Memristor Crossbar PUF based Lightweight Hardware Security for IoT

Mesbah Uddin, Aysha S. Shanta, Md Badruddoja Majumder, Md Sakib Hasan, and Garrett S. Rose

Proceedings of the IEEE Internation Conference on Consumer Electronics (ICCE), Las Vegas, NV, Jan 2019.

Citation Information (BibTex):

```
@INPROCEEDINGS{ICCE:Uddin2019,
  author="M. Uddin and Aysha S. Shanta and M. B. Majumder
  and Md Sakib Hasan and Garrett S. Rose
  title="Memristor Crossbar PUF based Lightweight Hardware
  Security for IoT",
  booktitle="IEEE Internation Conference on Consumer
  Electronics (ICCE)",
  month="Jan",
  year="2019",
  address="Las Vegas, NV"
}
```

Memristor Crossbar PUF based Lightweight Hardware Security for IoT

Mesbah Uddin, Aysha S. Shanta, Md. Badruddoja Majumder, Md. Sakib Hasan, and Garrett S. Rose
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville, Tennessee
Email: {muddin6, mmajumde, ashanta1, mhasan4, garose}@utk.edu

***Abstract*—Billions of IoT (Internet of Things) devices are being introduced to our everyday life each year. Most of them have little to no security features embedded in them. But these devices connect to internet and communicate with other devices all the time which pose a great security and privacy risk. Due to the resource constraint nature of these devices, initially security was not being considered as an essential feature. Therefore, any security features to be included in these devices should be minimal in area, energy consumption and added delay. Physical unclonable function (PUF) has emerged as a low-overhead solution for a variety of security concerns in recent years. In this paper, we have presented a simple security method for small IoT devices based on PUFs. The idea is to secure the backup data during sleep mode of an embedded processor or when power is unavailable for a batteryless system. With the emergence of nano-technology, memristors are being widely explored due to their non-volatility, low footprint among other advantages. Our proposed security system utilizes a memristor based PUF along with memristors as a non-volatile backup memory. The proposed system is very lightweight as this domain demands and provide reasonable security.**

I. Introduction

In the era of connectivity and smart technology, there is a plethora of small devices that connect to the Internet and other networking sources with very limited security features. Many of these devices may contain time-sensitive data. Most of these devices run on battery with very limited power available to them and some are even batteryless, dependent on harvested energy alone. Therefore, to save as much power as possible and also due to their intermittent nature of data sensing operation, they often go to sleep or ultra low power mode before backing up their sensitive states and other information. Now, these backup data are usually saved in non-volatile memory and are vulnerable to probing and malicious read. In this work, we are mainly focused on providing a security for the backup data in these resource constraint devices. Now, since the data in an IoT device are mostly temporary, the information is no longer useful after a certain period of time. Therefore, their security requirements are relaxed in terms of complexity of code-breaking algorithms but has to be very lightweight in terms of area occupied, power consumed and delay introduced. In this work, we have focused our attention on developing an ultra lightweight security for the backup data of these devices.

We have considered memristors as backup memory for this work. Memristors are one of the most promising candidates for next-generation post-CMOS devices. The term 'memristor' was first coined by Leon Chua in [1] and later experimentally demonstrated by HP labs in [2]. Memristors have multiple stable resistive states and can retain their states without any power for a long time. Thus a memristor essentially has a zero-standby power and is useful for IoT devices where power failure can occur for a long period of time.

Physically unclonable functions or PUFs have been used as a solution to a variety of security concerns including cloning, IC counterfeit, piracy etc [3]. PUFs have to be lightweight and thus emerging nano-devices like memristors are being explored in a wide range of hardware security domain including PUFs. In this work, we have used a slightly modified version of a memristor crossbar PUF, called the XbarPUF [4] which was later expanded and improved in [5]–[7]. This PUF is based on the variations in stochastic switching properties of memristors arising from large manufacturing process variations. In an XbarPUF, two sets of crossbar columns of memristors are compared to generate a single bit response. Since these memristors are spatially very close in a crossbar architecture, any environmental change should affect them both equally and thus relative characteristics are retained against varying environmental condition.

