

Article

A Trustworthy Key Generation Prototype Based on DDR3 PUF for Wireless Sensor Networks

Wenchao Liu, Zhenhua Zhang, Miaoxin Li and Zhenglin Liu *

School of Optical and Electronic Information, Huazhong University of Science and Technology, No. 1037 Luoyu Road, Wuhan 430074, China; E-Mails: saintlawhd@gmail.com (W.L.); zzh373576922@gmail.com (Z.Z.); l.alveira@gmail.com (M.L.)

* Author to whom correspondence should be addressed; E-Mail: liuzhenglin@hust.edu.cn; Tel.: +86-27-8754-1768; Fax: +86-27-8761-1255.

Received: 12 March 2014; in revised form: 15 May 2014 / Accepted: 25 June 2014 /

Published: 30 June 2014

Abstract: Secret key leakage in wireless sensor networks (WSNs) is a high security risk especially when sensor nodes are deployed in hostile environment and physically accessible to attackers. With nowadays semi/fully-invasive attack techniques attackers can directly derive the cryptographic key from non-volatile memory (NVM) storage. Physically Unclonable Function (PUF) is a promising technology to resist node capture attacks, and it also provides a low cost and tamper-resistant key provisioning solution. In this paper, we designed a PUF based on double-data-rate SDRAM Type 3 (DDR3) memory by exploring its memory decay characteristics. We also described a prototype of 128-bit key generation based on DDR3 PUF with integrated fuzzy extractor. Due to the wide adoption of DDR3 memory in WSN, our proposed DDR3 PUF technology with high security levels and no required hardware changes is suitable for a wide range of WSN applications.

Keywords: WSN; PUF; DDR3; key generation; fuzzy extractor

1. Introduction

WSNs are vulnerable to various attacks in hostile environments [1,2]. Within a typical node capture attack, an adversary can gain full control over the captured sensor nodes. Moreover, all contents of the non-volatile memory, including cryptographic secrets and other valuable data can be extracted through

semi/fully-invasive physical attacks (e.g., micro-probing, reverse engineering *etc.* [3]). The cryptographic key leakage often leads to the failure of the whole security mechanism [4]. Therefore, a tamper-resistant and cost-efficient secure key provisioning scheme against node capture attack is very important to WSN.

Physically unclonable function (PUF) is a collection of functional constructions which meets a number of naturally identifiable characteristics such as randomness, uniqueness, physical unclonability, and possibly tamper-resistant [5]. The PUF presents a unique mapping between challenges (inputs) and responses (outputs) based on the device's hardware-intrinsic features. Those features are originated from the subtleties of the operating conditions as well as random variations that are imprinted into the device during the manufacturing process. Any physical attacks will destroy the inherent physical features with very high probability. Consequently, the response would also be destroyed.

There are several advantages over using PUF for secret key generation and storage. The PUF is uniquely linked to a given device such that one cannot reproduce it or manufacture a device with a precise key. In addition, the PUF avoids the need for storing secret keys in non-volatile memory as keys can be extracted from the device only when required. Moreover, PUF could be tamper-resistant, which is innate immunity to node capture attacks.

However, the PUF responses cannot be used directly as a cryptographic key. Due to the influence of environmental noises, the responses are likely to be slightly different for each evaluation for the same challenge. On the other hand, the PUF responses may not be uniformly distributed. To derive a unique and almost uniform key from PUF responses, a fuzzy extractor can be used [6].

1.1. Previous Work

Pappu *et al.* initially introduced the idea of PUF [7,8] based on the response (scattering) obtained when shining a laser on a bubble-filled transparent epoxy wafer. A number of experiments were performed to test the characteristics of the constructed PUF. The inter- and intra-distance measures were evaluated for the obtained Gabor hashes. This resulted in an average inter-distance $\mu_{\text{inter}} = 49.79\%$ ($\sigma_{\text{inter}} = 3.3\%$) and an average intra-distance $\mu_{\text{intra}} = 25.25\%$ ($\sigma_{\text{intra}} = 6.9\%$). The implement of optical PUF is rather laborious that a laser and a tedious mechanical positioning system are needed.

Gassend *et al.* proposed silicon PUF [9], which used manufacturing process variations in ICs with identical masks to uniquely characterize each IC. The statistical delay variations of transistors and wires in the IC were used to create a parameterized self-oscillating circuit and its frequency measurements are collected to characterize each IC. The integration of PUF into the underlying manufacturing variability of ICs makes it much more suitable for practical application.

Ring oscillator PUF, as introduced in [10], use a different approach toward measuring small random delay deviations caused by manufacturing variations. The output of a digital delay line is inverted and fed back to its input, creating an asynchronously oscillating loop. Due to random manufacturing variations in the circuit latency, the exact frequency will also be random and unique per device. They tested a ring oscillator PUF with division compensation on four FPGA devices obtaining $\mu_{\text{inter}} \approx 10 \times 10^{-3}$ and $\mu_{\text{intra}} \approx 0.1 \times 10^{-3}$ with measurements taken over a 25 °C temperature interval.

The flip-flop PUF was introduced by Maes *et al.* in [11]. The presented measurement results show that the entropy in the measured flip-flop power-up values is rather limited due to the presence of a strong bias, namely the strong preference of most flip-flops to power up with a zero value.

