sensors

Article

Intrinsic Physical Unclonable Function (PUF) Sensors in Commodity Devices

Shuai Chen [1], Bing Li [1] and Yuan Cao [2,*]

[1] Shenzhen Research Institute, SEU-FiberHome Joint Research Center,
School of Cyber Science and Engineering, School of Microelectronics, Southeast University,
Nanjing 210000, China; chenshuai_ic@seu.edu.cn (S.C.); bernie_seu@seu.edu.cn (B.L.)

[2] College of Internet of Things Engineering, Hohai University, Changzhou 213000, China

* Correspondence: caoyuan0908@gmail.com

Received: 10 April 2019; Accepted: 24 May 2019; Published: 28 May 2019

Abstract: The environment-dependent feature of physical unclonable functions (PUFs) is capable of sensing environment changes. This paper presents an analysis and categorization of a variety of PUF sensors. Prior works have demonstrated that PUFs can be used as sensors while providing a security authentication assurance. However, most of the PUF sensors need a dedicated circuit. It can be difficult to implemented in commercial off-the-shelf devices. This paper focuses on the intrinsic Dynamic Random Access Memory (DRAM) PUF-based sensors, which requires no modifications for hardware. The preliminary experimental results on Raspberry Pi have demonstrated the feasibility of our design. Furthermore, we configured the DRAM PUF-based sensor in a DRAM PUF-based key generation scheme which improves the practicability of the design.

Keywords: physical unclonable functions; PUF sensor; DRAM PUF; key generation scheme

1. Introduction

In the internet of things (IoT) era, billions of smart devices are connected and interact with each other. A large number of sensor nodes are distributed in the network for sensing the world. The data collected from the sensors are used to trigger the whole system to interact. However, the sensing, collecting and communication of sensor data are vulnerable to attacks [1].

The sensors are working in a challenging world. This variable and hash environment results in the sensors being prone to failure. Furthermore, the sensors that are distributed all over the world can be physically accessed by the attacker. Therefore, they are also vulnerable to physical attacks. For example, some physical attacks focus on the long-term private key stored in non-volatile memory (NVM) that is assumed to be secure. The secret data stored in NVM like Read-Only memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM) and flash can be recovered even after erasures [2]. Non-invasive, semi-invasive and invasive attacks [3] can extract the private key, making NVM the weak link in many security implementations.

Furthermore, the sensor nodes are limited in computation, memory and power because of the resource constraints. Therefore, certain traditional security solutions cannot be embedded in it. Also, there are billions of connected devices with different manufacturers and service providers. Thus the nodes may not have global identifications. Therefore, it is hard to authenticate the identity of each node to countermeasure the false ones [1].

Another emerging threat for sensor nodes is sensor spoofing attacks [4]. The attackers can spoof a false analog signal to the sensor which may cause malfunctions. It is hard to address this attack because sensors cannot inherently distinguish between malicious and non-malicious signals. One promising solution is the so-called sensor fusion [4]. By comparing the sensed data from various sensors, the malicious signal can be detected. However, for some low-cost sensor nodes, the overhead of multi-sensors is unacceptable. Therefore, the low-cost intrinsic sensors that do not use an analog-mixed circuits are worth studying.

Physical unclonable functions (PUFs) [5] exploit the random variability of nano-scaled manufacturing variations to achieve tamper resistance, and physical and mathematical unclonablility. It has become an indispensable primitive to countermeasure the aforementioned security issues, since:

- PUF provides the possibility for low-cost key storage and authentication which is not vulnerable to physical attacks [6,7]. No secret data needs to be stored in NVM. Instead, the secret key or identification is derived from the physical properties of the PUF when needed.
- Some PUFs are very sensitive to environmental parameter changes. Therefore, one can use the PUF response to sense environmental changes. For example, a ring oscillator (RO) PUF can measure the temperature in the Field-Programmable Gate Array (FPGA) boards [8]. The digital PUF is easily embedded in the Application Specific Integrated Circuit (ASIC) or FPGA without analog circuits. Therefore, a PUF-based sensor is a good candidate to countermeasure a spoofing attack where the outputs from multiple traditional and PUF sensors can be compared to catch an anomaly [9].
- Some digital PUFs can be used as a fusion of low-cost key storage, authentication and sensor. It is easy to be implemented in digital devices because it does not require any analog-mixed process.

However, most of the existing PUF sensors rely on dedicated circuits that are very difficult, if not impossible, to find in off-the-shelf commodity devices [8–10]. The addition sensors may not meet the requirements of some low-cost systems. Therefore, some intrinsic PUF instances within standard hardware that do not need any dedicated circuits or hardware modifications can be evaluated as PUF sensors to overcome the requirements of the off-the-shelf devices. In this paper, we propose a temperature sensor leveraging Dynamic Random Access Memory (DRAM) PUF in commodity devices. Our method is based on the existing DRAM circuits and does not require adding any hardware circuits in the device.

Our key contributions are as follows:

- Implementation of an intrinsic DRAM PUF-based temperature sensor in off-the-shelf commodity devices.
- Test the feasibility of the DRAM PUF-based sensor and configure it in a DRAM PUF-based key generation scheme.

The rest of the paper is organized as follows. Background and some related security issues are introduced in Section 2. In Section 3, we summarize the existing PUF sensors. A novel temperature sensor based on DRAM PUF is proposed in Section 4. Evaluation of the novel temperature PUF is presented in Section 5 and we discuss this work in Section 6. The conclusion is presented in Section 7.

2. Background

2.1. Physical Unclonable Functions

Since the introduction of an optical PUF in 2002 [11], researchers have proposed various PUF designs. The digital PUFs are the most popular components. The essence of a digital PUF is a hardware circuit with unique binary or analog behavior which depends on the integrated circuit (IC) manufacturing variations, e.g., delays, frequencies or capacitances. The process variations are randomness, even the manufacturer can not predict or clone it. Hence, PUFs have been proposed as an important building block for security systems. For example, PUFs can be used in a lightweight key storage scheme [6,12] or authentication and identification scheme [13,14] which does not need any NVM to store the secret data. The private key is derived from the PUFs during run time instead of being stored in the NVM. Thus, it can be used to protect against certain NVM attcks.

Most of the digital PUF designs (e.g., Arbiter PUF [15] and Ring Osillator (RO)-PUF [16]) require the design of dedicated circuits which tend to be rather complex in design and manufacturing. Also, it is difficult, if not impossible, to find these dedicated circuits in existing commodity devices. Therefore, people are becoming more and more interested in intrinsic PUFs, e.g., Static Random-Access Memory (SRAM) PUF [17] and DRAM PUF [18]. DRAM PUF is the focus of this work.

2.2. Secure Key Management Scheme for Sensor Networks

Key management is considered the most critical component of security systems [19], as the leakage of keys makes even the toughest cryptographic system pliable. It is the same in the trust management scheme in the wireless sensor network (WSN) [20,21]. To maintain the tolerant level of trust among the sensor nodes, trust management is established to authenticate the genuine and fake sensor nodes. In the trust management system, a robust and lightweight key management scheme is critical.

