DETECTING POWER ATTACKS ON RECONFIGURABLE HARDWARE

Adrien Le Masle, Wayne Luk

Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2BZ, UK
email: {al1108,wl}@doc.ic.ac.uk

ABSTRACT

We present a novel framework to detect power attacks on crypto-systems implemented on reconfigurable hardware. We monitor the device supply voltage with a ring oscillator-based on-chip power monitor. In order to detect the insertion of power measurement circuits onto a device's power rail, a power attack detection strategy taking into account abnormal supply voltages and power rail resistance values is developed. Our strategy is integrated into an on-chip attack detector. The entire framework implementation only takes 3300 LUTs of a Spartan-6 LX45 FPGA, which is 12% of the total area available. Our results on an AES and RSA crypto-system show that our attack detection framework can reach false-positive and false-negative rates as low as 0% over all our test cases if proper operating margins are set.

1. INTRODUCTION

Encryption algorithms are designed to make brute-force attacks or exhaustive key search computationally infeasible and to resist cryptanalysis based on theoretical weaknesses of the algorithm. However, the physical implementation of an encryption algorithm can leak information and create security flaws. Attacks exploiting these physical flaws are called side channel attacks.

Since their initial publication [1], a relevant type of side channel attacks called power attacks have been extensively studied. Power attacks recover the key of the encryption algorithm by using one or multiple power traces, which are directly correlated to the switching of the transistors inside the device. They have been successfully demonstrated on many common encryption methods, including private key encryption such as DES [2] and AES [3], finite field based public key encryption such as RSA [4] and Diffie-Hellman, and elliptic curve based public key encryption [5]. Theoretically, power attacks can be used to attack any cryptosystem with a key-dependent power consumption.

Field Programmable Gate Arrays (FPGAs) are suitable platforms for implementing cryptographic algorithms in particular, and computationally demanding applications in general. The structure of FPGAs makes them particularly fit for pipelined applications, which is the case for most of the basic cryptographic operations. Moreover, a pure hardware implementation of a cryptographic algorithm is inherently less vulnerable than its software counterparts which are usually run in a multi-tasking operating system. However, without adopting suitable countermeasures, an FPGA implementation is as vulnerable to power attacks as its software counterparts running on a processor.

Two major types of countermeasures exist in order to make an implementation resistant to power attacks. Masking countermeasures randomize the intermediate values processed by the cryptographic device. Hiding countermeasures remove the data dependency of the power consumption. The common goal of these two countermeasures is to make the power consumption of the device independent of the encryption key. However, we are not aware of previous work on detecting power attacks targeting reconfigurable hardware.

In this paper, we present a novel approach to detect power attacks on reconfigurable hardware. Our main contributions are:

- A general framework to detect the insertion of a power measurement circuit onto a device's power rail
- An on-chip power monitor circuit to monitor variations in the device supply voltage
- A power attack detection strategy taking into account abnormal supply voltages and power rail resistance values
- An on-chip attack detector circuit implementing this strategy
- An evaluation of our attack detection framework on a simple crypto-system implemented on Spartan-6 FPGA boards

The entire framework implementation only takes 3300 LUTs of a Spartan-6 LX45 FPGA, which is 12% of the total

The support of BlueRISC, Inc, the FP7 EPiCS (Engineering Proprioception in Computing Systems) Project, the UK EPSRC (Engineering and Physical Sciences Research Council), HiPEAC, and Xilinx is gratefully acknowledged.

area available. Our results on an AES and RSA crypto-system show that our attack detection framework can reach false-positive and false-negative rates of 0% over three different test cases for random samples of 100 000 input pairs, if proper operating margins are set.

The rest of the paper is organised as follows. Section 2 explains the background relevant to our work. In section 3, we present our attack detection framework and the implemented attack detection strategy. Section 4 evaluates our framework on a crypto-system built around an RSA and an AES core. Finally, section 5 concludes the paper.

2. BACKGROUND

2.1. Power analysis methods and countermeasures

Power analysis relies on the fact that the energy consumed by a hardware module depends on the switching activity of its transistors. Hence, by measuring the power consumed by a chip performing a given cryptographic operation, an attacker can recover information about the data being processed and the secret keys used.

