Statistics and Secret Leakage

[Published in *ACM Transactions on Embedded Computing Systems* **3**(3):492–508, August 2004.]

Jean-Sébastien Coron[1], David Naccache[1], and Paul Kocher[2]

[1] Gemplus Card International
34 rue Guynemer, Issy-les-Moulineaux, F-92447, France
{jean-sebastien.coron,david.naccache}@gemplus.com
[2] Cryptography Research, Inc.
607 Market street, 5-th floor, San Francisco, CA 94105, USA
paul@cryptography.com

Abstract. In addition to its usual complexity assumptions, cryptography silently assumes that information can be physically protected in a single location. As one can easily imagine, real-life devices are not ideal and information may leak through different physical channels.
This paper gives a rigorous definition of leakage immunity and presents several leakage detection tests. In these tests, failure *confirms* the probable existence of secret-correlated emanations and indicates how likely the leakage is. Success *does not refute* the existence of emanations but indicates that significant emanations were not detected *on the strength of the evidence presented*, which of course, leaves the door open to reconsider the situation if further evidence comes to hand at a later date.

Keywords. Cryptography, side-channel analysis.

1 Introduction

In addition to its usual complexity postulates, cryptography silently assumes that secrets can be physically protected in tamper-proof locations.

All cryptographic operations are physical processes where data elements must be represented by physical quantities in physical structures. These physical quantities must be stored, sensed and combined by the elementary devices (*gates*) of any technology out of which we build tamper-resistant machinery. At any given point in the evolution of a technology, the smallest logic devices must have a definite *physical extent*, require a certain *minimum time* to perform their function and dissipate a minimal *switching energy* when transiting from one state to another.

The physical interpretation of data processing (a discipline named the *physics of computational systems* [18]) enables fundamental comparisons between computing technologies and provides physical lower bounds on the area, time and

energy required for computation [2, 10]. In this framework, a corollary of the second law of thermodynamics states that in order to introduce *direction* into a transition between states, energy must be lost irreversibly. A system that conserves energy cannot make a transition to a definite state and thus cannot make a decision (compute) ([18], 9.5). In tamper-resistant devices this inescapable energy transfer must (at least appear to) be independent of the machine's secret parameters. Despite extensive (and expensive) government-level research over the last forty years, most tamper resistance references are hardly accessible : TEMPEST's NACSIM 5100A, NATO's AMSG 720B and the SEPI proceedings [22, 21] are a few such examples. France's DISSI/SCSSI recommendation 400 is public but its six most informative parts are only accessible on a need-to-know basis. The rapid development of sophisticated (but often insecure) digital communication systems have created new academic and commercial interest in tamper resistance. Although the FIPS 140-1 standard [20] includes physical tamper resistance requirements, new standards such as Common Criteria [8] are currently being developed to provide a more comprehensive framework for tamper resistance testing. Several insightful papers about physical attacks (*e.g.* [1]) and fault attacks (*e.g.* [3, 4]) have been written, and these continue to be subjects of active research. This paper analyzes an area of recent interest – side-channel attacks – which exploit correlations between secret parameters and variations in timing [12], power consumption [13], and other emanations from cryptographic devices to infer secret keys.

This work is organized as follows : we start by introducing a general framework which is side-channel, algorithm and device-independent; this will yield a formal definition of leakage immunity (section 2), we will then present a collection of leakage detection tests (section 3) and experiment their effectiveness with a simple RLC filter (section 4).

2 What can we ideally expect ?

We view the tested hardware as a probabilistic Turing machine **H** with alphabet Σ, having a *start* and a *stop* state. **H** operates on the following one-way infinite tapes :

• a private read-only *key tape* $\mathcal{K}$ containing the key material which is the attacker's target,

• a public read-only *input tape* $\mathcal{M}$ which in practice contains the machine's input (program, plaintexts to encrypt, messages to sign, ciphertexts to decrypt *etc*),

• a private read-only *noise tape* $\mathcal{N}$ representing the noise added to the side channel by the attacker's measurement equipment and processes,

• a private *work tape* $\mathcal{W}$ containing the device's work variables,

• a public write-only *emanation tape* $\mathcal{E}$ (representing the side-channel information), and

• a public *output tape* $\mathcal{O}$ containing the hardware's output (plaintext decrypted or signature computed by $\mathbf{H}$ *etc.*).

$\mathbf{H}$ is finite expected time. *i.e.* there is a function f such that, on inputs of length n, $\mathbf{H}$'s expected computation time (number of state transitions elapsed from start to stop) does not exceed $f(n)$. As is usual, we also assume that there is a polynomial r such that $\mathbf{H}$ never writes more than $r(n)$ characters (including blanks) on $\mathcal{E}$ when the length of $\mathcal{M} \cup \mathcal{K}$ is n. Actually, the most complete model also includes a private read-only *random tape* $\mathcal{R}$ (the device's internal random number generator) used whenever a random number is required in a computation (*e.g.* a DSA signature or the generation of a fresh session key).

If $\mathbf{H}$ is given an empty $\mathcal{W}$, a noise tape $\mathcal{N}$ with $\eta \in \Sigma^\omega$, an input tape $\mathcal{M}$ with $\mu \in \Sigma^\omega$, a random tape $\mathcal{R}$ with $\rho \in \Sigma^\omega$ and is then run with $\kappa \in \Sigma^*$ on $\mathcal{K}$ then the contents of $\mathcal{E}$, denoted $\mathbf{H}_{\eta,\rho}(\kappa, \mu)$ (interpreted as the device's emanation, collected during some particular experiment) is well-defined. If we omit mention of η and ρ then $\mathbf{H}(\kappa, \mu)$ (the expected emanations characterizing a device keyed with κ and μ) is a *probability space*. The non-initialized hardware $\mathbf{H}$ can thus be seen as a *family of probability spaces*.

