Detecting Delay Anomalies Introduced by Hardware Trojans using Chip-Averaging and an On-Chip High Resolution Embedded Test Structure

D. Ismari, C. Lamech*, Swarup Bhunia~, F. Saqib+, and J. Plusquellic
ECE Dept. University of New Mexico, *Intel Corp., +Florida Institute of Technology, ~University of Florida

ABSTRACT

A hardware Trojan (HT) detection method is presented that is based on measuring and detecting small systematic changes in path delays introduced by capacitive loading effects or series inserted gates of HTs. The path delays are measured using a high resolution on-chip embedded test structure called a time-to-digital converter (TDC) that provides approx. 25 ps of timing resolution. A calibration method for the TDC as well as a chip-averaging technique are demonstrated to nearly eliminate chip-to-chip and within-die process variation effects on the measured path delays across chips. Path delay tests are applied to multiple copies of a 90 nm custom ASIC chip which incorporates two copies of an AES macro. The AES macros are exact replicas except for the insertion of several additional gates in the second copy, which are designed to model HTs. Statistical detection methods are used to isolate and detect systematic changes introduced by these additional gates and a set of Trojan emulation circuits also inserted into the macros.

1. INTRODUCTION

Hardware trust has emerged as a major concern for government and industry personnel, as made evident from a wide variety of issues raised at recent technical meetings [1][2]. Unlike hardware security which provides a 'value add' to products, hardware trust is something that customers expect, similar to the expectations they have regarding manufacturing defects. Unfortunately, providing a high assurance, trusted product is much more difficult than providing high quality, defect-free chips. This is true because the random nature of manufacturing defects makes it possible to find nearly all of them with test vectors that provide high levels of fault coverage. Hardware Trojans (HTs), on the other hand, are designed and inserted by intelligent adversaries with the deliberate intention of making them nearly impossible to activate with arbitrary test vectors, akin to the difficulty of guessing a 256 AES key using only information provided by a chosen message analysis.

However, there are alternative techniques for detecting HTs that do not require activation. Drawing on the analogy with AES, a technique called differential power analysis greatly simplified the task of extracting the AES key. Similarly, parametric testing methods provide a similar advantage for detecting HTs by carrying out a structural analysis of the IC, as opposed to a functional analysis. The advantage of a structural analysis over a functional analysis relates to the number of tests that need to be applied to attain sufficient coverage of HTs. Structural testing, as the name implies, focuses on testing each of the elements in the netlist or layout, and therefore, the number of tests is related linearly to the size of the circuit. In contract, functional analysis requires an exponential number of test vectors, which is not practical except for very small chips.

Any parameter of the IC, including dynamic current, leakage, delay, EMI, hot spots, etc. can be targeted by parametric methods. Delay-based parametric methods detect delay anomalies introduced by the capacitive loading of HT wires and by series inserted HT gates. In contrast to random defects, the anomalies introduced by HTs are systematic in nature, i.e., showing up in multiple copies of the ICs in a similar fashion, and can be identified by comparing the signal behavior of the chips with that of a golden (HT-free) simulation model. The challenge of implementing parametric testing methods is dealing with chip-to-chip and within-die process variations effects. Failing to properly account for the natural variations that occur in the power and performance characteristics of chips results in false negative decisions (a determination that the chip does not have an HT when it does) and false positive HT decisions (a decision that it has an HT when it does not).

In this paper, we investigate an HT detection method that analyzes a chip's delay characteristics. An on-chip measurement structure called a time-to-digital (TDC) is used to obtain high resolution (approx. 25 ps) measurements of path delay. Chip-to-chip and within-die process variation calibration methods are proposed as a means of improving resilience to false negative and false positive detection decisions. Experiments are carried out on 44 copies of a custom, 90 nm test chip which incorporate two instances of Advanced Encryption Standard (AES) macro. The contributions of this work are as follows:

- Application of our proposed detection methods to commercially synthesized implementations of actual functional units and chips.
- Demonstration of a high resolution, embedded time-to-digital converter (TDC) for measuring path delays.
- Calibration of chip-to-chip variations by tuning a control parameter of the TDC, and calibration of within-die variations using a chip-averaging technique. To the best of our knowledge, this is the first experimental, measurement-based method designed to isolate systematic anomalies in path delays in ASICs.
- Identification of systematic delay anomalies in two identical copies of a functional unit with fan-out and payload HTs inserted only in the second copy.
- Determination of the resolution limits of our proposed statistical detection methods.

The reminder of the paper describes the chip design, experimental setup and test chip data analysis, as well as the statistical outlier techniques that are used to detect the systematic delay anomalies introduced by HTs.

2. BACKGROUND

The insertion of an HT can impact the delay of paths in a circuit in two ways as shown in Fig. 1. A *fan-out* HT gate connects to

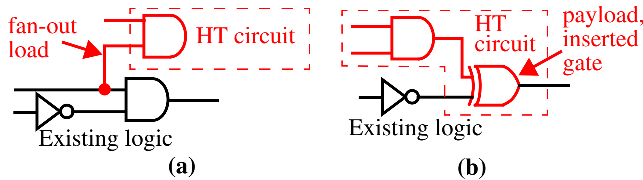

Fig. 1. (a) Fan-out HT gate and (b) payload HT gate.

existing wires in the circuit, that allow the HT to monitor circuit state for activation conditions. The connections add capacitive load to the original path, which in turn impacts delay. The second *payload* HT gate is inserted in series with the original path (see Fig. 1(b)), and is designed to maliciously modify circuit state when the HT activates. The inserted gate also adds delay to paths in the original circuit, and this delay is typically larger than the delay introduced by capacitive loading of fan-out HT gates.

The authors of [3] were the first to address the HT issue. They use transient power supply currents to identify HTs in chips. In [4], logical circuit analysis is carried out using automatic test pattern generation (ATPG) after identifying target sites where HT's are most likely to be inserted. In [5], the authors propose adding observability enhancing changes to the least controllable nodes of the circuit. In [6], the authors extend the state space of the circuit by introducing MUXes that can select the Q or Q_BAR outputs of the flip-flops. The authors of [7] use a dynamically distributed multi-processing element trust determination scheme that observes the outputs of the elements to determine trust.

