Simulation Models for Side-Channel Information Leaks

Kris Tiri[1]
tiri@ee.ucla.edu

[1]Electrical Engineering Dept.
UC Los Angeles, USA

Ingrid Verbauwhede[1,2]
ingrid@ee.ucla.edu

[2]Dept. ESAT/SCD-COSIC
K.U.Leuven, Belgium

ABSTRACT
Small, embedded integrated circuits (ICs) such as smart cards are vulnerable to so-called side-channel attacks (SCAs). The attacker can gain information by monitoring the power consumption, execution time, electromagnetic radiation and other information that is leaked by the switching behavior of digital CMOS gates. Ever since power attacks have been introduced in 1999, many countermeasures have been proposed. Often a significant increase in security has been touted. We will show that in order to assess the effectiveness of a countermeasure, a correct simulation model of the side-channel information leaks is vital. We will show that seemingly correct approximations can lead to completely flawed results.

Categories and Subject Descriptors
B.5 [**Hardware**]: Register-Transfer-Level Implementation; B.6 [**Hardware**]: Logic Design; B.7 [**Hardware**]: Integrated Circuits; E.3 [**Data**]: Data encryption.

General Terms
Design, Security, Verification.

Keywords
Simulation Model, Countermeasure, Side-Channel Attack, Differential Power Analysis, Encryption, Smart Card, Security IC.

1. INTRODUCTION
Most embedded applications, such as cellular phones and PDAs, require security and privacy protection. Much design effort is thus spent in developing secure protocols and selecting strong encryption algorithms to achieve the security level envisioned in the specifications. Yet, the security IC, which precisely provides the support for the required algorithms and protocols, emerges more and more as the main vulnerability. Due to physical and electrical effects, it broadcasts information that is related to the secret key. Information, such as execution time and power consumption, has been used to find the secret key with so-called side-channel attacks. SCAs are a real threat for any device of which the security IC is easily observable such as smart cards and embedded devices [1],[2]. SCAs have been effective in extracting the key of microprocessor-, DSP-, FPGA- and ASIC-based encryption systems.

Side-channel attacks are not a new practice. They have been of all times. One of the most well-know examples is the safecracker who uses his fingers and ears to feel and listen the tumblers impact each other while turning the dial. By observing when the lock's tumblers fall into place, he can crack the combination lock quickly and much faster than anyone could open the safe by trying every possible combination. While present-day side-channel attacks on electronic systems concentrate on electromagnetic radiations, execution time or power consumption, acoustic emanations have not been overlooked. The sounds of a personal computer (PC) also appear to be a rich source of information on CPU activity and it has been suggested to break encryption algorithms by simply listening to a PC [3].

Side-channel attacks provide information to find the secret key quicker than with a brute force attack. A brute force attack on the Advanced Encryption Standard (AES) algorithm, in which you try each and every possible value of the 128 bit key, is impossible with today's technology. Schneier even wrote in 1998 that there is not enough silicon in the galaxy or enough time before the sun burns out to brute-force triple-DES [which uses a 112 bit key] [4]. With the differential power analysis (DPA) attack, however, we have been able to find the key of an unprotected ASIC AES implementation in less than three minutes, from the start of the measurements to the end of the analysis [5]. It shows that security is only as strong as its weakest link.

The differential power analysis attack [6] is based on the fact that logic operations in standard static CMOS have power characteristics that depend on the input data. Power is only drawn from the power supply when a 0 to 1 output transition occurs. The attack relies on statistical analysis to retrieve the information from the power consumption variation that is correlated to the secret key. The attack can be mounted without precise knowledge of the architecture and implementation. It is only necessary to know which algorithm is being used and to have access to plaintext or ciphertext data. The DPA is effective even if power variations are overshadowed with measurement errors and power dissipation from other processing elements on the die. After a sufficient number of power acquisitions, a signal will emerge from the noise, as the signal to noise ratio ideally increases with the square root of the number of measurements [7]. Makeshift measures, such as the randomization of the execution sequence or the addition of a random power consuming module or a current sink, have been proven unsuccessful in thwarting power attacks [8],[6],[9].

