DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication

Fatemeh Tehranipoor, *Student Member, IEEE*, Nima Karimian, *Student Member, IEEE*,
Wei Yan, *Student Member, IEEE*, and John A. Chandy, *Senior Member, IEEE*

***Abstract*— A physically unclonable function (PUF) is an irreversible probabilistic function that produces a random bit string. It is simple to implement but hard to predict and emulate. PUFs have been widely proposed as security primitives to provide device identification and authentication. In this paper, we propose a novel dynamic-memory-based PUF [dynamic RAM PUF (DRAM PUF)] for the authentication of electronic hardware systems. The DRAM PUF relies on the fact that the capacitor in the DRAM initializes to random values at startup time. Most PUF designs require custom circuits to convert unique analog characteristics into digital bits, but using our method, no extra circuitry is required to achieve a reliable 128-bit PUF. The results show that the proposed DRAM PUF provides a large number of input patterns (challenges) compared with other memory-based PUF circuits such as static RAM PUFs. Our DRAM PUFs provide highly unique PUFs with a 0.4937 average interdie Hamming distance. We also propose an enrollment algorithm to achieve highly reliable results to generate PUF identifications for system-level security. This algorithm has been validated on real DRAMs with an experimental setup to test different operating conditions.**

***Index Terms*— Dynamic RAM (DRAM), ID extraction, physically unclonable function (PUF), randomness, reliability, system-level security, uniqueness.**

I. Introduction and Motivation

In RECENT years, security has grown into a critical issue in modern information systems. Electronic hardware security, in particular, has emerged as one of the most serious challenges due to electronic devices penetrating every aspect of our society. Due to globalization trends, intellectual property (IP) vendors and system integrators have to deal with various counterfeiting issues more than ever and this surge in counterfeit hardware has driven the need for more secure chip authentication. Among the sources of counterfeit chips are reintroduced discarded chips into the supply chain and fabrication of cheap copies that pass as authentic without significant scrutiny. Since the IP owner cannot be present during the fabrication process, this makes integrated circuit (IC) designs increasingly vulnerable to malicious modifications [1]. As a means to uniquely identify chips, researchers have proposed using the random process variations (PVs) that naturally occur during the manufacturing process. These effects include PVs such as the size of transistors, capacitors, resistors, and other components. These are unavoidable for the most part, and must be accounted during the design and layout process. However, these random process variabilities can be used to our advantage if we use them to generate unique intrinsic identifiers. This is the idea behind physically unclonable functions (PUFs), which was first proposed by Gassend *et al.* [2]. They developed the first silicon PUFs through the use of intrinsic PV in deep submicrometer ICs. They used the intrinsic process variability of silicon devices during manufacturing to produce unique, random and unclonable digital responses and called it a PUF. Generally speaking, PUFs should present *unpredictable*, *robust*, and *unclonable* characteristics. A PUF's inputs and outputs map a specific set of challenges to a set of corresponding responses, which are called challenge–response pairs (CRPs) [3]. In other words, a PUF is a multiple-input (challenges) multiple-output (responses) function that has hard-to-predict dependency between the outputs and its inputs. The functional relationship between the challenge and response looks like that of a random function. Because the PUF is derived from random PV, it is very difficult, if not impossible, to predict the responses from a particular challenge or construct a function to do so in hardware or in software. A PUF is a promising solution to many security issues due to its ability to generate a unique identifier to an IC that can resist cloning attempts as well as physical tampering. However, maintaining large databases of PUF challenge response pairs and dealing with PUF errors makes it difficult to use PUFs reliably. Yan *et al.* [4] presented an innovative approach to authenticate PUF challenge response pairs on IC chips. In [5], PUFs were proposed to be used for device authentication and unique ID generation. In terms of security, PUFs show better resilience to tampering compared with other solutions and methodologies. However, the reliability of the responses of a PUF is vulnerable to various operating conditions such as temperature, voltage, and aging effects. Thus, ensuring the stability of PUF responses is essential to the viability of a particular PUF technology.

In this paper, we explore the possibility of intrinsic PUFs within commercial off-the-shelf (COTS) dynamic RAM (DRAM) ICs. We describe how to use the signatures to prevent modifications and uniquely identify and/or authenticate

Manuscript received December 17, 2015; revised April 29, 2016 and July 22, 2016; accepted August 31, 2016. Date of publication September 29, 2016; date of current version February 22, 2017. This work was supported by Comcast Corporation and Honeywell Corporation.

The authors are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 USA (e-mail: fatemeh.tehranipoor@uconn.edu; nima.karimianbahnemiri@uconn.edu; wei.2.yan@uconn.edu; john.chandy@uconn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2016.2606658

electronic devices. The motivation to examine these devices is that DRAMs have some unique advantages.

A. Large Input Pattern

Because of the large number of available bits in a typical DRAM, one can generate a large set of input challenges and correspondingly large output responses. This characteristic of the DRAM PUF is very valuable, which can make it to be distinct among all kinds of intrinsic PUFs.

B. Cost Effective

Since many computer systems have some form of DRAM on board, DRAMs can be used as an effective system-level PUF, as has been presented in [6] as well. It is also much cheaper than static RAM (SRAM). Thus, DRAM PUFs could be a source of random but reliable data for generating board identifications (chip ID). The advantage of the DRAM PUF is based on the fact that the stand-alone DRAM already present in a system on a chip can be used for generating device specific signatures without requiring any additional circuitry or hardware [7]. PUFs intrinsic to DRAM ICs have not been explored extensively. Ours is one of the first works in which a DRAM has been used as a system-level security PUF.

There have been two other investigations into using DRAMs as PUFs based on varying write cycles [7] or refresh cycles [8]. Refresh-based DRAM PUFs depend on the variation of decay of bits due to not refreshing the DRAMs. However, the time required to observe these decays can extend to hours making it impractical as a PUF. Modifying the write cycles and using the variability in write reliability as a PUF response are highly effective, but require significant modifications to the DRAM memory controllers in order to manage these different cycle times. Our approach, instead, depends on the startup values of a DRAM and as such is relatively quick and does not need to change the memory architecture at all.

Thus, our primary contribution is the identification of a DRAM PUF based on startup values. We examine the effect of various operating conditions such as temperature variation, voltage variation, and aging, which may influence the behavior of the DRAM PUFs. In addition, we propose a selection mechanism, which we call the *enrollment algorithm* to isolate highly stable bits within the large set of available bits in a DRAM. The final part is the validation of the PUFs considering reliability, randomness of the data, and uniqueness of the IDs. Previous works on DRAM PUFs have been based on decays due to delayed refresh [8] or memory remanence [9] and are highly timing dependent and thus can take significant amounts of time to evaluate the PUF response. Keller *et al.* [8] varied the temperature from 34.4 °C to 68.5 °C; however, ours is ranged from 0 °C to 80 °C, which can support a wide range of temperatures. We also considered the voltage variation, aging effects, and the uniqueness of the selected ID bits in Sections IV-C2, IV-C3, and VI-C, respectively, which they did not. Refresh-based DRAM PUFs are especially limited in their practicality because of the large delays (320 s) needed to generate a PUF response. This previous work also did not fully evaluate the stability of the PUFs under environmental conditions or consider important PUF metrics such as uniqueness and randomness. Our work is the first comprehensive evaluation of DRAM PUFs that are based on startup behavior rather than delay-based techniques and are thus not restricted by the delay times.

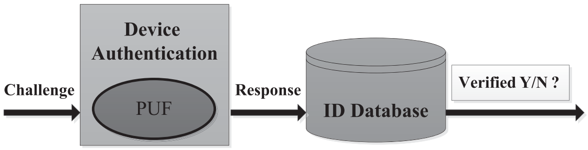


Fig. 1. System-level PUF.

The rest of the paper is organized as follows. Section II describes some required definitions in detail. The DRAM PUF description and properties are illustrated in Section III. Section IV shows the results of tests of real DRAM PUFs under different operating conditions such as temperature, voltage, and aging. Then, our enrollment algorithm for selecting the most stable bits is described in Section V. Section VI analyzes the experimental results and shows how valid the results are—particularly, measured results that demonstrate the effectiveness of DRAM PUF in terms of uniqueness, reliability, and randomness compared with the other kinds of PUFs. Finally, concluding remarks and future works are given in Section VIII.