Referring to the usual definition of statistical indistinguishablity ([16], page 70) we define leakage immunity as follows :

Definition 1 : $\mathbf{H}$ *is leakage-immune if for all distributions* $\{K, M\}$ *and* $\{K', M'\}$*, the distributions* $\mathbf{H}(K, M)$ *and* $\mathbf{H}(K', M')$ *are statistically indistinguishable.*

Although this definition is overly cautious, it seems impossible to come up with a meaningful alternative that captures the distinction between breaking $\mathbf{H}$ in a harmful and a non-harmful sense (probably because of the imprecise meaning of the word *harmful*, which typically becomes clear only *after* $\mathbf{H}$ is broken). This is however, compensated by the fact that leakage immunity *guarantees* that no information on κ can be inferred from $\mathcal{E}$, whatever the attacker's strategy is. Needless to say, we know of no system which is secure in this sense.

In this light, vulnerabilities to timing and power consumption attacks, electromagnetic monitoring and microprobing are nothing but *specific* manners of not being leakage-immune.

Related work : In an independent work Chari *et al.* formalized a similar definition of leakage immunity ([5], section 2.1). Actually, after assuming this similar definition the two contributions differ : Chari *et al.* describe a provably secure instance whereas we develop tests capable of detecting secret leakage (*cryptophthora*) in unknown implementations.

3 What can we practically hope to achieve ?

Ideally, only a physical in-depth analysis of the device (an *a priori* test) could rule out the existence of emanations or quantify the leakage under some assumptions. Such insider analyses (which should be ideally conducted by the device's

manufacturer) would directly point to the origins of the leakage, provide an objective evaluation of the device's limitations and be more insightful than the black-box tests (also called *blind* or *a posteriori tests*) described hereafter.

It appears quickly that perfect proofs of concept are unavailable for a variety of reasons such as the limited precision of analog simulators or the extreme complexity of the analyzed devices (let alone the vendors' reluctance to reveal design details and the analysis' financial cost). VHDL synthesis provides a powerful capability to optimize designs for gate count or speed. To achieve this, synthesis tools have built-in timing analyzers that can automatically calculate worst case time delays, setup and hold conditions and use this information to selectively optimize the circuit where needed. The result is an automatically synthesized product which gate-level design has been computer generated. In an ideal situation, the designer should not need to examine this gate-level design (others apparently do that [14]), but until synthesis tools are more tightly merged with ASIC layout tools, there is always some amount of uncertainty (typically around $\pm 4\%$ for products such as Synopsys' PowerMill and PowerGate) on the device's spectral and temporal power consumption features.

First generation simulators ($\cong$ 1985) used the digital simulation results to infer the local capacitance C switched by each switch on each node. The power dissipation was then approximated by $CV_{\rm dd}^2 f$ where $V_{\rm dd}$ and f denote the supply voltage and clock frequency applied to $\mathbf{H}$. Recent packages use gate-level current simulation and recursive device partition to achieve better precision.

The tests presented in this paper are *specifically* designed to be cryptosystem and technology independent and should be soon available as an experimental postlayout library.

3.1 Significance tests

We are thus obliged to reason with partial information and find reliable *black-box* tests that exhibit *evidence* of leakage; the outcome of such tests may confirm or contradict what human judgement might lead to expect, but at least, conclusions will be objective and capable of statistical justification.

Statistics provide procedures for evaluating likelihood called *significance tests*. In essence, given two collections of samples, a significance test evaluates the probability that both samples could rise by chance from the same parent group. If the test's answer turns out to be that the observed results could arise by chance from the same parent source with very low probability we are justified in concluding that, as this is very unlikely, the two parent groups are most certainly different. Thus, we judge the parent groups on the basis of our samples, and indicate the degree of reliability that our results can be applied as generalizations. If, on the other hand, our calculations indicate that the observed results could be frequently expected to arise from the same parent group, we could have easily encountered one of those occasions, so our conclusion would be that a significant difference between the two samples was not proven (despite the observed difference between the samples). Further testing might, of course, reveal a genuine

difference, so it would be wrong to claim that our test *proved* that a significant difference did not exist; rather, we may say that a significant difference was *not demonstrated on the strength of the evidence presented*, which of course, leaves the door open to reconsider the situation if further evidence comes to hand at a later date. In practice, we would apply about twenty different tests to **H** (four of which are described in this paper) and if it passes these satisfactorily, we only consider it to be *possibly-resistant* (an experiment can only prove that something actually happens, but no finite number of trials can ever prove that something will never happen).

The non-technical reader may prefer this analogy : to challenge the hypothesis that a lake **H** contains no fishes (forms of information leakage) an *a-priori* tester would dive and inspect each portion of the lake. Although exhausting, such an inspection may definitely *prove* that there are no fishes in the lake. An *a posteriori* tester would rather throw different hooks into the water hoping that a fish will eventually bite one of them (for one single captured fish will *refute* the assumption, thereby making the economy of an underwater inspection). Failure to find fish proves nothing (*e.g.* the hooks may simply not be adapted to the species inhabiting the lake) but comforts the tester's empirical confidence in the correctness of his assumption.

Note that a very similar situation occurs in randomness tests [6, 9, 11, 17] where, if a sequence behaves randomly with respect to the *a posteriori* tests $T_1, T_2, \ldots, T_n$ one can not be sure that it will not be rejected by a further test T_{n+1}; yet, successive tests give more and more confidence in the randomness of the sequence without any *a priori* information about the structure of the random number generator.

