

# Implementation of hardware security module using elliptic curve cryptography for cyber-physical system

**B. Muthu Nisha, J. Selvakumar**

Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, India

## Article Info

### Article history:

Received Nov 18, 2024

Revised Sep 9, 2025

Accepted Oct 9, 2025

### Keywords:

Cyber-physical system  
Field programmable gate array  
Hardware security module  
Multi-core  
Smart network meter  
Sustainable development goal

## ABSTRACT

The vision of sustainable development goal 9 (SDG 9) is realized through the integration of innovative technologies in the cyber-physical system (CPS). This work focuses on a smart network meter (SNM) application, designed to manage the extensive big data analytics required for processing and analyzing vast amounts of aggregated data in a short period. To address these demands, an advanced explicitly parallel instruction computing (AEPIC) approach is employed, leveraging a multi-core hardware security module (HSM) built on the elliptic curve cryptography (ECC) algorithm. Implementing the algorithm on various field programmable gate arrays (FPGAs) ensures adaptability to different hardware configurations, delivering scalable and optimized performance for big data aggregation in SNM applications. The proposed module showcases exceptional performance in design analysis. The Virtex-7 FPGA demonstrates excellent suitability for big data analytics in smart network applications, with dynamic power consumption accounting for 55% of total power and an on-chip power of 0.542 watts.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

J. Selvakumar

Department of Electronics and Communication Engineering, SRM Institute of Science and Technology  
Kattankulathur 603203, India

Email: selvakuj@srmist.edu.in

## 1. INTRODUCTION

Sustainable development goal (SDG) 9.4 focuses on upgrading infrastructure and industries by 2030 to enhance sustainability. It highlights resource-use efficiency and the adoption of clean, environmentally sound technologies [1]-[3]. Achieving this objective requires innovative solutions tailored to each country's capabilities, supporting sustainable and inclusive industrial growth. In this context, the smart energy meter (SEM) emerges as a pivotal technology for advancing smart grid infrastructure [4]-[6].

In addition to advancing the "smartness", a cost-effective smart network meter (SNM) is an optimal solution to mitigate network congestion through decentralized data processing, enabling smart energy metering independent of utility server data [7]. However, this SNM depends on the processing capability of Arduino Uno. Arduino Uno supports up to 5 consumers. In this device, the absence of robust security mechanisms leaves such systems vulnerable to cyber-physical attacks.

To address these vulnerabilities, a hardware security module (HSM) is imperative for secure communication and data integrity within smart grid networks [8]. The HSM, a tamper-resistant device, improves encryption, manages cryptographic keys, and safeguards against tampering, ensuring the reliable operation of SNM. As HSMs necessitate efficient public key cryptographic algorithms. Elliptic curve cryptography (ECC) is the ideal choice for such applications [9].

In parallel, ECC, independently introduced by Miller (1986) and Koblitz (1987), offers equivalent security to RSA with significantly smaller key sizes, making it ideal for resource-constrained environments. For instance, a 160-bit ECC key matches the security of a 1024-bit RSA key, offering pros such as reduced circuit area, lower memory requirements, decreased power consumption, and enhanced performance. These attributes make ECC well-suited for multi-core HSM integration in smart grid infrastructure. Furthermore, ECC's adoption in standards such as IEEE 1363 and NIST highlights its reliability and extensive recognition in modern security protocols [10].

This research used an advanced explicitly parallel instruction computing (AEPIC) approach, leveraging multi-core HSMs with finite-field ECC cores optimized for the Koblitz standard curve over the Galois field. Implemented on a field programmable gate array (FPGA) [11], the proposed solution ensures efficient processing, secure data handling, and scalability, meeting the growing demands of big data analytics in SNM applications. The motivation for this work comes from:

- SNMs in earlier implementations or prototypes have relied on Arduino Uno, which, due to its limited processing power, memory, and storage, is unsuitable for handling the big data required in more advanced applications.
- The Arduino Uno's limitations highlight the need for more powerful and efficient hardware solutions. In contrast, the FPGAs, with their inherent parallelism, can approach multiple data streams simultaneously, significantly upgrading throughput and efficiency.
- FPGAs are therefore well-suited for network environments that require high-speed data processing and real-time analytics.