The contributions of this paper follows: (1) identifying the security needs and requirements of resource limited IoT devices, (2) proposing a PUF-based lightweight security system, and (3) circuit-level demonstration of the proposed security system and overhead analysis.

The paper is organized as follows: section II first provides a short introduction to memristor, memristor PUF and the idea of a one time pad for security. Section III provides the limitations and requirements of security measures in this particular IoT domain and presents the concept of our proposed system. Section IV explains our proposed system in details and describes different components of it. Section V presents the experimental results of the PUF and provides the overhead of our security design. Finally, future prospects and concluding remarks are provided in section VI and VII, respectively.

II. Background

A. Memristor

The memristor or also known as RRAM, ReRAM (resistive random access memory) or more specifically transition metal-oxide may be constructed from a wide variety of materials.

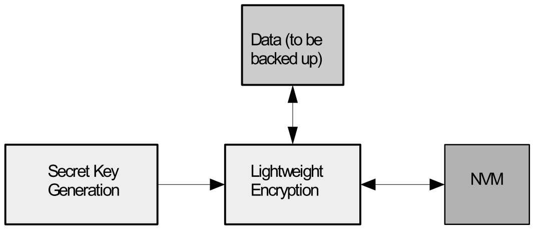

Fig. 1. High-level conceptual block diagram of the proposed system

Memristors are non-volatile, have relatively high off-on ratio, lower operating voltage, very high integration density, CMOS-compatible and analog in nature. Most metal-oxide memristors are usually bi-polar and thus current flows through them in both directions. Most of them are also binary i.e. they usually have two stable memristive states: a high resistive state (HRS) and a low resistive state (LRS). For memory purposes, HRS can be considered a '0' (little current) and LRS can be considered a '1' (high current). By applying an appropriate voltage of correct magnitude and direction for a finite duration of time, these two states can be reached.

B. Memristor PUF: the XbarPUF

In an XbarPUF [6], the memristors are arranged in a dense-crossbar architecture. The challenges are applied along the rows of the crossbar and responses are taken from each two adjacent crossbar columns. Each challenge bit can affect two adjacent rows. For a challenge bit of '1', a positive set voltage pulse is applied to switch the memristors in that row and no voltage is applied to the adjacent row. For a challenge bit of '0', opposite scenario happens i.e. the first row remains unaffected but a positive set pulse is applied to the second row. All the challenges of a PUF are applied simultaneously. A 1-bit response is generated after comparing the changes in memristance between adjacent columns by applying a small read voltage pulse along the rows. To apply a new challenge, all memristors should be returned to their original states. Therefore, a negative voltage pulse is applied at the beginning of each response-generation period to reset all the memristors to HRS. Thus the XbarPUF has three phases of operation: reset-all, challenge and read.

C. One Time Pad

The one time pad or also often called the Vernam cipher is the perfect cryptographic algorithm providing maximum security [8]. In this cryptosystem, the plaintext is paired with a random secret key and that key is changed every time a new encryption is needed, thus theoretically ensuring the best security. However, this one time pad or OTP is not used much in practice due to the difficulties in its requirements. The key must be (1) truly random, (2) as long as the plaintext, (3) cannot be used more than once. The distribution and secure storage of the key are also big concerns. However, the requirements are mostly fulfilled or relaxed in our proposed system. First, in embedded or IoT domain, the plaintext or data to be backed up is small and, therefore, the key can actually be made as large as the data. Second, in theory, PUFs can produce true random number as responses from non-repetitive challenges. Third, the key need not to be stored directly since PUF can generate the key anytime if the same challenge is applied. By ensuring a different challenge input each time, the requirement of producing different key can be fulfilled.