For side-channel variation detection, authors propose methods to decrease noise to better measure minimal power draw [8]. Authors look at region-based stimulation and compare global power consumption [9]. Characterization of regions side-channel signature and observing outliers compared to a set of estimation results are proposed in [10]. Another outlier technique is proposed in [11] where comparisons between a golden model and power signals of a chip. The authors of [12] leverage a calibration technique to identify regional current and frequency fluctuations. Reference [13] takes the characterization to gate-level leakage power and delay measurements.

For delay variation detection techniques, [14] demonstrates a shadow register setup to measure finite delays with minimal test vectors. [15] describes an embedded test structure called REBEL for measuring path delays and a simple statistical method for detecting HTs. The authors of [16] discuss a method of on-chip sensors to measure delay and determine HT's without the need of a golden model. [17] uses fault-injection to create clock glitches with variable clock periods to determine a delay fingerprint. [18] presents a scalable method of circuit partitioning to measure the delay at gate-level. The authors of [19] look to the foundry approximated models to determine expected delay variations. [20] creates a path delay fingerprint of HT-free chips as a statistical model which can be used to detect HTs. In [21], the authors look at register-to-register delay of a large number of paths to determine if an HT has been inserted. [22] presents a method to use clock sweeping to obtain the path delay signatures. The authors of [23] propose a self-referencing technique that compares behaviors of identical functional units, thereby eliminating the need for a golden simulation model. This technique is expanded on in [24] by identifying path symmetries and comparing the delays across multiple instances of the path on the same chip. We utilize the functional

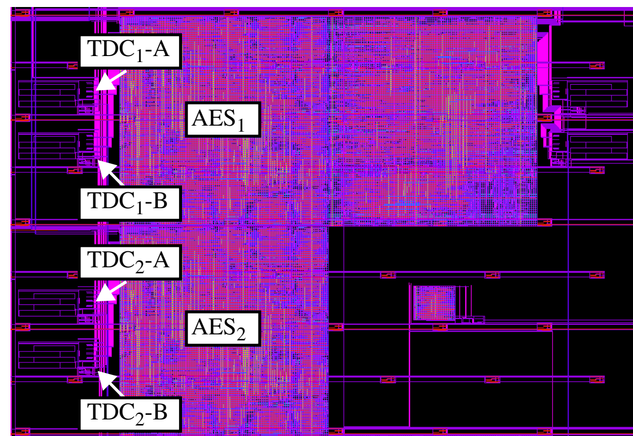

Fig. 2. Chip layout showing AES MUTs and TDCs.

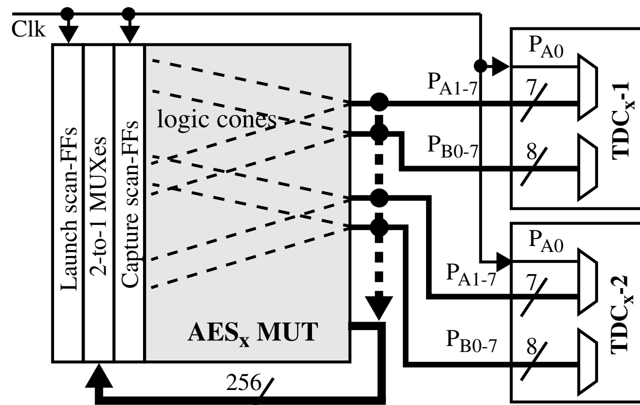

Fig. 3. Block diagram of TDC connections to an AES macro-under-test (MUT).

unit self-similarity method in this paper to circumvent the golden simulation model.

3. CHIP DESIGN

The test chip architecture consists of 2 macros-under-test (MUTs), labeled AES_1 and AES_2 for Advanced Encryption Standard, as shown in the layout of Fig. 2. Two copies of an embedded test structure called the time-to-digital converter or TDC (described below) are associated with each MUT [25]. The TDCs are capable of providing accurate delay measurements of signals propagating through the MUTs to its outputs.

A block diagram illustrating the connectivity between an AES MUT, labeled AES_x, and the TDCs labeled TDC_x-1 and TDC_x-2 is shown in Fig. 3. Each TDC has 16 inputs. The first input connects to the clock while the remaining 15 connect to outputs of the AES MUT. This allows signals propagating through logic cones to 30 of the AES outputs to be timed by one of the TDCs[1]. The Clk input serves as a means of characterizing the TDCs and as a reference path for delay testing as explained below. A special set of 'Launch scan-FFs' are added to the MUT to allow any arbitrary two vector test to be applied. This is accomplished by scanning the

1. Additional MUXes or a scheme similar to that described in [15] can be used to access the remaining outputs.

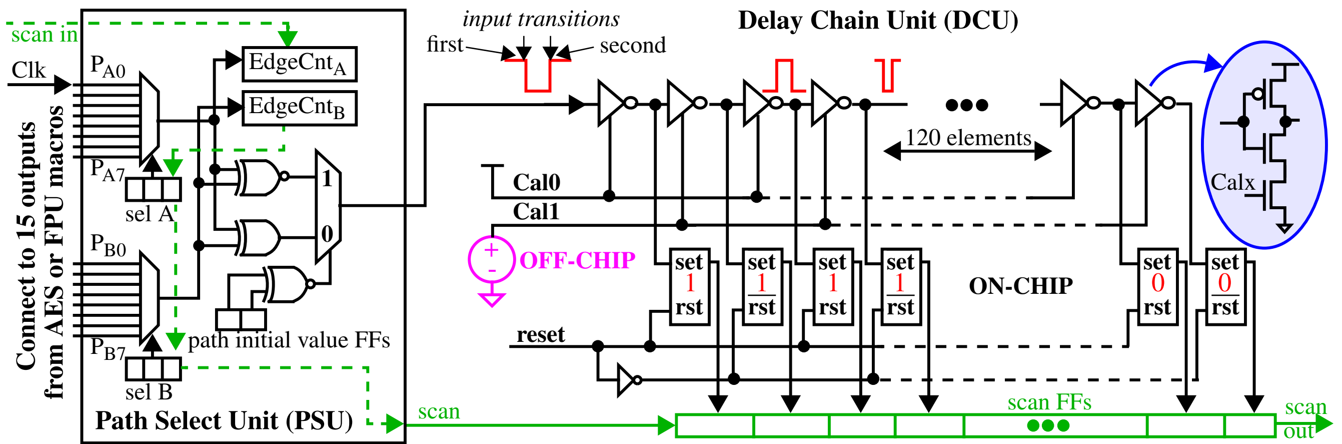

Fig. 4. Pulse Shrinking Time-To-Digital Converter (TDC).

first vector into the 'Capture scan-FFs' row and the second vector into the 'Launch scan-FFs' row. The Clk can then be used to launch transitions onto the inputs of the MUT.