In paper [22], the author divided the key management schemes into symmetric, asymmetric and hybrid, based on the encryption techniques. For the scenario of dynamic WSN, paper [23] proposed a dynamic key management scheme by refreshing the pairwise keys periodically or on demand. There is also an existing survey on key management in WSN that classifies the key management schemes as key pre-distribution schemes, hybrid cryptography schemes, one-way hash schemes, key infection schemes, and key management in hierarchy networks. In the key pre-distribution schemes, one lightweight solution is that all the nodes only need to store a master secret key. When used in the field, the key management scheme is initiated by the global master key. However, due to the NVM-based key storage scheme, the previous works showed that any micro-controller, FPGA, secure memory, smart-card and even ASIC can be attacked successfully by several attack methods [1,24,25]. The whole WSN will be compromised if one sensor node key is promised.

One complement to the aforementioned leakage attacks is proposed in paper [26]. The authors proposed a public-key encryption scheme that is resistant to key leakage attack. However, the public-key based crypto desigh is too expensive to implement in the low-cost systems.

A natural defense would be to store the secret key in tamper-resistant hardware. For example, paper [27] proposed a countermeasure using the coating layer and paper [28] presents a construction by the error detection codes that is resilient to key leakage. However, the traditional tamper-resistant hardware might also vulnerable to some attacks, e.g., [25,29]. Furthermore, it is difficult to implement in the resource-constrained sensor nodes and the off-the-shelf devices.

PUF provides the possibility for a tamper-resistant, low-cost key storage and authentication scheme [7,30]. The advantages of this combination would be:

- Tamper-resistant. The key is extracted from the nano-scale manufacturing variations, not “burned” in the NVM like EEPROM. Therefore, even an invasive attack cannot compromise the secret key.
- Low-cost. For some intrinsic PUFs, the security system does not need to add any dedicated circuit in hardware. For example, the implementation of DRAM PUF [31] just requires the firmware modifications.
- Combing node identity. Integrating node identity in the process of key production will make a system more secure [32]. It is also helpful for the resistance of node replication attacks [33]. PUFs can be seen as the “fingerprints” of hardware, it can be used in identification and key generation. In the PUF-based key generation scheme, the key is extracted from the hardware feature. Obviously, this feature can be seen as the identity of nodes.

3. PUF Sensor

Based on the roles of PUF in PUF sensors, we classify the current PUF sensors as PUF-protected sensors and PUF-based sensors.

3.1. PUF-Protected Sensor

The so-called PUF-protected sensor is a variety of sensor that leverages the functionality of a conventional PUF to authenticate the sensor nodes or protect the sensed value.

The PUF-protected sensor was first proposed in paper [34]. As shown in Figure 1, the conventional PUF is co-mingled with the sensor so that the sensor value is determined by both the physical quantity and PUF response. In paper [34], the signals of the offset generator are selected randomly by the PUF response to generate the final sensed value. Like the PUF-based authentication process, the proposal also has an enrollment phase to store some challenge-measurement-response pairs. When used in the field, the micro-controller just accepts the sensed data that passed the verification process. Therefore, the PUF-protected sensor becomes a promising mechanism for securing remote sensors.

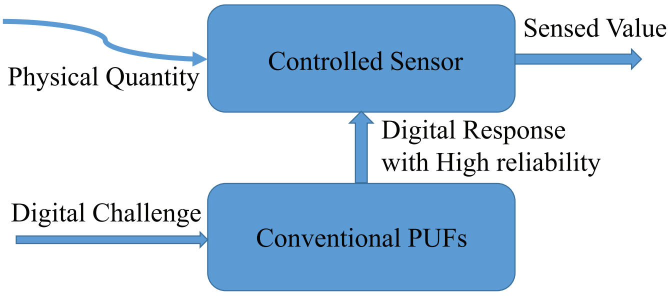


Figure 1. Physical unclonable function (PUF)-protected sensor.

Cao et al. extracted some PUF features from the CMOS image sensor [35] to improve the image sensor as a trusted entity. Each pixel can generate a 1-bit PUF response based on the fixed pattern noise resulting from manufacturing variations. Therefore, each image sensor can generate a unique and reliable signature for the pictures using the hash function. This design can be implemented on the existing CMOS image sensors without a dedicated circuit. It can be used in a PUF-based perceptual image hash scheme to carry out the image content birth certification.

3.2. PUF-Based Sensor

As shown in Figure 2, the PUF-based sensors evaluate the environmental parameters based on the environment-dependent-behavior of the PUF. Usually, the PUF-based authentication and key generation scheme has two steps: The enrollment phase in the security environment, and the authentication or key generation phase when used in the field. Most PUFs exhibit unreliability problems due to inherent sensitivity to the environmental conditions, e.g., temperature and supply voltage [36]. This unwanted fact gives us a new idea to sense environmental changes.

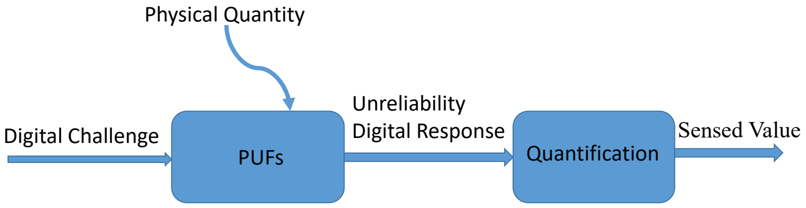


Figure 2. PUF-based sensor.

Paper [37] presented a micro-electro-mechanical (MEM) relay based RO-PUF to sense pressure and provide authenticity. Compared to the conventional RO-PUF, the “sensorPUF” leveraged the MEM relay inverter to replace the CMOS inverter. The MEM relay inverter can sense the pressure changes and influence the behavior of RO-PUF to a unique but deterministic function. Therefore, the sensed value has both the pressure feature and the hardware “fingerprints” feature to realize an authenticated pressure measurement. However, it is hard to implement this design in some low-cost IoT devices.

Compared to the aforementioned work, paper [10] proposed a universal RO-PUF to sense voltage. By leveraging the sensitivity challenges, the authors also investigated a challenge selecting a method to improve the sensing capability. Similarly, paper [9] presented another type of voltage sensor based on the error rate or universal Glitch PUF.

Furthermore, the oscillation frequency is sensitive to temperature changes [8,38]. It can be used as the thermal sensor in FPGA to monitor the die temperature. This temperature sensor also can be used as a possible malicious application of the thermal covert channel. The transmitter can encode the transmitted data into heat patterns and the RO-PUF based temperature sensor can detect temperature changes in the receiver [39,40].

4. Proposed Intrinsic PUF Sensor Based on DRAM PUF

Although the aforementioned design allows the sensor to inherently provide assurance of authenticity by co-mingling sensing and unique hardware features, it is difficult, if not impossible, to be found in the existing off-the-shelf commodity devices. Therefore, the design and implementation of intrinsic PUF sensors that do not need to add any dedicated hardware are necessary. This paper briefly presents a novel intrinsic temperature sensor based on the decay feature of DRAM PUF.

