At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection

Jie Li
Charles L. Brown Department of
Electrical and Computer Engineering
University of Virginia
Charlottesville, USA
jieli@virginia.edu

John Lach
Charles L. Brown Department of
Electrical and Computer Engineering
University of Virginia
Charlottesville, USA
jlach@virginia.edu

Abstract—**New attacker scenarios involving integrated circuits (ICs) are emerging that pose a tremendous threat to national security. Concerns about overseas fabrication facilities and the protection of deployed ICs have given rise to methods for IC authentication (ensuring that an IC being used in a system has not been altered, replaced, or spoofed) and hardware Trojan Horse (HTH) detection (ensuring that an IC fabricated in a non-secure facility contains the desired functionality and nothing more), but significant additional work is required to quell these treats.**

This paper discusses how a technique for precisely measuring the combinational delay of an arbitrarily large number of register-to-register paths internal to the functional portion of the IC can be used to provide the desired authentication and design alteration (including HTH implantation) detection. This low-cost delay measurement technique does not affect the main IC functionality and can be performed at-speed at both test-time and run-time.

Keywords-path delay characterization; hardware security; IC authentication; hardware Trojan Horse detection

I. INTRODUCTION

Hardware security has recently become a cause for serious concern in a variety of IC design, fabrication, testing, and deployment scenarios. IC authentication and HTH detection are two concepts that are gaining traction in the research community, but significant additional work is necessary to achieve the required level of IC protection.

A. Motivation: IC authentication

A strong security protection mechanism for ICs remains a challenge. For some applications where hardware authentication is used, such as Smartcard systems, designers face the difficulty of preventing adversaries from hacking and cloning the ICs embedded in these systems. Sometimes the ICs can be so vulnerable to different types of attacks that the security of the ICs can be compromised with relatively little effort. For example, Nohl et al. [8] have shown that they were able to reconstruct a cipher from its implementation in a widely-used RFID tag using image analysis of circuits and protocol analysis. Therefore, once an RFID tag is obtained by an adversary, an identical tag capable of passing authentication can be cloned.

Several new techniques for IC authentication, such as the Physical Unclonable Functions (PUF) [3], leverage IC manufacturing variability and are able to extract unique signatures from every manufactured IC based on non-functional characteristics such as delay and power. A number of PUF designs [4][10] have been implemented to show the feasibility of IC authentication using PUFs, most of which are circuit-delay-based silicon PUF designs. However, one limitation of these PUF designs is that the challenge-response pairs for generating the signature of the IC are limited in the length of the signatures they can provide with low area overhead (1 bit output for every combinational path pair), as they do not directly measure the path delay of the circuit but instead use an n-bit input vector as the challenge to generate a 1-bit response to indicate the relative propagation speed of two identical (by design but not fabrication) paths. This simple mechanism may make it easy for the attackers to construct precise timing models about the PUF circuits and use them to predict the response of a given challenge and therefore spoof the IC and pass authentication.

B. Motivation: HTH detection

IC designers (including governments and defense agencies) are increasingly outsourcing their IC fabrication to third parties in whom they may not have complete trust. This creates a high security risk for the systems that employ these ICs, as an attacker may have embedded malicious circuitry (e.g. HTHs) into the original circuit or made some other design alteration. In these situations, it is critical that the users be able to detect the malicious design alteration at test-time or at least prevent it from causing problems at run-time. Current techniques for addressing this goal rely primarily on functional tests. However, the complexity of modern ICs prevents exhaustive testing and limits controllability and observability, and a well-hidden HTH is likely to go undetected.

C. At-speed delay characterization

This paper introduces a new method for IC authentication and HTH detection based on delay characterization. This method utilizes a cost-effective at-speed delay measurement

1-4244-2401-6/08/$20.00©2008 IEEE

technique [5] that was originally developed to characterize process variation. It can be applied on a large number of otherwise unobservable internal combinatorial paths to get accurate, precise data about register-to-register path delays. This technique can be used to generate longer signatures than existing PUF designs. Furthermore, unlike most techniques that are applied to non-functional paths specifically inserted to be used as a PUF, this technique can be performed on the functional paths of the core circuit without affecting timing and functionality, thereby making it significantly more difficult for an attacker to bypass, remove, or spoof the signature extraction unit.