3.2 Leakage detection tests

We start by transforming **H** into an experiment $c = \mathbf{H}(x)$ where x is the device's input (depending on the experiment, x can be a key, a message or the concatenation of both) and c the corresponding output; we denote by i the experiment's serial number. The device's emanation can be a scalar $e[i]$ (*e.g.* execution time), an array $\{e[i,0], e[i,1], \ldots, e[i,\tau-1]\}$ (*e.g.* power consumption) or a table :

$$\begin{pmatrix} e[i,1,0] & e[i,1,1] & \ldots & e[i,1,\tau-1] \\ \ldots & \ldots & \ldots & \ldots \\ e[i,\ell,0] & e[i,\ell,1] & \ldots & e[i,\ell,\tau-1] \end{pmatrix}$$

representing the simultaneous evolution of ℓ quantities (*e.g.* samples or microprobes) during τ clock cycles. The tests that we are about to describe operate on $e[i,\ldots]$ and use (existing) significance and randomness tests as basic building blocks :

Definition 2 : *When called with two sufficiently large samples X and Y, a significance test $S(X,Y)$ returns a probability α that an observed difference in some feature of X and Y could rise by chance assuming that X and Y were*

drawn from the same parent group. The minimal sample size required to run the test is denoted size(S).

Definition 3 : *When called with a sufficiently large sample set :*

$$X = \{x_1, \ldots, x_n\}$$

where each $x_i \in \mathbb{R}$ *is such that* $0 \leq x_i \leq 1$*, a randomness test* $R(X)$ *returns a probability* β *that some observed feature in* X *could rise by chance while sampling* n *times a random uniform distribution. The minimal sample size required to run the test is denoted* size(R).

Many randomness tests for binary strings exist and can be used in our construction after straightforward conversion (*e.g.* replace x_i by zero if $0 \leq x_i < 0.5$ and by one if $0.5 \leq x_i \leq 1$ *etc*). The tests mentioned in the following table are more or less standard and cover a reasonable range of statistical defects; they are easy to implement and sensitive enough for most practical purposes.

test R	**notation**	**description**
frequency test	`F-test`	[11], (page 55) 3.3.2;A
run test	`R-test`	[11], (page 60) 3.3.2;G

As for significance tests, we arbitrarily restricted our choice to the three most popular ones : the *distance of means*, *goodness of fit* and *sum of ranks*. The reader may find the description of these procedures in most undergraduate textbooks (*e.g.* [15, 19]) or replace them by any custom procedure compatible with definition 2 (we will come to that in section 3.4).

test S	**notation**	**description**
distance of means	`DoM-H-test`	[19], (pp. 240–242) 7.9
goodness of fit	`GoF-H-test`	[19], (pp. 294–295) 9.6
sum of ranks	`SoR-H-test`	[19], (pp. 306–308) 10.3

3.3 General vulnerability to timing attacks

This test challenges the claim : $2n$ *execution time measurements are insufficient to distinguish* $\mathbf{H}(\gamma_1)$ *from* $\mathbf{H}(\gamma_2)$ *with significant probability*.

• Select two inputs $\gamma_1 \neq \gamma_2$ (γ_j is typically a key, a message or the concatenation of both).

• Select a significance test S (*e.g.* amongst `DoM-H-test`, `GoF-H-test` and `SoRH-test`).

• For $j = 1$ and 2, feed $\mathbf{H}$ with γ_j and perform (under identical experimental conditions) $n \geq$ size(S) time measurements, we denote by $e_j[i]$ the i-th execution time obtained using γ_j.

• Compute :

$$\alpha = S(\{e_1[1], e_1[2], \ldots, e_1[n]\}, \{e_2[1], e_2[2], \ldots, e_2[n]\})$$

• If $\alpha > 1\%$ answer 'possibly' else answer 'no'.

Note : The reader could, of course, question the usefulness of this test for it would suffice to make sure that $e[i]$ is constant at some early design stage. Unfortunately, engineers usually build new systems on top of existing black boxes (*e.g.* compiled operating systems, commercially available chips *etc.*) which processing times depend on both γ_j and other unpredictable or undocumented parameters. The result is some global execution time distribution [12] where the contributions of γ_j and the other parameters are mixed.

3.4 General vulnerability to power consumption attacks

This test challenges the claim : $2n$ *power consumption curves (*τ*-sample long) are insufficient to distinguish* $\mathbf{H}(\gamma_1)$ *from* $\mathbf{H}(\gamma_2)$ *with significant probability.*

• Select two inputs γ_1 and γ_2 (γ_j is again a key, a message or the concatenation of both).

• Select a significance test S (*e.g.* amongst `DoM-H-test`, `GoF-H-test` and `SoRH-test`) and a randomness test R (*e.g.* amongst `F-test` and `R-test`).

• For $j = 1$ and 2, feed $\mathbf{H}$ with γ_j and perform (under identical experimental conditions) $n \geq \text{size}(S)$ power consumption acquisitions, we assume that each acquisition is τ-sample long, that $\tau \geq \text{size}(R)$ and denote by $e_j[i,t]$ the t-th sample of the i-th waveform obtained using γ_j.

• For $t = 0$ to $\tau - 1$ let :

$$\alpha[t] = S(\{e_1[1,t], e_1[2,t], \ldots, e_1[n,t]\}, \{e_2[1,t], e_2[2,t], \ldots, e_2[n,t]\})$$

• At this step $\{\alpha[0], \alpha[1], \ldots, \alpha[\tau-1]\}$ should be uniformly distributed if $\mathbf{H}$ is leakage-immune; consequently, let :

$$\beta = R(\{\alpha[0], \alpha[1], \ldots, \alpha[\tau-1]\})$$

• If $\beta > 1\%$ answer 'possibly' else answer 'no'.

