p-ISSN : 2685-9106
e-ISSN : 2686-0384
ADI International Conference Series

Puf-trng for a Secure Industry 4.0

Eka Harsono[1], Chong Ming Tang[2]
harsono.ekajaya@gmail.com[1], cmtang.cm@gmail.com[2]
Universiti Tunku Abdul Rahman, Malaysia[1,2]

To cite this document :
Harsono, E., & Tang, C. M. . (2021). Puf-trng for a Secure Industry 4.0. Conference Series, 3(1), 584 - 592. https://doi.org/10.34306/conferenceseries.v3i1.394

Abstract

Modern technology such as IoT has made a significant change in the industry. By enabling communication between smart machines, the automation process would be more dynamic and adaptive. It will enable the machines to perform necessary tasks based on real data received from the other hardware. Additionally, the data recorded in the machines will further help the engineer to easily check the problem and the information needed for future development. However, unsecured data exchanges will compromise the data and the security of the devices. Popular security measures such as encryption and decryption are commonly used to solve the information exchange. However it is possible to further improve the performance for both encryption and decryption process by implementing a TRNG (True Random Number Generator). TRNG requires a strong entropy such as the proposed Physical Unclonable Function (PUF) which produces a response based on the physical variations of device parameters during the fabrication process.

Keywords: IoT, Hardware Security, TRNG, Physical Unclonable Function, PUF.

I. INTRODUCTION

Industry 4.0 relies greatly on communication between intelligent devices to further improve the automation process. Therefore, data security and integrity are essential. IoT is one of the innovations where automated machines in a smart factory can communicate and work with other hardwares in a cooperative network.

Any network needs to be protected from unauthorized access, data theft and system manipulation. Extensive information exchange causes unsecure information to be exposed. It requires extensive security measures to ensure information integrity. Information encryption and decryption techniques are commonly deployed to secure these data exchanges. Both encryption and decryption require a key during the information exchange. A Physical Unclonable Function Truly Random Number generator (PUF-TRNG) as a key generator, will ensure that the key is more secure.

Since most modern day machines are controlled by microcomputer systems usually in a single chip (SoC) it will not have a lot of resources to run conventional software-based security measures. Hardware-based security modules such as PUF (Physical Unclonable Function) are more desirable. The PUF used produces a response based on the physical variations of process parameters that happen during the device fabrication process.

This research will propose a novel TRNG module which utilizes PUF response as a seed to produce a randomized sequence. PUF response was extracted from the power-up state of memory elements from the smart nodes in the IoT.

This paper is structured as follows, Section 2 will explain the concept of Random Number Generator (RNG) and PUF which will be a source for the TRNG seed. The design of the proposed PUF-TRNG will be discussed in Section 3, with its result and analysis in Section 4. The conclusion will be laid out in Section 5.

II. LITERATURE REVIEW

a. Physical Unclonable Function (PUF)

Physical Unclonable Function (PUF) is a suitable and novel approach to create this unique key as an identity for the devices (Schaumont, P. 2009). PUF utilizes natural physical characteristics (for example, the skewed structure, thickness variation and dimension variation of a fabricated semiconductor structure) to produce the unique key. PUF is unclonable

because hardware variation is a natural occurrence with no human intervention. (Maes, R. 2013). Fig. 1 illustrates the concept of PUF in general.

A PUF response must be unique and of a truly random nature but for each device it must be reproducible. In earlier works, the true random nature of the power-up state of a flip-flop is used as a PUF. PUF works by CRP (Challenge Response Pair) mechanism. The number of CRP determines whether PUF are weak or strong. Weak PUF are suitable for microcontroller implementation since they only have a single CRP. Strong PUF are more suitable for complex hardware such as cashier systems or ATMs. Additional storage is required to store multiple CRP combinations. However, Strong PUF are more vulnerable since it will be easier to guess the PUF by comparing the CRP response. Power-up ramp is the challenge while the undefined value of flip-flop outcome is the responses (Ayoub, M. 2017).

PUF is implemented into various security measures such as cryptography, hash function, authentication process, true random number generator and hardware fingerprint. PUF is a suitable candidate for various IoT security measures. (Bayoumi, M. 2017).

b. Random Number Generator (RNG)

Random number generator (RNG) is a set of instructions to produce a set of random sequences so as not to be confused with a randomness discussed in the PUF subsection. RNG was classified into two different groups as True Random Number Generator (TRNG) and Pseudo Random Number Generator (PRNG).