II. Preliminaries and Definitions

A. System-Level Security

Most electronic systems are not designed with security in mind, and as a result, there are always threats from attackers to alter these systems and leak secret information from them. Even if the systems are securely designed, there is no assurance that the delivered system is authentic. System-level security mechanisms can use a subsystem on the board to prevent any altering or modification in system functionality and stop or reset system if any anomalous behavior has been detected. While it is difficult to authenticate the trustworthiness of any particular IC on a system board, a unique identifier, such as a PUF embedded in an IC, can be used and gives the IC a unique identity. However, with COTS parts, a PUF or chip ID may not be available, so mechanisms for intrinsic PUF identification are needed. In this paper, we present an intrinsic DRAM PUF that can be used to authenticate electronic systems on which DRAMs are present.

Fig. 1 shows an example of how a PUF can be used for device authentication. A system integrator who wants to authenticate a particular IC will issue a known challenge to the PUF embedded within the device. The PUF will respond with a response that can be verified with a trusted database.

B. Physically Unclonable Functions

We now look more closely at what exactly a PUF is. A PUF can provide a hardware specific unique signature or a “fingerprint” for an IC that can be leveraged to mitigate several security vulnerabilities. PUFs rely on manufacturing PVs to create unique identifiers or secret keys that can be used for various security applications including authentication and secure access. One strong characteristic of a PUF is that it cannot be reverse engineered easily. There are two fundamental requirements for building a PUF: *random* and *uncontrollable* variations. The variations must be random, thereby drastically reducing the probability that a unique signature will be repeated. In addition, the variations must be uncontrollable such that an adversary cannot clone the devices. In this paper, we use DRAM as PUFs to generate unique IDs for the system to make the systems more secure and reliable.

1) Process Variation: PV is a widely recognized phenomenon in modern CMOS technologies. PVs are important side effects in manufacturing process and completely uncontrollable. They will dictate that the output is nearly as likely to be a logic "0" as it is a logic "1." This probabilistic status for the output voltage is undesirable for conventional digital logic systems but can be leveraged in the implementation of PUF circuits. The effect of PV makes each piece of PUF to be *unique* and *unclonable*.

2) Current PUF Technologies: Since PUFs have gained considerable attention in the past few years, it has yielded several proposed approaches for the realization of these functions. So far, various kinds of PUFs have been proposed for key generation/ID, such as ring oscillator PUF [10], arbiter PUF, and clock PUF. Many methods have already been proposed for identification and authentication of ICs such as in [5] and [11]. Of particular interest are memory-based PUFs, which are attractive because most electronic systems have some type of memory included. Memory-based PUFs are usually based on the measurement of startup values of memory cells. Flash memory is a nonvolatile memory (NVM) that has been proposed as a memory-based PUF in [12] and [13]. An SRAM PUF is one existing-memory-based PUF, which has been presented in [14]–[16]. An SRAM PUF can generate a device-individual fingerprint using the startup behavior of its cells. Researchers have also proposed potential PUFs using future memory technologies such as memristors [17], [18], spintronic memories [19], and MRAM-based PUFs [20].

Memory-based PUFs are susceptible to possible attacks. Storing the secret in NVM memory such as flash represents vulnerability and may disqualify the NVM PUF for high-security applications that need to protect against invasive probing attacks. Volatile memories are considered safe against invasive attacks when implemented with a tamper detection circuit that shuts off the power supply to prevent compromise of the stored data as indicated in [21]. This is based on the assumption that the volatile memories lose their data immediately when the power supply is turned OFF. In some cases, when the memory is power OFF, an attacker can physically access to the system, extract the valuable, and secret information from the memories, although one solution is that the data should be always encrypted and stored to the memories [22].

It has been demonstrated that it is possible to physically clone SRAM PUFs with regard to the work of Helfmeier *et al.* [23]. In their work, when challenged, the physical clone could produce a response that was identical to the original device. Fault injection attacks can also change SRAM contents. The technique uses focused ion beam (FIB) probes to read out memory values from the SRAM to build the clone. Potentially, DRAM is also susceptible to these attacks, but the DRAM cell structure places the storage capacitors physically below the transistors, thus making it extremely hard to probe with FIB. Second, because the capacitances are so small and we are observing very small voltage changes (voltage reference), the act of probing it will change the measurement. As a result, it is much more difficult for a DRAM being attacked compared with other kinds of memories.

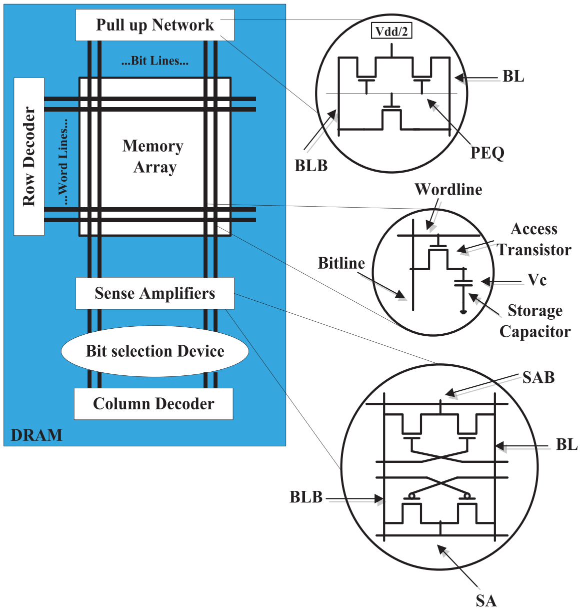


Fig. 2. Memory structure of a one-transistor DRAM array.

C. Dynamic Random Access Memory

DRAM is a type of physical memory used in many electronic systems. It is the most common type of memory in use today, which hold more data than SRAM and is significantly less expensive to manufacture. SRAM requires four times the amount of space for a given amount of data compared with DRAM. The disadvantage of DRAM is that it needs frequent power refreshing to retain its charge. Since reading a DRAM discharges its contents, a power refresh is required after each read. Apart from reading, just to maintain the charge that holds its content in place, a DRAM must be refreshed after a specified number of cycles. The term dynamic indicates that the memory must be constantly refreshed or it will lose its contents. Inside a DRAM chip, each memory cell holds 1 bit of information and is made up of a transistor and a capacitor. The capacitor holds the bit of information, a 0 or a 1. This is an efficient way to store data in memory, because it requires less physical space to store the same amount of data than if it was stored statically. The transistor acts as a switch that lets the control circuitry on the memory chip read the capacitor or change its state. Fig. 2 shows the structure of a one-transistor DRAM cell where each cell has a transistor and capacitor pair.

D. Problem Statement

Reliability and uniqueness are always important issues that hinder PUFs' practical applications. The stability of PUFs under various operating conditions has been a serious concern facing different kinds of PUFs. We tested our DRAM PUFs under several operating conditions such as temperature variations, voltage variations, and aging in order to consider their effects on stability. Hence, we constructively apply an enrollment algorithm to select the most reliable bits for PUF IDs.

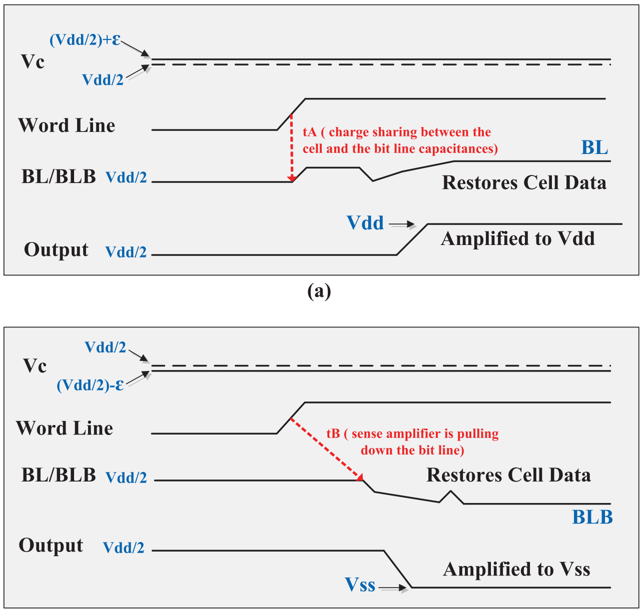
III. DRAM PUF Description and Properties

A. DRAM PUF Use Cases

PUFs can be authenticated by either using CRPs or ID-based authentication protocols. One challenge exploits the variation in one configuration to generate a response in a PUF supporting CRP authentication. Thus, this kind of PUF is required to have a large number of configurations. The generated DRAM PUFs can be used for various use cases as well. It can be directly used as a key or indirectly used to protect the entire system. In [6], it has been used as the board ID. Our primary contribution is the identification of a DRAM PUF based on startup values. However, DRAM PUFs have not been explored extensively.

