Differential Public Physically Unclonable Functions: Architecture and Applications

Miodrag Potkonjak, Saro Meguerdichian, Ani Nahapetian, and Sheng Wei
Computer Science Department
University of California, Los Angeles
{miodrag, saro, ani, shengwei}@cs.ucla.edu

ABSTRACT

We have developed an ultra low power (well below 1 nanojoule per transaction), ultra high speed (less than 1 nanosecond), and low cost (a few hundred gates) public physically unclonable function (PPUF). We have also developed the first PPUF-based smart card (SC). We analyze and demonstrate the security of this new SC against several families of potential security attacks.

Categories and Subject Descriptors

B.7 [**Hardware**]: Integrated Circuits

General Terms

Design, Security

Keywords

Authentication, hardware security, PUF, PPUF, smart cards

1. INTRODUCTION

We present a practical zero-knowledge proof authentication technology, which is independent of traditional cryptography techniques. The underlying idea behind the technology is a circuit, whose execution takes microseconds or less, but whose simulation for a given input vector takes at least seconds or minutes or more. In our technology, instead of storing authentication information on an IC, authentication is made possible by the speed with which timing information about a chip can be determined. An attacker without access to the physical IC will not be able to determine the requested information in the given time period, even given a complete characterization of the circuit and fabrication and design information.

The integrated circuit authentication technique that we propose leverages process variation (PV). PV causes gates designed to be identical to be different in terms of structural and operational properties, such as timing delay and power consumption. This is a result of the manufacturing process, given the intense feature scaling of industrial CMOS technology. By leveraging PV among identically designed gates and ICs, we can develop a delay-based authentication technique that helps us distinguish between unique ICs and authenticate them. Many ICs with the same digital key stored in them can exist; however, it is nearly impossible to manufacture two ICs with the same delay characteristics, since reducing PV is still one of the most difficult technical challenges of technology scaling. This protocol can be modeled as public key encryption, where the underlying one-way function is the ability of the authentic circuit to answer the query in the very limited time given. Simulation of the circuit will be very time-consuming and power hungry, and as a result an attacker will be unable to simulate the answer in time.

A significant application of such a circuit is as a new, more secure, and more time-efficient smart card. A smart card (SC) is an embedded system in card-like packaging, with an IC containing a microprocessor and non-volatile memory. In contrast to more pervasive passive cards (such as memory cards or the very common magnetic stripe cards) that can only store information, the smart card is an active device that is capable of processing data and making decisions. This capability to record and process information in its own non-volatile, physically protected memory, compounded with cryptographic protocols that protect the exchange of rights between the SC and the accepting computer, makes the smart card a powerful and convenient financial and digital rights management tool. Other applications include trusted remote sensing [1]

2. RELATED WORK

The primary basis for our approach is inherent PV and gate-level uniqueness in modern and future silicon CMOS technologies [2]. There are several difficult technological problems that preclude fabrication of ICs with the exact feature sizes of gates and wires and levels of doping. They include wafer lattice structure imperfections, non-uniform dopand distribution, mask alignment, and chemical / mechanical polishing [2].

PV exists among gates when gates across ICs are designed to be identical, but due to manufacturing limitations are different and unique in terms of structural and operational properties, such as timing delay and power consumption. As transistors have shrunk in size, the percentage difference in this property has grown. For example, identical gates may have up to 30% difference in timing delays in current 45 nm technology [3]. Furthermore, variability can be increased

DAC 2011, June 5-10, 2011, San Diego, California, USA.

using intentional random doping and through light exposure. By leveraging PV between identically designed gates and ICs, we can develop a delay-based authentication technique that authenticates ICs, taking advantage of the near impossibility to manufacture two ICs with the same delay characteristics.

