RESEARCH PAPER

Hardware Authentication and Encrypted Computation Using a XOR-Arbitar PUF-Based Secure ALU Architecture

P. Narsimha[1*], **Dr. V. Kumara Swamy**[1], **Dr. T. Ramaswamy**[1]
[1]Department of ECE, Sreenidhi Institute of Science and Technology, Hyderabad - 501302, Telangana, India
[1]M.Tech (P. Narsimha)
[1]Professor and Dean of Academics (Dr. V. Kumara Swamy)
[1]Associate Professor and MTech Coordinator (Dr. T. Ramaswamy)
***Corresponding Author:** P. Narsimha, M.Tech, Department of ECE, Sreenidhi Institute of Science and Technology, Hyderabad - 501302, Telangana, India

ABSTRACT: ***Hardware security has become essential for embedded and Internet of Things (IoT) systems due to the growing risks of hardware cloning, unauthorized access, and physical tampering. This paper proposes a Physically Unclonable Function (PUF)-based Secure Arithmetic Logic Unit (ALU) that integrates hardware authentication, tamper protection, and secure data processing within a lightweight architecture. The proposed design employs an XOR Arbiter PUF to generate a unique device-specific secret key, which is verified through an authentication module before enabling ALU operation. A tamper control mechanism prevents unauthorized execution by disabling the ALU whenever authentication fails. Upon successful verification, the ALU performs secure arithmetic, logical, and comparison operations, and the generated results are protected using lightweight XOR-based encryption prior to transmission or storage. The architecture is implemented in Verilog HDL and validated on a Xilinx FPGA using Vivado. Experimental results demonstrate reliable authentication, effective tamper detection, secure computation, and encrypted output generation with low hardware overhead, making the proposed design well suited for FPGA-based embedded systems, IoT devices, and other resource-constrained secure computing applications.***
Keywords: Nexys -4 DDR FPGA Board, ALU (Arithmetic Logic Unit) Architecture, XOR-Arbitar PUF, Encryption based security for Authentication of Hardware.

How to cite this article: P. Narsimha, V. Kumara Swamy, T. Ramaswamy. Hardware Authentication and Encrypted Computation Using a XOR-Arbitar PUF-Based Secure ALU Architecture. Int J Drug Deliv Technol. 2026;16(79s):844-849. DOI: 10.25258/ijddt.16.79s.151.

I. INTRODUCTION

As embedded systems, the Internet of Things (IoT) devices and FPGA-based computing systems are growing in scale, their demand for secure hardware level security has grown as well. In conventional systems, cryptographic keys are usually kept in memory, thus they can be easily duplicated, reverse engineered, extracted, accessed, and tampered with physically. Hence, it is essential to have hardware based security solutions for safeguarding the identity and sensitive computations [3], [6], [11].

Physical Unclonable Functions (PUFs) generate device-specific responses by taking advantage of unpredictable manufacturing differences in Integrated Circuits (ICs). The responses may be employed in the process of hardware authentication, hardware identification and dynamically generating secret-keys without permanently storing an external memory the full key [1]–[3]. In addition to its lightweight implementation and low hardware overhead, PUF-based authentication is well suited for FPGA and resource-constrained IoT systems [4, 5, 12].

But, there are potential machine-learning and modelling attacks against conventional Arbiter PUFs. The response complexity of XOR Arbiter PUF architectures is enhanced by XORing the output of several arbirers in chains. These architectures offer greater unpredictability and better response-prediction attack resistance [8, 9]. PUFs have also been recently shown to be useful in secure edge computing, authentication, and reconfigurable hardware platforms [10], [15], [16].

The Arithmetic Logic Unit (ALU), the main computing unit in digital processors, is used to carry out arithmetic, logical and comparison operations. Conventional ALUs typically perform operations with no checks on the authenticity of the hardware, allowing illegitimate operations to be carried out and data to be manipulated. Therefore, the combination of PUF-based authentication and ALU can be used to guarantee that only once the device is authenticated would computation be performed.

This work suggests a Secure ALU Architecture based on a PUF, which combines XOR Arbiter PUF, key generation, key verification, tamper control, secure computation, encryption, and decryption. A reference key is stored and compared

to the generated PUF key. If the authentication is successful, the ALU carries out the selected operation, otherwise, the tamper controller disables the ALU and prevents unauthorized operation. The result of the calculation is protected with a simple XOR encryption.