RNG was important for various security measures Such as a key for encryption and decryption (Guo. Y). A truly random sequence would significantly improve the security performance since it would not be easy to guess. On the other hand, a pseudo RNG was more predictable since it has an ending cycle for each seed (Bhattacharjee K). Therefore, PRNG was not secure to be used to generate keys for encryption and decryption.

RNG Require entropy to generate the output. The entropy for both TRNG and PRNG were different in nature. TRNG exploits natural entropy, which came from natural phenomena which were truly random and unpredictable. PRNG utilizes a fixed seed, therefore it's possible to guess the sequence produced by PRNG. However, PRNG may be used as TRNG as long as the entropy given as a seed came from a natural event.

c. RUMPS401

RUMPS401 is an MPSoC produced by the UTAR VLSI research center (Hartono, D. (2014)). RUMPS401 is a versatile MPSoC powered by four ARM Cortex-M0 cores. The ARM Cortex-M0 has low power consumption yet powerful enough for most of the requirements of a smart IoT node. Fig. 2 shows RUMPS401 layout and physical form.

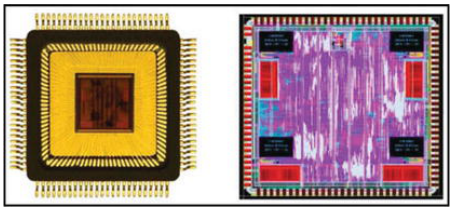


Fig. 2. RUMPS401 layout and physical form

The RUMPS401 is equipped with an adaptive Network on Chip (NOC) which allows the processor cores to communicate with one another (Lokananta. F. (2015)). It supports both hardware and software bootloader mechanisms to import the software easily. It implements Power Management to reduce the power consumption of the chip while the processor goes into sleep mode.

For security modules, the RUMPS401 is equipped with a hardware AES accelerator which enables RUMPS401 to perform 128-bit AES encryption. More importantly, the RUMPS401 design has a total of 26,926 flip-flops. Therefore, the RUMPS401 is an ideal candidate for building an IoT system with PUF-TRNG as the security implementation.

III. PUF-TRNG

a. Random Balanced Flip-Flop (RBFF)

The behaviour of the flip-flop might differ based on their physical variation. Random Balanced Flip-Flop is a term used for flip-flops which have a finely balanced structure due to their unique manufacturing variation. Due to the finely balanced nature, the flip-flop can power-up in the “0” or “1” state with a slightly different environmental perturbation in the power-up process.

This research will use RBFF as seed for a True Random Number Generator (TRNG). A TRNG has been implemented using the RUMPS401 and the result showed that it passed all

NIST Randomness test parameters. The illustration on the RBFF characteristics is shown on fig. 3. Elements 2 and 3 are RBFF since both flip-flops are able to produce both 0 and 1 during multiple power-up iterations. The X axis represents the number of iterations and the Y axis represent the state of the element at each available iterations.

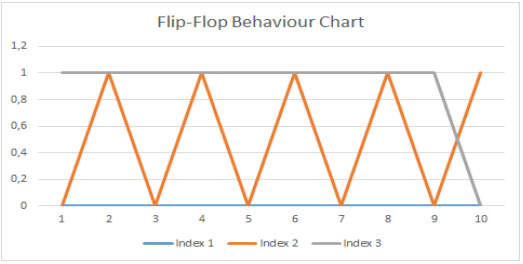


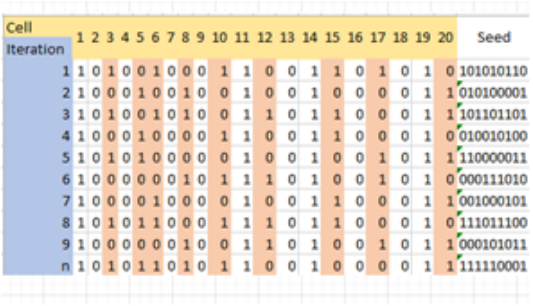
Fig. 3. FF Behaviour Chart.

b. TRNG Seed Extraction Implementation

In this research a TRNG was created by exclusively selecting RBFF as an element for the seed since the RBFF power-up states would be random for every power cycle iteration.