4.1. DRAM PUF

DRAM is pervasively used in existing embedded systems. As shown in Figure 3, the DRAM cell consists of a transistor and a capacitor. Each cell stores 1-bit data in the capacitor and can be accessed through the transistor. The cells are grounded in a 2-dimensional array, where each row is connected to a word line and column linked in a bit line. In each cell the capacitor leaks the charge over time which causes data to flip from the previous contents. Therefore, DRAM chips usually have a periodical self-refresh module to recharge the capacitor on time which is controlled by the memory controller.

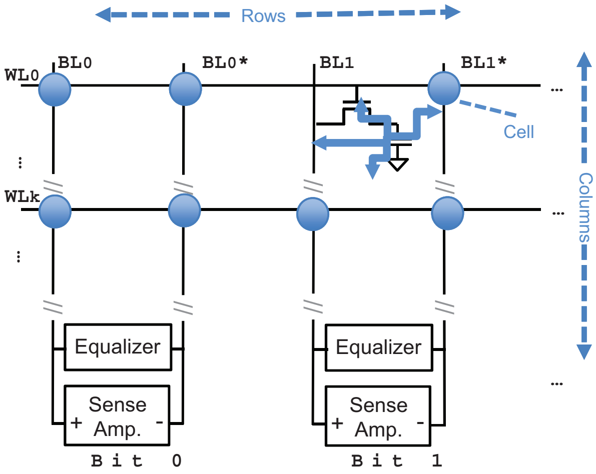


Figure 3. The architecture of DRAM.

Due to the manufacturing variations among DRAM cells, some cells leak faster than others. After a certain delay time, enough charge has leaked crossing the threshold from some cells such that the stored logical bit flips. For the other cells, the contents stay stable. This behavior heavily depends on the random manufacturing variations and environment (e.g., temperature). The random data flips allow DRAM to be a good candidate for PUF [31]. It can be used as a run-time accessible DRAM PUF in the key generation or authentication scheme in commodity devices [18]. Beyond that, Tehranipoor et al. [41] attempted to use the random start-up value of DRAM as a PUF. However, this method needs to control the power supply of the DRAM chip like the SRAM PUF. It is difficult to be implement in commodity off-the-shelf devices. In [42], the authors introduced error patterns bound to manufacturing variations of DRAM by reducing DRAM read access latency below the minimum value in data-sheet specifications to implement DRAM PUF. In this scenario, the computer system needs at least two DRAM ranks because normal read latency should be maintained at least in one rank to keep systems operational. Therefore, this method is difficult to implement in low-cost embedded systems. For example, we tried to implement this proposal in Raspberry Pi (Rpi) B+. However, Rpi B+ just have one rank. The system crashed when we set the value of reading latency as less than 2. In our study, we use the decay-based DRAM PUF as the covert channel to sense temperature changes in the commodity off-the-shelf devices.

4.2. Implementation of DRAM PUF on Raspberry Pi B+

We implemented and tested our sensor on three Rpi B+ development boards which are the most popular commodity embedded platform. Each board have a Broadcom BCM 2835 systems on chip (SOC) module which includes a 700 MHZ ARM11 76JZF-S processor and a VideoCore IV that implements a 512 MB Double Data Rate SDRAM (DDR2) memory.

We modified an open source firmware of Rpi [43] in order to get the privilege of DRAM decay control. The refresh of the whole DRAM has to be disabled as we can not control part of the DRAM address on Rpi. Similar to the previous work in the paper [31], we set the selective refresh by loops over all memory address that need to be refreshed by issuing a read to the first word in every DRAM row. Therefore, during query process of DRAM PUF, the other applications can operate normally.

Figure 4 shows the structure of DRAM PUF on Rpi B+. There are three important parameters for DRAM PUF: PUF address, initial value and decay time. PUF address is the DRAM address that supposed to be used as PUF. The initial value is a set of digital data that used to initiate the PUF address before the PUF query. In the PUF query, decay time indicates how long the DRAM PUF is disable refreshed. In our implementation, we set a 16 MB DRAM PUF with initial value = 1 and decay time = 60 s. These parameters can be compiled in the kernal of Rpi B+ or acquired from the upper computer via Universal Asynchronous Receiver/Transmitter (UART). And then, the PUF query code running on Graphics Processing Unit (GPU) can gain these parameters from Central Processing Unit (CPU) by the mailbox. The programs running on the CPU and GPU are only able to communicate via the mailboxes. All the aforementioned work has been published on the paper [44]. In paper [44], the decay feature of DRAM was used as covert channel leveraging the PoP architecture.

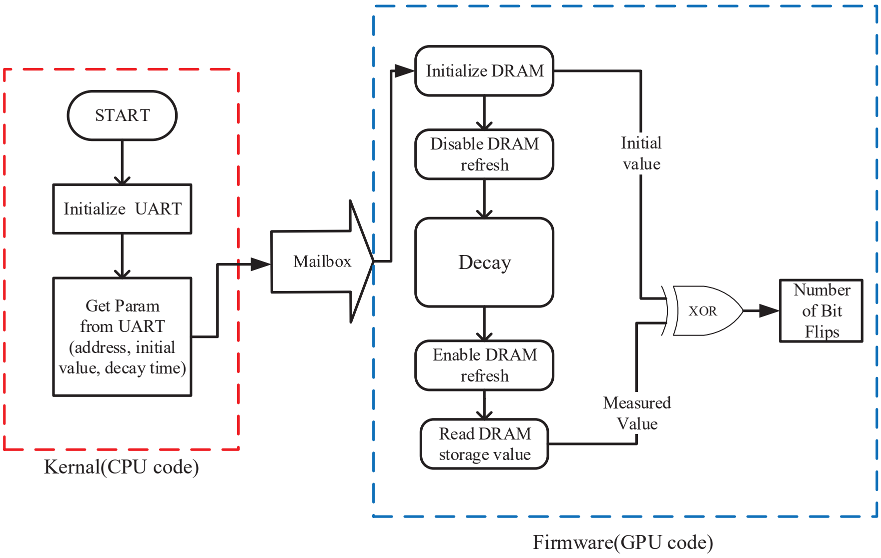


Figure 4. Structure of DRAM PUF implementation on Raspberry Pi (Rpi) B+.

4.3. Embedded in the PUFs-Based Key Generation Scheme

The DRAM PUF-based key generation scheme has two phases: enrollment and key generation. In the enrollment phase, the security system generates helper data $h = GEN(r)$, where $GEN()$ is the generation process of helper data algorithm [45] and r is the output of DRAM PUF. h can be generated and stored in the devices or the data center. In the key generation process, when used in the field, the device queries the DRAM PUF and receives a measurement r'. By the helper data algorithm, the system can regenerate the original DRAM output $r = REGEN(R', h)$ if the Hamming Distance of r and r' is smaller than the error-correction capability of helper data algorithm, where $REGEN()$ is the error correction process of the helper data algorithm. In the DRAM PUF-based sensor, the Hamming Distance of r' and r is the number of bit flips caused by the temperature changes. Therefore, as shown in Figure 5, the cross-correlation between the errors in DRAM PUF-based key generation process and the temperature variations is established. The novel temperature sensor is integrated into the conventional PUF-based key generation scheme.