A physically one-way function [4][5][6], whose silicon implementation is better known as a physically unclonable function (PUF), is a different type of hardware authentication approach. A PUF is similarly a complex phenomenon and practically impossible to clone. It is inexpensive, admitting no compact mathematical representation, and intrinsically resilient to attacks that aim to extract the digital key [7]. Gassend et al. [4] introduced delay-based silicon PUFs that leverage the intrinsic PV of deep submicron technologies, where a set of challenge-response pairs are used to perform authentication tasks. However, the PUF was initially secret key cryptography with a limited range of applications and the need for storing pairs of challenges-response vectors. Not only does this increase the cost of operation and reduce the number of security protocols that can be built using PUFs, but it may also be exploited as a security vulnerability.

Recently, Beckmann and Potkonjak proposed the first public PUF (PPUF) structure and used it to construct secure protocols for the exchange of secret keys and for public-key cryptographic communication [8]. The essential idea is that the owner of a PPUF publishes its gate-level characteristics, such as delay or leakage energy, which serve as its public key. Therefore, anybody can simulate a small number of input vector challenges using significant run times, but only the owner of the PUF is capable of executing billions or more inputs in any reasonable amount of time. Their approach exploits signal glitching and requires ultra accurate, ultra high frequency clocks and long execution times, at least in the range of hours. Our approach both eliminates the need for accurate clocking and reduces the execution time by more than orders of magnitude.

Our authentication technique offers distinct advantages over the previous research, specifically by requiring no prior storage of challenge-response pairs and by carrying out the low power authentication process in extremely small amounts of time. There is no secret hidden in the IC itself, nor in the authenticating party's database. The "secret," the gate delay characteristics, is publicly known and made available; however, only with the physical IC can authentication be done successfully. Thus, it is not susceptible to possible attacks, including side channel attacks, man-in-the-middle attacks, and cloning. One important way that our new architecture differs from the previous work is that our equivalent of the physical one-way function [9] can be public.

3. PRELIMINARIES

The delays of gates in submicron technologies are subject to significant PV. A spectrum of manufacturing factors that include unavoidable physical and chemical phenomena, such as silicon lattice imperfections, unequal number of dopants, imperfect mask alignment, and non-uniform chemical mechanical polishing, result in significant gate delay variability. Intel [3] reported that already in 180 nanometer CMOS logic families the variation in gate delay on different ICs was up to 30% from the nominal value. This technology is rather old today; thus, we used it in order to conduct a very conservative and pessimistic evaluation of our approach. The delay variation in modern technologies such as 32 nm is significantly (around 3 times, according to IBM and Cadence data) larger.

To authenticate the SC, the authenticator must determine the proper circuit output. The authenticator accomplishes this by simulating the SC output using the derived gate-level characteristics. Gate-level characterization (GLC) aims to recover post-silicon gate-level properties of ICs, which are unique due to the presence of PV [10][11][12][13][14][15]. GLC calculates the relevant characteristics of each gate using a limited number of nondestructive measurements of the combinatorial SC circuit.

Utilizing a linear programming (LP) formulation, we are able to properly characterize the sizes of the gates, and hence their timing characteristics. Each constraint involves knowledge of the circuit architecture and the timing, switching power, and/or leakage power measured for a pair of input vectors over a specific clocking period.

4. DIFFERENTIAL PPUF-BASED SMART CARD

Beckmann's PPUF leverages the frequent glitching of XOR gates and requires a very high clock resolution. In addition, their public-key PPUF-based cryptographic protocol requires processing of a huge number (e.g. many billions) of input vectors. Finally, their protocol is restricted by the execution of remote exchanges of secret keys or transfer of private messages using public key cryptography frameworks. In order to overcome these limitations, we have created several new concepts.

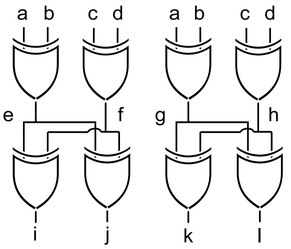
4.1 New Security Primitive

First, we have completely eliminated the need for ultra accurate clock manipulation by developing the first differential PPUF (dPPUF) architecture. In a dPPUF, two signals race along two or more paths that are nominally equal but unique due to PV. Furthermore, we have developed a different way of using PPUFs for public-key authentication that requires processing of at most two input vectors, reducing by several orders of magnitude both the execution time and required energy.