The proposed architecture is described in Verilog HDL and synthesized using the Xilinx Vivado Design Suite [13] and [14]. It is designed to offer the hardware authentication, tamper protection, secure ALU execution and encrypted output generation in a single FPGA design. Hence, it is suitable for embedded applications where security is required, embedded IoT, trusted processors, smart cards, resource-constrained systems with security applications.

II. LITERATURE SURVEY

Shanta et al. [1] proposed a Physically Unclonable and Reconfigurable Computing System (PURCS) which incorporates device authentication, logic locking and reconfigurable computation. The architecture is based on process variations and chaos based logic and can generate device-specific challenge–response pairs. The design shown was reported to be highly unique, uniform and robust against common modeling attacks and therefore is potentially applicable to secure computing architecture.

Zalivaka et al. [2] explored reliable PUF based authentication of FPGA devices. They concentrated on increasing the stability of Arbiter-PUF responses and decreasing the susceptibility to a machine-learning-based attack. These reliability-enhancing methods play a crucial role in the authentication of FPGA as environment and implementation variations can generate unreliable pairs of challenge and response.

Gao et al. [3] gave a general overview about the working principles, implementation strategies, security applications and practical constraints of PUFs. The main challenges that were identified in the study were reliability, environmental sensitivity, modelling attacks, and hardware integration. It also brought up PUFs as a promising option for authentication, anti-counterfeiting, generation of secret keys, and a lightweight hardware security. The verifiable publication is Nature Electronics, vol. 3 pp. 81–91, 2020, rather than Electronics, vol. 9, no. 11.

For the resource constrained IoT devices, Nabeel and Farooq [4] presented a lightweight PUF-based authentication scheme. This protocol relies on the PUF responses that are device dependent, which lessens the need for permanently stored secret keys and computationally complex cryptographic functions. This method can be applied for secure embedded systems that have limited storage, power and communication requirements.

Tao and Plusquellic [5] implemented FPGA-oriented PUF architectures to ensure security in embedded applications. They focused on designing PUF circuits in configurable FPGA resources and assessing response uniqueness, reliability and hardware cost. FPGA-based PUF designs offer a flexible platform for device identification, key generation and authentication.

M. Tehranipoor and A. Asadizanjani [6] talked about the present threats and importance of hardware-based trust mechanisms in hardware security. The challenges they discussed were counterfeit integrated circuits, reverse engineering, hardware Trojans, supply chain attacks, and secret-key exposure. It has highlighted the implementation of authentication, tamper resistance and secure key management in the modern electronic systems.

Li et al. [7] proposed a new real-time image-acquisition chip based on a triple-hybrid encryption system including AES, ECC and SRAM-PUF to mitigate the risks. The PUF, AES and ECC are used for chip authentication, anti-cloning protection, high data speed encryption and cryptographic key protection, respectively. The architecture was designed and realized on FPGA and ASIC platforms and proved the possibility to combine authentication and encryption in a single hardware platform.

Current research focuses are mainly on PUF authentication, reliable response generation, lightweight protocols or encrypted hardware, separately. A unified architecture which combines PUF-based key generation, key verification, tamper control, authenticated ALU execution, and encrypted output generation with low FPGA resource overhead is thus required.

III. PROPOSED METHODOLOGY

Proposed method workflow and steps are explained in detail below,

Fig.3.1 The Overall Workflow of the Proposed System

The proposed PUF-based Secure Arithmetic Logic Unit (ALU) architecture integrates hardware authentication, tamper protection, secure computation, and data encryption to provide a lightweight security solution for FPGA-based embedded systems. As illustrated in Fig. 1, the architecture consists of an XOR Arbiter PUF, key generator, key register, key verification unit, tamper controller, secure ALU, encryption module, and decryption module. The operation of the proposed system is described as follows.

Step 1: Challenge-Based Key Generation

The authentication process begins by applying an 8-bit challenge to the XOR Arbiter PUF. The PUF comprises three arbiter chains that generate unique responses for the applied challenge. These

responses are XORed to produce a reliable response bit. The response bits are accumulated over successive clock cycles in the key generator to construct an 8-bit device-specific secret key.

