking the link between the manipulated value and the power consumption: this means practically that no attacker shall be able to anticipated any sensitive manipulated value by technique such as masking.

In fact this kind of countermeasure can be bypassed, by another class of side channel attack generalization of the DPA seen previously. Instead of doing hypothesis on one manipulated variable, those Hight Order DPA attacks do hypothesis one multiple intermediate value.

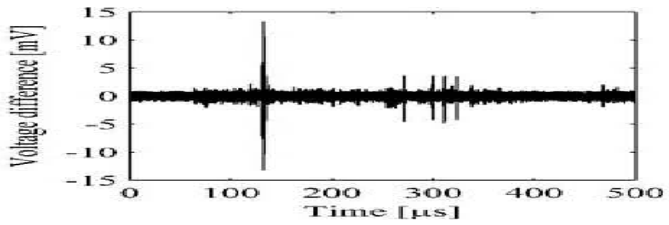


Figure I .1: Successful DPA attack

More than the number of sub key hypothesis -which is squared for HODPA square- the main difference between DPA and -true- HODPA is that DPA attack target one variable manipulated in moment in time, τ whereas HODPA target multiple variable manipulate at moment in time τ_1, τ_2, ...

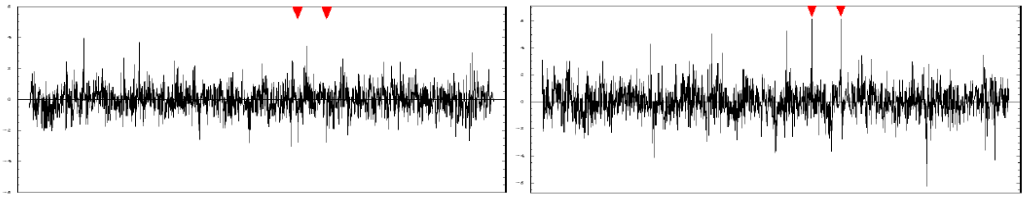


Figure I .2: Unsuccessful *vs* successful HODPA attack

I .4.5.1 A particular case: The superposition attack

The superposition attack proposed by M-L.Akkar and L.Goubin against their own implementation ' the Transforming masking method', see I .3.2. This is a second order DPA attack in theory because the selection function used is function of two distinct sub-keys. But in practice, it is nearly as simple as an usual DPA attack. '#1.333 HODPA'

The idea is as following: in a second order DPA attack, the most difficult thing is to localize the time when the precise needed values are manipulated, but on the contrary, localizing a whole DES round is often quite easy. So instead of correlating precise parts of the consumption traces, the attacker will just correlate the whole trace of the first and the last round.

The main idea is to perform a front attack with a plain-text M, making an hypothesis on a part of K_1, in the same time to perform a back-end attack with a cypher text C, making another hypothesis on a part of K_16. For one S-Box there will have 4096 possibilities for a sub-key hypothesis.

As we can see, by example with picture *26* , every output of SM-Box are masked with a value depending on the main mask, the thing that allow an attack to be possible is the fact that the value $P^{-1}(R1_{0-31} \oplus R1_{32-63})$ is always the same.

Then we have:

$$T = S(EP(IP^{-1}(C)_{32-63}) \oplus K_{16}) \oplus P^{-1}(IP(R)_{32-63} \oplus IP(R)_{0-31}))$$
$$\oplus S(E(IP(M)_{32-63}) \oplus K_1) \oplus P^{-1}(IP(R)_{32-63} \oplus IP(R)_{0-31})))$$

$$T = S(E(IP^{-1}(C)_{32-63}) \oplus K_{16}) \oplus S(E(IP(M)_{32-63}) \oplus K_1)$$

Key idea : that the value T does not depend on the random masking value.

On the next page is explained the different step of the algorithm, one can note that the presented version, taken from Akkar and Goubin [2003], is a mono-bit DPA attack and that T.Messerge's multi-bit attack, would also work just like a CPA attack.

Algorithm 1: Superposition Attack on S-boxes

Input: Curves resulting of the addition of traces from round 1 and 16

```
y ← 1 ;
for i = 0 j < 64 i++ do
    for j = 0 j < 64 j++ do
        for All the Curves do
            Separate the curves in two packets depending on one bit of the part of T which is corresponding to
            the attacked S-box and assuming that the part of K1 and K16 corresponding to the attacked S-box
            are i and j.
        Average and subtract the separated curves.
Choose the value of i and j where the greatest peak appears.
Check the coherency the i and j has to be compatible. ;
return i, j;
```

Note that this algorithm requires as an input to sum two round of a DES, if there is a problem of alignment this will dangerously compromise the attack.

I .4.5.2 General HODPA

In classical DPA attack once that the trace have been processed to remove noise, aligned and so on it sure that if an attack was launched on those traces then peaks would occurred exactly when the targeted variable is manipulated. This is assuming few hypothesis:

- power consumption was measured while the targeted variable was manipulated
- a relevant power model is available and compatible with the countermeasure.

In HODPA as multiple manipulate variable are targeted, and as those variables are manipulated at different moment in time, let's say τ_1 and τ_2, it wouldn't make any sense to launch this attack on recorded trace, the trace to be process align and then separate in packet are in fact traces that have not been recorded.

$\mathcal{T}_i(t)$, attacked traces, for HODPA targeting two variables, in function of $\mathcal{C}_i(t)$ the recorded trace:

$$\mathcal{T}_i(t) = \mathcal{C}_i(t) - \mathcal{C}_i(t + \tau_2 - \tau_1)$$

This way for $t = \tau_1$ the fist contribution of the attacked curve $\mathcal{C}_i(t)$ will give pike relative to the manipulation of the first variable while at the the second contribution of the attacked curve $\mathcal{C}_i(t + \tau_2 - \tau_1)$ will give pike relative to the manipulation of the second variable. Morality: with such a trace $\mathcal{T}_i(t)$, peak will pop up at $t = \tau_1$ revealing that the the two targeted variable are manipulated at τ_1 and τ_2, . Note that on a natural case, $\tau_2 - \tau_1$ is not known , moreover τ_2 and τ_1, then all the possibility would have to be tested .... Huge increase of the complexity.

Chapter II

Asymmetric encryption

II .1 RSA: description

II .1.1 References

More in Menezes [1996] :

- fundamentals, chapter 2: "number theory", "finite fileds", "abstract algebra"
- Asymetrical PKCS, chapter 8: "RSA public key encryption"

We recall the classic notation: $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$, for this document all the computation will take place in the following 'set': If n is a positive integer:

$$\mathbb{Z}/n\mathbb{Z} = \{0, , ..., n-1\}$$

where classical addition, subtraction, and multiplication are defined just like in $\mathbb{Z}$ with the amendment that results must be in the set $\mathbb{Z}/n\mathbb{Z}$.

II .1.2 Modular arithmetic

Definition. $x \in \mathbb{Z}/n\mathbb{Z}$ is said to be invertible in the ring $\mathbb{Z}/n\mathbb{Z}$
if there exist $y \in \mathbb{Z}/n\mathbb{Z}$ such that $x * y = 1$ in $\mathbb{Z}/n\mathbb{Z}$

Notation:
the set of invertible of $\mathbb{Z}/n\mathbb{Z}$ is noted $(\mathbb{Z}/n\mathbb{Z})^\times$
the number of invertible of $\mathbb{Z}/n\mathbb{Z}$ is noted $\phi(n)$

Definition. A prime number is a natural number that has exactly two distinct natural number divisors: 1 and itself.

Definition. Two integers a and b are said to be co-prime if they have no common positive factor other than 1.

Theorem. *If $n = p * q$ where p and q are prime numbers,*
Then $\phi(n) = (p-1)(q-1)$

Theorem. *Euler's theorem*
If $x \in \mathbb{Z}/n\mathbb{Z}$ with x co-primewith n, then $x^{\phi(n)} = 1$ in the set $\mathbb{Z}/n\mathbb{Z}$

RSA stands for Ronald Rivest, Adi Shamir et Leonard Adleman who first publicly described it in 1976, but it might be true that it was known before by some governmental agencies.

Keys generation

- Choose two distinct, 'same size' big prime numbers p and q.
- Compute $n = pq$. It define the set $\mathbb{Z}/n\mathbb{Z}$ where all the computation will take place.
- Compute $\phi(n) = (p-1)(q-1)$ it must be kept secret.
- Compute $\lambda(n) = lcm(p-1, q-1)$.

- Choose an integer e such that $1 < e < \lambda(n)$ and $gcd(e, \lambda(n)) = 1$; i.e., e and $\lambda(n)$ are co-prime. [1]. Note that implies $\forall x \in \mathbb{Z}/n\mathbb{Z} \quad x^{e \times d} = 1$, then (n, e) is released as the public key exponent.
- Compute d the modular multiplicative inverse of e in $\mathbb{Z}/\phi(n)\mathbb{Z}$. This is often computed using the extended Euclidean algorithm. d is kept secret this is the private key of the crytposystem.

(n, e) is the published private key
(n, d) is the private key: it suffice to recover any original message, however

$$d, \lambda(n), \phi(n), p, q \text{ must be kept secret as any of these value is enough to find } d$$

Encryption/Decryption

Alice transmits publishes her public key (n, e) and keeps the private key secret.
Bob then wishes to send message M to Alice. He first turns M into an integer $0 < m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher-text c corresponding to:

$$c = m^e \bmod n$$

Alice can recover m from c by using her private key exponent d by the following computation:

$$m = c^d \bmod n$$

Given m, she can recover the original message M by reversing the padding scheme.

Sign/verify Alice wants to transmit a unencrypted message m and to sign too, then she concatenates her message with the same message encrypt with her private-key then everyone can verify encrypting the signature that was with the message to which it is concatenate.

Cypher/Sign Depending on which exponent is used to do the operation on a public data, this algorithm can either sign a data and everyone can verify it or allow everyone to cypher a data to the private key's recipient.

RSA example

Key generation:

- we choose $p = 61$ and $q = 53$.
- $n = p * q = 3233$ is computed.
- $\phi(n) = \phi(p * q) = \phi(p) * \phi(q) = (p - 1) * (q - 1) = 3120$
- Choosing a number for e leaves you with a single check: that e is not a divisor of 3120. $e = 17$.
- Compute d such that it is the modular multiplicative inverse of e modulo $\phi(n)$: $d = 2753$ since $17 * 2753 = 46801$ and $46801 \bmod 3120 = 1$, this is the correct answer.

[1] here λ is function, many description ignore this part as most of the time Euler's function will provide the same result ... but not always, see Carmichael's totient

The public key is $(n = 3233, e = 17)$.
The private key is $(n = 3233, d = 2753)$.

Encryption/decryption:

- Encryption: let's assume that $m = 65$,
  $c = m^e \bmod n = 65^{17} \bmod 3233 = 2790$
- Decryption: we have received $c = 2790$,
  $m = c^d \bmod n = 2790^{3120} \bmod 3233 = 65$

RSA example CRT version

As $n = pq$ exponentiations mod p and mod q will always be faster than exponentiation mod n, moreover a link (an isomorphism!) exists between $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ and $\mathbb{Z}/n\mathbb{Z}$ via the Chinese remainder theorem -CRT-. Hereafter is an example
We already have chosen $p = 61$, $q = 53$, $e = 17$, $d = 2753$
Encryption: starting from $m = 65$

- $m_p = m \mod p = 4$
- $m_q = m \mod q = 12$
- $\phi(p) = p - 1 = 60$
- $\phi(q) = q - 1 = 52$
- $e_p = d \mod \phi(p) = 17$
- $e_q = d \mod \phi(q) = 17$
- $s_p = {m_p}^{e_p} \mod p = 45$
- $s_q = {m_q}^{e_q} \mod q = 34$
- $p_{-1} = p^{-1} \mod q = 38$
- $q_{-1} = q^{-1} \mod p = 20$
- $s = s_p * q * q_{-1} + s_q * p * p_{-1} \mod n = 2790$

Decryption: starting from $s = 2790$

- $s_p = s \mod p = 45$
- $s_q = s \mod q = 34$
- $\phi(p) = p - 1 = 60$
- $\phi(q) = q - 1 = 52$
- $d_p = d \mod \phi(p) = 53$
- $d_q = d \mod \phi(q) = 49$
- $m_p = {s_p}^{d_p} \mod p = 4$
- $m_q = {s_q}^{d_q} \mod q = 12$
- $p_{-1} = p^{-1} \mod q = 38$
- $q_{-1} = q^{-1} \mod p = 20$
- $m = m_p * q * q_{-1} + m_q * p * p_{-1} \mod n = 65$

Remark on CRT implementation Obviously there are more computations involved in CRT version, but they are much faster.

The computation involving the more resources is exponentiation with the CRT version the speed-up is a time 4.

Additionally many operation can be precomputed once and for all like e_p, e_q, p_{-1}, q_{-1} and stored to be reused each time

Finally the last formula Gauss's recombination is not used in practice.

In the end CRT is a best practice, because of the speed up.

Correctness

We have to prove that $(m^e)^d \bmod n = m$:
first let's remark that the definition of d implies that $ed = 1 + k\phi(n)$ where k is an integer.
$(m^e)^d \bmod n = m^{e \times d} \mod n$
$(m^e)^d \bmod n = m^{1+k\phi(n)} \mod n$
$(m^e)^d \bmod n = m \times m^(\phi(n))^k \mod n$
Assuming that m is co-prime to n we can applies Euler's theorem and: $(m^e)^d \bmod n = m1^k \mod n$
$(m^e)^d \bmod n = m \mod n$

Please note that if m is not relatively prime to n, Euler's theorem can not be applied and further considerations are mandatory.

II .1.3 Cryptographic problems

Asymmetrical cryptography rely on Complexity Theory which is a branch of the theory of computation in theoretical computer science and mathematics that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other.

What interests cryptographer very asymmetrical problems : problems very easy to do in a way but extremely difficult to do in the other way.

A typical example of asymmetrical problem is the factorization: given two numbers it is very easy to compute their product but, asymptotically, it is extremely difficult to factorize a product. This in this precise problem with two primes factor that relies the security of RSA.
Definition:

- Let n product of two prime numbers p and q what is called the factorisation problem is: knowing n find p and q such that $n = pq$.
- To break RSA means if (n, e) be a RSA public key to find the private key: d such that $ed = 1 \mod \phi(n)$.

Propositions:

Knowing a RSA public key and a RSA private key then we can know the factorization of n. And we saw that thanks to the extended Euclidean algorithm if we know a factorization of n it is easy to find the private key.

So, breaking RSA is equivalent to solve the factorization problem.

The RSA problem Definition:

Let (n, e) be a RSA public key what is called the RSA problem is :

knowing c co-prime to n, to find m such that $c = m^e \mod n$.

It means to be able to find one plain-text from it's cypher-text, without finding the private key.

The following assertion is not proven yet:
If someone can resolve the RSA problem many times then he break RSA. And then maybe that the RSA problem is easier than the factorization problem.

II .2 Asymmetrical implementation in smart cards

Notation: In the following part of the text:

- $(\mathbb{G},\ \times)$ is a commutative group
- x, y and z three elements belonging to this group
- n an positive integer
- $ln_b(x)$ is the Neperian logarithm in base b of x *i.e.* $ln_b(x) = ln(x)/ln(b)$

Without an explicit mention those algorithms work for every commutative group [2] $\mathbb{G}$. Note that is was not always possible to say 'the improvement of this algorithm lies in the multiplication only', so be careful. By example the Infineon technologies ZDN algorithm is an algorithm improving at the same time, multiplication and reduction.

II .2.1 Number representations

This section will be mainly dedicated to the different ways to represent numbers *i.e.* integers or relative numbers. Plan id the following:

- Classical b-arry representations
- Some binary representations
- Non adjacent representations

AIM: Change the representation of a an integers or relative numbers for a new one. What will be obtained might be a less simple, or might requires more amount of digits but will be but richer in term of algebraic/computational proprieties(s).

Definition. Canonical vs Non-Canonical representation

- 'canonical' is used to indicate a standard way to represent a mathematical object under the form of a mathematical expression. This a particular convention that was accepted as standard among an important number of possible conventions.
- 'non-canonical' by opposition means that the convention used to to represent a mathematical object under the form of a mathematical expression is not he standard one, therefore this expression is meaningless unless convention to which it is referring is specified.

Example:
In geometry, the canonical way to represent objects is referring to a Cartesian coordinate system system and no-one wonder what type of mathematical object represent the following expression $(x-h)^2+(y-k)^2=\rho^2$, (circle centered on (h,k) and of radius $|\rho|$).

[2] On of the first property of commutative, or Abelian groups, is to give a sense to Binomial theorem and its generalization (multinomial theorem)

II .2.1.1 Mathematical notation

Two way to interpret Euclidean division:

Definition. Euclidean division:
Given a, b positive integers there exist a unique couple of integers (q, r)
such that decompose uniquely a in the famous Euclidean form.
i.e. Given $a, b \in \mathbb{N}$ $\exists!(q, r) \in \mathbb{N}^2$, such that:

$$a = b \times q + r \text{ with } 0 \leq r < b$$

As a consequence each number, can be decomposed the following way:
$n = 2^k \times b + r$ with $0 \leq r < 2^{k-1}$.
$n = 2^k \times b + r$ with $-2^{k-1} < r \leq 2^{k-1}$

The first equality is the canonical one, it allows representation in any b-basis with positive digits, whereas the second one, not canonical, enable representations in any binary basis with positive and negative digits but with smaller absolute values -Useful for ECC, RSA with NAF exponent-.

Iterating the division process on a from the highest power of b to 0, a link is naturally created between polynomial in b and representation in basis b.

Possible properties of b-basis representations
* position relativity: representation in which the localization of a digit matters
* completeness for $\mathbb{S}$: each element of a set of number $\mathbb{S}$ can be represented
* uniqueness for $\mathbb{S}$: each element of $\mathbb{S}$ admits only one representative.
* homogeneity: representation in which all operations are performed in the same way.
* optimality: representation in which a propriety is minimized -Length, Hamming weight, many others- among a larger family of representation.

Remarks:
* For positional representation system: the basis is always written the same manner, 10, consequence of link between b-arry representation/polynomial expansion of variable b:
$10 = 10_{10}$, $2 = 10_2$, $b = 10_b$, $-b = 10_{-b}$
* Interesting properties are the span and shape of $\mathbb{S}$:
is this including negative number? is this a perfect, quite, strongly not, symmetrical span?
* In the following section, we always have $a \in \mathbb{S}$

Notation:
* $b \in \mathbb{N}^*$, basis will be b or $-b$
* $\mathcal{D}$ is the digit space.
* t number of bits of the considered architecture

II .2.1.2 Classical b-arry representations

Here are approached some classical general representations.

- Unary notation: representation with $b = 1$, *aka* 'Peano Unary system'.
  Simplest way to represent integers numbers...
  For that representation to include 0 we impose digit space to admits two elements
  /!\ Formal -but non consistent with the general one- scripture

$$a_1 = \sum_{i=0}^{t-1} a_i \text{ and } \forall i, a_i \in \mathcal{D} = \{0, 1\}$$

* completeness for $\mathbb{S} = [\![0, t]\!]$
* no uniqueness neither a positional system
* the value 0 can only be implicitly represented by an empty digit string
* addition is in fact concatenation, homogeneous

- b-arry notation: general canonical representation in base b, for $b \in \mathbb{N}\backslash\{0, 1\}$:
starting from the least significant digit, number is obtained via a polynomial in b which coefficients are the digits of the number representative.

$$a_b = \sum_{i=0}^{t-1} a_i \times b^i \text{ and } \forall i, a_i \in \mathcal{D} = \{0, ..., b-1\}$$

* unique representation for $\mathbb{S} = [\![0, b^t - 1]\!]$
* homogeneity

- Signed b-arry notation: representation in base $b \in \mathbb{N}\backslash\{0, 1\}$, *aka* 'Sign and Magnitude in base b'

$$a_{b,signed} = a_t \sum_{i=0}^{t-1} a_i \times b^i \text{ and } \forall i, a_i \in \mathcal{D} = \{0, ..., b-1\}$$

* completeness for $\mathbb{S} = [\![1 - b^{t-1}, b^{t-1} - 1]\!]$
* non unique representation + 0 and -0
* non homogeneous system !
* Simple to implement
* Symmetrical span

- Nega b-arry notation: canonical representation in base $-b$ for $b \in \mathbb{N}\backslash\{0, 1\}$:

$$a_{-b} = \sum_{i=0}^{t-1} a_i \times (-b)^i \text{ and } \forall i, a_i \in \mathcal{D} = \{0, .., b-1\}$$

* completeness for a $\mathbb{S} = [\![-b\frac{b^t-1}{b+1}, \frac{b^t-1}{b+1}]\!]$
* completeness for $\mathbb{S}$ without bit sign !
* uniqueness for $\mathbb{S}$
* although arithmetic operations are more complicated, homogeneity of operations !
* /!\ $\mathbb{S}$ not centered on 0, strongly asymmetrical span /!\

- One's complement in base b

$$a_{b,1*} = (b^t - 1)_b - a_b$$

* except MSB, each bit is a power of b

- Two's complement in base b

$$a_{b,2*} = b_b^t - a_b$$

* except MSB, each bit is a power of b, except for the most significant bit,
whose weight is the negative of the corresponding power of b.

II .2.1.3 Some binary representations

- Binary notation: canonical representation base 2:

$$a_2 = \sum_{i=0}^{t-1} a_i \times 2^i \text{ and } \forall i, a_i \in \mathcal{D} = \{0, 1\}$$

* Homogeneity
* Complete for $\mathbb{N}$
* Uniqueness for $\mathbb{N}$

- Signed binary: canonical representation in base 2, *aka* 'Sign and Magnitude in base 2':

$$a_{2,signed} = -1^{a_{t-1}} \sum_{i=0}^{t-2} a_i \times (2)^i \text{ and } \forall i, a_i \in \mathcal{D} = \{0, 1\}$$

+ Simple to implement.
+ Useful for floating point representation.
- Sign bit independent of magnitude
- can be both + 0 and -0 (Makes math hard to do).

- One's complement canonical representation

$$a_{1*} = (2^t - 1)_2 - a_2 = \overline{a_2},$$

Except the MSB, each bit is a power of 2
The MSB is seen as the negative of $(2^t - 1)$.
And with a written as $a_{1*} = \{a_{t-1}, a_{t-2}, ..., a_1, a_0\}$

$$a = -a_{t-1}(2^t - 1) + \sum_{i=0}^{t-2} a_i 2^i$$

* completeness for $\mathbb{S} = [\![1 - b^{t-1}, b^{t-1} - 1]\!]$
* non uniqueness: two zeros $a_i = 0$ and $a_i = 1$
* Feature: +0 and -0 return TRUE when tested for zero, FALSE when tested for non-zero.

- Two's complement canonical representation

$$\text{If } a \geq 0\text{:} \quad a_{2*} = a_2 \quad \text{If } a \leq 0\text{:} \quad a_{2*} = 2_2^t - a_2 = \overline{a_{2_{Signed}}} + 1$$

Except the MSB, each bit is a power of 2
The MSB is seen as the negative of the corresponding power of 2.
And with a written as $a_{2*} = \{a_{t-1}, a_{t-2}, ..., a_1, a_0\}$

$$a = -a_{t-1}2^t + \sum_{i=0}^{t-2} a_i 2^i$$

* completeness for $\mathbb{S} = [\![-b^{t-1}, b^{t-1} - 1]\!]$
* homogeneity
* there is one zero only.
* multiplying 2's complement operands takes just about same amount of hardware as multiplying unsigned operand

- Nega binary: canonical representation in base -2:

$$a_{-2} = \sum_{i=0}^{t-1} a_i \times (-2)^i \text{ and } \forall i, a_i \in \mathcal{D} = \{-1, 0\}$$

* completeness for $\mathbb{S} \subset \mathbb{Z}$
* /!\$\mathbb{S}$ not centered on 0, strongly asymmetrical span/!\ * basic routines requires some efforts.

- Binary Signed Digit representation:
Here is relaxed on condition on the digit of the representation, leading to non canonical representation. Here they have value in $\{-1, 0, 1\}$, but the formal decomposition remain the same:

$$a_{-1,0,1} = \sum_{i=0}^{t-1} a_i \times 2^i \text{ with } a_i \in \mathcal{D} = \{-1, 0, 1\}$$

The consequence it that number are not uniquely decompose: this is not canonical.
On the other hand this additional degree of liberty allow to choose between decompositions available.

- Canonical Binary Signed Digit representation: *aka* 'Binary-NAF'
  And precisely, without ambiguity, its name is '2-width Binary NAF' [3]
  Here is relaxed on condition on the digit of the representation, leading to non canonical representation. Here they have value in $\{-1, 0, 1\}$, but if the formal decomposition remain the same, one additional condition namely 'no adjacent 1' lead to one representation.

$$a_{2-NAF} = \sum_{i=0}^{t-1} a_i \times 2^i \text{ with } a_i \in \mathcal{D} = \{-1, 0, 1\}$$
$$\forall i \in [\![0, t-1]\!], a_i \times a_{i+1} = 0$$

- Binary Alternating Greedy expansion this is a signed binary expansion, with the property that consecutive non zero digit -even if separated by some 0- are opposed. useful in several left-to-right algorihms
- Zeckendorf representation:
  A number can be decomposed in a unique way as a sum of Fibonnaci numbers, in which there is no two consecutive Fibonnaci number used. There exist a signed digit Zeckendorf representation.
  $31 = 1010010_F = F_8 + F_6 + F_3$

II .2.1.4 Basic in geometry of numbers

Recall: finite field definition

The two following definitions are the start-point of a mathematical theory called Algebraic Theory of Number, also-known-as 'geometry of numbers'.

Definition. Algebraic number, algebraic integer

$\overline{\mathbb{Z}} := \{x \in \mathbb{C} / \exists P \text{ unitary}, P \in \mathbb{Z}[X]^* / P(x) = 0\}$, ring of algebraic integers

$\overline{\mathbb{Q}} := \{x \in \mathbb{C} / \exists P \in \mathbb{Z}[X]^* / P(x) = 0\}$, field of algebraic numbers

$\theta \in \overline{\mathbb{Q}}$ it is the root of many $P \in \mathbb{Z}[X]^*$, among them P can be chosen irreducible, then the degree of P, d is called algebraic degree of θ. We define

$$\mathbb{Q}[\theta] := \{x \in \mathbb{C} / \exists P \in \mathbb{Z}[X] \text{ and } x = P(\theta) \}$$

This is a $\mathbb{Q}$-vector space of dimension d, a basis is $\{1, \theta, \theta^2, \ldots, \theta^{d-1}\}$

Definition. 'ring of integers an algebraic number field'
Let $\mathbb{K}$ be a field of algebraic number *i.e.*$\mathbb{K} = \mathbb{Q}[\theta]$.

a ring can be defined, $\mathcal{O}_{\mathbb{K}} = \overline{\mathbb{Z}} \cap \mathbb{K}$. Note that the field of fraction of the ring $\mathcal{O}_{\mathbb{K}}$ is $\mathbb{K}$.