SRAM PUF was proposed in [12], originates from the mismatch of the symmetry of the memory cell, which is again caused by silicon manufacturing variability. The responses of SRAM PUF can be obtained from the power-up value of uninitialized SRAM cells. In [13], extensive experiments on SRAM PUFs were performed by collecting the power-up state of 8190 bytes of SRAM from different memory blocks on different FPGAs. The results show an average inter-chip Hamming distance of $\mu_{\text{inter}} = 49.97\%$ between two different blocks and the average intra-chip Hamming distance of $\mu_{\text{intra}} = 3.57\%$ within multiple measurements of a single block under typical environment and $\mu_{\text{intra}} < 12\%$ for large temperature deviations.

The PUF properties for 28-nm process field-programmable gate arrays (FPGAs) are examined by Hori *et al.* in [14]. In their work, the performance of arbiter PUF and delay-based PUF are evaluated on Kintex-7 FPGAs on 10 SASEBO-GIII boards and Artix-7 on eight Nexys4 boards. A performances comparison with those on the 45-nmSpartan-6 and 65-nm Virtex-5 were analyzed. The results show that both APUFs and PL-PUFs on the 28-nm FPGAs demonstrated good properties that were no worse than those on 45- and 65-nm FPGAs. Therefore, the PUFs can be implemented using the state-of-the-art process technology and used for various security-sensitive applications.

Among all the proposed silicon PUF instantiations in literature, memory-based intrinsic PUF is the most promising solution for secure key provisioning scheme in WSNs. Memories are standard components in sensor nodes. Cryptographic keys can be generated without any change to the existing hardware architecture. DRAM PUF [15,16] utilizes the decay signature of the memory cells with one transistor/one capacitor structure. The manufacturing process introduces miniscule variations to the decay features of these cells. Keller [15] customized a DDR3 controller to disable the refresh of the DRAM cells for PUF usage. Since most sensor nodes are not reconfigurable, it is hard to reproduce the PUF response without the customized DDR3 controller. Rosenblatt [16] introduced a random intrinsic chip ID generation method using decay fails in 32 nm SOI embedded DRAM. The intrinsic feature was detected by adjusting the wordline low voltage of embedded DRAM. The adjustment is still hard to realize without changing the circuits of the sensor nodes. In general, existing DRAM PUF solutions are not applicable to resource constrained WSN without hardware modifications.

1.2. Our Contribution

To study the decay feature of DDR3, a proper test system has been developed and set up. The key generation skeleton based on DDR3 PUF is established, which is suit for WSN. The challenges are power-switch time as well as the address and offset of memory cells with logic 1 before power off while the responses are power-on value of those cells after specific decay interval. According the experimental result, an efficient prototype of 128-bit key generation based on DDR3 PUF in conjunction with fuzzy extractor is proposed. Some implementation specific details as well as the PUF key performance are discussed.

1.3. Paper Outline

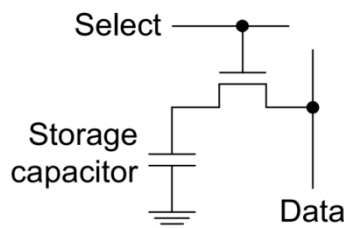
The rest of this paper is organized as follows: Section 2 explores the decay feature of DDR3 and describes the key generation framework. The experimental setup and DDR3 decay characterization are presented in Section 3. The post-processing process for reliable key generation is discussed in Section 4. Section 5 concludes the paper.

2. DDR3 PUF Exploration

2.1. Decay Feature of DDR3

The foundation of DDR3 PUF lies in the underlying decay feature of DRAM cells. A DRAM cell consists of a capacitor connected by a pass transistor to the bit line as shown in Figure 1. The presence or absence of charge in the capacitor determines whether the cell contains a “1” or “0”. This charge would leak off the capacitor due to the sub-threshold current of the cell transistor. The leaky nature varies among DRAM cells due to the miniscule variations introduced by manufacturing process. To prevent the charge leakage, a refresh operation is required which re-charges the capacitor to hold its value.

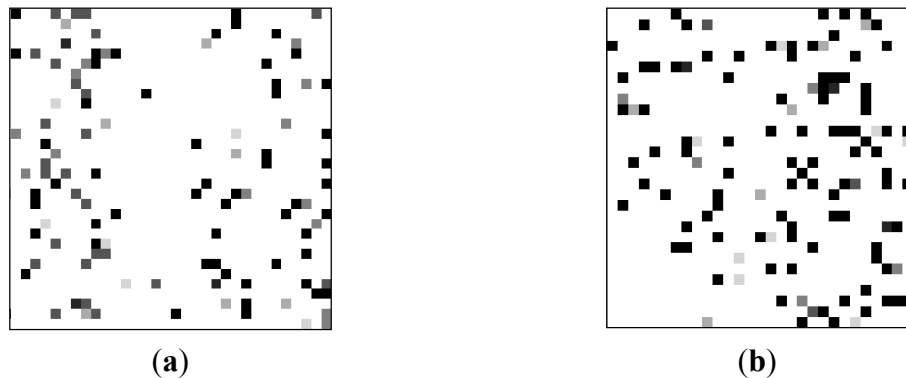
Figure 1. DRAM cells structure.