B. Potential DRAM PUF Implementations

As indicated earlier, DRAM memory cells are composed of a paired transistor and capacitor. While ideally every DRAM cell should be identical, manufacturing imperfections cause slight physical variations in each cell. Moreover, every DRAM cell has its own physical trait. Therefore, the leakage effects on the storage nodes will vary as well. These physical variation characteristics can be used to develop PUFs. The only previous work on DRAM PUF has been based on altering or disabling the refresh cycle [8]. Hashemian *et al.* [7] developed an authentication methodology based on a DRAM PUF to provide resilience to counterfeit attacks. Modern DRAM chips have a built-in self-refresh module, as they not only require a power supply to retain data but must also be periodically refreshed to prevent their data contents from fading away from the capacitors in their ICs. The essential approach with refresh-based DRAM PUFs is to initialize all cells to "1," and then after some time, with refresh turned OFF, some of the cells will leak to "0." The randomness of which cells leak to "0" provides the opportunity for a PUF. The difficulty with using these refresh- or retention-based methods for a PUF is that it may take several minutes to hours for sufficient cells to flip to "0." Another potential approach is to use the remanence property of DRAMs. Contrary to popular belief, DRAMs can hold their values for surprisingly long intervals without power. DRAM cells retain their contents for a few seconds to minutes at room temperature. In fact, it has been demonstrated that sensitive information can be extracted from volatile memories due to data remanence effects [24], [25]. Based on our examination on DRAMs, the remanence approach is not feasible for constructing PUFs; however, remanence effects can be used for creating true random number generators [26]. In this PUF approach, instead of turning refresh off, we turn the power OFF to the DRAM, thereby accelerating the cell leakage, thus reducing the challenge time from hours to minutes.



(a)

(b)

Fig. 3. Timing diagram of a DRAM read operation of an uncharged cell biased to (a) V_{dd} or (b) V_{ss} due to PVs.

C. Startup-Value-Based DRAM PUF

In our observation of DRAM refresh and remanence properties, however, we observed that certain DRAMs actually exhibit behavior similar to SRAMs, i.e., they have seemingly random startup values. In other words, the cells do not initialize to "0" as would be expected. Thus, as with SRAMs, these startup values provide a potential for creating a PUF. The reason for this random startup behavior can be explained by the interaction of precharge, row decoder, and column select lines when the device is powered ON. Fig. 2 shows the structure of a typical DRAM array. Bits are stored either by charging the storage capacitor to V_{DD} or discharging it to ground. The timing diagram of the DRAM read operation of an uncharged cells is shown in Fig. 3. In order to reduce the electric field stress on the capacitor, one of the plates of the capacitor is usually biased to $(V_{\mathrm{DD}}/2)$. Before the reading operation, the signal to precharge the bit lines (PEQ) is disabled. In normal operation, before reading the cell, the bitlines (BL and BLB) and sensing nodes (SA and SAB) are precharged to $(V_{\mathrm{DD}}/2)$, and when the wordline is activated, the bitlines voltage will change slightly depending on the capacitance of the storage capacitor. This slight change is detected by the sense amplifier as a "1" (V_{dd}) or "0" (V_{ss}), as shown in Fig. 3. In other words, the level of BL and BLB nodes eventually reaches the operating voltage (V_{dd}) or ground (V_{ss}), respectively [27]. At startup, however, the storage capacitor has neither been charged to V_{DD} nor discharged to ground. Thus, at startup, the nominal voltage of each capacitor (V_c) is equal to the bias voltage $(V_{\mathrm{DD}}/2)$, which is equal to the bitline precharge voltage. Thus, when read, the sense amplifier is equally likely to read a "1" or "0." However, because of manufacturing variations, the storage capacitance of each bit will have slight differences, which leads to biasing of each bit to either a “1” or a “0.” This behavior is what allows the startup values of the DRAM to function as a PUF.

In the remainder of this paper, we examine the suitability of DRAM startup values to be used as a PUF.

IV. Real DRAM PUF Evaluation

In this section, we describe the results of a variety of experiments that we performed on actual DRAMs in order to assess the suitability and effectiveness of the proposed DRAM PUF [6].

A. Experimental Setup

We used a set of 1-Mbit HM51100AL CMOS DRAMs [28] in dual in-line package (DIP) packages. Our setup essentially consists of four parts: 1) data acquisition experimental setup, the FPGA-based development board (Spartan 6 FPGA); 2) the power supply and digital storage oscilloscope; 3) the breadboard-based circuit (extension circuit); and 4) the host PC. During data collection, the power supply supplies voltage to the extension circuit (off-chip DRAMs) that is mounted on the breadboard and we check the voltage levels using the oscilloscope. The communication between the host PC and the FPGA is composed of two connections:

1) USB connection, which is used for FPGA configuration download;
2) a high density serial connector, which is used for data communication among the PC software, ISE Design Suit 14.7, and the software running on the FPGA (developed using Xilinx EDK).

In other words, the FPGA was programmed to control the test sequence supplied to the DRAM chip and transmit the outputs of the DRAM to a computer using an on-board USB–UART module.

B. Uniformity of the Dynamic RAM

In this paper, we start with an examination of the uniformity of a DRAM-based PUF. Ideally, 50% of the bits should be “1” and 50% should be “0.” For each of the eight DRAMs that were used for startup value experiments, we took ten measurements of the uniformity, i.e., the percentage of bits that were “1” or “0” at startup. As shown in Fig. 4, without any write operation to the DRAM cells, nearly half of the cell values are one at startup. Even though they are not perfectly uniform, the uniformity is close enough to ideal that with proper bit selection it can be used as a PUF. The error bar shows the average, minimum, and maximum percentages of “1”s values of each DRAM (DRAM1–DRAM8) across different trials. As an example for DRAM1, the average, minimum, and maximum percentages of “1”s among all the ten measurements are 53.35%, 50.73%, and 56.78%, respectively. As can be seen, there is a slight bias to “1” in all the DRAMs. In addition, we examined the distribution of “1”s within some of the DRAMs (DRAM1, DRAM2, and DRAM3) to make sure that the “1”s were not concentrated in particular areas of the DRAM. We analyzed 1k sections of the three DRAMs, and even within these smaller 1k sections, the percentage of “1”s varied from around 48% to 55%, close to the ideal 50%, showing that the uniformity is valid across the entire DRAM. This metric defines how uniform is the proportion of “1”s and “0”s in the response bits of a PUF. If the PUF has a bias toward 1 or 0 in its responses, then the attacker can potentially guess the response. For an ideal PUF, the proportion of “1”s and “0”s in its responses should be equal.

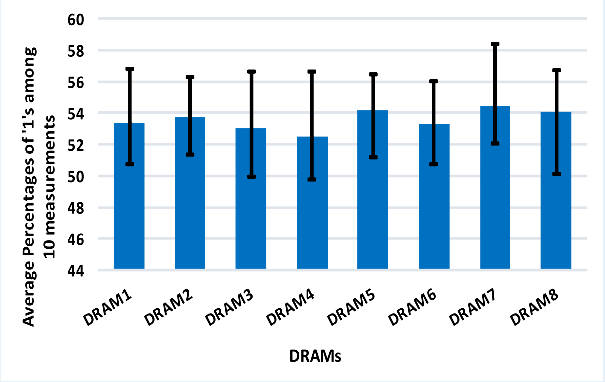


Fig. 4. Uniformity across ten measurements for each of the eight DRAMs (percentage of “1”s at startup).

C. DRAM PUF Evaluation Under Different Environmental Operating Conditions

In this section, we examine the stability of the DRAM PUF bits under various environmental operating conditions. A stable bit is a bit that does not change in any trial and remains the same over different measurements of the same or different conditions. There are various parameters that can affect PUF stability, such as PV, PUF activity, temperature, and supply voltage. Others have proposed PUFs that take into account both process and environmental variations such as crosstalk, which magnifies chip-to-chip signature randomness and uniqueness [29]. One of the advantages of our work is the stability evaluation against different operating conditions for more than one DRAM. We did all the experiments for three DRAMs, which we will call them DRAM1, DRAM2, and DRAM3. We explored the differences between reliable and unreliable DRAM cell values and the impact of operating conditions on them. To make a PUF highly reliable across its lifetime, unstable bits that are easily flipped by different operating conditions should not be used.