In earlier studies, many HSM techniques have been deployed, such as Sami *et al.* [12] demonstrated the efficacy of the custom hardware security module (CHSM) in handling security challenges associated with heterogeneous integration in the semiconductor industry. Xie *et al.* [13] presented the stepwise decreasing-based heuristic algorithm (SDH) and the interference balancing-based heuristic algorithm (IBH) for HSM-based multicore systems, concentrating on securing in-vehicle networks while meeting stringent delay constraints. Cabrera-Gutiérrez *et al.* [14] highlighted that HSMs serve as hardware-based roots of trust, offering physical protection and an additional security layer within system architectures. Their work primarily emphasized the integration of public-key cryptography algorithms and standards, aiming to enhance overall security by combining HSMs with blockchain technologies in Industrial IoT systems. Murtaza *et al.* [15] carried out a defense-in-depth strategy to secure through HSM, its users, and related services from various threats, ranging from basic shoulder surfing to sophisticated man-in-the-middle attacks. This strategy notably strengthened cryptographic key security by eliminating the need for stored keys and integrating multi-factor authentication. Lin *et al.* [16] initiated a novel tri-functional module using resistive random-access memory (RRAM), which was experimentally demonstrated for the first time, showing its possible for use in tightly restricted internet of things (IoT) applications. According to the previous literature, it is clear that:

- The use of HSMs and heterogeneous computing fabrics for SNMs has been unexplored.
- This paper proposes the implementation of a HSM with heterogeneous compute fabrics, utilizing ECC to enhance both secure module and data handling capabilities for SNMs.
- Combining ECC with multi-core design for HSMs offers a more effective security solution compared to existing big data handling methods.
- To address these challenges, advanced AEPIC is a novel approach to enhance the performance and security for SNMs.

This research is structured as follows: section 2 provides an overview of the proposed computational considerations for implementing HSMs. It outlines the methodology used and presents the pseudo-code modelling of the proposed algorithm. Section 3 discusses the results obtained, evaluating the effectiveness of the proposed HSM design. Finally, section 4 concludes the paper by summarizing the findings and suggesting directions for future research.

## 2. RESEARCH METHOD

The proposed research framework is illustrated in Figure 1. It started with choosing an appropriate cyber-physical system (CPS) application to evaluate a suitable FPGA processor for secure big data computing.

### 2.1. Protection setup

The SNMs act as a cyber-physical interface within a CPS application. This can be built on the safety specification in Figure 2. It is essential for maintaining the security and reliability of smart grid infrastructure.

It applies strict security measures and protocols to protect data integrity, prevent unauthorized access, and contribute to a robust platform for managing communications between smart meters and the grid network [1], [17].