To explore the decay feature, a proper test scheme has been developed to simulate typical WSN application scenarios in which the sensor nodes are often intermittently-powered. We write a “1” to every cell in a DRAM region, represented by the charge kept in the capacitors. Then, we power off the DRAM to terminate the refresh operation. After a pause, we cycle the DRAM power. Each cell is read and recorded as PASS if the charge has not leaked to logic “0” and FAIL if the charge has leaked to logic “0”. We repeat the above operations multiple times to minimize the influence of the environmental noises. To show decay feature of each DDR3 cells in an intuitional way, we transforms test data to a grayscale images illustrated in Figure 2.

As shown in Figure 2, the pixel position stands for the address of each DRAM cell. The decay feature is represented by the grayscale of the associated pixel which is equal to the total number of the failed test times divided by iteration times. The grayscale signifies the preference of decay feature in each cell. For example, a black pixel means the associated DRAM cell is always leaked to logic “0” during the interval time between power switching; while a white pixel means the leakage never happens during the interval. The decay feature between different DDR3 cells is different. This is hold for different DDR3 devices by comparing Figure 2a to Figure 2b. Thus the distribution of decay feature is unique.

Figure 2. DRAM cells decay feature with power-switch interval time of 120 s: (a) DDR3 device 1; (b) DDR3 device 2.



The decay feature also varies with the interval time. If it is too short, almost no changes have occurred. If given enough time, the charges will vanish and no information can be extracted. Therefore, the optimal interval time should be chosen based on extensive profiling process. This innate decay feature enables DDR3 to be used as a PUF instantiation.

2.2. Key Generation Framework

The PUF responses cannot be used directly as a cryptographic key for two reasons. First, due to the influence of the existing environmental noises, the responses from the same DDR3 PUF could be slightly different each time. Second, the keys directly extracted from the DDR3 PUF responses are not uniformly distributed. Fuzzy extractor is used to as a necessary post-processing step of the PUF responses. The fuzzy extractor with an error correct code (ECC) module and a universal hash module work together in two phases: enrolment and key reconstruction. First, we introduce some terms [17] used in post-processing algorithm.

Statistical distance: Let X and Y be two random variables with range U . Then the *statistical distance* between X and Y is defined as:

$$\Delta(X, Y) \equiv \frac{1}{2} \sum_{u \in U} |P[X = u] - P[Y = u]| \quad (1)$$

If two distributions have statistical distance of (at most) $\varepsilon \geq 0$, they are ε -close:

$$X \approx_{\varepsilon} Y \Leftrightarrow \Delta(X, Y) \leq \varepsilon \quad (2)$$

Min-entropy: Let X be a random variable. The Min-entropy of X is defined as:

$$H_{\infty}(X) \equiv \min_{u \in U} \{-\log(P[X = u])\} = -\log\left(\max_{u \in U} P[X = u]\right) \quad (3)$$

We say that X is a k -source if $H_{\infty}(X) \geq k$.

Strong extractor: Let the seed U_d be uniformly distributed on $\{0,1\}^d$. We say that a function $\text{Ext}: \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ is a (k, ε) strong random extractor if, for all random variables X in $\{0,1\}^n$ independent of U_d with $H_{\infty}(X) \geq k$:

$$(Ext(X, U_d), U_d) \approx_{\varepsilon} (U_m, U_d) \quad (4)$$

where U_m is uniformly distributed in $\{0,1\}^m$ independently of X and U_d .

Universal hash [18]: Let H be a finite collection of hash functions from $\{0,1\}^n$ to $\{0,1\}^m$, we say that H is universal hash if, for every $x, y \in \{0,1\}^n$ with $x \neq y$:

$$P_{h \in H}[h(x) = h(y)] \leq 2^{-m} \quad (5)$$

Leftover hash lemma: Let X be a random variable with universe U and $H_{\infty}(X) \geq k$. Fix $\varepsilon > 0$. Let H be a universal hash family of size 2^d with output length $m = k - 2 \log(1/\varepsilon) + O(1)$. Define:

$$Ext(x, h) = h(x) \quad (6)$$

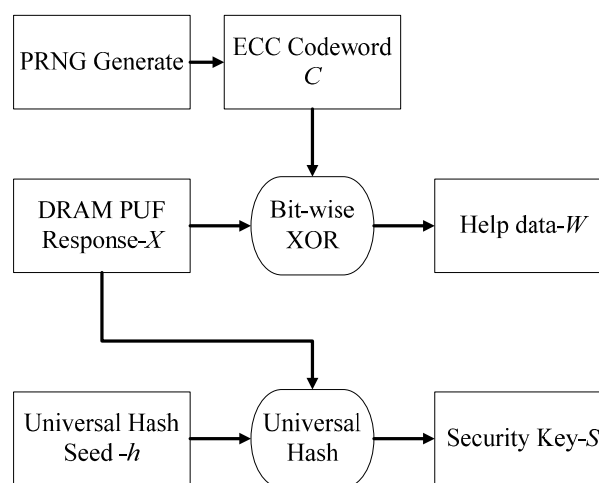
Then Ext is a strong $(k, \varepsilon/2)$ extractor with seed length d and output length m .

The Statistical distance between random variables X and a uniformly distributed Y reflects the randomness extent of X . The min-entropy provides a lower bound on the amount of randomness contained in the decay feature of the DDR3. The leftover hash lemma (LHL) shows how to construct extractors from universal hash families. Moreover, we can exactly calculate how many random bits can be derived from a response with $H_{\infty}(X) \geq k$ by fixing the statistical distance.