We start with baseline (Base) measurements [nominal condition (NC)] with the temperature set to 25 °C and the voltage to 5 V. For each DRAM, we took ten measurements whereby we read all 1 048 576 (2^{20}) startup bits. The result from Table I shows that for DRAM1, 37.9% of the startup values are read as “0” across all the ten measurements, and likewise, 43.5% are read as “1” across all the ten measurements. Thus, 81.4% of the bits are marked stable and the remaining 18.6% of bits, which read as both “0” and “1” on different measurements, are marked as unstable. In Table I, stability means that a bit is stable purely against the NC. The results show that the majority of bits are stable, meaning that one can have high

TABLE I

DRAM STABILITY ACROSS DIFFERENT NCs

DRAM1		DRAM2		DRAM3	
Bit Value		Bit Value		Bit Value	
0	1	0	1	0	1
37.9%	43.5%	27.9%	37.6%	26.6%	37.7%
81.4%		65.5%		64.3%	

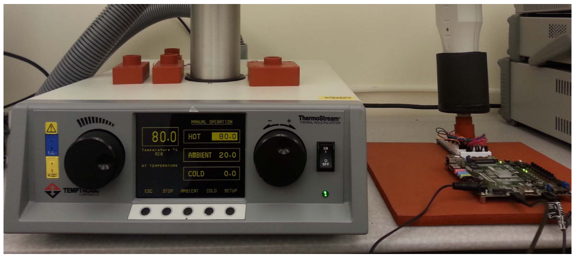


Fig. 5. Experimental setup (left) with Xilinx Spartan-6 FPGA (right) under the test using the ThermoStream system for HT and LT variations.

TABLE II

DRAM STABILITY UNDER DIFFERENT TEMPERATURE CONDITIONS COMPARED WITH THAT UNDER NCs

	DRAM1	DRAM2	DRAM3
	% of stable bits	% of stable bits	% of stable bits
NC-HT	78.8%	64.4%	49.8%
NC-LT	49.9%	54.4%	44.3%

TABLE III

DRAM STABILITY UNDER DIFFERENT VOLTAGE CONDITIONS COMPARED WITH THAT UNDER NCs

	DRAM1	DRAM2	DRAM3
	% of stable bits	% of stable bits	% of stable bits
NC-HV	55.4%	54.3%	30.5%
NC-LV	43.3%	26.7%	23.8%

confidence that in normal operating conditions, the bits will be the same at reconstruction as they were at enrollment. In the subsequent sections, we will examine the effects of varying operating conditions on stability.

1) Impact of Temperature Variation on Stability of DRAM PUF: Temperature plays a significant role in memory decay by affecting power leakage at the transistor level. It has been demonstrated that temperature variations have a much greater impact on bit stability than supply voltage variations for the SRAM PUF [14]. We performed the experiments by sweeping the temperature from 0 °C to 80 °C using a ThemoStream system (Temptronic TP04100A ThermoStream thermal inducting system), which is shown in Fig. 5. The ThermoStream system is a full-featured air stream system that delivers controlled temperature with speed and precision to devices and modules for thermal cycling and testing. In Table II, we show the bit stability under both high-temperature (HT) (80 °C) and low-temperature (LT) (0 °C) conditions. NC-HT and NC-LT compare the stability of the DRAM data under HT and LT conditions with the NC stability, respectively, i.e., the percentage of nominal stable bits (see Table I, where one can observe that the DRAM data remain stable under the HT and LT temperature variations). In Table II, stability means comparing a bit against the NC. Twenty measurements at LT and HT are taken (ten from each temperature) to illustrate the changes of startup values under temperature variations. First, we derive the stable bits among the ten measurements for each condition, i.e., those remaining the same across all ten measurements. Second, we find the stable bits among the ten measurements of the NC. Finally, we identify which bits are stable across both sets of bits and the output provides the final stability results as shown in Table II. For DRAM1, our data indicate that at HT, 78.8% of cells remain stable. At LT for DRAM1, however, only 49.9% of cells remain stable, indicating that LT has more of an effect on bit stability compared with HT. In addition, the stability varies slightly across different devices.

2) Impact of Voltage Variation on DRAM PUF Stability: Similar to the temperature variation, here, we observe the effect of voltage variation on PUF stability. We vary the nominal supply by 10% up and 10% down and observe the PUF's stability. Twenty measurements of startup values are taken at low voltage (LV) (4.5 V) and HV (5.5 V) (ten from each voltage). Table III contains the bit stability under both HV and LV conditions for different DRAMs. Again, NC-HV and NC-LV compare the stability of the DRAM data under HV and LT (LV) conditions with the NC stability, respectively. Note that in Table III, stability means comparing a bit against the NC. At HV, for DRAM1, our data indicate that 55.4% of cells are stable, and at LV, 43.3% of cells are stable. As with temperature, we see that voltage variations can have an impact on the bit stability. The reason is the structure of a DRAM cell, which consists of capacitor and a transistor. The startup values of a DRAM are dependent on the bias voltage of the capacitor, and very slight variations in the power supply voltage can alter the voltage differential across the capacitor. Similar to DRAM1, voltage variation has an effect on DRAM2 and DRAM3 also. Table III indicates that LV has more of an effect on bit stability than HV and this is true of all three DRAMs.

3) Impact of Aging on DRAM PUF Stability: Finally, we explore the potential impact of aging on the stability of the DRAM PUF. Several aging mechanisms can affect reliability during the lifetime of an IC. VLSI phenomena such as bias temperature instability (BTI), hot carrier injection (HCI), electromigration, and temperature-dependent dielectric breakdown (TDDB) are some of the causes of aging. As was mentioned in [30], among the BTIs, negative BTI (NBTI) affecting pMOS has more donating aging effect compared with positive BTI (PBTI) affecting nMOS. NBTI is enhanced by HT and high supply voltage. They both increase the threshold voltage and decrease the speed of CMOS transistors. A high switching rate in a circuit as well as excess supply voltage can enhance the HCI effect. A high operating voltage as well as higher temperatures can accelerate TDDB, a failure mechanism in MOSFETs.

In order to test the effects of aging on these DRAMs, we accelerated the aging process by performing burn in of the DRAM using the ThermoStream burn-in system. We did 8 h of HT aging at 80 °C to approximate the effects of 6 months of aging. In Table IV, a comparison of the stability of the

TABLE IV

DRAM STABILITY UNDER AGING CONDITION COMPARED WITH THAT UNDER NCs

	DRAM1	DRAM2	DRAM3
	% of stable bits	% of stable bits	% of stable bits
NC-AA	71%	65%	55.1%

TABLE V

STABILITY OF DRAMs DUE TO AGING OVER A PERIOD OF 1 YEAR

	DRAM1	DRAM2	DRAM3
	% of stable bits	% of stable bits	% of stable bits
Pre-aging Condition			
Sep. 2014	88.9%	91.6%	89.7%
Aged DRAMs			
Sep. 2014	87.1%	90.1%	80.2%
Feb. 2015	86.4%	85.1%	83.1%
Mar. 2015	90.2%	83.2%	78.0%
Apr. 2015	85.8%	82.4%	76.6%
Jul. 2015	87.3%	81.7%	81.3%
Aug. 2015	86.7%	81.2%	80.1%

aged DRAM data with the NC stability is provided, where NC-AA refers to the aging condition. The amount of stability degradation is not constant for each device. Table IV shows that after aging, still 71%, 65%, and 55.1% of the cells remain stable after aging across different measurements for DRAM1, DRAM2, and DRAM3, respectively.

We also aged DRAMs on September 2014 for the first time. Then we deliberately allowed the DRAMs to remain under normal conditions, but not powered ON, to see if the chips recover any aging effects over time. Table V shows the effect on stability over time and the percentages of "1"s and "0"s are the average values of stable bits among different measurements for each DRAM from September 2014 to August 2015. As can been seen, after almost 1 year, the aging does not have permanent effects on the stability in most cases. The percentages of "1"s and "0"s changed, but overall, the effect seems to be within 5%. In fact, it seems that the initial aging measurement was within the margin of error that we see over time, and one could make the case that aging has very little effect on permanent behavior of the PUF.

