Power Analysis of the t-Private Logic Style for FPGAs

Zachary N. Goddard, Nicholas LaJeunesse, and Thomas Eisenbarth
Worcester Polytechnic Institute, Worcester, MA 01609, USA
Email: {zngoddard,nplajeunesse,teisenbarth}@wpi.edu

***Abstract*—The goal of t-private circuits is to protect information processed by the circuit. This work presents the first practical power analysis evaluation of t-private logic style for FPGAs. Following the synthesis technique introduced at HOST 2012, a t-private S-box of the Present block cipher is synthesized and analyzed with respect to side channel leakage. The analysis is performed on simulated power traces as well as real power measurements taken from an implementation on a Virtex 5 FPGA. Classical Correlation power analysis and Correlation enhanced collision analysis are applied to detect first order leakages. Our results reveal a remaining first-order side channel attack vulnerability.**

I. Motivation

Implementation attacks are a common threat to cryptographic cores in hardware. In many embedded applications, the attacker has physical access to the chip containing the crypto core. That access can be abused to recover critical secrets, namely cryptographic keys. Common attack techniques include invasive attacks such as probing as well as non-invasive attacks such as power, EM and timing attacks. Power and EM side channel attacks are particularly interesting, because, unlike probing attacks, the equipment and hardware specific knowledge necessary to perform them is minimal. At the same time, power and EM attacks are extremely difficult to completely prevent. One of the common and well studied techniques to hinder power analysis is masking. Masking splits internal states of logic into two or more random shares so that secret information can only be extracted from their combination. It has been found that masking can be overcome by higher order side channel analysis. However, if implemented correctly, first order (or in general lower order) side channel attacks are prevented, while higher order attacks are known to need much higher number of traces to succeed, yielding a practical security for many applications. Nevertheless, the price is high in terms of the introduced area overhead and the care that is needed to implement the masking countermeasure correctly.

t-private circuits are a countermeasure that have originally been proposed to counter probing attacks. It is fairly similar to the above-described approaches, as it is also based on the principle of secret sharing. In [1] the authors show a methodology where t-private circuits can be efficiently mapped to FPGAs. While side channel resistance is not explicitly claimed in [1], the work suggests that t-private circuits protect against a *stronger adversary*, i.e. the probing adversary that can observe internal states directly, instead of an aggregate leakage through the common power supply. The application of the t-private logic as a Differential Power Analysis (DPA) countermeasure was studied in more detail in [2]. That work describes their method of finding circuits with a low DPA robustness and replacing that logic with a t-private countermeasure to mitigate these problems. The paper, however, does not describe the overall DPA resistance of t-private logic.

In this work we show that the achieved protection against probing is not inclusive, i.e. while probing attacks on certain lines might be prevented, an aggregate first-order leakage is still sufficient to recover secret internal states in a DPA scenario. In this respect, we present the first practical side channel analysis of the t-private logic style for FPGAs as presented at HOST 2012. For this we implement the S-box of the Present block cipher as a 1-private circuit on a Xilinx Virtex 5 FPGA. We analyze the resistance of the implementation to first-order attacks, as countermeasures with two shares can achieve first order side channel resistance. Similarly 1-private logic only promises resistance to one internal signal being probed. We follow the evaluation methodology established by works such as [3], [4], [5], which use Correlation Power Analysis (CPA) [6] and the Correlation Enhanced Collision Analysis (CCA) [7] to analyze their designs for first-order side channel leakage.

II. The t-private Countermeasure

Private circuits were proposed in [8] in 2003 as a countermeasure to invasive attacks. This work states that probing attacks are a more powerful adversary than DPA because the attacker 'can read your brain' by directly observing logic values of internal nets. t-private circuits are a masking countermeasure that splits the values into $t+1$ shares. The value of t determines the security level of the circuit, i.e. the number of internal probes per clock cycle the circuit is resistant to. A circuit designed with $t = 1$ will be resistant to 1 probe per clock cycle. For $t = 1$ the inputs to a circuit are split into two shares, the original input is XORed with a mask value and the mask value itself. The original work [8] described how to make t-private encoders, decoders, PRNGs, AND gates, OR gates, XOR gates and inverters. The protection is created by encoding the true value of the input into t shares as described in Ishai et al. [8]. The construction of basic t-private gates has been described in [1]. These gates have been optimized for size from the original t-private gates and the new gates are described in [8]. The gates operate on both encoded nets