3.1 Time-to-Digital Converter (TDC)

The TDC is designed to measure the relative delay between two output signals from the MUT or one MUT output signal and the Clk. The TDC is implemented as two components, labeled Path Select Unit (**PSU**) and Delay Chain Unit (**DCU**) in the schematic of Fig. 4. Scan FFs in the PSU, labeled 'Sel A' and 'Sel B', drive the inputs of two 8-to-1 MUXes, which, in turn, select a specific pairing of MUT outputs, one from the group labeled P_{Ax} and one from group labeled P_{Bx}.

Path delay tests are carried out by applying 2 vectors in sequence to the inputs of the MUTs. The output values from the two selected paths to be timed by the TDC after the 1st vector is applied are latched into the 'path initial value FFs' in Fig. 4. These control values select the output of either the XOR or XNOR gate to generate a negative pulse for the DCU (see annotation in Fig. 4). For example, if both path values are '0' or both are '1', then the XNOR gate is selected because its output under these conditions is '1'. The arrival of an edge on one of the MUT outputs propagates to the XOR or XNOR and generates the 1-to-0 transition of the negative pulse, and an edge (arriving later) on the second output generates the 0-to-1 transition of the pulse. The 'EdgeCnt$_{A/B}$' components count the number of transitions that occur on each of the paths as a means of determining whether any glitching occurred. Although some types of glitching can be tolerated by the TDC, the relative timing value produced by the TDC can be corrupted by glitching and therefore we use only path tests in which the EdgeCnts are '1' for both paths. Paths for which this is true are called **stable paths**.

The difference in the delays of the two paths is captured in the width of the negative pulse. The TDC is designed to 'pulse shrink' the negative pulse produced by the XOR/XNOR as it propagates down a current-starved inverter chain. As the pulse moves down the inverter chain, it activates a corresponding set of set-reset latches to record the passage of the pulse, where activation is defined as storing a '1'. A thermometer code (**TC**), i.e., a sequence of '1's followed by a sequence of '0's, represents the digitized delay difference between the transitions occurring on the paths. A longer sequence of '1's (up to 120 in our version) reflects a wider initial negative pulse width.

A call-out of a current-starved inverter used in the delay chain is shown on the far right side of Fig. 4. The NFET transistor with input labeled 'Calx' implements the current-starving mechanism. The Calx inputs are driven by two control voltages, labeled 'Cal0' and 'Cal1'. The current-starved inputs of all the even numbered inverters (numbered starting with 0) are connected to Cal0 while the inputs of the odd numbered inverters are connected to Cal1. This type of configuration allows independent control over the propagation speed of the two transitions associated with the negative pulse. For example, increasing the voltage on Cal1 toward the supply voltage allows the odd numbered inverters to switch more quickly. With Cal0 fixed at a specific voltage, larger Cal1 voltages allows the pulse to survive longer in the delay chain because it takes longer for the trailing edge to catch up to the leading edge. All latches up to the point where the pulse disappears store a '1', while those beyond that point store '0'.

We determined that the highest resolution and lowest noise is achieved by setting Cal0 to the supply voltage. Cal1 is tuned for each chip in a calibration process described below. An off-chip voltage source drives the Cal1 control signal in our experiments. Alternatively, this input signal can be derived from an on-chip resistor ladder network (a feature we intend to add to our next test chip). The TDC occupies an area of 176 um x 60 um (10 km^2).

The on-chip integration of the TDC makes it vulnerable to manipulation by an adversary. However, the high sensitivity of the TDC to systematic changes in path delay in combination with an externally controlled calibration method (both described below) significantly increase the challenge for an adversary to tamper with the instrument and defeat our verification methodology.

3.2 TDC Sensitivity Analysis

The Cal1 input allows the timing resolution to be tuned, trading off range and resolution. The curves in Fig. 5 illustrate the tradeoff using data collected from one of the 90 nm chips and one of the TDCs. The digital clock manager (DCM) from a Xilinx Zynq FPGA is used to drive a pulse into the Clk input shown on the left side of Fig. 4. The DCM allows the width of the pulse to be tuned with a timing resolution of approx. 18 ps. The x-axis of Fig. 5 plots the pulse width which could be varied from approx 1.2 ns (the smallest possible between the FPGA and the test chip board) up through approx. 14 ns. The y-axis plots the TC produced by the TDC, a value between 0 and 120. The individual curves plot data using different values of Cal1. For lower Cal1 values, e.g., 600

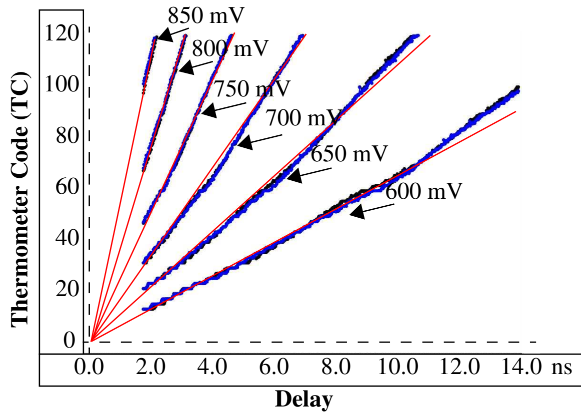

Fig. 5. TDC sensitivity analysis: TC value produced by the TDC (y-axis) as the input pulse width is varied (x-axis) using a variety of Cal1 values (separate curves).

mV, the timing resolution is approx. 130 ps per TC value while for higher values, e.g., 850 mV, it increases to approx. 20 ps per TC. The calibration process described above tunes Cal1 to values between 750 mV and 850 mV across all chips, which provides approx. 25 ps of timing resolution on average in our experiments.