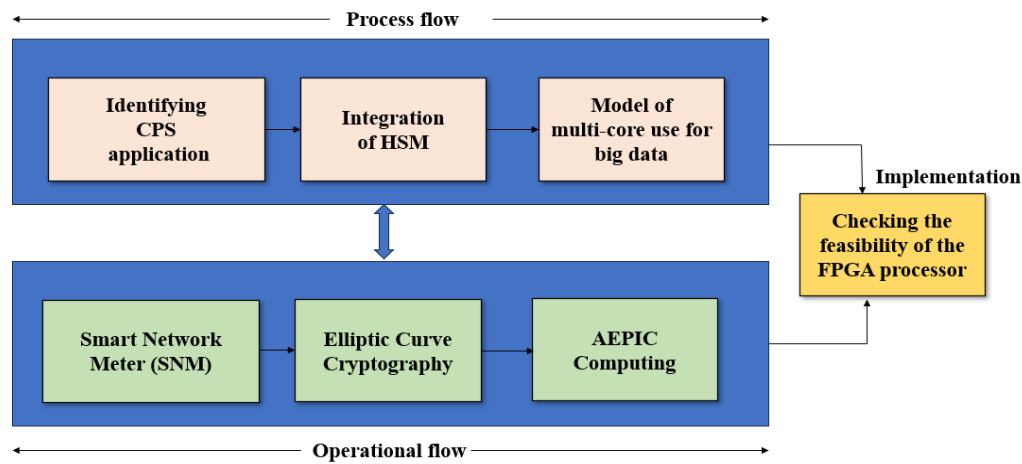


Figure 1. Proposed research framework

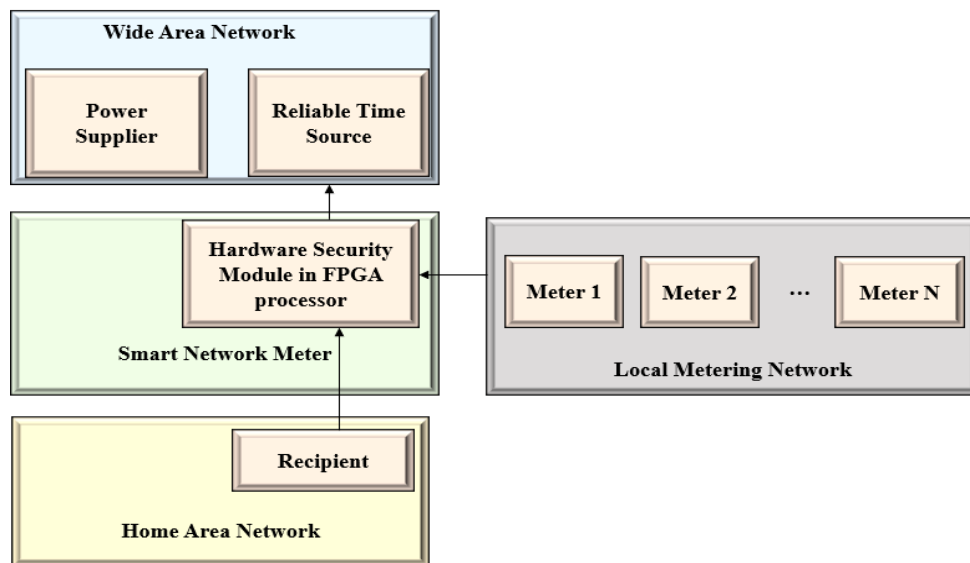


Figure 2. Safety specification for SNM (modified from [17])

## 2.2. Attributes of a two finite fields-based elliptic curve cryptography core

Binary or characteristic-two finite fields (FF) ( $GF(2^m)$ ) are a subset of FFs where the order is a power of two. These fields are particularly most suitable for hardware implementation and binary computer systems due to their inherent effectiveness. The components of  $GF(2^m)$  are represented as binary polynomials, with coefficients restricted to values of 0 or 1. This section outlines the algorithms used for executing FF arithmetic operations in the proposed study. Parallel FF reductions are fundamental to all FF operations and play an essential role in ensuring efficient execution of arithmetic instructions in  $GF(2^m)$ . The proposed HSM design emphasizes the importance of optimizing these operations for secure and efficient ECC [18].

## 2.3. Hardware security module functional computation

The development of ECC within the base of a HSM can be visualized as a pyramid, as shown in Figure 3. Overall, ECC serves as the base of the security mechanisms in HSM. Beneath this, ECC critical

operation is composed of a series of point additions, point doublings, and point multiplications. At the basic level, these point operations rely on modular arithmetic operations, including addition, subtraction, and multiplication [19]. These modular operations, in turn, are integrated as multiple-precision operations on an n-bit processor, supporting efficient computation in the HSM design.

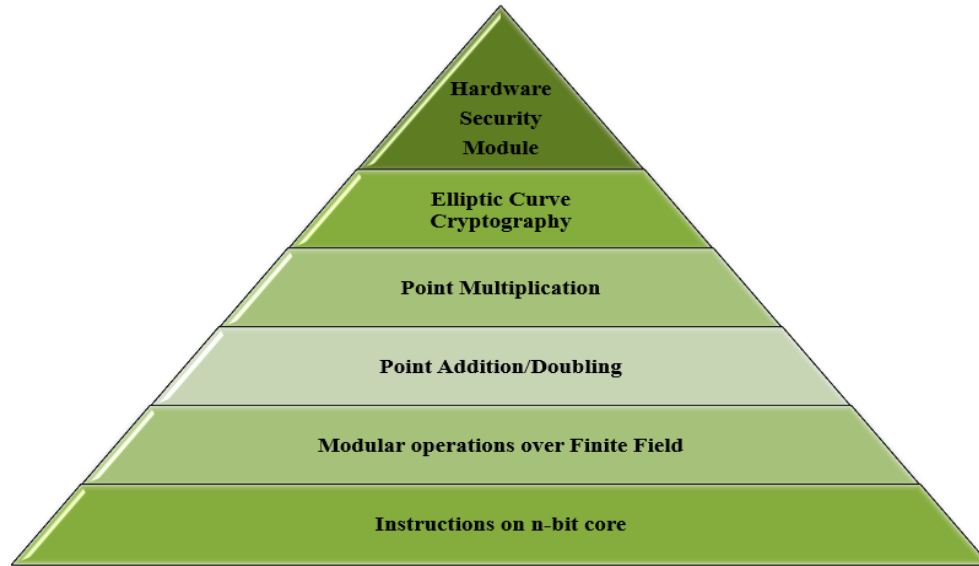


Figure 3. Functional pyramid of proposed HSM (modified from [19])