Probably the best way to explain the new dPPUF architecture is to consider its ultra small realization shown in Figure 1. Each side of the circuit contains 4 XOR gates arranged into 2 levels, where the inputs of the second level come from the outputs of the first. If the input switches from 0000 → 0101 at time $t = 0$: output i will switch at times $t = 6, 11$; j at $t = 9, 12$; k at $t = 8, 10$; and l at $t = 7, 12$. Therefore, the left side will be faster for the left output (i switches before k), while the right side will be faster for the right (l switches before j). However, we can see that if the input were to instead switch from 0000 → 1000, the results would be reversed.

4.2 Authentication Protocol

dPPUF-based secure authentication leverages the time gap between the execution of two consecutive challenge input vectors on an IC and the corresponding simulation time. Delays of logical gates in modern technologies are in the picosecond range. Thus, the delay of a chain of a few dozen gates is well bellow one nanosecond. For example, much more complex modern processors have clock rates of a few



δ_{ae}	3	δ_{ei}	7
δ_{be}	4	δ_{fi}	2
δ_{cf}	5	δ_{ej}	5
δ_{df}	4	δ_{fj}	8
δ_{ag}	4	δ_{gk}	5
δ_{bg}	5	δ_{hk}	4
δ_{ch}	8	δ_{gl}	7
δ_{dh}	4	δ_{hl}	3

Figure 1: Small dPPUF example with left and right PPUFs, where δ_{ij} denotes delay from input i to output j of a gate. Different outputs are faster for different inputs.

GHz; if we assume that there are h levels of gates in the dPPUF, it is easy to see that the required number of operations for simulation is an exponential function of h. Therefore, it is easy to make the simulation time excessive even with a very limited number of gate levels.

The new SC operates in the following three phases: (i) dPPUF characterization; (ii) challenge and response; and (iii) verification. The first phase is conducted before the pertinent service institution issues any challenge to a customer SC. In order to enable public key-based authentication, we assume that the complete architecture and the relevant manifestational characteristics of each gate, such as its delay, leakage power, or switching power, are publicly available. Specifically, we focus on delay characteristics, because they intrinsically require exponential simulation run time without any additional measurement or non-linear circuitry. Note that there are multiple ways to create both PPUFs and dPPUFs that use leakage power as manifestational phenomenon with exponential run times.

5. DIFFERENTIAL PPUF ARCHITECTURE

In this section, we introduce the first dPPUF architecture. We start by identifying and analyzing the design, operational, and security desiderata and their ramifications on PPUF architectures. Then, we propose and analyze a booster-represser-based dPPUF architecture.

The PPUF desiderata include: (i) conceptually clear and simple, and technologically inexpensive, implementation; (ii) ultra low power, low area, and very high speed PPUF; (iii) stability against operational and environmental conditions; (iv) suitability for inexpensive, in-field, and accurate characterization; (v) resiliency against security attacks; (vi) large, flexible, and controllable simulation time; (vii) scalability; and (viii) low input/output (IO) requirements.

5.1 Ramifications

1) Differential PPUF architecture. As in initial PUFs, we use as our basic mechanism the race between signal paths. There are two major conceptual novelties. First, signal paths are formed using the standard logical case where the challenge input is the input vector. The second innovation is that instead of having a race between two paths, we have a race between two sets with an exponential number of paths. Therefore, the need for accurate clock capturing of glitch temporal characteristics is eliminated; we use only the frontier signals. Because some frontier signals do not cause transitions, any emulation effort will require exponential time.

2) Fast and low power NAND and XOR gates. We use only NAND and XOR gates. XOR gates are slower and less energy efficient than NAND gates, but it can be easily shown that the XOR gate is the most efficient realization of a boosting cell with maximum efficiency (Section 5.2). We also use only local interconnect that connects only the cells between neighboring layers of gates.