3.3 HT Emulation Circuit

We inserted a set of 20 HT emulation circuits (HTECs) into each of the AES MUTs (the placement is identical in both AES units). A schematic representation of the HTEC is shown in Fig. 6. HTECs are inserted in series with an upstream and downstream gate at a variety of places in the MUTs' netlists. It consists of a complementary NFET/PFET *pass gate* that can be enabled or disabled in a controlled fashion using a scan chain FF. An additional PFET, labeled *AV-PFET*, is connected in parallel with the pass gate, whose gate is controlled from an external voltage source using the *analog ctrl pin*. With the pass gate disabled, the analog ctrl pin can be used to change the resistance between the upstream gate and downstream gate, which, in turn, impacts the delay added by the HTEC. We refer to the voltage applied to the analog ctrl pin as the analog voltage or **AV**. The HTEC cell is designed to allow the magnitude of the delay anomaly to be tuned, which in turn, is key to evaluating the capabilities and limitations of our proposed detection methods.

3.4 Dealing with Process Variations

3.4.1 Calibrating Chip-to-Chip Variations

The ability to tune the TDC using the Cal1 input signal provides a mechanism to virtually eliminate chip-to-chip process variations effects. These effects include both a global shift and scaling of all path delays to values above or below the nominal value. We implement a calibration process that computes, for a fixed value of Cal1, the average TC from a set of stable paths. A binary search process then tunes the Cal1 value (and re-tests the paths) until the average TC equals a user specified value. We choose a value of 60 in our experiments, which is 1/2 the maximum TC value of 120. Note that calibration not only addresses chip-to-chip process variation effects but also reduces measurement bias in the on-chip TDCs.

3.4.2 Calibrating Within-Die Variations

Within-die variations currently represent one of the most significant challenges for parametric HT detection methods. We have developed a chip-averaging method that significantly reduces the adverse effects of within-die variations. Chip-averaging simply

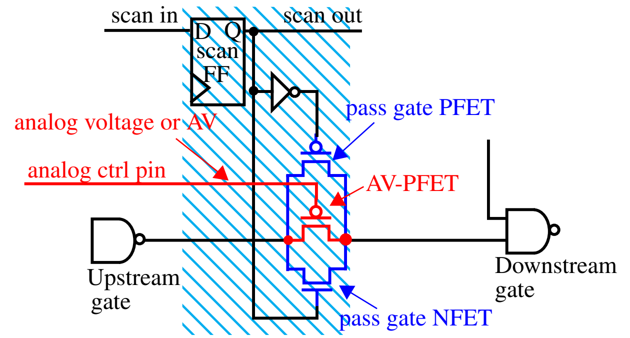

Fig. 6. HT Emulation Circuit. Red highlighted components are used to change the delay characteristics.

computes the average delay of a path from measurements made from all chips (or a statistically significant sample). If within-die variations are random, then the averaging operation will eliminate them, revealing any type of systematic delay anomaly, i.e., an increase in delay that consistently occurs in all chips. The experimental results that we show in this paper illustrate that within-die variations are largely random and therefore, can be nearly eliminated using chip-averaging. This is a very powerful feature, and to the best of our knowledge, has not been previously proposed and/or demonstrated in hardware. Eq. 1 gives the expression for computing a chip-averaged-delay (CAD), where subscripts *A*,*P* indicate the AES unit (0 or 1) and the path ID (0 to n).

$$\mathrm{CAD}_{A,\,P} = \frac{1}{\#\,\mathrm{chips}} \sum_{i\,=\,1}^{\#\,\mathrm{chips}} D_{A,\,P} \qquad \textbf{Eq. 1.}$$

3.4.3 Path Selection Criteria

Only paths that are classified as stable in both AES units across all 44 chips are used in this analysis. Also, a path is discarded from the analysis if the measured delay from any of the chips exceeds a 4 σ limit. The 4 σ limit for each path is derived by computing the standard deviation using the delays from all of the chips. Therefore, a delay that exceeds the limit is an outlier in the population. This catches paths that are unstable but are able to produce a single rising or falling edge and therefore are missed by the EdgeCnt check as discussed in reference to Fig. 4. The criteria for a path to be included in the analysis is very strict because we do not want glitching or small delay defects to bias the CAD value away from its nominal value, i.e., its value without process variations.

4. EXPERIMENTAL RESULTS

4.1 HT Emulation Circuit Experiments

A randomly selected sequence of 500 test vector pairs are applied to the two AES MUTs simultaneously. The same sequence is applied to each AES MUT to make it possible to directly compare the measured path delays. In these experiments, all HTECs pass gates are disabled as the vectors are applied, which means the pass gate controlled by the scan FF in Fig. 6 is in a high-impedance state and only the AV-PFET is conducting. Therefore, the delays of signals connected to the TDC inputs from paths sensitized from a HTEC site within the MUTs will depend on the value of AV. The same sequence of 500 vectors is applied 11 times to the 2 AES MUTs, each with a different AV applied. The values used for AV vary from 0.0 V (referred to as the **HT-free** value since this