Note : The test's *effectiveness* depends on the manner in which S and R handle the random variables defined by the device's underlying physics. Since our procedure does not assume any specific law of physics, inadequate choices of S and R will not result in *false* evaluations[1] but may stubbornly return the answer 'possibly' and fail to reflect an existing leakage (remember, we presume $\mathbf{H}$ *innocent until proven guilty*; failure to ask pertinent questions will not convict an innocent but may eventually force the detective to free $\mathbf{H}$ for lack of evidence).

At this point, preliminary planning and some hardware insight appear necessary. Figure 1 shows a CMOS logic inverter. The inverter can be looked upon as a push-pull switch : `in` grounded cuts off the top transistor, pulling `out` high. A high `in` does the inverse, pulling `out` to ground. CMOS inverters are the basic building-block of all digital CMOS logic, the logic family that has become dominant in very large scale integrated circuits (VLSI) [23].

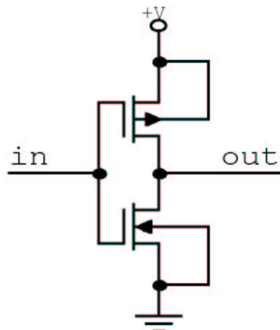

Fig. 1.

CMOS power dissipation has three different origins : the *static dissipation* due to leakage current drawn continuously from the power supply, the *dynamic dissipation* due the charging and discharging of internal load capacitances (stray) and the *short-circuit dissipation* due to transistor switching.

Static dissipation : In theory, unclocked CMOS circuits consume no quiescent current other than the small reverse-bias leakage between diffusion regions and the substrate plus some sub-threshold conduction (typically 10 nA to 10 μA, depending on the device's size). The source-drain diffusions and the n-well diffusion form reverse-biased parasitic diodes whose leakage contributes to static power dissipation. The quiescent power dissipation per gate is thus governed by the diode equation :

$$P_{\text{qu}} = i_{\text{s}}(e^{qV/kT} - 1) \times V_{\text{dd}}$$

where i_{s} is the reverse saturation current, V the diode voltage, q the electronic charge (1.6×10^{-19} C), k denotes Boltzmann's constant (1.38×10^{-23} J/K) and T is the device's temperature.

The total static power dissipation P_{st} is simply the sum of the individual P_{qu} contributions over all the gates composing $\mathbf{H}$ and is, at least in theory, independent of γ_j for large irregular chips. However, EEPROM avalanche injection requires a programming voltage (denoted V_{pp}) which is higher than V_{dd}. In a smart-card, V_{pp} is generated by a hybrid circuit having a specific P_{st} profile making EEPROM operations easy to characterize. Variations in P_{st} due to large bus driving were also observed experimentally.

Short-circuit dissipation : During transition from 0 to 1 or *vice-versa*, the device's n and p transistors are on for a short period of time. This results in a short *data-dependent* current pulse from V_{dd} to V_{ss}. The spike also depends on the clock's rise/fall time and, as confirmed experimentally with at least one smart-card chip, slow edges can increase the pulse's amplitude.

[1] provided, of course, that the chosen S and R comply with definitions 2 and 3.

Suggested guideline 1 : *When tested,* **H** *should be clocked with a signal which rise/fall times are long (unless the device's detectors forbid or filter such signals).*

Assuming that rise and fall times are equal ($t_\uparrow = t_\downarrow = t_\updownarrow$), that the junctions' β are equal[2] and that the technology's $V_{\rm tp}$ and $V_{\rm tn}$ are equal ($V_{\rm t}$ denotes the *threshold voltage*, the gate-source voltage at which drain current begins to flow; $V_{\rm t}$ is typically in the range of 0.5 to 5V in the forward direction), it can be shown that the short-circuit power dissipation is :

$$P_{\rm sc} = \frac{\beta}{12}(V_{\rm dd} - 2V_{\rm t})^3 \times t_\updownarrow f$$

Dynamic dissipation : Finally, current is also required to charge and discharge the internal capacitive loads. Denoting by C the load capacitance and by f the clock frequency, it is easy to show (under the assumption that $t_\updownarrow$ is much smaller than $1/f$) that the dynamic power dissipation is :

$$P_{\rm dy} = CV_{\rm dd}^2 f$$

As C is increased, $P_{\rm dy}$ progressively starts to dominate $P_{\rm st}$ and $P_{\rm sc}$ and a rough frequency domain analysis performed on a popular chip seems to suggest that $P_{\rm sc} \cong 15\%P$, $P_{\rm st} < 5\%P$ and $P_{\rm dy} > 80\%P$ where $P = P_{\rm sc} + P_{\rm st} + P_{\rm dy}$ is the device's total dissipation.

Suggested guideline 2 : *The definitions of P_{sc} and P_{dy} imply (and experiments confirm) that an important Hamming distance between γ_1 and γ_2 should increase the test's performances.*

Selection guidelines for R : As we have just seen, current is required to charge the internal capacitances during switching. Charging and discharging is not instantaneous (as a rule of thumb, a capacitor charges or decays to within 1% of its final value in five RC time constants) and therefore, data-dependent power consumption differences should not be *isolated incidences* in sufficiently sampled experiences. The genuine long leakage bursts will therefore be better discriminated from the random effects of chance[3] (false alerts) by randomness tests that are sensitive to *concentrations* of abnormally low values. Frequency tests are fairly good at spotting such defects and should suffice for most applications. The run test (which reacts to unusually long increasing or decreasing sequences, corresponding to the gradual charging and discharging of C) tends to give slightly better results. For technology-specific purposes, Kolmogorov-Smirnov's test can also be tuned to maximize sensitivity to *known* differences with respect to location, dispersion and skewness.

