A Morphable Physically Unclonable Function and True Random Number Generator using a Commercial Magnetic Memory

Mohammad Nasim Imtiaz Khan, Chak Yuen Cheng, Sung Hao Lin, Abdullah Ash- Saki, and *Swaroop Ghosh
*School of EECS, Pennsylvania State University, University Park, PA, USA
Email: *szg212@psu.edu

***Abstract*— In this work, a morphable security primitive using commercial magnetic memory is proposed which can be utilized as a Physically Unclonable Function (PUF) and as a True Random Number Generator (TRNG) by manipulating the write time and the number of write pulses respectively. The intra-HD, inter-HD, energy, bandwidth and area of the proposed PUF is found to be 0, 46.25%, 0.14pJ/bit, 0.34Gbit/s and 0.385μm²/bit (including peripherals) respectively. The proposed TRNG provides all possible outcomes with a standard deviation of 0.0062, correlation coefficient of 0.05 and an entropy of 0.95. The energy, bandwidth and area of the proposed TRNG is found to be 0.41pJ/bit, 0.12Gbit/s and 0.769μm²/bit (including peripherals). The proposed TRNG has also been tested with NIST test suite.**

I. Introduction

The proposed morphable security primitive can be used both as TRNG and PUF by manipulating the write time and the number of write pulses to a commercial toggle MRAM chip (Table I). This is based on the observation that intrinsic/extrinsic Process Variation (PV) in the MTJ changes its write latency. Therefore, if the write latency is chosen carefully, it will statistically flip/not flip the same bit in two different chips (useful for PUF). Furthermore, the same bit will be stochastically flipped if written multiple times with the same data (useful for TRNG) which we call intrinsic variation. Fig. 1(a) presents the basic concept. A 128KB commercial MRAM chip can be converted to 64KB TRNG and 64KB PUF, or to 128KB PUF/TRNG. First, we flush the cells of both TRNG and PUF. Then we write all 1's by manipulating the write time to flip 50% of the bits (i.e., 50% switching probability). Due to stochastic nature of the bitcell and PV, different chips will be written with different data. However, note that the MRAM will be re-written every time a random number needs to be generated to realize TRNG whereas the MRAM will be written only one time (during enrollment phase) to realize the PUF. The written data to PUF addresses is a function of PV and cannot be cloned.

Prior works [1-4] consider bitwise normal distribution of switching probability. However, we observe that real memory implementations offer narrow distribution and, some columns/rows can be stuck to 0/1 (Fig 1(b)). Furthermore, bitwise biasing is not possible and different biasing for each row is also impractical. *To the best of our knowledge, this paper makes the first attempt to observe and address these practical challenges. To the best of our knowledge, this is the first experimental demonstration of PUF/TRNG using commercial MRAM.*

II. PUF

The proposed PUF has two phases:

(a) Enrollment phase: The entire address space is flushed (write all 0's) and then, write all 1's with write time ($T_{50\%}$) to switch 50% of the bits. Next, the entire address space will be read to extract the random responses. This step will be repeated for few chips to gather correct statistics and identify the unresponsive columns/rows (no variation) for all chips. This gives highest uniqueness for different chips.

(b) Authentication phase (Fig. 8(d)): Challenge (address) will be sent to the chip which will send the response by reading the data stored in that address. If a set of challenge-response pairs matches (which is enough to distinguish a chip uniquely), the chip will be authenticated.

Enrollment phase is just a one-time operation performed by manufacturer. Authentication phase (involves read operation only) needs to be performed in run-time.

Table I: Characteristic of the MRAM chip used in this work

Parameter	Value
Capacity	16Mbit
Read/Write Cycle	35ns
Address/Data Bus Length	21/8
Retention Time	>20years
AC stand by Current	9-14mA
AC Active Current (Read/Write)	60-68mA/152-180mA

III. TRNG

First, the memory address is flushed by writing all 0's at rated write time. Then, 1's will be written with write time for 75% switching probability (i.e., the ratio of number of 1's and 0's will be 75%). The written data will be considered as the first random number. Then, the same address will be written with all 1's with the same write time (without flushing). From our findings, the ratio of number of 1's and 0's reduces to ~50% (observation (b), Section II). This written data will be the second random number. If we write again, ratio of number of 1's and 0's reduces to ~35%. We read and get the third random number. This process can be repeated for other write times and for other addresses to generate more random numbers. Fig. 12(b) shows the outcomes of 10,000 cases for the proposed TRNG.

IV. Conclusions

We proposed TRNG and PUF using commercial MRAM and manipulation of write time and number of writes. We have analyzed the practical implications and addressed them to achieve high quality.

References

[1] Iyenger, A. et al, "Spintronic PUFs for Security, Trust, and Authentication." JETC, 2016.
[2] Ghosh, S., et al. "Spintronics for associative computation and hardware security." MWSCAS, 2015.
[3] Fukushima, A., et al., "Spin dice: A scalable truly random number generator based on spintronics," JAP Express, 2014.
[4] Vatajelu, E. I., et. al, "Security primitives (PUF and TRNG) *with STT-MRAM," VTS, 2016.*

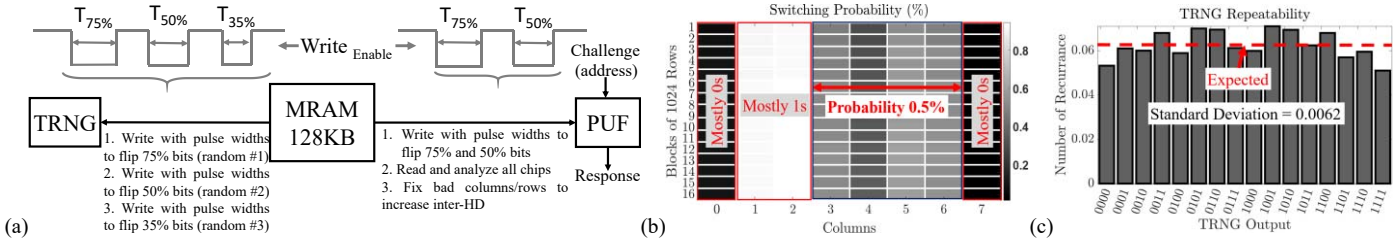

Fig. 1 (a) Morphable security primitive using MRAM; (b) switching probability distribution of 16KB MRAM. First 16384 rows are grouped to 16 blocks (1024 rows/block); (c) repeatability of TRNG outcomes out of 10,000 responses.

NSF CNS – 1722557, CCF – 1718474, DGE – 1723687 and DGE – 1821766, and SRC 2847.001.