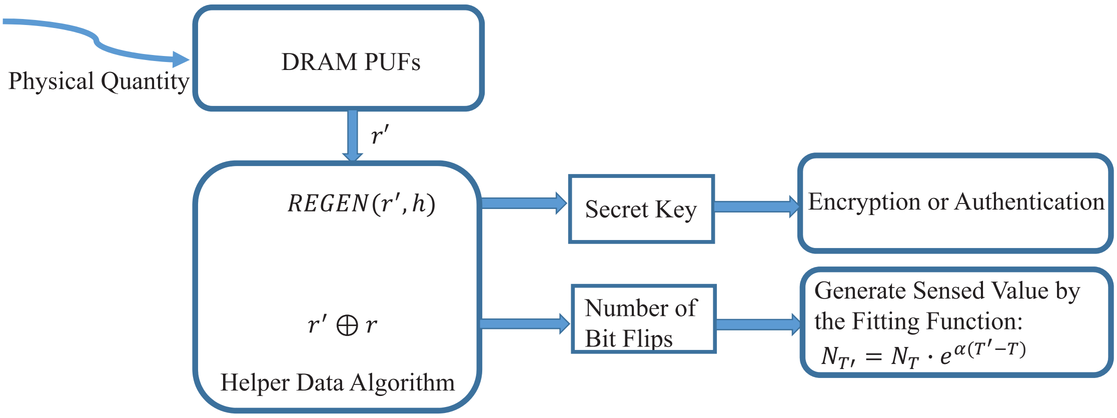

Physical Quantity
DRAM PUFs
r'
$REGEN(r', h)$
$r' \oplus r$
Helper Data Algorithm
Secret Key
Encryption or Authentication
Number of Bit Flips
Generate Sensed Value by the Fitting Function:
$N_{T'} = N_T \cdot e^{\alpha(T'-T)}$

Figure 5. Module of DRAM PUF-based sensor fused with the key generation scheme.

For DRAM PUF, one promising key generation scheme was proposed in paper [31]. The authors discretize the decay feature by divided the DRAM cells into fast cell, slow cell and no cell. The decay time of fast cells is shorter than slow cells. Some randomly selected fast cells and slow cells, which are either extremely fast or extremely slow, are used to generate a key. In our tests, we simplified the implementation of this scheme by ordering these selected cells in their physical address. The contribution of this scheme is that the output of this proposal can be capable of higher randomness with higher reliability. The reliability of this scheme has been tested in paper [31].

5. Experimental Set Up and Evaluation

5.1. Experimental Set Up

Figure 6 presents the schematic of the experimental set up used to verify the feasibility of our design. It includes a thermal chamber with a thermal chamber controller; three Rpi B+ boards running the modified open source firmware that can communicate with the workstation via UART; and a workstation running the control scripts. The automatic test process is the following:

1. The workstation sets the temperature of thermal chamber by the thermal chamber controller and starts the loop to monitor the temperature.
2. When the temperature requirements are reached, the workstation writes the parameters of the DRAM PUF to the CPU.
3. Execute the DRAM PUF query process on GPU and count the number of bit flips.
4. Restart from step 1 for next set of parameters.

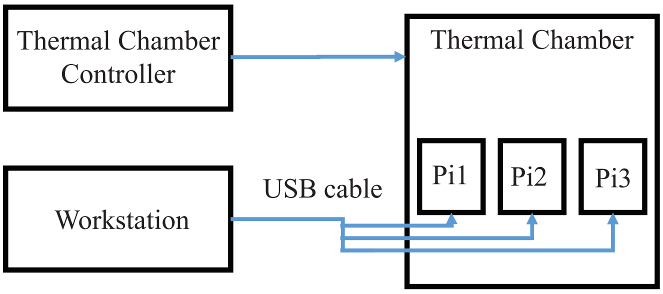


Figure 6. Experimental set up.

5.2. Test Results

We measured the temperature sensor instances on three Rpi B+ boards (Rpi1, Rpi2 and Rpi3) with temperature t = 15 °C, 20 °C, 25 °C, 30 °C, 35 °C, 40 °C and decay time 60 s. In each device, we measured one 16 MB DRAM in stride in the free address area of DRAM.

Decay time and temperature are two parameters that affect the number of bit flip for DRAM PUF. In our proposal, we evaluate the temperature, leveraging the number of bit flips of DRAM PUF. In Figure 7, we show the dependency between temperature and number of bit flips under certain decay times. The temperature changing was achieved using a thermal chamber. Although decay time affects the number of bit flips significantly, it does not influence the dependency between temperature and decay time. The temperature characteristics of DRAM PUF are very similar for different decay times. In the following tests, we evaluate our DRAM PUF under 30 s.

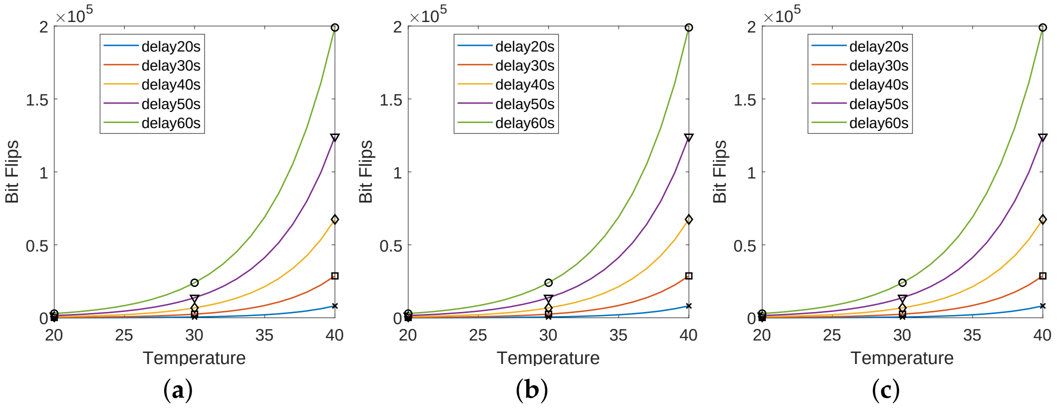
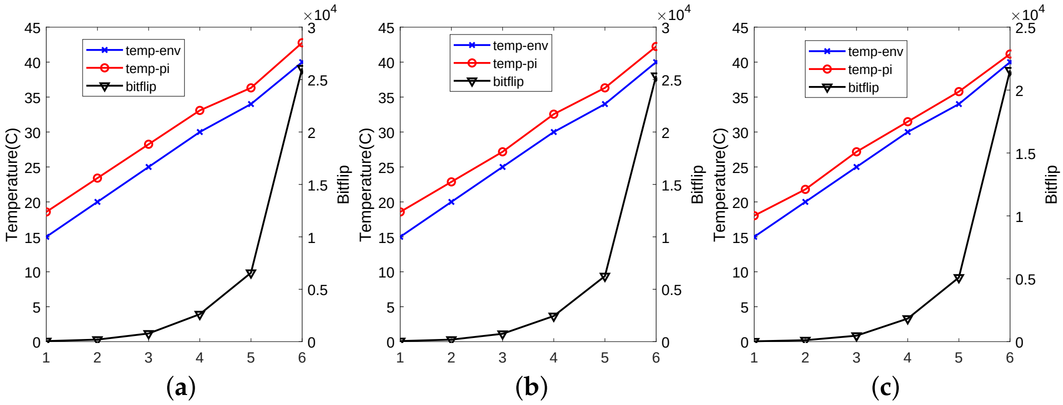


Figure 7. Relation between the temperature, decay time and number of bit flips measured on three Raspberry Pi B+ boards. (**a**) Test results of RPi1. (**b**) Test results of Rpi2. (**c**) Test results of Rpi3.