Proposition. *Quadratic field*
For $d = 2, \mathbb{Q}[\theta]$ is snoted sliglty differently; $\mathbb{Q}[\theta] =$

$$\mathbb{Q}[\sqrt{\theta}] := \{x \in \mathbb{C} / x = a + b \times \theta \text{ with } a, b \in \mathbb{N}\}$$

[3] the width refers to the number among which only one will be allowed to be a non zero.

and is named a quadratic field, according to the sign of Δ_P, we be talked about 'Imaginary quadratic field' ou 'Real' quadratic field'.

Theorem. *'of classical imaginary quadratic field'*
Let be $\mathbb{Q}[\sqrt{d}]$ for some $d \in \mathbb{Z}$, square free additionally let's assume that $d < 0$, then can explicit $\mathcal{O}_{\mathbb{K}}$:

$$\mathcal{O}_{\mathbb{K}} = \mathbb{Z}[\alpha] \text{ with } \alpha = \sqrt{d} \text{ if } d = 2, 3 \mod 4 \text{ and } \alpha = \frac{1+\sqrt{d}}{2} \text{ if } d = 1 \mod 4.$$

Illustrations:

d	-1	-2	-3	-7	-11
θ	$1+\sqrt{-1}$	$\sqrt{-2}$	$\frac{3+\sqrt{-3}}{2}$	$\frac{1+\sqrt{-7}}{2}$	$\frac{1+\sqrt{-11}}{2}$
$\mathbb{K} = \mathbb{Q}[\theta]$	$\mathbb{Q}[1+\sqrt{-1}]$	$\mathbb{Q}[1+\sqrt{-1}]$	$\mathbb{Q}[\frac{1+\sqrt{-3}}{2}]$	$\mathbb{Q}[\frac{1+\sqrt{-7}}{2}]$	$\mathbb{Q}[\frac{1+\sqrt{-11}}{2}]$
$\mathcal{O}_{\mathbb{K}} = \mathbb{Z}[\alpha]$	$\mathbb{Z}[\sqrt{-1}]$	$\mathbb{Z}[\sqrt{-2}]$	$\mathbb{Z}[\frac{1+\sqrt{-3}}{2}]$	$\mathbb{Z}[\frac{1+\sqrt{-7}}{2}]$	$\mathbb{Z}[\frac{1+\sqrt{-11}}{2}]$

Table II .1: θ chosen as non zero, non unit with $|\theta| > 1$

II .2.1.5 Non Adjacent Forms

By opposition to unsigned digit notation, here *increase the size of the representation* can induce lower hamming weight and as a consequence a *faster exponentiation.*

What's in the literature? Groscot [1984]

Joye et al. [1997] first part: present binary NAF and a natural converters. key paper.

Joye and Tymen [2001] introduce the compactification of a binary NAF representation, leading a n-bit number to be represented as 2-NAF by n+1 bit [4] - key paper.

Muir [2004] internship about NAF techniques and generalisation - NADS & JDS-

Heuberger and Krenn [2012] provide a bit more maths background for the complex NAF case

Blake et al. present complex NAF and give example for the following cases:

$$\tau = 1 + \sqrt{-1} \quad \tau = \sqrt{-2} \quad \tau = \frac{3+\sqrt{-3}}{2} \quad \tau = \frac{1+\sqrt{-7}}{2} \quad \tau = \frac{1+\sqrt{-11}}{2}$$

Aranha et al. [2011] Was the fastest scalar multiplication for few months in 2011.

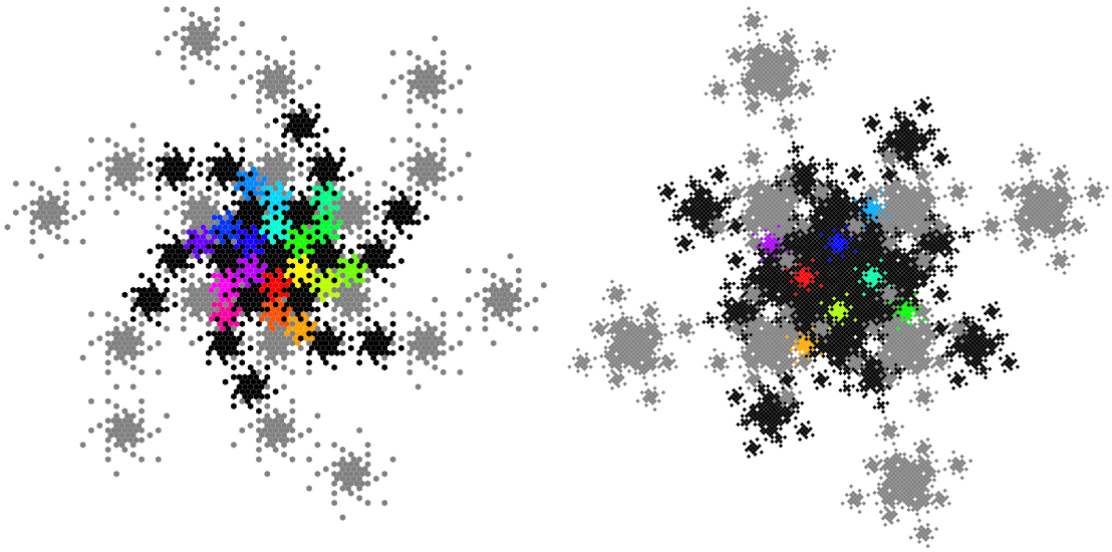


Figure II .1: 3-NAF on the left, 4-NAF on the right - for different $\tau \in \mathbb{C}$

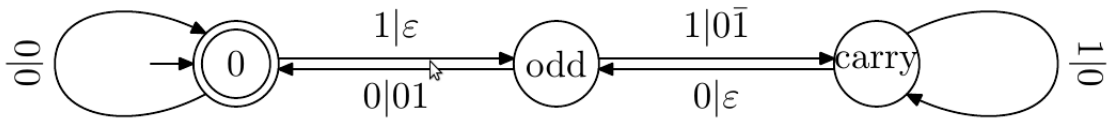
- Binary Non Adjacent Form, 2-adic NAF form, 1-width 2-adic NAF form:
  Condition on the digits are relaxed allowing the coordinate -1 in addition to 0 and 1:

$$\mathcal{D} = \{-1, 0, 1\}$$

'NAF' meaningless vs ' 1-width 2-adic NAF' meaningful
'1-width' refers the maximum value in $\mathcal{D}$,
'2-adic' refers to base's expansion, 2

Theorem (Reitwiesner' 1960)**.** *For a $a \in \mathbb{N}$, there exits a unique signed binary expansion noted, $(a_i)_{0 \leq i \leq t-1}$, called the 1-width 2-adic NAF form of a such that:*

$$a = \sum_{0}^{t-1} a_1 \times 2^i, \text{ with } a_i \in \mathcal{D} = \{-1, 0, 1\}$$
$$\forall i \in [\![0, t-1]\!], a_i \times a_{i+1} = 0$$



$$23_{10} = \{ \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad \}_2$$
$$23_{10} = \{ \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad \text{-}1 \quad \}_2$$
$$23_{10} = \{ \quad 1 \quad 0 \quad \text{-}1 \quad 0 \quad 0 \quad \text{-}1 \quad \}_2$$

Binary to binary 1-NAF definitions are not incompatible:

$$42_{10} = \{ \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad \}_2$$

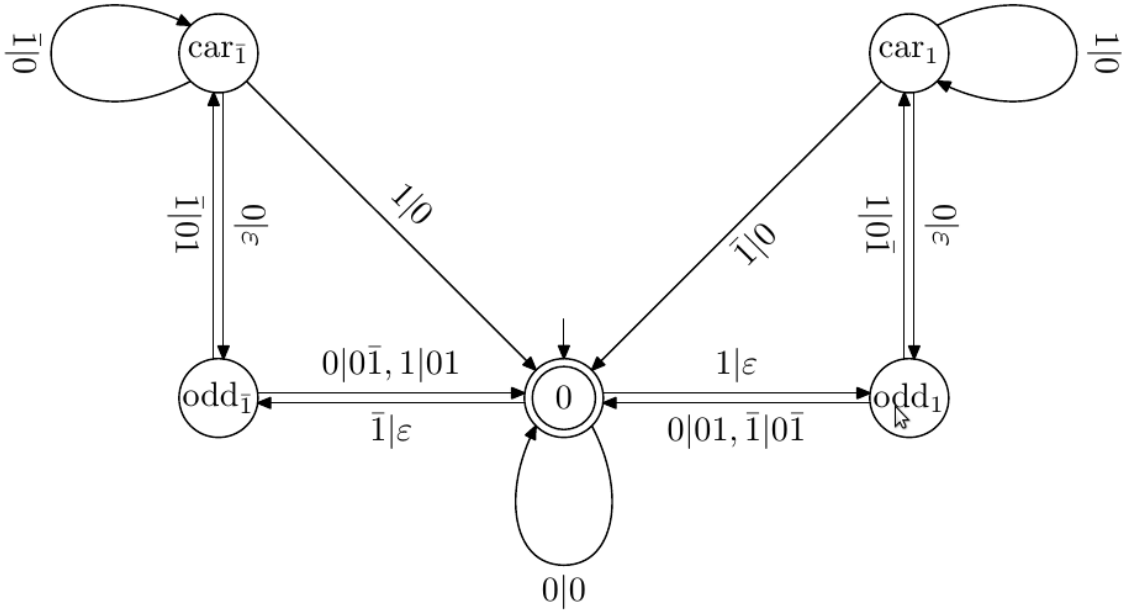


Figure II .2: Visual proof for previous affirmation

Definition. Optimality for a given digits space : An expansion of a is said to be optimal, if it minimizes the weight among all the of a with digits out of $\mathcal{D}$.

Theorem (Reitwiesner' 1960)**.** *binary-NAF of integers is optimal among all expansion using the following digits* $\mathcal{D} = \{0, \pm 1\}$.

Theorems:

- 2-adic NAF representation have , at most, one digit longer than its a binary representation.
- Each integer has a unique 2-adic NAF representation.

1-width 2-adic NAF form: ECC vs RSA

RSA case: 'Square-and-Multiply' has to be changed to 'Square-and-Multiply-or-Divise' and the value $x^{-1} \mod n$ shall be precomputed.

ECC case: 'Double-and-Add' algorithm should be changed to 'Double-and-Add-or-Substract' and the value $-P$ should be precomputed.
But here is a trick : in many ECC implementation compute the opposite of a point is very easy: in this case: binary NAF applied to ECC saved a pre-computed value by comparison to RSA the same representation applied to RSA.

[4] bit and not digit !

- Extended Binary NAF or k-width 2-adic NAF, canonical representation
  extension of the 2-NAF: the base is still $\tau = 2$, but the digit space is increased ...

$$a = \sum_{0}^{t-1} a_I \times 2^i, \text{ with } a_i \in \mathcal{D} = \{0, \pm 1, \pm 3, .., \pm 2^{k-2} - 1\}$$
$$\forall i \in [\![0, t-1]\!], \text{ Among } k \text{ consecutive digits max one of them is non-zero}$$
$$\text{Each non zero digit is odd.}$$

  Theorems:

  - k-width 2-adic NAF representation, at most, one digit longer than its binary representation.
  - Each integer has a unique k-width 2-adic NAF.

  Theorem: Avanzi, Muir & Stinson' 2004
  With $\tau = 2$, $k \geq 2$, extended binary NAF of each integer is optimal for $\mathcal{D} = \{0, \pm 1, \pm 3, .., \pm 2^{k-2}\}$.

General converter The following routine returns the ω-width binary NAF of a binary

Algorithm 2: function $f_\omega(n)$

Input: $n \in \mathbb{N}$
Output: $n \in \mathbb{N}$
if $n = 0 \mod 2$ **then**
  return $n/2$
$r \leftarrow -2^{k-1} < r \leq 2^{k-1}$ with $n = b \times 2^k + r.$;
return $n \leftarrow (n - r)/2^\omega$

Algorithm 3: function $g_\omega(n)$

Input: $n \in \mathbb{N}$
Output: $n \in \mathbb{N}$
if $n = 0 \mod 2$ **then**
  return 0
$r \leftarrow -2^{k-1} < r \leq 2^{k-1}$ with $n = b \times 2^k + r.$;
$0^{k-1}r$

Algorithm 4: Binary to ωNAF converter

Input: $n \in \mathbb{N}$
Output: a string of digits
$\alpha \leftarrow$ " **while** $n \neq 0$ **do**
  $\alpha \leftarrow g_\omega(n) \,||\, \alpha$;
  $n \leftarrow f_\omega(n)$
return α

examples
$g(2)(n)$ will take the values: $0, 01, 01, 01$
$f(2)(n)$ will take the values: $42, 21, 5, 1, 0$
then the 2NAF form of 42 is 0101010_2 meaning $42_{10} = 2^5 + 2^3 + 2^1$.

$g(3)(n)$ will take the values: $0, 00 - 3, 003$
$f(3)(n)$ will take the values: $42, 5, 1, 0$
then the 3NAF form of 42 is $300 - 30_2$ meaning $42_{10} = 3 \times 2^4 - 3 \times 2^1$.

$g(4)(n)$ will take the values: $0, 00 - 3, 003$
$f(4)(n)$ will take the values: $42, 21, 3, 0$
then the 4NAF form of 42 is $300 - 30_2$ meaning $42_{10} = 3 \times 2^4 - 3 \times 2^1$.

$g(5)(n)$ will take the values: $0, 0000 - 11, 0002$
$f(5)(n)$ will take the values: $42, 21, 1, 0$
then the 5NAF form of 42 is $000010000 - 10_2$ meaning $42_{10} = \times 2^7 - 11 \times 2^1$..

k-width 2-adic NAF form: ECC vs RSA ... example with $k = 3$

RSA case: 'Square-and-Multiply' has to be changed to 'Square-and-Multiplyor-Divise'
and the values $\{-1, \pm 3\}$ shall be precomputed.

ECC case: 'Double-and-Add' algorithm should be changed to 'Double-and-Add-or-Substract'
and only the value $3P$ should be precomputed, in the case of a curve giving cheap inversion.

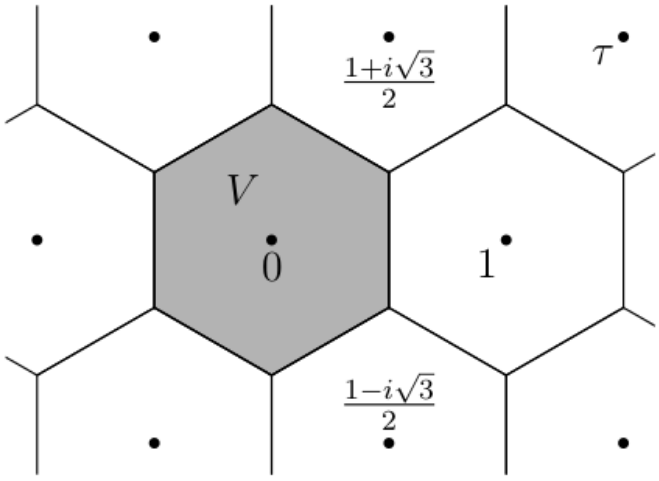
- Real Non-Binary based NAF: Non-Binary NAF and Non-Binary k-NAF:
  Here simply the base τ is not two any more, but another prime.
- Complex NAF and k-NAF form:
  This time is $\tau \in \mathbb{C}^*$ and is algebraic integer over $\mathbb{Z}$ of degree 2 [5] with $|\tau| > 1$ and $k \geq 2$

  Definition. *Zero & local Voronoi Cells:*
  The Voronoi cell at origin is :

$$V := \{z \in \mathbb{C} : \forall y \in \mathbb{Z}^*[\tau], \|z\| \leq \|z - y\|\}$$

  By definition the Voronoi Cell for the point $u \in \mathbb{Z}[\tau]$ corresponding to set $\mathbb{Z}[\tau]$ where u is then called centre of V_u, or the lattice point corresponding to V_u, is such as:

$$V_u := \{z \in \mathbb{C} : \forall y \in \mathbb{Z}[\tau], \|z - u\| \leq \|z - y\|\}$$



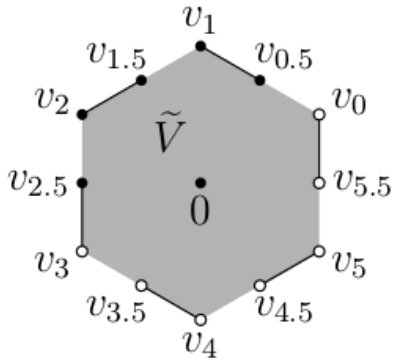
Voronoi cell V for 0 corresponding to the set $\mathbb{Z}[\tau]$ with $\tau = \frac{3}{2} + \frac{1}{2}\sqrt{-3}$.

Remark that the limit case given by equation of the definition $\|z\| = \|z - y\|$ share the plan in two via the bisection of segment $[zy]$ which fully make sense when z is an extremal point.

[5] *i.e* root of some $x^2 - p \times x + q \in \mathbb{Z}$[x]

... From now on $\mathcal{D}$ is assumed to be a 'Restricted Voronoi Cell'

Definition. *Restricted Voronoi Cell:*
Previous definition was unclear about to which cell borders -vertices + edges- belong to. Each edge is split in two equal part, defining three points of each of them, defining twice more points and portions of line than there was of vertices and edges, then those elements are fairly shared (segment , points) on each side of the origin. This new cell is called 'Restrictive Voronoi Cell', and noted $\widetilde{V}$



Restricted Voronoi cell $\widetilde{V}$ for 0 corresponding to the set $\mathbb{Z}[\tau]$ with $\tau = \frac{3}{2} + \frac{1}{2}\sqrt{-3}$.

Figure II .3: *if $stuff \in V_u$ then its symmetrical don't.*

... From now on $\mathcal{D}$ is assumed to be a 'Reduced Residue Digit Set'

Definition. Reduced Residue Digit Set
For $\mathcal{N}(\tau^k) \leq 12$, we note

$$R = \{x \in \mathcal{O}_{\mathbb{K}} \text{ such that } \tau \not| x\}$$

The set $\mathcal{D} \subseteq \mathbb{Z}[\tau]$ is called 'reduced residue digit set modulo τ^k', if it consist of 0 and exactly one representative for each residue class of $R \mod \tau^k$ that is not divisible by τ.

More precisely:
if a class contain a unit u then: $\mathcal{D} := \mathcal{D} \cup \{u\}$
if a class do not contain a unit: $\mathcal{D} := \mathcal{D} \cup \{u \text{ such that } |u| \leq \mathcal{N}(\tau^k)\}$

Definition: *RDS k-Width τ-adic Non Adjacent Form* or *k-Width NADS:*
Let's $\eta = (\eta_j)_j \subset \mathcal{D}^{\mathbb{Z}}$. The representation η is called k-Width τ-adic Non Adjacent Form if each factor $\eta_{j+k-1}...\eta_j$, *i.e.* each block of length k contains, at most, one non-zero digit.

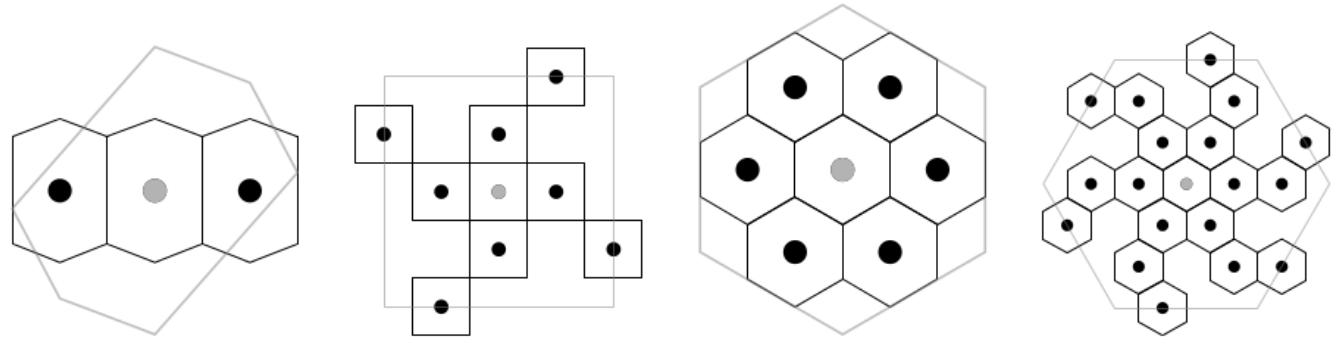
Theorem

Let's $k > 2$, $\mathbb{K} = \mathbb{Q}[\tau]$ and $\mathcal{D}$ be a k-Width NADS and
we have $\forall x \in \mathcal{O}_{\mathbb{K}}$, $\exists!$ RDS k-Width τ-adic NAF for x.

Remark: in certain cases the condition $k > 2$ can be relaxed:
for $\tau = \frac{3+\sqrt{-3}}{2}$, $k > 1$
for $\tau = \frac{1+\sqrt{-7}}{2}$, $k > 1$
for $\tau = \frac{1+\sqrt{-11}}{2}$, all k

... From now on element of $\mathcal{D}$ are assumed to be a 'Representatives of Minimal Norm'

Definition. Representative of Minimal Norm
Let τ be an algebraic integer, imaginary quadratic and let $\eta \in \mathbb{Z}[\tau]$ be not divisible by τ.
Then if $\eta \in \tau^k \times \tilde{V}$ is called 'Representative of Minimal Norm'



(a) Digit set for $\tau = \frac{1}{2} + \frac{1}{2}\sqrt{-7}$ and $w = 2$.

(b) Digit set for $\tau = 1 + \sqrt{-1}$ and $w = 4$.

(c) Digit set for $\tau = \frac{3}{2} + \frac{1}{2}\sqrt{-3}$ and $w = 2$.

(d) Digit set for $\tau = \frac{3}{2} + \frac{1}{2}\sqrt{-3}$ and $w = 3$.

- Minimal norm representatives digits set modulo τ^k for several τ and k.
- For each τ, V_τ is drawn; the large cell is $\tau^k \times V$.
- the situation c) verify the previous definition

... From now on element of $\mathcal{D}$ are assumed to be a 'Minimal Norm Representative Digit Set'
Definition - *Minimal Norm Representative Digit Set:*
Let τ be an algebraic integer, imaginary quadratic and let $\mathcal{D}$ be reduced residue digit set modulo τ^k consisting of representative of minimal norm if its residue class, then $\mathcal{D}$ is called 'Minimal Norm Representative Digit Set'.

.... And here it is:

Definition *k-Width τ-adic Non Adjacent Form:*
Let's $\eta = (\eta_j)_j \in \mathcal{D}^{\mathbb{Z}}$. The representation η is called k-Width τ-adic Non Adjacent Form if each factor $\eta_{j+k-1}...\eta_j$, *i.e.* each block of length k contains, at most, one non-zero digit.

Theorem. *For each lattice point $x \in \mathbb{Z}[\tau]$ and $\mathcal{D}$ be a k-Width MNR we have $\forall x \in \mathbb{Z}[\tau]$, $\exists!$ MNR k-Width τ-adic NAF for x.*

Theorem: Avanzi, Muir & Stinson' 2004
In the complex case with $\tau = 2$, $k \in \{2, 3\}$, the τ^k-NAF is optimal.

Theorem: Heuberger' 2010
In the complex case with $k \in \{4, 5, 6\}$, the τ^k-NAF is NOT optimal.

Theorem: Krenn' 2011
In the following cases: $k \geq 4$, $\|p\| \geq 3$ or $k = 3$, $\|p\| \geq 5$
In the complex case with $k \in \{4, 5, 6\}$, the τ^k-NAF is optimal.

Example: binary to binary NAF conversion
An intuitive algorithm without pretending to any efficiency is 'starting form LSB if two consecutive non zeros digit are meet change the smallest for -1 and propagate the change.

The process will always terminates with a valid NAF representation of n, process start and begin at position entirely determined by n_2, noting that the it is a fully reversible process.

Important remark: Strictly speaking the difference between a Square & Multiply and a 2^3-arry method is only a change in the representation from base 2 to base 2^3, and using the same exponentiation algorithm.

II .2.1.6 Illustration of complex NAF:

This section is dedicated to give an basically meaningful example of use of complex NAF, the given example is for primary field, then

Let assume P is a point on the hereafter defined curve$\mathbb{E}$, and let n and integer and nP to be computed.

$$\mathbb{E}({\mathbb{F}_5}^m) : y^2 = x^3 - x + 2 \text{ over } {\mathbb{F}_5}^m$$

How many point on the curves ${\mathbb{E}({\mathbb{F}_5}^m)}$? depending on m
How many point on the curves $\mathbb{E}(\mathbb{F}_5)$? let's see

x	x^3-x+2	y	y^2	
0	2	0	0	$\mathcal{O}$
1	2	1	**1**	$(3,\pm1)$
2	3	2	4	$(3,\pm4)$
3	**1**	3	4	
4	2	4	**1**	

Figure II .4: $\mathbb{E}(\mathbb{F}_5) = \{\mathcal{O}; (3, \pm 1)\}$

Then we define the Frobenius isomorphism as usual:

$$\sigma = \begin{cases} \mathbb{E} \longrightarrow \mathbb{E} \\ (x, y) \mapsto (x^5, y^5) \end{cases}$$

Then according to REFERENCE, Frobenius morphism has for characteristic polynomial: $\Phi^2 - a \times \Phi + q$ where $a = q + 1 - \mathbb{E}(\mathbb{F}_q)$ and here $q = 5$ That is to say $\Phi^2 - 3 \times \Phi + 5$

The previous decomposition lead us to use representation in $\mathbb{Q}[\sqrt{-11}]$

Avoid weakness on ${\mathbb{E}({\mathbb{F}_5}^m)}$, m should not be stupidly chosen.

Then ω-width τ-adic NADS is prepared, with $\tau = \frac{1+\sqrt{-11}}{2}$, According to the value of ω, the digit space of $\mathbb{D}$ is defined and ω-width τ-adic NADS conversion algorithm is defined.