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

Algorithmic countermeasures modify the algorithm in order to decorrelate the power consumption and the data being processed. For instance with masking (e.g. [10]), a random "mask" is added to the data prior to the encryption and removed afterwards without changing the encryption result. The distinguishing feature of algorithmic countermeasures is that mathematically they are DPA resistant. In practice, however, proposed solutions actually have been insecure [11]. The reason is that for the theoretical proof flawed essential preconditions had been assumed. It has been pointed out that for instance employing masked CMOS gates is only an effective countermeasure if glitches are not taken into account [12]. Glitches are the spurious signal transitions of a logic gate caused by differences in input signal arrival times. Yet, this is not a correct power model for standard static CMOS logic, which is the default logic style in standard cell libraries used for security ICs. It shows that side-channel resistance cannot be isolated at one abstraction level. It also shows that an accurate simulation model is essential to evaluate the side-channel resistance of a proposed countermeasure.

Hardware countermeasures change the behavior of the operations invoked by function calls. Their goal is to reduce the signal to noise ratio sufficiently such that a DPA attack becomes de facto infeasible. This is the case if the time required to perform the number of measurements is larger than the lifetime of the secret key. The lifetime of a key is often expressed by the number of encryptions for this key. A distinction can be made based on the abstraction level that the measures are implemented. At the gate level, specialized circuit styles and a place & route approach exist such that each individual gate has a quasi data-independent power dissipation [13]. At the macro level, a macro can be replaced by two macros, with completely different power consumption profiles, which are randomly inserted into the datapath [14]. At the instruction level, secure versions of instructions can be called that process the normal and the complementary version of the operands simultaneously [15]. At the function level, functions are modified such that subroutines occur with the same order and redundant subroutines are inserted if necessary [16].

The DPA resistance of hardware countermeasures is evaluated through experimental analyses. The quality of the assessment, however, is only as good as the simulation model used. In contrast with simple power analysis (SPA), in which a single power acquisition is recorded and inspected visually to identify relevant power fluctuations, DPA uses statistical methods that detect small power variations. A visual inspection of the power consumption profile or even a statistical measure is not a suitable figure of merit to define DPA resistance. It has been shown that an attack is possible with a mere 1% absolute variation and 2‰ standard deviation on the cycle-to-cycle power consumption obtained from transistor level simulations [13]. The resistance against DPA is quantified with the MTD, which is the number of measurements to disclosure. Until a formula has been found that accurately accounts for all the dynamics involved in defining the signal to noise ratio, this will require an actual attack using an adequate power simulation model and correct assumptions on the attack. Furthermore, standard encryption algorithms, such as DES, AES, and ECC should serve as benchmark circuits such that a trustworthy comparison can be made between countermeasures concerning the resistance and the associated overhead in execution time, power consumption, area, and byte code.

The experimental result sections of the manuscripts that introduced the macro level [14] and instruction level [15] countermeasures are simply restricted to before and after power consumption profiles based on cycle accurate simulators with an error band of at least 10% when compared with transistor level simulations using extracted interconnect capacitances. Even with such a first order approximation on the power consumption, a DPA attack on for instance the macro level measure would have revealed that the upperbound on the increase of the MTD is a factor two. Indeed, it is sufficient to only utilize the power acquisition when e.g. the macro with the high power dissipation has been selected in the datapath. Because of the random selection process between the two macros, only two times as much measurements are needed to have the required number of measurements of the original MTD. In [16], conclusions are based on 62 power measurement acquisitions. This is insufficient to draw any conclusion. A differential power analysis exploits the law of large numbers to increase the signal to noise ratio. It can still be done in a very short time, compared to exhaustive search of the key space. With our differential power analysis setup, which uses a 2GHz digital sampling oscilloscope and a standard GPIB interface, we have made up to 400 acquisitions a second, including data transfer. Such a setup only requires 4 minutes to make 100,000 power measurements.

In this manuscript, we will describe how at different abstraction levels a DPA attack with associated power simulation model and countermeasure can be conceived. We will show that if one proposes a countermeasure at a certain abstraction level, one needs to assure that the assumptions made are also supported at the lower abstraction levels. Furthermore, in a case study, we will see how the accuracy of the power simulation model significantly influences the outcome of a DPA. The main contributions of this paper are that (1) we point out that an actual DPA must be performed with the correct accuracy on the power simulation model as the quality of the resistance assessment of a countermeasure is only as good as the simulation model; and (2) that we provide some inside information with concrete numbers on the performance and the setup of an actual differential power analysis.