3) Interleaved layout. There is a very minimal difference in operational and environmental conditions for devices that are physically close. Therefore, to take advantage of this simple but powerful observation, we place corresponding gates of the left and right PPUFs next to each other.

4) Multiple outputs at several layers of gates. We measure delays of gates using clock sweep techniques [16][17]. Our architecture is such that we can easily form an exponentially large system of equations that are not collinear. Still, to further improve the accuracy of gate delay characterization, we store some or all intermediate signals into flip-flops (FFs). Note that while this step would completely break the security of traditional PUFs, the security of the procedure is not compromised because this data is anyhow public; protocol security is based on the difficulty of quick enough simulation.

5) High multiplicity of racing paths. Traditional PUFs have two racing paths. They employ Shannon's paradigm of input diffusion and nonlinearities to preserve unpredictability. Our dPPUF relies only on the difficulty of simulation. In order to ensure unpredictability, we use partly randomized and partly maximally interacting interconnect networks.

6) Booster-represser cells. The simulation time of any network that is formed using 2-input gates is bounded on the lower side by the number of gates and on the upper side by an exponential function of type $O(2^n)$. However, the key observation is that one has just to simulate the frontier of propagated signals. Therefore, architectures where one can predict which frontier signals will not cause transitions are not secure. We use represser cells to terminate in an unpredictable way a subset of propagating signals; we use booster cells to ensure maintenance of a large set of candidates for the frontier.

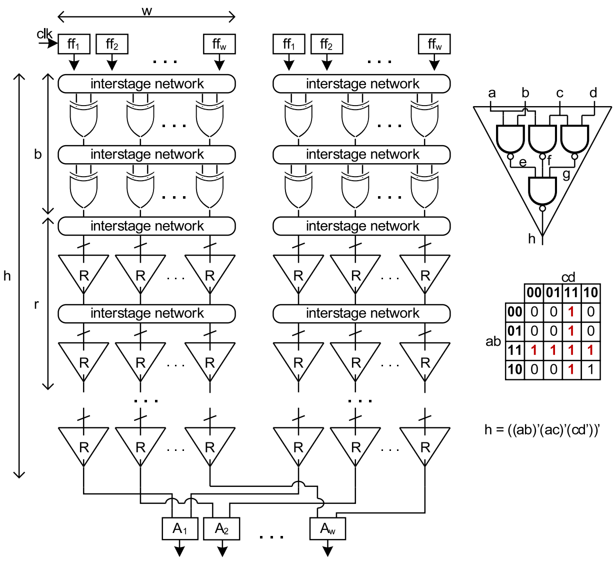
7) Linear size increase to exponential simulation time increase. Again, interleaved booster and represser gate cells directly provide the solution to this objective.

8) Partially randomized input vector overwriting and local random input generation. The essential limiting resource in all modern systems is IC I/O. If we enter a significant part of the input vector before it is completed, the attacker gains valuable additional time for computation. In order to enable the practical and inexpensive resolution of this problem, we propose two solutions: (i) overwrite a randomly selected subset of inputs or (ii) generate the complete input vector locally using a random number generator.

5.2 Architecture

Following the specified desiderata, we have developed the dPPUF architecture shown in Figure 2. The design has two nominally identical components: left and right. However, due to PV, the delays of each gate are different.

Each component receives its inputs from the same or different sets of FFs with cardinality w and consist of h stages of interstage networks and sets of gate-based cells. Finally, the left and the right components provide inputs to w arbiters that produce the final vector output. The signal that first arrives, from either the left or right arbiter, locks it to a particular value, either 0 or 1.



	cd			
ab	**00**	**01**	**11**	**10**
00	0	0	1	0
01	0	0	1	0
11	1	1	1	1
10	0	0	1	1

h = ((ab)'(ac)'(cd'))'

Figure 2: dPPUF architecture of height h and width w. A challenge vector propagates through repeated layers of b booster (XOR) and r represser (R) cells (right), with partly maximally interacting and partly random interstage networks. Signals from 2 nominally identical but physically unique PPUFs race to arbiters (A_i) to produce the response.

