International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 06 | Jun 2026 www.irjet.net p-ISSN: 2395-0072

Design and Implementation of a Configurable Ring Oscillator Physically Unclonable Function (PUF) Using Logic Gates for AES-128 Key Generation

Prof. P. V. Sridevi M.E [1], PhD, Inaganti Himavardhini[2], Inapakurti Jahnavi[3], Didla John Paul[4], Damodarapatruni Manoj Karthik[5]

[1]*HOD, Department of Electronics and Communication Engineering, Andhra University College of Engineering (A), Andhra Pradesh, India*

[2,3,4,5]*(Students, Department of Electronics And Communication Engineering), Andhra University College of Engineering(A), Andhra Pradesh, India*

Abstract-*Physically Unclonable Functions (PUFs) have emerged as a lightweight hardware security primitive for device authentication and cryptographic key generation. This project presents a Configurable Ring Oscillator (RO) PUF using Hybrid Logic Gates, enhanced with an area-optimized hardware comparator unit and integrated with an AES-128 key generation block (10 rounds). The configurable RO-PUF exploits manufacturing process variations to generate unique and unpredictable responses, ensuring secure device identification without storing secret keys in memory. The proposed hybrid logic gate design reduces power consumption and improves oscillation stability. Furthermore, the conventional comparator is replaced with an area-efficient hardware comparator, significantly reducing silicon area and delay while maintaining high reliability. The generated PUF response is used as a 128-bit cryptographic key input for AES encryption, enabling secure and tamper-resistant key generation for embedded and IoT devices. The overall architecture provides a low-cost, lightweight, and secure solution suitable for modern hardware security applications.*

Key Words: Physically Unclonable Function (PUF), Ring Oscillator PUF (RO- PUF), Configurable PUF Architecture, Hybrid Logic Gates, Hardware Security, Area-Optimized Comparator, AES-128 Encryption, Cryptographic Key Generation.

1.INTRODUCTION

PUF is one of the most promising security primitives for resource constrained scenarios, e.g. the Internet of Things (IoT) applications. A PUF utilizes process variations to generate unique CRPs for each single chip. Even when manufactured under the same condition, the CRPs of a specific chip will be different and these unique CRPs can be used to prevent the adversary from an unauthorized copy of the chip. To date, various PUF structures have been proposed and they can be classified to delay based PUFs and memory based PUFs. The most cited designs include RO PUFs, Arbiter PUFs and SRAM PUFs [1]. CRO PUF, as an improved design of the conventional RO PUF, aims to acquire a high uniqueness, reliability and lower cost. A typical CRO PUF is composed of switching components and delay components, where the switching components provide the reconfigurability to the PUF structure when the design is completed and deployed. The MUXs in [2]–[4] and the tristate gates in [5] act as the switching components and selects which delay unit is involved in the construction of the CRO PUF. As CMOS integrated circuits (ICs) are reaching its limitation, nano-device based PUFs provide new possibilities for reliable security circuits. It is a great challenge for IC designers and foundries to scale down the devices to the nanoscale size since the unpredictable thickness and the cross-sectional area introduce large process variations. However, this is a great opportunity for the PUF designers since the multiple variations and noise sources can improve the uniqueness and randomness of the PUF responses. Moreover, the entropy of a PUF design can be increased significantly due to the unpredictable variations and noises. Among the emerging PUFs based on nanotechnology, the resistive random access memory (RRAM, also referred to as memristor or ReRAM [6]). based PUF designs are one of the most promising approaches. Conventional RRAM based PUF designs mainly utilize the resistance variations and mismatch of the RRAM at a fixed programming voltage or programming time

2. LITERATURE SURVEY

A.RRAM BASED PUFS RRAMs can be classified as bipolar or unipolar according to the switching behaviors. The bipolar RRAM based on metal oxide(shown in Fig. 1(a)) is one of the most promising candidates for applications in memory computing or configurable logic [7]. By applying a proper positive or negative pulse to the selected RRAM cell, the resistance of the RRAM will be switched between High-Resistance State (HRS) and Low-Resistance State

(LRS), corresponding to ON and OFF of the circuit. When the RRAM is configured in the low-resistive ON state, it works like a diode. On the contrary, if the RRAM is in the high-resistive OFF state, the current that can flow through the RRAM is very small. RRAM is a popular primitive for PUF designs, especially for memory based PUF designs. A survey of recent studies investigating emerging nano-electronic devices to build PUFs is given in [8]. Most of these designs rely on the resistance of the RRAMs and one major issue of the RRAM based PUF design is that the structures need extra analog reading circuits e.g. analog-to-digital converter (ADC) to generate the corresponding response. Furthermore, the resistance of the RRAM is strongly related to the operating temperature and the programming voltages or time. These make them unreliable and conventional RRAM based circuits suffer serious variability [9]. Besides, for crossbar based RRAM PUFs, the accumulated sneak current will also make them unstable and unreliable [10].