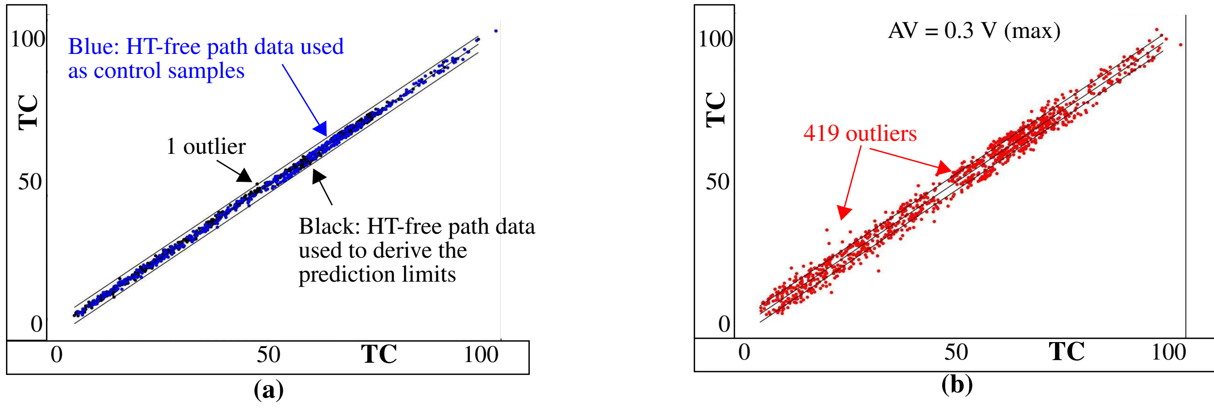

Fig. 7. (a) Regression analysis of HT-free control samples from HTEC experiments, (b) using HT data points with AV = 0.3 V showing outliers.

value produces the smallest delay) to 0.3 V in 30 mV steps. If a path is sensitized from a HTEC site, then the sequence of delays will change as the AV is varied. Note that the delays can increase or decrease depending on whether the edge effected by the HTEC cell is ahead or behind the edge on the other path (as indicated earlier, the TDC computes the relative delay between two paths). We use this feature to distinguish between HT and HT-free timing measurements.

Each TDC can be used to time 64 different combinations of the P_{Ax} and P_{Bx} inputs as each vector is applied. The 8 P_{Bx} inputs that are timed against TDC_x P_{A0} input, which is the Clk input, represent a special case because the arrival of the Clk remains constant for each test. Therefore, the length of the path driving the P_{Bx} input is proportional to the magnitude of the TC. The length of the paths under the other 56 combinations cannot be determined, however, but instead reflect the magnitude of the relative difference between the two path delays.

Although it is possible to obtain up to 128 path delay measurements from the two TDCs for each MUT for each vector pair by repeatedly applying the test vector pair and selecting a different combination using the *sel A* and *sel B* MUXes, we ignore data obtained for TDC_x-2 inputs P_{B3} through P_{B7}. As mentioned earlier, we inserted extra gates in AES_2 to model actual HTs and these inputs are on sensitized paths from these inserted gates. A special set of experiments are carried out to determine the detectability of these hand-inserted HTs in Section 4.2. Therefore, only 24 of the 64 path pairings within TDC_x-2 are used in these experiments, i.e., path inputs P_{B0} through P_{B2} inputs are tested against each of the P_{A0} through P_{A7} inputs. These 24 path pairings are combined with the 64 from TDC_x-1 to provide a total of 88 path pairing per vector. Therefore, the stable paths from 500 vectors * 88 path pairings, i.e., 44,000 path pairings, are selected for inclusion in this analysis.

HT-free data is obtained from path pairings whose delay does not change with AV (only the timing data obtained with AV set to 0.0 V is used). A separate set of HT-free **control** data is obtained from path pairing whose delay **does** change as AV is increased. In this case, only the delays measured with AV set to 0.0 V are used (which is also considered HT-free). The control data is used to determine how many false positive detections occur in our statistical detection method. Finally, the HT data is obtained using the HT-free control paths but with AV > 0.0 V. Given these constraints and those described earlier in Section 3.4.3, our post processing produced data sets with the following cardinality and characteristics:

- HT-free path pairings: 728
  These are used to derive the statistical limits.
- HT-free control path pairings: 557
- HT path pairings: (557 * 2) per AV
  Since each path that is affected by a HTEC occurs in both AES units, the number of HT paths that can be tested doubles. AVs between 0.03 V and 0.30 V in steps of 30 mV are used (for a total of 10 steps) in the HT experiments so 11,140 HT path pairings are tested altogether.

Regression analysis is applied to the scatterplot created using the 728 HT-free paths as shown in Fig. 7(a). Here, the CADs from paths in AES_1 are plotted against the CADs from the same paths in AES_2. The black data points represent the HT-free paths while the blue data points represent the HT-free control paths. A regression line and prediction limits are derived using the black data points. The limits were increased slightly above 3 σ (99.73) to 3.5 σ (99.9) to account for the slight increase in noise that occurs when two MUT paths are timed against each other (as opposed to paths that are timed against the Clk only). Only one outlier is present in the data as highlighted in the figure (from the HT-free data). All HT-free control data points fall within the limits. The average width of the band is approx. +/- 2.5 TCs, which means that uncompensated process variations and noise add uncertainty of approx. 63 ps.

Fig. 7(b) plots the HT data points with AV = 0.3 V (the largest value that we used in our experiments) on top of the prediction limits to illustrate the number of outliers for this scenario. The data point pairings in this case are created by using the AV = 0.0 V CADs from AES_1 as the first element of the pairing and the AV = 0.3 V CADs from AES_2 as the second element (and vise versa). In other words, the data point pairings are created from a combination of HT-free and HT CAD values. This is consistent with the self-referencing technique where it is assumed that the adversary only adds the HT to one of the functional units.

Table 1 shows the detection results, partitioned into rows according to the applied AV. The 1st column gives the AV value, the 2nd column gives the average increase in the TC (and the corresponding delay), while the last column gives the number of detections (and as a percentage of the total). The average increase

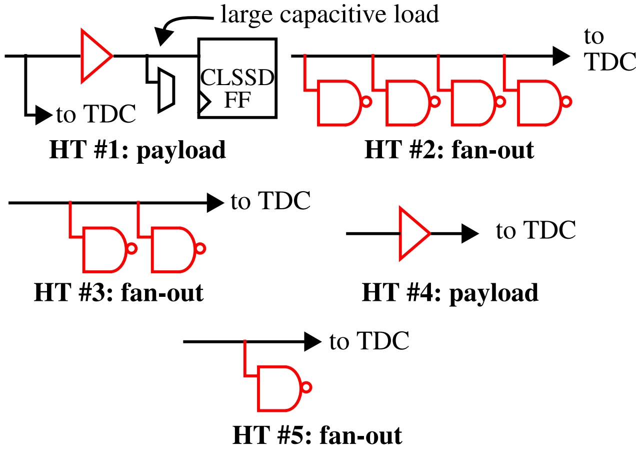

Fig. 8. Manually inserted fan-out and payload HTs in AES_2.