The remainder of this paper is organized as follows. The next section presents first the theory behind a general differential power analysis and subsequently the details of an attack on the AES algorithm. In section 3, simulation models of a DPA attack with power model and countermeasure are described at the following levels of abstraction: (1) instruction level; (2) register transfer level; and (3) transistor level. Section 4 presents the case study. Finally, a conclusion will be formulated.

2. DIFFERENTIAL POWER ANALYSIS

2.1 General Concept

In a DPA attack, measured power traces are compared with a prediction on the power consumption. To make the prediction a guess on the secret key is used. Only if the secret key hypothesis is correct, then the predicted and the actual power consumption are correlated.

The contribution of a component of the encryption module into the total power consumption of the device is estimated through a behavioral model of that component. The model calculates one or more state bits of the component from known plaintext or ciphertext data and from a guess on a subset of the secret key. If the guess was correct, the outcome is always equal to the actual state bits and is therefore correlated with the power consumption of the logic operations that are affected by the state bits. Measurement

errors and the power consumption of the other logic operations are uncorrelated.

Several statistical techniques are available to perform the comparison between the predictions and the measurements. The original DPA uses the distance-of-mean test [6]. In this technique, the measurements are divided over two sets based on the value of one state bit in the behavioral model. At the end, the difference is calculated between the typical supply currents of the two sets. If the difference has noticeable peaks, the guess on the secret key was correct. The peaks are actually the effect of the state bit on the instantaneous power consumption. Another common DPA uses the correlation test [17]. Here, a representative number of the measurements, such as the mean or the maximum supply current in a clock cycle, is correlated with the Hamming distance of two successive values of the state bits or in other words with the number of changing state bits in the behavioral model in a clock cycle. The correct key guess is the one that results in the highest correlation coefficient between the vector of representative measurements and the vector of Hamming distances.

2.2 AES Attack

AES operates on 128-bit data blocks and supports three key sizes (128, 192, and 256 bits). An encryption operation starts with a single addition (xor-operation) of the plaintext and the secret key followed by iteratively applying encryption rounds. An encryption round consists of the following operations: SubByte, ShiftRow, MixColumn and a key addition. The number of rounds depends on the key length. Each round requires a round key that is derived from the secret key. Figure 1 depicts a simple block diagram of the encryption datapath of an implementation of the 128-bit key, 128-bit data version of the AES algorithm. The AES core depicted performs an encryption in 11 cycles, with one round of the algorithm executed per clock cycle. The key scheduling routine is not shown.

AES has been designed with the limited resources of a typical 8-bit processor of a smart card in mind and most operations are byte oriented. Therefore, 8 state bits can be predicted using a guess on 1 key byte. For the attack, the influence of 1 state byte in the datapath on the power consumption of the AES core is estimated. This is done by calculating the number of changing bits of a byte in register RB, which provides the input to and stores the output of each encryption round.

We will compare the state of register RB in round eleven, which is the final encryption round, and the one after that, in which the encrypted data is known. As shown in Figure 1, RB in round eleven (D_{11}) can be found by tracing back the signal obtained after xor-ing the final ciphertext (C_{11}) and a key guess (K_{11}) through both the shift row operation and the substitution box. RB in the next round, during which we perform the supply current measurement, is the final ciphertext (C_{11}). The correct key byte is found by evaluating:

$$\max_{K_{11}} f_{cost}(K_{11}) = \mathrm{corr}(P_{model}, P_{measurement}) \quad (1)$$

$$\textit{where} \quad P_{model} = \mathrm{HamDist}(sub^{-1}(shiftrow^{-1}(K_{11} \otimes C_{11})), C_{11})$$

$$P_{measurement} = \max(I_{supply,11+1})$$

The cost function compares the estimations and the measurements with the correlation test. The correct key guess is the one that results in the highest correlation coefficient between the vector of