Selection guidelines for S : Since we made no assumption on the physical units or the range of $e_j[i,t]$, our test remains statistically sound even if we replace $e_j[i,t]$ by $\phi(t, e_j[i,t])$ where ϕ is an arbitrary function. The test will also remain

[2] note that unlike bipolar β which are unitless, FET β are measured in μA/V^2.

[3] strictly speaking, chance is never a cause, it only refers to a happening which occurs in the (apparent) absence of a cause.

valid if we replace samples by groups of samples. For instance, we may replace e by the least-squared :

$$\bar{e}_j[i,t] = \text{trend}(e_j[i,3t], e_j[i,3t+1], e_j[i,3t+2])$$

and (to better reflect the synchronous nature of $\mathbf{H}$) test $\bar{e}$ instead of e. $3t$ is only a toy example and acquisition frequencies which are integer multiples of f are good enough for most evaluations; more accurate results can nevertheless be obtained by deseasonalization :

Suggested guideline 3 : *Trigger the sampling operation by* $\mathbf{H}$*'s clock and analyze samples by groups corresponding to each clock cycle.*

Needless to say, ϕ could degrade or enhance the signals that we want to detect and a good selection of ϕ is crucial. This can be achieved by various techniques which are beyond the scope of this paper (*e.g.* apply geometric hashing [24] to sample groups corresponding to different clock cycles and tune feature extraction by simulated annealing).

Finally, the test should never be run in parallel on two devices of the same nominal type. If this is not respected, manufacturing spread is likely to be detected instead (or with) the data-dependent leakage by the test.

(Strongly) suggested guideline 4 : *Re-key the same device; do not use distinct devices (of the same nominal type) to collect* $e_1[i,t]$ *and* $e_2[i,t]$.

3.5 Correlation with the I/O's Hamming weight

While in the previous test we analyzed general forms of leakage, here we look for a *correlation* between e and the device's I/O. For doing so, we challenge the following claim : *power consumption variations do not increase or decrease with the Hamming weight of* $\mathbf{H}$*'s input or output.*

• Select k different inputs $\gamma_0, \ldots, \gamma_{k-1}$ such that $\hbar(\gamma_{i+1}) > \hbar(\gamma_i)$ where $\hbar(x)$ denotes the Hamming weight of x.

For instance, if the device's input is a string of bytes and if it is known that $\mathbf{H}$ is an 8-bit machine, the tester may set $k = 9$ and define γ_i to be a series of bytes of value $2^i - 1$. Let $\sigma(\hbar(\gamma_j))$ denote the standard deviation of $\{\hbar(\gamma_0), \ldots, \hbar(\gamma_{k-1})\}$.

• For $j = 0$ to $k-1$:

key $\mathbf{H}$ with γ_j and perform n power consumption acquisitions, we assume again that each acquisition is τ-sample long, that $\tau \geq \text{size}(R)$ and denote by $e_j[i,t]$ the t-th sample of the i-th waveform obtained using γ_j.

• Average the power consumption curves :

$$\bar{e}_j[t] = \frac{1}{n} \sum_{i=0}^{n-1} e_j[i,t]$$

and compute (the covariance and standard deviations are all taken over the variable j) for $t = 0$ to $\tau - 1$:

$$\rho[t] = \frac{\text{Cov}\,(\bar{e}_j[t], \hbar(\gamma_j))}{\sigma(\bar{e}_j[t])\sigma(\hbar(\gamma_j))}$$

• If, indeed, at all points in time there is no direct (negative or positive) correlation between the average power consumption and the Hamming weights of γ_j, the hypotheses $\rho[t] = 0$ should hold for $0 \leq t < \tau$ and since the statistic :

$$z[t] = \frac{\rho[t]\sqrt{k-2}}{\sqrt{1-\rho[t]^2}}$$

follows a t-distribution with $k - 2$ degrees of freedom, we can compute the probabilities :

$$\alpha[t] = t\text{-distribution}_{k-2}(z[t]) \quad \text{for} \quad t = 0, \ldots, \tau - 1$$

and make sure that $\{\alpha[0], \alpha[1], \ldots, \alpha[\tau - 1]\}$ is uniformly distributed by testing :

$$\beta = R(\{\alpha[0], \alpha[1], \ldots, \alpha[\tau - 1]\})$$

• If $\beta > 1\%$ answer 'possibly' else answer 'no'.

Note : This test can also be applied to the device's output by modifying the input arbitrarily until an output having a desired weight appears. This limits the test to moderate word sizes (typically < 32 bits) but appears sufficient in most situations.

3.6 Correlation between the leakage and external parameters

Although in theory, power consumption increases approximately linearly with the clock's frequency (as we have just seen, switching requires current and as frequency increases, switching becomes more frequent in time), other parameters such as the clock's shape, duty cycle, the external temperature or V_{dd} influence the leakage. The test presented in this section challenges the claim : *leakage is independent of the external parameters applied to* $\mathbf{H}$ *(such as the clock's shape, frequency, temperature,* V_{dd}*, etc.)*

We denote by θ_0 and θ_1 two different experimental conditions which might be qualitative (*e.g.* θ_0 is a square clock whereas θ_1 is a triangular one) or quantitative (*e.g.* θ_0 means $V_{\text{dd}} = 4\text{V}$ whereas θ_1 means $V_{\text{dd}} = 5\text{V}$).

