Cryptography Research, Inc
575 Market Street, 11th Floor
San Francisco, CA 94105

Phone 415 397 0123
Fax 415 397 0127
www.cryptography.com

Cryptography **Research Whitepaper**

Protecting FPGAs from Power Analysis

Version 1.0
April 20, 2010

Overview

Recent advances in the size and performance of FPGAs, coupled with advantages in time-to-market, field-reconfigurability and lower up-front costs, make FPGAs ideally suited to a wide range of commercial and defense applications [6]. In addition, FPGAs' generality and reconfigurability provide important protections against the introduction of Trojan horses during semiconductor manufacturing process[8]. As a result, FPGA applications increasingly involve highly-sensitive intellectual property and trade-secrets, as well as cryptographic keys and algorithms [7].

For such applications, FPGAs need to achieve a high level of tamper resistance in order to preserve confidential information and ensure system integrity. Systems that utilize FPGAs for cryptography may also need to comply with tamper-resistance security standards, including applicable Common Criteria protection profiles as well as the upcoming U.S. government FIPS 140-3 standard.

Non-invasive attacks, including both simple and differential power analysis (SPA and DPA), must be addressed by all FPGA-based systems that require any significant degree of tamper resistance. Power analysis attacks can be carried out by attackers with modest skill and resources, since power measurements can be collected and analyzed easily. If a design is not adequately protected, secrets such as sensitive data, IP, trade-secrets and cryptographic keys can be extracted, and adversaries could make unauthorized modifications to the device configuration.

This whitepaper introduces SPA and DPA, discusses how these vulnerabilities apply to FPGAs, and provides guidance about the types of countermeasures that can be implemented to protect FPGAs against these attacks.

Introduction to Simple and Differential Power Analysis

The energy consumed by a hardware device such as an FPGA depends on the switching activity of its transistors, which in turn depends on the operations it is performing. An attacker who is passively measuring a device's power consumption or electromagnetic emissions will recover some aggregated and noisy information related to the sensitive data being processed. SPA and DPA attacks [1] use the information obtained from power measurements to extract secret keys from a device.

SPA attacks recover the secret keys by directly observing features within individual power consumption measurements. Implementations that have significantly different power consumption depending on secret key bits are most vulnerable to SPA. For example, implementations of modular exponentiation for RSA or Diffie-Hellman commonly use a key-dependent sequence of square and multiply operations. Similarly, implementations of scalar multiplication in elliptic curve cryptosystems (ECC) generally use a key-dependent sequence of double and add operations. In each case, the pattern of these operations reveals the value of the key. For unprotected devices, this pattern can be observed from a single operation.

Figure 1 shows the power trace from an RSA operation using a standard square and multiply sequence. The square and multiply operations have visibly different power profiles that are easy to distinguish. The secret exponent can be recovered easily from the sequence of squares and multiplies. In particular, each '1' in the secret exponent consists of a squaring step (lower power) followed by a multiplication step (higher power), while a '0' in the exponent involves only a squaring step (lower power). In Figure 1, steps involved in squaring operations have been highlighted in green, while steps involved in multiplication are highlighted in red.