to create two outputs, that when XORed together return the true output value. Each output is dependent upon both encoded nets so a single gate value will not deterministically change the output of a gate. To apply this to the Present S-box a standard CMOS gate level design was created. This CMOS design was changed into a t-private design by replacing standard gates with their t-private equivalents. To semi-automatically implement the countermeasure, a synthesized Verilog netlist was translated to t-private RTL using a library of $t = 1$ private gate modules and a script. The translation is performed by a Python script that creates encoded equivalents of the original nets by replacing standard CMOS gates with their t-private counterparts. The design was then synthesized to maintain hierarchy so the gates would not get combined in LUTs.

The use of random mask bits is crucial to t-private circuits. Each input must be masked and there are internal masks needed for AND, OR, NOR, and NAND gates. The work described in [1] explains their use of EXor Sum Of Products (ESOP) circuits to save space in their hardware design. The application to Present does not use the ESOP form of the cipher. While the ESOP form only uses AND and XOR gates our design does not use this form. Instead, we explore the application of AND, OR, XOR and inverter gates described in the previous work [1].

III. Reference Implementation

Present is a lightweight block cipher that implements a very compact substitution permutation network. One of the design goals of the Present S-box was low area in hardware, which makes it a good and simple target for reference implementations. To create the Present S-box design t-private logic synthesis was used. This incorporates the creation of a synthesized netlist using a synthesis tool (Synopsys Design Compiler) and using their technology independent synthesis to create a netlist containing CMOS primitives that are easily parseable. The standard combinational logic designs can be translated to t-private logic by replacing the original CMOS gate with the t-private gates. The nets originally contained into the designs were then duplicated for our $t = 1$ design. The nets were connected to the respective inputs and outputs of the t-private gates. The Python tool creates a new design module that is then put through the rest of the standard design flow for FPGAs. The unprotected Present S-box has four input and four output bits. Thus for the $t = 1$ design, four independent input mask bits are generated also which gives 8 input bits and 8 output bits. The input nets are encoded so that one share contains the original value xored with the mask bit and the other share contains the mask bit. In addition, one internal mask bit for all and AND and OR gates is contained in the design. This RTL is synthesized using ISE Design Suite 14.6. The target platform is a Virtex-5 (XCV5VLX50-2ff324) on a SASEBO GII. Post synthesis reports for the t-private implementation show that the design synthesizes and maps to 96 Slice LUTs. While this design could be further optimized for size, it is a correct implementation of the t-private Present S-box.

a) Device Under Test Setup: The design has been developed so that encoded plaintexts and masks can be sent to the FPGA to be encrypted. The hardware design was replicated eight times to amplify the power consumption and thereby simplify the measurement and analysis of the recorded power traces. This step was necessary because the used oscilloscope did not have a sufficiently high resolution to observe the power consumption of a single S-box. Therefore there were 8 Present S-boxes placed in parallel to be able to distinguish the power consumption in the measurement setup. The control interface for the SPARTAN 3 is a reuse from the model created by AIST. The wrapper around an AES core was replaced to wrap around the 8 Present S-boxes. To ensure a clean leakage analysis, there is a register bank that stores plaintexts after they are written to the board by a Python interface. At the rising clock edge the values in the register bank get assigned to the inputs of the combinatorial logic cloud and a trigger signal asserts a high value. The output values from the combinatorial logic are continuously assigned to a separate register bank. Both inputs and masks are provided to the implementation from the test setup. This means that masks are externally generated to prevent dependent leakages from the mask generation.

IV. Power Analysis

In this section we describe the performed analysis of the t-private reference implementation. Two popular analysis methods are applied: a classical CPA and the correlation-enhanced collision attack.