#### 2.4. Advanced explicitly parallel instruction computing

AEPIC is the operation of improving the performance of an HSM through configuring its instruction set to suit a specific elliptic curve arithmetic algorithm [20], [21]. In particular, it highlights efforts to refine the algorithm across three processor cores. Each core executes FF operations using a customized instruction set specifically designed for arithmetic tasks, such as addition, squaring, and multiplication [22]. This customization consolidates selected operations into a single clock cycle, improving the algorithm's efficiency. Establishing interconnections between cores plays a critical role in managing data dependencies [23]. For instance, if core 1 relies on data generated by core 2 at a specific step, a connection between the two cores ensures seamless data flow, as illustrated in Figure 4. Similar interconnections are established to facilitate other data dependencies within the algorithm. In this research work, the multicore AEPIC design is implemented across three cores, as detailed in pseudo-code.

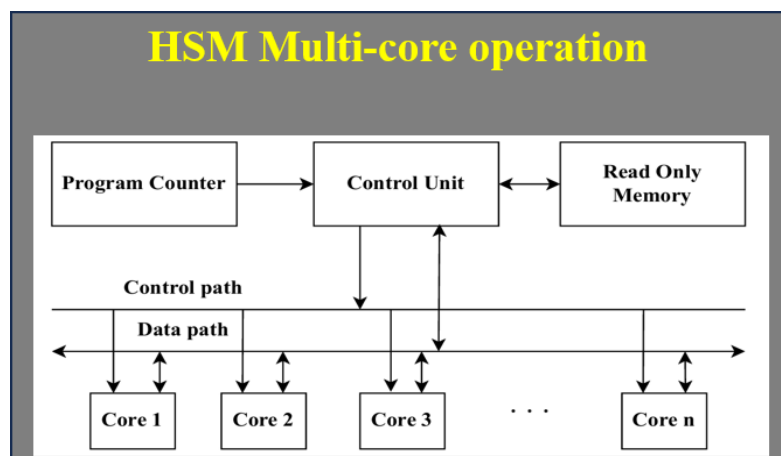


Figure 4. The design of multi-core using AEPIC computing (modified from [23])

## 2.5. Pseudo code overview for multi-core AEPIC design

The following steps outline the pseudo code of the proposed multi-core AEPIC design for efficient elliptic curve point multiplication.

Step 1. Objective of efficient computation of elliptic curve point multiplication  $bA$ , where  $b$  is a binary integer.

Step 2. Convert input point  $(p, q)$  from affine to projective coordinates for efficient computation.

Step 3. Core 1 initializes  $P1, R1$ ; Core 2 is assigned with no operation (idle during initialization). Core 3 computes  $R2 = (p + 0)^2$  for the iterative loop.

Step 4. Core 1 handles point addition  $P1R2$ , squares  $R1$ , and updates  $P2$  with modular reductions. Core 2 performs parallel tasks with  $P2R1$  and updates  $P2$ ; Core 3 computes modular multiplications, squarings, and updates  $R2$ .

Step 5. Core 1 computes modular inversions and multiplications for  $R1$  and  $W1$ , Core 2 performs similar operations for  $R2$  and  $W2$ , and Core 3 handles modular inversions and arithmetic for  $p$  and  $q$ .

Step 6. Some cores remain idle (indicated as "NO-OP") during specific stages to avoid resource conflicts, ensuring efficient pipeline usage.

---

### Pseudo Code for AEPIC [20], [21]

---

Input:  $A = (p, q) \in E(GF(2^n))$ , an  $i$ -bit integer  $B, b \leftarrow (b_{i-1}; \dots; b_1; b_0)_2$ .

Output:  $bA = (p_0; q_0)$ .