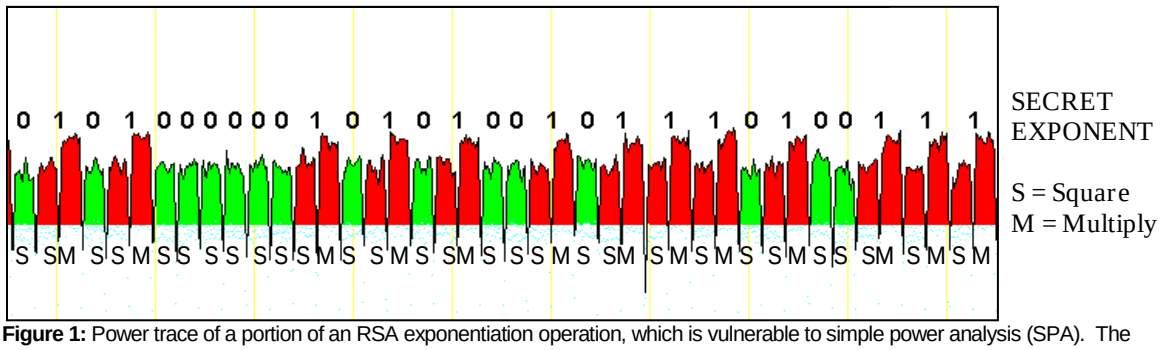

Figure 1: Power trace of a portion of an RSA exponentiation operation, which is vulnerable to simple power analysis (SPA). The square and multiply sequence are shown along with bits of the recovered secret exponent.

DPA attacks employ statistical techniques that combine multiple power consumption measurements to extract secrets. DPA is effective even when the information available from any individual cryptographic transaction is small and masked by other activity and noise. The basic concept behind DPA is that the overall power consumption of a device at a point in time is correlated to the computational intermediates it is processing at that time. By focusing on intermediates that depend only on a few bits of the key, it is possible to use power measurements to determine those bits of key. For every possible value of these key bits, one can predict the computational intermediate then look for correlations between the power measurements and bits of the predicted intermediate.

As shown in Figure 2, for a correct value of these key bits, correlation spikes are observed whenever the predicted intermediate is being processed. For all other (incorrect) values of the key bits, there are no correlation spikes, or the spikes are much smaller. Once these key bits are determined, the same divide-and-conquer approach can be repeated with other intermediates to determine the other bits of the key.

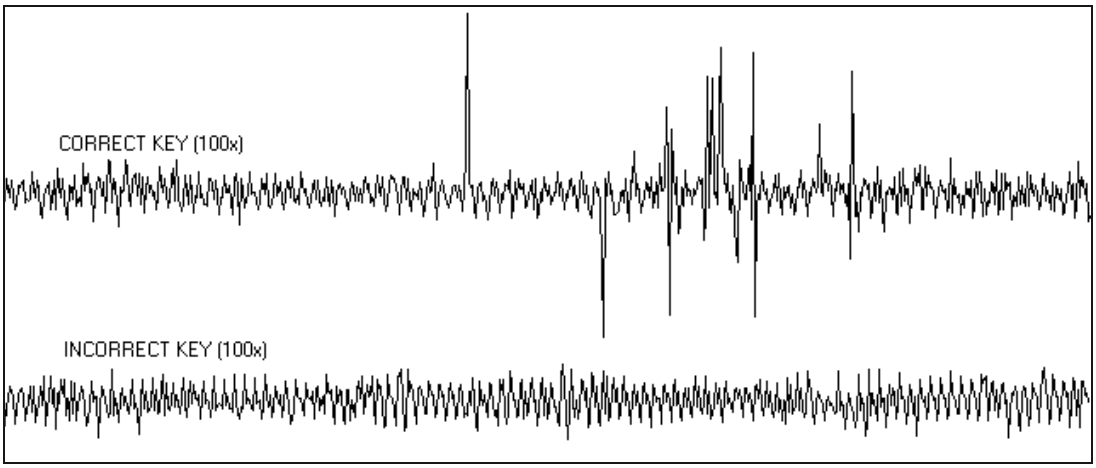

Figure 2: DPA: Correlation of power traces with a predicted intermediate for a correct guess (top) versus an incorrect guess (bottom).

The number of measurements required for a successful DPA attack against a given key depends on the signal-to-noise properties of the target system. Because the entire process can be automated easily, attacks involving even several million operations are straightforward using an off-the-shelf digital storage oscilloscope for data acquisition and an ordinary PC for analyzing the data.