We have two types of gate cells: boosters and repressers. The role of the booster cell is to increase the frequency of output switching with respect to the frequency of input switching. Note that for a two input gate, the ratio of these frequencies is at most 2. Represser cells nominally have the opposite role, i.e. to reduce the amount of switching. More precisely, the goal is eliminate a certain number of initial frontiers of the propagation signals in order to make simulation more time-demanding.

In our realization, we use a 2-input XOR gates as a booster. The XOR gate has a boosting factor of 2 because each switching of any input results in the switching of the output. As a represser cell, we use the small NAND-based circuit shown in Figure 2. Even though this cell's Karnaugh map has an equal number of 0s and 1s, the cell's output is less likely to switch for any given input switch because the 0s and 1s are grouped together and not interspersed (as in the Karnaugh map of the XOR gate). Therefore, by using a different number of booster and represser layers of gates, we can control the switching and hence the computational complexity of simulation.

6. PROBABILISTIC VERIFICATION

In the straightforward implementation, the verifier just simulates the PPUF. There are four main advantages of the verifier over any attacker:

(i) There is a sharp difference in the available computation time. While the attacker has only one clock cycle to provide the answer, the verifier has a much longer time. This difference is in particular high (several orders of magnitude) if the verifier uses pre-computation.

(ii) The attacker can choose to verify only a subset of the outputs. A simple, but important, observation is that the verifier does not have to simulate and verify all outputs. Specifically, if we assume that the likelihood of 0 or 1 for any output is $p = 1/2$, in order to have proof of strength (probability of coincidence) s, we need to simulate $n = \log(p/s)$ inputs. For example, for an ultra small likelihood of coincidence of 10^{-30}, we need to simulate 100 outputs.

(iii) The verifier may select to request answers to a number of challenges and verify only a subset of the responses. This approach linearly increases the required time and energy but exponentially decreases the probability that the attacker will guess the correct answer.

(iv) The verifier can keep subsets of the inputs common to two or more challenges. The benefit is due to the reduced simulation time. Since the attacker can neither guess nor quickly identify which parts of the previous input vectors are reused, he cannot take advantage of this approach.

7. SECURITY ATTACKS AND EVALUATION

In this section, we identify three classes of potential security attacks: (i) guessing (statistical analysis); (ii) technological; and (iii) protocol. In guessing attacks, the attacker observes a polynomial number of challenge-response pairs and tries to statistically analyze them in order to predict the answer to an unseen challenge. It is important to observe that in this case, there is no real benefit to selecting a training set for statistical modeling. Technological attacks are mainly based on cloning, emulation, side channel, and physical attacks. Finally, protocol attacks try to explore vulnerabilities in policies for the exchange of data between the SC and the verifier.

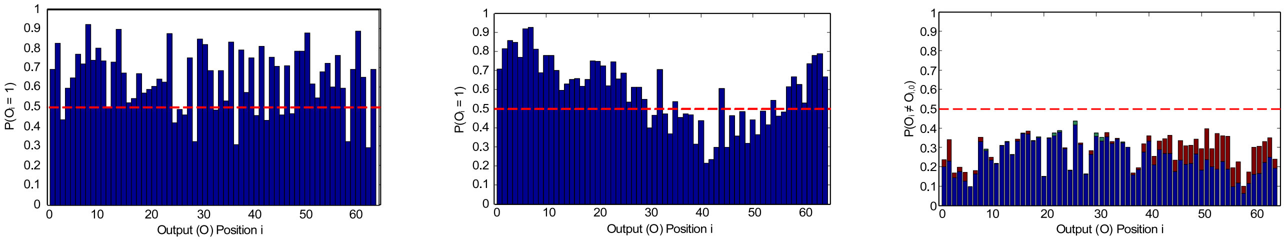
For statistical attacks, we conducted comprehensive simulations using a very small card with $h = 10$, $w = 64$, $b = 1$, and $r = 1$. We present the results for each simulation obtained using 10,000 input vectors.