Figure 8 shows the number of bit flips of each Rpi, environment temperature and SoC temperature. Every point in the plot represents a test result under one temperature conditions. We see that the number of bit flips significantly increases with the temperature rising. However, there are obvious differences in the change of slope rate of the environmental temperature and number of bit flips among all the devices. The curve slope of temperature is very stable compared with the number of bit flips.

Prior work has presented that the decay time (retention time) of DRAM cells decreases exponentially as the temperature increases [46]. In paper [31], the authors computed it by the formula $t'_{T'} = t \cdot e^{-\alpha(T'-T)}$. At temperature $T' > T$, DRAM PUF can generate a similar response under decay time $t' < t$. Furthermore, the number of bit flips is determined by the decay time under a certain temperature. Therefore, we analyse the relationship between temperature and number of bit flips by a simple fitting formula exponentially. The behaviour of bit flips $N_{T'}$ under temperature T' can be computed by known parameters N_T and T by:

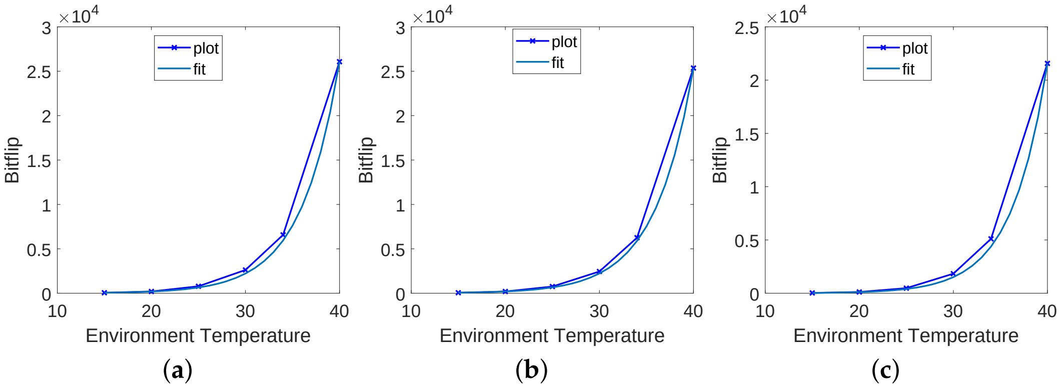
$$N_{T'} = N_T \cdot e^{\alpha(T'-T)} \tag{1}$$



(**a**) (**b**) (**c**)

Figure 8. Test results of Rpi1, Rpi2 and Rpi3. The temperature of thermal chamber, temperature sensor on the systems on chip (SOC) and the number of bit flips (right y-ray) are shown in each plot. The x-ray is the number of iteration of tests under different tests conditions. Due to the heat production of function operation, the curve of temperature sensor on the SoC are always higher than the thermal chamber. (**a**) Test results of RPi1. (**b**) Test results of Rpi2. (**c**) Test results of Rpi3.

Based on our measurements, we estimated α to be 0.2465 for Rpi1, 0.2432 for Rpi2 and 0.2659 for Rpi3. As shown in Figure 9, the smooth fitting curve coincides very well with the original line with "X" label.



(**a**) (**b**) (**c**)

Figure 9. Fitting line of Rpi1, Rpi2 and Rpi3. The smooth curve is the fitting line and the line with the "'X" lable is the original line. (**a**) Fitting line of Rpi1. (**b**) Fitting line of Rpi2. (**c**) Fitting line of Rpi3.

As shown in Table 1, the accuracy of our DRAM PUFs based temperature sensor can be within 4 °C. Due to the reason that the accuracy of our thermal chamber can only be stabilized at 4 °C to 5 °C, we implemented our tests under that range. However, we can still find the big gap between our test results, e.g., from 20 °C to 25 °C, the number of bit flips increased threefold. Therefore, the real accuracy of our proposal should be much better than our test results. Theoretically, if the number of bit flips caused by noise is 200 (this will be shown in the following contents), the temperature accuracy should be less than 3 °C ($T' - T = (ln((188 + 200) \div 188) \div 0.2465$).

Table 1. The number of bit flips under temperature 20 °C, 25 °C, 30 °C, 35 °C, 40 °C, where TR is the test results in thermal chamber and ER is the value evaluated by Equation (1).

Rpi	20 °C		25 °C		30 °C		34 °C		40 °C	
	TR	ER	TR	ER	TR	ER	TR	ER	TR	ER
Rpi1	198	188	781	647	2621	2219	6571	5948	26074	26103
Rpi2	193	196	751	660	2448	2227	6256	5891	25359	25348
Rpi3	117	106	476	400	1835	1511	5103	4378	21569	21584

In our proposal, the uniqueness of DRAM PUFs is an important parameter to make sure that the DRAM PUF can generate a unique ID and key for the sensor. However, the uniqueness evaluation method based on inter-Hamming distance is not suited for DRAM PUF, because the majority cells of DRAM PUF does not flip in short decay time. Therefore, paper [31] proposed to use the Jaccard index to evaluate the uniqueness of multiple DRAM PUF instances. For two sets A and B, the Jaccard index is defined as Equation (2). As shown in Figure 10, the inter-chip Jaccard index is 1.4981×10^{-4}. This small value indicates the high uniqueness of our DRAM PUF on Rpi.

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{2}$$

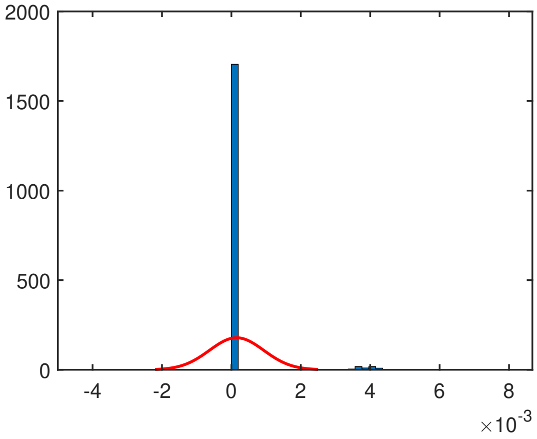


Figure 10. Inter-chip Jaccard index. We evaluate 16MB DRAM as DRAM PUF on each Raspberry Pi board. Given that we do not have enough boards to evaluate the inter-chip Jaccard index, we divided each 16 MB DRAM PUF into 20 parts with the same size. We consider each 0.8MB DRAM PUF as a unique model. Therefore we have 60 DRAM PUF models to operate the Jaccard index evaluation.

For the DRAM PUF-based key generation scheme in Section 4.3, we tested the randomness of the output. As shown in Table 2, the scheme can generate random bits under 40 °C, 30 s or under low temperature with longer decay time calculated by the aforementioned equation $t'_{T'} = t \cdot e^{-\alpha(T'-T)}$.