V. ENROLLMENT ALGORITHM FOR GENERATING PUF ID

A. Bit Selection Based on Neighbors' Stability Status

A key part of using a PUF to generate a unique ID or key is the enrollment process, i.e., the selection of bits to use for the ID. For example, in our 1-Mbit DRAMs, one could randomly select 128 bits to use as a key. However, Tables II–IV show that many of those bits would not be stable under different measurement conditions. Thus, during reconstruction, the bits may not be what they originally were. In this section, we describe an algorithm to select a set of bits for an ID/key that has a high likelihood of being stable. The key insight of the algorithm is that we use spatial information within the DRAMs to infer the stability of a bit cell. The approach is similar to the selection algorithm used by Xiao *et al.* [14] for SRAM PUF bit selection except that in our approach we take advantage of the fact that we have a better approximation of the layout of cells in the DRAMs. In other words, we have a grid for memory rows and columns that can give us a very good picture of the cell distribution in the memory array. Thus, spatial correlations (neighborhood stable cells) can be made in both x- and y-directions. In all, the algorithm uses spatial information within the DRAMs to infer the stability of a bit cell. In other words, stable neighbors provide better reliability than random selection. The basic algorithm is shown in Algorithm 1.

Algorithm 1 Highly Stable Bit Selection Algorithm

```
1: Apply $n$ measurements to each operating condition (NC, HT, LT, HV, LV).
2: Find bits that are stable across all $n$ measurements of all conditions for the specific DRAM that has been enrolled (DRAM1 or DRAM2 or DRAM3). Note that we do not perform aging during enrollment because it is not practical to age the chip because of the time involved.
3: Count the number of stable bits (ones and zeros separately) in each DRAM cells row.
4: Select rows $R$ that have more stable bits than selected thresholds ( $T1$ for '1's and $T2$ for '0's ) which have been selected based on experimentation in order to select 2048 (16 IDs) bits.
5: For each row $r \in R$, enroll bits $(r, j)$ that have a neighborhood of stable bits where $r$ is the row number and $j$ is the column number. The neighborhood of stable bits is defined such that row $r-1 \in R$ and $r+1 \in R$ and bits $(r, j-1)$ and $(r, j+1)$ are also stable.
```

The DRAM is organized as an array of cells—in our case for a 1-Mbit DRAM the array is 1024 rows × 1024 columns. We count the number of stable bits (1s and 0s) in each row and then select rows that have more stable bits than specific thresholds (T1 and T2). Thresholds have been chosen based on experimentation in order to select 2048 bits (16 128-bit keys). In the ideal case, half will be 0s and half 1s, among the 1-Mbit data in the next level of the algorithm. T1 and T2 are different in order to get the equal number of bits (1s and 0s) for the PUF ID bits. In fact, we have an algorithm to adjust the thresholds (T1 and T2). First, we selected a random threshold value and then based on the number of 1s and 0s that have been selected, our algorithm can change the threshold value to an upper or lower value in order to get 50% of "1"s (1024 bits) and 50% of "0"s (1024 bits). T1 and T2 are the thresholds for choosing rows that have more stable "1"s and "0"s, respectively. Among selected rows, those bits that have stable neighbors are also identified as potential highly stable bits suitable for enrollment as the PUF ID/key. Note that T1 and T2 should be selected in a way to find almost equal number of "1"s and "0"s bits for the IDs.

As we have mentioned in [14] that a stable cell surrounded by more stable cells is more likely to remain stable because its neighboring cells have experienced similar operational conditions such as aging effects. In other words, physically neighboring memory cells can strongly influence each other, in particular when they are physically connected. As an example, assume that row number i is selected as the one that has a large number of stable bits. If row numbers $i-1$ and $i+1$ also were selected, row i is one of the best that contains the most stable bits. After that, we can look at the neighbors of each

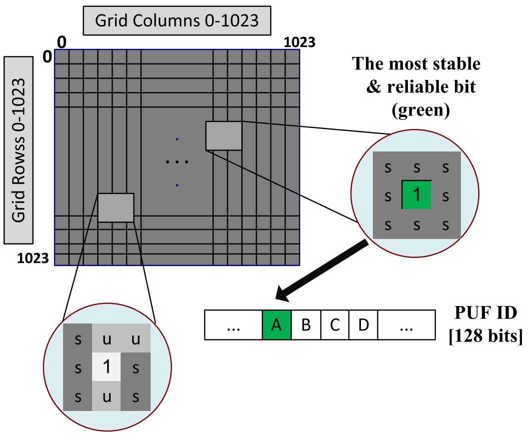


Fig. 6. Schematic of grids (rows and columns) on DRAM cells.

stable bit in that row to choose the ones that have more stable neighbors around it and finally is the best one to be used as ID bits. Note that after the enrollment process, the selected cells along with the responses would be stored in an external database. This database would then be used for authentication to generate challenges and evaluate responses.

As shown in Fig. 6, the most stable bits have been selected considering the neighboring cells. Basically, it shows how to select the most stable and reliable bits from the 1024 rows × 1024 columns grid for PUF ID considering the neighborhood cell stability approach. The more the number of neighbors is stable around a cell, the more reliable the cell is. In Fig. 6, the green cell at the center with value "1" has the best chance to be used as an ID bit since it has eight neighbors that are all stable. Similarly, the white cell is not very suitable to be used as an ID bit as it has several unstable neighbors around. We considered a 1024 × 1024 grid (rows and columns) on the memory cells. PUF ID bits have been selected from the bits that are stable among all the constraints (different operation conditions). Note that in Fig. 6, the characters "s" and "u" indicate the *stable bit* and *unstable bit*, respectively.

B. Our Selection Algorithm Versus Baseline Algorithm

As indicated earlier, we used our selection algorithm to select 16 128-bit keys from the available bits and compared it to a naive Base algorithm where 16 keys are selected at random from the 1-Mbit set. In both cases, the enrollment is done based on one, two, or three distinct measurements. In other words, $n = 1$, $n = 2$, or $n = 3$ in Algorithm 1. The Base case only uses one, two, or three nominal case measurements, whereas our selection algorithm uses 5–15 measurements—one, two, or three each for NC, LV, HV, LT, and HT. Aging is not used for enrollment because of the time involved and because it also shortens the lifetime of the device. Since our algorithm is a combination of a neighborhood selection algorithm and screening due to environmental measurements, we also evaluated each of these approaches separately. We evaluate the effectiveness of the selection algorithm by comparing the effect on reconstruction of the 16 keys. Ideally, on reconstruction, we should read back the same bits. Table VI contains a summary of our PUF ID reconstruction results. Reconstruction consists of reading the keys back ten times under all conditions (NC, LV, HV, LT, HT, and aging)—60 reconstructions for each of the cases in Table VI. The data show the number of bits that flipped in any of the reconstructions. As can be seen, the use of the enrollment algorithm (Algo) with just $n = 1$, i.e., five measurements, reduces the number of bit flips for DRAM1 from nearly 79% on average for Base to less than 14%, which is sufficient for using the PUF for chip identification. In fact, our results show that we can also use this PUF for key generation with minimal ECC check bits. Furthermore, using more measurements during enrollment can decrease the number of bit flips significantly to 2%–3%. It is interesting to note that while neighborhood selection (NS) and environmental selection (ES) are somewhat effective on their own, we get significantly better performance when both are used together (Algo). Note that Table VI is a worst case in that we examined the number of bit flips across multiple measurements including those under extreme operating conditions (40 measurements). Typically, however, reconstruction will be done under normal operating conditions. Table VII indicates the percentage of bit flips when reconstructing under just normal conditions. As can be seen, the bit flip rate is reduced to less than 3% with one set of measurements and to less than 1% with three sets of measurements.

VI. Analysis of Experimental Results and Their Validation

In this section, we first explain the multidevice evaluation on DRAMs based on different set of measurements, and then discuss the security metrics of uniqueness, randomness, and reliability followed by their results.

1) *Reliability:* Reliability is a measure of repeatability or consistency with which a PUF generates its response across environmental variations, temperature, voltage, and aging.
2) *Uniqueness:* Uniqueness (or interdie randomness) is a measure of how uncorrelated the response bits are across dies, and ideally, the response bits should differ with a probability of 0.5.
3) *Randomness:* Randomness (or intradie randomness) is a measure of the unpredictability of the response. This implies unpredictability of a response for a new challenge despite the prior knowledge of a large number of CRPs.