7.1 Guessing (Statistical Analysis) Attacks

1) A priori guessing. The attacker tries to guess each output O_i by observing a set of previous trials for a given card. Specifically, his goal is predict $P(O_i = c)$, where $c = 0$ or 1. Figure 3a shows the results of simulation for one representative chip with many inputs. The attacker may also have access to other instances of the SC, or knowledge of how they responded to the same challenge. In Figure 3b, we show $P(O_i = 1)$ for one representative input over many chip instances. Outputs for which this metric is very close to either 0 or 1 are highly predictable and thus suboptimal; ideally, output should be 0 or 1 with equal likelihood. However, the verifier can choose to only verify those inputs which are most unpredictable. We can see that even for a small circuit there are many such inputs.

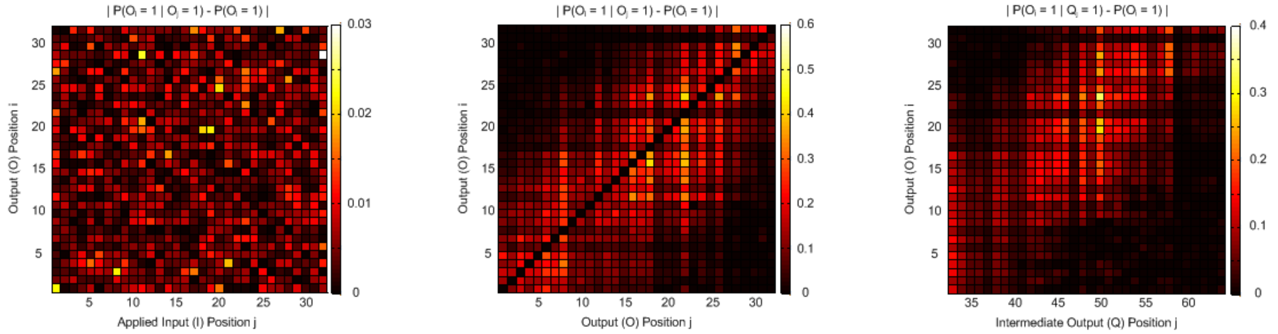
2) Conditional input- and output-based guessing. We can define four relevant metrics of the type $P(O_i = 1|I_j = 1)$, where I is the input vector, and four of the type $P(O_i = 1|O_j = 1)$. In order to maximize readability, we show $P(O_i = 1|I_j = 1) - P(O_i = 1)$ in Figure 4a and $P(O_i = 1|O_j = 1) - P(O_i = 1)$ in Figure 4b. Darker values are ideal, implying low correlation; note that for input-based guessing (Figure 4a) we observe essentially 0 correlation. There is moderate correlation between only some output pairs (Figure 4b), up to 60%.

3) Conditional intermediate output-based guessing. An effective attack on traditional PUFs is to simulate a small part of the PUF and use this result to guess the final output. However, our dPPUFs employ an exponential number of paths and are therefore not susceptible to this attack.



(a) (b) (c)

Figure 3: For $w = 64$ and $h = 10$, probability that an output bit will equal 1 (a) for one SC and (b) for different instances of the SC given the same input; (c) SAC, the probability that an output bit will switch for inputs of hamming distances 1 (blue) and 2 (increase: red and decrease: green). The red dashed lines depict the ideal case, where $P(O_i = 1) = 0.5$ for all i.



(a) (b) (c)

Figure 4: Correlation between output bits O_i and (a) input bits I_j, (b) other output bits O_j, and (c) intermediate output bits Q_j. Darker values are ideal, implying low correlation. For readability, correlation between an output bit and itself is set to 0.

To statistically prove this claim, we studied the probability $P(O_i = 1|Q_j = 1)$. In order to provide an upper bound on predictability from the intermediate results, we used the intermediate outputs Q only one layer before the actual outputs O. As in the previous subsections, Figure 4c plots $P(O_i = 1|Q_j = 1) - P(O_i = 1)$.