• For $u = 0$ and 1, subject $\mathbf{H}$ to θ_u and perform $v > \text{size}(S)$ times the test described in section 3.4. Let :

$$\alpha_u[\ell, 0], \ldots, \alpha_u[\ell, \tau - 1]$$

be the probability curve obtained during the ℓ-th experiment under θ_u.

• Select a significance test S and a randomness test R.

• For $t = 0$ to $\tau - 1$ let :

$$a[t] = S(\{\alpha_1[1,t], \alpha_1[2,t], \ldots, \alpha_1[v,t]\}, \{\alpha_2[1,t], \alpha_2[2,t], \ldots, \alpha_2[v,t]\})$$

• At this step $\{a[0], a[1], \ldots, a[\tau-1]\}$ should be uniformly distributed if the leakage is independent of θ. As for the previous tests, let :

$$\beta = R(\{a[0], a[1], \ldots, a[\tau-1]\})$$

• If $\beta > 1\%$ answer 'possibly' else answer 'no'.

Note : Here, success *does not imply possible-resistance* but indicates that if **H** leaks, the leakage (which may be important) does *not seem to vary* when θ_0 is replaced by θ_1 (we say that **H** is *possibly stable*).

Finally, in all experiments involving temperature, one should keep in mind that V_{GS} and β depend on temperature. This causes drifts in output current with changes in ambient temperature; in addition, the junction's temperature varies as the load voltage is changed (because of variation in the transistor's dissipation), resulting in departure from the FET's ideal behavior. Therefore, if we key **H** with γ_1, perform n acquisitions, replace γ_1 by γ_2 and perform n new acquisitions, the first (γ_1-type) acquisitions will progressively heat **H** while the acquisitions performed with γ_2 will take place in a thermodynamically stable device (at some point, **H**'s temperature will reach an equilibrium that depends on the clock's frequency, V_{dd} and the external temperature). This difference between $e_1[i,t]$ and $e_2[i,t]$ can therefore be misinterpreted by the test as a data-dependent one.

Suggested guideline 5 : *When collecting the power consumption curves, alternate acquisitions with γ_1 and γ_2.*

4 What can we (typically) get for a reasonable price ?

To evaluate our tests, we implemented the following 68HC05-based PIN-comparison routine on a popular smart-card chip :

```
      CLR   Result  ; Result = 0
      LDX   #$08    ; for X = 8 downto 1
more  LDA   k-1,X   ; {
      EOR   m-1,X   ; A = k[X-1] xor m[X-1]
      ORA   Result  ; A = A or Result
      STA   Result  ; Result = A
      DEX           ;
      BNE   more    ; }
      SEC           ; carry = 1
      SBC   #$00    ; if (Result==0) then carry = 1 else carry = 0
      CLRA          ; A = 0
      CLR   Result  ; Result = 0
      RTS           ; return(carry)
```

After running the DoM-H-test (appendix A) on the device, we added the RLC filter drawn in figure 2 and re-started from scratch.

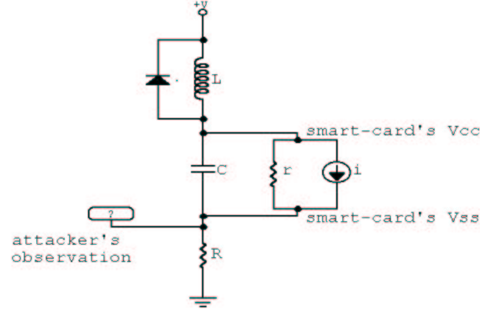

Fig. 2.

A (very) long list of defects makes this protection non-ideal and we *do not recommend* to adopt it in any practical application (actually, L even acts as an antenna that broadcasts signals correlated to the power consumption variations). We nevertheless proceeded to use this filter, which attenuates the input signal by :

$$\rho(\omega) = \frac{1}{r+R} \times \sqrt{(L+CrR)^2\omega^2 + (r+R-CLr\omega^2)^2}$$

to find out to what extent figure 2 departs from definition 1 (the diode is simply added to block the inductive kick; something like a 1N4004 is fine for nearly all cases).

Usual smart-card current consumption is roughly 10 mA for V_{dd}=5 V, whereby $r \cong 500\Omega$. Assuming that the resistor added by the attacker is small ($R \cong 10\Omega$) and using $C = 4.7$nF and $L = 1\mu$H we get a 27 dB attenuation for $f = \omega/(2\pi) =$ 3.57MHz.

Figure 3 shows the card's average ($n = 1000$) power consumption curve for $\mathtt{k}_0 = \mathtt{00...00}_{16}$ where the eight loop iterations can be easily distinguished.

Figure 5 shows[4] the α curve obtained when applying the DoM-H-test to curves obtained with $\mathtt{k}_0$ and $\mathtt{k}_1 = \mathtt{FF...FF}_{16}$ (for $\mathtt{m} = \mathtt{55...55}_{16}$ in both cases). The dashed line formed at the $\alpha \cong 0$ level points-out the clock-cycles where the $\mathtt{k}_0$ curves could be distinguished from the $\mathtt{k}_1$ ones.

As expected, a closer look at a problematic clock cycle (155) spotted by the test reveals a genuine difference between the two curves (figure 7).

Repeating the same experiment with the filtered card, figures 3, 5, 5 become 4, 6, 8 (y axis zoomed when necessary). Surprisingly, it appears that the filter

[4] axes cross at $\{0, -0.1\}$ to avoid plotting points on the x-axis.