Power Analysis Vulnerabilities of FPGAs

Background

Numerous papers have been published about power-analysis vulnerabilities of FPGA implementations. One of the earliest works [2] dealt with SPA attacks on Elliptic Curve Cryptography implemented on an FPGA. Other papers presented DPA attacks on the Advanced Encryption Standard (AES)[3] and Data Encryption Standard (DES)[4] on the same platform. While the details of exploitable leaks for FPGA implementations differ somewhat from those typically found on unprotected microprocessor or smart-card based implementations, the basic principles of SPA and DPA apply equally to cryptographic implementations in software, FPGAs, or ASICs. The flexibility and low cost of FPGAs make them a preferred platform for researchers investigating the power analysis vulnerabilities of cryptographic implementations, as well as for assessing the effectiveness of countermeasures. The FPGA-based Side-channel Attack Standard Evaluation Board (SASEBO) [5], developed by AIST, the National Institute of Advanced Industrial Science and Technology of Japan, is being used by several research groups to investigate power analysis.

SPA/DPA vulnerabilities in FPGAs may occur at the platform level or within the logic downloaded into the fabric. The following sections review the major issues at each level.

Platform-Level Vulnerabilities

Modern FPGA platforms provide a variety of hard IP blocks and features. Some FPGAs include microprocessors and math blocks, as well as security features such as bitstream decryption, password protection for system and security configuration data, protected segments, and storage for keys. These platform-level components and functionality are usually implemented within standard cell hardware or using an embedded microprocessor with on-chip ROM, rather than using the programmable fabric. In the absence of effective power analysis countermeasures, these components and their associated security protections could be subverted using SPA and DPA. Normally, vulnerabilities in these features cannot be corrected by end users of the FPGA and are common to all FPGAs of a given design, so the impact of vulnerabilities in platform-level elements can be severe. In some cases, these problems could be mitigated by avoiding or locking out the usage of vulnerable features or by using one-time programming.

Vulnerabilities in the FPGA Fabric

There is a large body of published literature showing how to mount power analysis attacks against different cryptographic algorithms and implementations on FPGAs. These papers show that, in the absence of countermeasures, cryptographic implementations in the FPGA fabric are highly susceptible to DPA attacks, and in some cases may also fall to SPA attacks.

As an example, Figure 3 shows a DPA attack on the sample AES implementation provided with the SASEBO-GII platform [6]. This is a straightforward efficient implementation of AES-128, with one round per clock cycle and the clock running at 24MHz. The top trace shows the average power trace from 10000 encryption operations, measured using a 1 • resistor at the V_{CC} side. The 11 dips correspond to the 11 clock cycles it takes to perform the AES operation. (For this implementation, there is an initial XOR between the key and data followed by 10 rounds). The bottom trace shows a correlation of power traces with a predicted intermediate at the beginning of round 10, for the correct guess of a key byte. The sharp rising edge in the correlation trace at the beginning of round 10 confirms that the guessed key byte is correct.