// Initialization of Affine Coordinates to Projective Coordinates

// core 1	core 2	core 3
1. $P1 \leftarrow (p + 0)$ ;	NO-OP;	$R2 \leftarrow (p + 0)^2$ ;
2. $R1 \leftarrow (1 + 0)$ ;	NO-OP;	$P2 \leftarrow p^4$ ;
3. NO-OP;	NO-OP;	$P2 \leftarrow P2 + c$ ;

// Point Multiplication Iterative Process

for  $k = i - 2$  down to 0 do

// core 1	core 2	core 3
1. $W1 \leftarrow P1 R2$ ;	$W2 \leftarrow P2 R1$ ;	$W3 \leftarrow P1 R1$ ;
		$X3 \leftarrow R1^4$ ;
2. $R2 \leftarrow (W1 + W2)^2$ ;	NO-OP;	$R1 \leftarrow (W3 + 0)^2$ ;
3. $W1 \leftarrow W1 W2$ ;	$W2 \leftarrow p R2$ ;	$W3 \leftarrow c X3$ ;
		$X3 \leftarrow p^4$ ;
4. $P2 \leftarrow W1 + W2$ ;	NO-OP;	$P1 \leftarrow W3 + X3$ ;

if  $(k \neq 0 \text{ and } b_i \neq b_{i-1})$  or  $(k = 0 \text{ and } b_k = 1)$  then

Swap( $P1, P2$ ), Swap( $R1, R2$ )

end if

end for

// Conversion from Projective to Affine Coordinates

// core 1	core 2	core 3
1. $W1 \leftarrow Inv(R1)$ ;	$W2 \leftarrow Inv(R2)$ ;	$W3 \leftarrow Inv(p)$ ;
2. $X1 \leftarrow P1 W1$ ;	$W2 \leftarrow P2 W2$ ;	$X3 \leftarrow (p + 0)^2$ ;
		$X3 \leftarrow X3 + q$ ;
3. $W1 \leftarrow p + X1$ ;	$W2 \leftarrow p + w2$ ;	NO-OP;
4. $W1 \leftarrow W1 W3$ ;	$W2 \leftarrow W2 W1$ ;	NO-OP;
5. NO-OP;	$W2 \leftarrow W2 + R3$ ;	NO-OP;
6. NO-OP;	$W2 \leftarrow W1 W2$ ;	NO-OP;
7. NO-OP;	$X2 \leftarrow W2 + q$ ;	NO-OP;

return  $bA = (p_0; q_0) = (X1, X2)$ .

---

## 3. RESULTS AND DISCUSSION

This work is implemented through Vivado HLS 2018.3. The results include the detailed design and the execution of the same on different FPGAs. An arithmetic logic unit (ALU) includes instructions from the program counter, data from read-only memory, and control signals from the control unit to perform the computations.

### 3.1. Arithmetic logic unit module with three-core parallelism

An ALU module in Figure 5 that performs three core operations on two input data (IA and IB), including point addition/doubling, multiplication, and repeated squaring. The corresponding output ports

(BP\_O1 and BP\_O2) are given the output signals from the point adder/doubling module and the multiplier module. The input wire of the repeated square field module is designated for the input signal IB.