1) CPA: CPA is one of the most popular and simple forms of Differential Power Analysis. It correlates the observed leakage to a hypothetical key dependent leakage. The correlation coefficient is a value used to distinguish or detect the correct key. Our leakage model is the Hamming distance of two consecutive S-box outputs, which is an approximation of the leakage of the output register of the S-box. This is because the Hamming distance is representative of the number of toggles that occur between the previous state to the current state. Toggles or switching, can be directly related to the power consumption of the circuit, thus making the Hamming distance an appropriate model for leakage present during the encryption.

2) Correlation-enhanced Collision Attack: The second analysis method used to evaluate our data the correlation enhanced collision attack (CCA). CCA has two interesting properties: it detects only first order leakages and it does not require a leakage model, making it a good leakage detection mechanism [4], [5]. In this attack, traces from a collection are separated into two sets of equal size. Traces from each set are then sorted by the S-box input (or output, as it is a one-to-one correspondence) value and averaged to create two sets of 16 average traces, one per set per value. Next, the averages of the two sets are compared using Pearson's correlation coefficient. To test whether the leakage is distinguishable, a key offset was simulated (by simply reordering the averages for offsets 0-15, where an offset of zero is the correct key guess). If the 'correct key', i.e. zero offset is distinguishable, this implies

that the implementation has a first order leakage that can be exploited as described in [7]. If there is no data-dependent leakage, then this test fails.

Both attacks are applied on a simulated leakage based on the toggle count model of the FPGA synthesis output as well as measurements of the reference design on the SASEBO-GII.

A. Simulated Power Leakage

The leakage simulation is based upon a toggle model. The 8 Present S-boxes were simulated after place and route using ISIM. The simulation is exhaustive for all input state transitions with randomly generated masks. During the simulation a VCD file is created that records the binary values for nets within the design. A Python script is used to parse this VCD file and collect the number of toggles over time. The toggle values are then filtered with a low pass filter with a cutoff of 150 Mhz and the simulated power traces are created.

B. Real Measurement Setup

The measurement setup used the Sasebo-GII board with an on-board Virtex-5 FPGA(XC5VLX50) as the encryption core operating at 3 MHz. A Tektronix MDO4104B-6 Oscilloscope measured the power signal via SMA-BNC cable connection to the 1Ω sense resistor on the Sasebo-GII. Measurements were taken at a sample rate of 5 GS/s with 8 bits of resolution.

Each design, one with constant masks and one with varying masks had a capture that consisted of 259,200 consecutive encryptions. A test vector of plaintext inputs was generated such that all possible input transitions are iterated over. This complete cycle was then repeated multiple times during the capture to gather the total sample size of 259,200 traces. During each encryption process, the design generates a trigger signal when the data is ready to be read in to the S-box, ensuring that every trace is properly aligned.

The simulated data was collected for the same state transitions. Power traces were generated using the filtered time varying toggles, as described in Section IV-A.

C. Evaluation Results

Using the results from the attacks on simulation and hardware implementation data the effectiveness of 1-private circuits against several first order attacks was determined. Evaluation was performed for two scenarios: In the first scenario the input mask set to zero. In the second scenario the input mask is chosen uniformly at random. As done in [5], we use *externally applied masking* to ensure that no leakage is due to input and output logic. The zero-mask case corresponds to an unprotected implementation and is used to analyze the leakage behavior of the circuit as well as to show that the measurement setup is able to properly capture leakages. The second scenario evaluates the effectiveness of the countermeasure to power analysis.

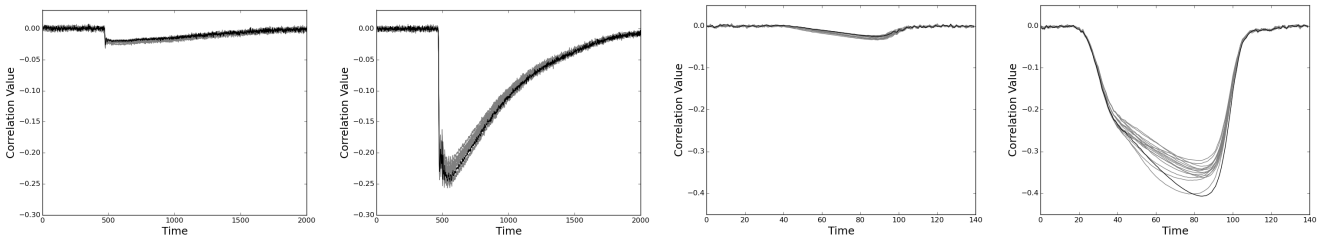
The CPA was performed on all 259,000 traces for each case. Fig. 1(a) and 1(b), depict CPA results for the regular 1-private and the zero-mask 1-private Present S-box respectively. Fig. 1(c) and 1(d) show the same results on the simulated leakage. The t-private and zero mask t-private scenarios show resistance to the classical CPA as the correct key remains hidden among the incorrect hypotheses. However it is important to note that while still hidden, the zero mask CPA results have much higher correlation values than their masked counterparts.