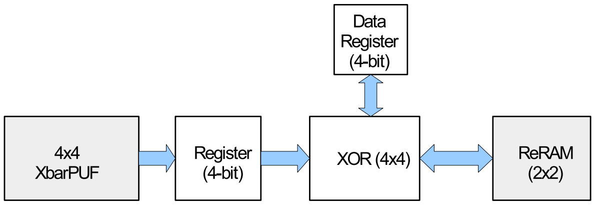

Fig. 2. Block diagram of the designed prototype system

III. Design requirements and the proposed system

Any IoT device or embedded system running on an external environment can be subject to data theft or malicious data manipulation or alteration. Specifically, during power saving or sleep mode, a processor saves its program states and other necessary registers or memory data in a non-volatile memory for successful and quicker wake-up operation and resume from last state before power-down to increase throughput. This memory can contain important temporary information that an adversary can collect easily by different invasive and non-invasive methods. To protect this memory, some form of encryption and/or data integrity checking method must be used. However, these security algorithms must be lightweight i.e. they should have minimal power and area consumption with low delay. Fig. 1 shows the conceptual block diagram of the proposed security method. The secret key generation block of this figure is essentially a PUF. Before each power down or sleep phase, a random challenge is applied to the PUF to generate a unique random response. Simple techniques like majority voting can be used to produce reliable response if needed. Then the data to be backed up can be encrypted with a ultra-lightweight cryptographic technique and then saved in a non-volatile memory (NVM). This way the data would be safe against malicious read. For this demonstration, we have used a simple bitwise XOR to encrypt the data. To restore the data, the same challenge is applied to the PUF again to generate the key. In this case, simply XORing the encrypted data with the PUF key would give back the original data. The idea is to provide robust security like an one time pad where they key is random and changed on each encryption-decryption operation. The added benefit of using a PUF is that the key is inherently random with pseudo-random challenge input, need not to be stored physically and can be generated on demand.

IV. System Description

Fig. 2 shows our implemented security system for an IoT. For this demonstration, we have implemented a small 4×4 modified XbarPUF which takes 4 challenge bits as input and produces a 4-bit response as the cryptographic key. In the first phase of operation, this PUF takes one cycle to generate the 4-bit key and that key is saved in a 4-bit register. In second phase, key is XORed with the data register and this encrypted data is written in an RRAM one bit at a time. Thus this phase takes 4 clock cycles for a 4-bit key. During decryption, the PUF is provided with the same challenge to produce the same response key and encrypted data is read from the RRAM bit by bit. Decryption is done by XORing the encrypted data with this key again and the decrypted data is saved back in the original data register. Here, the encryption and decryption operation are symmetric and share the same XOR block to reduce area. The idea of this security design is to try to implement as close as possible to a hardware one time pad [8]. If the PUF is provided with pseudo-random challenges as input, the responses would be random. Each time a power-down occurs and backup operation is needed, the PUF is provided with a new challenge and thus the resulted response or the key would be new too. Therefore, just like an one time pad, the cryptographic key would be changed each time it is used. Moreover, since the length of data to be backed up is not very long in these IoT devices, the key can actually be made as large as the data and thus any repetition based attack should not reveal the key.

A. *Redesigned XbarPUF for IoT*

The memristor crossbar PUF or the XbarPUF is the heart of operation of this system. In a regular XbarPUF [6], challenges are applied for a very short period of time, much smaller than the actual switching time of a memristor, to nudge the memristance towards HRS or LRS depending on the magnitude and direction of applied challenge voltage. Then a small read voltage pulse is applied to read the memristance of both crossbar column memristors. This whole process is repeated until one of the columns reach to either HRS or LRS first and thus declared the winner. This is the so-called read-monitored-write approach [9] where knowledge of precise switching time is not important. The problem with this approach is that the time to get a valid response is not predictable and thus not suitable to use within a processor based system.

To overcome these issues, we do not use the gradual read-monitored-write approach and use a single fixed pulse to nudge the memristors midway between HRS and LRS. Although, it requires prior knowledge of exact switching time of memristors, it makes the input-to-output sample time fixed and predictable and thus useful to incorporate in a processor based system. The memristor switching time can be determined beforehand from foundry or we can change the clock period to try with different frequencies and select whichever gives the most stable output. We choose the clock period to be around half the switching time of the memristor. Choosing a fixed low frequency could make both the column memristors reach to either LRS or HRS and thus eventually having little difference between them. Choosing very high frequency could make the change in memristance very small and thus could be impractical to differentiate in a circuit. For the switching time on the range of 50-100ns, we have chosen the pulse width of the clock to be 25ns. This technique doesn't require the use of arbiters or flip-flops and sense amplifiers are used instead to generate responses.