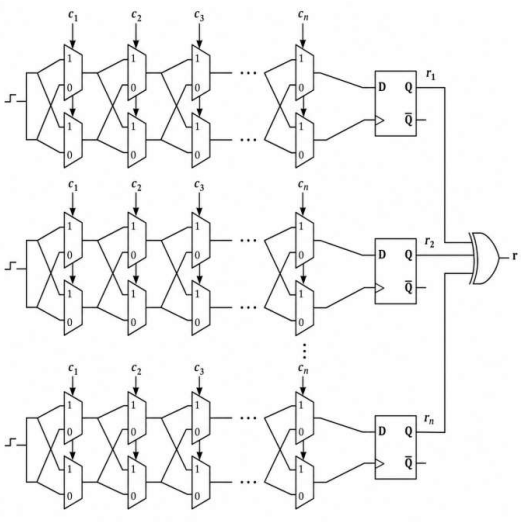

Fig2: Architecture of the Proposed XOR Arbiter PUF for Device-Specific Key Generation

1. *XOR Arbiter PUF Response*

$$□ = □_{1} \oplus □_{2} \oplus \ldots \oplus □_{□}$$

The individual responses generated by the parallel Arbiter PUF which are indicated by r1, r2,…$□_{□}$ chains are combined using XOR to obtain the final device-specific response bit which is indicated by r.

2. *Hardware Authentication*

$$□□□h = \begin{cases} 1 & □_{□□□} = □_{□□□} \\ 0 & □_{□□□} \neq □_{□□□} \end{cases}$$

Authentication is successful only when the PUF-generated key $□_{□□□}$ exactly matches the stored reference key $□_{□□□}$; otherwise, the ALU remains disabled.

In above equation , 1 indicates successful authentication while 0 indicates unsuccessful authentication.

3. *Output Encryption and Decryption*

$$□ = □_{□□□} \oplus □_{□□□}$$

$$□_{□□□} = □ + □_{□□□}$$

Here, $□_{□□□}$ is the ALU result $□_{□□□}$is the authenticated secret key, and C is the encrypted output. Applying the same key again recovers the original result.

Step 2: Key Storage and Verification

The generated secret key is stored in the key register and compared with the pre-stored reference key using the key verification module. Authentication is considered successful only when both keys match exactly. The verification output acts as the authentication signal for the remaining modules.

Step 3: Tamper Detection and Access Control

The authentication signal is provided to the tamper controller, which determines whether the ALU is permitted to operate. If the generated key matches the stored key, the tamper controller enables the ALU. Otherwise, the ALU remains disabled, preventing unauthorized hardware access and protecting the system from cloning and tampering attacks.

Step 4: Secure ALU Operation

After successful authentication, the secure ALU receives two 8-bit operands and a 4-bit control signal to perform the selected operation. The implemented ALU supports arithmetic, logical, and comparison functions required for secure embedded computing. If authentication fails, the ALU output is forced to zero.

Step 5: Output Encryption

To ensure confidentiality, the ALU output is encrypted using the authenticated PUF-generated key. A lightweight XOR-based encryption scheme is adopted, where the encrypted output is obtained by combining the ALU result with the secret key. This approach introduces minimal hardware overhead while protecting sensitive data during transmission or storage.

Step 6: Output Decryption

The encrypted output is decrypted using the same secret key through another XOR operation. Since XOR is a reversible operation, the original ALU result is recovered without any loss of information, ensuring correct and secure data retrieval.

Step 7: System Verification

The complete architecture is implemented in Verilog HDL and verified using Xilinx Vivado. Functional simulations confirm correct PUF-based key generation, successful hardware authentication, tamper detection, secure ALU operation, and accurate encryption and decryption of the output data

IV. RESULTS ANALYSIS

Proposed application results of both software and hardware implementation are explained in detail below,