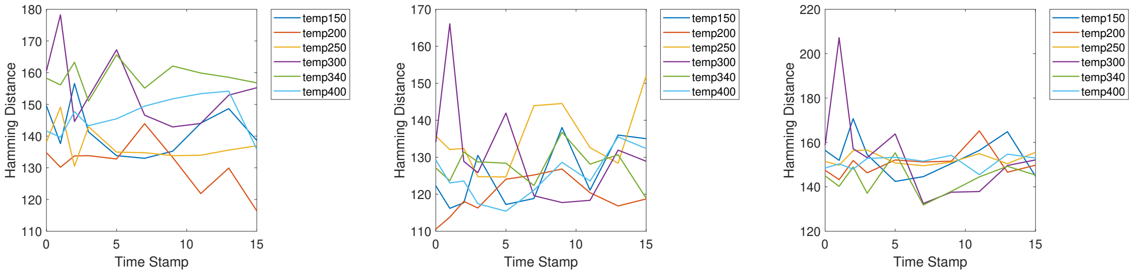
Furthermore, to evaluate the robustness of our proposal, we tested the aging and workload effects of DRAM PUF and analyzed the voltage effects by the related study.

All the accelerated aging experiments are performed using a thermal chamber. Theoretically, one day's test under 80 °C is equal to 18 months of operation under room temperature [47]. Therefore, as shown in Figure 11, after 15 days accelerated aging tests, we can evaluate nearly 270 months of aging effects of DRAM PUFs. Although there are some fluctuations, the test results do not present serious aging effects. Similar conclusions are also shown in paper [47].

Table 2. The randomness test results for the DRAM PUF leveraging NIST test suit for temperature 25 °C, 30 °C, 35 °C, 40 °C under decay time 30 s. It should be noticed that as a weak PUF, the length of the output from the DRAM PUF cannot meet some of the tests in the National Institute of Standards and Technology (NIST) test suit, e.g., the length of the bit strings should be longer than 10^6 for the Rank test. Therefore, we just listed the test results that meet the requirement. (N.O.T. is the Non Overlapping Template. FFT is the Fast Fourier Transform test.)

NIST Tests	**Pi1**				**Pi2**				**Pd i3**			
	25 °C	**30 °C**	**34 °C**	**40 °C**	**25 °C**	**30 °C**	**34 °C**	**40 °C**	**25 °C**	**30 °C**	**34 °C**	**40 °C**
Frequency	-	-	100%	100%	-	-	97%	97%	-	-	99%	99%
Blcok Frequency	-	-	100%	99%	-	-	100%	97%	-	-	100%	99%
Cumulative Sums	-	-	100%	100%	-	-	99%	96%	-	-	98%	99%
Runs	-	-	100%	99%	-	-	99%	97%	-	-	98%	99%
longest Run	-	-	99%	99%	-	-	0%	98%	-	-	0%	99%
FFT	-	-	95%	98%	-	-	96%	99%	-	-	99%	100%
N.O.T.	-	-	47%	100%	-	-	59%	91%	-	-	98%	98%

As shown in paper [48], voltage is one of the parameters that effect the junction leakage current $I_{leak} \propto e^{V_{appliedvoltage}}$. The relationship between the retention time of DRAM and I_{leak} can be expressed as $T_{ret} \propto C_s / I_{leak}$, where C_s is a parameter. Thus, as the increase of voltage, the retention time of DRAM decreased. For a certain decay time, there would be more bit flips. This may effect the robustness of DRAM based key generation scheme and the accuracy of the temperature sensor. Therefore, a stable supply voltage for DRAM chip is necessary for our proposal.



(**a**) (**b**) (**c**)

Figure 11. The aging test resuts of Rpi, Rpi2 and Rpi3. The x-ray is the days of the accelerated aging tests under 80 °C. The y-ray is the Hamming distance between the output of DRAM PUF before and after accelerated aging under different temperature. (**a**) Aging test results of Rpi1. (**b**) Aging test results of Rpi2. (**c**) Aging test results of Rpi3.

Furthermore, the working conditions of SoC may also influence the feature of our design. However, until now, the open source firmware for Raspberry Pi B+ cannot boot up a real operation system. Therefore, in this paper, we only tested the effects of two specific functions that operated on GPU. The difference between the two functions is reliable at 5%. These orders of magnitude are much smaller than the influence of temperature (as shown in Table 1). On the other hand, the SoC can mitigate the influence of workload by control the code operation when used in the field.

6. Discussion

6.1. Security Discussion

Since our intrinsic DRAM PUF relies on the intrinsic DRAM memory, memory protection is the premise for the security of our DRAM PUF-based key storage and sensor scheme. The memory used as PUF must be protected from tampering from all software outside the trust boundary. Furthermore, the privilege to control the register about DRAM decay control is another important security issue. The security system must make sure that only the legal component that is granted the highest privilege can access arbitrary memory without any limitations.

Isolation techniques for multi-core platforms that are based on resource partitioning offer a promising method for this security issue. For example, memory isolation techniques can control memory access rights on an embedded system. It relies on temporal partitioning of memory between trusted and untrusted code to create isolated memory for the execution of sensitive code. It can be used to protect a specified memory space by forbidding any illegal access to the memory address. The isolation techniques have been widely used in the existing computer security scheme, e.g., Intel Trusted Execution Technology (TXT) [49], Intel Software Guard Extensions (SGX) [50] and ARM TrustZone [51].

Although the memory resources that are used for DRAM PUF can be protected by the isolation technology aforementioned, another possible threat for our DRAM PUF-based sensor is the Rowhammer attack [52]. As shown in Figure 12, the attacker could try to introduce some random errors (bit flips) into DRAM PUF by repeatedly accessing adjacent rows which are legal for the attacker. If the DRAM PUFs address is discrete physically, this attack should be very powerful because there will be a lot of adjacent rows that can be used to operate the attack. On the contrary, if the DRAM PUF is physically successive as shown in Figure 12, the attack would be not very useful in our design because the rowhammer attack can only injure the borders of the DRAM PUF area. The number of bit flips introduced by this attack is very limited. Furthermore, two "empty-rows" can be used to isolate the DRAM PUF from the attacker.

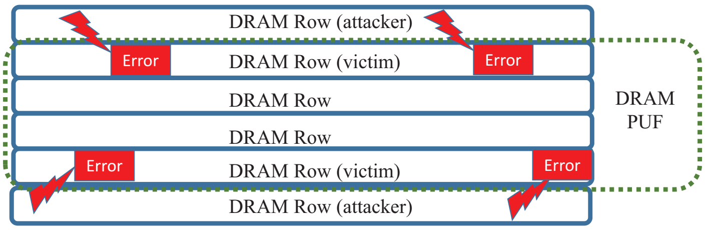


Figure 12. The rowhammer attacks for DRAM PUF.

6.2. Future Works

This novel DRAM PUF-based sensor still leaves a number of open research issues and questions that need to be addressed. The possible future work includes:

- We only verified the feasibility of the DRAM PUF-based sensor. Therefore, more comprehensive tests are necessary.
- The open source firmware [43] used in our implementation cannot boot up a whole operating system now. Therefore, we still are not clear about the influence of the operating code on the feature of the sensor.
- The query process of DRAM PUF needs several seconds of decay time. Therefore, it can not be used in certain real-time scenarios. Our future work will utilize more intrinsic PUF designs to address this issue.