4) Strict avalanche criterion (SAC). The SAC examines the correlation probability of the corresponding outputs of two input vectors that differ at exactly one position. Ideally, the probability of each output switching should be 1/2. In Figure 3c, we show dPPUF SAC for input hamming distances 1 (blue) and 2 (red/green). The red bars show an increase in probability toward the ideal 0.5, while the green bars show a decrease.

7.2 Technological and Protocol Attacks

1) Simulation attack. Figure 5 shows the simulation effort with exponential scale on a 1 GHz processor for a dPPUF with width $w = 64$ for various heights. Obviously, the growth is exponential. This does not necessarily render our approach unscalable for the verifier; recall that the verifier has the option of only verifying a subset of the outputs, so we can increase w to increase the simulation effort for the attacker while maintaining that of the verifier.

2) Storage of intermediate results. Input vectors very often share some subset of identical bits; thus, an attacker can attempt to store intermediate simulation results from previous input vectors to speed up later computation. Dynamic programming and other such techniques would not scale, since an exponential number of intermediate results would require an exponential amount of storage. Additionally, such an attack could be defeated by hashing the input vector before applying it to the dPPUF. For example, a large number of input bits could be shuffled in some random order such that only a small, unpredictable subset of them is applied to the dPPUF.

3) Special purpose hardware. Another type of attack is to use a very fast processor, perhaps a FPGA or ASIC, to simulate the PPUF. The key thing to note here is that it is easy to create an exponentially larger PPUF (in terms of computation effort) by just adding a few levels of gates; creating faster hardware simulation using FPGAs or ASICs, however, cannot be exponential.

4) Side channel attack. Side channel attacks are not a threat, since the power profile of a gate would not reveal any new information. GLC results are available as a public key.

5) Man-in-the-middle attack. Recall that the total amount of time given to the attacker is one or at worst two clock cycles, where he learns the challenge after the first cycle. Thus, man-in-the-middle attacks are also not possible, because there is not enough time for an attacker to transmit and receive data in time, or to otherwise communicate with any entity beyond the inserted SC.

6) Look-up table (LUT). The creation of a LUT of all possible challenge-response pairs is not feasible due to the exhausting number of pairs that need to be enumerated.

7) Cloning attack. A cloning attack, which would fabricate an identical SC, would not be possible due to PV. A counterfeit SC would not have the same gate characteristics; signals would race at different rates and produce different output vectors.

8) Old technology and FPGA emulation attacks. An especially pernicious attack is the old technology attack, where the attacker would create an IC with gates with a factor

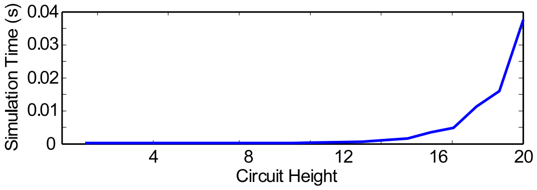


Figure 5: Exponential growth of simulation time required for a 1 GHz processor to simulate a dPPUF with $w = 64$.

larger timing delay, but with the same relative timing characteristics, using older technology that is not as susceptible to PV. The signals would race at the same relative rates as the authentic SC. Alternatively, the attacker could create a FPGA to emulate the SC. Such attacks are infeasible because intrinsic gate delay is not the only consideration. The delay of a gate is also affected by its logical effort, or driving capability, and its electrical effort, or gain, which is dependent upon the gates that it drives; the ratio between these components is very different for different technologies.

8. SENSITIVITY TO AGING, TEMPERATURE, AND VOLTAGE

In order for a SC to be effective under a variety of operational and environmental conditions, it must be resilient to variations in aging, temperature, and supply voltage. Effects such as negative-bias temperature instability (NBTI) can cause a gate to age, increasing its delay by up to 10% and thus changing the PPUF characteristics dynamically [18][19]. We can combat this effect by aging the SC to its maximum degree upon issuing it, raising permanently each gate's delay by 10%.