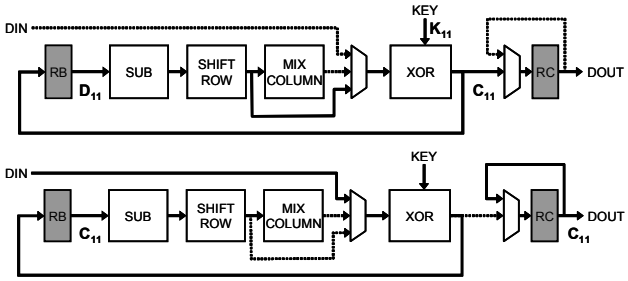

Figure 1. Encryption datapath of AES: round 11 (top); and round 11 + 1 (bottom).

Hamming distances and the vector of representative measurements, for which the maximum supply current in a clock cycle is used. For the remainder of this manuscript, we will refer to the representative measurement (such as maximum or mean value) of one acquisition as the measurement.

Note that the DPA is still an exhaustive search. Yet a divide-and-conquer approach reduces the search space drastically. A brute force attack on the AES algorithm requires 2^{128} key guesses to try all the possibilities of the 128 bit key. The DPA however, working byte per byte, only requires $16*2^8$ key guesses.

3. POWER CONSUMPTION MODEL

A power attack compares a power prediction with a power measurement. The power prediction comes from a fairly simple model of the device. Incorporating much more accuracy into the model than using Hamming weights and/or distances of a few select state bits is hard because in principle the attacker does not know the specifics of the implementation. In order to simulate a power measurement during the design of a secure IC, a power model is also used. This model, however, can be very accurate depending on the stage of the design. We will now discuss several power models at different abstraction levels.

At the instruction level, a measure for the dynamic power consumption is the number of instructions that are executed. Elliptic curve algorithms for instance can be -and have been- attacked because there is a difference in the number of instructions between a double and a double with add operation. The use of these two operations depends directly on the value of the key bits. Making the number of instructions independent of the actual operation being performed thwarts such a power analysis [18].

At the register transfer level, a measure for the dynamic power consumption is the toggle count. This number specifies how many output nets of all the gates make a 0 to 1 transition. Even if the same number of instructions is always executed, there will be a data dependent toggle count. Making the toggle count independent of the data being processed thwarts such a power analysis. In practice, this is done by employing dynamic differential logic, which is also known as dual rail with precharge logic (e.g. [19]). This logic style has a switching factor of 100% and does not suffer from glitches.

At the transistor level, a power consumption model includes the extracted capacitances. Even if the circuit has a constant toggle count there will be a different power consumption depending on the actual data being processed. If the capacitance at the true net is much larger than the capacitance at the false net, the power dissipation depends on which net is being toggled. Capacitance mismatches at the output of dual rail logic, which has perfect security

at the register transfer level, are visible in power attacks. A DPA attack mounted on SPICE simulations of an extracted layout showed that the secret key of a dynamic differential logic DES implementation can be revealed with less than 2,000 measurements if the capacitances do not match [13]. Making the load capacitance of both nets the same thwarts such a power analysis. In practice, this effect is obtained by using custom cells and/or a special place & route approach.

These examples show that the results of a simulated DPA attack depend on the power simulation model. They also show that side-channel resistance cannot be isolated at one abstraction level.

4. CASE STUDY

In this case study, we will assess the security of a protected version of the AES core shown in Figure 1. We will gradually refine the power model used and see that seemingly insignificant second order effects are actually very important.

The AES core has been implemented in constant power consuming logic. Independent of the input stimuli, every logic gate has a single charging event per cycle in which it charges a constant load capacitance. We used Wave Dynamic Differential Logic [19], in which static CMOS standard cells are combined, to implement the desired switching behavior. In order to match the total load at the differential outputs of the dual rail logic gates, the 2 output signals (true and false) are routed with parallel routes that are at all times in adjacent tracks of the routing grid, on the same layers and of the same length [13]. This balances geometric distances and routes both interconnects in the same environment.