2.2.1. Enrolment Phase

The flow chart of enrolment phase is shown in Figure 3. Responses X measured from DDR3 decay as well as a seed h are sent to the universal hash module to get an almost uniformly distributed security key S . At the same time, a help data W is derived from bitwise XOR operation between the response X and ECC codeword C . The output of a PRNG Generate with uniform distribution is encoded to the ECC codeword C . The help data W and seed h will be saved and used in the key reconstruction process later.

Figure 3. The flow chart of enrolment phrase.



The mathematical expression of enrolment phase is shown in Equation (7):

$$\begin{cases} X \oplus C = W \\ \{X, h\} \xrightarrow{\text{Universal Hash}} S \end{cases} \quad (7)$$

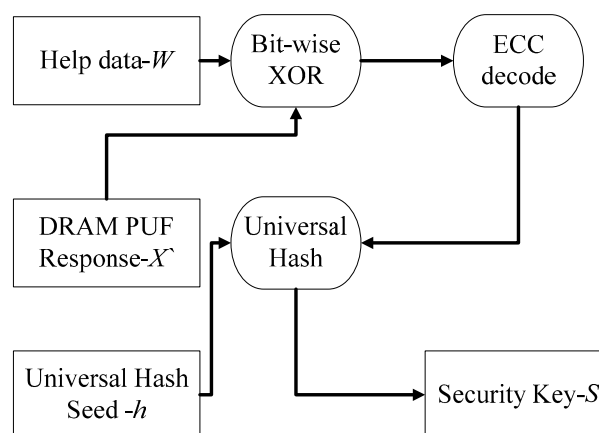
Our goal is to extract almost uniformly distributed key from a response. The response X can be considered as a random variable due to the unpredictability. Our assumption is that we do not know the exact distribution but we know it comes from some class D of distributions. We would like D to be as general as possible, while still being able to extract good random bits.

The storage of the universal hash seed h is safe as it is independent to security key S . There is certain risk to store the helper data W . When the attacker get the W , the response X might be extracted by performing bitwise XOR operation between W and all possible ECC codeword. However, if the min-entropy of random ECC codeword is much larger the response, the leakage is safe. Thus those data can be stored in non-volatile memory without being encrypted. A more detailed procedure will be discussed in Section 4.

2.2.2. Key Reconstruction Phase

The flow chart of key reconstruction phase is shown in Figure 4. A new response X' which might be slightly different from the response in enrolment phase is measured. Then, the bitwise XOR operation is performed between X' and the helper data W . An ECC decode module is applied to the result. The ECC output as well as the seed h is input to universal hash module. Finally, the security key S is reproduced.

Figure 4. The flow chart of key reconstruction phase.



The mathematical expression of the key reproduce phase is described in Equations (8):

$$\left\{ \begin{array}{l} X' \oplus W = C' \\ C' \xrightarrow{\text{ECC decond}} C \\ e = C' \oplus C \\ X' \oplus e = X \\ \{X, h\} \xrightarrow{\text{Universal Hash}} S \end{array} \right. \quad (8)$$

An error version ECC codeword C' is calculated by bitwise XOR operation between X' helper data W . Then an ECC decode procedure is performed to correct errors in C' . If the error correct ability of ECC is stronger compared with the difference between X' and X , the ECC codeword C used in enrolment phase can be decoded. An error vector e is equal to C bitwise XOR with C' . X can be

retrieved by bitwise XOR between e and X' . The security key S can be reproduced by input the retrieved X as well as the seed h into universal hash module.

3. Experimental Setup and Results

3.1. Metrics

To quantify the experimental results, several metrics are defined. The DDR3 PUF response is described in the form of binary string $\{0,1\}^n$. Hamming distance is equal to the number of positions at which the two responses are different in terms of binary string. Normalized Hamming distance is equal to Hamming distance divided by the length of the response binary string times 100%.

Normalized Inter-Hamming distance is equal to normalized Hamming distance between two responses obtained from different DDR3 PUF. This measure expresses how easy it is to distinguish two DDR3 based on their decay feature. In general, it measures the uniqueness of the PUF responses with respect to the entire device population. Its value should be as close to 50% as possible.

Normalized intra-Hamming distance is equal to normalized Hamming distance between two responses obtained from the same DDR3 PUF. This measure reflects the degree of similarity between the two responses. Its value should be much small than normalized Inter-Hamming distance.

Decay rate is the number of “0” in a response binary string divided by the length of the response binary string times 100%.

Normalized min-entropy is equal to the min-entropy obtained from responses of the same DDR3 PUF divided by the length of the response binary string. The min-entropy is calculated by the method in [12]. Normalized min-entropy is a measure of the randomness in the responses.

3.2. Experimental Platform and Setup

The feasibility of DDR3 PUF is validated through experiments on a Xilinx VC709 evaluation board. The decay feature of two Micron DDR3 SODIMM and two Kingston DDR3 SODIMM is tested. The DDR3 is labeled as M1, M2, K1 and K2 respectively. The Built-In Self-Test (BIST) [19] example project in Xilinx EDK with DDR3 IP core can be used to access the DDR3 devices. The system memory map of BIST project is shown in Figure 5.