This precise delay characterization technique can also be applied for HTH detection by extracting the non-functional path delay characteristics to detect malicious circuit alterations. Even relatively small changes in a design can cause significant changes in the delay characteristics of the altered path and even neighboring paths. Therefore, by comparing the delay projections at design-time (when the manufacturer extracts the delay model of the original circuit, including projected manufacturing variations) and delay measurements at test-time (when the user measures the actual path delays after any non-trusted IC fabrication), we are able to evaluate the consistency of the statistical delay models and hence detect whether the design has been altered. However, some dormant HTHs may still be difficult to detect at test-time, requiring a backup method for HTH detection when it is sprung at run-time. Given that the delay measurement technique is at-speed and does not affect the circuit functionality, it is possible to monitor these delay characteristics continuously in the field at run-time. When an HTH is sprung, the delay characteristics will change, enabling fast detection and ultimately containment even when a functional monitor does not detect anomalous behavior.

The remainder of this paper is organized as follows. Section II discusses related work. Section III explains the technique for precise path delay measurement. Section IV introduces the process for generating unique signatures and applying them to IC authentication, and Section V discusses the approach for HTH detection. Section VI presents the experimental results of this approach, and Section VII presents a summary and future work.

II. Related Work

A. PUFs

The basis of the IC authentication method detailed in this paper is signatures generated from the non-functional characterization of an IC, with manufacturing variations ensuring the uniqueness and unclonability of the resulting generated signatures. This is also the basis of much of the work that has been done on PUFs, which are functions that map a set of challenges to responses that are generated from, and hence reflect, the unique physical characteristics of each device [3][4][10]. As a result, PUFs can provide higher security than other soft-key-based cryptographies because they extract the security information from the physical system itself, rather than storing this information in non-volatile memory.

Fig. 1 shows a sample silicon PUF based on a delay circuit for generating the challenge-response pairs. An n-bit challenge vector C is input to the n-stage string of switch blocks as the control vector – one bit for each switch block. Every switch block has two inputs, two outputs, and a control bit. Depending on the value of the control bit, the inputs of the switch block will either go straight through to the outputs or will be exchanged. In this way, the circuit can create a pair of delay paths for each input C vector. To evaluate this circuit, a rising edge is also input to stimulate both of the circuit paths. This signal race through these two paths, and one of the outputs of the last switch block is used as the response to that challenge vector.

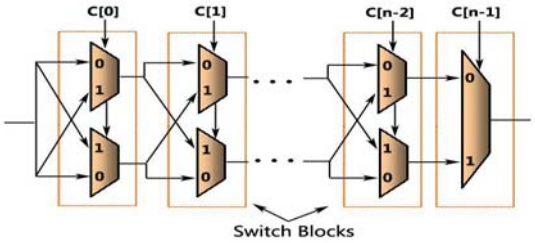

Figure 1. A typical PUF delay circuit

While such PUFs have been shown to be effective for IC authentication, most PUF circuits are rather simple and generate only one signature bit from every combinational path pair. It may therefore be possible for an adversary to construct precise timing models for the PUF circuits from already observed challenge-response pairs and use them to predict the responses to future challenges. The vulnerability to these model-building attacks forces security extensions to be built on top of this original delay circuit, which introduces additional overhead. In addition, the PUF circuits themselves are non-functional and impose a non-trivial overhead.

As described in Section III, the delay characterization technique described here generates a much longer signature for each individual delay path, making it more difficult for attackers to generate timing models. In addition, the paths being characterized are part of the main circuit functionality, both reducing area overhead and making it much more difficult to remove or sidestep the authentication mechanism.

B. HTH detection

While Trojan Horses have long been a concern in software, the rising concern about HTHs has led to several efforts to effectively detect and contain them, many of which depend on non-functional circuit characterization. For example, Agrawal et al. proposed a side-channel-based approach that extracts non-functional circuit information, including path delays, power consumption, and electromagnetic emanation profiles, that constitutes a “fingerprint” for every individual IC [1]. This approach is effective when detecting an HTH that would be activated by a trigger condition, where detection using purely functional testing is practically impossible. However, the extraction of the side-channel information introduces large

overhead and is time-consuming. Similarly, Wolff et al. provide an HTH taxonomy and propose an approach to detect HTHs by analyzing and choosing an optimal set of test vectors [12]. However, it also takes a long time to exhaustively analyze all of the input patterns.