A few adjustments must be made to the DPA attack presented in section 2.2. Dynamic logic alternates precharge and evaluation phases, in which all signals are reset to 0 and are computed respectively. Consequently, in order to determine the number of changing state bits, we do not compare the state of register RB between 2 consecutive clock cycles but between the precharge and evaluation phase within a clock cycle. Since all signals are at 0 in the precharge phase, the number of changing bits is simply the number of bits evaluating to 1, or in other words the Hamming weight of RB. To find the secret key, we will compare the power prediction in round 11 (Hamming weight of RB) with the power measurement in the evaluation phase of round 11.

The power measurement is mimicked with a simulation model. Every gate charges exactly one of the two differential output lines to 1 and does not suffer from glitches. As a result, the total load capacitance that is being charged by the AES core in the evaluation phase is the sum of all the individual capacitances attached to the nets that become 1. Because of the power supply inductance and the on-chip decoupling capacitance, the power supply current is a damped sinusoid of which the amplitude is proportional with the total load capacitance being charged. Using the total switched load capacitance is thus a truthful imitation of a real measurement.

To speed up the simulation, we calculated the total load capacitance of the logic cone of each byte in RB for each of its 256 possible states. A logic cone is the combinatorial logic affected by a set of registers and input ports. Since AES has been developed with byte operations in mind, the logic cones of the 16 bytes in RB do not overlap. Charging events between adjacent wires of different logic cones will occur at random and if a sufficient number of measurements are performed they have no effect. We also ignored the key addition. The total switched load capacitance of the AES core for a particular state of RB is the sum of the total load capacitances of each of the 16 bytes in RB. The total load capacitance switched of each byte depends on the particular state of that byte.

The interconnect capacitances have been extracted after place & route with the tool HyperExtract in Silicon Ensemble with Signal Integrity license. With the license, the capacitance per net is reported as one lumped value as well as a list of all the individual capacitances to ground and neighboring wires. For the example in Figure 2, the lumped capacitance C_{lump} is the summation of C_{gnd}, C_i, C_{ib}, C_j and C_{cpl} where C_{gnd} is the capacitance to the power lines and the substrate; C_{cpl} is the cross-coupling capacitance to the other net of the parallel route; and C_i, C_{ib}, and C_j are coupling capacitances to other nearby nets. The gate capacitances C_{gate} come from the library database.

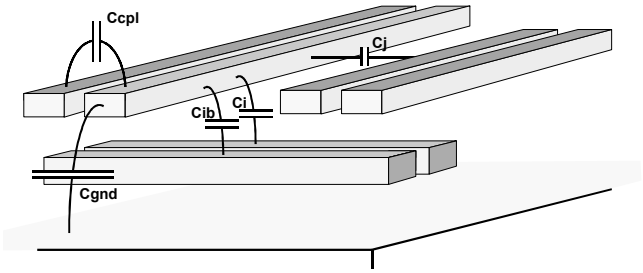

Figure 2. Interconnect capacitance decomposition.

The following four load models, with increasing accuracy, will be used in the power attack:

- *gate*: The capacitance per net is the sum of the gate capacitances it connects to ($C_{net} = C_{gate}$). This model would be used in an early security assessment after synthesis. The inclusion of a wire load model would have no effect. The wire load model would attach the same capacitance value to both the true and false net of a parallel route. (Note that this is even the case for some extraction tools such as Simcap after place & route.) Independent of the switching event of a gate, the same wire capacitance would switch. This would result in a constant contribution into the total load capacitance and would be of no influence in the correlation coefficient. For the same reason, only the input capacitances of the gates are included. Because of the structure of the gates, the intrinsic output capacitance at the true and false net of a gate is the same.
- *lump*: The capacitance per net is the lumped capacitance. ($C_{net} = C_{gnd} + C_{cpl} + C_i + C_{ib} + C_j$). This model would be used in a final security assessment after place & route.
- *gnd*: The capacitance per net is the capacitance to the power lines and substrate. ($C_{net} = C_{gnd}$). This model is a refinement of the lump-model. The lump-model includes all coupling capacitances independent of the state of the neighboring nets. Concentrating on C_{gnd} seems a sound choice. C_{gnd} is order(s) of magnitude larger than coupling capacitances C_j, C_i and C_{ib}. On the other hand, C_{cpl} is rather large because both nets of the parallel routes are always adjacent. Yet, C_{cpl} can be safely ignored. This capacitance will always be charged independent of the switching event of a gate: one of both differential lines will be charged, the other remains at 0.