Figure 5. System memory map of BIST project.

microblaze_0's Address Map				
microblaze_0_d_bram_ctrl	C_BASEADDR	0x00000000	0x00001FFF	8K
microblaze_0_i_bram_ctrl	C_BASEADDR	0x00000000	0x00001FFF	8K
Push_Buttons_5Bits	C_BASEADDR	0x40000000	0x4000FFFF	64K
LEDs_8Bits	C_BASEADDR	0x40020000	0x4002FFFF	64K
DIP_Switches_8Bits	C_BASEADDR	0x40040000	0x4004FFFF	64K
RS232_Uart_1	C_BASEADDR	0x40400000	0x4040FFFF	64K
IIC_EEPROM	C_BASEADDR	0x40800000	0x4080FFFF	64K
microblaze_0_intc	C_BASEADDR	0x41200000	0x4120FFFF	64K
debug_module	C_BASEADDR	0x41400000	0x4140FFFF	64K
axi_timer_0	C_BASEADDR	0x41C00000	0x41C0FFFF	64K
Linear_Flash	C_S_AXI_MEM0_BASEADDR	0x48000000	0x4FFFFFFF	128M
axi_bram_ctrl_0	C_S_AXI_BASEADDR	0xA0000000	0xA003FFFF	256K
DDR3_SDRAM	C_S_AXI_BASEADDR	0xC0000000	0xFFFFFFFF	1G

We first write a 1 to every cell from 0xD000 0000 to 0xD002 0000 of the DDR3. Then, we power off the evaluation board to terminate the memory refresh. After a pause, we power on the board. Each cell from 0xD000 0000 to 0xD002 0000 is read out. The experiments are repeated under 8 different intervals between power-switches. The specific interval time and experiment repeat times are list in Table 1.

Table 1. Experimental Setup.

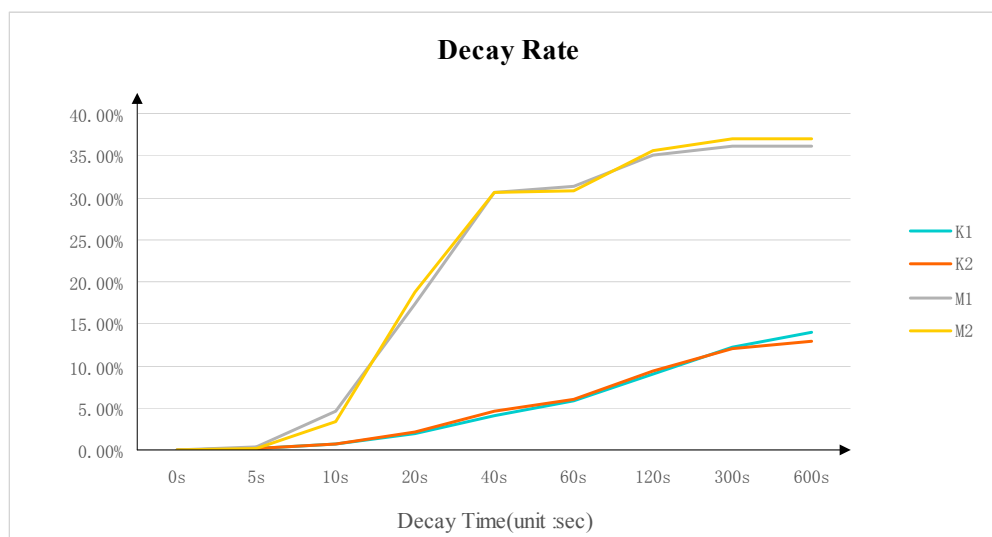
Group ID	Interval Time (unit: s)	Repeat Times (unit: repeat times per each DDR3 device)
1	5	50
2	10	50
3	20	50
4	40	50
5	60	50
6	120	50
7	300	50
8	600	50

The mean values of decay rate, normalized inter- and intra-Hamming distance and normalized min-entropy over 50 times experimental results at each intervals are analyzed.

3.3. Decay Rate

The decay rate of DDR3 is shown in Figure 6. It displays significant differences between the two brands in decay trend and decay speed. The decay rate from the same DDR3 brand and model is almost identical. The Micron DDR3 decayed quickly after 10 s and the decay rate becomes stable around 35% after 120 s. The decay trend of Kingston DDR3 is relatively more stable than the Micron's. The decay rate is around 12% after 300 s.

Figure 6. Decay rate for different DDR3 SODIMM.



3.4. Normalized Inter- and Intra-Hamming Distance

Figure 7 shows the average normalized intra-Hamming distance of the same DDR3 from multiple experiments at different power-switch time. Similar to the decay rate, the intra-Hamming distance rate varies between different brands. The average normalized intra-Hamming distance of Micron DDR3 is larger than that of the Kingston DDR3. Micron DDR3's average normalized intra-Hamming distance is around 3.6%; while Kingston DDR3's intra-Hamming distance rate is around 1.3%.

Figure 7. Normalized intra-Hamming distance for 4 DDR3 devices.