Simple power analysis (SPA) proceeds by direct observation of a power trace. An implementation whose power consumption is different depending on which bit of the secret key is being processed is vulnerable to SPA. Differential power analysis (DPA) uses statistical properties of multiple power traces. This method is introduced in [1]. DPA relies on the correlation between the power consumption of a module and the intermediate data it is computing at a given time.

In designing a secured hardware-based cryptosystem one needs to incorporate protections against SPA and DPA. In [6], these countermeasures are divided into two groups: masking and hiding countermeasures. Masking countermeasures randomize the intermediate values processed by the cryptographic device. The main goal is to make the power consumption independent of the intermediate values. They have been successfully applied to several encryption algorithms [7, 8].

Hiding countermeasures aim at removing the data dependency of the power consumption. Several hiding countermeasures exist such as power supply filtering, on-chip noise generation [9], wave dynamic differential logic (WDDL) and symmetrical routing [10, 11], and on-chip power regulation [12].

Both types of countermeasures focus on making an attack more difficult by decreasing the correlation between the power consumption and the actual computation. In practise no countermeasure can guarantee the security of the cryptographic system and several countermeasures are often used simultaneously. Hence being able to detect certain types of power attacks will further increase the security of a cryptographic device.

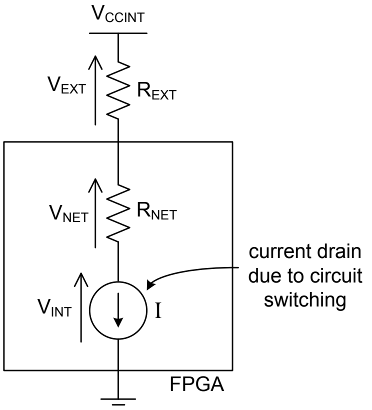

Fig. 1. FPGA power measurement model

2.2. FPGA power measurement

Fig. 1 shows a simplified model of the most common setting to measure the power consumption of an FPGA chip as presented in [5, 6]. A shunt resistor R_{EXT} (typically 1 ohm to 50 ohms) is inserted into the core logic power supply line V_{CCINT} of the FPGA. R_{NET} represents the equivalent resistance of the power rail. I is the current drain due to circuit switching. The power consumed by the FPGA is given by the following equations:

$$P = V_{INT}I = (V_{CCINT} - V_{TOT})I \tag{1}$$

$$V_{TOT} = V_{EXT} + V_{NET} \tag{2}$$

$$R_{TOT} = R_{EXT} + R_{NET} \tag{3}$$

If the voltage drop due to the resistors V_{TOT} is small compared with V_{CCINT}, the power consumption of the device can be approximated by the following equations:

$$P \approx V_{CCINT}I \tag{4}$$

$$I = V_{EXT}/R_{EXT} \tag{5}$$

$$\begin{aligned} I &= V_{TOT}/R_{TOT} \\ &= (V_{CCINT} - V_{INT})/R_{TOT} \end{aligned} \tag{6}$$

As shown in the equations, the power consumption of the FPGA is approximately proportional to the voltage drop across the resistors. An attacker with physical access to the voltage supply pin can obtain a power trace by inserting a shunt resistor R_{EXT} and measuring the voltage drop V_{EXT}.

Ideally the power measurement circuit should not modify the electrical behaviour of the FPGA board. However, R_{EXT} cannot be too small for a power attack to be possible. Even for 1 ohm, the voltage drop across the shunt resistor is not completely negligible. Hence the FPGA supply voltage V_{INT} is smaller than V_{CCINT}. In practise, V_{CCINT} needs to be increased after programming the device to allow the FPGA to run close to normal operating voltage. Even with

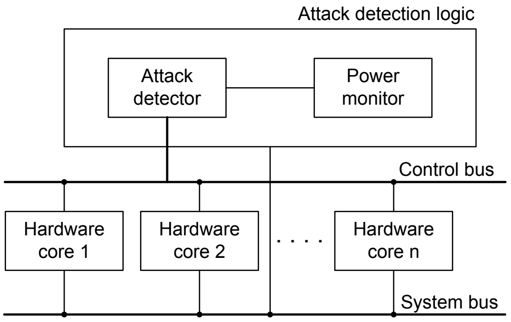

Fig. 2. Power attack detection framework

this compensation the FPGA supply voltage V_{INT} cannot be kept constant at all time. In fact, when some computation is running on the device, I and therefore V_{EXT} vary due to circuit switching. The variations in V_{EXT} are what enables the attacker to obtain power traces. However they also create variations in the FPGA supply voltage V_{INT} which would not occur without the shunt resistor.