– *sum*: The capacitance per net is the summation of the capacitances (1) of the gates; (2) to the power lines and substrate; and (3) to the nets that remain at 0 ($C_{net} = C_{gate} + C_{gnd} + (1\text{-}V_i).C_i + (1\text{-}V_{ib}).C_{ib} + (1\text{-}V_j).C_j$ *where* V_k is the value of net k). This model is a refinement of the gnd-model. It includes the coupling capacitances to the neighbouring nets and takes the state of these nets into account. Only if the neighboring wire remains at 0, the capacitance is charged and thus included.

Figure 3 shows the correlation coefficient between the power predictions and the power measurements in function of the number of measurements for each of the four load models. The figure is the result of an attack on the first keybyte. We stopped the simulation at 40,000 measurements. For the gate-model, the correlation coefficient of the secret key crosses the boundary of the correlation coefficients of incorrect key guesses at about 7,000 measurements. The measurements to disclosure is thus 7,000 for this model. The lump-model reports that perfect security is achieved: 40,000 measurements are not sufficient to disclose the secret key. The cross-over point of the gnd-model, on the other hand, is at 1,500 measurements. There is also a large resolution; there is no doubt about the correct key guess. The sum-model, which is the most refined model, assesses the security at 17,500 measurements.

This DPA attack shows that minor differences in the power model can have a big influence in a correct assessment of the security. The lump-model, which is the standard output of HyperExtract and which probably at first sight seemed the best choice as most simple but still correct model, produced an incorrect result as did the gnd-model. They overrate and underrate the security respectively. Even though the individual coupling capacitances are very small and are a second order effect when compared with the gnd capacitance, their total contribution in defining the security is important. Together they have a sufficient large variation to change the variation on the total power consumption.

The variation on the load capacitance of the attacked component, however, is not proportional to the MTD. Figure 4 shows the

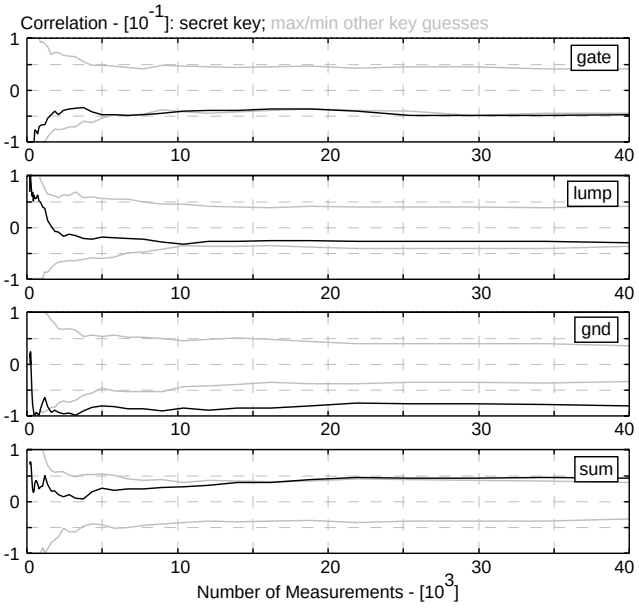

Figure 3. Measurements to disclosure.

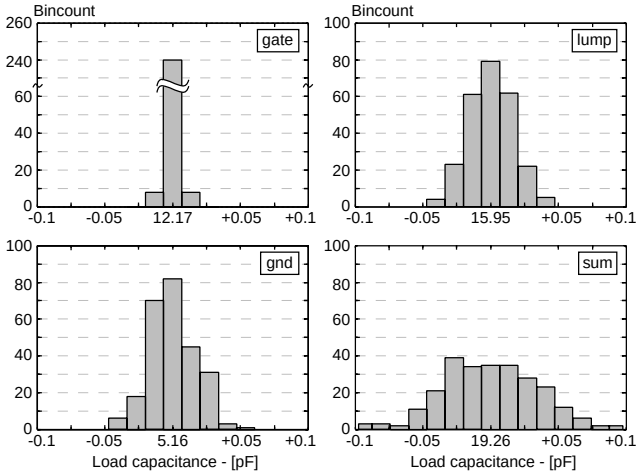

Figure 4. Histogram of the 256 load capacitances.

histogram of the capacitances associated with the 256 possible switching events for the logic cone of the first keybyte. The sum-model has a larger variation on the total load capacitance than the gnd-model. However, the former shows a larger MTD and would predict more resistance than the latter. The same is true for the lump-model and the gate-model. The power consumption variation alone does not tell the complete story and is not sufficient on its own as a figure of merit for DPA resistance.