7. Conclusions

In this work, we presented an analysis and categorization of the security key management schemes for sensor networks and PUF sensor designs. Previous works depicted that PUF can provide low-cost key storage and intrinsic sensors to countermeasure the security issues of physical attacks for NVM-based key storage and spoofing attacks. However, the existing PUF sensors can not be used in commodity off-the-shelf devices because of the dedicated circuits of PUF implementation. Our work demonstrates that intrinsic PUFs can be a good candidate to configure the PUF-based key storage and

PUF sensor in the commodity off-the-shelf devices without any hardware changes. An evaluation of the DRAM PUF found on the off-the-shelf commodity device–Rpi B+, showed the feasibility of a DRAM PUF-based temperature sensor. Moreover, we proposed a DRAM PUF-based key storage scheme that can configure the PUF sensor in it. The sensor process can be operated during the key generation process.

Author Contributions: Conceptualization, S.C. and B.L.; methodology, S.C. and Y.C.; software, S.C. and Y.C.; validation, S.C. and Y.C.; investigation, S.C.; resources, S.C.; data curation, S.C.; writing–original draft preparation, S.C.; writing–review and editing, Y.C.; visualization, S.C. and Y.C.; supervision, Y.C. and B.L.; project administration, Y.C.; funding acquisition, B.L.

Funding: This research was funded by Shenzhen Science Technology and Innovation Commission(SCSTI), No. JCYJ20170817115500476, National Nature Science of China(NSFC), No. 61571116 and Scientific Research Foundation of Graduate School of Southeast University, No. YBPY1869.

Acknowledgments: We would like to thank Jakub Szefer and Wenjie Xiong for guidance and help with the PUF implementation on Rpi. The test platform was based on my work at Yale University working with Jakub Szefer and Wenjie Xiong which has been published open-source at http://caslab.csl.yale.edu/code/popchannels/.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schneier, B. The Internet of Things Is Wildly Insecure—and Often Unpatchable. Wired (Online), 6 January 2014. Available online: http://www.wired.com/2014/01/theres-no-good-way-to-patch-the-internet-of-things-and-thats-a-huge-problem/ (accessed on 8 July 2016).
2. Skorobogatov, S.P. Data remanence in flash memory devices. In *International Conference on Cryptographic Hardware & Embedded Systems*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 339–353.
3. Torrance, R.; James, D.L. The State-of-the-Art in IC Reverse Engineering. In *CHES 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 363–381.
4. Shoukry, Y.; Martin, P.; Tabuada, P.; Srivastava, M. Non-invasive Spoofing Attacks for Anti-lock Braking Systems. In *CHES 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 55–72.
5. Gassend, B.; Clarke, D.; van Dijk, M.; Devadas, S. Silicon Physical Random Functions. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 18–22 November 2002; pp. 148–160.
6. Armknecht, F.; Maes, R.; Sadeghi, A.-R.; Sunar, B.; Tuyls, P. Memory leakage-resilient encryption based on physically unclonable functions. In *Towards Hardware-Intrinsic Security*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 135–164.
7. Kursawe, K.; Sadeghi, A.; Schellekens, D.; Skoric, B.; Tuyls, P. Reconfigurable Physical Unclonable Functions-Enabling technology for tamper-resistant storage. In Proceedings of the 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, HOST, San Francisco, CA, USA, 27–27 July 2009; pp. 22–29.
8. Franco, J.J.L.; Boemo, E.; Castillo, E.; Parrilla, L. Ring oscillators as thermal sensors in FPGAs: Experiments in low voltage. In Proceedings of the 2010 VI Southern Programmable Logic Conference, SPL, Ipojuca, Brazil, 24–26 March 2010; pp. 133–137.
9. Shimizu, K.; Sugawara, T.; Suzuki, D.; Srivastava, M.T. PUF as a sensor. In Proceedings of the 2015 IEEE 4th Global Conference on Consumer Electronics GCCE, Osaka, Japan, 27–30 October 2015; pp. 88–92.
10. Ma, H.; Gao, Y.; Kavehei, O.; Ranasinghe, D.C. A PUF sensor. Securing physical measurements. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, Kona, HI, USA, 13–17 March 2017; pp. 648–653.
11. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical One-Way Functions. *Sciences* **2002**, *297*, 2026–2030. [CrossRef] [PubMed]
12. Suh, G.E.; Devadas, S. MPhysical unclonable functions for device authentication and secret key generation. In Proceedings of the Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 9–14.
13. Ghaith, H.; Sunar, B. PUF-HB: A Tamper-Resilient HB Based Authentication Protocol. In *Applied Cryptography & Network Security, International Conference*; Springer: Berlin/Heidelberg, Germany, 2008.