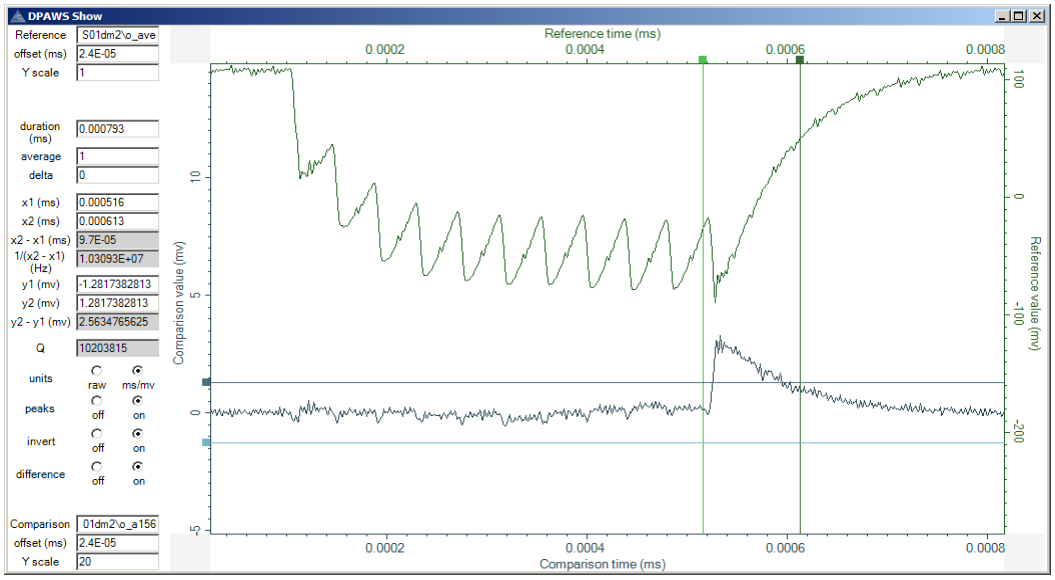

Figure 3: A DPA attack on the sample SASEBO AES implementation. The top trace is the average AES trace, and the bottom trace shows the correlation between power traces with a predicted round 10 intermediate for a correct guess of a key byte.

The DPA attack above was performed by externally invoking the AES block encryption operation on SASEBO approximately 10000 times with an unknown key and random data. Thus, less than 5ms of actual cryptographic computation time was observed, and a minute of processing time on a PC was required for the analysis to extract the entire 16-byte key using the Cryptography Research DPA Workstation™ analysis software.

In comparison with smart cards and other bandwidth-limited devices, adversaries can collect much larger data sets from FPGAs that perform bulk encryption or decryption. For example, Figure 4 shows a portion of an unprocessed power trace from an AES implementation on SASEBO-GII that performs bulk encryption using AES in CBC mode. A single power trace during bulk encryption contains tens of thousands of individual block encryption operations (or more) and can be collected and transferred to a PC in a matter of seconds. Figure 5 shows the results of a successful DPA attack using a single trace, which was performed by first decomposing the trace into its constituent block encryption operations then using DPA to analyze those operations.

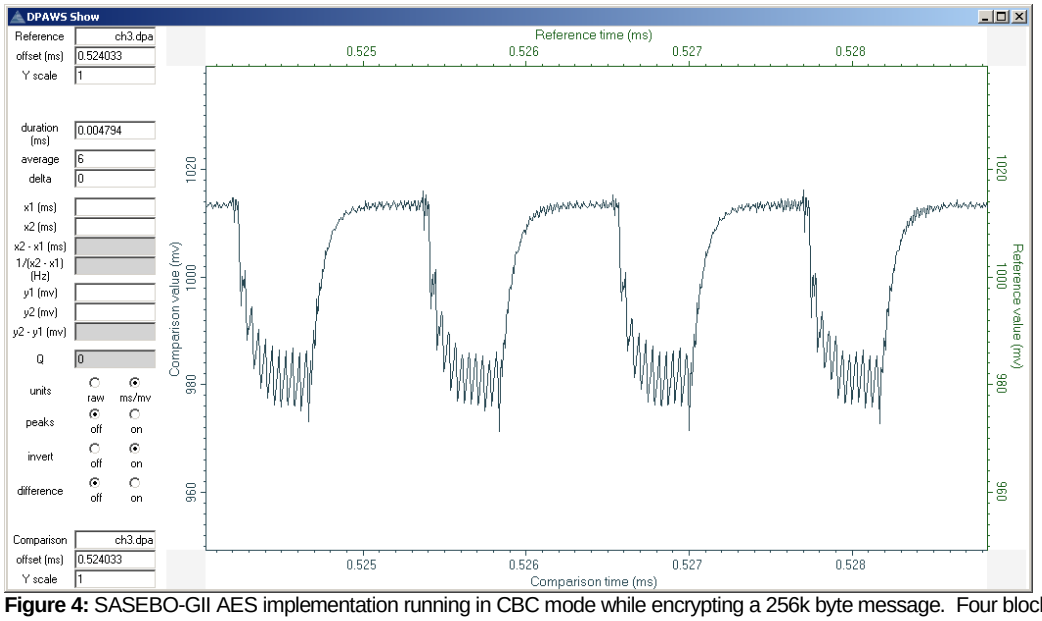

Figure 4: SASEBO-GII AES implementation running in CBC mode while encrypting a 256k byte message. Four blocks pictured, and the full trace contained measurements from 16384 AES block encryption operations.