1. Check that the considered representation exist
2. Convert n: compute $a + b \times \tau$ such that $n \equiv a + b \times \tau \mod (\tau^m - 1)$
   this trick has a name!
   since $(\tau^m - 1) \times P = \mathcal{O}$, we have $n \times P = (a + b \times \tau)P$
3. Convert $a + b \times \tau$ to ω-width τ-adic NADS
$$a + b \times \tau = \sum_{i=0}^{s} c_i \tau^{k_i}$$
4. for each $c \in \mathbb{D}$ pre-compute $Q_c = c \times P$
5. Exponentiation is done trough a Horner scheme:
   $(a + b\tau) \times P = \tau^{k_1}(\tau^{k_2-k_1}(...(\tau^{k_s-k_{s-1}} \times Q_{c_s} + Q_{c_{s-1}}) + ...+) + Q_{c_1}) + Q_{c_0}$

For binary fields, using normal basis, the cost of 'a Frobenius' is a shift

II .2.2 Group representations

II .2.2.1 Chinese Theorem of Remainders:

Elements of some finite fields $\mathbb{Z}/n\mathbb{Z}$, where n is a product of primes

- Canonical representation in $\mathbb{Z}/n\mathbb{Z}$,
  4 times slower tan CRT to achieve an RSA exponentiation.
- Uses of Chinese theorem of remainders, smaller number fastening the computations. Require a recombination step: Gauss, Garner, ...

  VS side channel cryptanalysis
  Because of the recombination step, that can be side channel analysed or even perturbed, the CRT implementation bring also some potential weaknesses.

Algorithm 5: Gauss recombination

Input: $p, q, S_p, S_q, q^{-1} \mod p, p^{-1} \mod q$
Output: $S = m^d \mod n$

$S \leftarrow S_p \times q \times (q^{-1} \mod p) + S_q \times p \times (p^{-1} \mod q)$;
return S

Remark: very natural but terribly slow as two modular inversions are required!

Algorithm 6: Unprotected Garner algorithm

Input: $p, q, S_p, S_q, q^{-1} \mod p$
Output: $S = m^d \mod n$

$t \leftarrow S_p - S_q$;
if $t < 0$ **then**
  $t \leftarrow t + p$;
$t' \leftarrow t \times (q^{-1} \mod p)$;
$S \leftarrow S_q + t' \times q$;
return S

Algorithm 7: Non conditional Garner algorithm

Input: $S_p, S_q, p, q, q^{-1} \mod p$
Output: $S = m^d \mod n$

$t_0 \leftarrow S_p - S_q$;
$t_1 \leftarrow t_0 + p$;
if $t_0 < 0$ **then**
  $t \leftarrow t_1$;
if $t_0 > 0$ **then**
  $t \leftarrow t_0$;
$t' \leftarrow t \times (q^{-1} \mod p)$;
$S \leftarrow S_q + t' \times q$;
return S

II .2.3 Multiplications

Beware that this is en attempt to separate each type operation -addition/soustration; multiplication; reduction; exponnetiation- whereas modern algorithm interleaves this operations ....

Algorithm 8: Non conditional DPA-protected Garner algorithm

Input: $S_p, S_q, p, q, q^{-1} \mod p$
Output: $S = m^d \mod n$

$t_0 \leftarrow S_p - S_q$;
$t_1 \leftarrow t_0 + p$;
if $t_0 < 0$ **then**
  $t \leftarrow t_1$;
if $t_0 > 0$ **then**
  $t \leftarrow t_0$;
$t' \leftarrow t \times (q^{-1} \mod p)$;
$S' \leftarrow S_q + t' \times (q + R)$;
$S \leftarrow S' \mod N$;
return S

AIM: realize the following operation: compute $x \times y$ in the group G, where G is a general group. In some groups special algorithm can speed up multiplication algorithm, thanks to their rich algebraic structure and property: this is the case of 'some' finite fields with Froebinius's isomorphism.

- Recursive method -*Neanderthal*- $-\infty$
  Multiply two numbers thanks to the definition of the multiplication relying on the one of the addition.
  Complexity: $\mathcal{O}(n \times 2^n)$.
- Knuth's schoolbook method -*Neanderthal adult*- This is the 0%
  old method that is learnt in elementary school, also know as the Shift-And-Add method. Note that the famous Andrei Kolmogorov stated that this method was optimal!
  Complexity: $\mathcal{O}(n^2)$.
- Gauss trick -*1852*- 0%
- Quarter square method -*Babylonian, 2000 BC*- 0%
  This method suppress multiplication for squarring, and is applcable since division by 4 is allowed and p is even.

  $$\frac{(x+y)^2}{4} - \frac{(x-y)^2}{4} = \frac{1}{4}[(x^2 + 2xy + y^2) - (x^2 - 2xy + y^2)] = xy$$

  Or taking advantage of the parity:

  $$\lfloor \frac{(x+y)^2}{4} \rfloor - \lfloor \frac{(x-y)^2}{4} \rfloor = xy$$

- Revisited quarter square method -*F.J.Taylor, 1981*- 0%
  Quarter square method adapted to modular multiplication.
  Method bounded to the same restrictions than the previous one.
  Taylor pre-computes: $\forall\, 0 \leq x < p, MEM(x) = 4^{-1}x^2 \mod p$
  The output is directly reduced $\mod p$, *i.e.* no reduction and no mulitplication.
  But $\#MEM = pn$ bits.
- Booth's algorithm-*1951*- 20%
  This is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation, used in the Infineon's ZDN algorithm. This multiplication algorithm is based on a changed of the representation of numbers in two's complement notation.
- Karatsuba's Method -*1962*- 0%
  Divide and conquer applied to the classical multiplication algorithm: numbers to multiply are divided in two equal parts, then using Gauss's trick the result is obtained with only three multiplication (instead of four).

Note that the week after Kolmogorof state that any algorithm would not be faster than n^2, one of his student, Anatolii Alexeevich Karatsuba, proved him he was wrong with this algorithm.
Complexity: $\mathcal{O}\left(3n^{log_2(3)}\right) \approx \mathcal{O}(3n^{1.55})$.

- Toom-Cook Algorithm -*1963*- $-\infty$
Generalise the previous method dividing each number to be multiplied in k parts and it is typically used for intermediate-size multiplications, before using the asymptotically faster Schonhage Strassen algorithm. Not applicable to smart card.
Complexity: $\mathcal{O}\left(n^{log(5)/log(3)}\right) \approx \mathcal{O}(n^{1.465})$

- Schonhage-Strassen algorithm -*1971*- $-\infty$
Computation by isomorphism using FFT in rings with $2^n + 1$, it is used in practice for numbers with more than 10,000 to 40,000 decimal digits. Not applicable to smart card.
Complexity: $\mathcal{O}(nlog(n)log(log(n)))$. warning

- Kochanski multiplication's -*Kochanski 1985*- 20%
Kochanski multiplication is an algorithm that allows modular arithmetic (multiplication or operations based on it, such as exponentiation) to be performed efficiently when the modulus is large (typically several hundred bits). Widely used in constrained environment.

- Furer's algorithm -*2007*- $-\infty$
Improvement of the Schonhage-Strassen's algorithm.
Complexity: $\mathcal{O}\left(nlog(n)2^{log^*(n)}\right)$ where log^* is the iterated logarithm. Not applicable to smart card.

- Hardware multiplier -*2000's*- 100%
Especially dedicated multiplier, with a high efficiency. According to a highly valuable source, namely wikipedia, hardware multiplier are, in general multiplying by n, using the operation deduced from the following representation:
  - Canonical binary representation
  - Booth encoding

  But bit are not automatically scan 1 by 1.

- Masked Multiplication -*2000's*- 100%
This is a generic counter-measure, to hide usage of multiplicands.

$$p \times q = (p - x) \times (q - x) + x \times (p + q) - x^2$$

  Simple method to blind operands

- 'As fast as you want' Nota, there exist a theoreme stating:
Given $\epsilon > 0$ there exists a multiplication algorithm such that the number of elementary operation $T(n)$ needed to multiply two m-bit digit numbers satisfies:

$$T(n) < c(\epsilon) \times n^{1+\epsilon}$$

II .2.4 Squarring

Maybe TBD

II .2.5 Reduction

Problem: T a number to reduce modulo N, respectively $2n$ and n bit long.
Aim: to deal efficiently with reduction modulo N, computation in $\mathbb{Z}/N\mathbb{Z}$.

References:
Bosselaers et al. [1993]
Hasenplaugh et al. [2007]
Cao et al. [2014]

Duality between Multiplication and Modular Reduction: Fischer and Seifert [2005]

- Euclidean method method - *2000BC-* 0%
  A complete Euclidean division is achieved:
  $T = qN + r$ with $0 \leq r < N$

  Example:
  Let's to be computed $x.y \mod N$ with $N = 119$, and $x = 63$, $y = 57$:
  $63.57 \mod 119 = 3591 \mod 119 = 119.30 + 21 \mod 119 = 21 \mod 119$
  But this requires a costly division which process is hidden...

- Knuth's "scholar" method - *1969-* 0%
  Improve the previous algorithm in the sense that only a partial Euclidean division is performed.

- Montgomery's reduction - Montgomery [1985] - 60%
  Montgomery reduction algorithm is a pillar of modern modular calculus. It exists in many versions, with computation by bloc or globally with interleaved multiplication or not, and many more.

  Computation by isomorphism: computation are not performed in the original group $\mathbb{Z}/N\mathbb{Z}$ but rather in another representation of this ring.
  for R co-prime with N, group isomorphism for R co-prime with N : $\phi \begin{cases} \mathbb{Z}/N\mathbb{Z} \longrightarrow \mathbb{Z}/N\mathbb{Z}' \\ x \longmapsto x.R \end{cases}$ the operation corresponding [6]to the multiplication in $\mathbb{Z}/n\mathbb{Z}$ is $M_R(a, b) \mapsto a.b/R$

Algorithm 9: Montgomery's reduction algorithm *redct* function

Input: $T \in \mathbb{Z}/R.N\mathbb{Z}$ to reduce modulo N
$R, N \in \mathbb{Z}/N\mathbb{Z}$ such that $pgcd(R, N) = 1$
$N' \in \mathbb{Z}/R\mathbb{Z}$ such that $R.R' - N.N' = 1$
Output: $T.R^{-1} \mod N$

```
m ← T.N' mod R ;
t ← (T + m.N).R^{-1} ;
if t ≥ n then
    t ← T mod N;
return t;
```

we have an algorithm:

- achieving a division by R, that can be chosen under condition
- reducing its input modulo N
- without divisions by N, if $T \leq R.N$

Proof: It is clear that $T = T + m.N \mod N$, with $R.R' - N.N' = 1$ and $m = T.N' \mod R$ we have: $T + m.N = T - (T \mod R)$ then $T + m.N \leq RN + RN$ so that $0 \leq t \leq 2N$.

[6] Since $\phi(a).\phi(b) = a.b.R^2$, and $\phi^{-1}(a.b.R^2) = a.b.R$

Function *redct*is used to multiply in Montgomery domain but also to as 'transfer':
Thanks to: $redct(x.R^2) = x.R \mod N = \phi(x)$ and $redct(x) = x.R^{-1} \mod N = \phi^{-1}(x)$.
In practice we can choose R regarding the modulus pre-compute $R^2 \mod N$, $R^{-1} \mod N$

Example: Let's to be computed $x.y \mod N$ with $N = 119$, and $x = 63$, $y = 57$:
Define isomorphism : choose a hardware compatible R with the condition $R > N$, $R = 2^7 = 128$
Precomputed values : $R^2 \mod N = 81$, $R^{-1} \mod N = 53$
Transfer data into Montgomery Domain:

$$\phi(x) = x.R \mod N = redct(x.R^2) = redct(63.81) = 91$$
$$\phi(y) = y.R \mod N = redct(y.R^2) = redct(57.81) = 37$$

Multiplication in Montgomery domain:

$$M_R(\phi(x), \phi(y)) = redct(91.37.53) \mod 119 = 70$$

Return the result from the Montgomery domain:

$$x.y \mod N = redct(70) = 21$$

All the division was by R, if R is a power of two for the hardware this only a shift. Secondly R^2 and R^{-1} are precomputed values.

- Barrett's reduction -*Barrett [1986]* Widely used

Involves one pre-computation: $\mu = \lfloor \frac{2^{2n}}{N} \rfloor$.
The reduction take the form:

$$R = T - \lfloor \lfloor \frac{T}{2^n} \rfloor \frac{\mu}{2^n} \rfloor N$$

Requires: two +1 n-bit multiplications and on subtraction.

Example: $x.y \mod N$ with $N = 119$, and $x = 63$, $y = 57$
Precomputed values : $\mu = \lfloor 2^{16}/119 \rfloor = 550$

The reduction take the form
$R = 3591 - \lfloor \lfloor \frac{3591}{2^8} \rfloor \frac{550}{2^8} \rfloor 119$
$R = 3591 - \lfloor 14 \times 2.148 \rfloor 119$
$R = 3591 - 30.119$
$R = 3591 - 3570$
$R = 21$

- Iterative folding -*Hasenplaugh et al. [2007]* 0%

Iterates a divide and conquer approach to have smaller multiplication, in Barrett's algorithm.
This involve more pre-computations: $\forall i, 1 \leq i \leq F, M^{(i)} = 2^{(1+2^{-i})n}$
and for the final Barrett step : $\mu = \lfloor \frac{2^{2n}}{M} 2^{2^{-F}n} \rfloor$

The reduction take the form
$N^{(0)} = N$
$N^{(i)} = N^{(i-1)} \mod M^{(i)} + \lfloor \frac{N^{(i-1)}}{M^{(i)}} \rfloor M^{(i)} \forall i, 1 \leq i \leq F$
$R = N^{(F)} - \lfloor \lfloor \frac{N^{(F)}}{2^{(1+2^{-F})n}} \rfloor \frac{\mu}{2^{2^{-F}n}} \rfloor M$

Optimal for $F = 2$

Example: Let's to be computed $x.y \mod N$ with $N = 119$, and $x = 63$, $y = 57$:
Precomputed values : N is 8 digits, we set $F = 2$:
$M^{(1)} = 2^{(1+2^{-1})8} = 2^{12} = 4096$

$M^{(2)} = 2^{(1+2^{-2})8} = 2^{10} = 1024$
And for the final Barrett step :
$\mu = \lfloor \frac{2^{16}}{119} 2^{2^{-2}8} \rfloor = \lfloor \frac{2^{16}}{119} 4 \rfloor = 2202$

The reduction take the form
$N^{(0)} = 3591$
$N^{(1)} = 3591 \mod 2^{12} + \lfloor \frac{3591}{2^{12}} \rfloor M^{(1)} = 3591 \mod 2^{12} + \lfloor \frac{3591}{2^{12}} \rfloor 4096 = 3591$
$N^{(2)} = N^{(1)} \mod 2^{(1+2^{-2})8} + \lfloor \frac{3591}{2^{10}} \rfloor M^{(2)} = 3591 \mod 2^{10} + \lfloor \frac{3591}{2^{10}} \rfloor 1024 = 519 + 3072 = 3591$
$R = N^{(2)} - \lfloor \lfloor \frac{N^{(2)}}{2^{(1+2^{-2})n}} \rfloor \frac{\mu}{2^{2^{-2}n}} \rfloor M$
$R = 3591 - \lfloor \lfloor \frac{3591}{2^{10}} \rfloor \frac{\mu}{2^2} \rfloor 119$
$R = 3591 - 3550.5 * 119$

$\mu = \lfloor 2^{16}/119 \rfloor = 550$

The reduction take the form
$R = 3591 - \lfloor \lfloor \frac{3591}{2^8} \rfloor \frac{550}{2^8} \rfloor 119$
$R = 3591 - \lfloor 14 \times 2.148 \rfloor 119$
$R = 3591 - 30.119$
$R = 3591 - 3570$
$R = 21$

- Sedlack's reduction -*1987*- 0%
- Quisquater reduction -*1990*- 0%
  Source: Joye [2012] Patent protected
- ZDN algorithm -*Infineon technologies 1990's*- 0%
  A far effective, hardware-compatible algorithm, ZDN based modular multiplication was developed by Infineon. ZDN based modular multiplication replaces the multiplication and reduction operations with a single operation, which the system can execute in a single clock-cycle. This algorithm implements a look-ahead Booth (LABooth) multiplication with ZDN based (Zwei Drittel N , 2/3N in German) modular reduction (LARed). This algorithm further improves on register constraints because it ensures that the partial product remains at approximately 2/3N. With this algorithm the multiplication and modular reduction are calculated completely in parallel WTF!.

  VS side channel cryptanalysis:
  * immune against SvsM discrimination

Barrett vs Montgomery
Similarities:
Both require pre-computing various constants for a given modulus N. Their input range is $[0, N^2)$.
Their last las step lies in $[0, 2n)$, with a final fix-up step to reach the output range of $[0, n)$.
They perform 2 internal multiplications per reduction.
Their reduction phases avoid division or remainder by non-powers-of-2.

Differences:
Montgomery reduction requires expensive conversion into and out of 'Montgomery form', whereas Barrett reduction operates on regular numbers directly.
Montgomery is based on modular congruences and exact division, whereas Barrett is based on approximating the real reciprocal with bounded precision.

Consumption: "

If the modulus is n bits long, Montgomery: two n-by-n bit multiplications yielding $2n$ bits. Barrett: one $2n$-by-n bit multiplication yielding $3n$ bits, plus a n-by-n bit multiplications yielding $2n$ bits,

Usage:
Montgomery reduction is suitable for modular exponentiation but not for working with various unrelated numbers where Barrett reduction is a good candidate for both,but more expensive.

II .2.6 Exponentiations

AIM: realize the following operation $y = x^n$.

Warning !

Note that of the following algorithms might be implemented indifferently in left-to-right or for right-to-left, or also in atomic or non atomic version, and so on and so on... A lot of combination are possible. Warning

II .2.6.1 Convention and Names !

The famous routine **"Square & Multiply"** was named this way for a descriptive purpose: when we are using it we are effectively squaring and multiplying.

On the other hand it's mathematical name is **"dichotomic exponentiation"** insisting on the fact that to achieve the exponentiation, is proceeded to recursive calls to a sub-routine and that for each call there are two possibility multiplication.

The name **"binary method"** has been given with the same spirit. Starting from now we adopt this kind of naming for every algorithm relative to exponentiation. In this section will be viewed other algorithm scanning not one bit but several bits of the exponent at each recursion. To have a clear naming we extend the previous convention to this other class of algorithms with the following convention:

2-ary method: each recursion 2^1 possibilities and bit scanned 1 by 1.
2^k-ary method: each recursion 2^k possibilities and bit scanned k by k.

Some might object that could be said that a 3-arry method is scanning bits 3 by 3.
Example: What about x^{27} without multiplying?
If you have an efficient algorithm to compute third power and you build on it an exponentiation algo with at each recursion a possible multiplication by x^0, x^1, x^3. This algorithm works in a representation of 27 in base 3, and can't be named with the other convention. It's named trichotomic exponentiation, ternary method or cube and multiply.

Known algorithms:

- Recursive method - *Neanderthal-* $-\infty$
  This method is simply applying the definition of the exponentiation as a set of exponentiation to obtained the desired result.

  Enjoy: some academic paper from 1986 has been found considering that this algorithm was competitive.

  This method will always return the worst of all exponentiation chains: $\{1, ..., n\}$ which length $l(n)$ is n. To compute x^n, n successives power of x are computed !

II .2.6.2 Two Square & Multiply algorithms and the factor method

- LtoR Square & Multiply - *Chandah-sutra of Pingala, a classic Hindu, 400AC-* 0%
  Method designed following the observation that squaring can faster than multiplying. A square is done each time that is scanned another bit of the exponent. A multiplication is done conditionally depending on the value ofthe scanned bit.

  Since $n > 4$ this is computationally more efficient than naive exponentiation. This method uses the same addition chain that 'Multiply always' but the same one than the LtoR Binary method.

Figure II .5: $d_2 = 01100100...$

Algorithm 10: LtoR dichotomic exponentiation

Input: $x, d = d_{t-1}...d_1 d_{0_2}$
Output: $y = x^d \mod n$

```
1 y ← 1 ;
2 for i ← t − 1 to 0 do
3 |   y ← y^2 mod n
4 |   if d_i = 1 then
5 |   |   y ← y × x mod n
6 return y;
```

Example : LtoR Square & Multiply: $n = 23_{10} = 10111_2$ the method starts from the MSB.

$y = 1$
$y := y^2$ $y := y * x$ $(y = x)$
$y := y^2$ $(y = x^2)$
$y := y^2$ $y := y * x$ $(y = x^5)$
$y := y^2$ $y := y * x$ $(y = x^{11})$
$y := y^2$ $y := y * x$ $(y = x^{23})$

Powers of x successively computed: $x\ x^2\ x^4\ x^5\ x^{10}\ x^{11}\ x^{22}\ x^{23}$, length 8

Names
Algorithm also known as: dichotomical exponentiation, Square & Multiply, binary method.
Do not mistaken this algorithm with its atomic version as it is not using a specialized routine for squaring it is also called sometime 'Multiply always'...

Note that the first non trivial step is always the same as we have $d_{t_{min}-1}$, and will result in $y = x$ and therefore the it can be replaced for an if.

Which leads the number of operations to be performed to:

$$t_{min} + \omega_{\mathcal{H}}(d) - 2$$
$$\text{with } t_{min} = \lfloor log_2(d) \rfloor$$

- RtoL Square & Multiply- *Al Kashi, 1427 AC-* 0%

  Those two algorithms are exactly the same, those two versions are presented here. The fist version is scanning the bit form Right to Left, the second one is doing the conversion on the fly starting from d. $d_i = 1$ *vs* $d \leftarrow \lfloor \frac{d}{2} \rfloor$.

Algorithm 11: RtoL dichotomic exponentiation -On the fly version-

Input: $x, n \in \mathbb{N}$, $x \leq n$
Output: $y = x^d \mod n$

```
1 y ← 1 ;
2 local ← x ;
3 while d ≠ 0 do
4 |   if d mod 2 = 1 then
5 |   |   y ← y × local mod n ;
6 |   d ← ⌊d/2⌋;
7 |   local ← y^2 mod n
8 return y;
```

Algorithm 12: RtoL dichotomic exponentiation

Input: $x, n \in \mathbb{N}$, $x \leq n$, $d = d_{t-1}d_1d_{0_2}$
Output: $y = x^d \mod n$

```
1 y ← 1 ;
2 local ← x ;
3 for i ← 0 to t − 1 do
4 |   if d_i = 1 then
5 |   |   y ← y × local mod n
6 |   local ← y^2 mod n
7 return y;
```

Example: RtoL Binary method: $n = 15_{10} = 1111_2$

$y = 1$	$l = x$	$(y = 1, l = x)$
$y := y * l$	$l := l^2$	$(y = x, l = x^2)$
$y := y * l$	$l := l^2$	$(y = x^3, l = x^4)$
$y := y * l$	$l := l^2$	$(y = x^7, l = x^8)$
$y := y * l$		$(y = x^{15}, l = x^{16})$

Powers of x successively computed: $x\ x^2\ x^3\ x^4\ x^7\ x^8\ x^{15}$, length 7: 6 operations
This is the smallest value of n for which the binary method is not optimal:
The factor method with $x^{15} = (x^3)^5$, lead to 5 operations

Example: RtoL Binary method: $n = 23_{10} = 10111_2$

$y = 1$	$l = x$	$(y = 1, l = x)$
$y := y * l$	$l := l^2$	$(y = x, l = x^2)$
$y := y * l$	$l := l^2$	$(y = x^3, l = x^4)$
$y := y * l$	$l := l^2$	$(y = x^7, l = x^8)$
	$l := l^2$	$(y = x^7, l = x^{16})$
$y := y * l$		$(y = x^{23}, l = x^{16})$

Powers of x successively computed: $x\ x^2\ x^3\ x^4\ x^7\ x^8\ x^{16}\ x^{23}$, length 8: 7 operations

Note that those algorithms are provided in a pedagogical way, in a real life implementation two multiplication can be skipped: the $d_i = 0$ step $y \leftarrow 1 \times x \mod n$ can be exchanged for a if. The last square is also useless and can be prevented.

Which leads the number of operations to be performed to:

$$t_{min} + \omega_{\mathcal{H}}(d) - 2$$
$$\text{with } t_{min} = \lfloor log_2(d) \rfloor$$

- Square & Multiply Always 0%

  This a counter-measure, to prevent multiply vs non multiply discrimination.
  The SPA vulnerability of S&M algorithms come from a possible square versus multiply discrimination, coming from the algorithmic implementation of those routines.

  Solutions:

  - *'Mutliply Always' i.e. Naive 'Square & Multiply'*
    replace squarings by multiplications, but slower still a condition remains ...
  - *'Square & Multiply Always'*
    Whatever the need 'always multiply': if no multiplication is requirred place a bogus multiplication

Figure II .6: LtoR Square & Multiply Always illustration

  - *'Joye's Multiply Always' i.e. Atomic 'Square & Multiply'*
    replace the squaring algorithm by a multiplying one: fully atomic algorithm.

- Square & Multiply on the fly reduction- *G.R.Blakley, 1983 AC-* 0%

- Nota on Square *vs* Multiply operands:

  When applying Square & Multiply algorithm, additionally to the fact that the dedicated routines are different, there is structural difference between the two fundamental operation:

  - *Square operations* possesses a variable operand, likely to change at each execution.
  - *Multiply operations* possesses a static operand, the x to elevate to a certain power.

- Factor Method *D.E.Knuth, 1908 AC-* 0%

  This method is recursive and is neither better or worst than the binary method.
  if d is prime, compute $x \times x^{d-1}$
  if not and $d > 3$, compute $x^d = x^p \times x^{d'}$ with p prime.

  The Factor Method is better than the binary method *in average.*

  Example: factor method: $n = 15_{10}$
  $x^{15} = (x^3)^5$ and $x^3 = x \times x^2$
  $x^{15} = x^3 \times (x^3)^4$
  $x^{15} = x^3 \times ((x^3)^2)^2$
  Powers of x successively computed: $x\ x^2\ x^3\ x^6\ x^{12}\ x^{15}$, length 6: 5 operations
  This example is the first where the factor method is better than the binary method.

  But the factor method is not always better ...

  Example: factor method: $n = 33_{10} = 100001_2$
  $x^{33} = (x^3)^{11}$ and $x^3 = x \times x^2$
  $x^{33} = x^3 \times (x^3)^{10}$
  $x^{33} = x^3 \times ((x^3)^2)^5$
  $x^{33} = x^3 \times (x^6)^5$
  $x^{33} = x^3 \times x^6 \times ((x^6)^2)^2$
  Powers of x successively computed: $x\ x^2\ x^3\ x^6\ x^{12}\ x^{24}\ x^{30}\ x^{33}$, length 8: 7 operations
  Whereas binary method requires $6 + 2 - 2 = 6$ operations...

II .2.6.3 Atomic Square & Multiply

- Atomic Square & Multiply - Joye et al. [2004], *Link* *Joye 2003-* 0%

  Generic counter measure principle, based on the concept of 'side channel indistinguishability', applicable to virtually all algorithms to protect them for SPA.