TABLE I
PERFORMANCE OF MODIFIED XBARPUF USED IN THIS DESIGN

Security metrics	Uniqueness(%)	Bit-aliasing(%)	Reliability(%)
Average	50.090	50.833	99.904

B. *RRAM for non-volatile storage*

For this demonstration, we are encrypting a 4-bit data with 4-bit PUF key. Therefore, we need a 4-bit non-volatile memory. We have designed a simple 2×2 RRAM for that purpose. It can be accessed one bit at a time and thus needs a total of 4 clock cycles to either write or read all 4 bits. Only 4 memristors along with the address decoder/counter are needed to implement this simple memory element.

C. *Other blocks and control circuits*

Four XOR gates are used to implement the shared encryption-decryption block. Three sets of 4-bit registers are used for holding the original data, PUF key and encrypted/decrypted data. One two-bit counter is needed as addresses for the RRAM and the data register. Other smaller logic blocks like multiplexers, decoders and primary logic gates are used to control different parts of the design.

V. RESULTS, ANALYSES AND DISCUSSION

First, we have evaluated our PUF by running Monte Carlo statistical analysis. Although, it is only a demonstration, using Monte Carlo, we have analyzed 500 chip instances, and for 500 cycles each. The results for important PUF metrics are listed in Table I. As we can see, our proposed PUF shows excellent uniqueness and bit-aliasing which indicates the random nature of the response bits. Reliability is also very good, although not 100% as a typical cryptographic algorithm demands. However, in our proposed system, the PUF only needs to be reliable in a single set of encryption and decryption cycles and causes no problem if the PUF produces different response in a completely different time and can actually increase the entropy of the PUF this way.

Fig. 3 shows the waveforms to demonstrate how this system works. In the first cycle, a 4-bit PUF response is generated and saved in a 4-bit register. In next four cycles, each key bit is XORed with one bit from data register and the RRAM memory bit is updated according to the XORed value. The signals 'SET_on_one' and 'RESET_on_zero' in Fig. 3 show that the first two encrypted bits are '0' (being RESET) and the next two bits are '1' (being SET) in this case. Decryption starts with first reading the RRAM bit-by-bit. Like a memory write, memory read also takes 4 clock cycles. In last stage of decryption, the PUF is enabled again to produce the key and encrypted data read from the RRAM is XORed with the key

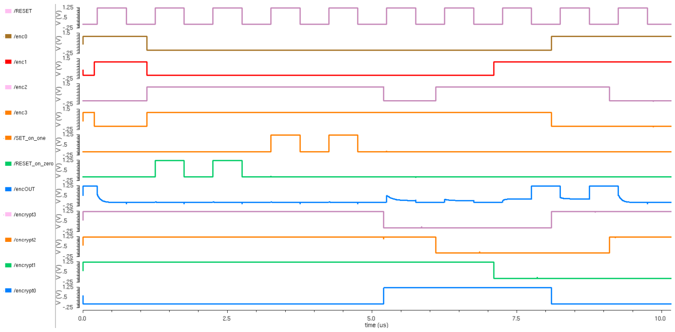

Fig. 3. Waveform showing the encryption-decryption operation of the designed prototype

to produce the original data again. The decrypted data is saved back to the data register in this cycle also.

Table II shows the overhead of our proposed system. The encryption and decryption phases involve a lot of memristor read-write operations and consume larger amount of power compared to write-only and read-only stages for the RRAM. Both the key generation phases again have just a single cycle delay. Where for an N-bit key and thus N-bit memory, it takes N cycles each to write and read all the bits.

The total GE (gate equivalent) count compared to a unit-sized NAND of our whole system is approximately 547. Majority of this number comes from the large pass-gates used for read-write control circuits of RRAM. Fortunately, these control blocks can be shared among many row-columns and thus GE count won't increase much with increasing RRAM sizes. There are also a total of 64 memristors in the XbarPUF (4×4) and 4 memristors in the RRAM(2×2). Memristors in a crossbar architecture takes very little area compared to even a single transistor and thus they contribute to a large reduction in overall area.