The power consumption of a device can also be measured using an electromagnetic (EM) probe or a contactless current probe. In this paper, we only focus on the power measurement circuit of Fig. 1.

3. POWER ATTACK DETECTION FRAMEWORK

The integration of our power attack detection framework into a typical crypto-system is shown in Fig. 2. A typical System-on-Chip (SoC) for cryptography consists of several hardware cores communicating through a system bus. Some cores implement critical cryptographic functions such as RSA, AES, or random number generation. Other non-critical cores are used to handle generic tasks such as communication (UART, Ethernet cores, ...), clock generation, etc. Usually a simple processor core controls the system. Our attack detection logic consists of two main modules:

Power monitor. The power monitor measures the on-chip power of the FPGA. Its input is a value correlated to the on-chip power consumption, such as the average supply voltage across the power network of the FPGA. Its output is a value proportional to the input that can be easily interpreted by the attack detector.

Attack detector. The attack detector receives information about the state of each hardware module in the system. Using the information given by the power monitor and the knowledge of which hardware modules are running by reading the control bus, the attack detector checks whether the power consumption of the device stays within a pre-defined range.

The following sections describe our implementations of the power monitor and the attacks detector in details.

3.1. Ring oscillator-based power monitor

As shown in section 2.2, a power measurement circuit creates variations in the FPGA supply voltage V_{INT} that would not typically occur without shunt resistor. Ring oscillators (ROs) are perfect candidates to monitor these variations. Since the circuit switching speed of an FPGA is correlated with its supply voltage V_{INT}, the oscillation frequency of a RO is affected by the supply voltage [13]. A linear approximation can be used to model the relationship between the FPGA supply voltage V_{INT} and the oscillation frequency f_R:

$$f_R \approx k_0 V_{INT} + f_0 \tag{7}$$

where k_0 and f_0 are positive constants. Note that f_R also depends on the chip temperature. However, for a low number of inverters in the RO, we can neglect the variations with temperature compared to the variations with voltage [13].

As described in [12], the major challenge of using a RO to measure the FPGA's supply voltage is the trade-off between resolution and response time. In order to obtain a sufficient resolution, we need to accumulate enough oscillations from the RO. This implies running the RO for a long period of time, which increases the measurement period. However, increasing the measurement period decreases the number of measurements that can be taken per second and therefore reduces the sampling rate of the power monitor. The solution presented in [12] is to evenly distribute a network of ROs among the FPGA. The oscillations from each RO are accumulated locally during a fixed amount of time. All the accumulated values are then summed together and used as the power measurement. This solution allows a much better resolution and a higher sampling rate at the expense of some area overhead. It provides a more consistent measurement because the effect of voltage variations within the FPGA is averaged. Moreover by averaging the values of uniformly distributed ROs, we also reduce the influence of random wire delay variations on the power monitor reading.

Some recent FPGAs such as the Xilinx Virtex 6 have system monitor capabilities through ADCs that could also be used to monitor the supply voltage V_{INT} [14]. However the sampling rates of these ADCs are low (200 kHz for the Virtex 6 system monitor [15]). A RO-based power monitor has a much higher sampling rate (around 8 MHz in our implementation) that can be scaled according to the area available. Moreover, ROs can be built using primitives that are available to all commercial FPGAs.