  Here is presented the version to protect the Square & Multiply algorithm against the SPA. Carefully note that k is simply a boolean.

Algorithm 13: Atomic Square & Multiply - LtoR version

Input: $x, n \in \mathbb{N}$, $x \leq n$, $d = d_{t-1}d_1d_{0_2}$
Output: $y = x^d \mod n$

```
1 R_0 ← T.N' mod R R_1 ← T.N' mod R ;
2 i ← t − 1 k ← 0 ;
3 while i ≥ 0 do
4 |   R_0 ← R_0 × R_k ;
5 |   k ← k ⊕ d_i ;
6 |   i ← i − k
7 return y;
```

Example: LtoR version - Atomic Square & Multiply: $n = 23_{10} = 10111_2$

$R_0 = 1$	$R_1 = x$	$(k = 0 \;\; i = 4)$
$R_0 = 1$		$(k = 1 \;\; i = 4)$
$R_0 = x$		$(k = 0 \;\; i = 3)$
$R_0 = x^2$		$(k = 0 \;\; i = 2)$
$R_0 = x^4$		$(k = 1 \;\; i = 2)$
$R_0 = x^5$		$(k = 0 \;\; i = 1)$
$R_0 = x^{10}$		$(k = 1 \;\; i = 1)$
$R_0 = x^{11}$		$(k = 0 \;\; i = 0)$
$R_0 = x^{22}$		$(k = 1 \;\; i = 0)$
$R_0 = x^{23}$		$(k = 0 \;\; i = -1)$

Remark that the same power of x than in Square & Multiply - LtoR version

$$x \;\; x^2 \;\; x^4 \;\; x^5 \;\; x^{10} \;\; x^{11} \;\; x^{22} \;\; x^{23}$$

So this algorithm is: using the same addition chain that Square & Multiply to get the final result as a LtoR S&M, but in much more side-channel resistant way:

- no condition: it is doing exactly the same thing at each recursion
- no squaring vs multiplication discrimination can be achieve from an algorithmic point of view.

The price is no special routine is used to square, automatically slowing down the whole process

Disambiguation
Because of the absence of squaring it is then frequently called 'multiply always' algorithm, please notice that is is ambiguous indeed, see page 62.

Conventional Names
"Joye's multiply always, LtoR version", "Atomic Square & Multiply, LtoR version"

Known Vulnerability
It have to be understood that this implementation is, by definition, absolutely invulnerable to squaring vs multiplication discrimination **from an algorithmic point of view**.

But it does not mean that it is invulnerable to squaring vs multiplication discrimination. Indeed squaring vs multiplication discrimination can be achieved focusing on the hamming weight.

II .2.6.4 Scanning digits in base 2^k

- Window Method: the LtoR 2^kary method: - *Brauer 1939*- 0%
  Generalisation of the previous method, the bits of the exponent, in base 2, are scanned k by k, this method requires pre-computed values. Principle: k squares are done each time scanning k other bits of the exponent A multiplication by one of the $2^k - 3$ pre-computed value is done depending on the scanned group of bits..

  Only the precompuation of the x^{d_j} such that d_j appears in the representation of d is needed.

Figure II .7: LtoR 2^kary method illustration, with $k = 3$

Algorithm 14: LtoR 2^k-ary method

Input: $x, n \in \mathbb{N}$, $x \leq n$, $d = d_{t-1}d_1d_{0_2}$

$x_0 \leftarrow 1$;
for $i = 0 \rightarrow 2^k - 1$ **do**
  $x_i \leftarrow x_{i-1} \times x$;
$y \leftarrow 1$;
for $i = t - 1 \rightarrow 0$ **do**
  $y = x^{2^k}$;
  if $d_i > 0$ **then**
    $y = y \times x_i$;
return y;

Example 1: LtoR 2^k-ary method
Let's assume that $m = 8$ *i.e.* $k = 3$ with $n = 326_{10} = 101\ 000\ 110_2$ the method start scanning the bits from their MSB but three by three. The precomputed values are: x^2, ..., x^7

$y := 1$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^5$ $(y = x^5)$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^0$ $(y = x^{40})$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^6$ $(y = x^{326})$

Example 2: LtoR 2^k-ary method
Let's assume that: $n = 11651101_{10} = 101\ \ 100\ \ 011\ \ 100\ \ 100\ \ 000\ \ 011\ \ 101_2$

$y := 1$ $y := y \times x^5$ $(y = x^5)$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^4$ $(y = x^{44})$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^3$ $(y = x^{355})$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^4$ $(y = x^{2\,844})$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^4$ $(y = x^{22\,756})$
$y := y^2$ $y := y^2$ $y := y^2$ $(y = x^{182\,048})$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^3$ $(y = x^{1\,456\,387})$
$y := y^2$ $y := y^2$ $y := y^2$ $y := y \times x^5$ $(y = x^{11\,651\,101})$

To get this result a certain decomposition of n was used which lead to 21 squarings and 7 multiplications.

Name & convention The 2^kary method may also be called k-bits Window method, insisting on the number of bit scanned at each recursion...

- Window Method: LtoR Optimized 2^k-ary exponentiation: - 0%
  Optimization of the previous method, only even power of x are stored only.
  Require $2^{k-1} - 1$ precomputed values overall.

Algorithm 15: Optimized LtoR 2^k-ary method

Input: x, $d = d_{t-1}d_1d_{0_{2^k}}$
$x_0 \leftarrow 1$;
$x_1 \leftarrow x$;
$x_2 \leftarrow x^2$;
for $i = 1 \rightarrow 2^{k-1} - 1$ **do**
  $x_{2i+1} \leftarrow x_{2i-1} \times x_2$;
$y \leftarrow 1$;
for $i = t - 1 \rightarrow 0$ **do**
  $y = x^{2^k}$;
  define: $d_i = 2^{h_i} \times u_i$ where u_i is odd and h_i maximal ;
  $y = (y^{2^{k-h_i}} \times x_{u_i})^{2^{h_i}}$
return y;

Example: LtoR Optimized 2^k-ary exponentiation
$\overline{n = 326_{10}} = 101\ 000\ 110_2$ and $k = 3$ *i.e.* $m = 8$ the method starts scanning the bits from their MSB but three by three. The precomputed values are: x^3, x^5, x^7.

$y := 1$				
$y := y^2$	$y := y^2$	$y := y^2$	$y := y \times x^5$	$(y = x^5)$
$y := y^2$	$y := y^2$	$y := y^2$	$y := y \times x^0$	$(y = x^{40})$
$y := y^2$	$y := y^2$	$y := y \times x^3$	$y := y \times x^2$	$(y = x^{326})$

Practical remark:

On a SPA point of view it is crucial to know whether or not the implementation is saving pre-computation !

- if precomputed value are optimized: the goal become to discriminate elevation to the 8^{th} power from multiplication by $(x^i)_{2 \leq i \leq 7}$ and in this case we will always have sequence of elevation to the 8^{th} power then a multiplication (or not). But the pattern of the curve should match exactly the key.
- in this case we will not always have sequence of elevation to the 8^{th} power then a multiplication (or not). Compare the last line of the previous example and the same line with the 8-arry method.

- Window Method: RtoL 2^kary method: 0%

Algorithm 16: RtoL 2^k-ary method

Input: $x, n \in \mathbb{N}$, $x \leq n$, $d = d_{t-1}d_1d_{0_2}$

$x_0 \leftarrow 1$;
for $j = 1 \rightarrow 2^k - 1$ **do**
  $R_j \leftarrow 1$;
$y \leftarrow x$;
while $m \leq n$ **do**
  $d \leftarrow n \mod m$;
  if $d \neq 0$ **then**
    $R[d] \leftarrow R[d] \times y$;
  $y \leftarrow y^m$;
  $n \leftarrow \lfloor n/m \rfloor$;
$R[n] \leftarrow R[n] \times y$;
$y \leftarrow R[m-1]$;
for $j = m - 2 \rightarrow 1$ **do**
  $R[j] \leftarrow R[j] \times R[j+1]$;
  $y \leftarrow y \times R[j]$;
return y;

Example : RtoL 2^k-ary method, $m = 8$ *i.e.* $k = 3$ with $n = 326_{10}$

Initialization phase:
$R = \{1, 1, 1, 1, 1, 1, 1\}$, $y := x$, $m = 8$, $n = 326$

While loop:
$d = 326 \mod 8 = 6$, $R = \{1, 1, 1, 1, 1, x, 1\}$, $y := x^8$, $n = 40$
$d = 40 \mod 8 = 0$, $R = \{1, 1, 1, 1, 1, x, 1\}$, $y := x^{64}$, $n = 5$

End while:
$R = \{1, 1, 1, 1, x^{64}, x, 1, 1\}$, $y := 1$

Recomposition phase:
$j = 6, R = \{1, 1, 1, 1, x^{64}, x, 1\}$, $y := x$
$j = 5, R = \{1, 1, 1, 1, x^{65}, x, 1\}$, $y := x^{66}$
$j = 4, R = \{1, 1, 1, x^{65}, x^{65}, x, 1\}$, $y := x^{131}$
$j = 3, R = \{1, 1, x^{65}, x^{65}, x^{65}, x, 1\}$, $y := x^{196}$
$j = 2, R = \{1, x^{65}, x^{65}, x^{65}, x^{65}, x, 1\}$, $y := x^{261}$
$j = 1, R = \{x^{65}, x^{65}, x^{65}, x^{65}, x^{65}, x, 1\}$, $y := x^{326}$

Variants:
right to left version, optimized version

II .2.6.5 Sliding window algorithms

- LtoR static 2^k sliding window method: 0%

  Adapt the previous method in minimizing the size of the window when possible. First of all maximum length for the window has to be defined, then then bit are scan adapting the window size to isolate the zeros.

  Complexity:

  The algorithm requires $\omega - 1 + n$ squares and, at most, $2^{\omega-1} - 1 + \frac{n}{\omega}$ multiplications.

Algorithm 17: LtoR static ω-sliding window method

Input: $x, n \in \mathbb{N}$, $x \leq n$, $d = d_{t-1}d_1d_{0_2}$, k
Output: $y = x^d \mod n$

```
1  x_1 ← x ;
2  x_2 ← x^2 ;
3  for i = 0 → 2^k do
4      x_i = x_{2i-1} × x_2 ;
5  y ← 1 ;
6  i ← t − 1 ;
7  while i ≥ 0 do
8      if d_i = 0 then
9          y ← y^2;
10         i ← i − 1;
11     else
12         s ← max(i − w + 1, 0) ;
13         while d_s = 0 do
14             s ← s + 1
15         u ← {d_i...d_s}_2;
16         y ← y^{2^{i−s+1}} × x_u;
17         i ← i − s + 1;
18 return y;
```

Example 1: LtoR static ω-sliding window method

Let's assume that $n = 11651101_{10} = 1011\ \ 000\ \ 111\ \ 00\,1\ \ \ 000000\ \ 111\ \ 0\ \ 1_2$ and $\omega = 4$

$y := 1$					$y := y \times x^{11}$
$y := y^2$	$y := y^2$	$y := y^2$			
$y := y^2$	$y := y^2$	$y := y^2$			$y := y \times x^7$
$y := y^2$	$y := y^2$				
$y := y^2$					$y := y \times x$
$y := y^2$	$y := y^2$	$y := y^2$	$y := y^2$	$y := y^2$	$y := y^2$
$y := y^2$	$y := y^2$	$y := y^2$			$y := y \times x^7$
$y := y^2$					
$y := y^2$					$y := y \times x$

$n = 11651101_{10} = 1011\ \ 000\ \ 111\ \ 00\,1\ \ \ 00000\ \ 0111\ \ 01_2$ and $\omega = 4$

$y := 1$					$y := y \times x^{11}$
$y := y^2$	$y := y^2$	$y := y^2$	$y := y^2$		
$y := y^2$	$y := y^2$	$y := y^2$			$y := y \times x^7$
$y := y^2$	$y := y^2$	$y := y^2$			
$y := y^2$	$y := y^2$				$y := y \times x$
$y := y^2$					
$y := y^2$	$y := y^2$	$y := y^2$	$y := y^2$	$y := y^2$	$y := y \times x^7$
$y := y^2$	$y := y^2$	$y := y^2$	$y := y^2$		$y := y \times x$

- LtoR dynamical 2^k sliding window method: 0%
  Adapt the previous method in minimizing the size of the window when possible. First of all maximum length for the window has to be defined, then then bit are scan adapting the window size to isolate the zeros.

Algorithm 18: LtoR dynamical k-Sliding window algorithm

Input: $x, n \in \mathbb{N}$, $x \leq n$, $d = d_{t-1}d_1d_{0_2}$, k
Output: $y = x^d \mod n$

```
1  x_1 ← x ;
2  x_2 ← x^2 ;
3  PreComputation = [x_1, x_2];
4  for i = 0 → 2^k do
5      PreComputation[i] = PreComputation[2i − 1] × x_2 ;
6  y ← 1 ;
7  i ← ⌊ln_2(d)⌋ ;
8  while i > 0 do
9      d ← n mod m;
10     if d_i = 0 then
11         y = y^2;
12         i = i − 1;
13     else
14         find the longest bit string e_i...e_l such that i − l + 1 ≤ k and e_l = 1;
15         y = y^{i−l+1} × x_{e_i···i+k−1};
16 return y;
```

Variants

The here presented algorithm is one with dynamical length of window. Another simpler version would be to define the multiplication with the bit string of constant length, giving up the condition '$e_l = 1$'. Also possible algorithm is right to left version

- The two type of sliding window method in a nutshell
  illustration with $w = 111001010001_2$

CLNW: $\underline{111}\ 00\ \underline{101}\ 0\ \underline{001}_2$
VLNW: $\underline{111}\ 00\ \underline{101}\ 000\ \underline{1}_2$

II .2.6.6 The Montgomery Ladder

- Montgomery ladder technique - Joye and Yen [2002], *Link* - 0%

Algorithm 19: Montgomery ladder technique

Input: x, $d = d_{t-1}d_1d_{0_2}$
Output: $y = x^d \mod n$

```
1  R0 ← 1 ;
2  R1 ← x ;
3  for i = t − 1 to 0 do
4      if di = 0 then
5          R1 ← R0 × R1;
6          R0 ← R0^2
7      else
8          R0 ← R0 × R1;
9          R1 ← R1^2
10 return R0
```

Example: Montgomery ladder technique:
$n = 23_{10} = 10111_2$

$R_0 = 1$	$R_1 = x$
$R_0 = x^1$	$R_1 = x^2$
$R_0 = x^2$	$R_1 = x^3$
$R_0 = x^5$	$R_1 = x^6$
$R_0 = x^{11}$	$R_1 = x^{12}$
$R_0 = x^{23}$	$R_1 = x^{24}$

Note that the following powers of x are successively computed: x x^2 x^3 x^4 x^5 x^6 x^{11} x^{12} x^{23} x^{24}.

VS side channel cryptanalysis
This technique is vulnerable to the doubling attack

II .2.6.7 Randomized algorithms

- Random RtoL k-ary method - Tunstall [2005], *Link* - 0%
  Is taken advantage of RtoL specificities to introduce some randomness.

Algorithm 20: Random Order - RtoL k-ary exponnetiation

Input: $x \in \mathbb{G}$, $d \in \mathbb{N}$, $n \in \mathbb{N}$, r number of values to store in memory
Output: $y = x^d \mod n$

```
for i = 1 → k − 1 do
    R[i] ← 1_G ;
S[0] ← x ;
for i = 1 → r − 1 do
    S[i] ← S[i − 1]^k ;
for i = 0 → r − 1 do
    D[i] ← n mod k ;
    n ← ⌊n/k⌋
γ ← r − 1;
while n > 0 do
    τ = RandInteger(0..r − 1) ;
    if D[τ] ≠ 0 then
        R[D[τ]] ← R[D[τ]] × S[τ]
    S[τ] ← S[γ] ;
    D[τ] ← n mod k ;
    n ← ⌊n/k⌋ ;
    γ ← τ
for i = r − 1 → 0 do
    if D[i] ≠ 0 then
        R[D[i]] ← R[D[i]] × S[i]
y ← R[m − 1] ;
for i = k − 2 → 1 do
    R[i] ← R[i] × R[i + 1] ;
    y ← y × R[i]
return y;
```

Example 1: Taken form the original article
let's compute $z = x^{738530}$ Let's considere a 2^2-ary method, $m = 4$

- Setup phase

$$R = \{1, 1, 1\}$$

we fix $r = 10$, therefore we have:

$$S = \{x, x^4, x^{16}, x^{64}, x^{256}, x^{1024}, x^{4096}, x^{16384}, x^{65536}, x^{262144}\}$$
$$D = \{2, 0, 2, 3, 0, 1, 0, 1, 3, 2\}$$

Computation can be done by treating the elements of S and D in an arbitrary order.
$\gamma = 9$

- Main loop phase
  An arbitrary $\tau \in \{0, ..., 9\}$ is chosen, and we compute $R[D[\tau]] = R[D[\tau]].S[\tau]$
  (except when $D[\tau]$ is equal to zero when no operation is performed).
  $R[1] = S[5].S[7] = x^{1024}.x^{16384} = x^{17408}$
  $R[2] = S[0].S[2].S[9] = x^{16}.x^{262144} = x^{262161}$
  $R[3] = S[3].S[8] = x^{64}.x^{65536} = x^{65600}$

- Recombination phase
  $z = R[1]R[2]^2R[3]^3 = x^{17408}x^{2.262161}x^{3.65600} = x^{738530}$

II .2.6.8 NAF algorithms

Most of exponentiation algorithms can be modified to be adapted to NAF representation, **algorithms 21 and 23** illustrate this on a S&M algorithm point out that NAF-conversion can be performed on the fly if necessary.

Most important is the is the difference between NAF representation applied to RSA and applied to ECC see **algorithms 24 and 25** , for ECC certain curve allow almost 'free' inversion, the precomputed value can be optimized.

- LtotR S&M , NAF version- *Reitweisner's NAF 1960-* 0%
  Simply apply the binary NAF representation to the exponent fastening the square and multiply algorithm. This version of S&M requires a saved pre-computation: x^{-1}.

Algorithm 21: LtoR NAF S&M

Input: x, $d = d_{t-1}d_1d_{0_2}$
Output: $y = x^d$
1 Pre-compute x^{-1} ;
2 $y \leftarrow 1$;
3 **for** $i = t-1$ ***to*** 0 **do**
4 $\quad y \leftarrow y^2$;
5 $\quad$ **if** $d_i = 1$ **then** $y \leftarrow y \times x$;
6 $\quad$ **if** $d_i = -1$ **then** $y \leftarrow y \times x^{-1}$;
7 **return** y

Example : LtoR S&M, NAF version- *Reitweisner's NAF 1960*
Let's compute x^n with $n = 23_{10} = 10111_2 = \{\ 1,\ 0,\ -1,\ 0,\ 0,\ -1\ \}_{2NAF}$:

$y := x$		$(y = x)$
$y := x^2$		$(y = x^2)$
$y := y^2$	$y := y \times x^{-1}$	$(y = x^3)$
$y := y^2$		$(y = x^6)$
$y := y^2$		$(y = x^{12})$
$y := y^2$	$y := y \times x^{-1}$	$(y = x^{23})$

Remark that the intermediate computed exponentiations: $x^1,\ x^2,\ x^4,\ x^3,\ x^6,\ x^{12},\ x^{24},\ x^{23}$, length 8.

- RtoL S&M Binary-NAF - 0%

Algorithm 22: RtoL S&M Binary-NAF

Input: $x, d = d_{t-1}d_1d_{0_{2-NAF}}$
Output: $y = x^d \mod n$

```
1 l ← x ;
2 y ← 1 ;
3 for i ← 0 to t − 1 do
4 |   if d_i = 1 then y = l × y ;
5 |   if d_i = −1 then y = l^{−1} × y ;
6 |   l ← l^2
7 return y;
```

- RtoL S&M Binary-NAF On-the-fly conversion - 0%

  The following algorithm interesting is not, it just illustrate the fact that a NAF conversion can be done inside or outside the exponentiation algorithm, and this information is crucial on a SPA point of view.

Algorithm 23: RtoL S&M Binary-NAF On-the-fly conversion

Input: $x, d = d_{t-1}d_1d_{0_{2-NAF}}$
Output: $y = x^d \mod n$

```
1  y = 1 ;
2  l ← x ;
3  while d ≥ 1 do
4  |   if d mod 2 = 1 then
5  |   |   u ← 2 − (k mod 4) ;
6  |   |   d ← d − u ;
7  |   |   if u = 1 then y = x × y ;
8  |   |   if u = −1 then y = x^{−1} × y ;
9  |   d ← d/2 ;
10 |   l ← l^2
11 return y;
```

- Window NAF method - For number exponentiation *Reitweisner's NAF 1960-* 0%

Algorithm 24: LtoR k-widht binary NAF - for costly inversion

Input: $x, d = d_{t-1}d_1d_{0_{\omega NAF}}$
Output: $x = x^d$

```
1 Compute x_i = x^i for i ∈ {±1, ±3, ..., ±2^{ω-2} − 1} ;
2 y ← 1;
3 for i ← t − 1 to 0 do
4 |   y ← y^2 ;
5 |   if d_i ≠ 0 then
6 |   |   y = y × x_{d_i} ;
7 return y
```

Example: Window NAF method $n = 23_{10} = 10111_2 = \{1,\ 0,\ -1,\ 0,\ 0,\ -1,\ \}_{2NAF}$
Pre-computed values: $x^2,\ x^{-1},\ x^{-2}$.

$y := 1$		$y := y \times x^2$	$(y = x^2)$
$y := y^2$	$y := y^2$	$y := y \times x^{-2}$	$(y = x^6)$
$y := y^2$	$y := y^2$	$y := y \times x^{-1}$	$(y = x^{23})$

Remark that the following intermediate exponentiation were computed: $x^1,\ x^2,\ x^4,\ x^6,\ x^8,\ x^{12},\ x^{24},\ x^{23}$, length 8.

- Window NAF method - For point multiplication *Reitweisner's NAF 1960-* 0%
Here are saved negative pre-computations taking advantage of the arithmetic of elliptic curves: for some curve inversion is trivial. On some faster curve (inversion sacrificed) it can become a problem: some use projective coordinates for the accumulator Q, possibly also for the $(P_i)_{i\in\{1,3,...,2^{\omega-1}-1\}}$
There exist other representation of curve slowing down their speed to the profit of the inversion.

The special shape of the precomputed, only even values, is linked to the size of the ωNAF form, which by definition as only odd values and not to any savings.

Apply the ω NAF representation to the exponent fastening the window method. This algorithm is built for ECC this way .

Algorithm 25: 2^k-ary method, NAF version - for cheap inversion

Input: $x, d = d_{t-1}d_1d_{0_{\omega NAF}}$
Output: $Q = dP$

```
1 Compute P_i = d_iP for i ∈ {1, 3, ..., 2^{ω-1} − 1} ;
2 Q ← 0 ;
3 for i = t − 1 to 0 do
4 |   Q ← 2Q ;
5 |   if d_i ≠ 0 then
6 |   |   If d_i > 0 Q = Q + d_iP ;
7 |   |   If d_i < 0 Q = Q − d_{−i}P
8 return P
```

II .2.6.9 Square free algorithms

Clavier et al. [a]

$$x \times y = \frac{(x+y)^2 - x^2 - y^2}{2}$$

$$x \times y = \frac{(x+y)^2}{2} - \frac{(x-y)^2}{2}$$

Notation use for compactness and efficiency

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 2 & 1 & 1 & 1 & 2 & 1 \\ 2 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 0 \\ 1 & 1 & 3 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 3 & 3 & 3 & 0 & 3 & 3 & 1 & 1 & 3 & 1 \end{pmatrix}$$

II .2.7 what exponentiation is about

Bergeron et al. [1994]
Bernstein [2006]

The hereafter section is a little bit more theoretical but necessary to understand what exponentiation is about. After each example, we saw the key role of intermediate exponentiations to obtain the desired final result, study intermediate computation is the key for very fast exponentiation.

In fact this problem is a very good example of theoretical mathematics that have a very concrete application. The concrete application in which we are interested is answer the to the following question:

what is the minimal number of multiplications necessary to perform a given exponentiation ?

Addition chain

Let's formalize this: due to the propriety of the exponential function this problem can be described as a problem of addition, and conduct to the following definition:

Definition: A finite sequence of positive integers $1 = a_0, a_1, a_r = n$ is called an addition chain for the number n iff for each element a_i, but the first a_0, there exist two elements in the list a_j and a_k such that:

$$a_i = a_j + a_k \text{ with } k \leq j \leq i - 1$$

For example $\{ 1, 2, 4, 5, 8, 10, 13\}$, length 7.

Definition: The shortest addition chain for n is an addition chain for n with the smallest possible number of elements. The length of this shortest addition chain for n is noted $l(n)$.
For example $\{ 1, 2, 3, 6, 12, 13\}$, length $6 = l(13)$.

Algorithm 26: Exponentiation in term of Addition chain

Input: x, an addition chain , $\{n_1, ..., n_l\}$, computing d
Output: $y = x^d \mod n$

```
1 y ← 1 ;
2 for i = 1 to l do
3 |  (s, r) as n_i = n_s + n_r  y ← x^{n_s} × x^{n_r}
4 return y;
```

Definition: A Brauer chain is an addition chain in which every member after the first is the sum of the immediately preceeding element and a previous element (possibly the same element). More formally:

$$a_i = a_{i-1} + a_k \text{ with } k \leq i - 1$$

For example $\{ 1, 2, 3, 6, 7, 13\}$ is one, $\{ 1, 2, 4, 5, 8, 13\}$ is not.

Example: Shortest addition chain for $n = 23_{10} = 10111_2$
Saved value: value: x^4, x^5.

$y := y \times x$ $(y = x^1)$
$y := y^2$ $(y = x^4)$
$y := y \times x$ $(y = x^5)$
$y := x^4 \times x^5$ $(y = x^9)$
$y := y^2$ $(y = x^{18})$
$y := y \times x^5$ $(y = x^{23})$
Remark that the following intermediate exponentiation were computed: x^1, x^2, x^4, x^5, x^9, x^{18}, x^{23}, length 7.

Addition/Subtraction chain

Practical consideration

to find the shortest addition chain is 'NP difficult', and interesting algorithm shall be hardware compatible (no smart car runs a 5-ary method with 5 choices of multiplication at each iteration of the algorithm or at least its seems).

A same addition chain can be computed with different number of step depending on the algorithm (S&M *vs* Atomic S&M), changing the speed but impacting also the side channel leakage.

also exists other techniques fastening the computation of a product of several exponentiation at the same time Strauss -1964-,Yao and Pippenger -1976-

II .2.7.1 Physical threat: side channel

M/M^2 attack zero attacks (not working with CRT) doubling attack novack attack Note: There is no logic in the order in which the different attacks are listed !