Delay of a gate is also proportional to temperature, which can vary dramatically even across the IC. Nevertheless, we can heat up the circuit by issuing several challenge vectors (thus causing its gates to switch and generate heat), then check the consistency of its response vectors to its increase in temperature at each iteration. We can adopt a similar approach to overcome variations in supply voltage by applying several voltages and checking consistency.

Ultimately, however, the more sophisticated approach to guaranteeing resilience to any operational or environmental variations is to re-characterize all gates using GLC before and after each usage, to obtain the current PPUF characteristics and ensure that they did not change significantly during use.

9. CONCLUSION

We have introduced the differential public physically unclonable function (dPPUF), built using only standard logic gates. The dPPUF eliminates the need for an ultra accurate clock and enables essentially zero knowledge authentication, where previous usage provides negligible help to potential attackers. Authentication is conducted in less than 1 nanosecond using less than 1 nanojoule. Compounded with a security protocol that exploits the huge gap of more than ten orders of magnitude between the execution time for a random input vector of the dPPUF (less than 1 nanosecond) and the corresponding simulation time, the dPPUF is the basis for a new generation of smart cards. We have demonstrated exceptionally high security of the smart card even for a very modest dPPUF size of 1,600 gates placed in 10 logic layers.

10. ACKNOWLEDGMENTS

This work was supported in part by the NSF under awards CNS-0958369 and CNS-1059435.

11. REFERENCES

[1] M. Potkonjak et al., "Trusted sensors and remote sensing", *IEEE Sensors*, pp. 1104–1107, 2010.

[2] K. Bernstein et al., "High-performance CMOS variability in the 65-nm regime and beyond," *IBM Research and Development*, vol. 50, nos. 4–5, pp. 433–450, 2006.

[3] S. Borkar et al., "Parameter variations and impact on circuits and microarchitecture," *DAC*, pp. 338–342, 2003.

[4] B. Gassend et al., "Silicon physical random functions," *ACM CCS*, pp. 148–160, 2002.

[5] J. Guajardo et al., "Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions," *ISF*, vol. 11, no. 1, pp. 19–41, 2009.

[6] K. Kursawe et al., "Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage," *HOST*, pp. 22–29, 2009.

[7] R. Anderson et al., "Cryptographic processors-a survey," *IEEE Proceedings*, vol. 94, no. 2, pp. 357–369, 2006.

[8] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically Unclonable Functions," *Info. Hiding*, pp. 206–220, 2009.

[9] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," *DAC*, pp. 9–14, 2007.

[10] S. Wei et al., "Gate-level characterization: foundations and hardware security applications," *ACM/IEEE DAC*, pp. 222-227, 2010.

[11] S. Wei and M. Potkonjak, "Scalable segmentation-based malicious circuitry detection and diagnosis, *ICCAD*, pp. 483–486, 2010.

[12] S. Wei et al., "Malicious Circuitry Detection Using Thermal Conditioning," *IEEE TIFS*, 2011.

[13] S. Wei and M. Potkonjak, "Scalable hardware Trojan diagnosis," *IEEE Transactions on VLSI Systems*, 2011.

[14] S. Wei and M. Potkonjak, "Integrated circuit security techniques using variable supply voltage," *ACM/IEEE DAC*, 2011.

[15] M. Potkonjak, et al., "Hardware Trojan horse detection using gate-level characterization," *DAC*, pp. 688–693, 2009.

[16] J. S. J. Wong, P. Sedcole, and P. Y. K. Cheung, "Self-measurement of combinatorial circuit delays in FPGAs," *TRETS*, v. 2, n. 2, pp. 1–22, 2009.

[17] Y. Kim, et al., "Low-cost gate-oxide early-life failure detection in robust systems," *VLSI Circuits Symposium*, pp. 1–2, 2010.

[18] M. Agarwal, B. Paul, M. Zhang, and S. Mitra, "Circuit failure prediction and its application to transistor aging," *VTS*, pp. 277–286, 2007.

[19] S. Meguerdichian and M. Potkonjak, "Device aging-based physically unclonable functions," *ACM/IEEE DAC*, 2011.