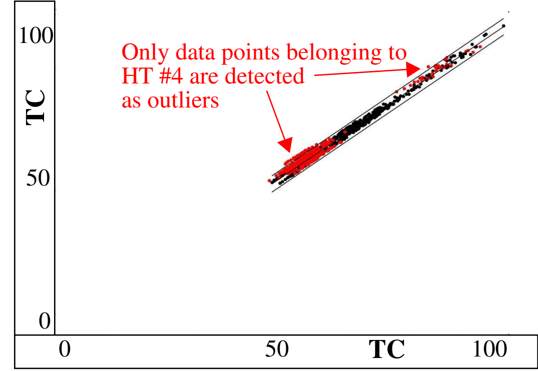

Fig. 9. Regression analysis of fan-out/payload HTs.

in TC is computed using the differences in the HT-free control CADs (with AV = 0.0) and the HTs CADs (with AV > 0.0 V) across all 1,114 path pairings. As indicated earlier, 1 TC ~= 25 ps. Although some of the 40 HTECs (20 in each AES) are detected for lower values of AV, detecting them all is not possible until the delay anomaly becomes greater than 1 TC. A 5X increase in the level of confidence of a positive detection occurs when the anomaly reaches 2 TCs. For perspective, the fanout-of-4 gate delay is approx. 35 ps for a 1X inverter in this technology. Therefore, these results indicate that its is possible to detect an extra inverter in a path with reasonably high confidence.

Table 1: HT detection results from HTEC experiments.

AV (V)	Average increase in TC (ps) over HT-free path	Number (%) of paths that detect an HT (of 1,114)
0.03	0.27 (7 ps)	3 (0.3%)
0.06	0.53 (13 ps)	7 (0.6%)
0.09	0.79 (20 ps)	34 (3.1%)
0.12	1.05 (26 ps)	67 (6.0%)
0.15	1.30 (33 ps)	115 (10.3%)
0.18	1.55 (39 ps)	176 (15.8%)
0.21	1.79 (45 ps)	246 (22.1%)
0.24	2.02 (51 ps)	302 (27.1%)
0.27	2.25 (56 ps)	359 (32.2%)
0.30	2.46 (62 ps)	419 (37.6%)

4.2 Fan-out and series-inserted HT Experiments

The objective of these experiments is to determine if it is possible to detect a set of HTs implemented with standard cell logic gates connected as fan-out loads or in series with paths in the AES_2 MUT. Similar to the self-similarity strategy taken in the previous section, we compare path delays from two 'nearly' identical copies of the AES to determine if systematic anomalies can be measured and detected on paths that posses an HT.

4.2.1 Fan-out and Series Inserted Trojans

As mentioned earlier, we purposely made the layouts for AES_1 and AES_2 (including the TDCs) nearly identical. The Cadence synthesis and place and route tools were used to create one copy of the layout that was then placed into the two locations as shown in Fig. 2. The layout editor was then used to make small changes in the copy labeled AES_2. In particular, filler cells were removed and additional gates were added as shown in Fig. 8. The red components represent the HT while the black components represent the existing elements in the layout. Each HT was designed to effect only one of the AES outputs. HT #1 adds a buffer which isolates the large capacitive load present on the capture FF input from the wire that connects to the TDC input. Therefore, the delay along paths to the output for AES_2 should be less than the delay for AES_1 which does not include the isolation buffer. HTs #2, #3 and #5 add capacitive load to existing wires while HT #4 includes a series inserted buffer. Therefore, the path delays from AES_2 for HTs 2 through 5 are expected to be longer than those measured from AES_1.

4.2.2 Methodology

We again applied a set of 500 random vector pairs to the 2 AES MUTs. Unlike the analysis carried out in the previous section, all TDC path measurements are carried out using the Clk as reference, i.e., only P_{A0} from Fig. 3 is used, and all HTECs are disabled. The HTs referred to in Fig. 8 affect TDC_2-2 P_{B3} through P_{B7} inputs while the remaining 11 TDC inputs (8 from TDC_2-1 and 3 from TDC_2-2) are HT-free.

4.2.3 Regression Detection Results

Fig. 9 plots the CADs for AES_1 along the x-axis against the CADs for AES_2 along the y-axis (similar to the plots described in reference to Fig. 7). A regression line and 3 σ prediction limits are derived using 676 HT-free CADs, while the HT CADs for 468 paths are shown in red. Path delays begin at approx. TC = 50 because the Clk edge always arrives on the TDC PA_0 before any of the transitions propagating through the AES MUTs arrive on the TDC PB_x inputs. Outliers occur in the region as labeled. Unfortunately, only HT #4 is detected using regression analysis. Therefore, regression is not sensitive enough to detect most of these subtle HT delay anomalies.

4.2.4 Trending Detection Results

In this section, we develop an alternative detection method that is based on the analysis of trends in the CAD differences. The trends introduced by systematic anomalies are captured by computing a **difference** CAD or DCAD value defined by Eq. 2.

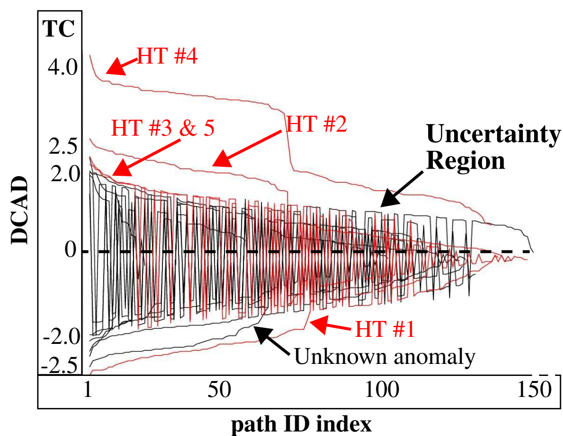

Fig. 10. DCAD values on y-axis for paths tested through each of the TDC inputs, sorted on the magnitude of the differences from largest (left) to smallest (right).

$$\mathrm{DCAD}_{A,\,P} = \mathrm{CAD}_{1,\,P} - \mathrm{CAD}_{0,\,P} \qquad \textbf{Eq. 2.}$$

We partition the DCAD values computed for the stable paths under all 500 vectors into 16 groups, with each group corresponding to one of the HT or HT-free TDC inputs. Each of the groups of DCAD values are plotted as a curve in Figs. 10 and 11. All 16 curves are superimposed in Fig. 10 while Fig. 11 partitions the curves into HT and HT-free groups. The data points in each curve represent all of the stable paths that were tested. We only include paths that go through the HT site for the HT TDC inputs, and ensure that all paths included in the HT-free curves are HT-free.