Most of the time in smart card evaluation only known plain-text attack can be performed

. DPA Attack on RSA private key: Can be found in paper named: "A DPA attack on RSA in CRT mode Authors: Witteman, Van Woudenberg, Menarini Employer: Riscure Date of Publication:

Key words: Iterative attack

———————————-

. Zero-Exponent-Multiple-Data DPA: (ZEMD-DPA) Can be found in paper named: 'Power analysis attacks of modular exponentiation in smartcards." Authors: Messerges, Dabbish, Sloan Employer: Motorola Labs, University of Illinois Date of Publication: 1999 Type of attack: Recursive attack, mono-nit attack, multi-bit attack, known plain-text Countermeasure: Square-and-Multiply-Always? Key words: ZEMD-DPA, Iterative attack Idea : DPA on bit(s) of the secret key during modular exponentiation Can not append on RSA-CRT implementation ———————————-

. Zero-Exponent-Multiple-Data CPA: (ZEMD-CPA)

Can be found in paper named: "Power Analysis for Secret Recovering and Reverse Engineering of Public Key Algorithms" Authors: Amiel, Feix, Villegas Employer: Inside contactless, Gemalto Date of Publication: 2007? Type of attack: Recursive, mono-nit attack, multi-bit attack, known plain-text Countermeasure: Exponent blinding? Key words: ZEMD-CPA, Iterative attack Idea: CPA on bit(s) of the secret key during modular exponentiation. Can not append on RSA-CRT implementation

———————————-

. CPA on Multiplicand Data during Square-and-Multiply exponentiation: Authors: Amiel, Feix, Villega ? Employer: Inside contactless, Gemalto Date of Publication: 2007? Countermeasure: Square-and-Multiply-Always? Exponent blinding? Type of attack: Non-recursive, known plain-text Key words: Square-and-Multiply, Correlation, Multiplicands Idea: During an exponentiation, for each multiplication (as opposed to squaring), one of the multiplicands is constant Remark: can append on full multiplicands or only on a part of them d is thus recovered with a single correlation

---

. Cross Correlation attack: Can be found in paper named: "Cross correlation attack on RSA" Authors: Witteman, Van Woudenberg, Menarini Employer: Riscure Date of Publication: 2009? Against a counter measure: Square-and-Multiply-Always Countermeasure: Exponent blinding Key words: Square and multiply always, cross correlation Idea: in mono-bit square and multiply always, considering couple of operation: SM, SMd, and MS do not share multiplicands whereas MdS does.

.Correlation during the initial reductions Authors: Amiel, Feix, Villega ?? Employer: Inside contactless, Gemalto Date of Publication: 2007? Countermeasure:

Type of attack: Non-recursive, known plain-text Key words: CRT, modular reduction, Correlation, Multiplicands Idea:

. Correlation during CRT modular exponentiations: Authors: Amiel, Feix, Villega ?? Employer: Inside contactless, Gemalto Date of Publication: 2007? Countermeasure: Square-and-Multiply-Always? Exponent blinding? Type of attack: Non-recursive, known plain-text Key words: CRT, Square-and-Multiply, Correlation, Multiplicands Idea: do correlation on the multiplicand's value to recover simultaneously dp and dq Remark: can append on full multiplicands or only on a part of them Applicable to Barret reduction. Not to Montgomery reduction

---

. Correlation during the CRT recombination: Authors: Amiel, Feix, Villega ?? Employer: Inside contactless, Gemalto Date of Publication: 2007? Countermeasure:

Type of attack: Non-recursive, known plain-text Key words: CRT, Square-and-Multiply, Correlation, Multiplicands Idea: During an exponentiation, for each multiplication (as opposed to squaring), one of the multiplicands is constant

---

. The Big-Mac Attack (Big Mac) Title: "Sliding windows succumbs to big mac attack." Authors: Walter Date of Publication: 2001

. CPA . DPA attack with recombination of bit of p . Attacking Blinded RSA-CRT with Montgomery Multiplication . A Timing Attack against RSA with the Chinese Remainder Theorem . Perturbating RSA Public Keys: An Improved Attack . Side Channel Attack Resistant Implementation of Multi-Power RSA using Hensel Lifting

Without message blinding: ACPA Schindler(2000) CPA of tomoeda 2006 CPA of Primas 2010 KPA of Hlavac lattice ACPA of Ruppeldtova

Messerges, Dabbish, Sloan's Single-Exponent, Multiple-Data (SEMD) DPA Attack Multiple-Exponent, Single-Data (MESD) DPA Attack - weaker than ZEMD ???

1.6 Second order Analysis A refined version of the SPA analysis can be used in order to be able to synchronize two signals having a relationship between each other (for instance, being able to match a decision-making event about

II .2.7.2 Physical threat: fault injection

THE reference: Joye et al. [2012], published in 2012, countains more than 400 bibliographical references

Due to the mathematical complexity of some laser attack, which a complete description would force me to do a lot of 'recall' about abstract maths and would dramatically increase the size of this document, the goal for this part are:

1 - list the known papers, sum-up practical content.
2 - evaluate their efficiency in term of bit(s) per fault.
3 - evaluate their feasibility in term of countermeasures.

Two remarks, as stupids than fundamental:

All attacks by fault injection, assume a *fault model*.
The more unprecise the model is the more realistic the attack.
Many of the fault model are 'impossible' to check.

FI targetting straightforward RSA:

- Boneh - Boneh et al. [1997], *Link*
  aka paper countaining 'The Bellcore attack' , 'The Flip bit attack I'

  Arround this paper: Sorry but there is no Dr Bellcore, 'Bellcore' stands for Bell Communications Research a formerly famous center of research now closed. The Bellcore attack is in reality the 'Boneh-DeMillo-Lipton attack' among the first one to deals with FI.

  Paper's sum-up:
  Present FI attack breaking RSA in its CRT version with one faulty signature and no particular no fault model. Then, authors consider attack Fiat-Shammir scheme, Schnorr's scheme, RSA in RtoL version assuming a fault in some register , breaking those system with a much more larger number of fault.

  Summing up *CRT-RSA's vulnerability to hardware faults* *'The Bellcore attack'*

  Type of attack:
  unknown & reproducible input attack - Dr Bellecore's version
  known & not reproducible input attack - Dr Lenstra's version

  Targets: Every RSA algorithm using the Chinese theorem of the remainders.

  Fault model No particular fault model is assumed, any random value for S'_p respecting the condition $S-S'$ is not divisible by p is suitable for this attack. Following Gauss, the authors decompose the recombination the following way:

  $$S = a \times S_p + b \times S_q$$

  Then, they assume one of the partial encryption were faulted:

  $$S' = a \times S'_p + b \times S_q$$

  Taking in account that statistically $S - S'$ is not divisible by p

  $$q = gcd(S - S', n)$$

  Informed that an important paper was to be published but ignoring the details, Arjen Lenkstra found a version of this attack requiring only one faulty signature and the message. Also exit

another version of this attck by Jorn-Marc Schmidt, in his master repport, finding the key with two faulty signastures and their messages.

Summing up ***Breaking other implementations of RSA*** *'The Flip bit attack I'*

Type of attack: randomly choosen plaintext attack, correct signature not mandatory
reproductible plaintext are not necessary
adaptative recovering algorithm: fault from other model are tolerated.

Targetted: Targets straightforward RSA using the S&M algorithm in LtoR version.

Fault model A single random bit has been flipped in the output register of the S&M algorithm in RtoL version.

Theorem: (Efficiency) With probability at least 1/2 , the secret exponent s can be extracted from a device implementing the first exponentiation algorithm by collecting (n/m)log(n) faults and $O(2^m n^3)$ RSA encryptions for testting motives, for any $1 \leq m \leq n$. For small public exponent d this takes $O(2^m n^4)$ time. For random d it takes $O(2^m n^5)$ time.

Notations:

Variable	Description	Status
$l = \frac{n}{m} log_2(n)$	number of faults	known
$(M_i)_{1 \leq i \leq l}$	Set of random of random messages	known
$(E_i)_{1 \leq i \leq l}$	Set of corresponding signatures	unknown
$(E_i^{'})_{1 \leq i \leq l}$	Set of faulted signatures	known
$(k_i)_{1 \leq i \leq l}$	index of the of fautled loop for $E_i^{'}$	unknown
$s_n s_{n-1} ... s_1$	bit of the secret exponnent	unknown
$s_n s_{n-1} ... s_{k_i}$	bit already guessed	known
$s_{k_i - 1} s_{k_i - 2} ... s_{k_{(i-1)}}$	to guess bits	unknown

Algorithm 27: Boneh's flip bit attack recovering algorithm

Input: $x, n \in \mathbb{N}$, $x \leq n$, $d = d_{t-1} d_1 d_{0_2}$

$y \leftarrow 1$;
for *all lenght* $r = 1, 2, ...$ **do**
  for *all r-bits candidates* $u = u_{k_i - 1} u_{k_i - 2} ... u_{k_i - r}$ **do**
    form full candidate: $\omega = \sum_{j=k_i}^{n} s_j 2^j + \sum_{j=k_i - r}^{k_i - 1} u_j 2^j$;
    test full candidate: $\exists ? e \in \{0, ..., n\} / (E_j^{'} \pm 2^e M_j^{\omega})^d = M_j \mod N$;
    if *yes* **then**
      output: $u_{k_i - 1} u_{k_i - 2} ... u_{k_i - r}$
    if *no* **then**
      reject candidate

Finally, the set of index of the faulted loop for $E_i^{'}$ is assumed to be sorted thanks the natural order, consequence with probability $p > 50\%$, we have: $k_{i+1} - k_i < m$. This section of the article finishes with a proof that false positive*i.e.* wrong candidate that passed the test, are rare.

- Bao - Bao et al. [1998",], *Link*
  aka apper countaining the 'flip bit attack'

  Around the attack: The attack presented by Sciventure is in fact the second one much more realistic in its fault model. One of the very first paper on the subject.

  Paper's sum-up: Attack the RSA algorithm in its straightforward version, the ElGamal signature scheme, the Schnorr signature scheme, and the DSA. RSA is attacked in two different ways: the first attack aim to flip on bit of the message, the other one aiming to flip on bit of the exponent.

  Fault model: Unrealistic: the first fault model - precisely flip one bit in m^{2^i} - appears to be completely unaplicable. Certainly that to flip one bit of an exponent is an (difficult but) achievable objective ...

Type of attack:
randomly chosen & reproducible plain-text attack

Target: S & M algorithm in LtoR and RtoL version.

Counter measure Nothing said bout that, because the authors give their own counter measure to their attack.

Summing up ***Attacking the RSA Scheme*** *'flip bit attack II'*

Notation: let m be the plain-text, c the cyphertext and t and their number of bits, with this, the authors define, which allow them to write the cypher text as a product:

$$\forall i \in [0, t-1] \; m_i = m^{2^i} \mod n$$
$$c = {c_{t-1}}^{d_{t-1}} ... {c_i}^{d_i} ... {c_1}^{d_1} {c_0}^{d_0} \mod n$$

Attack I: the message has been faulted, with a single bit flip:

$$c' = {c_{t-1}}^{d_{t-1}} ... {c'_i}^{d_i} ... {c_1}^{d_1} {c_0}^{d_0} \mod n$$

Then $\frac{c'}{c} = \frac{{c'_i}^{d_i}}{{c_i}^{d_i}}$ can be evaluated, on the other hand can be also calculated the t^2 possible values, if a match is found then i is known, $d_i = 1$.
Limitation
* only one m_i contain one bit of error
* no propagation of error is tolerated, if m is modified, all m_i are

Attack II: Approximately the same faulting only one bit of the exponent.

Limitation
* only one d_i contain one bit of error

- Joye - Joye et al. [1997], *Link*
  aka 'The Flip bit attack III'

  Paper's sum-up:

  This paper is extended the work of Boneh and Boa 'Flipped bit attack I & II' the following way. Fisrt is recalled the previous attacks, then they propose an extension to LUC [7] cryptosystem, KMOV cryptosystem and finally give minors improovment. [8]

- Yen - Joye and Yen [2001], *Link*
  aka 'Safe error attack' -

  Paper's sum-up:
  Authors introduce safe error, the attacker did modify something but it did not have any effect, from this information can be deduced. This attack is extremely generic and can be applied to a lot of situation. On the other there is no way to check that something has been changed.

  Applicability: none.

  Remark this work has been continued by some author distinguishing M safe error and C safe error.

- Schmidt - Schmidt and Herbst [2008], *Link*

  Paper's sum-up:
  Present a FI attack breaking straightforward RSA skipping squarrings. Applicable to most the exponentiation algorithm.

[7] LUC is a public-key cryptosystem developed by a group of researchers in Australia and New Zealand. The cipher implements the analogs of ElGamal -LUCELG-, Diffie-Hellman -LUCDIF-, and RSA -LUCRSA- over Lucas sequences.
[8] KMOV is an elliptic curve based analogue to RSA

Summing up *Attack* LtoR S&M

Fault model: be able to skipp a determined squaring.

Targetted: RtoL & LtoR: S&M, S&MA, 2^k-arry method, sliding window.

Type of attack: random known plaintext attack & reproducible input attack, correct signature mandatory.

Counter measures Ineffective: Square & Multiply always. Effective: the authors are claiming that they can overcome 'most of SPA countermeasure' by skipping the phase where this counter measure is applied. With their super fault model this trivial: -'each time that I want to skipp an operation it works'-. Practically, 'most of SPA countermeasure' defeats this attack.

Initialization:
get Sig_0 -skip the last squaring- and the correspondent non faulted signature.
Then, if the last square has been genuinely skipped, we shall have:

$$Sig = \begin{cases} {Sig_0}^2 \mod n & \text{for } e_0 = 0 \\ {Sig_0}^2 \times m^{-1} \mod n & \text{for } e_0 = 1 \end{cases}$$

Do this operation till the previous equation has been verified and then e_0 deduced.

Induction:
Then when all the first $k-1$ bit of the exponent has been obtained, if the right square has been genuinely skipped, we shall have:

$$Sig_k = \begin{cases} Sig_{k-1} \mod n & \text{for } e_{k-1} = 0 \\ Sig_{k-1} \times m^{2^{k-1}} \mod n & \text{for } e_{k-1} = 1 \end{cases}$$

- Boreale - Boreale [2006], *Link*

Paper's sum-up:
Present a FI attacks breaking straightforward RSA in RtoL version using the Jacobi Symbol, using a practical fault model: that external perturbation, or glitch, may cause a single modular multiplication to produce a truly random result. Two attacks are presented, the second one having relaxed condition.

Type of attack: known & reproducible plaintext attack .

Targetted: RtoL *S&M* only.

Counter measures:
Unefficient: Blind masking of the message: m replaced by $r^e \times m \mod N$.
Efficient: verify the signature by checking that $S^e = m \mod N$, exponnent masking, various delays, modulus blinding-?-.

Fault model: Practical: change the result of a certain multiplication for a random one.

Mathematical background: Jacobi symbol generalizes Legendre's ones, which value $\left(\frac{a}{p}\right)$, for p prime, means:

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } \exists\, x \neq 0 \in \mathbb{Z}/p\mathbb{Z}/a = x^2 \mod p \\ -1 & \text{if } \nexists\, x \neq 0 \in \mathbb{Z}/p\mathbb{Z}/a = x^2 \mod p \\ 0 & \text{if } a = 0 \mod p \end{cases}$$

A link: Wiki about Jacobi & legendre symbols:

Summing up *Attack* LtoR *S&M*

Assumptions:
1- Each modular multiplication/squaring operation takes a constant time, say δ clock cycles, and δ is a constant known to the attacker.
2- Time taken by control-flow instructions is ignored, we view the algorithm as a sequence of modular multiplications. Each phase i takes either δ or 2δ.

3- A glitch applied onto the device during the execution of a modular multiplication will result in a random value $r \in \mathbb{Z}/2^s\mathbb{Z}$ to be written in the involved register in place of the multiplication's correct result.
4- For message m it is assumed that $\left(\frac{m}{N}\right) = 1$, if equal to -1 some equation shall be slighlty modified, the case where $\left(\frac{m}{N}\right) = 0$ in unlikely : it implies $m = p$ or q

The authors begin to define T_i the moment when happend the i^{th} operation while an encryption is performed. As the attacker already knows d_0 the first fault is done around $t = T_1$, more precisely, $T_1 > t > T_1 - \delta$, this will provok a fault in the squarring of the phase $i - 1$.

The obtained signature can be written the following way, using the classical notation $c_i = m^{2^i} \mod n$

$$S' = c_0^{d_0} c_1^{d_1} \ \ldots \ c_{i-1}^{d_{i-1}} (r)^{d_i} (r^2)^{d_{i+1}} \ \ldots \ (r^{2^{l-i-1}})^{d_{l-1}} \mod n$$

Taking in account hypothesis 4, then it is clear that, except $(r)^{d_i}$, each divisor of S' has Jacobi Symbol different from -1. Therefore $\left(\frac{S'}{N}\right) = -1$ implies $\left(\frac{r^{d_i}}{N}\right) = -1$ and then $d_i = 1$. On the other hand to obtain $\left(\frac{S'}{N}\right) \neq -1$ suggest that the more probable is that $d_i = 0$.

Authors finishes this part with an evaluation of the probability $\{d_i = 1 \| \left(\frac{S'}{N}\right) \neq -1\}$.

If the moment of the i^{th} operation is difficult to estimate, the attack is run severasl time ≈ 50.

Software simulation:
On a 768-RSA, 5000 faults are enough to recover, in 30 minutes, the whole key in 70% of the cases.

FI targetting CRT-optimized RSA:

- Coron - Coron et al. [2009a]

  Type of attack:
  reproducible input
  partially known plain-text
  RtoL exponentiation algorithm only
  Result:
  CRT: one faulty encryption is enough
  non CRT: several faulty encryption are required.
  Fault localisation:
  CRT:
  non CRT: one of the two CRT exponentiation
  Relying on:
  Recent improvement of Coppersmith's algorithm to find small roots of multivariate polynomial.
- Amhuller - Fault attack on CRT-RSA concrete result and practical approach - 2007
- Coron - Fault Attacks and Countermeasures on Vigilants RSA-CRT Algorithm - 2010
- Naccache - Modulus fault attack aginst RSA-CRT Signatures - CHES2011
- Fouque - Attacking RSA-CRT signatures with fault Montgommery Multiplication - CHES2012

Electro magnetic fault injection are the most powerful... wtf

II .3 Physical threats and counter measures

II .3.1 Physical threat

Vocabulary: in some part of the literature can be found standard expression like 'exponent blinding' and 'message masking', we will consider blinding and masking as perfect synonymous considering that the expression 'exponent masking' make sense.

an intro better than :

Frequently in so-called 'bullet proof programming' all important parameters such as an RSA modulus can be stored many times with different mask, and after that their integrity have been checked, one modulus of the xored modulus is transformed to an arithmetic masked modulus, and then is ready for use. Algorithms have been setup to securely switch from one type of masking another.

Quiqskatter in the 2000's or more recently: "Debraize - efficient and provably secure methods from switching from arithmetic from boolean masking - CHES2012"[9]

II .3.1.1 Counter measures to side channel

$$\text{Boolean masking: } x = x \oplus m$$
$$\text{Arithmetic masking: } x = (x - m) \bmod 2^k$$

The following countermeasures **rely on the arithmetic masking scheme**: using some classical mathematics identities, the result is computed in a way which is not the most natural neither the quicker.

Transparent Countermeasure: Exponent blinding, ...
Non Transparent Countermeasure: Multiplicative Message blinding, ...

- Exponent blinding -*Coron Ches'1999*- 0%
  The direct computation: $c = m^e \bmod n$ is replaced by a random one:

  $$c = m^{e+r\times\phi(n)} \bmod n$$

  Also known as *'Coron's first Countermeasure'* where r is a random number, the result is not changed thanks to Euler's theorem that is to be changed for each encryption, this countermeasure will impact the number of bits of the exponent, and then the number of visible patterns.

  Practical remarks:

  $$r \textbf{ shall be small to regard to } m\textbf{!}$$

  Remarks: registers and transparency

  - carefully note that if the intermediate computation are reduced modulo n then this countermeasure has strictly no effect. Intermediate computation will only be masked when the register size would have been increased sufficiently.
  - When the masked result has been calculated then the correct result is simply obtain by reduced modulo n. No special operation depending of the mask have to be performed: this is a *transparent mask*.

[9] Do you know any use of a boolean masking in a asymmetrical crypto stuff? If yes communicate!

- Multiplicative Message blinding -*Kocher' 1995*- 0%

  The direct computation: $c = m^e \mod n$ is replaced by a random one:

$$\begin{aligned} m^{'} &= m \times r \mod n \\ c^{'} &= m^{'e} \mod n \end{aligned}$$

  Non transparent countermeasure: recover the original message need an extra operation

$$c = c^{'} \times r^{-e} \mod n$$

  Practical remarks:

  r and m share the same size !

Remarks

- Computational cost of 'random numbers'
  Generation of a random mask r and the extra exponentiation are expensive. Some manufacturer prefer to cheat a bit generating small numbers and from them creating the 'random number' expected, thanks to some available transformation. (a 32 bytes 'random R' can be generate by example with $R = r^k \mod n$ from to really random number of two bytes each)
- Note that a choice is possible to remove the mask at then end of the signature forge or while verifying this one if the mask is available see Schaum's blind signature.
- an extra operation is required to recover the original signature: non transparent mask !

- Additive Message blinding -- 0%
  The message is replaced by a random one:

$$m^{'} = m + r \times n \mod (k \times n)$$

To obtain the desired signature a extra reduction is necessary:

$$c = c^{'} \mod n$$

Practical remarks:

r shall be small to regard to m !

- Exponent splitting. pick-up a random r and form $r^* = e - r$ and compute separately :

$$S_r = m^r \mod n \text{ and } S_{r'} = m^{r^{'}} \mod n,$$

Finally

$$S = S_r \times S_{r'}$$

Practical remarks:

r and m share the same size !

Splitting is considered to be a very secured solution against side channel:

- the implementation shall be SPA resistant
- r shall be small in regard of e

- Modulus blinding -*C.Giraud ' 2006*- 0%
  The modulus replaced by a random one:

$$n^{'} = k \times n$$

Once that the final result is obtained it is then reduced modulo n to unmask the result. Where r is a random number, the result is not changed thanks to Euler's theorem. If r is changed frequently, on a Square and Multiply Always implementation, this countermeasure will impact the number of bits of the exponent, and then the number of burst.

- Multiplicative Message blinding -*kocher ??*- 0%
  The direct computation: $c = m^e \mod n$ is replaced by a random one:

$$c^{'} = (r \times m)^e \mod n$$

Exponentiate as usual, then remove the mask applying: $f : x \to x/r^e \mod n$

$$c = (r \times m)^e / r^e \mod n$$

Remarks:

- where r is a random number, under the following conditions exponent splitting is considered to be a very secured solution against side channel: the implementation shall be SPA resistant and both new exponents $e - r$, practically r is always small.
- the exponent can be split in much more than 2 parts...
- this double the time required to obtain the exponentiation...

Important variants: The famous 'Schaume Blind Signature'
This way to use this countermeasure, is more used for protocol reasons. The interest is to have a protocol allowing a person to sign a blinded message: the signature is genuine but the real data signed is blinded.

$$c^* = (r^d \times m)^e \mod n = r \times m^e \mod n$$
$$m = c^{*d} \times r^{-1}$$

II .3.1.2 Counter measures side channel: slowing down the exponentiation

Algorithms presented in the previous subsection were only interested in being fast, without particular concern about the side channel security, algorithms hereafter presented favourize the security on the efficiency.

- Randomized exponentiation -*??*- 5%
  The idea of this counter measure is to change regularly the type of algorithm used for exponentiation.

  By example to use the binary method in LtoR and RtoL version 'alternatively', using a Random Number Generator.

- Montgomery ladder technique -*Messerge-Dabish-Sloan Ches'1999*- 0%
  The idea of this counter measure is to change regularly the type of algorithm used for exponentiation. By example to use the binary method in LtoR and RtoL version 'alternatively', using a Random Number Generator.

- Overlapping window methods -*Itoh, Yajima, Takenaka, Tori Ches'2002*- 0%
  The idea of this family of counter measures is to use the 2^k-ary method in a redundant way. In the normal 2^k-ary method when a window is scanned and that the relative computation took place the window is shift of k bits. Here the idea is to shift the window of less than k bits.

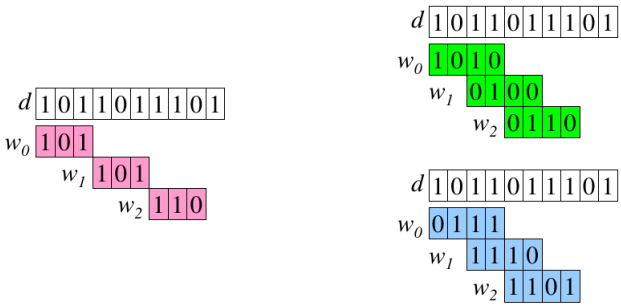


Figure II .8: Overlapping window methods illustration

Let's assume that a 2^5-ary method with an overlapp of 2 bits: the window is each time shifted of 3 bits instead of 5. During the first exponentiation, the three first bits will be read and stored normally in the window but the two last bits will be two random ones, and then the

exponentiation of this window would take place. At the next round of the exponentiation, this random will be compensate, and so on.

- Clavier's square always method -*Clavier 2011*- 0%
Revisiting the Babylonians method replacing multiplications by squarings, thanks to an old identity :

$$x \times y = (\frac{x+y}{2})^2 - (\frac{x-y}{2})^2$$

It also has the advantage to be easily parallizable and it is always faster than Montgomery ladder technique. In practice only atomic version of this technique should be used, can be optimized with a the sliding window method.

VS side channel cryptanalysis
immune against SvsM discrimination

II .3.1.3 Counter measures to fault injection

- Shamir's trick -*1999*- 0%
A probabilistic test - *e.g.* not always working - detecting if the forging has been faulted, the probability of detection increase with the length of r.

Algorithm 28: Shamir's trick *'Extend and reduce modulus'*

Input: m, d, r, r_p , r_q , n
Output: $y = x^d \mod n$, if genuine

```
1 S_{r×p} = m^e mod r × p ;
2 S_{r×q} = m^e mod r × q if S_{r×p} ≠ S_{r×q mod r} then
3     abort
4 return S = CRT(S_{r×p}, S_{r×q});
```

Remarks

- Require the knowledge of d -and not d_p, d_q-
- Does not detect fault during the recombination stage
- significantly longer exponentiation

- A faster countermeasure -*1999*- 0%
A probabilistic test - *e.g.* not always working - detecting if the forging has been faulted, the probability of detection increase with the length of r.