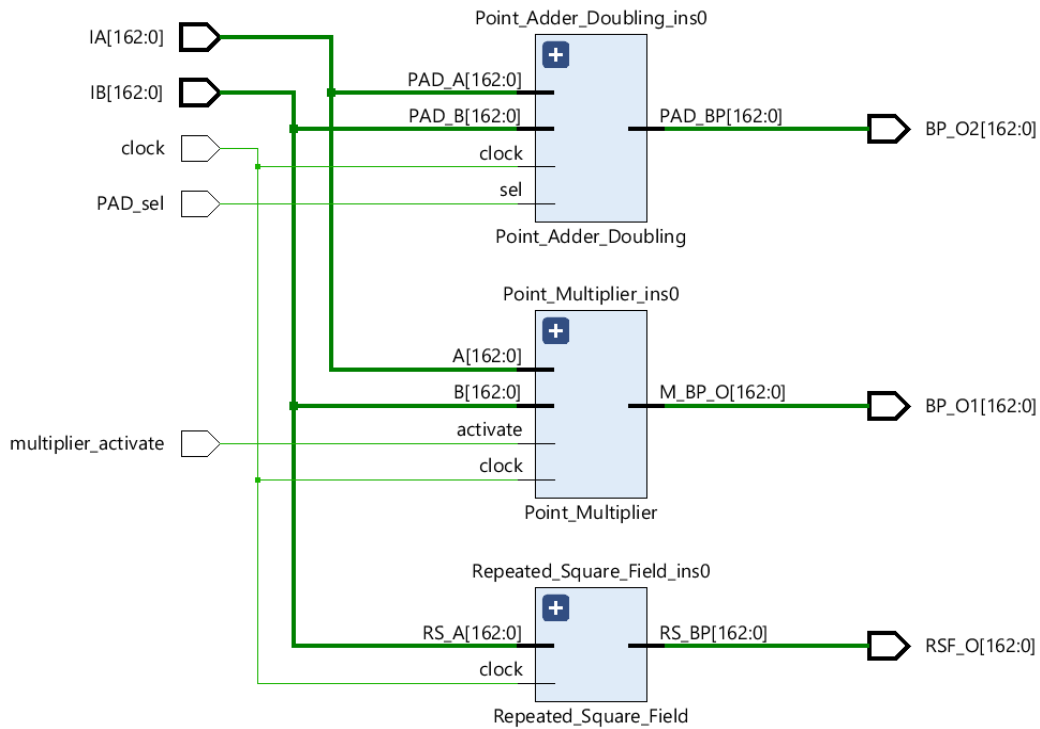


Figure 5. Design of three cores by ALU in Vivado HLS

Based on the control signals (multiplier\_activate and PAD\_sel), the ALU executes the required operations and generates the corresponding results. The output port RSF\_O is designated for the repeated square field module's output signal.

### 3.2. Point addition/doubling module computation

Based on Figure 6, the point addition and doubling operations are handled by a module called 'point\_adder' defined in the design. Four inputs are required by the module: clock, sel, PAD\_A, and PAD\_B. A clock signal is used to enable synchronous operation. The input, which has a value of 0 for addition and 1 for doubling, is a selection signal that chooses the operation to be carried out. The input operands, PAD\_A and PAD\_B, are both 163-bit vectors. PAD\_BP is the module's sole output and represents the outcome of the operation. This value is stored in a 163-bit vector. Registers and signals are used by the module internally to carry out the necessary computations.

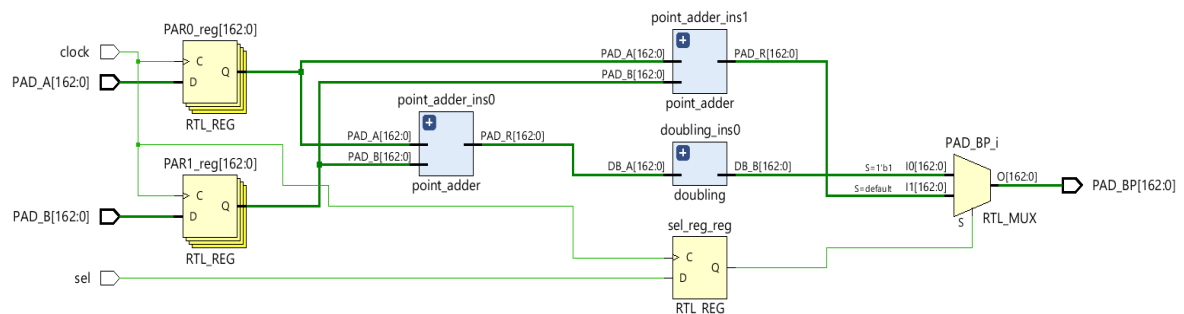


Figure 6. Comprehensive design of ECC point addition/doubling module using Vivado HLS

The values are kept in two registers, PAR0 and PAR1, which are part of it. The module performs the addition operation using two instances of the adder module. These instances generate the addition results using the values from PAR0 and PAR1 as inputs. To complete the addition/doubling process, the module also contains an instance of the doubling module. It generates the doubling/addition result. The value of sel\_reg is used to calculate the output, PAD\_BP. If sel\_reg is set to 1, which denotes doubling/addition, however, if sel\_reg is 0, it signifies only addition. In inference, the 'point\_adder' module gives users the option to execute doubling/addition operations depending on the value of the 'sel' signal.

On each positive edge of the clock signal, the values of PAD\_A and PAD\_B are correspondingly placed in the PAR0 and PAR1 registers. The value of the 'sel' signal is recorded in the sel\_reg register. The adder module creates two instances, point\_adder\_ins0 and point\_adder\_ins1. Using the data from PAR0 and PAR1, point\_adder\_ins0 performs addition, and the outcome is saved in PAD\_ins0\_R. Addition is carried out by point\_adder\_ins1 using the identical values from PAR0 and PAR1, and the outcome is saved in PAD\_ins1\_R. Doubling\_ins0 is created as an instance of the doubling module, which uses the value from PAD\_ins0\_R to perform doubling addition and stores the outcome in DB\_ins0\_R. The value of sel\_reg determines the final bypass output, PAD\_BP: The value of DB\_ins0\_R is assigned to the output PAD\_BP when sel\_reg is 1 (meaning doubling/addition). The value of PAD\_ins1\_R is allocated to the output PAD\_BP when sel\_reg is set to 0 (meaning addition).