VI. Future Work

The randomness of the response bits generated by the PUF is crucial to the security of this system. To generate unique and random responses, challenges to the PUF should be non-repetitive. To ensure that, on-chip pseudo-random number generator (RNG) or a separate source of randomness might be needed. To prevent same challenges being applied more than once, the challenge can be made dependent on other factors like current program states, previous states, noise etc. This is something we are working on right now. To detect malicious write on the RRAM, a simple and very lightweight data integrity checking might be used. One such example for RRAM is in [10]. PUF-based encryption mitigates malicious read while the data integrity checking can detect any malicious write operation. If after decryption, any unintended data write is found, program states can simply be flushed out and the system can be restarted to prevent any erroneous calculation.

VII. Conclusion

We have built a small prototype of our proposed system in transistor level. We can estimate the performance for a larger system by scaling up the results from different components. Any rigorous encryption-decryption based security with large power-delay-area overhead is not practical for these small IoT devices and, therefore, are not considered. The overhead of the implemented system has to be as low as possible and our aim was to achieve towards that. Any backed up temporary data of an IoT device are argued to be valid and important only for a small period of time and thus a reasonably secure system can achieve that purpose. Since our implemented system would change the key each time a backup operation is needed, it can thus ideally can behave like a one time pad. PUF adds the benefit of not requiring the key to be saved separately and can be generated whenever needed. Memristor based circuits help to reduce the overall area and also work as a non-volatile storage. Our proposed system shows what researchers should consider while designing security for these IoT system and how emerging technology can help to reduce overhead and improve performance.

TABLE II
Overhead in different phases of the system

Overhead	Enc. key generation	Memory write	Memory read	Dec. key generation
Power (μW)	24.61	0.79	0.44	24.96
Delay (clock cycles)	1	N	N	1

Acknowledgment

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-16-1-0301. Any opinions, finding, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

References

[1] L. O. Chua, "Memristor-the missing circuit element," *IEEE Trans. on Circuit Theory*, vol. 18, no. 5, pp. 507–519, September 1971.

[2] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.

[3] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *44th ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2007, pp. 9–14.

[4] G. Rose and C. Meade, "Performance analysis of a memristive crossbar PUF design," in *52nd ACM/EDAC/IEEE Design Automation Conf. (DAC)*, June 2015, pp. 1–6.

[5] M. Uddin, M. B. Majumder, G. S. Rose, K. Beckmann, H. Manem, Z. Alamgir, and N. C. Cady, "Techniques for improved reliability in memristive crossbar PUF circuits," in *IEEE Comp. Society Annual Symp. on VLSI (ISVLSI)*, July 2016, pp. 212–217.

[6] M. Uddin and *et.al.*, "Design considerations for memristive crossbar physical unclonable functions," *J. Emerg. Technol. Comput. Syst.*, vol. 14, no. 1, pp. 2:1–2:23, Sep. 2017.

[7] M. Uddin, M. B. Majumder, and G. S. Rose, "Robustness analysis of a memristive crossbar PUF against modeling attacks," *IEEE Transactions on Nanotechnology (TNANO)*, vol. 16, no. 3, pp. 396–405, May 2017.

[8] R. Horstmeyer, I. M. Vellekoop, S. Assawaworrarit, B. Judkewitz, and C. Yang, "Physical key-protected one-time pad," *Scientific Reports, Nature*, vol. 3, no. 3543, December 2013.

[9] H. Manem and G. S. Rose, "A read-monitored write circuit for 1t1m multi-level memristor memories," in *IEEE International Symposium of Circuits and Systems (ISCAS)*, May 2011, pp. 2938–2941.

[10] M. B. Majumder, M. Uddin, G. S. Rose, and J. Rajendran, "Sneak path enabled authentication for memristive crossbar memories," in *Hardware-Oriented Security and Trust (AsianHOST), IEEE Asian*, 2016, pp. 1–6.