Remarks

- Require the knowledge of d -and not d_p, d_q-
- Does not detect fault during the recombination stage

Algorithm 29: A faster countermeasure

Input: m, d, r, r_p , r_q , n
Output: $y = x^d \mod n$, if genuine

$S^{'} = CRT(S_{r\times p}, S_{r\times q})$;
$r = rnd(32\text{bits})$;
$S = S^{'} \mod n$;
$S_j = S^{'} \mod j$;
$S_{p_r} = m^{d_p \mod r-1} \times j$;
$S_{q_r} = m^{d_q \mod r-1} \times j$;
$S_j^{'} = CRT(S_{p_r}, S_{q_r})$;
if $S^{'} \neq S_j^{'} \mod n$ **then**
  abort
return $S \mod r$;

- Joye & Ciet's trick' -*2002*- 0%
  A probabilistic test - *e.g.* not always working - detecting if the forging has been faulted, the probability of detection increase with the length of r.

Algorithm 30: A faster countermeasure

Input: m, d, r, r_p , r_q , n
Output: $y = x^d \mod n$ if genuine

$S^{'} = CRT(S_{r\times p}, S_{r\times q})$;
$r_1 = rnd(32\text{bits})$, $r_2 = rnd(32\text{bits})$;
$S = S^{'} \mod n$;
$S_j = S^{'} \mod j$;
$S_p^{\star} = m^{d_p} \mod p \times r_1, S_p^{'} = m^{d_p \mod \phi(r_1)} \mod r_1$;
$S_q^{\star} = m^{d_q} \mod q \times r_2, S_p^{'} = m^{d_p \mod \phi(r_2)} \mod r_2$;
if $S_p^{\star} \neq S_p^{'} \mod r_1$ *or* $S_q^{\star} \neq S_q^{'} \mod r_2$ **then**
  abort
return $S \mod r$;

Remarks

  - Does not require the knowledge of d -and not d_p, d_q-
  - Detect fault during the recombination stage

- Masked Montgomery ladder technique -*Fumaroli 2006*- 0%
  Exactly the same algorithm than the previous one but with a random value added. Exist also in a side channel analysis and fault attack resistant version: a checksum is initiated, recalculated at the end of each loop and removed at the end of the algorithm.

Chapter III

About template attacks

Original paper, presenting Template Attacks on stream cyphers with Point Of Interest
Chari et al. [2002], *Link*

III .1 The original template attack

Template attacks[1] have been introduced by Chari, Rao, and Rohatgi, from IBM research, in 2002, originally to attack stream-cypher and block-cypher cryptosystems and being able to break a system with very few recorded traces, possibly one.

Belonging to the family of side-channel attacks as it exploits the fact that power consumption of depend on the data processed, it is however very different from SPA/DPA and their variants of all order as template attack is a *two steps attack*.

First step, the setup phase:

A template attack begins exactly as DPA attacks does by selecting a target: a variable that appears *or not* during the computation and that will be written as $h = h(m, k)$ a function of the plain-text and the key. Another parallel with standards DPA attacks, is that exhaustive enumeration of the values taken by this targeted variable will have to be practical in this attack.

For each of these possible values, many traces shall be recorded -around 1000- so that a 'template' could be built, characterizing with precision the noise of this packet by a multivariate normal distribution the noise that can be observed. This elaborated mathematical model, permit to approach a more complex leakage model than the linear one, as DPA/CPA is usually doing.

Note that in this description already lies several attacks in the definition of h, all with different feasibilities, we can cite the all inputs of an S-boxes, only the keys related to this S-box, or the output of an S-box, or its hamming weight, etc... Depending on this definition there will have more or less templates to built and then evaluate: and without surprise the more templates there is the more powerful the attack is but also the less practicable the attack becomes...

To initiate the setup phase the definition of h have to be chosen. Then recorded traces will be sorted depending on the particular value $\hat{h}$ that h has for each of them. Note that to characterise efficiently the distribution of the noise associated to one of the packet thereby defined each packet shall have its elements the most 'randomly chosen' [2].

[1] Medwed and Oswald [2008], *Link* , Schmidt and Kim [2008], *Link* , Herbst and Medwed [2008], *Link*
[2] a clearer definition of this meaningless expression is possible

The template associated to the particular value $\hat{h}$ of the variable h, is the pair $\mathcal{T}_{\hat{h}} = (m_{\hat{h}}, \Sigma_{\hat{h}})$, where:

- $m_{\hat{h}}$ is the average of the $\hat{h}$-packet
- $\Sigma_{\hat{h}}$ is built upon the noise covariance matrices of the $\hat{h}$-packet.

Wikipedia - *Covariance matrix* :
let's $\vec{X}$ be a vector composed of n random variables $(X_i)_{1 \leq i \leq n}$, each of them having a finite variance, then the covariance matrix $\Sigma_{\vec{X}}$ is a $n \times n$ matrix and is defined by is defined by its components:

$$\text{on the diagonal } \Sigma_{\vec{X}} = var(\vec{X})$$
$$\text{out of the diagonal } \Sigma_{\vec{X}}(i,j) = Cov(X_i, X_j)$$

Where :

$$Cov(X_i, X_j) = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]$$

THEN: Keeping the same notations, to compute the covariance matrix $\Sigma_{\hat{h}}$, requires to define :

$$X_1 = \{ \text{ first points of all traces } \}$$
$$X_2 = \{ \text{ second points of all traces } \} ...$$
$$\text{defining } X \text{ as a vector with } poi \text{ elements}$$

The matrix $\Sigma_{\hat{h}}$ shall be a $poi \times poi$ matrix, because of size of $t - m_h$ (poi) in the following formulas.

Second step, the attack phase:

At this point, after the setup phase, what we manage to built a formula to evaluate the conditional probability $\mathscr{P}$(A|B) of an event A,'a trace is recorded' , happening if B, '$\mathcal{H} = h$', is certain.
A trace t is recorded:

$$\mathscr{P}(t|h) = \frac{1}{\sqrt{(2\pi)^p |\Sigma_h|}} \exp^{(-\frac{1}{2}(t-m_h)^t {\Sigma_h}^{-1}(t-m_h))}$$

A noise x is recorded:

$$\mathscr{P}(x_h|h) = \frac{1}{\sqrt{(2\pi)^p |\Sigma_h|}} \exp^{(-\frac{1}{2}{x_h}^t {\Sigma_h}^{-1} x_h)}$$

Where:

- $|\Sigma_h|$ is the determinant of the Σ_h matrix
- ${\Sigma_h}^{-1}$ is the inverse of the Σ_h matrix

The Maximum likelihood principle:

When the attacker have a trace t he/she then evaluate the probability $\mathscr{P}(t|h)$ all the possible value of h. The output is constituted of all the value k, linked to h such that the corresponding probability are sorted form the most probable to the less one.

The main interest of this attack is that the two steps can be done on two different, but identical, chips. The setup phase can then be achieved on a clone of the targeted device, and in a second time only the attack phase is performed on the targeted device, with very few power consumption records.

Proposition:
Under the Gaussian assumption and if only recorded trace, noted t, is available, the maximum likelihood principle while applied to two equally possible hypothesis simplifies to the following comparison:

$$(t - m_{h0})^t {\Sigma_{h0}}^{-1} (t - m_{h0}) - (t - m_{h1})^t {\Sigma_{h1}}^{-1} (t - m_{h1}) \leq \ln(|\Sigma_{h0}|) - ln(|\Sigma_{h1}|)$$

where a decision is made in favor of H_1 if the above inequality is true and in favor of H_0 otherwise.

III .2 Practical improvements

The previous attack is very powerful indeed due to the elaborated mathematical model underlying but also absolutely impracticable in the real world, even with good computers for the same reason. Hereafter are listed the two way of making Template Attack more feasible and finally a remark on the importance of signal processing before building the templates.

- Point Of Interests: '*reduced traces to some specific points*'
  The idea is to perform the attack not on the whole traces but only on traces reduced to few decades of interesting points. For each value of the selected variable has been recorded a packet of n_i traces, of average $\mu_i(t)$ and of variance $\sigma_i(t)$, then different functions of time can be considered to define Points of interest with the abscissas of their highest pikes, sorted in term of efficiency:
  - Chari*&al.* in Chari et al. [2002], *Link* difference of average signals: First proposed method for selection of points for some i and j select only the points where large difference shows up.
    $$d(t) = \mu_i(t) - \mu_j(t)$$
  - Rechberger*&al.* in Rechberger and Oswald [2008], *Link* Sum Of Difference of average signals: Filter some noise but positive and negative quantity compensate each other and hide informations. They also showed the crucial importance of two parameters to choose point of interest: a minimum distance of one clock cycle and a heigh greater than the noise floor.
    $$sod(t) = \sum_{i<j}(\mu_i(t) - \mu_j(t))$$
  - Gierlichs*&al.* in Gierlichs et al. [2006], *Link* : Sum Of Squared Difference of average signals:
    solve the previous problem but the noise is more present also, small contribution are crushed.
    $$sosd(t) = \sum_{i<j}(\mu_i(t) - \mu_j(t))^2$$
  - Agrawal*&al.* in Agrawal et al. [2003], *Link* : Sum of squared t-values:
    This method seems to be the chosen one in most of the cases nowadays.
    $$sost(t) = \sum_{i<j} \frac{(\mu_i(t)-\mu_j(t))^2}{(\frac{\sigma_i^2}{n_i}+\frac{\sigma_j^2}{n_j})(t)}$$
    where n_i and n_j are size of the different packet.
- Principal Component Analysis: '*analyse only most importnant part of* Σ_h'
  Archambau *&al.* in Archambeau et al. [2006], *Link* published an article to apply the famous statistical method called Principal component analysis which technique reduces the dimension of the covariance matrix by projection into the subspace spanned by the eigenspaces of the highest eigenvalues.
- Signal processing and acquisition: *'analyse transformed traces'*
  - Rechberger*&al.* in Rechberger and Oswald [2008], *Link* In a practical study advised to perform template attacks not on the time domain but on the frequency domain with significant improvement of the result, especially for noisy traces.
  - El Aabid*&al.* in Elaabid and Guilley [2012], *Link* In a practical study showed the crucial importance of two parameters: the chronological synchronization and the vertical scale. They shall be the same for all traces.
- Reduced matrix: *'definition of the matrix* Σ_h *can be simplified'*

  1- Fill only the diagonal, computation of Σ_h^{-1} trivial

  2- Stochastic attack, see section III .6.

Normally nowadays every template attacks shall take in account those approaches.

III .3 Template based DPA attacks:

Two ways to turn template attacks to more DPA like attack, *i.e.* to attack recovering bits of the keys from 'a lot' of traces. The first one is applying template attacks to several available traces the second one is skipping the setup phase and give a new metric for DPA.

III .3.1 Template attack with several traces

Bayes' theorem allows us to evaluate the probability of the event "the sub key used was k given that x is recorded".

$$\mathscr{P}(\mathcal{H} = \hat{h}|t) = \frac{\mathscr{P}(t|\hat{h})\mathscr{P}(\hat{h})}{\sum_j \mathscr{P}(t|h_j).\mathscr{P}(h_j)}$$

It also permit us to consider the case of a set of traces $\mathbf{T}$ available during the attack phase:

$$\mathscr{P}(\mathcal{H} = \hat{h}|\mathbf{T}) = \frac{\left(\prod_{i=1}^{D} \mathscr{P}(t_i|\hat{h})\right).\mathscr{P}(\hat{h})}{\sum_{l=1}^{H}\left(\left(\prod_{i=1}^{D} \mathscr{P}(t_i|h_l)\right).\mathscr{P}(h_l)\right)}$$

III .3.2 DPA-Template attack

The classical DPA-decision metric can be improved thanks the notion of template even is no template is actually build using the inequality mentioned previously.

Let H_i be one of the considered hypothesis by a DPA attack, and H_v the value of the selection function. To those two equally possible hypothesis can be applied the inequality presented earlier. Then the obtained metric is not evaluable because two parameters are not known problem solved by giving an estimation of those two.

This attack is among the most powerful side channel attack, because it can efficiently adapted to makes algorithms, this is this attack, in its naive version, that Inspector implemented note however that the selection of the point of interests is critical.

III .4 Template attacks on symmetrical algorithms

What has been published

- In 2003, Chari*&al.* in Chari et al. [2002], *Link* gave the first description of a two steps side channel attacks, with elaborated model for noise.
- In 2003, Agrawal*&al.* in Agrawal et al. [2002], *Link* improved significantly the template attack combining multiple side channel such as power and EM simultaneously. They also improved the DPA attack defining a new metric by using the Gaussian assumption, turning the DPA attack to a two steps attacks and if the setup phase was impossible to perform they give a way to

approximates this one. Quoting Elisabeth Oswald 'Template based DPA attack constitute the strongest the strongest kind of DPA attack.

- In 2005, Agrawal*&al.* in Agrawal et al. [2005], *Link* defined the 'single bit template attack' where the targeted variable is a single bit and the 'template enhanced DPA attack' mixing template and DPA attack (Warning) They also break a masked DES and AES basically building template with a chip with a biased RNG and then exploiting those on the same chip with a perfect RNG.

- In 2006, Mangard*&al.* in **?**, *Link* the authors claim that 'in the scenario of template attacks, masking does not improve the security of an implementation... ' They used template based DPA attacks to attack masked version of the DES and AES, reference to the masks in Akkar and Giraud [2001], *Link* and Blomer et al. [2004], *Link* , with devastating conclusions, if an biased PRNG is available during the setup phase! The attack phase just this line is changing.

$$\mathscr{P}(t|h_j) = \sum_j \mathscr{P}(t|h_j \wedge m).\mathscr{P}(m)$$

In this same paper, combination of HODPA and template attack is also studied: to unleash the maximum of correlation in a 2^{nd} order attack or to force a bias in the collected traces.

- In 2007, El Aabid*&al.* in Aabid et al. [2007], *Link* claimed that the real target of template attack was the key schedule. Instead of the naive definition of h sorting the traces depending on value of the sub-key used, he improved the sorting using the following functions:
$h = k$
$h = k \oplus LS(k)$
$h = k \oplus LS^2(k)$
where LS is the left shift function used in the key schedule of the DES algorithm. In this article they recognize that the first function is suitable for a 'blind' attacker

- **?**, *Link* .

Definitions for h

Here only the case of the DES algorithm is considered and *Input* and *Output* will represent the respective input and output of some S-box.

$h = Input$	2^{12} templates to build
$h = k$	2^6 templates to build
$h = k \oplus LS(k)$	2^6 templates to build
$h = k \oplus LS^2(k)$	2^6 templates to build
$h = Output$	2^4 templates to build
$h = \omega_{\mathcal{H}}(Input)$	5 templates to build

Matrix

	Naked DES	*Splited DES*	*Masked DES* [3]	*Masked DES* [4]
References				

●				Chari et al. [2002]
●				Agrawal et al. [2002]
●		○		Rechberger and Oswald [2008]
●		●	?	Agrawal et al. [2005]
●				Aabid et al. [2007]

[3]The transforming masked method Akkar and Giraud [2001], *Link* (one mask)
[4]The transforming masked method Akkar and Giraud [2001], *Link* (two masks)

III .5 Template attacks on asymmetrical algorithms

In this section we only consider the two most used asymmetrical algorithms, namely RSA and ECC, because of their common point that is they share a family of algorithms to perform the central operation on which mainly depends their security.

This is modular exponentiation for RSA and scalar multiplication for ECC, both taking place in some finite fields. In RSA the the first of those algorithms is the well known square-and-multiply algorithm to which is corresponding the double-and-add algorithm for elliptic curves.

What has been published

- In Medwed and Oswald [2008], *Link* is showed the feasibility of template attacks on asymmetrical algorithm: on a unmasked ECDSA implemented with the double-and-add-always algorithm and also with the sliding window version of this algorithm. They gave also conditions so their attack to work with scalar blinding [5] and for Point Blinding [6]. A counter measure is to randomize the base point [7].
- In Herbst and Medwed [2008], *Link* is showed the feasibility of template attacks on masked asymmetrical algorithm: on a RSA implemented with a masked version of the Montgomery ladder exponentiation.
- In Amiel et al. [2009], *Link* is showed a theoretical presentation of template attacks on atomic versions of the double-and-add algorithm and of the square-and-multiply algorithm. In this article is said that a device with a high level of side channel leakage, even with a masked exponent, could be vulnerable to their attack.
- In Hanley et al. [2011b], *Link* is showed an extension of the previous article giving a practical presentation of template attacks on *atomic* versions of the double-and-add algorithm and of the square-and-multiply algorithm. Plus, the attack described in this paper do not require any open device to built its template.
- Schindler [2011], *Link* revisit the paper of P.A Fouque "Power attack on small RSA public exponent" which can if some bits exponent are known the recover the all exponent, to make it more error tolerant and then more practical. Case of Square and always Multiply or for Double and always Add.

III .6 The stochastic attack

Schindler*&al.* in Schindler et al. [2005], *Link* publish a variant of template attack: the Stochastic attack also known as the regression model. This attack find the key answering to the following question: 'among all the linear leakage models that can be build with N simulations of the targeted variable and the recorded trace, which in the end corresponds the best to the recorded trace ?'

First let's assume that the deterministic part of the leakage is simply $\delta(x) = \alpha_{-1} + \sum_{i=0}^{n} \alpha_i . x_i$ and that the target variable h got n bits, and that $\left(l_{\hat{h},i}\right)_{1 \leq i \leq N}$ is one measure set of N measurements.

$$L = \begin{bmatrix} l_{h,1} \\ \vdots \\ l_{h,N} \end{bmatrix}$$

[5] corresponding to exponent blinding in RSA
[6] corresponding to message blinding in RSA
[7] corresponding to nothing in RSA

Now for every possible value $\hat{h}$ of h let's assume that $\left(v_{\hat{h},i}\right)_{1\leq i\leq N}$ are the N corresponding hypothesis about the deterministic part of the leakage. And let's take a look at the contribution of each bits of v:

$$M = \begin{bmatrix} 1 & v_{\hat{h},1}[0] & \dots & v_{\hat{h},1}[n-1] \\ & \vdots & & \vdots \\ 1 & v_{\hat{h},N}[0] & \dots & v_{\hat{h},N}[n-1] \end{bmatrix}$$

The leakage model given by the hypothesis '$h = \hat{h}$' is:

$$\alpha_{\hat{h}} = (\alpha_{\hat{h},-1}, \dots, \alpha_{\hat{h},n-1}) = (M^{\top}.M)^{-1}.(M^{\top}.L)$$

And the signal of decision for this hypothesis is:

$$\Delta_{\hat{k}} = \frac{|L - M.\alpha_{\hat{k}}|_2}{\sqrt{Var(L)}}$$

Then the most probable value for h is the one minimising $\Delta_{\hat{k}}$

Question: some claim that the stochastic method is just a normal template attack replacing the covariance matrix for the identity matrix... ?

III .7 The power consumption model & notations

The classical presentation is the following one. It is assumed when considering a sensitive variable V_h [8] , the leakage, L to be composed of two parts: a deterministic part, δ and the noise independent from V_h, *i.e.* independent from the plain text and the key.

$$L_h = \delta(V_h) + B$$

Then if N measurements are done, the previous equation implies:

$$\forall i \leq N, \quad l_{h,i} = \delta(v_{h,i}) + b_i$$

One of the more general *symmetrical* model for approaching the function δ of a variable x of d is given by:

$$\delta(x) = \alpha_{-1} + \sum_{i=0}^{n} \alpha_i.x_i + \sum_{i_1 \neq i_2 = 0}^{n} \alpha_{i_1,i_2}.x_{i_1,i_2} + \ldots + \sum_{i_1 \neq \ldots \neq i_d = 0}^{n} \alpha_{i_1,\ldots,i_d}.x_{i_1,\ldots,i_d}$$

Hypothesis to simplify this model:

-LID: Leakage Interpolation Degree:
A good approximation of δ can be obtained with a polynomial of smaller multivariate degree, $n < d$.

-IBL: Independent Bit Leakage:
A good approximation of δ can be obtained with a linear function, $n = 1$.

-EHQ: Equivalent Homogeneous Contribution:
A good approximation of δ can be obtained assuming that each homogeneous polynomial constituting δ have independently the same coefficient.

Questions :
DPA/CPA etc IBL ?, EHQ ?
Template attacks LID with n=2? EHQ ?

[8] This notation ...

Chapter IV

Latex definition & tricks

NIST [2009], *Link* Staff [2002], *Link* Des [2001], *Link* Cetricom [2000], *Link* of Standards et al. [2000], *Link* NIST [1977], *Link* ISO/IEC/BSI [1999], *Link* Christopher [2012], *Link* Elaabid [2012], *Link* Verneuil [2012], *Link* Marcel [2012], *Link* Timmerman [2011], *Link* Timmerman [2011], *Link* Gierlichs [2011], *Link* Papachristodoulou [2010], *Link* Hogenboom [2010], *Link* Shah Kruti R. [2010], *Link* Timmerman [2011], *Link* Berzati, *Link* ?, *Link* Waldyr [2008], *Link* Clavier [2007], *Link* DeCanniere [2007], *Link* Sylvain [2007], *Link* Giraud [2007], *Link* Lemke-Rust [2007], *Link* Patrick [2007], *Link* Piret [2005], *Link* Muir [2004], *Link* Bevan [2004], *Link* Marcel [2003], *Link* Hasselstrom [2003], *Link* E.Oswald [2003], *Link* ?, *Link* Batina [2003], *Link* Messerges [2000], *Link* Dehem [2000], *Link* Kenneth, *Link* Kenneth, *Link* Joye, *Link* Sedgewick and Wayne [2012], *Link* Joye et al. [2012], *Link* van Tilborg and Jajodia [2011], *Link* Tahe [2008], *Link* Andre [2012], *Link* Mangard et al. [2006], *Link* Stern et al. [2004], *Link* Goldreich [2012], *Link* Hankerson et al. [2012], *Link* Goldreich [2001b], *Link* Goldreich [2001a], *Link* Rankl and Effing [2000], *Link* Schneier [2000], *Link* Silverman [2000], *Link* Barth [1994], *Link* Menezes [1996], *Link* Koblitz [1991], *Link* Mukhopadhyay [2015], *Link* CHES [2015], *Link* ?, *Link* Bond et al., *Link* Heyse and Guumlneysu [2012], *Link* Ruohrmair and van Dijk [2012], *Link* Fei et al. [2012], *Link* Reparaz et al. [2012], *Link* Schloumlsser et al. [2012], *Link* Kerckhof et al. [2012], *Link* Lee et al. [2012], *Link* Gottert et al. [2012], *Link* Debraize [2012], *Link* Debraize [2012], *Link* Cheng et al. [2012], *Link* Sarkar and Maitra [2012a], *Link* Bernstein and Schwabe [2012], *Link* Czypek et al. [2012], *Link* Faust et al. [2012], *Link* Bilgin et al. [2012], *Link* Guneysu et al. [2012], *Link* Knezevic et al. [2012], *Link* Debraize [2012], *Link* Fouque et al. [2012], *Link* Briais et al. [2012], *Link* Moradi and Mischke [2012], *Link* Moss et al. [2012], *Link* van der Leest et al. [2012], *Link* Vielhaber [2012], *Link* Gerard and Standaert [2012], *Link* Maes et al. [2012], *Link* Banik et al. [2012], *Link* Medwed et al. [2012], *Link* Vielhaber [2012], *Link* Matsuda and Moriai [2012], *Link* Oren et al. [2012], *Link* Goubin and Martinelli [2011], *Link* Kim et al. [2011], *Link* Clavier et al. [2011], *Link* Fan et al. [2011], *Link* CHES [2011], *Link* Tunstall and Joye [2010], *Link* Rivain and Prouff [2010], *Link* Berzati et al. [2010], *Link* Longa and Gebotys [2010], *Link* CHES [2010], *Link* Batina et al. [2009], *Link* Renauld et al. [2009], *Link* Coron et al. [2009a], *Link* Coron and Kizhvatov [2009], *Link* CHES [2009], *Link* Baddam and Zwolinski [2008], *Link* CHES [2008], *Link* Bogdanov et al. [2007], *Link* Paillier and Verbauwhede [2007], *Link* Le et al. [2006], *Link* Prouff et al. [2006], *Link* Stebila and Theriault [2006], *Link* Archambeau et al. [2006], *Link* Brier et al. [2006], *Link* Fouque et al. [2006], *Link* Gierlichs et al. [2006], *Link* CHES [2006], *Link* Joye et al. [2005], *Link* Agrawal et al. [2005], *Link* Canright [2005], *Link* Peeters et al. [2005], *Link* Popp and Mangard [2005], *Link* Schindler et al. [2005], *Link* CHES [2005], *Link* Hars [2004], *Link* Brier et al. [2004], *Link* Fouque et al. [2004], *Link* Ledig et al. [2004], *Link* Schramm et al., *Link* CHES [2004], *Link* Agrawal et al. [2003], *Link* Coron and Tchulkine [2003], *Link* Fouque and Valette [2003],