B. RRAM CMOS HYBRID DESIGNS RRAM CMOS hybrid

Design offers an opportunity to accomplish high-density circuit [11], since the RRAM has been proven CMOS-compatible and it can be integrated in the metal layers over the CMOS layer [12]. Previous research have shown that RRAM based switch box (SB) can be applied to the design of high performance and high density FPGAs [12], [13]. Furthermore, the RRAM CMOS hybrid circuits can accelerate neural network computations [14]. Inspired by the idea of the RRAM CMOS hybrid designs in the FPGA, hn-CRO PUF is proposed in the paper. Different from the previous RRAM based PUF designs [15]–[17], in this paper, the RRAMs in this work are configured at the fixed state and work like a programmable nano-switch while the CMOS inverters act as the delay components, as shown in Fig. 1(b), which makes the proposed hn-CRO PUF more reliable.

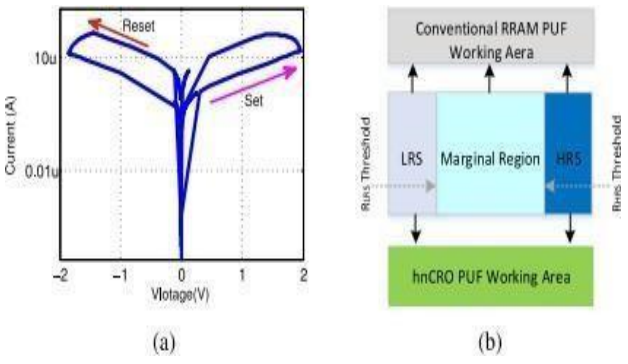

Fig. 1. RRAM circuits: (a) I-V curve of the RRAM, and (b) working area of the RRAM PUF.

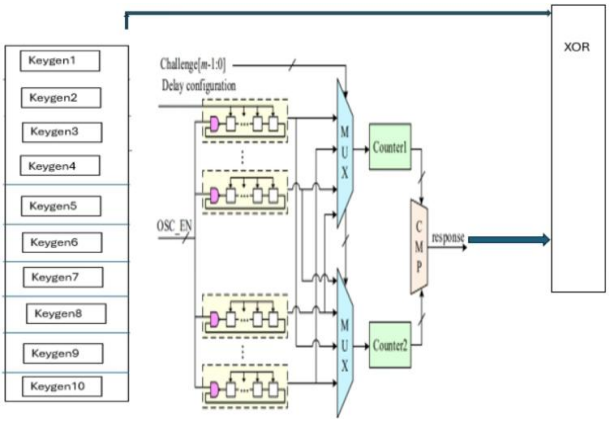

Fig. 2. Proposed Block diagram

The proposed architecture consists of five major blocks. Multiple challenges are applied sequentially. Each challenge produces one response bit.

After 128 comparisons:

Bits are concatenated → 128-bit PUF key. Optional filtering improves reliability.

1. Configurable Ring Oscillator Array
2. Challenge Generator / Selector
3. Area-Optimized Hardware Comparator
4. Response Stabilization & Key Formation
5. AES-128 Encryption Core (10 Rounds)
6. The overall flow is:

Challenge → RO-PUF → Comparator → 128-bit Key → AES Encryption

The overall architecture of the proposed PUF, which does not include the controlling part, is shown in Fig. 2 Based on the proposed DCUs, we can construct a hybrid configurable RO (HC-RO) with a NAND gate as the leading unit. Meanwhile, the following two rules must be abided by:

- The RO_I of current DCU should connect to the RO_F of the previous DCU and the RO_F of current DCU should connect to the RO_I of the next DCU.
- The total number of reverse logic units in a HC-RO should be odd, which includes the leading NAND gate, DCU-1, DCU-3 and inverter if exists. Fig. 3 shows an example of our hybrid configurable RO. As we can see, a NAND gate is placed at the beginning of the RO and the total number of reverse

logic units is three (i.e., one NAND gate, one DCU-3 and one DCU-1), exactly an odd number. When RO_EN is low, RO_F holds high all the time. Once RO_EN turns to high,

the whole RO can oscillate at 2 5 = 32 kinds of frequencies determined by the CI[4:0]. Of course, from the perspective of area cost, it would be better to use DCU-1 or DCU-3 as the stage unit.