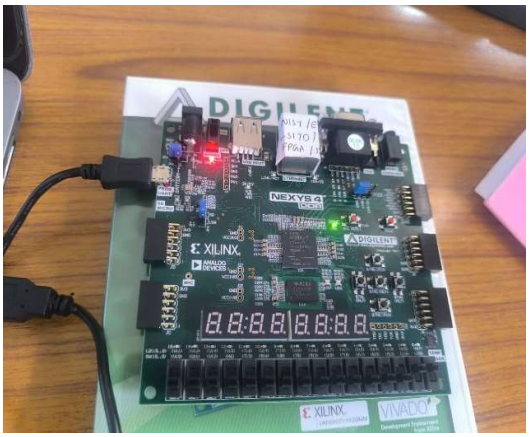

Fig.4.1 Nexys -4 DDR FPGA Board used in this Application

Nexys-4 DDR FPGA Board Used for Hardware Implementation. The proposed application is validated using hardware Nexys-4 DDR board for which code is developed on Vivado-Xilinx software. Before execution of ALU, XOR Arbiter PUF, modules of decryption and encryption, FPGA gives the required hardware modules which help in integration. FPGA authentication is performed using PUF-based secrete key before start of ALU operations. VIO interface as well as Vivado Hardware Manager are utilized for monitoring results of ALU, keys which are generated and stored, operands, etc. Implementation on hardware with FPGA board shows the possibility of practical applications of devices for IOT, higher computational security, embedded systems with FPGA boards due to lightweight nature of encryption, higher security with hardware authentication.

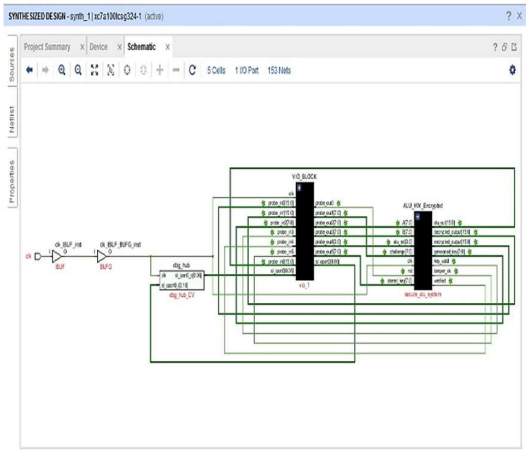

Fig.4.2 VIO Hardware based Architecture Implementation on FPGA Board

The above figure shows hardware-based schematic of proposed system with virtual input and output designed using Vivado Xilinx in the post synthesis phase. In the hardware testing VIO block helps to get real time updates of the signals present in the FPGA, which avoids additional circuits and switches. Vivado hardware manager can monitor different parameters which are important for this application such as ALU output, PUF challenge, A and B which are input operands. ALU computation is enabled only when reference key is matched with key generated by PUF with the use of block named 'secure_alu_system'. Total systems FPGA verification can be done in real time using interception present in hardware level with the use of architecture of encryption, computation with high security, authentication proposed and VIO interface.

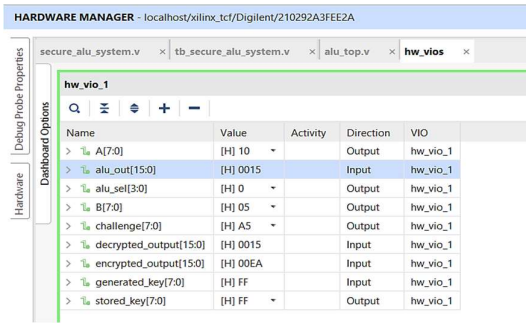

Fig. 4.3. Output of PUF-Based Secure ALU System with VIO real time Monitoring

VIO interface and Vivado hardware manager are used to verify the architecture of secure ALU in real time scenarios. The operands present in above example are B=0x05 and A=0x10 and the PUF observed is 0xA5. In above figure , ALU operations are enabled due to hardware authentication as 0xFF is reference key and 0xFF is the generated key by device. Two input operands are getting addition operation as command alu_sel=0 generates the 0x0015 result. Even after decryption received 0x0015 result which is matching with original value.

Thus, the hardware observation confirms correct authentication, secure ALU execution, encryption, and decryption on the implemented FPGA system.