The element used as a TRNG seed was selected by choosing an element which satisfies the RBFF criteria. The RBFF does not always produce the same state across multiple power-up iterations. However, not every RBFF was usable as TRNG elements. Certain RBFF elements may have a biased distribution which will weaken the entropy strength. Additionally, a low count of selected RBFF elements will also affect the entropy strength. Unfortunately, if the RBFF which were closer to the 100% biased were de-selected, it would reduce the number of usable RBFF elements. A trade off was therefore necessary.

The remainder of unselected RBFF elements would not be used in the TRNG seed. Fig. 4 was used to illustrate the extraction of the TRNG seed. The red column defines a finely balanced flip-flops element which may produce random power-up states.



Cell Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Seed
1	1	0	1	0	0	1	0	0	0	1	1	0	0	1	1	0	1	0	1	0	101010110
2	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	1	010100001
3	1	0	1	0	0	1	0	1	0	0	1	1	0	1	1	0	0	0	1	1	101101101
4	1	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	010010100
5	1	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1	1	110000011
6	1	0	0	0	0	0	0	1	0	1	1	1	0	1	0	0	1	0	1	0	000111010
7	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	001000101
8	1	0	1	0	1	1	0	0	0	1	1	1	0	1	1	0	0	0	1	0	111011100
9	1	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1	1	000101011
n	1	0	1	0	1	1	0	1	0	1	1	0	0	1	0	0	0	0	1	1	111110001

Fig. 4. Example of TRNG Seed Extraction.

IV. RESULTS & ANALYSIS

a. NIST Test

To analyze the entropy strength, the binary bitstream from the TRNG would be evaluated by the NIST Test suite. The strength of the entropy was gauged by the NIST test results. The strength of the TRNG seed may vary across different devices due to uncontrollable physical variations. The NIST randomness tests consist of 15 different tests. The list of the tests were shown in table fig. 5.

No	Name of Test	Purpose
1	monobit	count 0 and 1 frequency ratio
2	monobit within a block	count 0 and 1 frequency ratio
3	Runs Test	count how often is 0 or 1 appear continously
4	Runs Test within a block	count how often is 0 or 1 appear continously
5	Binary Matrix Rank Test	Detecting linear dependent pattern through matrix
6	Spectral Test	Count the frequency of 0 and 1 by inspecting the peak of DFT
7	Non-Overlapping template test	Check whether a sequence contain certain string
8	Non-Overlapping template test	Check whether a sequence contain certain string considering overlapping case
9	Maurers Universal Test	Compression test to check whether there is a compressable pattern
10	Linear complexity	Measuring a complexity of the sequence
11	Serial Test	Measuring overlapping patterns occurrence
12	Approximate Entropy Test	Calculate a frequency of overlapping blocks between adjacent blocks.
13	Cusum Test	Measuring the occurrence of consecutive 1 and 0
14	Random Excursion Test	Measuring the occurrence of consecutive 1 and 0
15	Random Excursion Variant Test	Measuring the occurrence of consecutive 1 and 0

Fig. 5. NIST Tests

The input data for the NIST test was a binary sequence extracted from the power-up state of the RBFF elements in the RUMPS401 flip-flops. This constitutes a seed for the TRNG. The data from 50 power-up iterations of the RUMPS401 were collected. The minimum size required in order to perform all tests in NIST was 1 MB. Therefore, a 20 KB binary bitstream had to be generated by using the C++ PRNG algorithm from each seed produced by the 50 power-up iterations. The visualization for binary construction was shown in Fig. 6.

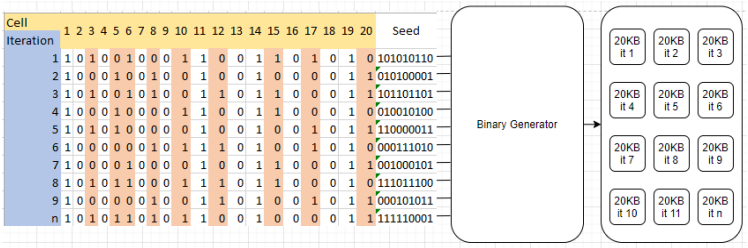


Fig. 6. Binary file construction visualization

b. Results Analysis

To analyze the entropy strength, the binary bitstream from the TRNG would be evaluated by the NIST Test suite. The strength of the entropy was gauged by the NIST test results. The strength of the TRNG seed may vary across different devices due to uncontrollable physical variations.