A. Multidevice Evaluation on DRAMs, Based on Different Set of Measurements

Here, we consider the effect of different operational conditions on various devices. We selected the three same DIP DRAMs, DRAM1, DRAM2, and DRAM3, to be tested under different conditions. These tests were applied for the entire 1-Mbit memory of each DRAM. Here, we compare the results among DRAM1, DRAM2, and DRAM3. For each condition, we did two to ten tests, collected data, and the percentage of stability among those data (for example, for the HT condition, we had two to ten different measurements). A bit is

TABLE VI

PERCENTAGE OF BIT FLIPS ACROSS MULTIPLE MEASUREMENTS (TEN SETS) UNDER BASE, NS, ENVIRONMENTAL SCREENING (ES), AND COMBINED (ALGO) APPROACHES

	DRAM1				**DRAM2**				**DRAM3**			
	Base	NS	ES	Algo	Base	NS	ES	Algo	Base	NS	ES	Algo
n=1	78.5%	13.5%	19.8%	13.8%	78.3%	20.6%	15.4%	13.5%	75.6%	18.3%	21.7%	10.4%
n=2	76.1%	8.7%	11.3%	3.8%	72.5%	13.5%	7.1%	7.7%	72.2%	14.4%	9.9%	8.8%
n=3	73.9%	6.6%	6.9%	2.3%	71.8%	7.6%	4.7%	3.1%	71.9%	8.5%	7.5%	3.8%

TABLE VII

PERCENTAGE OF BIT FLIPS ACROSS MULTIPLE MEASUREMENTS UNDER NORMAL OPERATING CONDITIONS

	DRAM1		DRAM2		DRAM3	
	BLine	Algo	BLine	Algo	BLine	Algo
n=1	47.3%	2.1%	50.4%	2.3%	49.5%	3.1%
n=2	45.9%	1.5%	46.9%	1.4%	44.2%	1.7%
n=3	45.2%	0.9%	45.7%	0.8%	43.1%	1.1%

marked stable if it has the same result for all measurements. Fig. 7 shows the percentages of "1"s and "0"s under different operational conditions and different set of measurements (2, 4, 6, 8, and 10) among DRAM1, DRAM2, and DRAM3. In most conditions, the degradation differences between the percentages of two measurement conditions and ten measurements are less than 10%. Fig. 7 shows that by increasing the number of measurements, the percentage of stable bits, "1"s and "0"s (both of them), does not decrease very fast. Thus, most bits remain stable across multiple measurements.

B. DRAM PUFs' Reliability Evaluation

High-volume manufacturing of PUF circuits requires test techniques to evaluate the quality of manufactured PUFs. PUF circuits are expected to show high reliability, uniqueness, and randomness. Reliability is an important feature of PUFs, which denotes their ability to produce the same response for a particular challenge. Generally, reliability of a PUF means that a given PUF can regenerate the same bits consistently. In our work, we chose stable bits based on the random selection and the proposed highly stable bit selection algorithm as discussed in the previous section. Various measurements from different operational conditions (HT, LT, etc.) were used for each DRAM to apply the bit selection algorithm on them. Then based on the number of distinct measurement approach ($n = 1$ or $n = 2$), we can determine which approach produced a fewer bit flips during the reconstruction phase. Our results show clearly that there is a relationship between better stability with bit selection and a higher number of distinct measurements (n), as shown in Tables VI and VII. We use Hamming distance (HD) across different PUF measurements as the basis of our metric. To estimate the reliability metric, an n-bit response (R_i) from challenge C and from chip i should be extracted at normal operating condition (room temperature and normal supply voltage). The same challenge C is applied to chip i at a different operating condition to extract an n-bit response ($R_{i,2}$). In the same way, T samples can be collected from chip i at different operating conditions. Hence, the average reliability metric (r) is estimated as [31]

$$r_i = \frac{1}{T}\sum_{t=1}^{T}\frac{\mathrm{HD}(R_i, R_{i,t})}{n} \times 100\% \qquad (1)$$

where $R_{i,t}$ is the tth sample of R_i. The reliability metric shows the average number of reliable PUF responses. Ideally, this value should be 0.

For measuring intradie HD, we consider 48 IDs (16 IDs associate with each DRAM). Each ID has been compared with different measurements of every operating condition, such as NC, HT, LT, HV, LV, and aging. Fig. 8 shows the distribution of intradie HD of 48 IDs from three DRAMs under various conditions. As it is shown, most of the IDs are stable under different conditions.

C. DRAM PUF Uniqueness Evaluation

In this section, we evaluate the uniqueness of the DRAM PUFs. In particular, uniqueness means that the responses resulting from evaluating the same challenge on different PUF instances should not be similar. The uniqueness of a PUF circuit among a population of PUF circuits manufactured depends on various factors such as the PV of a particular manufacturing process, any manufacturing defects, and the metric used to evaluate uniqueness. Interdie HD can be used to evaluate the uniqueness of the PUF data. It is typically used that averages the HD among the responses of various PUFs over multiple CRPs. Assume that there are k chips and R_i and R_j are the n-bit responses to a challenge C from chips i and j, respectively. Then the interdie HD among the k chips is defined as

$$\text{Inter-die HD} = \frac{2}{k(k-1)}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\frac{\mathrm{HD}(R_i, R_j)}{n} \times 100\%. \qquad (2)$$

Ideally, the HD between the responses should differ 50% of total responses bits.

We calculated the average interdie HD among all pairs of IDs that were extracted from the different PUFs (DRAM1, DRAM2, and DRAM3) based on our bit selection algorithm. Fig. 9 shows the distribution of interdie HD of the 48 IDs from the three DRAMs. The average HD is 0.4937 and close to the ideal 0.5. Hence, the proposed DRAM PUFs can provide unique identifiers. As shown in Fig. 9, the HD points tend to be very close to the mean of the set, as can be seen by the very small standard deviation of 0.055.

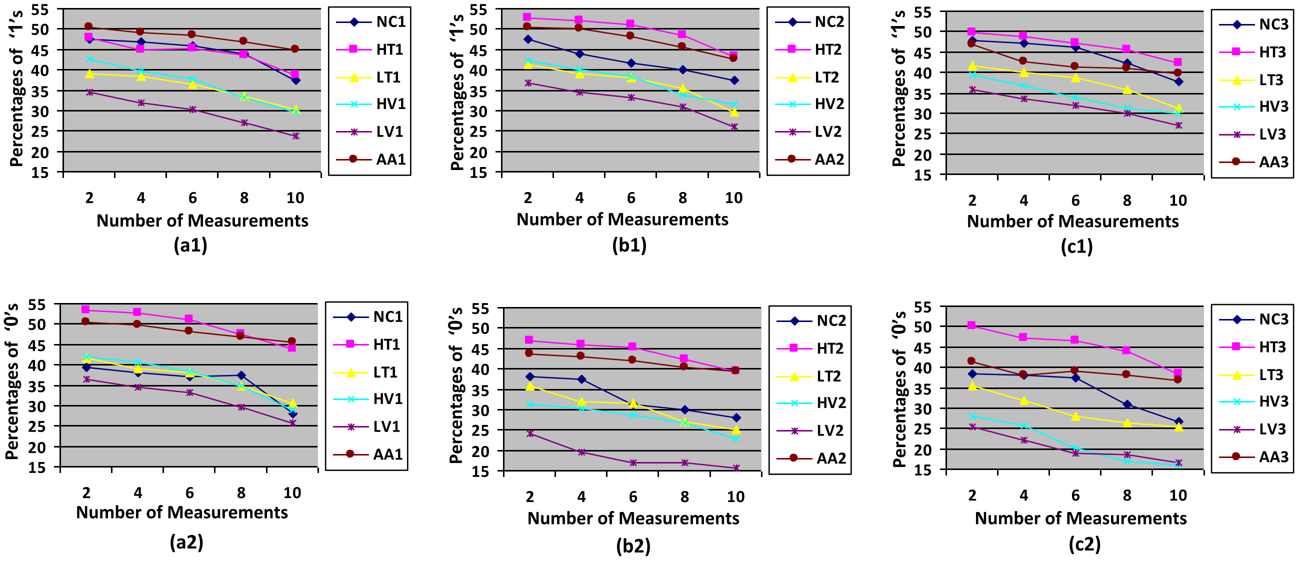


Fig. 7. Mutidevice evaluation. (a1)–(c2) Stability (percentages of "1"s and "0"s) across a different set of measurements for DRAM1, DRAM2, DRAM3, respectively.