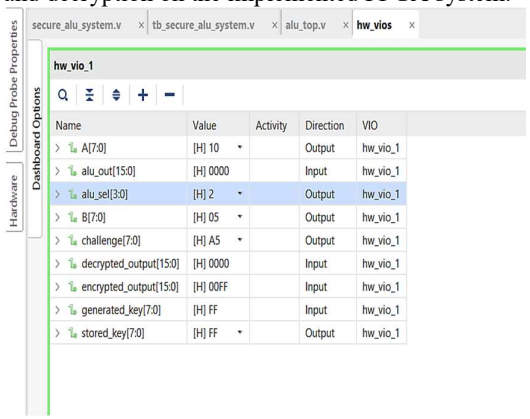

Fig. 4.4 PUF-based ALU system with verification of AND operation in hardware

VIO interface and Vivado is utilised for secure alu operations in the verification of FPGA in real time. A=0x10 and B=0x05 are the two operands considered for this operation with 0xA5 is PUF challenge. 0xFF is the PUF key generated and it is

matching with 0xFF which is reference key which confirms the authentication of hardware. Alu_sel=0x2 is selected for operation of bitwise AND which gives the output of 0x0000. Authentication key is used for encryption of results of secure ALU. The same key is used for decryption to recover the original results. This indicates that the authentication is proper , AND calculations are secured with encryption/decryption design for given data.

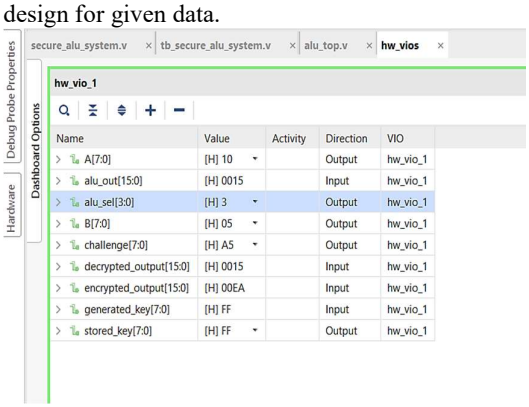

Fig. 4.5. PUF-Based Secure ALU results for OR Operation with verification in real time hardware

In PUF based ALU, OR operation is performed with hardware verification with the help of VIO interface and Vivado hardware manager. 0xA5 is taken as PUF challenge with A=0x10 as first operand and B=0x05 is the second operand. Reference key and the PUF hey generated are matching with 0xFF which indicates that the hardware authentication is performed successfully.

Alu_sel=03 is taken to perform operations with bitwise OR operation. The alu_out after encryption obtained is 0x0015 while same results obtained after decryption which indicates that hardware authentication is successful with securing the OR operation.

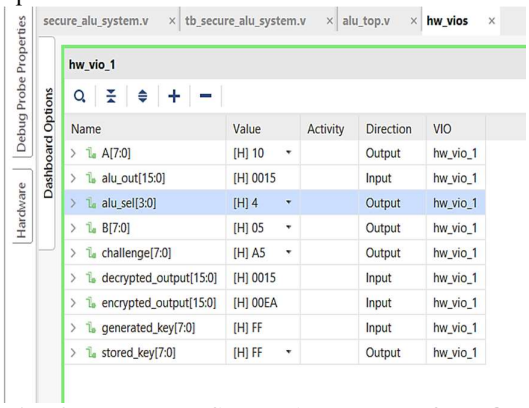

Fig. 6. PUF-Based Secure ALU results for XOR operation with verification in real time hardware.

In PUF based ALU, XOR operation is performed with hardware verification with the help of VIO interface and Vivado hardware manager. A=0x10 and B=0x05 are the two operands taken as input to verify ALU results with hardware implementation and 0xA5 is the PUF challenge. Authentication of hardware got successful as 0xFF which is reference key is matching with the key generated. Alu_sel= 0x4 which indicates that the operation is selected as XOR. The decrypted results are 0x0015 which is matching with original input data which is obtained by applying decryption to encrypted data as 0x00EA which indicates successful working of computation of XOR is secure and correct.

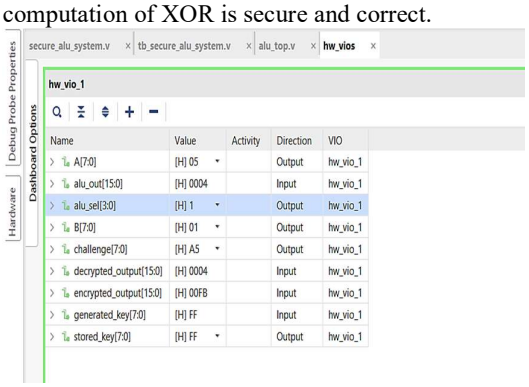

Fig. 6. PUF-Based Secure ALU results for subtraction operation with verification in real time hardware.