### 3.3. Point multiplier module computation

The detailed design of the ECC multiplier module for point multiplication using Vivado HLS is shown in Figure 7. The module accepts the four inputs clock, activate, A, and B. The clock signal is used in the synchronous operation to keep the internal activities of the module. The computation process is managed by the activated input, which regulates when the multiplication operation should start. An operand for multiplication is represented by operands A, B as a 163-bit input vector. These inputs supply the information required for the module to carry out the intended multiplication operation.

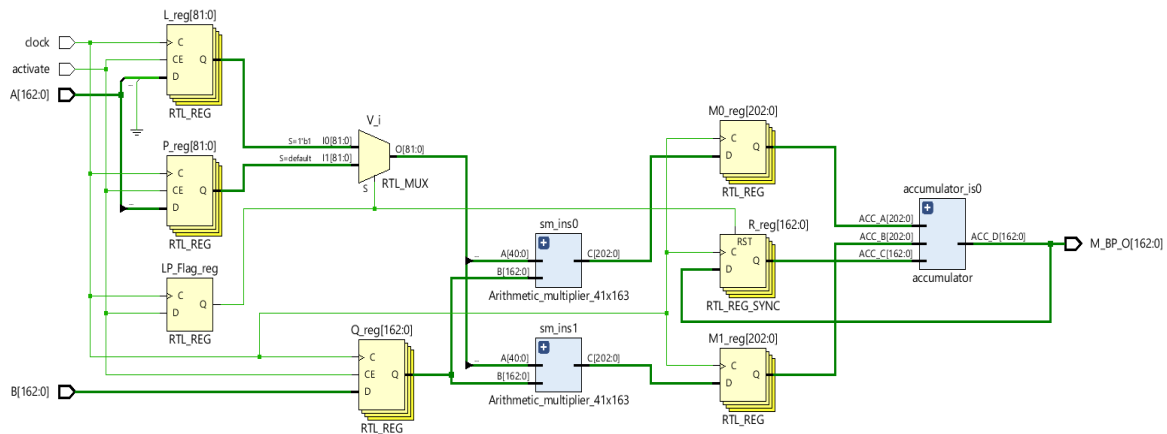


Figure 7. Detailed design of ECC multiplier module using Vivado HLS

The output, M\_BP\_O, represents the result of the multiplication operation. The finished product is stored in a 163-bit vector. Three intermediate signals, designated sm\_ins0\_A, sm\_ins0\_B, and sm\_ins0\_C, are present in the first arithmetic multiplier. The module uses these signals to carry out the calculations required for the first multiplier. Similar to the first multiplier, arithmetic multiplier number two includes three intermediate signals sm\_ins1\_A, sm\_ins1\_B, and sm\_ins1\_C.

Registers L, P, Q, M0, M1, and R store specialized values needed for the multiplication operation. When necessary, the signal 'clear' is used to clear the register R. The LP\_Flag is a flag that represents the computation's state. The module also utilizes an intermediary signal to select between L and P based on the LP\_Flag. Based on the current computation state, it chooses which value will be applied in subsequent computations. An accumulator is employed with an intermediary signal to store the final product.

### 3.4. Repeated square module computation

The ECC repeated squaring module for field arithmetic operation has a 163-bit input signal (RS\_A) that is fed into the module, which outputs a 163-bit signal (RS\_B). Based on the values of the input signal RSM\_A, the assign statements in the Verilog code are used to assign values to each bit of the output signal

RSM\_B. Bitwise XOR operations and various combinations of the input bits are used to assign a value to each bit of the RSM\_B structure. RSM\_B[0] is given the XOR result of RSM\_A[80], RSM\_A[158], and RSM\_A[0] in the first assign statement. Similar to this, later assignment statements determine the values of additional RSM\_B bits depending on various combinations of input bits, which is shown in Figure 8.