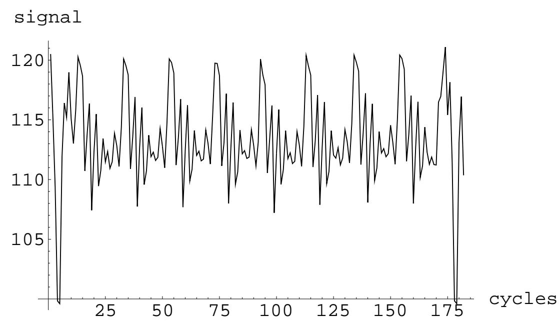

Fig. 3. Average power consumption

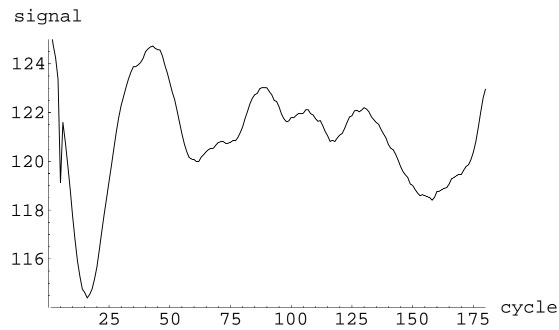

Fig. 4. Filtered average power consumption

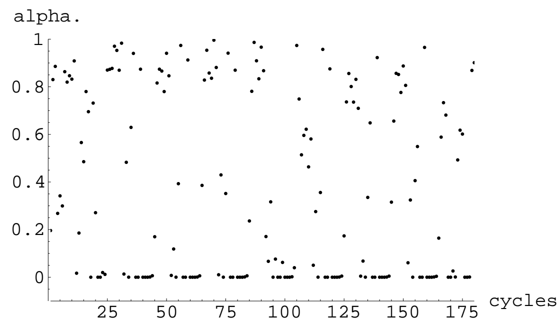

Fig. 5. The α curve

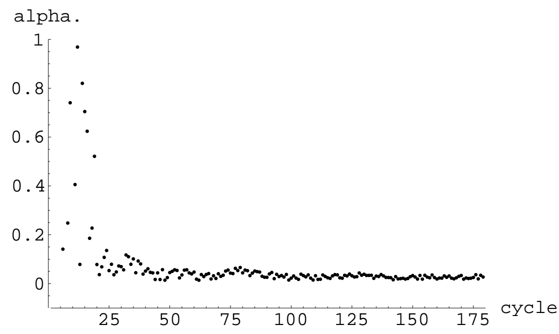

Fig. 6. The filtered α curve

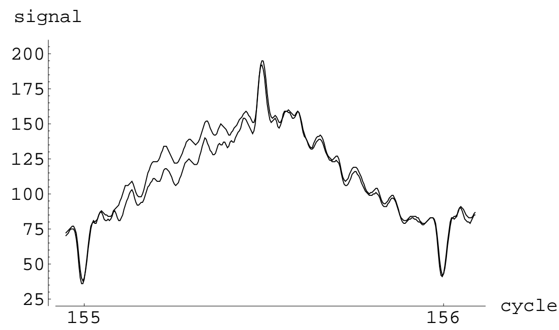

Fig. 7. The α curve at cycle 155

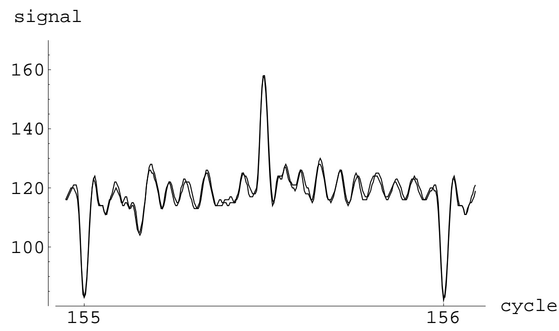

Fig. 8. The filtered α curve at cycle 155

increased the number of samples in which the test failed ! The explanation of this counter-intuitive observation is the following : L and C act as energy accumulators and average the power consumption differences into the future. When a first difference occurs, L and C start averaging it, thereby contaminating the coming samples. Since our routine *repeats* the *same* comparison eight times, the power consumption quickly reaches (for $\mathtt{k}_0$ and $\mathtt{k}_1$) two different (yet individually stable) signal levels, detected by test.

More effective power consumption compensators exist. These are based on *active* components (FETs) that *dissipate* power[5] whenever the card does not. The design of such protections is somewhat technical given the need to eliminate HF peaks (let alone insensitivity to V_{dd}, clock and temperature variations). Active protections also increase the circuit's global power consumption, which might be very problematic in some applications (*e.g.* mobile telephony).

Data-related dissipation has specific spectral characteristics and it appears useless to waste energy in order to overcome variations in frequencies where consumption is data-independent. For example, rough spectral estimates indicate that only 30 to 40% carefully triggered (and this is *precisely* where the difficulty is) extra dissipation might be enough to complement the data-dependent components in most chips. It is therefore our belief that the best long-term solutions involve minimizing data dependent side channels and building cryptography that inherently tolerates some information leakage, as opposed to the (energy-consuming) solution consisting of brutally flattening the power consumption curve.

A The difference of mean test

The `DoM-H-test` (*e.g.* [7]) is a significance test returning a probability α that an observed difference *in the means* of two sample sets X and Y could rise by chance, assuming that X and Y were drawn from the same parent population.

In other words, the test challenges the hypothesis : $\mu[X] \stackrel{?}{=} \mu[Y]$ where $\mu[i]$ denotes the mean of set i.