In contrast to these existing HTH detection techniques, the approach described here provides a low cost procedure for HTH detection that can be applied in every phase of IC life cycle, including design-time, test-time, and run-time. The capability of run-time monitoring will be especially useful to detect and contain those HTHs undetected at test-time.

III. Path Delay Characterization

A. Architecture and approach

In this section, the implementation details for the delay characterization technique [5] are presented, with the core concept being the use of negatively-skewed shadow registers placed at the end of a number of combinational paths. The number and selection of these paths are determined by the designer and must be based on considerations of IC authentication and HTH protection and the die-area costs.

As shown in Fig. 2, the *Main Circuit* is one register-to-register combinational path delay that is to be characterized, and the registers on this path are triggered by the main system clock (*clk1*). The components outside the dotted box are the testing circuitry inserted under the characterization approach. The *Shadow Register* takes the same input as the *Destination Register* in the *Main Circuit*, but it is triggered by the shadow clock (*clk2*), which runs at the same frequency as *clk1* but at a controlled phase offset. The results that are latched by the *Destination Register* and the *Shadow Register* are compared during every clock period until a discrepancy is identified.

To this point, there are no significant differences between this technique and Razor [2]. However, in the Razor approach, *clk2* is positively skewed to trigger the *Shadow Register* a fixed amount of time after the *Destination Register*, and both clock frequencies are increased until the destination register on any path stores a different (i.e. incorrect) value than its shadow register. Once this occurs, a recovery process is initiated, and the system clock is reduced.

In the technique described here, *clk1* remains constant, so the functionality of the *Main Circuit* is not affected by the delay characterization testing. It is this key difference that enables at-speed testing without adding any complexity to the testing process. Instead of increasing the frequency of either clock, a negative phase shift (i.e. negative skew) is applied to *clk2* in small steps, triggering the *Shadow Register* before the *Destination Register* by a precisely controlled amount of time.

Depending on the purpose of the delay characterization (parameter variation characterization, IC authentication, HTH detection, etc.), the *Main Circuit* may run for some amount of time at each skew step to enable a variety of data to exercise difference delay paths. During each clock period, the results in the *Shadow Register* and the *Destination Register* are compared, and a 1-bit flip-flop stores the *Result Bit* of that comparison. If the comparison result is ever unequal (i.e. the *Shadow Register* has latched an incorrect value by triggering before the data has settled), the *Result Bit* is set to '1' and stays that way regardless of future comparisons. After the circuit has run at one skew step for some time, the *Result Bit* is read out via a scan chain connecting the 1-bit flip-flops at the end of every tested path. The delays for paths that set their *Result Bit* to '1' for the first time at that skew step have been characterized to a precision of the skew step size. The *clk2* negative skew is then increased another step, and the process repeats itself until all of the tested path delays have been characterized. Again, none of this testing affects the functionality of the *Main Circuit*, enabling orthogonal at-speed testing.

[5] addresses a number of technical issues related to this approach, and experimental results on a set of FPGAs have shown the utility of this delay characterization mechanism. (For example, it does not matter if the routing delays from the output of the *Combinational Path* to the *Destination Register* and the *Shadow Register* are different, as a valid delay signature is still extracted.) However, some of these issues will directly impact the uniqueness of extracted signatures for IC authentication and the sensitivity of HTH detection. The first issue is the precision of the clock skew control – the higher the precision, the higher the signature uniqueness and HTH detection sensitivity. In our experiments, a skew step of 160ps is used, which is the highest resolution that can be supported by our experimental device, the Xilinx Virtex-II FPGA.

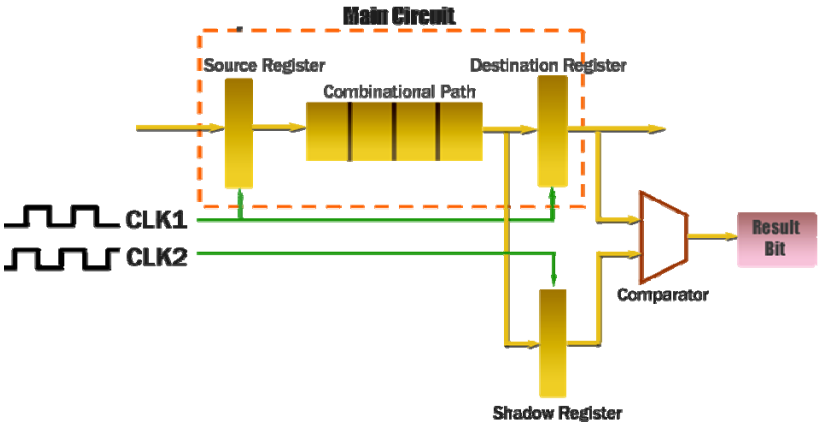

Figure 2. Delay characterization with negative-skewed shadow register [5]