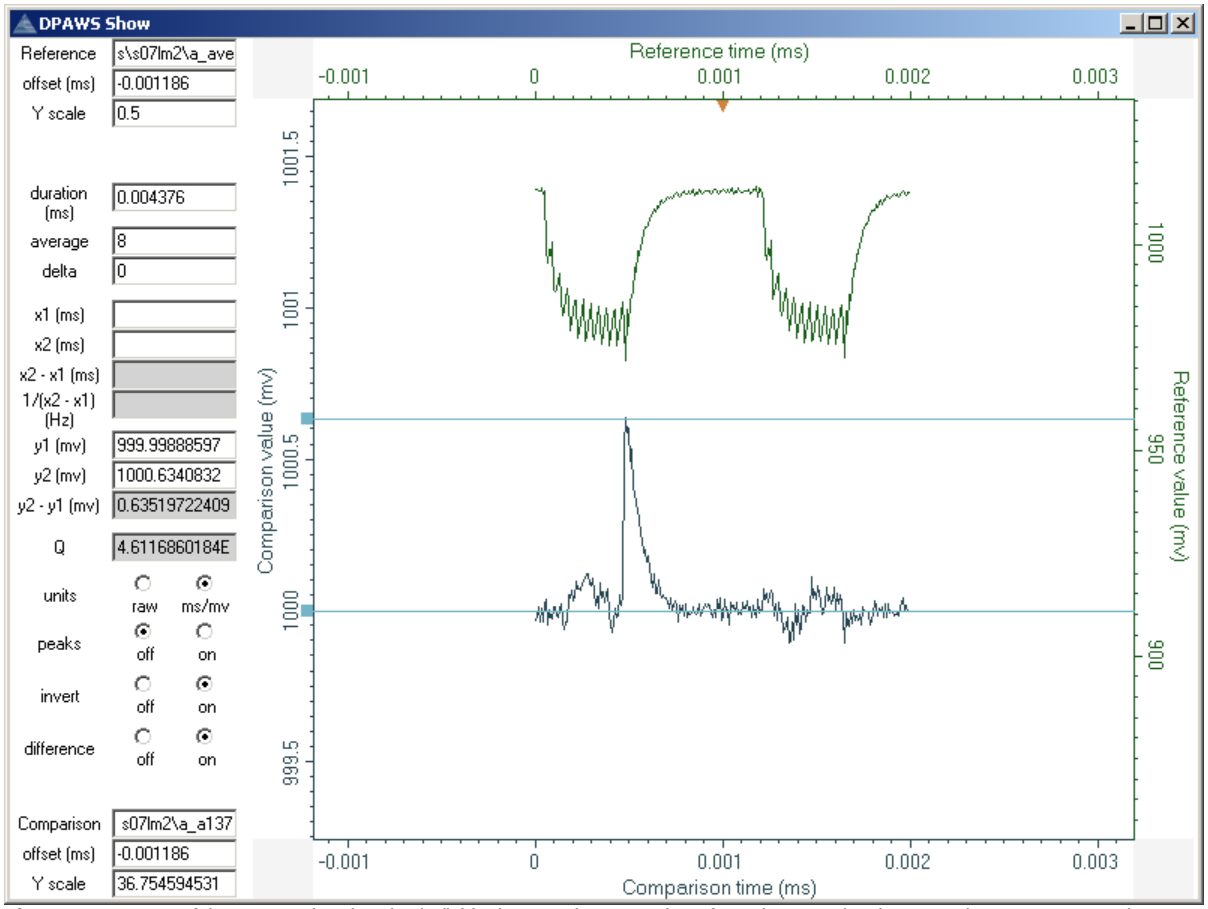

Figure 5: A successful DPA attack using the individual encryption operations from the trace in Figure 5. The average trace is shown on top and the correlation trace for the correct guess (137) for byte 7 of round key in the last round is shown below.