By virtue of the CLT, the experimental averages of X and Y (respectively $\bar{X}$ and $\bar{Y}$) are approximately Gaussian, independent, of expectations $\{\mu[X], \mu[Y]\}$ and variances $\{\sigma[X]^2/n[X], \sigma[Y]^2/n[Y]\}$; where $n[U]$ denotes the number of elements in the set U.

We can therefore compute the reduced Gaussian variable :

$$\epsilon = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{s[X]^2}{n[X]} + \frac{s[Y]^2}{n[Y]}}}$$

($s[U]$ denotes the standard deviation of the set U) and look-up its corresponding value in the CDF Gaussian table which yields the hypothesis' significance α

[5] instead of *averaging* it.

representing the probability that the reduced deviation will equate or exceed in absolute value a given ϵ.

α	0.000	0.010	0.020	0.030	0.040	0.050	0.060	0.070	0.080	0.090
0.00	∞	2.576	2.326	2.170	2.054	1.960	1.881	1.812	1.751	1.695
0.10	1.645	1.598	1.555	1.514	1.476	1.440	1.405	1.327	1.341	1.311
0.20	1.282	1.254	1.227	1.200	1.175	1.150	1.126	1.103	1.080	1.058
0.30	1.036	1.015	0.994	0.974	0.954	0.935	0.915	0.896	0.878	0.860
0.40	0.842	0.824	0.806	0.789	0.772	0.755	0.739	0.722	0.706	0.690
0.50	0.674	0.659	0.643	0.628	0.613	0.598	0.583	0.568	0.553	0.539
0.60	0.524	0.510	0.496	0.482	0.468	0.454	0.440	0.426	0.412	0.399
0.70	0.385	0.372	0.358	0.345	0.332	0.319	0.305	0.292	0.279	0.266
0.80	0.253	0.240	0.228	0.215	0.202	0.189	0.176	0.164	0.151	0.138
0.90	0.126	0.113	0.100	0.088	0.075	0.063	0.050	0.038	0.025	0.013

α is obtained by adding the two numbers appearing in the margins (for instance : for $\epsilon = 1.960$ table[ϵ]=0.000+0.05 $=$ 0.05), except for small values where the following table should be used :

α	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
ϵ	3.291	3.891	4.417	4.892	5.327	5.731	6.109

PC users may prefer Mathematica's standard `DoM-H-test` (`Statistics` package) or use α = `2(1-N[CDF[NormalDistribution[0,1],`ϵ`]])` instead of the CDF Gaussian table.

Acknowledgements The authors are grateful to Philippe Anguita, Olivier Benoît, Cyril Brunie, Christophe Clavier, Benjamin Jun, Pascal Moitrel and Yiannis Tsiounis for their valuable comments.

References

1. R. Anderson and M. Kuhn. Tamper resistance – a cautionary note. *The second* USENIX *workshop on electronic commerce*, pages 1–11, 1996.
2. C. Bennett. Logical reversibility of computation. *IBM Journal of R&D*, 17:525–532, 1973.
3. E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. *Proceedings of Crypto' 97*, pages 513–525, 1997.
4. D. Boneh, R. DeMillo, and R. Lipton. On the importance of checking cryptographic protocols for faults. *Proceedings of Eurocrypt' 97*, pages 37–51, 1997.
5. S. Chari, C. Jutla, J. Rao, and P. Rohatgi. Towards sound approaches to couteract power-analysis attacks. *Proceedings of Crypto' 99*, pages 398–412, 1999.
6. J.-S. Coron. On the security of random sources. *Proceedings of PKC'99, Springer-Verlag*, pages 29–42, 1999.
7. F. Edgeworth. Observations and statistics : an essay on the theory of errors of observation and the first principles of statistics. *Transactions of the Cambridge philosophical society*, pages 138–169, January 1885.
8. ISO/IEC 15408-1:1999(E). Information technology – security techniques – evaluation criteria for it security. *International Organization for Standardization and International Electrotechnical Commission*, 1999.

9. B. Jun and P. Kocher. The intel random number generator. *Cryptography Research white paper,* http://www.cryptography.com/intelRNG.pdf, 1999.
10. R. Keyes. Physical limits in digital electronics. *Proceedings of the IEEE*, 63, 1975.
11. D. Knuth. The art of computer programming, seminumerical algorithms. 2, 1969.
12. P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. *Proceedings of Crypto' 96*, 1996.
13. P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. *Proceedings of Crypto' 99*, 1999.
14. O. Kommerling and M. Kuhn. Design principles for tamper-resistant smart-card processors. *Proceedings of the* USENIX *workshop on smartcard technology*, 1999.
15. R. Langley. Practical statistics. *Dover publications, New-York*, 1968.
16. M. Luby. Pseudorandomness and cryptographic applications. *Princeton computer science notes*, 1996.
17. U. Maurer. A universal statistical test for random bit generators. *Journal of Cryptology*, 5,2, 1992.
18. C. Mead and L. Conway. Introduction to VLSI systems. *Addison-Wesley*, 1980.
19. I. Miller, J. Freund, and R. Johnson. Probability and statistics for enginners. *Prentice Hill*, 1990.
20. NIST. Security requirements for cryptographic modules. *National Institute of Standards and Technology, Federal Information Processing Standards Publication 140-1*, 1994.
21. SEPI'91. Symposium on electromagnetic security for information protection, Rome (Italy). 1991.
22. SPI'88. Primo simposio nazionale su sicurezza elettromagnetica nella protezione dell'informazione, Rome (Italy). 1988.
23. N. Weste and K. Eshraghian. Principles of CMOS VLSI design. *Addison-Wesley*, 1993.
24. H. Wolfson. Geometric hashing, an overview. *IEEE Computational Science and Engineering*, 4,4, 1997.