Side-channel Power Resistance for Encryption Algorithms using Dynamic Partial Reconfiguration (SPREAD)

N. Bete, F. Saqib*, C. Patel+, R. Robucci+ and J. Plusquellic
University of New Mexico, University of North Carolina, Charlotte*, University of Maryland, Baltimore County+

This hardware demonstration investigates countermeasures to side-channel-based attack mechanisms. In particular, a dynamic partial reconfiguration (DPR) method is proposed for FPGAs to make techniques such as differential power analysis (DPA) difficult and/or ineffective. We call the technique SPREAD, for Side-channel Power Resistance for Encryption Algorithms using DPR. SPREAD is designed to introduce diversity, and uncertainty, in the analysis of power supply transient signals. The proposed technique involves frequently changing the implementation characteristics of components of the Advanced Encryption Standard (AES) algorithm (while preserving the functionality) using DPR methods. Replicated primitives within AES, in particular, the SBOX, are synthesized to multiple implementations. During encryption/decryption, SBOX components are randomly selected and replaced dynamically with one of these implementations. The implementations are stored within FPGA Block RAM resources and a state machine coordinates with AES to carry out periodic DPR. The diversity of the implementations changes their delay characteristics and removes correlations in the power traces, making it difficult to identify the correct key.

1 Introduction

Security and trust have become critically important for a wide range of existing and emerging microelectronic systems including those embedded in aerospace and defense, industrial ICS and SCADA environments, automotive and autonomous vehicles, data centers, communications and medical healthcare devices [1]. The vulnerability of these systems is increasing with the proliferation of internet-enabled connectivity and unsupervised in-field deployment.

Authentication and encryption are heavily used for ensuring data integrity and privacy of communications between communicating devices. The security of the system depends on the key being securely stored and remaining private within the chip when encryption and decryption is taking place. Unfortunately, these assumptions are no longer valid, and in fact, adversaries can apply invasive and semi-invasive techniques, generally referred to as side-channel techniques, to extract information from chips that was traditionally considered private [2]. A wide variety of techniques have emerged that measure analog signals as a means of extracting internal secrets from the chip. The term *side-channel* refers to techniques developed for this purpose, and include methods that analyze leakage current, dynamic power (transient currents) and electromagnetic emissions. Used alone or in combination with fault injection techniques, where adversaries purposefully introduce clock and power glitches, such techniques can allow adversaries to steal secret keys and other private information in hours or days, effectively defeating the algorithmic protections engineered into the security algorithms.

This hardware demonstration investigates countermeasures to side-channel-based attack mechanisms. In particular, we focus on developing methods that are designed to make differential and correlation power analysis, referred to as DPA [3] and CPA [4], ineffective as an attack vector. DPA and CPA are particular problematic because 1) they enable high resolution visibility into the gate-level switching behavior of the chip, 2) they are semi-invasive and non-destructive, requiring only bench-top test and measurement equipment (that are widely available and decreasing in cost), and 3) with additional time and processing, they have been shown to be successful even when circuit level countermeasures are employed.

Our research is focused on leveraging the dynamic partial reconfiguration capabilities available in modern FPGA-based system-on-chip hardware platforms. Reconfigurable hardware is increasingly being integrated into microprocessor environments and therefore, the opportunity to leverage DPR is expanding. The proposed technique involves rapidly changing the implementation characteristics of components of encryption algorithm (while preserving the functionality) using DPR methods as a means of reducing correlations that are leveraged to deduce the top key byte candidates in cryptographic algorithms.

DPA derives its power by averaging power traces measured from an underlying **invariant** circuit implementation. We propose to change small components of the circuit implementation, e.g., one or more of the SBOX instantiations of the AES algorithm, randomly and rapidly while encryption/decryption is being carried out using DPR [5]. A set of different instantiations are stored within FPGA Block RAM resources (or secure processor side memory) and a DPR Controller state machine running in parallel with the cryptographic primitive synchronizes with AES to enable periodic reconfiguration.

Several strategies are investigated for creating the different instantiations, including methods which add ‘wire stubs’ to a fixed implementation as a means of changing its load capacitance and corresponding delay and power trace behavior. Implementation diversity can also be introduced by making small inconsequential changes to the behavioral description and/or timing constraints of, e.g., an SBOX component, and then using the FPGA synthesis tools to add diversity automatically. The goal is to create a set of instantiations which produce different power traces and then to swap the different instantiations in and out during cryptographic operations. DPA/CPA techniques, oblivious to the swapping, would then average power traces from different instantiations. The mixed set of power traces create random artifacts in the averaged traces, reducing or eliminating correlations that allow the target key byte to be properly deduced. The SPREAD state machine manages the swapping process, and is designed to minimize stalls to the encryption engine which is running in parallel with SPREAD.

In addition to investigating the diversity of implementations as a countermeasure to DPA/CPA, we also conduct a

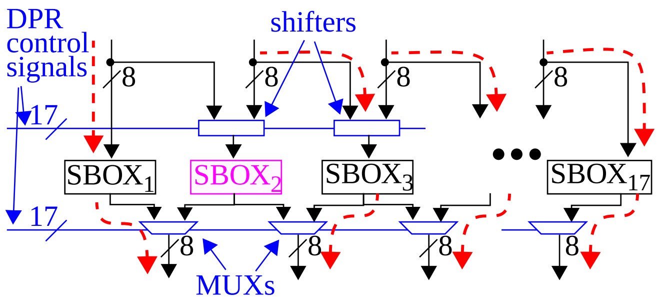

Fig. 1. Implementation strategy that allows any of the SBOXs to be reconfigured by creating a 'hole' while allowing the entire AES engine to continue encypting or decrypting.

power analysis on the power signature that is generated by the DPR operation itself. It is important that the adversary is not able to track the SBOX configurations over time as swapping operations are carried out otherwise he/she may be able to separate the traces corresponding to one particular configuration in the attack. We show that the DPR traces are very similar across two different implementations, as expected since the partial bitstream used to re-program the SBOXs are identical in size and only a subset of the configuration bits are different.