However, recent advancements have provided much higher precision and accuracy for nominal costs. For example, a novel variable-delay element with an inverter and a digitally adjustable resistor was shown to have a clock skew resolution of 1ps in 180nm fabrication technology [9].

A few other factors may also affect the precision of the delay measurement, such as metastability in the shadow registers as the clock edge is moved, hence causing erroneous register values. Also, the impact of various random environmental and noise factors (e.g. clock jitter) may cause different readings of the delay values. Therefore, based on the precision requirements of the application, multiple measurement runs may be needed to get correct and precise delay values. However, these variation effects are typically not large and can be ignored if they are not larger than the skew step resolution.

B. Implementation

The Xilinx Virtex-II test-bed was programmed with designs constructed in VHDL. The generation of the clock skew step and the dynamic phase-shifting function is provided by the Digital Clock Manager (DCM) module on the FPGA board. The DCM produces an output clock signal that can be dynamically phase-shifted a set amount beyond the input clock signal, which enables us to keep increasing the negative skewed phase of the output clock signal step-by-step. The time resolution for each skew step depends on both the characteristics of the board and the period of the input clock, and as mentioned above, the minimum skew step in our experiments is 160ps.

To determine the actual delay of a path, we record the total number of skew steps that the attached shadow register has experienced before the corresponding result bit is set to '1' for the first time. Given the resolution of a single skew step, we know the total negative phase shift for that shadow register. Hence, we can determine the delay by simply subtracting this phase from the total system clock period.

For a large number of circuit paths to be measured, the result bits are scanned out using a scan chain to minimize I/O pin overhead. Therefore, the total time required for one delay measuring process is $s*(c+p)$, where s is the number of skew steps per process (how large the phase shift must be to make all paths fail), c is the number of cycles spent at each skew step (how many input data combinations must be run to be reasonably sure a path that should fail at a particular skew step does fail), and p is the number of paths being measured (how many clock cycles are necessary to scan out the result bits after each skew step). These are all designer- or user-defined parameters, and all of the experiments that we performed ran for several hundred cycles per measurement. Since these tests are performed at the IC's operating frequency, this is a short amount of time for IC authentication and test-time HTH detection, especially since these tests are performed in parallel with normal circuit operation. For run-time HTH detection, the path delays can be re-measured every $s*(c+p)$ clock cycles, although a fast-acting HTH sprung during operation may be able to avoid detection long enough to cause the desired damage.

C. On-die temperature monitoring

One issue that arises when non-functional characteristics are used for IC authentication is that many such characteristics, including delay and power, are dependent on temperature and voltage variations. It has been shown that the delays of PUF path pairs vary proportionally to each other when temperature changes and when the voltage on a shared rail changes.

However, in the method detailed in this paper, the responses are generated by direct delay measurements of a large number of internal functional paths of the core circuit, rather than from PUF pairs. Therefore, voltage and temperature will impact the measurements and must be compensated for in challenge-response scenarios. Voltage variations are not a major issue, as they are transients that unlikely to be reoccur if a challenge is repeated (similarly, clock jitter proved to be a non-factor in on-chip experiments), but temperature will always be an issue.

To overcome this problem, this method incorporates on-die temperature monitoring [6][7][11], which uses a ring oscillator as the clock input of a counter. As shown in Fig. 3, the oscillator is embedded within the main circuitry, and its switching frequency is temperature-dependent. At the end of a delay measurement, the counter is scanned out with the delay-based signature from the result bits, and the counter value is used as a proxy for the temperature. The temperature-delay relationship is well-known, so the challenge-response pairs that are generated pre-deployment do not need to include every possible operating temperature, as the authenticator can calculate the effective response from the reported temperature and delay signature.