The same analysis was performed on the simulation data. First shown are CPA results for both designs: t-private Present with all-zero mask bits and t-private with 5 random mask bits per clock cycle. The solid black line represents correct key guesses in all plots below. The leakage model for the CPA was the Hamming distance between S-box outputs. In the attacks performed on the t-private designs the decoded 4 bit output values for the S-box were used for previous and current ciphertext and 259,200 power traces were collected. In the CPA for the t-private fully protected design the correlation was not able to escape the noise floor, i.e. the attack would fail. This design was protected against this leakage model for 259,200 traces and lowers the peak correlation by about a factor of 10 compared to the zero mask values. We assumed a Hamming distance leakage for two consecutive S-box outputs for the CPA. This model is incorrect, as a 0-value input mask still results in a masked output, i.e. even if the input mask is zero, the output will be shared.

Next, the CCA was performed across 259,000 traces for each scenario. The number of traces was determined to allow for all input states and transitions to be accounted for within the test. Figure 2 depicts the correlation value of all 16 key hypotheses at a single point on the power trace versus the number of traces analyzed. Each figure is scaled on the x axis by a logarithmic scale representing the number of traces and the y axis is showing the correlation value for a point determined in the attack results. The graphs show where the values escape the noise floor. Fig. 2(a) and 2(b) show the CCA analysis on real measurements for random and all-0 fixed masks. The correct hypothesis shows clear distinguishability above all other guesses from about 130,000 traces in the masked case and 10,000 in the zero mask case. The CCA results for simulation are depicted in Fig. 2(c) and 2(d). We can see that leakage begins to become evident at about 1,000 traces for the zero mask case and 5,000 traces for the unmasked case.

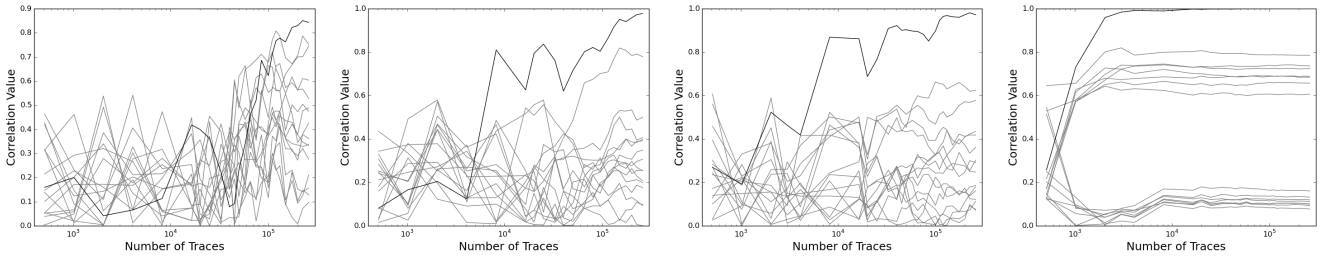
D. Differences between Simulation and Hardware Capture Attack Results

As shown in previous sections there are similarities and differences between the simulation and hardware attack results. The real measurements are the output of a 8 bit A/D converter that samples at best at 1GS/s, while simulation based on the filtered toggle values achieves a much higher resolution in both dimensions. Furthermore, the amount of Gaussian noise added to the simulated design was a small amount to allow for the variance to be enough for the classical attacks to function without zero variance errors. The simulation data is deterministic for a certain input state transition. Each net in the design is also contributes equally to simulated power consumption. This is not always true for hardware because capacitive loads differ between nets.