Protecting FPGAs from Power Analysis

Cryptography Research discovered SPA and DPA in the 1990s and developed the fundamental techniques for securing systems against these attacks. Defending against DPA requires the careful application of countermeasure techniques, but many existing products have implemented countermeasures and have passed stringent requirements and tests for DPA resistance. These include a range of commercial security products such as chips used in smart cards, electronic passports, trusted platform modules (TPMs), and others. These same set of techniques can generally be used to protect FPGA-based cryptographic implementations from attacks.

At a high level, general categories of countermeasures to DPA include:

- Leakage reduction: These techniques make the set or sequence of operations less dependent on the key or secret intermediates. Balancing techniques to reduce variation in the power consumption can also be employed, although using these methods on FPGAs may require extra care due to asymmetries within the routing infrastructure. The overall goal of leakage reduction strategies is to reduce the leakage signal-to-noise ratio, increasing the number of power measurements an adversary would require for a successful attack.

- Noise introduction: These techniques add different types of noise into the power consumption measurements available to the attacker, reducing the leakage-signal to noise ratio. Noise can be generated in the amplitude domain (e.g., by consuming random amounts of power) or in the temporal domain (e.g., by randomizing operation timing). As with leakage reduction, these countermeasures increase the number of power traces required by an adversary.

- Obfuscation: By keeping algorithms secret, the attacker is forced to perform reverse engineering along with power analysis. Such countermeasures typically do not provide any security once an adversary understands the operation of the obscure function, but can increase the initial effort required for an attack. Because the cost of subsequent attacks is not increased, obfuscation-based countermeasures should be used with caution, but still may be better than having no protection at all.

- Incorporating randomness: These categories include a broad range of techniques for randomizing the data manipulated by the device in ways that still produce the correct result. For public key systems, techniques for masking or blinding of data and keys can be particularly effective. Similarly, for symmetric algorithms such as AES, techniques for masking intermediates and tables can be effective. These techniques force the attacker to employ more complex attacks, such as higher order DPA that requires a larger number of measurements.

- Protocol level countermeasures: These approaches involve designing the cryptographic protocols to preserve security even if some information leaks from each cryptographic operation. Secrets are continually refreshed and updated during cryptographic operations, so that an attacker is never able to get sufficient information to solve for any particular value. Variants of these constructions are applicable to both on-line applications (such as challenge-response authentication to a server) and off-line applications (such as firmware loading), and can accommodate both interactions with trusted servers as well as fully peer-to-peer protocols. While these methods cannot be used with legacy protocols lacking integrated protocol-level protections, designers who have the flexibility in the protocols can use these methods to achieve the highest level of security against power analysis attacks.

Because DPA attacks use signal processing to amplify leaked information, systems generally benefit from using multiple countermeasures. As a result, designers need to consider which approaches to use, given both their application's security requirements and engineering constraints. The flexibility of FPGAs permits designers to iteratively refine and test their implementations till the desired level of DPA-resistance is achieved.

Requirements and Standards

There are several security requirements and standards that are applicable to FPGAs when used in systems processing sensitive information. Current DoD Anti-Tamper (AT) requirements such as DoD Instruction 5200.39 require the protection of critical program information, which could include the data being processed by an FPGA as well as the design itself. The FIPS 140-3 standard (in draft form as of this whitepaper) specifies requirements for cryptographic devices used by the U.S. government for sensitive applications, and also includes requirements for resistance to power analysis attacks. DPA resistance has been a requirement under major Common Criteria (CC) protection profiles. In addition, commercial customers also define additional requirements relevant to FPGAs for applications such as those where there are concerns about IP theft, product counterfeiting, and exposure of cryptographic keys. As a result, FPGAs used for a range of government and sensitive commercial applications need resistance to power analysis attacks.