3.2. Attack detector

The attack detector has two operating modes: calibration mode and monitoring mode. In calibration mode, the power characteristics of each hardware core are determined. These

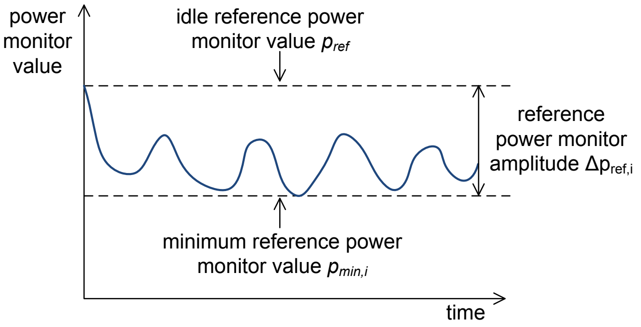
power monitor value
idle reference power monitor value p_{ref}
reference power monitor amplitude $\Delta p_{ref,i}$
minimum reference power monitor value $p_{min,i}$
time

Fig. 3. Calibration of a hardware core

power characteristics depend on the FPGA board on which the system is running, in particular on the design of the power supply circuit of the board. Hence calibration is FPGA-dependent and has to be performed for each type of boards on which the crypto-system is implemented.

First p_{ref}, the reference power monitor value when the system is idle is determined. In the idle state, the switching activity of the transistors and therefore the current I and the voltage drop V_{TOT} are minimum. Here no measurement circuit is present, therefore the voltage drop only depends on the equivalent resistance of the power rail R_{NET}, and V_{TOT} is equal to V_{NET}. A low voltage drop leads to a high power monitor value.

Then we determine the minimum reference power monitor value $p_{min,i}$ of each core i, corresponding to the highest voltage drop. In this step we provide each core with a sample of inputs representative of the operating conditions of the core. The reference power monitor amplitude of core i is:

$$\Delta p_{ref,i} = p_{ref} - p_{min,i} \tag{8}$$

p_{ref}, $p_{min,i}$ and $\Delta p_{ref,i}$ are shown in Fig. 3. In most cases we can only find approximations for these values as all the possible input values of the cores cannot be tested. Hence, we allow margins on p_{ref} and $\Delta p_{ref,i}$, respectively m_{ref} and $m_{ref,i}$. We define:

$$p^*_{ref} = p_{ref}(1 + m_{ref}) \tag{9}$$
$$\Delta p^*_{ref,i} = (p^*_{ref} - p_{min,i})(1 + m_{ref,i}) \tag{10}$$

In monitoring mode, the attack detector is given p^*_{ref} and $\Delta p^*_{ref,i}$ for each core i. The attack detector monitors the instantaneous power monitor reading $p(t)$ and the instantaneous power amplitude:

$$\Delta p(t) = p^*_{ref} - p(t) \tag{11}$$

The attack detector records which hardware modules are currently running by reading their `start` and `done` signals on the control bus.

We assume that at time t, a subset $S(t)$ of the n hardware cores are running. An attack flag is raised if:

$$p(t) > p^*_{ref} \text{ or} \tag{12}$$
$$\Delta p(t) > \sum_{i \in S(t)} \Delta p^*_{ref,i} \tag{13}$$

Eqn. 13 uses the fact that several cores running in parallel should not reach an amplitude higher than the sum of their reference amplitudes. This is an upper bound limit. One could also obtain reference values for every possible combination of cores running in parallel. However this may not be possible for a crypto-system with a large number of cores.

On one hand, the reference power monitor amplitude of a core $\Delta p_{ref,i}$ is a relative value. $\Delta p_{ref,i}$ is proportional to the total resistance R_{TOT} on the reference FPGA power rail. As a matter of fact, let us assume that the switching of hardware core i leads to a variation of current:

$$\Delta I_i = I_{i,max} - I_{i,min} \tag{14}$$

If V_{CCINT} is fixed, the maximum and minimum FPGA supply voltages are respectively:

$$V_{INT,max} = V_{CCINT} - R_{TOT} I_{i,min} \tag{15}$$
$$V_{INT,min} = V_{CCINT} - R_{TOT} I_{i,max} \tag{16}$$