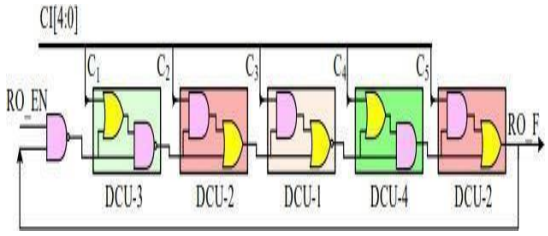

Fig. 3. An example of hybrid configurable RO

For the falling time, let us assume RO_I is 1 at the beginning, thus leading RO_F to be 1 no matter what value CI holds. When RO_I pulls down, the 'B' input of the OR gate will pull down. Also, the output of the AND gate (i.e., AND_OUT) will pull down, no matter what value CI holds. That is to say the 'A' input of the OR gate will also pull down. Hence, the RO_F will always follow the RO_I to pull down. However, the falling time from the RO_I to RO_F is not the same for different CI values: 1) When CI = 1, the initial value of the AND_OUT is 1. So when RO_I pulls down, it must take a time of dAND_10 to pull down the AND_OUT firstly. After that, it will take another time of dOR_00 to pull down the RO_F. In other words, it takes a total time of (dAND_10 + dOR_00) for RO_F to become 0 after RO_I pulls down. 2) When CI = 0 (as shown in Fig. 5(b), Fig. 5(d)), because the initial value of AND_OUT is already 0, it only takes a time of dOR_00 for RO_F to become 0 after RO_I pulls down. In conclusion, if the DCU-2 is used as a stage of the RO, the value of CI can decide whether an AND gate is included in the path during falling phases. For the rising time, let us assume RO_I is 0 at the beginning, thus leading RO_F to be 0 no matter what value CI holds. when RO_I pulls up, the 'B' input of the OR gate will also pull up. Hence, the RO_F will certainly follow RO_I to pull up, no matter what value AND_OUT is. However, the rising time from the RO_I to RO_F varies a little for different CI values:

1) When CI = 1 , the initial value of the AND_OUT is 0. So when RO_I pulls up, the AND_OUT (i.e., the 'A' input of the OR gate) is rising during the period of dAND_11.

2) When CI = 0 , the AND_OUT always holds 0.
The generated PUF key is directly used as the AES key.
Area-Optimized Hardware Comparator

3. WORKING

1. Counters count oscillations from both ROs during a fixed time window.
2. The hardware comparator compares the two counter values.
3. Output bit generation:

- If RO1 > RO2 → output 1
- If RO1 < RO2 → output 0

Why Area Optimized?

- Reduced logic gates
- Lower silicon area
- Faster comparison
- Lower power consumption

Output: 1-bit PUF response.

AES Operations:

AES performs 10 rounds of encryption:

1. SubBytes
2. ShiftRows
3. MixColumns
4. AddRoundKey

This produces secure encrypted output.

Importance

- Key is **never stored** in memory.
- Key is generated on-chip when needed.
- Prevents key theft and cloning.

Output: Encrypted ciphertext.

4. RESULTS

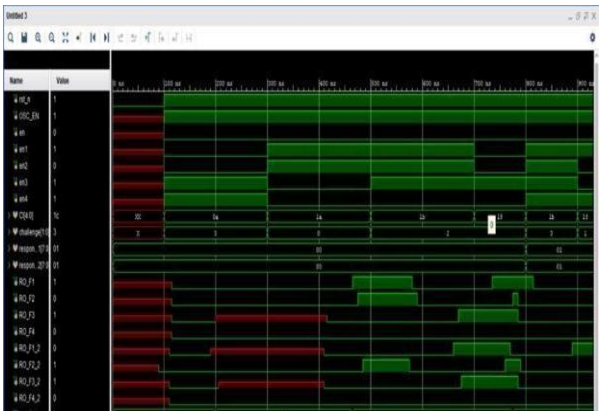

Fig. 4. PUF functional simulation waveform

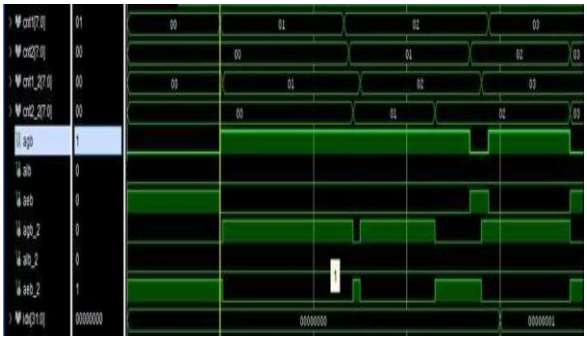

Fig. 5. PUF outputs

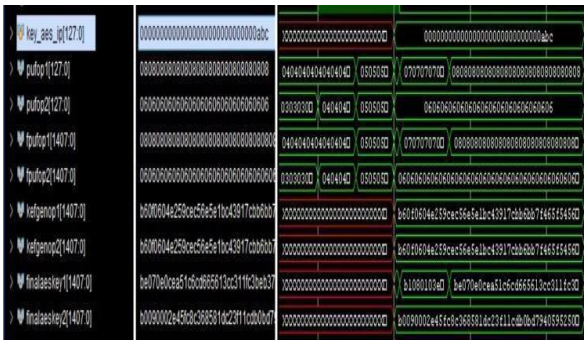

Fig. 6. AES outputs

ADVANTAGES

1. Security Advantages :

- Generates unique and unclonable keys from intrinsic chip variations.
- Eliminates the need for non-volatile memory key storage, preventing key extraction attacks.
- Strong resistance to:
  - Physical tampering
  - Side-channel attacks
  - Reverse engineering