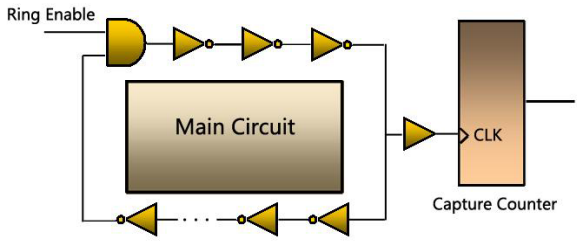

Figure 3. Ring-oscillator temperature monitor

This technique is both area- and power-efficient and has been shown to provide highly accurate temperature measurements, although multiple sensors may have to be used if there are large spatial temperature gradients. In addition, the incorporation of temperature into the IC authentication protocol greatly increases the number of challenge-response pairs and further confounds an attacker's ability to create a delay model from observed challenge-response pairs.

D. Die-area costs

As discussed in [5], the die-area costs of the testing circuitry can be categorized as one-time cost (fixed overhead regardless of number and size of paths) and per-path cost. One-time cost involves the circuitry controlling the *clk2* skew and the temperature monitoring circuitry. (Depending on the size of the circuit and the extent of the spatial temperature

gradients, additional temperature monitors may be necessary.) Per *n*-bit path cost involves one *n*-bit shadow register, one *n*-bit comparator (*n* 2-bit XOR gates and one *n*+1-bit OR gate), and one 1-bit flip-flop for the result bit. Table 1 shows the equivalent gate counts (as estimated by the Xilinx ISE 8.2i software) for these hardware costs with *n*=32.

The one-time DCM costs are fairly high, but most modern circuits have a large number of register-to-register paths, enabling these costs to be amortized over all of the paths. Based on the complexity of the circuit and the desired level of delay measurements, the per-path cost as a percentage can be large for certain circuits, especially when the size of the logic on a measured path is relatively small. However, the per-path costs can be managed by the designer in terms of how many paths are covered, how many bits are covered per-path, and if multiple paths (or sub-paths) can share the same testing circuitry, all of which also affect the uniqueness of extracted signatures for IC authentication and the sensitivity of HTH detection.

TABLE I. EQUIVALENT GATE COUNT FOR TESTING COMPONENTS

	Component	Equiv. Gates
One-time cost	DCM	6469
	Temp. Monitor	906
Per 32-bit path cost	Shadow Register	256
	Comparator	157
	Result Bit	8

IV. IC AUTHENTICATION

This section discusses how this delay characterization technique can be applied in IC authentication. As with PUF-based authentication techniques, the authenticator needs to keep a number of records of different challenge-response pairs for each individual IC. As shown in Fig. 4, each time an IC needs to be authenticated, the authenticator sends a challenge (usually a bit string) to the IC and waits for it to generate the response based on the measurements of its path delays. The response is generated by combining the measured delays of a number of circuit paths (which has been shown to be unique because of process variations) and the temperature reading from the ring-oscillator clocked counter. Therefore, the response can be regarded as the signature for that specific IC. After the authenticator gets the response from the IC, it will compare this response to the one in its record (a large number of challenge-response pairs must be collected pre-deployment). If they are the same, then the IC is authenticated, otherwise it will be suspected.

Fig. 5 shows the process of IC signature extraction. First, an arbitrary number of paths inside the circuit are chosen for the generation of the challenge-response pairs. Then *a pair* of challenge vectors is sent to the controller as the data input of the source register of those selected paths. The first vector is used to set the initial value of the source register, and the other vector is used to load into the source register at the next cycle. We keep applying this pair of data patterns and using the approach in Section III to measure the delay for this specific data to be propagated through the path. Since the delay of the paths will be different based on the data going through it, we will get different delays for different pairs of input data vector (i.e. different responses to different challenges). After performing the challenge at one skew step, the result bits are scanned out to determine which paths failed for the first time at that step, and the code for that step composes the signature portion associated with that path.