Link CHES [2003], *Link* Trichina et al. [2002], *Link* Agrawal et al. [2002], *Link* Chari et al. [2002], *Link* Itoh et al. [2002], *Link* Joye and Yen [2002], *Link* CHES [2002], *Link* Akkar and Giraud [2001], *Link* Clavier and Joye [2001], *Link* Sakurai [2001], *Link* CHES [2001], *Link* **?**, *Link* Coron and Goubin [2000], *Link* CHES [2000], *Link* Goubin and Patarin [1999], *Link* Messerges et al. [1999], *Link* DaRolt et al. [2012], *Link* Hutter et al. [2012], *Link* Shiqian WANG, *Link* Medwed, *Link* Martin, *Link* Akkar et al. [2004], *Link* Akkar and Goubin [2003], *Link* Wheeler and Needham, *Link* Coron et al. [2009b], *Link* Okeya et al. [2004], *Link* Patarin [2003], *Link* Biehl et al. [2000], *Link* Coron et al. [1999], *Link* Kocher et al. [1999], *Link* Koblitz [1998], *Link* Kocher [1996], *Link* Barrett [1986], *Link* Kochanski [1985], *Link* Prouff and Rivain [2013], *Link* Bertoni et al. [2013], *Link* Standaert et al. [2009], *Link* Ciet et al. [2003], *Link* Boneh et al. [1997], *Link* Brickell et al. [1992], *Link* Standaert et al. [2010], *Link* Boneh et al. [1998], *Link* Gierlichs et al. [2012], *Link* Aranha et al. [2011], *Link* Schmidt et al., *Link* Joye and Tunstall [2009], *Link* Clavier et al. [a], *Link* Clavier et al. [b], *Link* Fumaroli et al., *Link* Bauer et al., *Link* Houssem et al. [2012], *Link* Batina et al. [2012], *Link* Witteman et al. [2011], *Link* Michael et al. [2011], *Link* CTRSA [2011], *Link* CTRSA [2010], *Link* Berzati et al., *Link* **?**, *Link* Fischlin, *Link* CTRSA [2008], *Link* **?**, *Link* Abe [2007], *Link* Schramm and Paar [2006], *Link* Oswald et al. [2006], *Link* Pointcheval [2006], *Link* CTRSA [2005], *Link* Fischer and Seifert [2004], *Link* CTRSA [2004], *Link* CTRSA [2003], *Link* CTRSA [2002], *Link* Brown et al. [2001], *Link* CT-RSA [2001], *Link* Gierlichs et al. [2010], *Link* Coron et al. [2010], *Link* Bouffard et al., *Link* Guilley and Pacalet [2004], *Link* Muller and Valette, *Link* Antipa et al. [2003], *Link* Joye and Tymen [2001], *Link* Piret et al. [2012], *Link* Coron [2008], *Link* Le et al. [2007], *Link* Hasenplaugh et al. [2007], *Link* Ross et al., *Link* Delerablee et al. [2011], *Link* Möller [2012], *Link* Pan et al. [2011], *Link* Joye [2009], *Link* Amiel et al. [2009], *Link* Amiel et al. [2007], *Link* Walter [2003], *Link* **?**, *Link* Krämer et al. [2011], *Link* Walker [2000], *Link* Tunstall [2005], *Link* Standaert et al. [2006], *Link* Masmoudi et al. [2006], *Link* Fan et al. [2010], *Link* Park et al. [2012], *Link* Maghrebi et al. [a], *Link* Aumonier [2007], *Link* Hanley et al. [2009], *Link* Herbst and Medwed [2008], *Link* Medwed and Oswald [2008], *Link* Schmidt and Kim [2008], *Link* Miret et al. [2008], *Link* Rechberger and Oswald [2008], *Link* van Woudenberg et al. [2011], *Link* Schmidt and Herbst [2008], *Link* Boreale [2006], *Link* Ciet and Joye [2005], *Link* Regazzoni et al., *Link* de Ruiter and Poll [2012], *Link* Messerges et al., *Link* Messerges et al., *Link* Messerges et al., *Link* Ploog et al. [2001], *Link* Batina et al., *Link* Sidorenko et al. [2012], *Link* Rauzy et al. [2013], *Link* krenn [2012], *Link* Sarkar and Maitra [2012b], *Link* Belenky et al. [2012], *Link* Hanley et al. [2012], *Link* Oren and Wool [2012], *Link* Maghrebi et al. [b], *Link* Hanley et al. [2011a], *Link* Roy et al. [2011], *Link* Clavier et al. [2010], *Link* Dent [2009], *Link* Wang et al. [2008], *Link* Aabid et al. [2007], *Link* Bard et al. [2007], *Link* Clediere et al., *Link* Avanzi et al. [2005], *Link* Avanzi et al. [2005], *Link* Courtois and Bard, *Link* Bernstein [2008], *Link* Brier et al. [2003b], *Link* Lemmermeyer [2003], *Link* Ciet and Joye [2003], *Link* Frederic [2003], *Link* Brier et al. [2003a], *Link* Bogdanov and Kizhvatov [2012], *Link* Montgomery [1985], *Link* Brier et al. [2003b], *Link* Joye and Yen [2001], *Link* Krenn [2012c], *Link* Krenn [2012d], *Link* Krenn [2012a], *Link* Krenn [2012b], *Link* Heuberger [2009], *Link* Bergeron et al. [1994], *Link* Krenn [2012d], *Link* G. [2004], *Link* Blake et al., *Link* Elaabid and Guilley [2012], *Link* Moreno and Hasan [2011], *Link* Joye et al. [1997], *Link* Fürer, *Link* Mavroeidis et al., *Link* Mangard et al. [2011], *Link* Kulikowski et al. [2006], *Link* Dimitrov et al., *Link* Dimitrov et al., *Link* LeGal et al. [2010], *Link* Abdi and Williams [2010], *Link* Heuberger and Krenn [2012], *Link* Booth [1951], *Link* Tunstall et al., *Link* Lemke-Rust and Paar, *Link* Jiye et al., *Link* Bond et al., *Link* Lin, *Link* Karatsuba [1994], *Link* Poulakis, *Link* Kreuzer [2009], *Link* **?**, *Link* Avanzi et al. [2006], *Link* Abrahams [1999], *Link* Oluwatope et al. [2006], *Link* Hanley et al. [2011b], *Link* Papachristodoulou [2009], *Link* Dent et al., *Link* Bernstein [2002], *Link* Bernstein [2006], *Link* Koc [1994], *Link* Standaert et al., *Link* Joye [2012], *Link* Maria et al. [2011], *Link* Solinas [2000], *Link* Shannon [2001],

Link Fortnow and Homer [2003], *Link* Koc and Acar, *Link* Cilio et al. [2013], *Link*

From the natbib package cite: nocite: : the paper is in the biblio, but no link papers in the text
citeauthor: Chari et al.: cite an author name
citeauthor*: Chari, Rao, and Rohatgi: cite all authors name
citet: Chari et al. [2002]: cite an author with the link
citet*: Chari, Rao, and Rohatgi [2002]: cite all author with the link
citep: [Chari et al., 2002]: cite an author with the link
citep*: [Chari, Rao, and Rohatgi, 2002]: cite all author with the link
Chari et al. [2002], *Link*

Bibliography

Proceedings of the 2006 International Conference on Field Programmable Logic and Applications (FPL), Madrid, Spain, August 28-30, 2006, 2006. IEEE.

El Aabid, Guilley, and Hoogvorst. Template attacks with a power model. In ePrint Archive. *Link* .

Herve Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2010. *Link* .

Masayuki Abe, editor. *Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings*, Lecture Notes in Computer Science, 2007. *Link* .

Abrahams. Addition chains as test trees and a sequential variant - 1999. *DIMACS - Technical report*, 1999. *Link* .

Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The em side-channel(s). In CHES [2002]. *Link* .

Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. Multi-channel attacks. In CHES [2003]. *Link* .

Dakshi Agrawal, Josyula R. Rao, Pankaj Rohatgi, and Kai Schramm. Templates as master keys. In CHES [2005]. *Link* .

Mehdi-Laurent Akkar and Christophe Giraud. An implementation of des and aes, secure against some attacks. In CHES [2001]. *Link* .

Mehdi-Laurent Akkar and Louis Goubin. A generic protection against high-order differential power analysis. In *FSE*, 2003. *Link* .

Mehdi-Laurent Akkar, Regis Bevan, and Louis Goubin. Two power analysis attacks against one-mask methods. In *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, 2004. *Link* .

Amiel, Feix, Tunstall, Whelan, and Marnane. Distinguishing multiplications from squaring operations (revised paper). *LNCS 2009, volume 5394, p346-360, Springer*, 2009. *Link* .

Frederic Amiel, Benoit Feix, and Karine Villegas. Power analysis for secret recovering and reverse engineering of public key algorithms. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *Selected Areas in Cryptography*, Lecture Notes in Computer Science, pages 110–125. Springer, 2007. *Link* .

Neubauer Andre. *Coding Theory - Algorithms, Architectures, and Applications*. Springer editor, 2012. *Link* .

Adrian Antipa, Daniel R. L. Brown, Alfred Menezes, Rene Struik, and Scott A. Vanstone. Validation of elliptic curve public keys. In *PKC*, 2003. *Link* .

Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, and Julio López. Faster explicit formulas for computing pairings over ordinary curves. In LatinCRYPT, editor, *Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology*, EuroCrypt'11. Springer-Verlag, 2011. *Link* .

Cedric Archambeau, Eric Peeters, Francois-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In CHES [2006]. *Link* .

Sebastien Aumonier. Generalized differential power analysis. In ECRYPT [2007]. *Link* .

Roberto Maria Avanzi, Clemens Heuberger, and Helmut Prodinger. Minimality of the hamming weight of the $\tau - naf$ for koblitz curves and improved combination with point halving. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Roberto Maria Avanzi, Clemens Heuberger, and Helmut Prodinger. Scalar multiplication on koblitz curves using the frobenius endomorphism and its combination with point halving: Extensions and mathematical analysis. *Algorithmica*, 2006. *Link* .

Karthik Baddam and Mark Zwolinski. Divided backend duplication methodology for balanced dual rail routing. In CHES [2008]. *Link* .

Subhadeep Banik, Subhamoy Maitra, and Santanu Sarkar. A differential fault attack on the grain family of stream ciphers. In CHES [2012]. *Link* .

Bao, Deng, Han, Jeng, Narasimhalu, and Ngair. Breaking public key cryptosystems on tamper resistant devices in the presence of transient faults. *Security Protocols, 5th International Workshop, Paris, April 7-9*, 1998",.

Gregory V. Bard, Nicolas Courtois, and Chris Jefferson. Efficient methods for conversion and solution of sparse systems of low-degree multivariate polynomials over gf(2) via sat-solvers. *IACR Cryptology ePrint Archive*, 2007. *Link* .

Paul Barrett. Implementing the rsa public key encryption algorithm on a standard digital signal processor. In *CRYPTO*, 1986. *Link* .

W. Barth. *Complex Compact Surfaces*. 1994. *Link* .

Lailja Batina. Arithmetic and architectures for secure hardware implementations of public-key cryptography, 2003. *Link* .

Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, Francois-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual information analysis: a comprehensive study. *J. Cryptology*. *Link* .

Lejla Batina, Benedikt Gierlichs, and Kerstin Lemke-Rust. Differential cluster analysis. In CHES [2009]. *Link* .

Lejla Batina, Jip Hogenboom, and Jasper G. J. van Woudenberg. Getting more from pca: First results of using principal component analysis for extensive power analysis. In CTRSA [2012]. *Link* .

Aurélie Bauer, Éliane Jaulmes, Emmanuel Prouff, and Justine Wild. Horizontal and vertical side-channel attacks against secure rsa implementations. Lecture Notes in Computer Science. *Link* .

Yaacov Belenky, Zeev Geyzel, Michael Kara-Ivanov, and Avraham Entelis. Two exponentiation algorithms resistant to cross-correlation power analysis and to other known attacks. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

F. Bergeron, J. B ERSTEL, and S. B RLEK. Efficient computation of addition chains. 1994. *Link* .

Daniel Bernstein and Sanjit Chatterjee, editors. *Progress in Cryptology - INDOCRYPT 2011 - 12th International Conference on Cryptology in India, Chennai, India, December 11-14, 2011. Proceedings*.

Daniel J Bernstein. Pippenger exponentiation algorithm. Preprint. *Link* .

Daniel J Bernstein. Differential addition chains. Preprint. *Link* .

Daniel J. Bernstein. Rsa signatures and rabin-williams signatures: the state of the art. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Daniel J. Bernstein and Peter Schwabe. Neon crypto. In CHES [2012]. *Link* .

Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Keccak. In EuroCrypt [2013]. *Link* .

Alexandre Berzati. White-box cryptography. *Link* .

Alexandre Berzati, Cécile Canovas, Jean-Guillaume Dumas, and Louis Goubin. Fault attacks on rsa public keys: Left-to-right implementations are also vulnerable. In Fischlin. *Link* .

Alexandre Berzati, Cecile Canovas-Dumas, and Louis Goubin. Public key perturbation of randomized rsa implementations. In CHES [2010]. *Link* .

Regis Bevan. Éstimation statistique et sécuritée des cartes à puce Évaluation d'attaques dpa ´voluées, 2004. *Link* .

Ingrid Biehl, Bernd Meyer, and Volker Maller. Differential fault attacks on elliptic curve cryptosystems. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, 2000. URL *Link*. *Link* .

Begull Bilgin, Svetla Nikova, Ventzislav Nikov, Vincent Rijmen, and Georg Stuumltz. Threshold implementations of all 3x3 and 4x4 s-boxes. In CHES [2012]. *Link* .

Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors. *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario,Canada, August 12-13, 2010, Revised Selected Papers*, Lecture Notes in Computer Science, 2011.

Blake, Murty, and Xu Guangwu. Nonadjacent radix-τ expansions of integers in euclidean imaginary quadratic number fields. *Canadian Journal of Mathematics*. *Link* .

Johannes Blomer, Jorge Guajardo, and Volker Krummel. Provably secure masking of aes. In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography*, Lecture Notes in Computer Science, 2004.

Andrey Bogdanov and Ilya Kizhvatov. Beyond the limits of dpa: Combined side-channel collision attacks. *IEEE Trans. Computers*, pages 1153–1164, 2012. *Link* .

Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In Paillier and Verbauwhede [2007]. *Link* .

Mike Bond, Omar Choudary, Steven J. Murdoch, Sergei P. Skorobogatov, and Ross J. Anderson. Chip and skim: cloning emv cards with the pre-play attack. *CoRR*. *Link* .

Dan Boneh, Richard DeMillo, and Richard Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In EuroCrypt [1997]. *Link* .

Dan Boneh, Glenn Durfee, and Yair Frankel. An attack on rsa given a small fraction of the private key bits. In *ASIACRYPT*, 1998. *Link* .

Andrew D. Booth. A signed binary multiplication technique. 1951. *Link* .

Michele Boreale. Attacking right-to-left modular exponentiation with timely random faults. In FDTC, editor, *FDTC*, Lecture Notes in Computer Science, 2006. *Link* .

Antoon Bosselaers, Rene Govaerts, and Joos Vandewalle. Comparison of three modular reduction functions. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, 1993. *Link* .

Guillaume Bouffard, Tom Kefif, Jean louis Lannet, Ismael Kane, and Sergio Casanova. Accessing secure information using export file fraudulence. *Link* .

Sebastien Briais, Stephane Caron, Jean-Michel Cioranesco, Jean-Luc Danger, Sylvain Guilley, Jacques-Henri Jourdan, Arthur Milchior, David Naccache, and Thibault Porteboeuf. 3d hardware canaries. In CHES [2012]. *Link* .

Ernest F. Brickell, Daniel M. Gordon, Kevin S. McCurley, and David Bruce Wilson. Fast exponentiation with precomputation (extended abstract). In *Advances in Cryptology - EuroCrypt '92, Workshop on the Theory and Application of of Cryptographic Techniques, Balatonfared, Hungary, May 24-28, 1992, Proceedings*, 1992. *Link* .

Eric Brier, Christophe Clavier, and Francis Olivier. Modeling of digital substrate noise generation and experimental verification using a novel substrate noise sensor. ePrint Archive. *Link* .

Eric Brier, Christophe Clavier, and Francis Olivier. Optimal statistical power analysis. ePrint Archive. *Link* .

Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In CHES [2004]. *Link* .

Eric Brier, Benoit Chevallier-Mames, Mathieu Ciet, and Christophe Clavier. Why one should also secure rsa public key elements. 2006. *Link* .

M. Brown, D. Hankerson, J. Lopez, and A. Menezes. Software implementation of the nist elliptic curves over prime fields. In CT-RSA [2001]. *Link* .

David Canright. A very compact s-box for aes. In CHES [2005]. *Link* .

Zhengjun Cao, Ruizhong Wei, and Xiaodong Lin. A fast modular reduction method. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Çetin Kaya Koç and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, Lecture Notes in Computer Science, 1999.

Cetricom. Standards for efficient cryptography group: Elliptic curve cryptography 1. 2000. *Link* .

Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In CHES [2002]. *Link* .

Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, and Bo-Yin Yang. Solving quadratic equations with xl on parallel architectures. In CHES [2012]. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, 2000. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001,roceedings*, Lecture Notes in Computer Science, 2001. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, 2002. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, 2003. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, 2004. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, Lecture Notes in Computer Science, 2005. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, Lecture Notes in Computer Science, 2006. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, Lecture Notes in Computer Science, 2008. Springer. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, Lecture Notes in Computer Science, 2009. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, 2010. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, Lecture Notes in Computer Science, 2011. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, Lecture Notes in Computer Science, 2012. *Link* .

CHES, editor. *Cryptographic Hardware and Embedded Systems - CHES 2015 - 14th International workshop, Leuven, Belgium, September 9-12, 2015. Proceedings*, Lecture Notes in Computer Science, 2015. *Link* .

Goyet Christopher. Cryptanalsye algebrique par canaux auxiliaire, 2012. *Link* .

Mathieu Ciet and Marc Joye. Elliptic curve cryptosystems in the presence of permanent and transient faults. 2003. *Link* .

Mathieu Ciet and Marc Joye. Practical fault countermeasures for chinese remaindering based rsa (extended abstract). In FDTC, editor, *IN PROC. FDTC05*. IEEE, 2005. *Link* .

Mathieu Ciet, Tanja Lange, Francesco Sica, and Jean-Jacques Quisquater. Improved algorithms for efficient arithmetic on elliptic curves using fast endomorphisms. 2003. *Link* .

Washington Cilio, Michael Linder, Chris Porter, Jia Di, Dale R. Thompson, and Scott C. Smith. Mitigating power- and timing-based side-channel attacks using dual-spacer dual-rail delay-insensitive asynchronous logic d3l. *Microelectronics Journal*, 2013. doi: doi:10.1109/SECON.2010.5453826. *Link* .

Christophe Clavier. De la sécuritée physique des crypto-systèmes embarqués, 2007. *Link* .

Christophe Clavier and Marc Joye. Universal exponentiation algorithm. In CHES [2001]. *Link* .

Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. Square always exponentiation. In Bernstein and Chatterjee. *Link* .

Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. Square always exponentiation. In Bernstein and Chatterjee. *Link* .

Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. Horizontal correlation analysis on exponentiation. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylaine Roussellet, and Vincent Verneuil. Improved collision-correlation power analysis on first order protected aes. In CHES [2011]. *Link* .

Jessy Clediere, Thanh-Ha Le, Quoc-Thinh Nguyen-Vuong, and Cecile Canovas. Novel approaches for improving the power consumption models in correlation analysis. ePrint Archive. *Link* .

Jean-Sébastien Coron. A new dpa countermeasure based on permutation tables. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *SCN*, Lecture Notes in Computer Science, pages 278–292. Springer, 2008. *Link* .

Jean-Sebastien Coron and Louis Goubin. On boolean and arithmetic masking against differential power analysis. In CHES [2000]. *Link* .

Jean-Sebastien Coron and Ilya Kizhvatov. An efficient method for random delay generation in embedded software. In CHES [2009]. *Link* .

Jean-Sebastien Coron and Alexei Tchulkine. A new algorithm for switching from arithmetic to boolean masking. In CHES [2003]. *Link* .

Jean-Sebastien Coron, David Naccache, and Julien P. Stern. On the security of rsa padding. In CRYPTO [1999]. *Link* .

Jean-Sebastien Coron, Antoine Joux, Ilya Kizhvatov, David Naccache, and Pascal Paillier. Fault attacks on rsa signatures with partially unknown messages. In CHES [2009]. *Link* .

Jean-Sebastien Coron, David Naccache, Mehdi Tibouchi, and Ralf-Philipp Weinmann. Practical cryptanalysis of iso-iec 9796-2 and emv signatures. In *CRYPTO*, 2009b. *Link* .

Jean-Sebastien Coron, David Naccache, and Mehdi Tibouchi. Revisiting higher-order dpa attacks. In CTRSA [2010]. *Link* .

Nicolas Courtois and Gregory V. Bard. Algebraic cryptanalysis of the data encryption standard. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

CRYPTO, editor. 1999.

CT-RSA, editor. *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, Lecture Notes in Computer Science, 2001. *Link* .

CTRSA, editor. *Topics in Cryptology - CT-RSA 2002 - The Cryptographers' Track at the RSA Conference 2005, Proceedings*, Lecture Notes in Computer Science, 2002. Springer Berlin Heidelberg. *Link* .

CTRSA, editor. *Topics in Cryptology - CT-RSA 2005 - The Cryptographers' Track at the RSA Conference 2003, Proceedings*, Lecture Notes in Computer Science, 2003. Springer Berlin Heidelberg. *Link* .

CTRSA, editor. *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, 2004. Springer. ISBN 3-540-20996-4. *Link* .

CTRSA, editor. *Topics in Cryptology - CT-RSA 2005 - The Cryptographers' Track at the RSA Conference 2005, Proceedings*, Lecture Notes in Computer Science, 2005. Springer Berlin Heidelberg. *Link* .

CTRSA, editor. *Topics in Cryptology - CT-RSA 2008 - The Cryptographers' Track at the RSA Conference 2008, Proceedings*, Lecture Notes in Computer Science, 2008. Springer Berlin Heidelberg. *Link* .

CTRSA, editor. *Topics in Cryptology - CT-RSA 2010 - The Cryptographers' Track at the RSA Conference 2010, Proceedings*, Lecture Notes in Computer Science, 2010. Springer Berlin Heidelberg. *Link* .

CTRSA, editor. *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, Lecture Notes in Computer Science, 2011. Springer Berlin Heidelberg. *Link* .

CTRSA, editor. *Topics in Cryptology - CT-RSA 2012 - The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 - March 2, 2012. Proceedings*, 2012.

Peter Czypek, Stefan Heyse, and Enrico Thomae. Efficient implementations of mqpks on constrained devices. In CHES [2012]. *Link* .

Jean DaRolt, Amitabh Das, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre, and Ingrid Verbauwhede. A new scan attack on rsa in presence of industrial countermeasures. In Schindler and Huss [2012]. *Link* .

Joeri de Ruiter and Erik Poll. Formal analysis of the emv protocol suite. In Sebastian Modersheim and Catuscia Palamidessi, editors, *TOSCA*, Lecture Notes in Computer Science, 2012. *Link* .

Blandine Debraize. Efficient and provably secure methods for switching from arithmetic to boolean masking. In CHES [2012]. *Link* .

DeCanniere. Analisys and design of symmetric encryption algorithms, 2007. *Link* .

Dehem. Design of an efficient public-key cryptographic library for risc-based smart cards, 2000. *Link* .

C. Delerablee, T. Lepoint, P. Paillier, and M. Rivain. White-box security notions for symmetric encryption schemes. In Biryukov et al. [2011]. *Link* .

A. W. Dent, K. G. Paterson, and P. R. Wild. Extensions to Chaum's Blind Signature Scheme and OpenCoin Requirements. Preprint. *Link* .

Alexander W. Dent. A brief history of provably-secure public-key encryptionn. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Des. *Advanced Encryption Standard*, 2001. *Link* .

Vassil S. Dimitrov, Laurent Imbert, and Pradeep Kumar Mishra. The double-base number system and its application to elliptic curve cryptography. *Mathematics of Computation*. *Link* .

Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate side channel attacks and leakage modeling. *J. Cryptographic Engineering*, 2011. *Link* , *Link* .

ECRYPT, editor. *Proceedings of the Ecrypt Workshop Tools For cryptanalysis*, Lecture Notes in Computer Science, 2007.

Moulay Abdelaziz Elaabid. Experimentations avancees sur les attaques par template, 2012. *Link* .

Moulay Abdelaziz Elaabid and Sylvain Guilley. Portability of templates. *J. Cryptographic Engineering*, 2012. *Link* .

E.Oswald. On side channel attacks and the application of algorithm countermeasures, 2003. *Link* .

IACR Cryptology ePrint Archive, editor. International Association for Cryptologic Research's Cryptology ePrint Archive. *L*ink.

EuroCrypt, editor. *Advances in Cryptology - EuroCrypt '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, 1997.

EuroCrypt, editor. *Advances in Cryptology - EuroCrypt 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, Lecture Notes in Computer Science, 2013. Springer.

Junfeng Fan, Xu Guo, Elke De Mulder, Patrick Schaumont, Bart Preneel, and Ingrid Verbauwhede. State-of-the-art of secure ecc implementations: A survey on known side-channel attacks and countermeasures. In Jim Plusquellic and Ken Mai, editors, *HOST*. IEEE Computer Society, 2010. *Link* .

Junfeng Fan, Benedikt Gierlichs, and Frederik Vercauteren. To infinity and beyond: Combined attack on ecc using points of low order. In CHES [2011]. *Link* .

Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical leakage-resilient symmetric cryptography. In CHES [2012]. *Link* .

Yunsi Fei, Qiasi Luo, and A. Adam Ding. A statistical model for dpa with novel algorithmic confusion analysis. In CHES [2012]. *Link* .

Wieland Fischer and Jean-Pierre Seifert. High-speed modular multiplication. In CTRSA [2004]. ISBN 3-540-20996-4. *Link* .

Wieland Fischer and Jean-Pierre Seifert. Duality between multiplication and modular reduction. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Marc Fischlin, editor. *Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings*, Lecture Notes in Computer Science. *Link* .

Lance Fortnow and Steven Homer. A short history of computational complexity. *Bulletin of the EATCS*, 2003. *Link* .

Pierre-Alain Fouque and Frederic Valette. The doubling attack - why upwards is better than downwards. In CHES [2003]. *Link* .

Pierre-Alain Fouque, Frederic Muller, Guillaume Poupard, and Frederic Valette. Defeating countermeasures based on randomized bsd representations. In CHES [2004]. *Link* .

Pierre-Alain Fouque, Sebastien Kunz-Jacques, Gwenaelle Martinet, Frederic Muller, and Frederic Valette. Power attack on small rsa public exponent. 2006. *Link* .

Pierre-Alain Fouque, Nicolas Guillermin, Delphine Leresteux, Mehdi Tibouchi, and Jean-Christophe Zapalowicz. Attacking rsa-crt signatures with faults on montgomery multiplication. In CHES [2012]. *Link* .

Amhuller Frederic. Fault attack on crt-rsa concrete result and practical approach. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Guillaume Fumaroli, Emmanuel Mayer, and Renaud Dubois. First-order differential power analysis on the duplication method. In K. Srinathan, C. Pandu Rangan, and Moti Yung, editors, *INDOCRYPT*. *Link* .

Martin Fürer. Faster integer multiplication. In David S. Johnson and Uriel Feige, editors, *STOC*. *Link* .

Rizzo Ottavio G. On the complexity of the 2^k-ary and of the sliding window algorithms for fast exponentiation. 2004. *Link* .

Benoat Gerard and Francois-Xavier Standaert. Unified and optimized linear collision attacks and their application in a non-profiled setting. In CHES [2012]. *Link* .

Benedikt Gierlichs. Theoretic methods for power analysis on embedded cryptography, 2011. *Link* .

Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. stochastic methods. In CHES [2006]. *Link* .

Benedikt Gierlichs, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede. Revisiting higher-order dpa attacks. In CTRSA [2010]. *Link* .