2. Hardware Advantages:

- Hybrid logic gates reduce:
  - Power consumption
  - Propagation delay
  - Transistor count
- Area-optimized comparator:
  - Reduces chip area significantly
  - Improves comparison speed
  - Enables scalability for large PUF arrays

3. System Advantages:

- Lightweight and suitable for resource-constrained devices
- Scalable and reconfigurable architecture
- Low cost and easy VLSI implementation

APPLICATIONS

1. IoT Device Security:

- Secure boot and firmware authentication
- Device identity for smart home devices
- Key generation for wireless sensor networks

2. Embedded Systems:

- Secure microcontrollers and FPGAs
- Automotive ECU security
- Smart cards and RFID authentication

3. Cloud & Edge Security:

- Hardware root of trust
- Secure key provisioning
- Device-to-cloud authentication

4. Military & Defense:

- Secure communication hardware
- Anti-counterfeiting of electronic components

CONCLUSION

This project demonstrates a secure and area-efficient hardware security architecture using a configurable RO-PUF integrated with an AES-128 key generation block. The use of hybrid logic gates improves energy efficiency and stability, while the area-optimized comparator reduces hardware overhead. By eliminating the need for stored secret keys and generating cryptographic keys directly from silicon variations, the proposed system enhances hardware trust and resilience against attacks. The architecture is highly suitable for low-power, resource-constrained, and security-critical applications.

FUTURE SCOPE

Technical Improvements

- Implement Machine Learning attack resistance techniques
- Add error correction codes (ECC) to improve reliability
- Develop temperature and voltage variation compensation

Architecture Enhancements

- Integration with AES-256 or lightweight cryptography
- FPGA and ASIC fabrication for real-chip validation
- Implement Strong PUF architecture for large CRP sets

REFERENCES

1) J. Delvaux and I. Verbauwhede, "Side Channel Modeling Attacks on 65 nm Arbiter PUFs Exploiting CMOS Device Noise," IEEE Trans. Inf. Forensics Security, vol. 10, no. 3,pp. 725–738, Mar. 2015.

2) G. T. Becker, "The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs," in Proc. IEEE CHES, 2015, pp. 535–555.

3) R. Maes, " Physically Unclonable Functions: Constructions, Properties and Applications," Springer, 2016.

4) U. Ruhrmair and D. E. Holcomb, "PUFs at a Glance," in Proc. IEEE DATE, 2014, pp. 1–6.

5) Y. Gao et al., "Emerging Physical Unclonable Functions With Nanotechnology," IEEE Access, vol. 4, pp. 61–80, 2016.

6) Maiti and P. Schaumont, "Improved Ring Oscillator PUF: An FPGA-Friendly Secure Primitive," J. Cryptology, vol. 24, no. 2, pp. 375–397, 2011.

7) K. Yang et al., "A Physically Unclonable Function With BER < 1e−8 for Robust Chip Authentication Using Oscillator Collapse in 40 nm CMOS," IEEE ISSCC, 2015.

8) M. Tehranipoor and C. Wang, Introduction to Hardware Security and Trust. Springer, 2012.

9) J. Guajardo et al., "FPGA Intrinsic PUFs and Their Use for IP Protection," in Proc. IEEE CHES, 2007, pp. 63–80.

10) S. Katzenbeisser et al., "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions," IEEE Trans. Inf. Forensics Security, vol. 13, no. 2, pp. 352–365, 2018.

11) X. Xu et al., "A Reliable and Reconfigurable FPGA-Based Ring Oscillator PUF," IEEE Trans. CAD, vol. 37, no. 11, pp. 2783–2796, Nov. 2018.

12) H. Handschuh and H. Gilbert, "AES Candidate Algorithm Submission," NIST AES Proposal, 1999.

13) S. Sutar et al., "Lightweight Cryptography Implementation Using PUF - Based Key Generation," IEEE Access, vol. 9, pp. 114563–114575, 2021.

14) Y. Cao et al., "A 1.4 pJ/bit Fully Synthesizable PUF-Based Cryptographic Key Generator in 65 nm CMOS," IEEE JSSC, vol. 52, no. 10, pp. 2671–2682,Oct. 2017.

15) M. Hiller et al., "Systematic Analysis of RO PUFs and Their Secure Key Generation Capabilities," IEEE Trans. Dependable Secure Comput., vol. 17, no. 3, pp. 553–566.