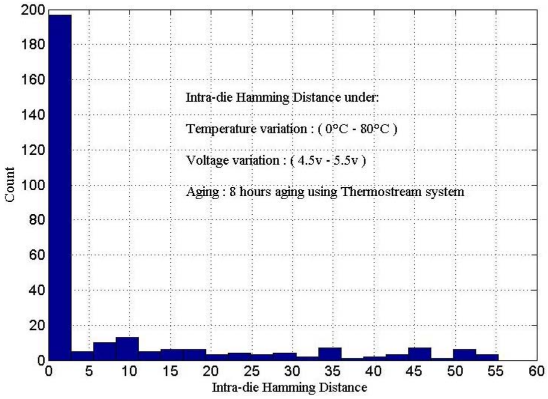


Fig. 8. Distribution of intradie HD among 48 (3 × 16) DRAM-based PUFs under different operating conditions.

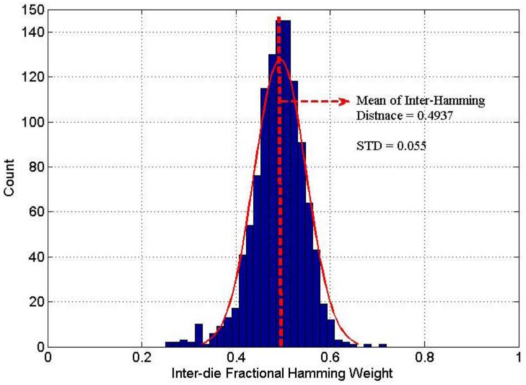


Fig. 9. Distribution of interdie HD of three DRAMs among the different extracted IDs.

D. DRAM PUFs' Randomness Evaluation

In PUF design, the randomness of data is very important as it can prevent the prediction of the cell values or the ID bits. In other words, perfectly random data mean that the PUF cells are generated independently of each other, and the value of the next cell cannot be predicted, regardless of how many cells have already been produced. PUFs using intrinsic randomness are very attractive as they can be included in a design without applying any modifications to the manufacturing process. Note that if the HD uniqueness measure discussed above is 50%, it does not mean that data are necessarily random. To evaluate the randomness of a PUF, statistical tests such as the NIST test [32], machine learning techniques, Shannon entropy, and min-entropy can be applied to the PUF data. Here, we considered min-entropy as a metric to estimate the unpredictability (randomness) of our DRAM PUF data.

Min-entropy is an approach for estimating the randomness of the PUF responses based on experimental data [33]. In particular, min-entropy indicates how many bits of a PUF response are uniformly random. In this literature, we estimate the entropy and min-entropy of the responses of all available PUFs. We have three DRAMs and for each of them, 16 IDs were selected based on the algorithm. Min-entropy is estimated as

$$P_{\mathrm{MAX}} = \mathrm{MAX}\{\mathrm{Hwt}(i), 1 - \mathrm{Hwt}(i)\} \tag{3}$$

$$\text{Min-entropy} = \frac{1}{128}\sum_{i=1}^{128}(-\log_2(P_{MAX}(i))) \tag{4}$$

where i is the number of ID bits. First, we have to consider $\mathrm{Hwt}(i)$ for each bit over all IDs. In fact, the Hamming weight of a bit $\mathrm{Hwt}(i)$ is defined as the number of nonzero bits. The min-entropy that has been calculated based on 4 is 0.9483. This value is approximately close to the ideal case min-entropy of 1.

E. Discussion

In this section, we evaluate security analyses, such as reliability, uniqueness, and randomness, for two different cases of ID extraction.

TABLE VIII

QUALITY EVALUATION OF IDs FROM DRAM PUFs

	DRAM1		DRAM2		DRAM3	
	Case1	**Case2**	**Case1**	**Case2**	**Case1**	**Case2**
No. IDs	263	16	178	29	205	22
Reli.	81%	100%	75.4%	100%	70.7%	100%
Unique.	0.539	0.503	0.581	0.514	0.448	0.492
Min_Ent.	0.89	0.99	0.78	0.96	0.86	0.98

1) *Case1:* Here, our goal is to find the maximum number of IDs using Algorithm 1. The enrollment and reconstruction are exactly similar to the results in Tables VI and VII except that the thresholds ($T1$ and $T2$) have been set to zero.
2) *Case2:* In this case, we try to find the maximum number of IDs with 100% reliability. To achieve this goal, 40 measurements from different operating conditions have been considered to extract the bits that are stable. Then we apply Algorithm 1 on the stable bits to directly select the ID bits. In other words, for the enrollment phase, all measurements of all conditions (40 measurements) are considered instead of one measurement of each condition (five measurements). Similar to Case1, the difference between the results from **Case2** in Tables VI and VIII is that the thresholds (T1 and T2) in Algorithm 1 are zero for Case2. Note that in Table VIII, Reli. is the reliability, Unique. is the uniqueness, and Min-Ent. is the min-entropy.

From Table VIII, the maximum number of IDs for DRAM1, DRAM2, and DRAM3 are 263, 178, and 205 in Case1, and 16, 29, and 22 in Case2, respectively. As is shown in Table VIII, reliability, uniqueness, and min-entropy have been calculated in different cases for different DRAMs using (1), (2), and (4), respectively. As a result of using all measurements for Case2, we are able to get 100% reliability, better uniqueness metrics, and a min-entropy that is very close to ideal. However, we are not able to enroll as many IDs in Case2 because of the restrictive enrollment process. As can be seen, the reliability results in Table VIII are lower than what are shown in Tables VI and VII. The reason is that here the thresholds are zero instead of the optimal thresholds that we tried to select in Section V.

VII. DRAM PUF AS A SYSTEM SECURITY SOLUTION

Since DRAMs are a system/board level component, they offer an opportunity to use the DRAM PUF to authenticate the system. However, since the DRAM is not embedded within another IC, it is potentially subject to attacks that may compromise the PUF. Two main vulnerabilities are that the pins are easily probed and the DRAM can be removed/replaced from the board. Since the pins of the DRAM are easily accessible, an attacker could exhaustively read all the memory cells in the DRAM and store the startup values. Using these stored values, the attacker could then theoretically reproduce the PUF responses of that DRAM. Note that because of the difficulty of replicating PV, it would be impossible to actually replicate a DRAM with the same physical responses. Instead, however, an attacker could add nonvolatile storage to the DRAM to store the startup values. This makes the attack relatively impractical because of the large size of typical DRAMs. Essentially, one would need to more than double the cost of the DRAM to add the nonvolatile storage. Moreover, the attack would need to determine whether the DRAM is in startup or not, meaning that there would need to be extra circuitry to detect writes to the DRAM. A simple approach would turn OFF the nonvolatile storage and turn ON the actual DRAM on detection of the first write. That could easily be defeated by simply writing a few random cells at startup, thus requiring the attacker to monitor writes to every cell and thus increasing the cost of the attack significantly. In addition, if suspicious of attacks, it would be relatively easy to determine that the DRAM package has an extra embedded nonvolatile storage component by imaging the DRAM package.

As to the second vulnerability, an attacker could remove the DRAM package from an authentic system and place it in an another potentially counterfeit system and thus allow the second system to be authenticated as valid. While this is possible, it does not allow the attacker to create a new "authentic" system without having access to a previously valid system. Presumably, the previously authentic system is nonfunctional or has been taken out of the supply chain, because otherwise the cost of the authentic system would make the need to counterfeit it irrelevant. The attacker would need one authentic system for every counterfeit system. To address this problem, manufacturers must keep a tight control of their supply chain and ensure that any authentic systems that have been taken out of the supply chain must either destroy the DRAMs or simply remove the DRAM responses from the database.

VIII. CONCLUSION

This paper identified unexpected startup behavior in a DRAM that could allow the DRAM to be used as a PUF primitive with proper bit selection to maintain high reliability and stability. We presented a novel PUF ID generation approach and evaluated the practicality of our PUF with empirical data that was obtained from a set of real DRAMs. Our experiments showed that temperature, voltage, and aging can have a major impact on DRAM PUF stability. The paper proposes a specific enrollment algorithm for generation of a stable PUF using memory cells within a DRAM module. The evaluation of DRAM gives insight into the randomness of the cells' startup values and their stability. The experimental results demonstrate that our algorithm is very effective at finding the most stable bits to be used as a 128-bit identifier. We also show that the DRAM PUFs' randomness and uniqueness metrics are close to ideal. While DRAM PUFs may not exhibit the same levels of stability as newer SRAM PUF techniques, they offer an opportunity for PUFs in systems that do not have SRAMs, require a high number of PUF CRPs, or want higher density than available with SRAM. Despite the many advantages of the DRAM PUF, there are still many challenges. As a future work, we are currently investigating further enhancements to the algorithm to improve the enrollment process. We also intend to observe the suitability of DRAM remanence for use as a PUF.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of Comcast or Honeywell.