In the above figure, PUF based ALU design using VIO interface and Vivado hardware manager is performed on FPGA board. Two operands such as A=0x05 and B=0x01 are taken as input and subtraction operation is performed with 0xA5 as PUF challenge. 0xFF is the reference key which matches with generated key and hence the authentication of hardware is successful. alu_sel = 0x1, is selected for subtraction operation so obtained result is 0x04.

The decrypted results also obtained as 0x0004 which shows that decryption is successful and hence PUF based authentication works perfectly on FPGA with exact data recovery , secure operation of subtraction in FPGA hardware board.

REFERENCES

[1] A. S. Shanta, M. B. Majumder, M. S. Hasan, and G. S. Rose, "Physically Unclonable and Reconfigurable Computing System (PURCS) for Hardware Security Applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 3, pp. 405–418, 2020.

[2] S. S. Zalivaka, A. A. Ivaniuk, and C. H. Chang, "Reliable and Secure PUF-Based Authentication for FPGA Devices," *IEEE Access*, vol. 8, pp. 146402–146412, 2020.

[3] N. Gao, S. F. Al-Sarawi, and D. Abbott, "Physical Unclonable Functions: Challenges and

Opportunities for Hardware Security," *Electronics*, vol. 9, no. 11, 2020.

[4] M. Nabeel and M. A. Farooq, "PUF-Based Lightweight Authentication Protocol for IoT Devices," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4797–4808, 2021.

[5] Y. Tao and J. Plusquellic, "FPGA-Oriented Physical Unclonable Functions for Secure Embedded Systems," *Microprocessors and Microsystems*, vol. 82, 2021.

[6] M. Tehranipoor and N. Asadizanjani, "Hardware Security and Trust: Current Trends and Future Directions," *IEEE Design & Test*, vol. 38, no. 3, pp. 8–17, 2021.

[7] J. Li, Y. Luo, F. Wang, and W. Gao, "Design and Implementation of Real-Time Image Acquisition Chip Based on Triple-Hybrid Encryption System," *Electronics*, vol. 11, no. 18, Art. no. 2925, 2022.

[8] J. Delvaux and D. Gu, "Machine Learning Resistant PUF Architectures for Secure Hardware Applications," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1256–1268, 2022.

[9] A. Maiti and P. Schaumont, "Advanced FPGA-Based PUF Architectures for Lightweight Security," *IEEE Access*, vol. 10, pp. 84523–84536, 2022.

[10] Z. Chen, M. Wu, Y. Zhou, R. Li, J. Tan, and D. Ding, "PUF-CIM: SRAM-Based Compute-in-Memory With Zero Bit-Error-Rate Physical Unclonable Function for Lightweight Secure Edge Computing," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 31, no. 8, pp. 1234–1247, 2023.

[11] K. Xiao, M. Forte, and M. Tehranipoor, "Emerging Trends in Hardware Security Using Physical Unclonable Functions," *ACM Computing Surveys*, vol. 55, no. 2, pp. 1–34, 2023.

[12] H. A. Moghaddam and M. M. Kermani, "PUF-Based Secure Authentication Architectures for Resource-Constrained IoT Devices," *IEEE Access*, vol. 11, pp. 34621–34635, 2023.

[13] AMD Xilinx, *Vivado Design Suite User Guide: Synthesis (UG901)*. San Jose, CA, USA: AMD Xilinx, 2023.

[14] AMD Xilinx, *Vivado Design Suite User Guide: Implementation (UG904)*. San Jose, CA, USA: AMD Xilinx, 2023.

[15] N. W. Tun and M. Mambo, "Secure PUF-Based Authentication Systems," *Sensors*, vol. 24, no. 16, Art. no. 5295, 2024.

[16] R. Maes and I. Verbauwhede, "Recent Advances in FPGA-Based Physical Unclonable Functions and Their Applications," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 32, no. 4, pp. 567–580, 2024.