The counterintuitive observation that less variation does not necessarily mean more security is because the attack correlates a power prediction and a power measurement. If one predicts the power consumption more accurately, the attack will be more successful. Figure 5 shows the normalized variation of the prediction (Hamming weight) and the signal in the measurement (load capacitance of logic cone of the first keybyte). For clarity of the figure, the Hamming weights have been inversed if the power prediction and the power signal correlated negatively (i.e. for gate, lump, and gnd). The gnd-model, which showed the smallest

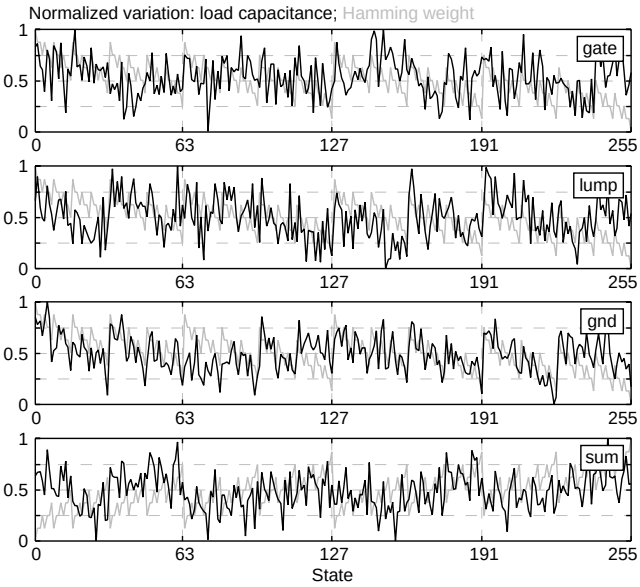

Figure 5. Comparison between power prediction and signal in power measurement.

MTD, has the largest resemblance. The correlation coefficient between the Hamming weights and the gnd-model is -0.117. This number is -0.061, -0.037 and 0.038 for the gate-,lump- and sum-model respectively.

A power signal with a larger variation, however, will be more visible in presence of noisy power variations coming from other modules on the die. The DPA resistance is a combination of both the correlation coefficient and the variation. A Taylor series approximation of the prediction of the MTD in [20] for low SNR and a probability of 0.9999 to discover the secret key shows that:

$$\mathrm{MTD} \approx \frac{28}{\rho_0^2 . \mathrm{SNR}} \qquad (2)$$

The MTD is thus inversely proportional with the square of the correlation coefficient ρ_0 between the power model and the signal in the power consumption. It is also inversely proportional with the signal to noise ratio SNR between the signal in the power consumption and the noise in the power consumption. In the case study, the differences in security came only from differences in ρ_0. The SNR was the same for each case, as all logic cones are independent of each other and have a similar power consumption variation [21]. If other noisy modules would have been present, however, the differences in variation of the signal in the power consumption (shown in Figure 4) would have changed the MTDs.

Constant power consuming logic pursues both a reduction in SNR, by reducing the variation on the signal in the power consumption, while the noise coming from the other modules on the die remains constant, and a reduction in ρ_0, by making the power consumption ideally a constant and thus independent of the power prediction. Other techniques that reduce ρ_0 have the potential disadvantage that another power prediction may accurately predict the signal in the power consumption. A 'random' power consuming countermeasure for instance is very weak if the randomness is not actually random and can accurately be predicted by another model than the Hamming distances. For instance, the randomness introduced in the macro level countermeasure of [14], only changed ρ_0 and the SNR at first sight. By only including the measurements of the macro with the high power consumption in the analysis, ρ_0 and SNR never changed. Influencing ρ_0, and SNR must be done with care.