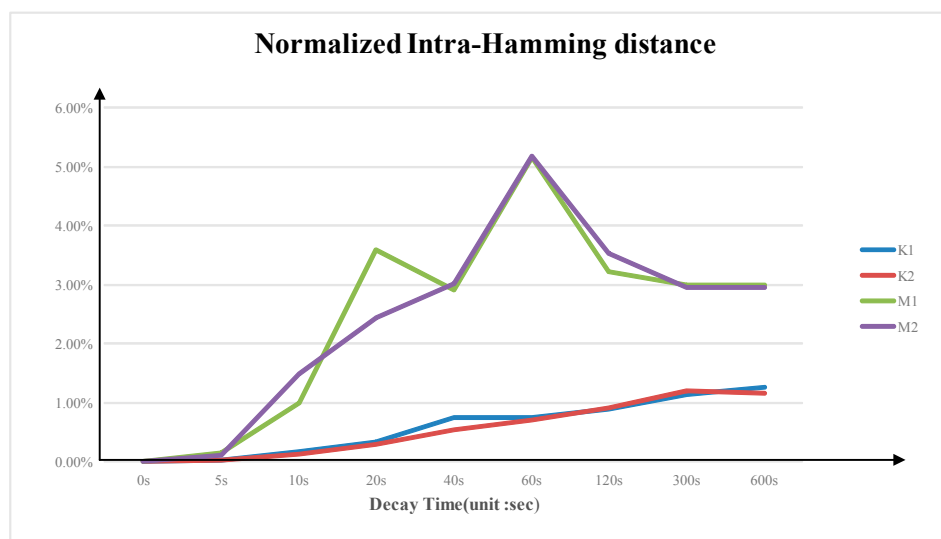
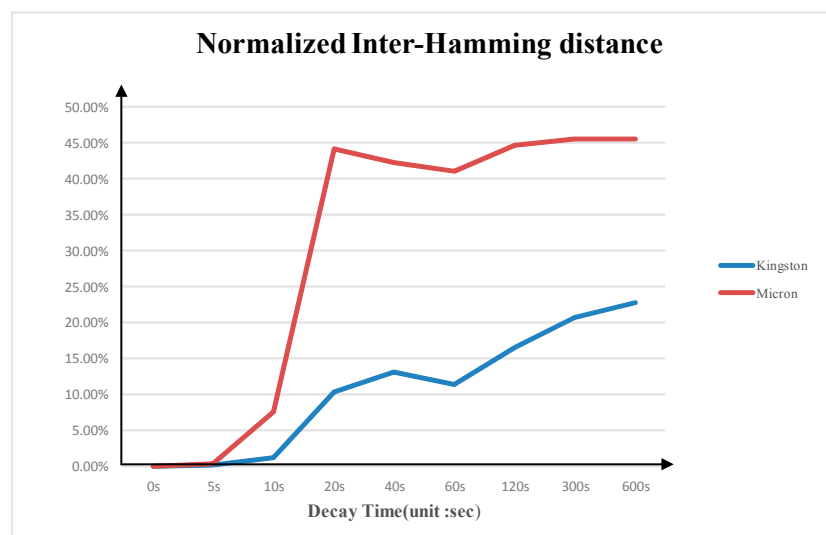


Figure 8 shows the average normalized inter-Hamming distance between different DDR3 with the same brand from multiple experiments at different power-switch time. The average normalized inter-Hamming distance of Micron DDR3 is larger than that of the Kingston DDR3. The average normalized inter-Hamming distance between two Micron DDR3s is around 45% and around 23% for the Kingston.

Figure 8. Normalized inter-Hamming distance between DDR3 devices with the same brand.

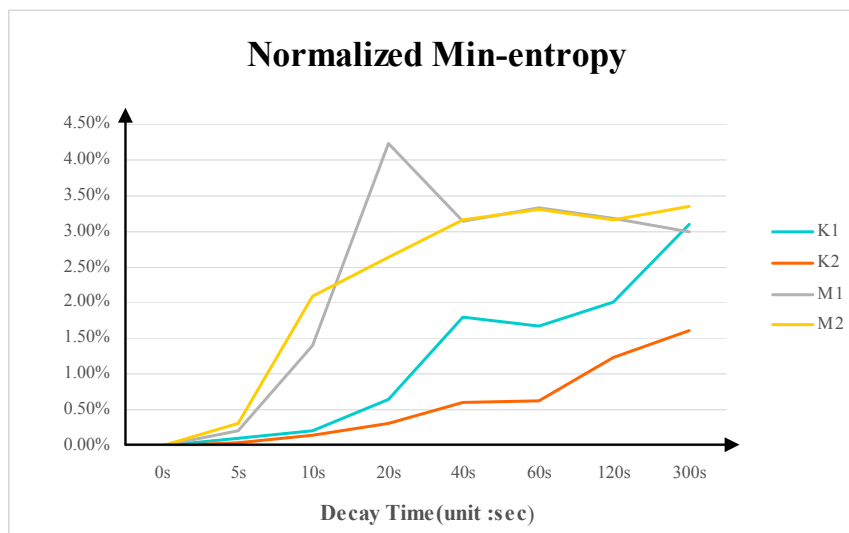


According to the test results, we validate the unique of DDR3 PUF with the fact that the average normalized intra-Hamming distance is much smaller than the inter-Hamming distance rate. By choosing the proper ECC code with error correction ability between the intra-Hamming distance rate and the inter-Hamming distance rate the DDR3 PUF response used in enrolment phase can be retrieved from the measured response.