Fig. 7, fig. 8 show the NIST tests results of 33 RUMPS401 chips with <90%, <80% bias rates respectively. Based on these NIST tests results, the <90% rate was shown to be the most suitable selection criteria as it was able to produce a random seed to pass a good number of the NIST tests. On the other hand, <80% rates increased the number of unusable chips and chips which fail most of the NIST tests.

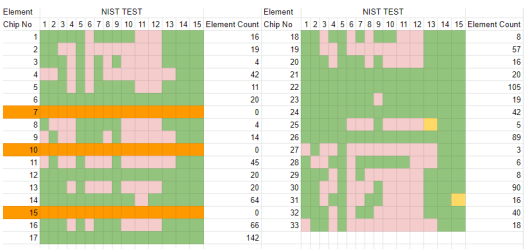


Fig. 7. 90% Distribution Rate 33 RUMPS NIST Tests Result

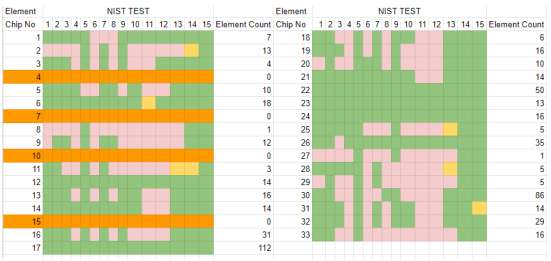


Fig. 8. 80% Distribution Rate 33 RUMPS NIST Tests Result

V. CONCLUSIONS

The power-up states of the RBFF (Random Balanced Flip-Flop) are sensitive to manufacturing process variations so that their responses are different over different locations on the same chips as well as across different chips.The balanced structures are finely balanced enough to produce different responses over repeated power-up to be suitable as a TRNG seed.

Based on the analysis, the result indicates that the PUF response was proven to be usable as a seed for the TRNG. However, not all PUF responses were able to produce a good randomized sequence.

The PUF response is heavily influenced by process parameter variations during wafer fabrication. Physical variation of the chip affects the strength of the TRNG. The NIST Test Suite was able to measure the strength of the TRNG entropy by evaluating the binary output stream that was produced by the TRNG module.

Not all RBFF elements might be used as TRNG seed on every device. By not selecting an RBFF which is biased too close to 100% ‘1’ or ‘0’ output state, it is possible to produce a TRNG seed. The less than '90% bias rate selection was able to produce 28 usable chips with 7 of them being able to pass all 15 NIST tests. This indicated chips which were able to produce seeds with strong entropy.

REFERENCES

[1] A. Maiti and P. Schaumont, “Physical Unclonable Function and True Random Number Generator : a Compact and Scalable Implementation,” Glsvlsi2009, pp. 425–428, 2009.

[2] R. Maes, Physically unclonable functions: Constructions, properties and applications, vol. 9783642413. 2013.

[3] A. Wael and M. Ayoub, “Physical and Mechatronic Security, Technologies and Future Trends for Vehicular Environment,” VDI-Fachtagung Automot. Secur. VDI Berichte, vol. 2310, pp. 73–95, 2017.

[4] T. Idriss, H. Idriss, and M. Bayoumi, “A PUF-based paradigm for IoT security,” 2016 IEEE 3rd World Forum Internet Things, no. February 2019, 2016.

[5] K. Bhattacharjee, K. Maity, and S. Das, “A Search for Good Pseudo-random Number Generators : Survey and Empirical A Search for Good Pseudo-random Number Generators :,” no. November, 2018.

[6] D. Hartono, “THE DESIGN AND IMPLEMENTATION OF A SCALABLE MULTI-PROCESSOR SYSTEM-ON-CHIP USING NETWORK COMMUNICATION FOR PARALLEL COARSE-GRAIN DATA PROCESSINGS,” no. August, 2014.