5. CONCLUSIONS

Side-channel resistance cannot be isolated at one abstraction level. Essential preconditions for high-level countermeasures must hold at the lower abstraction levels. The DPA is a powerful attack; countermeasures based on wrong assumptions on the side-channel information leakage will not stand. The case study has shown that the simulation-model must accurately reflect the reality and that approximations must be made cautiously. The security assessment is only as good and as trustworthy as the model used. Designers must assure that whichever the option is to change the number of measurements to disclosure -whether the correlation between the prediction and the measurements is reduced; or the variation of the signal is decreased; or the variation of the noise is increased- it can not be circumvented by the attacker.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (CCR-0098361).

7. REFERENCES

[1] M. Renaudin, F. Bouesse, P. Proust, J. Tual, L. Sourgen and F. Germain, "High Security Smartcards", DATE, pp. 228-232, February 2004.

[2] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Ravi, "Security as a New Dimension in Embedded System Design", DAC, pp.735-760, June 2004.

[3] A. Shamir, and E. Tromer, "Acoustic cryptanalysis", http://www.wisdom.weizmann.ac.il/~tromer/acoustic/, 2004.

[4] B. Schneier, "A Hardware DES Cracker", Crypto-Gram Newsletter, http://www.schneier.com/crypto-gram-9808.html#descracker, August 1998.

[5] K. Tiri, D. Hwang, A. Hodjat, B. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "A Side-Channel Leakage Free Coprocessor IC in 0.18µm CMOS for Embedded AES-based Cryptographic and Biometric Processing", DAC, June 2005.

[6] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis", CRYPTO, LNCS 1666, pp. 388-397, August 1999.

[7] T. Messerges, E. Dabbish, and R. Sloan, "Examining smart-card security under the threat of power analysis attacks", IEEE TC, Vol. 51, Issue: 5, pp. 541-552, May 2002.

[8] C. Clavier, J. Coron, and N. Dabbous, "Differential Power Analysis in the Presence of Hardware Countermeasures", CHES, LNCS 1965, pp. 252-263, August 2000.

[9] A. Shamir, "Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies", CHES, LNCS 1965, pp. 71-77, August 2000.

[10] N. Pramstaller, F. Gürkaynak, S. Häne, H. Kaeslin, N. Felber, and W. Fichtner, "Towards an AES Crypto-chip Resistant to Differential Power Analysis", ESSCIRC, pp. 307-310, September 2004.

[11] E. Oswald, S. Mangard and N. Pramstaller, "Secure and Efficient Masking of AES – A Mission Impossible?", Report 2004/134 in IACR Cryptology ePrint Archive, June 2004

[12] S. Mangard, T. Popp, and B. Gammel, "Side-Channel Leakage of Masked CMOS Gates", CT-RSA, Feb. 2005.

[13] K. Tiri, and I. Verbauwhede, "Place and Route for Secure Standard Cell Design", CARDIS, pp. 143-158, August 2004.

[14] L. Benini, A. Macii, E. Macii, E. Omerbegovic, F. Pro, et al., "Energy-aware design techniques for differential power analysis protection", DAC, pp. 36-41, June 2003.

[15] H. Saputra, N. Vijaykrishnan, M. Kandemir, M. Irwin, R. Brooks, S. Kim, et al., "Masking the energy behavior of DES encryption", DATE, pp. 84-89, March 2003.

[16] C. Gebotys "Design of secure cryptography against the threat of power-attacks in DSP-embedded processors", ACM TECS, Vol. 3, Issue 1, pp. 92-113, February 2004.

[17] J. Coron, P. Kocher, and D. Naccache, "Statistics and Secret Leakage", FC, LNCS 1962, pp. 157-173, Feb. 2000.

[18] J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," CHES, LNCS 1717, pp. 292-302, August 1999.

[19] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation", DATE, pp. 246-251, February 2004.

[20] S. Mangard, "Hardware Countermeasures Against DPA – A Statistical Analysis of Their Effectiveness", CT-RSA, LNCS 2964, pp. 222 - 235, February 2004.

[21] F. Mace, F. Standaert, I. Hassoune, J. Legat and J. Quisquater, "A Dynamic Current Mode Logic to Counteract Power Analysis Attacks", DCIS, November 2004