(a) CPA on power measurements (b) CPA on power with all-0 mask (c) CPA on simulated measurements (d) CPA on sim. with all-0 mask

Fig. 1. CPA results for the correct (black) and wrong (grey) key offset on the t-private S-box using the measured power leakage (a), (b), and simulated leakage (c), (d). The mask is all-0 for (b) and (d), hence there should be a first-order leakage. For (a) and (c) the mask is random.



(a) CCA on power measurements (b) CCA on power with all-0 mask (c) CCA on simulated leakage (d) CCA on sim. with all-0 mask

Fig. 2. Correlation coefficient for the correct (black) and wrong (grey) key offset for the CCA attack on the t-private S-box using the measured power leakage (a), measured power leakage when the mask is fixed to all-0 (b), simulated leakage (c), simulated leakage when the mask is fixed to all-0 (d). The correct hypothesis becomes obvious in all scenarios, though more than 100,000 measurements are needed for properly masked real power measurements.

The reason that these facts contribute to the magnitude of traces being lower for the CCA is that the CCA is based upon averages. Averaging the traces for a certain value aims to profile the trace and remove Gaussian noise. While the hardware data has a high amount of noise, the simulation data is very regular and has a low amount of noise. The averages also vary by a small amount which causes the precision to come into play. All of these contribute to the hardware collection data to require a higher number of traces to extract the key guess opposed to the simulation data.

V. Conclusion

This work presents the first practical side channel analysis of t-private circuits on FPGAs. In [8] t-private for $t = 1$ is said to be resistant to 1 probe per clock cycle. This is essentially stating that it is resistant to first-order DPA, since DPA is probing a single wire, the power wire outside the chip. However, our results show leakage when analyzed using the correlation enhanced collision attack after only 130,000 traces for the full t-private S-box implementation. The result was also verified in simulation by applying a toggle count analysis.

The analysis results are particularly interesting because the CPA was unsuccessful for our total number of traces collected at 259,000. This is an important observation showing that CPA is not an appropriate method for validating the effectiveness of a side channel countermeasure.

Acknowledgment

We would like to thank Joe Chapman from the MITRE Corporation for his support for this work. This material is based upon work supported by the National Science Foundation under Grant No. #1261399.

References

[1] J. Park and A. Tyagi, "t-Private Logic Synthesis on FPGAs," in *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*, pp. 63–68, June 2012.

[2] J. Park and A. Tyagi, "Towards making private circuits practical: Dpa resistant private circuits," in *VLSI (ISVLSI), 2014 IEEE Computer Society Annual Symposium on*, pp. 528–533, July 2014.

[3] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "A More Efficient AES Threshold Implementation," in *Progress in Cryptology – AFRICACRYPT 2014* (D. Pointcheval and D. Vergnaud, eds.), vol. 8469 of *Springer LNCS*, pp. 267–284, 2014.

[4] A. Moradi and O. Mischke, "How Far Should Theory Be from Practice?," in *Cryptographic Hardware and Embedded Systems – CHES 2012* (E. Prouff and P. Schaumont, eds.), vol. 7428 of *Springer LNCS*, pp. 92–106, 2012.

[5] A. J. Leiserson, M. E. Marson, and M. A. Wachs, "Gate-Level Masking under a Path-Based Leakage Metric," in *Cryptographic Hardware and Embedded Systems – CHES 2014* (L. Batina and M. Robshaw, eds.), vol. 8731 of *Springer LNCS*, pp. 580–597, Springer Berlin Heidelberg, 2014.

[6] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *Cryptographic Hardware and Embedded Systems — CHES 2004* (M. Joye and J.-J. Quisquater, eds.), vol. 3156 of *Lecture Notes in Computer Science*, pp. 135–152, Springer Berlin / Heidelberg, 2004.

[7] A. Moradi, O. Mischke, and T. Eisenbarth, "Correlation-Enhanced Power Analysis Collision Attack," in *Cryptographic Hardware and Embedded Systems – CHES 2010* (S. Mangard and F.-X. Standaert, eds.), vol. 6225 of *Springer LNCS*, pp. 125–139, 2010.

[8] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in *Advances in Cryptology - CRYPTO 2003*, no. 2729 in Springer LNCS, 2003.