REFERENCES

[1] N. Karimian, F. Tehranipoor, M. T. Rahman, S. Kelly, and D. Forte, "Genetic algorithm for hardware Trojan detection with ring oscillator network (RON)," in *Proc. IEEE Int. Conf. Technol. Homeland Secur. (HST)*, Apr. 2015, pp. 1–6.

[2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. 9th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2002, pp. 148–160. [Online]. Available: http://doi.acm.org/10.1145/586110.586132

[3] A. Maiti, I. Kim, and P. Schaumont, "A robust physical unclonable function with enhanced challenge-response set," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 333–345, Feb. 2012.

[4] W. Yan, F. Tehranipoor, and J. A. Chandy, "A novel way to authenticate untrusted integrated circuits," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 2015, pp. 132–138.

[5] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th ACM/IEEE Design Autom. Conf.*, Jun. 2007, pp. 9–14.

[6] F. Tehranipoor, N. Karimian, K. Xiao, and J. Chandy, "DRAM based intrinsic physical unclonable functions for system level security," in *Proc. 25th ed. Great Lakes Symp. VLSI*, 2015, pp. 15–20.

[7] M. S. Hashemian, B. Singh, F. Wolff, D. Weyer, S. Clay, and C. Papachristou, "A robust authentication methodology using physically unclonable functions in DRAM arrays," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, San Jose, CA, USA, 2015, pp. 647–652. [Online]. Available: http://dl.acm.org/citation.cfm?id=2755753.2755902

[8] C. Keller, F. Gurkaynak, H. Kaeslin, and N. Felber, "Dynamic memory-based physically unclonable function for the generation of unique identifiers and true random numbers," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 2740–2743.

[9] W. Liu, Z. Zhang, M. Li, and Z. Liu, "A trustworthy key generation prototype based on DDR3 PUF for wireless sensor networks," in *Proc. Int. Symp. Comput., Consum. Control (IS3C)*, 2014, pp. 706–709.

[10] C. Yin, G. Qu, and Q. Zhou, "Design and implementation of a group-based RO PUF," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2013, pp. 416–421.

[11] E. Ozturk, G. Hammouri, and B. Sunar, "Physical unclonable function with tristate buffers," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 3194–3197.

[12] Y. Wang, W.-K. Yu, S. Wu, G. Malysa, G. E. Suh, and E. C. Kan, "Flash memory for ubiquitous hardware security functions: True random number generation and device fingerprints," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2012, pp. 33–47.

[13] P. Prabhu *et al.*, "Extracting device fingerprints from flash memory by exploiting physical variations," in *Proc. 4th Int. Conf. Trust Trustworthy Comput. (TRUST)*, 2011, pp. 188–201. [Online]. Available: http://dl.acm.org/citation.cfm?id=2022245.2022264

[14] K. Xiao, M. T. Rahman, D. Forte, Y. Huang, M. Su, and M. Tehranipoor, "Bit selection algorithm suitable for high-volume production of SRAM-PUF," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, May 2014, pp. 101–106.

[15] M. Potkonjak and V. Goudar, "Public physical unclonable functions," *Proc. IEEE*, vol. 102, no. 8, pp. 1142–1156, Aug. 2014.

[16] G.-J. Schrijen and V. van der Leest, "Comparative analysis of SRAM memories used as PUF primitives," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2012, pp. 1319–1324.

[17] G. S. Rose, N. McDonald, L.-K. Yan, B. Wysocki, and K. Xu, "Foundations of memristor based PUF architectures," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit.*, Jul. 2013, pp. 52–57.

[18] P. Koeberl, U. Kocabaş, and A.-R. Sadeghi, "Memristor PUFs: A new generation of memory-based physically unclonable functions," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, San Jose, CA, USA, 2013, pp. 428–431. [Online]. Available: http://dl.acm.org/citation.cfm?id=2485288.2485390

[19] A. Iyengar, K. Ramclam, and S. Ghosh, "DWM-PUF: A low-overhead, memory-based security primitive," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, May 2014, pp. 154–159.

[20] E. I. Vatajelu, G. Di Natale, M. Indaco, and P. Prinetto, "STT MRAM-based PUFs," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, San Jose, CA, USA, 2015, pp. 872–875. [Online]. Available: http://dl.acm.org/citation.cfm?id=2755753.2757014

[21] C. Cakir, M. Bhargava, and K. Mai, "6T SRAM and 3T DRAM data retention and remanence characterization in 65 nm bulk CMOS," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2012, pp. 1–4.

[22] S. Kannan, N. Karimi, O. Sinanoglu, and R. Karri, "Security vulnerabilities of emerging nonvolatile main memories and countermeasures," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 34, no. 1, pp. 2–15, Jan. 2015.

[23] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, "Cloning physically unclonable functions," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, Jun. 2013, pp. 1–6.

[24] M. Gruhn and T. Muller, "On the practicability of cold boot attacks," in *Proc. 8th Int. Conf. Availability, Rel. Secur. (ARES)*, Sep. 2013, pp. 390–397.

[25] P. Gutmann, "Data remanence in semiconductor devices," in *Proc. 10th Conf. USENIX Secur. Symp. (SSYM)*, vol. 10. Berkeley, CA, USA, 2001, Art. no. 4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251327.1251331

[26] F. Tehranipoor, W. Yan, and J. A. Chandy, "Robust hardware true random number generators using DRAM remanence effects," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2016, pp. 79–84.

[27] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis & Design*. New York, NY, USA: McGraw-Hill, 2003.

[28] *Jameco Part Number 42219*, (2015). [Online]. Available: http://www.jameco.com/Jameco/Products/ProdDS/42219.pdf

[29] X. Wang and M. Tehranipoor, "Novel physical unclonable function with process and environmental variations," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2010, pp. 1065–1070.

[30] X. Wang *et al.*, "Aging adaption in integrated circuits using a novel built-in sensor," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 1, pp. 109–121, Jan. 2015.

[31] A. Maiti and P. Schaumont, "The impact of aging on a physical unclonable function," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 9, pp. 1854–1864, Sep. 2014.

[32] A. Rukhin *et al.*, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," DTIC, Tech. Rep. ADA393366, 2001.

[33] R. Konig, R. Renner, and C. Schaffner, "The operational meaning of min- and max-entropy," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4337–4347, Sep. 2009.

Fatemeh Tehranipoor (S'15) received the B.S. degree in computer hardware engineering from Mazandaran University, Mazandaran, Iran, in 2011, and the M.S. degree in computer hardware engineering from Shahid Beheshti University, Tehran, Iran, in 2013. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA.

Her current research interests include hardware security and trust, designing novel physical unclonable functions and true random number generations, system level security approaches, field-programmable gate array-based design and implementation, hardware Trojan detection and prevention, system authentication, Internet-of-Thing systems and applications, machine learning, and 3-D integrated circuit technologies and issues.

Nima Karimian (S'15) is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Connecticut, Storrs, CT, USA.

He was a Research Assistant with the Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran. He has five years of experience in biometrics, specifically ECG. His current research interests include machine learning, deep learning, pattern recognition, biometrics authentication and identification, optimization problem, artificial neural network, physical unclonable functions design, Internet-of-Things, medical devices, and hardware security primitives.

Wei Yan (S'15) received the master's degree in electronic engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA.

He has been involved in field-programmable gate array-based (FPGA) systems and memory reliability since 2009. His projects were solid-state device, high-speed interface, parallel error correcting code, embedded flash translation layer, fault-tolerant algorithm, and other memory systems. Currently, he is involved in IC authentication, ECC optimization, and embedded system simulation. His current research interests include digital system architecture, FPGA-based design, and hardware security.

John A. Chandy (M'91–SM'07) received the B.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1989, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana-Champaign, in 1993 and 1996, respectively.

He held the executive and engineering positions in software companies, where he was particularly involved in clustered storage architectures, tools for the online delivery of psychotherapy and soft-skills training, distributed architectures, and unstructured data representation. He is currently a Professor and the Associate Head of the Electrical and Computer Engineering with the University of Connecticut, Storrs, CT, USA, where he is also the Interim Director with the UConn Center for Hardware Assurance, Security, and Engineering and the Co-Director with the Comcast Center for Cybersecurity Innovation. His current research interests include high-performance storage systems, reconfigurable computing, embedded systems security, distributed systems software and architectures, and multiple-valued logic.