3.5. Normalized Min-Entropy

Figure 9 shows the normalized min-entropy for the tested DDR3 memories at different power-switch time. The trend of normalized min-entropy is stable when the decay trend is stable. Micron DDR3's normalized min-entropy is around 3.3% and K1 normalized min-entropy is about 3% while the value for K2 is around 1.5%.

Figure 9. The normalized min-entropy for the tested DDR3 memories at different power-switch time.



Normalized min-entropy quantifies the randomness of the PUF responses. The result shows that the DDR3 PUF is not uniformly distributed. In order to generate a 128-bit uniform key, the Min-entropy of DDR3 PUF should be at least 128. This means the number of cells used in DDR3 PUF is equal to 128 divided by normalized min-entropy.

4. Key Generation Prototype

The key generation prototype mainly focuses on determining the proper challenges of the PUF as well as the parameters of fuzzy extractor.

4.1. Challenges

According the experimental results, the decay rate of Micron 1 DDR3 will be stable after 120 s. Thus the power-switch time is equal to 120 s. The offset of memory bits is equal to the target key length divided by the min-entropy per bit that is about 3878. We set the offset to 4080 with a margin.

By using this challenge, a response in the form of a 4080-bit binary string will be obtained in each measurement. The min-entropy of the 4080-bit response is about 134.64.

4.2. ECC Parameters

The length of ECC codeword is the same as the response length. The BCH forms a large class of powerful random error-correcting cyclic codes [20]. In order to simplify the process in encode and decode the 4080-bit response is divided into 16 groups with 255-bit per group. Some BCH code with block length 255 are listed in Table 2.

Table 2. Some BCH code with block length 255.

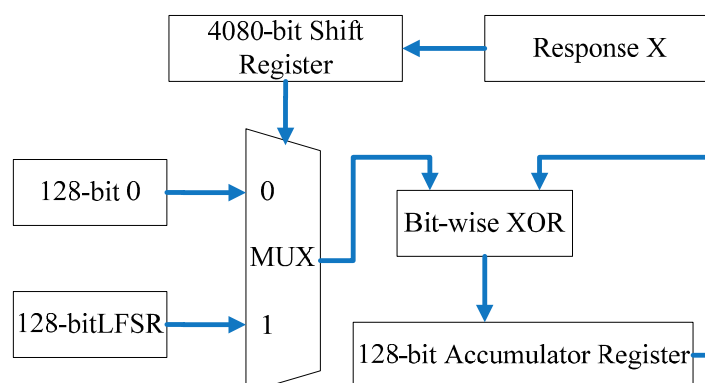
n	k	t	Error Correct Ability	Normalized Min-Entropy
255	63	30	11.76%	24.71%
255	55	31	12.16%	21.57%
255	47	42	16.47%	18.43%
255	45	43	16.86%	17.65%
255	37	45	17.65%	14.51%
255	29	47	18.43%	11.37%
255	27	55	21.57%	10.59%
255	13	59	23.14%	5.10%
255	9	63	24.71%	3.53%

We randomly choose 16 BCH (255, 37, 45) codewords. The error correct ability of each codeword is 17.64% which is between the normalized intra-Hamming distance and normalized inter-Hamming distance of Micron 1 DDR3. Moreover, the codewords can be deemed as a random variable with 14.5% normalized min-entropy which is larger than the value of response. This implies the bitwise XOR operation between the response and an ECC code will not leak the entropy of the response.

4.3. Universal Hash Parameters

The Toeplitz matrix is a universal hash instantiation [21]. It can be represented by a linear-feedback shift register (LFSR). The Toeplitz matrix transforms the 4080-bit response to a 128-bit almost uniform key as shown in Figure 10.

Figure 10. The structure of universal hash.



The primitive polynomial for the 128-bit LFSR is:

$$f(x) = 1 + x^{32} + x^{47} + x^{58} + x^{90} + x^{121} + x^{128} \quad (9)$$

The key can be calculated as:

$$key = m(1) \oplus m(2) \oplus \dots \oplus m(i) \oplus m(4080) \quad (10)$$

where $m(i)$ switches between a 128-bit 0 or the content of the 128-bit LFSR depending on the value at position i of the response binary string. The $m(1)$ is the seed of the LFSR.

According to [6] the output key length, the min-entropy of input response and the statistical distance between the key and universal distribution on $\{0,1\}^{128}$ should satisfy the equation below:

$$m = k - 2\log(1/\varepsilon) + 2 \quad (11)$$

The statistical distance of 128-bit key between uniformly distributed is about 0.05 with $m = 128$ and $k = 134.64$.

Several keys have been generated from the other responses of the Micron 1 DDR3. After the universal hash process the normalized min-entropy in the key is about 70.7%, which is much bigger than its value of the responses before process.

5. Conclusions

In this paper we use the decay feature of DDR3 to build a PUF optimized for WSN key generation. Several evaluation metrics such as decay rate, normalized inter- and intra-Hamming distance and Min-entropy per bit are used to evaluate DDR3 PUF. The decay rate will be stable after some power-switch time. The unique decay feature is demonstrated with the fact that the intra-Hamming distance rate is much smaller than the inter-Hamming distance rate. The Min-entropy per bit of decay feature is acceptable for PUF usage. The challenges are power-switch time as well as the address and offset of memory bits with logic 1 before power off while the responses are power-on value of those bits after specific decay interval. We construct a fuzzy extractor to eliminate the influence of the existing environmental noises and make key almost uniformly distributed. The parameters of fuzzy extractor are calculated according to the experimental result. A prototype of 128-bit key generation based on DDR3 PUF with integrated fuzzy extractor is presented. A 128-bit key is extracted from 4048-bit DRAM cells. Future works may involve environmental testing and large scale evaluation with more sample chips. With tamper-resistance at zero silicon cost and wide adoption in sensor nodes, DDR3 PUF is a promising key provisioning solution for various WSN applications.