Hence the maximum variation of supply voltage during the operation of core i is:

$$\Delta V_{INT} = R_{TOT} \Delta I_i \tag{17}$$

Using eqn. 7 the variation in oscillation frequency of the power monitor ring oscillators is:

$$\Delta f_R \approx k_0 \Delta V_{INT} = k_0 R_{TOT} \Delta I_i \tag{18}$$

As Δf_R is directly proportional to $\Delta p_{ref,i}$, a change in R_{TOT} approximately leads to a proportional change in $\Delta p_{ref,i}$. Clearly at a fixed time t and under the same configuration for each core, this reasoning also applies to $\Delta p(t)$. In particular, adding a shunt resistor R_{EXT} to the reference device in order to perform a power attack will lead to Δp being higher than $\Delta p_{ref,i}$ at a certain time t_d, as $\Delta p_{ref,i}$ have been obtained on the reference device without shunt resistor. This will be detected by eqn. 13.

On the other hand, the idle reference power monitor value p_{ref} and the instantaneous power monitor reading $p(t)$ are absolute values. The combination of eqn. 12 and eqn. 13 fixes $p(t)$ in a given range. This makes sure that an attacker in not tampering with the supply voltage of the FPGA.

Fig. 4 shows the different operating conditions detected by the attack detector. In the time frame of this graph, we assume that the number of hardware cores running in parallel is kept constant. Let us define:

$$p_{min}(t) = p^*_{ref} - \sum_{i \in S(t)} \Delta p^*_{ref,i} \tag{19}$$

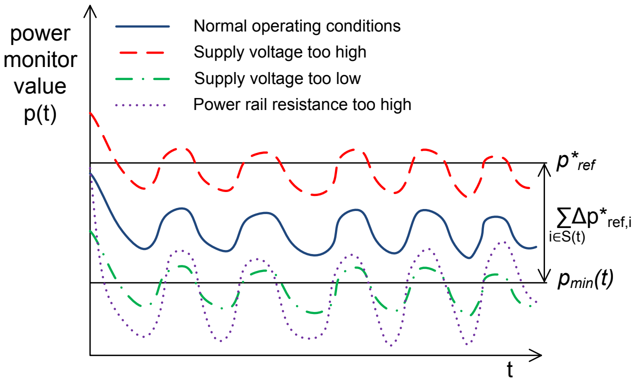

Fig. 4. Operating conditions monitored by the attack detector

In normal operating conditions, at time t the power trace $p(t)$ always stays between p^*_{ref} and $p_{min}(t)$. If the supply voltage is too high, p raises over p^*_{ref} and the attack flag is raised according to eqn. 12. If either the supply voltage is too low or the supply voltage is at a normal level but the power rail resistance is too high, p falls below p_{min} at a time t_d. Hence $\Delta p(t_d)$ is higher than $\sum_{i\in S(t_d)} \Delta p^*_{ref,i}$ and the attack flag is raised according to eqn. 13.

Depending on the application, different strategies can be adopted when an attack is detected. The critical cores can be reset to prevent the current power acquisition from being carried on, new keys can be generated making the corrupted keys useless, or the keys can be erased from the cryptographic device, making the device unusable.

4. RESULTS

Our experimental setting is based on two Pico E-101 FPGA boards. The Pico E-101 has a Spartan-6 LX45 FPGA. One of the boards is modified with a power measurement circuit consisting of a 1 ohm shunt resistor inserted in the 1.2V power line. The 3.3V and 1.2V switching regulators are also replaced with more stable low dropout regulators. The 1.2V can be adjusted via a variable resistor. We use a Tektronix MSO 2024 oscilloscope with a 200 MHz bandwidth and a 1 GHz sampling rate for all our measurements.

We consider a crypto-system with five main cores running at 20 MHz: the power attack detection system, a 512-bit RSA core (core number 0) [16], a 128-bit AES core (core number 1) [17], a Microblaze processor controlling the different cores and a UART core for communication.

The power monitor has 144 1-inverter ring oscillators uniformly distributed on the FPGA and placed and routed the exact same way. The power monitor reading is updated at 8 MHz. We determine that such a power monitor can detect variations of supply voltage as low as 5 mV. The entire attack detection framework consisting of the power monitor and the attack detector only takes 3300 LUTs. This represents 12% of the area available. For comparison, standard countermeasures such as masking and WDDL have area overheads of respectively 20% to 300% [7] and 3 to 10 times [10, 11].