Many testing labs and DPA-testing products are equipped to perform DPA-testing on FPGA platforms. For example, the DPA Workstation™ from Cryptography Research is fully integrated with the SASEBO-GII Platform and has been used to evaluate a broad range of FPGA designs. In addition, many independent security evaluation labs provide testing for power analysis vulnerabilities. A first step for vendors is to use these testing resources to obtain a baseline assessment of information leakage from their cryptographic implementations on FPGAs.

DPA Countermeasure Licensing

Cryptography Research discovered SPA and DPA and has been awarded patents on the countermeasures required to protect products against these attacks. The company owns and actively licenses more than sixty issued U.S and international patents covering the fundamental countermeasures for DPA attacks. A patent license from CRI is required to make, use, sell, offer to sell, or import products utilizing DPA countermeasures. Over 4.5 billion security products are made each year under CRI's DPA countermeasure licensing program. Leading manufacturers with licenses include Actel, Atmel, EM Microelectronics, Infineon, Inside Contactless, NXP, Renesas, Samsung, ST Microelectronics, and Toshiba.

To help meet the needs of FPGA customers, Actel has obtained a license from Cryptography Research, and is offering FPGAs that already include a full license under the Cryptography Research DPA patent portfolio. By selecting these products, customers' licensing needs are addressed for those Actel chips. For more information on DPA mitigation techniques in Actel FPGAs visit www.actel.com/products/solutions/security/analysis.aspx. Actel is also working to add power analysis countermeasures to FPGA platform features, including bitstream decryption.

For products using FPGAs that are not fully pre-licensed by the FPGA maker, customers need to contact Cryptography Research for a separate license.

References

[1] Paul Kocher, Joshua Jaffe, Benjamin Jun, "Differential Power Analysis," *Advances in Cryptology - Crypto 99 Proceedings, Lecture Notes In Computer Science Vol. 1666*, M. Wiener, (Ed.), Springer-Verlag, 1999, pp. 388–397. (Whitepaper available at http://www.cryptography.com/public/pdf/DPA.pdf)

[2] Sıddıka Berna Ors, Elisabeth Oswald and Bart Preneel , "Power-Analysis Attacks on an FPGA – First Experimental Results" , *Cryptographic Hardware and Embedded Systems – CHES 2003, Proceedings, Lecture Notes in Computer Science, Vol 2779,* Colin D. Walter, Çetin Kaya Koç, Christof Paar (Eds.), Springer 2003, pp. 35-50.

[3] Francois-Xavier Standaert, Sıddıka Berna Ors, Bart Preneel, "Power Analysis of an FPGA - Implementation of Rijndael: Is Pipelining a DPA Countermeasure", *Cryptographic Hardware and Embedded Systems – CHES 2004, Proceedings, Lecture Notes in Computer Science, Vol 3156,* Marc Joye, Jean-Jacques Quisquater eds, Springer 2004, pp. 30-44.

[4] François-Xavier Standaert, Siddika Berna Örs, Jean-Jacques Quisquater, Bart Preneel, "Power Analysis Attacks Against FPGA Implementations of the DES", *Field Programmable Logic and Application, Proceedings – FPL 2004*, *Proceedings, Lecture Notes in Computer Science, Vol 3203,* Jürgen Becker, Marco Platzner, Serge Vernalde (Eds.): *Springer 2004,* pp 84-94

[5] Side-channel Attack Standard Evaluation Board (SASEBO), http://www.rcis.aist.go.jp/special/SASEBO/index-en.html

[6] Dylan McGrath, "Gartner: ASIC design starts to fall by 22% in '09", EE Times, http://www.eetimes.com/news/semi/showArticle.jhtml?articleID=216401584

[7] Jessica Davis, "FPGAs storm military spending", EDN (Electronic Design, Strategy, News), http://www.edn.com/article/CA6670749.html

[8] Steve Trimberger, Trusted design in FPGAs, In *Proceedings of the 44th Annual Design Automation Conference* (DAC '07). ACM, New York, NY, pp 5-8.

Contact Information

For more information about this whitepaper contact:

Pankaj Rohatgi, Technical Director, Hardware Security Solutions, Cryptography Research, Inc.