Acknowledgments

This work is supported in part by National Natural Science Foundation of China (Grant No. 61176026, No. 61202469 and No. 61376026). We appreciate the denotation of STM32 development boards by ARM University Program. We also appreciate Heng Zhao and Gen Pei for their help on DDR3 memory testing experiments.

Author Contributions

The work presented here was carried out in collaboration between all authors. Wenchao Liu and Miaoxin Li co-prepared the manuscript, co-designed the post-processing process for reliable key generation. Zhenhua Zhang carried out all the experiments. Zhenglin Liu proposed the ideas and revised the manuscript. All of the authors read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Pathan, A.K.; Lee, H.W.; Hong, C.S. Security in wireless sensor networks: Issues and challenges. In Proceedings of the 8th International Conference on Advanced Communication Technology, Phoenix Park, Gangwon-Do, Korea, 20–22 February 2006; pp. 1043–1048.
2. Wang, Y.; Attebury, G.; Ramamurthy, B. A survey of security issues in wireless sensor networks. *IEEE Commun. Surv. Tutor.* **2006**, *8*, 2–23.
3. Semi-Invasive Attacks—A New Approach to Hardware Security Analysis. Available online: <http://core.kmi.open.ac.uk/download/pdf/72289.pdf> (accessed on 25 February 2014).
4. Halderman, J.A.; Schoen, S.D.; Heninger, N.; Clarkson, W.; Paul, W.; Calandrino, J.A.; Feldman, A.J.; Appelbaum, J.; Felten, E.W. Let us remember: Cold-boot attacks on encryption keys. In Proceedings of the 17th USENIX Security Symposium (Sec '08), San Jose, CA, USA, 28–29 July 2008.
5. Maes, R.; Verbauwhede, I. Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions. In *Towards Hardware-Intrinsic Security*; Springer Berlin Heidelberg: Berlin, Germany, 2010; pp. 3–37.
6. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97–139.
7. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical one-way functions. *Science* **2002**, *297*, 2026–2030.
8. Pappu, R. Physical One-Way Functions. Ph.D. Thesis, Massachusetts Institute of Technology, Boston, MA, USA, 2001.
9. 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.
10. Gassend, B. Physical Random Functions. Master's Thesis, Massachusetts Institute of Technology, Boston, MA, USA, 2003.
11. Maes, R.; Tuyls, P.; Verbauwhede, I. Intrinsic PUFs from flip-flops on reconfigurable devices. In Proceedings of the 3rd Benelux Workshop on Information and System Security, Eindhoven, The Netherlands, 13–14 November 2008.
12. Holcomb, D.E.; Burleson, W.P. Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *IEEE Trans. Comput.* **2009**, *58*, 1198–1210.

13. Guajardo, J.; Kumar, S.S.; Schrijen, G.-J.; Tuyls, P. FPGA Intrinsic PUFs and Their Use for IP Protection. In *Cryptographic Hardware and Embedded Systems—CHES 2007*; Springer Berlin Heidelberg: Berlin, Germany, 2007; pp. 63–80.
14. Hori, Y.; Kang, H.; Katashita, T.; Satoh, A.; Kawamura, S.; Kobara, K. Evaluation of Physical Unclonable Functions for 28-nm Process Field-Programmable Gate Arrays. *J. Inf. Process.* **2014**, *22*, 344–356.
15. Physically Unclonable Functions for Secure Hardware. Available online: <http://www.nano-tera.ch/pdf/posters2012/QCrypt53.pdf> (accessed on 1 March 2014).
16. Rosenblatt, S.; Cestero, A.; Robson, N.; Kirihata, T.; Iyer, S.S. Field Tolerant Dynamic Intrinsic Chip ID Using 32 nm High-K/Metal Gate SOI Embedded DRAM. *IEEE J. Solid-State Circuits* **2013**, *48*, 940–947.
17. Cliff, Y.; Boyd, C.; Nieto, J.G. How to Extract and Expand Randomness: A Summary and Explanation of Existing Results. In *Applied Cryptography and Network Security*; Springer Berlin Heidelberg: Berlin, Germany, 2009; pp. 53–70.
18. Carter, J.L.; Wegman, M.N. Universal classes of hash functions. *J. Comput. Syst. Sci.* **1979**, *18*, 143–154.
19. Virtex-7 XT VC709 Connectivity Kit Getting Started Guide. Available online: http://www.xilinx.com/support/documentation/boards_and_kits/vc709/2013_2/ug966-v7-xt-connectivity-getting-started.pdf (accessed on 1 November 2013).
20. Lin, S.; Costello, D.J. *Error Control Coding*, 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2004; pp. 130–155.
21. Hugo, K. LFSR-based hashing and authentication. In Proceedings of the 14th Annual International Cryptology Conference, Santa Barbara, CA, USA, 21–25 August 1994; pp. 129–139.

© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).