14. Van Herrewege, A.; Katzenbeisser, S.; Maes, R.; Peeters, R.; Sadeghi, A.R.; Verbauwhede, I.; Wachsmann, C. Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs. In Proceedings of the Financial Cryptography & Data Security-International Conference, Kralendijk, Bonaire, 27 Februray–2 March 2012.
15. Majzoobi, M.; Koushanfar, F.; Devadas, S. FPGA PUF using programmable delay lines. In Proceedings of the IEEE International Workshop on Information Forensics & Security, Seattle, WA, USA, 12–15 December 2011.
16. Maiti, A.; Casarona, J.; McHale, L.; Schaumont, P. A large scale characterization of RO-PUF. In Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 13–14 June 2010; pp. 94–99.
17. Selimis, G.; Konijnenburg, M.; Ashouei, M.; Huisken, J.; de Groot, H.; van der Leest, V.; Schrijen, G.-J.; van Hulst, M.; Tuyls, P. Evaluation of 90 nm 6T-SRAM as physical unclonable function for secure key generation in wireless sensor nodes. In Proceedings of the IEEE International Symposium on Circuits and Systems, Rio de Janeiro, Brazil, 15–18 May 2011; pp. 567–570.
18. Rosenblatt, S.; Chellappa, S.; Cestero, A.; Robson, N.; Kirihata, T.; Iyer, S.S. A self-authenticating chip architecture using an intrinsic fingerprint of embedded DRAM. *IEEE J. Solid-State Circuits* **2013**, *48*, 2934–2943. [CrossRef]
19. Serpanos, D.N.; Voyiatzis, A.G. Security challenges in embedded systems. *Trans. Embedded Comput. Syst. TECS* **2013**, *12*, 66. [CrossRef]
20. Momani, M.; Challa, S.; Alhmouz, R. Can we trust trusted nodes in wireless sensor networks? In Proceedings of the 2008 International Conference on Computer and Communication Engineering, ICCCE, Kuala Lumpur, Malaysia, 13–15 May 2008; pp. 1227–1232.
21. Momani, M.; Challa, S. Survey of Trust Models in Different Network Domains. *Int. J. Ad Hoc Sens. Ubiq. Comput.* **2010**, *1*, 1. [CrossRef]
22. Zhang, J.; Varadharajan, V. Wireless sensor network key management survey and taxonomy. *J. Netw. Comput. Appl.* **2009**, *33*, 63–75. [CrossRef]
23. He, X.; Niedermeier, M.; De Meer, H. Dynamic key management in wireless sensor networks: A survey. *J. Netw. Comput. Appl.* **2013**, *36*, 611–622. [CrossRef]
24. Viega, J.; Thompson, H. The state of embedded-device secu-rity (Spoiler alert: It's bad). *IEEE Secur. Privacy* **2012**, *10*, 68–70. [CrossRef]
25. Arulraj, J.; Pavlo, A.; Dulloor, S.R. Let's talk about storage & recovery methods for non-volatile memory database systems. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Australia, 31 May–4 June 2015; pp. 707–722.
26. Naor, M.; Segev, G. Public-Key Cryptosystems Resilient to Key Leakage. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2009; pp. 18–35.
27. Posch, R. Protecting devices by active coating. *J. Univ. Comput. Sci.* **1998**, *4*, 652–668.
28. Gaubatz, G.; Sunar, B.; Karpovsky, M.G. Non-linear Residue Codes for Robust Public-Key Arithmetic. In Proceedings of the Fault Diagnosis & Tolerance in Cryptography, Third International Workshop, FDTC, Yokohama, Japan, 10 October 2006.
29. Anderson, R.; Kuhn, M. Tamper Resistance—A Cautionary Note. In Proceedings of the Conference on Second Usenix Workshop on Electronic Commerce USENIX Association, Oakland, CA, USA, 18–20 November 1996.
30. Tuyls, P.; Škorić, B. Secret key generation from classical physics: Physical uncloneable functions. In *AmIware Hardware Technology Drivers of Ambient Intelligence*; Springer: Dordrecht, The Netherlands, 2006; pp. 421–447.
31. Schaller, A.; Xiong, W.; Anagnostopoulos, N.A.; Saleem, M.U.; Gabmeyer, S.; Skoric, B.; Katzenbeisser, S.; Szefer, J. Decay-Based DRAM PUFs in Commodity Devices. *IEEE Trans. Dependable Secure Comput.* **2018**, *16*, 462–475. [CrossRef]
32. Chen, X.; Makki, K.; Yen, K.; Pissinou, N. Sensor network security: A survey. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 52–73. [CrossRef]
33. Parno, B.; Perrig, A.; Gligor, V. Distributed detection of node replication attacks in sensor networks. In Proceedings of the 2005 IEEE Symposium on Security and Privacy, S & P, Oakland, CA, USA, 8–11 May 2005; pp. 49–63.
34. Rosenfeld, K.; Gavas, E.; Karri, R. Sensor physical unclonable functions. In Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST, Anaheim, CA, USA, 13–14 June 2010; pp. 112–117.

35. Cao, Y.; Zhang, L.; Chang, C. Using image sensor PUF as root of trust for birthmarking of perceptual image hash. In Proceedings of the 2016 IEEE Asian Hardware-Oriented Security and Trust, Asian HOST, Anaheim, CA, USA, 13–14 June 2016; pp. 1–6.
36. Roel Maes, L. Physically Unclonable Functions: Constructions, Properties and Applications. In *Properties and Applications*, 1st ed.; Springer Publishing Company: Berlin, Germany, 2016; p. 193.
37. Tang, J.; Karri, R.; Rajendran, J. Securing pressure measurements using SensorPUFs. In Proceedings of the 2016 IEEE International Symposium on Circuits and Systems, ISCAS, Montreal, QC, Canada, 22–25 May 2016; pp. 1330–1333.
38. Sun, J.; Bittner, R.; Eguro, K. FPGA side-channel receivers. In Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays, International Symposium on FPGA, Monterey, CA, USA, 27 February–1 March 2011; pp. 267–276.
39. Iakymchuk, T.; Nikodem, M.; Kępa, K. Temperature-based covert channel in FPGA systems. In Proceedings of the 6th International Workshop on Reconfigurable Communication-Centric Systems-on-Chip, ReCoSoC, Montpellier, France, 20–22 June 2011; pp. 1–7.
40. Tian, S.; Szefer, J. Temporal Thermal Covert Channels in Cloud FPGAs. In Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Seaside, CA, USA, 24–26 February 2019.
41. Tehranipoor, F.; Karimina, N.; Xiao, K.; Chandy, J. DRAM based intrinsic physical unclonable functions for system level security. In Proceedings of the Great Lakes Symposium on VLSI, Pittsburgh, PA, USA, 20–22 May 2015; pp. 15–20.
42. Sutar, S.; Raha, A.; Raghunathan, V. D-PUF: An intrinsically reconfigurable DRAM PUF for device authentication in embedded systems. In Proceedings of the IEEE International Conference on Compilers, Architectures, and Sythesis of Embedded Systems, CASES, Pittsburgh, PA, USA, 2–7 October 2016; pp. 1–10.
43. Badea, L.; Rosenzweig, A.; Brooks, K. rpi-open-firmware. Available online: https://github.com/christinaa/rpi-open-firmware (accessed on 5 February 2019).
44. Chen, S.; Xiong, W.; Xu, Y.; Li, B.; Szefer, J. Thermal Covert Channels Leveraging Package-On-Package DRAM. In Proceedings of the International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Rotorua, New Zealand, 5–8 August 2019.
45. Delvaux, J.; Gu, D.; Schellekens, D.; Verbauwhede, I. Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* **2015**, *34*, 889–902. [CrossRef]
46. Liu, J.; Jaiyen, B.; Kim, Y.; Wilkerson, C.; Mutlu, O. An experimental study of data retention behavior in modern DRAM devices. In Proceedings of the ACM SIGARCH Computer Architecture News, Tel-Aviv, Israel, 23–27 June 2013; Volume 41.
47. Tehranipoor, F.; Karimian, N.; Yan, W.; Chandy, J.A. Investigation of DRAM PUFs reliability under device accelerated aging effects. In Proceedings of the IEEE International Symposium on Circuits & Systems, Baltimore, MD, USA, 28–31 May 2017.
48. Hamamoto, T.; Sugiura, S.; Sawada, S. On the retention time distribution of dynamic random access memory (DRAM). *IEEE Trans. Electron Devices* **1998**, *45*, 1300–1309. [CrossRef]
49. INTEL CORPORATION. Intel® Trusted Execution Technology Measured Launched Environment Programming Guide. Available online: https://www.intel.sg/content/www/xa/en/software-developers/intel-txt-software-development-guide.html (accessed on 13 August 2016).
50. McKeen, F.; Alexandrovich, I.; Berenzon, A.; Rozas, C.V.; Shafi, H.; Shanbhogue, V.; Savagaonkar, U.R. Innovative Instructions and Software Model for Isolated Execution. *Int. Workshop Hardw. Arch. Support Secur. Privacy* **2013**, *10*, 10.
51. ARM. Building a Secure System using TrustZone Technology. 2009. Available online: http://www.arm.com (accessed on 1 May 2009).
52. Kim, Y.; Daly, R.; Kim, J.; Fallin, C.; Lee, J.H.; Lee, D.; Wilkerson, C.; Lai, K.; Mutlu, O. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. In Proceedings of the ACM SIGARCH Computer Architecture News, Minneapolis, MN, USA, 14–18 June 2014; pp. 361–372.