The RSA and AES cores are calibrated using respectively 100 and 1000 random input pairs on the original Pico E-101 board. The UART is only used to get the RSA and AES inputs from a PC and to send the results back after the computation is finished. Hence the UART core is never running in parallel with the RSA or the AES core. The processor only waits for an interrupt during an RSA or AES computation and we can therefore neglect its power consumption during this phase. The calibration values obtained are:

$$p_{ref} = 7853 \quad p_{min,0} = 7496 \quad p_{min,1} = 7508 \qquad (20)$$

We consider three test cases. The first two cases are respectively an RSA and an AES encryption working alone. The last case consists of several AES encryptions working in parallel with an RSA encryption. Note that an AES encryption finishes in 8 clock cycles whereas an RSA encryption finishes in between 275 968 and 551 936 clock cycles depending on the number of ones in the 512-bit key. We consider 100 000 runs of each of the three test cases for a random set of inputs using a different seed from the one used in the calibration phase.

Three operating conditions are considered: the original board on which the cores have been calibrated, the modified board with a higher supply voltage ($V_{INT} = 1.25V$) and the modified board with a shunt resistor $R_{EXT} = 1\Omega$. Operating conditions with a supply voltage lower than 1.2V cannot be tested due to the characteristics of the low-dropout regulator used on our modified board. For $R_{EXT} = 1\Omega$, V_{INT} is compensated so that the value of p in the idle state is as close to p^*_{ref} as possible. This ensures that an attack flag is not raised because of a low supply voltage as shown by the green dotted/dashed curve in Fig. 4. This compensation requires access to the value of the power monitor, which an attacker would not acquire easily. In practise an attacker would simply compensate to set V_{INT} to its nominal value when the system is idle. This is less precise than our fine tuning using the power monitor value. Hence this case is a best case scenario in an attacker's point of view.

The results are reported in Table 1. In the left part of the table, we only consider the effect of eqn.12. If no margin is allowed on p_{ref}, we see that the detector gives false-positives (respectively 3.8%, 0.11% and 1.8% for RSA, AES and RSA+AES), that is the attack flag is raised when the crypto-system is operating in normal conditions. If the margin m_{ref} on p_{ref} is too wide, false-negatives start to appear. For $m_{ref} = 5\%$, the raise in the power supply ($V_{INT} = 1.25V$) is not detected in more than 98% of the cases. A margin of 1% does not lead to any false-positives or false-negatives. In the right part of the table, we set $m_{ref} = 1\%$ and we study the effect of eqn.13. With no margin $m_{ref,i}$ on the $\Delta p_{ref,i}$,

Table 1. Detected attacks (% of total runs - of which % of high voltage detections)

		p_{ref}	$p_{ref} + 1\%$	$p_{ref} + 5\%$	$\Delta p_{ref,i}$	$\Delta p_{ref,i} + 10\%$	$\Delta p_{ref,i} + 50\%$
RSA	Original	3.8 - 100	0 - NA	0 - NA	98.6 - 0	0 - NA	0 - NA
	$V_{INT} = 1.25V$	100 - 100	100 - 100	0 - NA	100 - 100	100 - 100	100 - 100
	$R_{EXT} = 1\Omega$	0.001 - 100	0.001 - 100	0 - NA	100 - 0	100 - 0.004	100 - 0.006
AES	Original	0.11 - 100	0 - NA	0 - NA	1.6 - 0	0 - NA	0 - NA
	$V_{INT} = 1.25V$	100 - 100	100 - 100	0.13 - 100	100 - 100	100 - 100	100 - 100
	$R_{EXT} = 1\Omega$	0.001 - 100	0.001 - 100	0 - NA	100 - 0.004	100 - 0.004	99.7 - 0.004
RSA+AES	Original	1.8 - 100	0 - NA	0 - NA	2.7 - 0	0 - NA	0 - NA
	$V_{INT} = 1.25V$	100 - 100	100 - 100	0.02 - 100	100 - 100	100 - 100	100 - 100
	$R_{EXT} = 1\Omega$	0.001 - 100	0.001 - 100	0 - NA	100 - 0.02	100 - 0.02	100 - 0.003

the false-positive rate can be as high as 98.6%. As soon as we set a margin of at least 10% this rate drops to 0%. False-negatives start to appear from $m_{ref,i} = 50\%$ for AES where the shunt resistor R_{EXT} is not detected 0.3% of the cases. Note that the voltage compensation when $R_{EXT} = 1\Omega$ leads to a few high voltage conditions being detected.