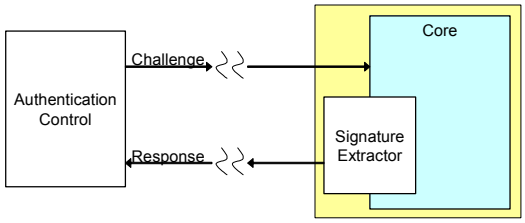

Figure 4. IC authentication

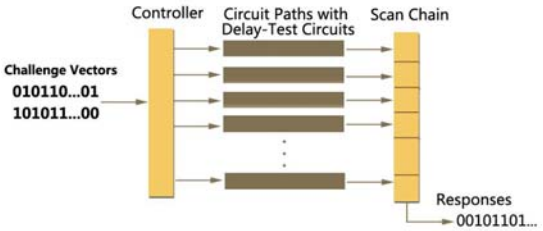

Figure 5. IC signature generation

When this technique is applied to a large number of paths and a high skew step resolution is used, the generated signatures can be significantly longer than those generated by PUFs. For example, if an 8-bit binary string is used to represent the path delay in number of skew steps and a specific path experienced 128 skew steps before the shadow register stores the wrong value, then the signature portion associated with that path is 10000000. The actual time-period of each skew step indicates the precision of this delay measuring technique, which can be as small as tens of picoseconds. Given this discrete representation of the path delay, there is still a possibility that the delays of the same circuit path on two different ICs will fall into the same binary representation, but experiments have shown that variations are high enough that unique signatures will be generated if a large number of paths contribute to the signature.

While experiments have achieved reliable and consistent challenge-response pairs at various temperatures, this signature extraction technique is susceptible to noise. The IC authentication protocol must therefore be designed with this in mind, either defining a range of acceptable responses to challenges or performing a limited number of additional challenges if the first response is not exactly correct.

V. HTH DETECTION

The delay characterization technique discussed in Section III can also be used in HTH detection, in that it can retrieve the statistical distribution of the measured delays for every circuit path over a large number of ICs. When the delay measurement circuit is embedded at design-time, the designer can create a statistical delay model (including projected

manufacturing variations) for all the paths in the circuit. At test-time, the designer can invoke the embedded delay measurement mechanism to create a statistical distribution of delays for every path. These test-time measurements are then compared to the design-time projections, and any substantial statistical difference creates a suspicion of design alteration. Even small design alterations can measurably impact path delays, and implanted HTHs affect the delays of neighboring circuitry due to crosstalk effects. While a difference in delay statistics may be a result of another factor, it may provide cause to perform more rigorous functional tests or even perform a reverse engineering analysis on a sample of ICs.

While this technique provides HTH capabilities to complement functional tests, dormant HTHs may still be difficult to detect at test-time. Techniques for triggering dormant HTHs at test-time by choosing an optimal set of test vectors have been developed [12], but a backup method for detection of HTHs sprung at run-time may still be required. Therefore, all deployed ICs are first characterized under normal operating conditions, and delay characteristics will be monitored continuously in the field, further emphasizing the benefits of this at-speed delay measurement technique not affecting the main circuit functionality. When an HTH is sprung, these delays will change, enabling detection and ultimately containment even when a functional monitor does not detect anomalous behavior. However, as discussed in Section III-B, if a well-hidden HTH acts fast when sprung at run time (e.g., it executes for only several clock cycles to perform a specific action and then goes dormant again), then this brief alteration of circuit path delays will most likely go undetected by our approach.

One of the key considerations for both test-time and run-time HTH detection is the desired level of sensitivity for path delay alteration, as being too sensitive could lead to false positives and being too insensitive could lead to false negatives. Therefore, the sensitivity must be set based on the specific application and on how this delay measurement technique is used in conjunction with other HTH detection techniques.

VI. Experimental Evaluation

We have evaluated the feasibility of this signature extraction mechanism by implementing a 32-bit floating point adder on an FPGA. The adder is a 6-stage pipelined adder, and each stage was treated as a single register-to-register path for delay measurement. The original VHDL code for this circuit was downloaded from the OPENCORES website [12], and the delay testing and signature extraction circuitry was added to this code. The code was synthesized using the Xilinx ISE 8.2i software and then implemented in two (supposedly) identical Xilinx Virtex-II Pro FPGA boards.

To evaluate this signature generation mechanism, we first input a set of challenge pairs into the adder and then read the responses from the output. As described in Section IV, the tested circuit paths keep applying the same challenge pair to the input of its source register, and the skew is increased until the shadow path fails, at which point a 6-bit signature is generated for that path based on the current skew step. In our experiment, we applied this technique to three of the six pipeline stages and therefore generated an 18-bit delay signature.