The values are sorted from left-to-right on the magnitude of the DCAD value, with the largest differences on the left. The shaded region labeled "Uncertainty Region" represents the noise floor, i.e., the point at which the DCAD values in the curves begin jumping between negative and positive values in a random fashion. The cone-like shape, with a maximum width of +/- 2 TCs on the left, indicates that noise is proportional to the magnitude of the difference. The ordering of the data points along the x-axis also correlates to path length, with the data points on the left side of the curves produced by the longest paths.

The curves above and below the Uncertainty Region capture a systematic component of variation in path delay, i.e., a shift in delay that is similar in all chips. The 5 HT curves are labeled, and are consistent with the expected behavior as described in Section 4.2.1. For example, HT #1 isolates the large load capacitance of the CLSSD capture FF and therefore, the delay in AES_2 is smaller than the delay in the AES_1 (the nominal version), producing a curve with negative values. Larger positive increases in delay are expected in the curves for HT #2 and #4, while the expected increase in delay of HT #3 and #5 should be smaller. The average TC over the left portion of the curves is 2.5, 3.75, 2.0 and 2.0 for these 4 HTs, which is consistent with the expectations. Note that the curves for HT #3 and #5 are very close to the noise floor, making it difficult to decide with high confidence that a systematic anomaly exists. Therefore, fan-out loads of 1 or 2 gates represent the most challenging HT characteristics to detect. However, the trend of the data points for the HT inputs to remain for at least some fixed interval above or below the 0 line provides evidence of a systematic anomaly. Notice that this representation, unlike the

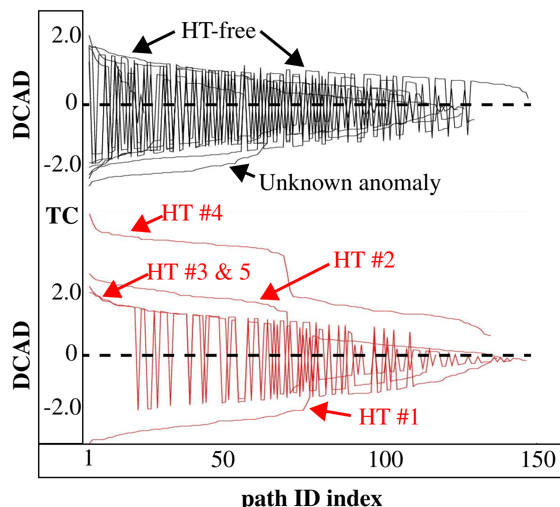

Fig. 11. Same as Fig. 10 except HT-free and HT results are separated.

regression curve in Fig. 9, clearly shows that HT #1 and #2 are anomalous.

All 11 of the HT-free curves in Figs. 10 and 11 fall within the Uncertainty Region as expected except for 1 curve. The reason that a systematic anomaly exists in the curve labeled 'Unknown anomaly' is unclear. We carefully checked the layout for wiring differences between the 2 AES units for this TDC input and found only small differences in capacitance coupling introduced by cross-over wires that are present in AES_2 but not in AES_1. We do not believe these are significant enough to introduce the observed anomaly. We also tested each of the 4 TDCs across all chips using a series of clock pulses (similar to that described in reference to Fig. 5), computed chip-averages and compared the CAD values of TDCs connected to AES_1 with those computed for AES_2. The differences were less than 1 TC, which indicates that the TDCs themselves are not introducing the anomaly. The fact that the anomaly is present, and is consistent across a large number of paths and chips makes it likely that there is a physical difference in the layouts (which we cannot find), or there is some type of systematic process variation (which is unlikely because this would effect more than just one of the TDC inputs).

4.2.5 Analysis of the Effectiveness of Chip-Averaging

In this section, we quantify the level of reduction in noise and within-die variations provided by chip-averaging. As indicated above, the CAD values are computed from the TC values measured from 44 chips. Fig. 12 plots the individual chip TCs for a set of 5 HT-free path pairings, labeled P_1 through P_5, as the first 44 values along the x-axis and the CAD value as the last (right-most) data point. The two curves associated with each path pairing are derived from each of the two AES units. The length of the path delay difference increases from P_1 through P_5 as reflected in the range of TCs. The variation in the individual chip values is caused by noise and within-die process variations. Given all 5 path pairings are HT-free, the CAD values for each pairing of paths on the far right should be equal in the absence of noise. The difference between the CAD values, defined earlier as DCAD, in all 5 cases is close to the ideal case of 0. The largest DCAD value is 1.2, as given by P_3 CAD values of 79.2 and 78.0. The range of variation in the individual chips, on the other hand, is given by 85 - 74 = 11.

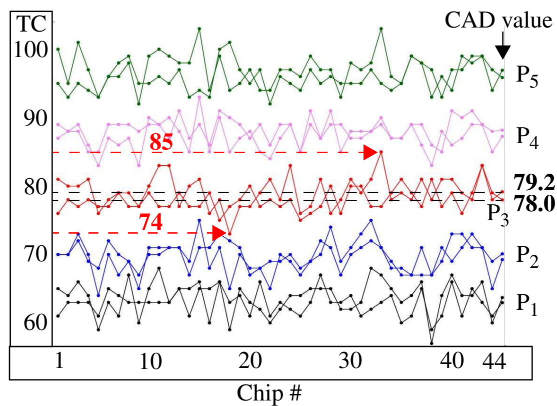

Fig. 12. TCs from individual chips and the chip-averaged delay value (CAD).