We can conclude from Table 1 that for margins $m_{ref} = 1\%$ and $m_{ref,i} = 10\%$ the attack detector has false-positive and false-negative rates of 0% across all our experiments.

5. CONCLUSION

This papers presents a novel framework to detect power attacks on crypto-systems implemented on reconfigurable hardware. We monitor the device supply voltage with a ring oscillator-based on-chip power monitor. In order to detect the insertion of power measurement circuits onto a device's power rail, a power attack detection strategy taking into account abnormal supply voltages and power rail resistance values is developed. Our strategy is integrated into an on-chip attack detector. Our results on an AES and RSA crypto-system implemented on a Spartan-6 LX45 FPGA shows that our attack detection framework can reach false-positive and false-negative rates of 0% if proper operating margins are set. The implementation of the framework only takes 3300 LUTs, which is 12% of the total area available.

Current and future work includes testing our method for lower shunt resistor values, confirming the negligible effect of temperature experimentally, extending our framework to the power consumption of individual processor instructions, investigating other on-chip power measurement methods, and exploring attack detection of electromagnetic attacks.

6. REFERENCES

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *CRYPTO '99*, pp. 789–789.

[2] F.-X. Standaert, S. B. Ors, J.-J. Quisquater, and B. Preneel, "Power analysis attacks against FPGA implementations of the DES," in *FPL '04*, pp. 84–94.

[3] F. Standaert, F. Mace, E. Peeters, and J. Quisquater, "Updates on the security of FPGAs against power analysis attacks," in *ARC '06*, pp. 335–346.

[4] T. Messerges, E. Dabbish, and R. Sloan, "Power analysis attacks of modular exponentiation in smartcards," in *CHES '99*, pp. 724–724.

[5] S. Ors, E. Oswald, and B. Preneel, "Power-analysis attacks on an FPGA: First experimental results," in *CHES '03*, pp. 35–50.

[6] S. Mangard, E. Oswald, and T. Popp, "Power analysis attacks: Revealing the secrets of smart cards," Springer (2007).

[7] F. Regazzoni, Y. Wang, and F.-X. Standaert, "FPGA implementations of the AES masked against power analysis attacks," in *COSADE 2011*, pp. 56–66.

[8] C. Rebeiro and D. Mukhpodhyay, "Power attack resistant efficient FPGA architecture for Karatsuba multiplier," in *VLSID '08*, pp. 706–711.

[9] T. Güneysu and A. Moradi, "Generic side-channel countermeasures for reconfigurable devices," in *CHES '11*, pp. 33–48.

[10] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *DATE '04*.

[11] P. Yu and P. Schaumont, "Secure FPGA circuits using controlled placement and routing," in *CODES+ISSS '07*, pp. 45–50.

[12] A. Le Masle, G. C. T. Chow, and W. Luk, "Constant power reconfigurable computing," in *FPT 2011*.

[13] J. Franco, E. Boemo, E. Castillo, and L. Parrilla, "Ring oscillators as thermal sensors in FPGAs: Experiments in low voltage," in *SPL '10*, pp. 133 –137.

[14] Xilinx, "Virtex-6 FPGA System Monitor, User Guide UG370 (v1.1)," June 2010.

[15] ——, "Virtex-6 FPGA Data Sheet: DC and Switching Characteristics, DS152 (v3.4)," January 2012.

[16] A. Le Masle, W. Luk, J. Eldredge, and K. Carver, "Parametric encryption hardware design," in *ARC '10*, pp. 68–79.

[17] "Aoki laboratory webpage," http://www.aoki.ecei.tohoku.ac.jp/crypto/.