Table II shows the delay responses (in hexadecimal) for three different challenge vector pairs (the temperature readings in each case were identical). As can be seen from the table, the responses of the two ICs for all three challenge pairs are all different, which is a reflection of the significant inter-die process variations. However, as the total number of ICs increases, the possibility that two have the same challenge-response pair will also increase. This problem can be overcome by increasing the total length of the signature (i.e. increase the number of paths for the measurements of path delay) or increase the number of input challenge pairs on the same path. In addition, we can further split each of the 32-bit paths into 32 single-bit paths and then measure the path delay for each single-bit path. In this way, we can generate a very long, unique response, even for relatively simple circuits.

TABLE II. Variability of challenge-response pairs

		Response	
Challenge		Chip 1	Chip 2
	1	23b2e	1fa2c
	2	27b3d	24a29
	3	2bcf6	27c75

We also investigated the impact of temperature changes (which were artificially induced) on the path delays using the technique discussed in Section III-C and observed delay variations for the same path under different temperature conditions. Table III shows the path delays for one sample path in the 32-bit adder circuit under different temperature conditions (indicated by the ring oscillator sensor frequency). The measured delays are in nanoseconds, computed by the methods introduced in Section III-B.

TABLE III. Temperature-dependent delay measurements

Temps.	Ring Oscillator Frequency (MHz)	Path Delay (ns)
1	29.141	7.3
2	28.382	7.6
3	28.138	7.7

VII. Summary and Future Work

This paper addressed the increasingly important problems of IC authentication and HTH detection through the use of an at-speed delay measurement technique. These delay measurements can be used to generate long, unique signatures for authentication purposes based on manufacturing variations, and the measurements can be performed on functional parts of the circuit. In addition, these delay measurements can be used to detect design alterations, including the presence of HTHs, at both test-time and run-time. While this approach does not guarantee HTH detection, it can be used in conjunction with other HTH detection mechanisms (none of which provide

100% detection capability) to improve the probability of detection while minimizing false positives.

These techniques were evaluated on FPGAs, but future work includes the design and fabrication of custom ICs to evaluate the authentication approach across a large number of physical ICs and to experiment with the test-time and run-time detectability of various design alterations. We will then be able to carry out more in-depth analyses and statistical modeling to evaluate and verify these approaches.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grants CNS-0716443 and IIS-0612049. The authors would like to thank the anonymous reviewers for their helpful suggestions.

REFERENCES

[1] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," IEEE Symposium on Security and Privacy, pp. 296-310, 2007.

[2] D. Ernst, et al., "Razor: circuit-level correction of timing errors for low-power operation," IEEE Micro, vol. 24, no. 6, pp. 10-20, Nov. 2004.

[3] B. Gassend, "Physical random functions," Master Thesis, February 2003.

[4] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," ACM Conference on Computer and Communications Security, pp. 148-160, 2002.

[5] J. Li and J. Lach, "Negative-skewed shadow registers for at-speed delay variation characterization," IEEE International Conference on Computer Design, pp. 354-359, 2007.

[6] S. Lopez-Beudo and E. Boemo, "Making visible the thermal behavior of embedded microprocessors on FPGAs – a progress report," ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 79-86, 2004.

[7] S. Lopez-Beudo, P. Pernas, and E. Boemo, "Thermal testing on reconfigurable computers," IEEE Design and Test of Computers, vol. 17, no. 1, pp. 84-91, Jan-Mar 2000.

[8] K. Nohl, D. Evans, Starbug, and H. Plotz, "Reverse-engineering a cryptographic RFID tag," 17th USENIX Security Symposium, in press, 2008.

[9] M. Saint-Laurent and M. Swaminathan, "A digitally adjustable resistor for path delay characterization in high-frequency microprocessors," Southwest Symposium on Mixed-Signal Design, pp. 61-64, 2001.

[10] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," Design Automation Conference, pp. 9-14, 2007.

[11] S. Velusamy, W. Huang, J. Lach, M. Stan, and K. Skadron, "Monitoring temperature in FPGA based SoCs," IEEE International Conference on Computer Design, pp. 634-640, 2005.

[12] F. Wolff, C. Papachristou, S. Bhunia, and R. Chakraborty, "Towards trojan-free trusted ICs: problem analysis and detection scheme," pp. 1362-1365, Design, Automation and Test in Europe, 2008.

[13] www.opencores.com