Benedikt Gierlichs, Jorn-Marc Schmidt, and Michael Tunstall. Infective computation and dummy rounds: Fault protection for block ciphers without check-before-output. In Alejandro Hevia and Gregory Neven, editors, *LATINCRYPT*, Lecture Notes in Computer Science, pages 305–321, 2012. *Link* .

Christophe Giraud. Attack on embedded cryptosystems and corresponding countermeasures, 2007. *Link* .

Oded Goldreich. *Fundation of Cryptography 1st volume*. Cambridge university press, 2001a. *Link* .

Oded Goldreich. *Principal componnet analyis*. Springer, 2001b. *Link* .

Oded Goldreich. *Fundation of Cryptography 2ndst volume*. Princetown university, 2012. *Link* .

Norman Gottert, Thomas Feller, Michael Schneider 0002, Johannes Buchmann, and Sorin A. Huss. On the design of hardware building blocks for modern lattice-based encryption schemes. In CHES [2012]. *Link* .

Louis Goubin and Ange Martinelli. Protecting aes with shamir's secret sharing scheme. In CHES [2011]. *Link* .

Louis Goubin and Jacques Patarin. Des and differential power analysis (the 'duplication' method). In Çetin Kaya Koç and Paar [1999]. *Link* .

H. Groscot. Estimation of some encryption functions implemented into smart cards. In *Advances in Cryptology: Proceedings of EuroCrypt 84, A Workshop on the Theory and Application of of Cryptographic Techniques, Paris, France, April 9-11, 1984, Proceedings*, 1984.

Sylvain Guilley and Renaud Pacalet. Differential power analysis model and some results. In Jean-Jacques Quisquater, Pierre Paradinas, Yves Deswarte, and Anas Abou El Kalam, editors, *In proceedings of CARDIS 2004*. Kluwer Academic Publishers, 2004. *Link* .

Tim Guneysu, Vadim Lyubashevsky, and Thomas Poppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In CHES [2012]. *Link* .

Hankerson, Menezes, and Vanstone. *Guide to Elliptic Curve Cryptography - 2nd edition*. Princetown university, 2012. *Link* .

Neil Hanley, Michael Tunstall, and William P. Marnane. Unknown plaintext template attacks. In WISA. *Link* .

Neil Hanley, Michael Tunstall, and William P. Marnane. Using templates to distinguish multiplications from squaring operations. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Neil Hanley, Michael Tunstall, and William P. Marnane. Using templates to distinguish multiplications from squaring operations. *International Journal of Information Security*, 2011b. *Link* .

Neil Hanley, HeeSeok Kim, and Michael Tunstall. Exploiting collisions in addition chain-based exponentiation algorithms. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Laszlo Hars. Long modular multiplication for cryptographic applications. In CHES [2004]. *Link* .

Hasenplaugh, Gaubatz, and Gopal. Fast modular reduction. In *Computer Arithmetic, 2007. ARITH '07. 18th IEEE Symposium on Arithmics*. IEEE Computer Society, 2007. *Link* .

Karl Hasselstrom. Fast division of large integers. a comparison of algorithms, 2003. *Link* .

Christoph Herbst and Marcel Medwed. Using templates to attack masked montgomery ladder implementations of modular exponentiation. In WISA. *Link* .

Clemens Heuberger. Application of digital expnesion. 2009. *Link* .

Clemens Heuberger and Daniel Krenn. Analysis of ω-width naf to imaginary quadratic bases. *Journal of Number Theory*, 2012. *Link* .

Stefan Heyse and Tim Guumlneysu. Towards one cycle per bit asymmetric encryption: Code-based cryptography on reconfigurable hardware. In CHES [2012]. *Link* .

Jip Hogenboom. Efficient secure computation of aes, 2010. *Link* .

Maghrebi Houssem, Prouff Emmanuel, Guilley Sylvain, and Danger Jean-Luc. A first-order leak-free masking countermeasure. In CTRSA [2012]. *Link* .

Michael Hutter, Mario Kirschbaum, Thomas Plos, Jorn-Marc Schmidt, and Stefan Mangard. Exploiting the difference of side-channel leakages. In Schindler and Huss [2012]. *Link* .

ISO/IEC/BSI. Iso-iec 9797-1: Information technologies - security techniques - message authentication codes (macs) - part1: Mechanisms using a block cipher. 1999. *Link* .

Kouichi Itoh, Jun Yajima, Masahiko Takenaka, and Naoya Torii. Dpa countermeasures by improving the window method. In CHES [2002]. *Link* .

LIU Jiye, ZHOU Yongbin, YANG Shuguo, and FENG Dengguo. Generic side-channel distinguisher based on kolmogorov-smirnov test: Explicit construction and practical evaluation. *Chinese Journal of Electronics*. *Link* .

Joye, Quisquater, Bao, and Deng. Rsa-type signatures in the presence of transient faults. *Cryptography and Coding, 6th IMA International Conference, UK, December 17-19*, 1997. *Link* .

Joye, Chevalier-Mames, and Ciet. Low cost solution for preventing spa: side-channel atomicity. *IEEE*, 2004. *Link* .

Joye, Tunstall, Boneh, and Oswald. *Fault Injection in Cryptography*. Springer editor, 2012. *Link* .

Marc Joye. Elliptic curve cryptosystems in the presence of faults. *Link* .

Marc Joye. Highly regular m-ary powering ladders. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography*, Lecture Notes in Computer Science. Springer, 2009. *Link* .

Marc Joye. On quisquater's multiplication algorithm. In Naccache. *Link* .

Marc Joye and Michael Tunstall. Exponent recoding and regular exponentiation algorithms. In AfricaCrypt, editor, *AfricaCrypt*, Lecture Notes in Computer Science, pages 334–349. Springer, 2009. *Link* .

Marc Joye and Christophe Tymen. Compact encoding of non-adjacent forms with applications to elliptic curve cryptography. In Kwangjo Kim, editor, *Public Key Cryptography*, Lecture Notes in Computer Science. Springer, 2001. *Link* .

Marc Joye and Sung-Ming Yen. Optimal ltor binary signed digit recoding. *Public key cryptography*, 2001. *Link* .

Marc Joye and Sung-Ming Yen. The montgomery powering ladder. In CHES [2002]. *Link* .

Marc Joye, Pascal Paillier, and Berry Schoenmakers. On second-order differential power analysis. In CHES [2005]. *Link* .

Anatolii Alexeevitch Karatsuba. Complexity of multiplication]. 1994. *Link* .

Chu Kenneth. A crash course on compact complex surfaces. *Link* .

Stephanie Kerckhof, Francois Durvaux, Cedric Hocquet, David Bol, and Francois-Xavier Standaert. Towards green cryptography: A comparison of lightweight ciphers from the energy viewpoint. In CHES [2012]. *Link* .

HeeSeok Kim, Seokhie Hong, and Jongin Lim. A fast and provably secure higher-order masking of aes s-box. In CHES [2011]. *Link* .

Miroslav Knezevic, Ventzislav Nikov, and Peter Rombouts. Low-latency encryption - is 'lightweight = light + wait'? In CHES [2012]. *Link* .

Neal Koblitz. An elliptic curve implementation of the finite field digital signature algorithm. In *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, 1998. URL Link. Link .

Neil Koblitz. *A course in Number Theory and Cryptography*. 1991. Link .

Cetin Koc. Hight speed implementation of rsa algorithm. Preprint. Link .

Cetin Kaya Koc and Tolga Acar. Analyzing and comparing montgomery multiplication algorithms. *IEEE Micro*. Link .

Martin Kochanski. Developing an rsa chip. In *CRYPTO*, 1985. Link .

Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, 1996. Link .

Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In CRYPTO [1999]. Link .

Juliane Krämer, Dmitry Nedospasov, and Jean-Pierre Seifert. Weaknesses in current rsa signature schemes. In Howon Kim, editor, *ICISC*, Lecture Notes in Computer Science, pages 155–168, 2011. Link .

Daniel krenn. Analysis of the width-w non-adjacent form in conjunction with hyperelliptic curve cryptography and with lattices. In *IACR Cryptology ePrint Archive* ePrint Archive. Link .

Daniel Krenn. Non-adjacent forms and their playground. 2012a. Link .

Daniel Krenn. Analysis and optimality of the width-w non-adjacent form to imaginary quadratic bases. 2012b. Link .

Daniel Krenn. Analysis and optimality of w-naf to imaginary quadratic basesd. 2012c. Link .

Daniel Krenn. Non-adjacent forms: Optimality and analysis. 2012d. Link .

Martin Kreuzer. Algebraic attacks galore! *Groups Complexity Cryptology*, 2009. Link .

Konrad J. Kulikowski, Mark G. Karpovsky, and Alexander Taubin. Power attacks on secure hardware based on early propagation of data. In *IOLTS*. IEEE Computer Society, 2006. Link .

Thanh-Ha Le, Jessy Clediere, Cecile Canovas, Bruno Robisson, Christine Serviere, and Jean-Louis Lacoume. A proposition for correlation power analysis enhancement. In CHES [2006]. Link .

Thanh-Ha Le, J. Clediere, C. Serviere, and J.-L. Lacoume. Efficient solution for misalignment of signal in side channel analysis. In *ICASSP*. IEEE, 2007. Link .

Herve Ledig, Frederic Muller, and Frederic Valette. Enhancing collision attacks. In CHES [2004]. Link .

Jen-Wei Lee, Szu-Chi Chung, Hsie-Chia Chang, and Chen-Yi Lee. An efficient countermeasure against correlation power-analysis attacks with randomized montgomery operations for df-ecc processor. In CHES [2012]. Link .

Bertrand LeGal, Aurelien Ribon, Lilian Bossuet, and Dominique Dallet. Reducing and smoothing power consumption of rom based controller implementations. In SBCCI, editor, *SBCCI*. ACM, 2010. Link .

Kerstin Lemke-Rust. Model and algorithm for physical cryptanalysis, 2007. Link .

Kerstin Lemke-Rust and Christof Paar. Analyzing side channel leakage of masked implementations with stochastic methods. Link .

Franz Lemmermeyer. Conics - a poor man's elliptic curves. 2003. Link .

Tzu-Chun Lin. Algorithms on elliptic curves over fields of characteristic two with non-adjacent forms. *Link* .

Patrick Longa and Catherine H. Gebotys. Efficient techniques for high-speed elliptic curve cryptography. In CHES [2010]. *Link* .

Roel Maes, Anthony Van Herrewege, and Ingrid Verbauwhede. Pufky: A fully functional puf-based cryptographic key generator. In CHES [2012]. *Link* .

H. Maghrebi, J. L. Danger, F. Flament, S. Guilley, and L. Sauvage. Evaluation of countermeasure implementations based on Boolean masking to thwart side-channel attacks. In *Proc. 3rd Int Signals, Circuits and Systems (SCS) Conf*, a. *Link* .

Houssem Maghrebi, Sylvain Guilley, Claude Carlet, and Jean-Luc Danger. Classification of high-order boolean masking schemes and improvements of their efficiency. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Mangard, Oswald, and Popp. *Power analysis attacks, revealing the secrets of smart cards*. 2006. *Link* .

Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 2011. *Link* .

Medwed Marcel. Algorithmie cryptographique aspect securitaires et algorithmiques, 2003. *Link* .

Medwed Marcel. Protecting security aware devices against implementation attacks, 2012. *Link* .

Avanzi Roberto Maria, Heuberger Clemens, and Prodinger Helmut. Redundant -adic expansions i: non-adjacent digit sets and their applications to scalar multiplication. *Design Codes Cryptography*, 2011. *Link* .

Bar Martin. Improved template attacks. In *COSADE*, Lecture Notes in Computer Science. *Link* .

Kamoun Najeh Masmoudi, Bossuet Lilian, and Ghazel Adel. Experimental Implementation of 2ODPA attacks on AES design with flash-based FPGA Technology. In *Proceedings of IMC 2010* fpl [2006]. URL *Link*. *Link* .

Seiichi Matsuda and Shiho Moriai. Lightweight cryptography for the cloud: Exploit the power of bitslice implementation. In CHES [2012]. *Link* .

Dimitrios Mavroeidis, Lejla Batina andv Twan van Laarhoven, and Elena Marchiori. Pca, eigenvector localization and clustering for side-channel attacks on cryptographic hardware devices. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *ECML/PKDD*. *Link* .

Marcel Medwed. Randomizing the montgomery multiplication to repel template attacks on multiplicative masking. In *COSADE*, Lecture Notes in Computer Science. *Link* .

Marcel Medwed and Elisabeth Oswald. Template attacks on ecdsa. In WISA. *Link* .

Marcel Medwed, Francois-Xavier Standaert, and Antoine Joux. Towards super-exponential side-channel security with efficient leakage-resilient prfs. In CHES [2012]. *Link* .

Menezes. *Handbook of aplied cryptography*. 1996. *Link* .

Thierry Messerges. Power analysis attacks and countermeasure for cryptographic algorithm, 2000. *Link* .

Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Investigations of power analysis attacks on smartcards. In *Proceedings of the USENIX Workshop on Smartcard Technology*, WOST'99, Berkeley, CA, USA. USENIX Association. *Link* .

Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Power analysis attacks of modular exponentiation in smartcards. In Çetin Kaya Koç and Paar [1999]. *Link* .

Brown Michael, Hankerson Darrel, Lopez Julio, and Menezes Alfred. Software implementation of the nist elliptic curves over prime fields. In CTRSA [2011]. *Link* .

Josep M. Miret, Daniel Sadornil, Juan Tena, Rosana Tomàs, and Magda Valls. On avoiding zvp-attacks using isogeny volcanoes. In WISA. *Link* .

Bodo Möller. Algorithms for multi-exponentiation. In Ali Miri and Serge Vaudenay, editors, *Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*, SAC '01, London, UK, UK, 2012. *Link* .

Peter L. Montgomery. Five, six, and seven-term karatsuba-like formulae. *IEEE Trans. Computers*, 1985. *Link* .

Amir Moradi and Oliver Mischke. How far should theory be from practice? - evaluation of a countermeasure. In CHES [2012]. *Link* .

Carlos Moreno and M. Anwar Hasan. Spa-resistant binary exponentiation with optimal execution time. *J. Cryptographic Engineering*, pages 87–99, 2011. *Link* .

Andrew Moss, Elisabeth Oswald, Dan Page, and Michael Tunstall. Compiler assisted masking. In CHES [2012]. *Link* .

Muir. Efficient integer representation for cryptographic operation, 2004. *Link* .

Debdeep Mukhopadhyay. Fault analysis of cryptosystems: Attacks, countermeasures and metrics. In CHES [2015]. *Link* .

Frederic Muller and Frederic Valette. High-order attacks against the exponent splitting protection. *Link* .

David Naccache, editor. *Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday.*

FIPS PUBS NIST. Data encryption standard. 1977. *Link* .

FIPS PUBS NIST. *n186.3, Digital Signature Standard DSS*, 2009. *Link* .

National Institute of Standards, Technology (U.S.), Information Technology Laboratory (National Institute of Standards, and Technology). Digital signature standard (dss) [electronic resource]. 2000. *Link* .

Katsuyuki Okeya, Katja Schmidt-Samoa, Christian Spahn, and Tsuyoshi Takagi. Signed binary representations revisited. In *CRYPTO*, 2004.

A. O. Oluwatope, B. A. Ojo, and G. A. Aderounmu. A memory optimized public-key crypto algorithm using modified modular exponentiation (mme). 2006. *Link* .

Yossef Oren and Avishai Wool. Tolerant algebraic side-channel analysis of AES. 2012. *Link* .

Yossef Oren, Mathieu Renauld, Franncois-Xavier Standaert, and Avishai Wool. Algebraic side-channel attacks beyond the hamming weight leakage model. In CHES [2012]. *Link* .

Elisabeth Oswald, Stefan Mangard, Christoph Herbst, and Stefan Tillich. Practical second-order dpa attacks for masked smart card implementations of block ciphers. In Pointcheval [2006]. *Link* .

Pascal Paillier and Ingrid Verbauwhede, editors. *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, Lecture Notes in Computer Science, 2007. Springer. *Link* .

Jing Pan, Jasper G. J. van Woudenberg, Jerry den Hartog, and Marc F. Witteman. Improving dpa by peak distribution analysis. In Biryukov et al. [2011]. *Link* .

Louiza Papachristodoulou. Secure des implementation against high-order differential power analysis. Preprint. *Link* .

Louiza Papachristodoulou. Efficient secure computation of aes, 2010. *Link* .

Jong-Yeon Park, Dong-Guk Han, Okyeon Yi, and JeongNyeo Kim. An equidistant message power attack using restricted number of traces on reduction algorithm. *Conference on Ubiquitous Information Technologies & Applications*, 2012. *Link* .

Jacques Patarin. Luby-rackoff: 7 rounds are enough for $2^{n((1-\epsilon))}$ security. In *CRYPTO*, 2003. *Link* .

Longa Patrick. Accelerating the scalar multiplication on elliptic curve cryptosystems over prime fields, 2007. *Link* .

Eric Peeters, Francois-Xavier Standaert, Nicolas Donckers, and Jean-Jacques Quisquater. Improved higher-order side-channel attacks with fpga experiments. Lecture Notes in Computer Science, 2005. *Link* .

Gilles Piret, Thomas Roche, and Claude Carlet. Picaro - a block cipher allowing efficient higher-order side-channel resistance. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *ACNS*, Lecture Notes in Computer Science, 2012. *Link* .

Gilles-Francois Piret. Block ciphers: Security proofs, cryptanalysis, design, and fault attacks, 2005. *Link* .

Hagen Ploog, Sebastian Flügel, and Dirk Timmermann. Improved zdn-arithmetic for fast modulo multiplication. In *ICCD*. IEEE Computer Society, 2001. *Link* .

David Pointcheval, editor. *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, Lecture Notes in Computer Science, 2006. Springer. *Link* .

Thomas Popp and Stefan Mangard. Masked dual-rail pre-charge logic: Dpa-resistance without routing constraints. In CHES [2005]. *Link* .

Dimitrios Poulakis. Some lattice attacks on dsa and ecdsa. *Appl. Algebra Eng. Commun. Comput.* *Link* .

Preprint, editor.

Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: a formal security proof. In EuroCrypt [2013]. *Link* .

Emmanuel Prouff, Christophe Giraud, and Sebastien Aumaunier. Provably secure s-box implementation based on fourier transform. In CHES [2006]. *Link* .

Rankl and Effing. *Smart Card Handbook*. Wiley editor, 2000. *Link* .

Pablo Rauzy, Sylvain Guilley, and Zakaria Najm. Formally proved security of assembly code against leakage. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Christian Rechberger and Elisabeth Oswald. Practical template attacks. In WISA. *Link* .

Francesco Regazzoni, Thomas Eisenbarth, Johann Grossschadl, Luca Breveglieri, Paolo Ienne, Israel Koren, and Christof Paar. Power attacks resistance of cryptographic s-boxes with added error detection circuits. *2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. *Link* .

Mathieu Renauld, Francois-Xavier Standaert, and Nicolas Veyrat-Charvillon. Algebraic side-channel attacks on the aes: Why time also matters in dpa. In CHES [2009]. *Link* .

Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. Selecting time samples for multivariate dpa attacks. In CHES [2012]. *Link* .

Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of aes. In CHES [2010]. *Link* .

Anderson Ross, Bond Mike, Choudary Omar, Murdoch Steven J., and Stajano Frank. Might financial cryptography kill financial innovation? — the curious case of emv. In *Proceedings of the 15th international conference on Financial Cryptography and Data Security*, FC'11. *Link* .

Sujoy Sinha Roy, Chester Rebeiro, Debdeep Mukhopadhyay, Junko Takahashi, and Toshinori Fukunaga. Scalar multiplication on koblitz curves using tau2-naf. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Ulrich Ruohrmair and Marten van Dijk. Practical security analysis of puf-based two-player protocols. In CHES [2012]. *Link* .

Kouichi Sakurai. Efficient elliptic curve cryptosystems from a scalar multiplication algorithm with recovery of the y-coordinate on a montgomery-form elliptic curve. In CHES [2001]. *Link* .

Santanu Sarkar and Subhamoy Maitra. Side channel attack to actual cryptanalysis: Breaking crt-rsa with low weight decryption exponents. In CHES [2012]. *Link* .

YSantanu Sarkar and Subhamoy Maitra. More on correcting errors in rsa private. keys: Breaking crt-rsa with low weight. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Schindler. Exponent blinding does not always lift (partial) spa resistance. In Javier Lopez and Gene Tsudik, editors, *ACNS*, Lecture Notes in Computer Science, 2011. *Link* .

Werner Schindler and Sorin A. Huss, editors. *Constructive Side-Channel Analysis and Secure Design - Third International Workshop, COSADE 2012, Darmstadt, Germany, May 3-4, 2012. Proceedings*, 2012.

Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In CHES [2005]. *Link* .

Alexander Schloumlsser, Dmitry Nedospasov, Juliane Kraomer, Susanna Orlic, and Jean-Pierre Seifert. Simple photonic emission analysis of aes - photonic side channel analysis for the rest of us. In CHES [2012]. *Link* .

Jorn-Marc Schmidt and Christoph Herbst. A practical fault attack on square and multiply. In FDTC, editor, *FDTC*, Lecture Notes in Computer Science, 2008. *Link* .

Jorn-Marc Schmidt and Chong Hee Kim. A probing attack on aes. In WISA. *Link* .

Jörn-Marc Schmidt, Michael Tunstall, Roberto Avanzi, Ilya Kizhvatov, Timo Kasper, and David Oswald. Combined implementation attack resistant exponentiation. In *Proceedings of the First international conference on Progress in cryptology: cryptology and information security in Latin America*, LATINCRYPT'10, Berlin, Heidelberg. *Link* .

Schneier. *Applied Cryptography*. 2000. *Link* .

Kai Schramm and Christof Paar. Higher order masking of the aes. In Pointcheval [2006], pages 208–225. *Link* .

Kai Schramm, Gregor Leander, Patrick Felke, and Christof Paar. A collision-attack on aes: Combining side channel- and differential-attack. *Link* .

Robert Sedgewick and Kevin Wayne. *Alforithms* 4^{th} *edition*. Princetown university, 2012. *Link* .

Bhavika Gambhava Shah Kruti R. New approach of data encryption standard, 2010. *Link* .

Claude E. Shannon. A mathematical theory of communication. *Mobile Computing and Communications Review*, 2001. *Link* .

Mael Berthier Shiqian WANG, Thanh-Ha Le. When cpa and mia go hand in hand. In *COSADE*, Lecture Notes in Computer Science. *Link* .

Andrey Sidorenko, Joachim van den Berg, Remko Foekema, and Michiel Grashuis Jaap de V. Bellcore attack in practice. In *IACR Cryptology ePrint Archive* ePrint Archive. *Link* .

Silverman. *Arithmetic of Elliptic Curves Cryptography*. 2000. *Link* .

Jerome A. Solinas. Efficient arithmetic on koblitz curves. *Des. Codes Cryptography*, pages 195–249, 2000. *Link* .

British Standards Institute Staff. *Information Technology. Security Techniques. Message Authentication Codes (MACs). Mechanisms Using a Dedicated Hash-function*, 2002. *Link* .

François-Xavier Standaert, Christophe Petit, and Nicolas Veyrat-Charvillon. Masking with randomized look up tables - towards preventing side-channel attacks of all orders. In Naccache. *Link* .

François-Xavier Standaert, Gaël Rouvroy, and Jean-Jacques Quisquater. Fpga implementations of the des and triple-des masked against power analysis attacks. In *FPL* fpl [2006]. *Link* .

Francois-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EuroCrypt 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Lecture Notes in Computer Science, 2009. *Link* .

Francois-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order dpa. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security*, 2010. *Link* .

Douglas Stebila and Nicolas Theriault. Unified point addition formulaend side-channel attacks. In CHES [2006]. *Link* .

Jacques Stern, Louis Granboulan, Phong Nguyen, and David Pointcheval. *Conception et preuves dalgorithmes cryptographiques - Cours de magistere MMFAI Ecole normale superieure*. 2004. *Link* .

Guilley Sylvain. Contre-mesures geometriques aux attaques exploitant les canaux cachees, 2007. *Link* .

Serge Tahe. *Apprentissage du language C#*. 2008. *Link* .

Thijs R. Timmerman. Mutual information analysisfor side channel attacks, 2011. *Link* .

Elena Trichina, Domenico De Seta, and Lucia Germani. Simplified adaptive multiplicative masking for aes. In CHES [2002]. *Link* .

Michael Tunstall. Random order m-ary exponentiation. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP*, Lecture Notes in Computer Science. Springer, 2005. *Link* .

Michael Tunstall and Marc Joye. Coordinate blinding over large prime fields. In CHES [2010]. *Link* .

Michael Tunstall, Neil Hanley, Robert McEvoy, Claire Whelan, Colin Murphy, and William Marnane. Correlation power analysis of large word sizes. *IET Irish Signals and System Conference-ISSC 2007*. *Link* .

Vincent van der Leest, Bart Preneel, and Erik van der Sluis. Soft decision error correction for compact memory-based pufs using a single enrollment. In CHES [2012]. *Link* .

Henk C. A. van Tilborg and Sushil Jajodia, editors. *Encyclopedia of Cryptography and Security, 2nd Ed.* Springer, 2011. *Link* .

Jasper G. J. van Woudenberg, Marc F. Witteman, and Federico Menarini. Practical optical fault injection on secure microcontrollers. In FDTC, editor, *FDTC*, Lecture Notes in Computer Science, 2011. *Link* .

Vincent Verneuil. Elliptic curve cryptography and security of embedded devices, 2012. *Link* .

Michael Vielhaber. Reduce-by-feedback: Timing resistant and dpa-aware modular multiplication plus: How to break rsa by dpa. In CHES [2012]. *Link* .

Waldyr. Application of frobienuis expansion in ecc, 2008. *Link* .

Michael Walker, editor. *Speeding Up Elliptic Scalar Multiplication with Precomputation*, Lecture Notes in Computer Science, 2000. Springer Berlin Heidelberg. doi: 10.1007/10719994-9. *Link* .

Colin D. Walter. Longer keys may facilitate side channel attacks. In Mitsuru Matsui and Robert J. Zuccherato, editors, *Selected Areas in Cryptography*, Lecture Notes in Computer Science. Springer, 2003. *Link* .

Peng Wang, Dengguo Feng, Wenling Wu, and Liting Zhang. On the correctness of an approach against side-channel attacks. Cryptology ePrint Archive, Report 2008/497, 2008. *Link* .

David J. Wheeler and Roger M. Needham. Tea, a tiny encryption algorithm. In *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*. *Link* .

WISA, editor. *International Workshop Information Security Applications, Revised Selected Papers*, Lecture Notes in Computer Science.

Marc Witteman, Jasper van Woudenberg, and Federico Menarini. Defeating rsa multiply-always and message blinding countermeasures. In CTRSA [2011]. *Link* .