Therefore, chip-averaging reduces undesirable variations introduced by noise and within-die variations by almost a factor of 10, allowing small systematic variations introduced by HTs to be more easily detected.

5. CONCLUSIONS

In this paper, we present chip results of using a self-similarity-based method to detect hardware Trojans (HTs). An embedded test structure called a time-to-digital converter is used to obtain high resolution, e.g., 25 ps, measurements of path delays from two nearly identical copies of AES. A chip-averaging technique is shown to significantly reduce the adverse effects of within-die process variations on HT detection sensitivity. Statistical methods are used to detect path delay anomalies introduced by HT emulation circuits and HTs implemented by the insertion of standard cell logic gates. The chip-averaging technique is a generalized approach of dealing with the adverse effects of within-die process variations on HT sensitivity, that is also applicable to other side-channel measurements such as leakage current [26].

6. REFERENCES

[1] "Secure, Trustworthy, Assured and Resilient Semiconductors and Systems (STARSS) Workshop", sponsored by SRC/NSF, San Jose, CA, May 21, 2014, https://www.src.org/calendar/e005440/

[2] "Design for Security Working Meeting", sponsored by USC, ISI and US Army Research Office, Marina del Rey, CA, July 23, 2014, https://uscisi.atlassian.net/wiki/display/DFSWM

[3] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, "Trojan Detection using IC Fingerprinting", Symposium on Security and Privacy, 2007, pp. 296 - 310.

[4] F. Wolff, C. Papachristou, S. Bhunia, and R. Chakraborty, "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme", *DATE*, 2008, pp. 1362-1365.

[5] R. S. Chakraborty, S. Paul and S. Bhunia, "On-Demand Transparency for Improving Hardware Trojan Detectability", *HOST*, 2008, pp. 48-50.

[6] M. Banga and M. S. Hsiao, "ODETTE: A Non-Scan Design-for-Test Methodology for Trojan Detection in ICs," *HOST*, 2011, pp. 18-23.

[7] D. McIntyre, F. Wolff, C. Papachristou, S. Bhunia and D. Weyer, "Dynamic Evaluation of Hardware Trust," *HOST*, 2009, pp. 108-111.

[8] H. Salmani and M. Tehranipoor, "Layout-Aware Switching Activity Localization to Enhance Hardware Trojan Detection", *Trans. on Information Forensics and Security*, Vol. 7, Issue: 1, Part: 1, 2012, pp. 76-87.

[9] M. Banga and M. S. Hsiao, "A Region Based Approach for the Identification of Hardware Trojans", *HOST*, 2008, pp. 40-47.

[10] J. Zhang, Y. Haile and X. Qiang, "HTOutlier: Hardware Trojan Detection with Side-Channel Signature Outlier Identification", *HOST*, 2012, pp. 55-58.

[11] L. Wang, H. Xie and H. Luo, "A Novel Analysis Method of Power Signals for Integrated Circuits Trojan Detection", *International Symposium on Physical and Failure Analysis of Integrated Circuits*, 2013, pp. 637-640.

[12] C. Lamech, R. M. Rad, M. Tehranipoor, J. Plusquellic, "An Experimental Analysis of Power and Delay Signal-to-Noise Requirements for Detecting Trojans and Methods for Achieving the Required Detection Sensitivities," *Transactions on Information Forensics and Security*, Vol. 6, No. 3, 2011, pp. 1170-1179.

[13] M. Potkonjak, A. Nahapetian, M. Nelson and T. Massey, "Hardware Trojan Horse Detection using Gate-Level Characterization," *DAC*, 2009, pp. 688-693.

[14] D. Rai and J. Lach, "Performance of Delay-based Trojan Detection Techniques under Parameter Variations", *HOST* 2009, pp. 58-65.

[15] C. Lamech and J. Plusquellic, "Trojan Detection Based on Delay Variations Measured using a High-Precision, Low-Overhead Embedded Test Structure," *HOST*, 2012, pp. 75-82.

[16] A. Davoodi, L. Min and M. Tehranipoor, "A Sensor-Assisted Self-Authentication Framework for Hardware Trojan Detection", *IEEE Design & Test*, Vol. 30, Issue: 5, 2013, pp. 74-82.

[17] I. Exurville, J. Fournier, J.-M. Dutertre, B. Robisson and A. Tria, "Practical Measurements of Data Path Delays for IP Authentication & Integrity Verification", *Workshop on Reconfigurable and Communication-Centric Systems-on-Chip*, 2013, pp. 1-6.

[18] W. Sheng and M. Potkonjak, "Malicious Circuitry Detection using Fast Timing Characterization via Test Points", *HOST*, 2013, pp. 113-118.

[19] C. Byeongju and S. K. Gupta, "Efficient Trojan Detection via Calibration of Process Variations", *Asian Test Symposium*, 2012, pp. 355-361.

[20] Y. Jin and Y. Makris, "Hardware Trojan Detection Using Path Delay Fingerprints", *HOST*, 2008, pp. 51-57.

[21] Jie Li and John Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection", *HOST*, 2008, pp. 8-14.

[22] K. Xiao, X. Zhang, M. Tehranipoor, "A Clock Sweeping Technique for Detecting Hardware Trojans Impacting Circuits Delay," *IEEE Design & Test*, Vol. 30, No. 2, 2013, pp. 26-34.

[23] D. Du, S. Narasimhan, R. S. Chakraborty, and S. Bhunia, "Self-referencing: a Scalable Side-Channel Approach for Hardware Trojan Detection, *CHES*, 2010, pp. 173-187.

[24] N. Yoshimizu, "Hardware Trojan Detection by Symmetry Breaking in Path Delays", *HOST*, 2014, pp. 107-111.

[25] Stephan Henzler, "Time-to-Digital Converters", *SpringerLink*, Volume 29 2010, ISBN: 978-90-481-8627-3 (Print) 978-90-481-8628-0 (Online).

[26] I. Wilcox, F. Saqib and J. Plusquellic, "GDS-II Trojan Detection using Multiple Supply Pad V_{DD} and GND I_{DDQ}s in ASIC Functional Units", To appear, *HOST*, 2015.