Moreover, we show the number of possible configurations is exponential and introduce a nonce-driven (random) timing interval between swapping operations, which, when taken together, make SBOX tracking by the adversary unlikely to be successful.

2 SPREAD Design

The SPREAD Controller is a VHDL module that coordinates the DPR operations with a fully operational encryption engine, e.g., AES. SPREAD performs self-reconfiguration using Xilinx's internal configuration access port (ICAP) interface. Self-reconfiguration refers to techniques that run in the programmable logic (**PL**) that reconfigure other components in the PL, excluding itself.

The time taken to perform DPR using the ICAP interface is approx. 1 ms for smaller *pblocks* (pblocks are partial dynamic reconfigurable regions). Therefore, stopping cryptographic operations to carry out DPR would introduce a significant performance penalty on the encryption or decryption operations. To address this issue, we implement a *single-unit redundancy scheme* as shown in Fig. 1 using AES as the encryption engine. Each of the SBOX regions is reconfigurable. A set of 16 parallel SBOXs are needed in the 128-bit version of AES (the figure shows only a subset of them).

Our proposed scheme adds one additional parallel SBOX. The *DPR control signals* from SPREAD are used to create a 'hole' in the parallel configuration of the 17 SBOXs, by using shifters and MUXs to wire around the SBOX that is the target for reconfiguration. Annotation in Fig. 1 shows the routing configuration when SBOX$_2$ is the target. Since DPR can take place while the rest of the system continues to operate at full speed, encryption/decryption can continue with only 1 stall cycle to reconfigure the shifters and MUXs.

2.1 System Diagram

A block diagram of the proposed system that is applicable to FPGA SoC architectures is shown in Fig. 2. Security features that exist on the Processor Side of the SoC, such as Xilinx TrustZone, can be leveraged to ensure the partial bitstreams are loaded into BRAM using a secure general-purpose I/O (GPIO) interface. Once loaded, the operations carried out by the DPR Controller during encryption or decryption are as follows:

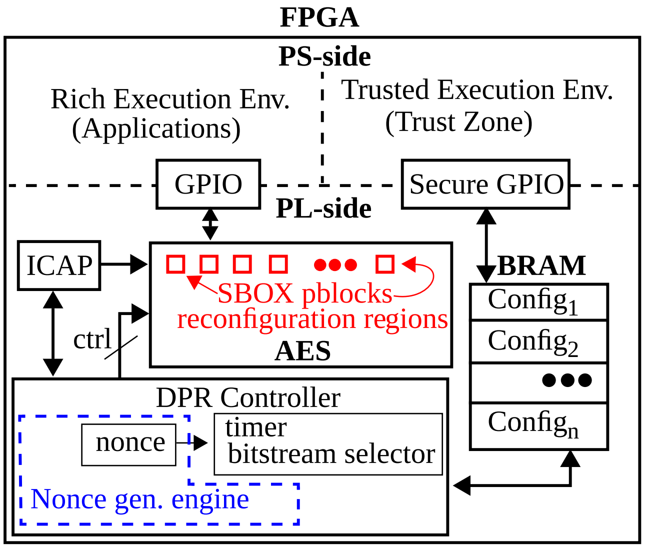

Fig. 2. Block diagram of the DPR Controller on a SoC FPGA.

- Start the nonce generation engine. The nonces are used to 1) randomize the time intervals between DRP operations, 2) select from among the configurations that have been loaded into the BRAM and 3) select the target reconfiguable regions within the cryptographic engine.
- Read the selected bitstream from BRAM, assert the appropriate control signals for reconfiguration of the selected cryptographic component, synchronize with the cryptographic engine to insert a stall cycle(s) as needed and execute the transfer protocol using the ICAP controller.

3 Observables in Hardware Demonstration

A Sakura-X will be used as the test platform in the hardware demonstration. A controller state machine, implemented in VHDL, will randomly select and re-program portions of a fully operational and functioning AES engine once every millisecond using the TRNG data of a physical unclonable function. The partial programming bitstrings will be loaded into PL-side BRAM from a trusted execution environment (TEE) running under Linux using Xilinx TrustZone. An oscilloscope will be used to measure power transient waveforms and a software version of the DPA algorithm will be used to average and correlate them to determine if peaks normally present without the countermeasure reduce or disappear when the countermeasure is enabled.

4 References

[1] K. M. Goertzel, "Integrated Circuit Security Threats and Hardware Assurance Countermeasures", Real-Time Information Assurance", *CrossTalk*, Nov./Dec. 2013.

[2] P. C. Kocher, "Timing Attacks on Implementations of Diffe-Hellmann, RSA, DSS, and Other Systems", *CRYPTO* 1996, LNCS 1109, 1996, pp. 104-113.

[3] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis" *Advances in Cryptology*, 1999, pp. 388-397.

[4] E. Brier, C. Clavier, F. Olivier, "Correlation Power Analysis with a Leakage Model", *CHES*, Vol. 3156, 2004, pp. 16-29.

[5] N. G. Bete, M. Nakka, J. Plusquellic, F. Saqib, C. Patel and R. Robucci, "Implementation Diversity and Dynamic Partial Reconfiguration for Impeding Differential Power Analysis Attacks on FPGA", shttp://www.hostsymposium.org/host2017/hardware-demo-list.php, HOST, 2017.