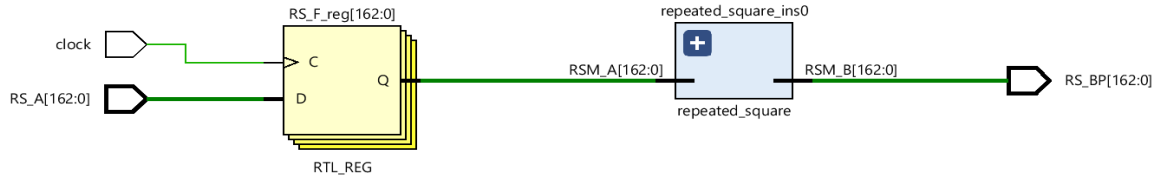


Figure 8. Detailed implementation of ECC repeated squaring module using Vivado HLS

### 3.5. Implementation on heterogeneous field programmable gate array compute fabric

A heterogeneous FPGA compute fabric leverages the strengths of various processing elements to create versatile and high-performance computing for big data analytic environment which is illustrated in Table 1 for 100 Mhz clock frequency.

Table 1. Comparative analysis of power and hardware consumption across various FPGA boards

FPGA board	Part	Hardware consumption		Total power (watts)
		LUTs	Flip flops	
Alpha-Data ADM-PCIE-7V3	xc7vx690tffg1157-2	25,391	7,048	0.626
Kintex Ultrascale Alphadataboard	xcku060-ffva1156-2-e	25,407	7,048	0.962
Artix-7 AC 701	xc7a200tffg676-2	Not feasible		
Alveo U200	xcu200-fsgd2104-2-e	25,412	7,048	2.798
Alveo U250	xcu250-figd2104-2L-e	25,414	7,048	3.625
Kintex-7 KC705	xc7k325tffg900-2	25,399	7,048	0.448
Kintex Ultrascale KCU105	xcku040-ffva1156-2-e	25,420	7,048	0.797
Kintex Ultrascale KCU116	xcku5p-ffvb676-2-e	Not feasible		
Kintex Ultrascale KCU1500	xcku115-flvb2104-2-e	25,404	7,048	1.584
Pico Computing M505	xc7k325tffg900-2	25,399	7,048	0.488
Virtex-7 VC707	xc7vx485tffg1761-2	25,398	7,048	0.542
Virtex-7 VC709	xc7vx690tffg1761-2	25,391	7,048	0.634
Virtex-Ultrascale VCU108	xcvu095-ffva2104-2-e	25,413	7,048	1.247
Virtex-Ultrascale VCU110	xcvu190-flgc2104-2-e	Not feasible		
Virtex Ultrascale+VCU118	xcvu9p-flga2104-2L-e	25,410	7,048	2.780
Virtex Ultrascale+VCU1525	xcvu9p-fsgd2104-2L-e	25,412	7,048	2.789
Zynq-7ZC702	xc7z020clg484-1	Not feasible		
Zynq-7ZC706	xc7z045ffg900-2			
Zynq Ultrascale+ZCU102	xczu9eg-ffvb1156-2-e			
Zynq Ultrascale+ZCU104	xczu7ev-ffvc1156-2-e			
Zynq Ultrascale+ZCU106	xczu7ev-ffvc1156-2-e			
Zynq Ultrascale+ZCU111	xczu28dr-ffvg1517-2-e			

The Alpha, Alveo, Kintex, Pico Computing, and Virtex-7 FPGA families provide a feasible platform for big data analytics for SNMs due to their efficient resource allocation and well-optimized design. In contrast, the Artix-7, Virtex-110 Ultrascale VCU, and Zynq families are less suitable for this application because they are not designed with the necessary optimizations for handling large-scale data analytics in such environments.

### 3.6. Power efficiency analysis for feasible and effective environments

Power efficiency analysis for static and dynamic conditions on feasible FPGA boards, including Alpha, Alveo, Kintex, Pico Computing, and Virtex-7 families, is shown in Figures 9(a)-(e). Static power, also known as leakage power, is the power consumed by the FPGA when it is powered on but not actively switching [24], [25]. It is primarily due to leakage currents that flow even when transistors are not switching states. Static power can be a significant component of the total power consumption, especially as process technology scales down to smaller geometries.

Conversely, dynamic power, also known as switching power, is the power consumed when the FPGA is actively switching states during operation [26], [27]. It is associated with the charging and



discharging of capacitive loads within the FPGA's circuitry. Dynamic power is typically the dominant component of power consumption during active operation. According to power reports, the Virtex-7 family FPGA board exhibits a favorable dynamic and static power consumption, accounting for 55% of the total power. The on-chip power consumption is 0.542 watts, with a junction temperature of 25.6 °C and a thermal margin of 59.4 °C. These characteristics make the Virtex-7 an excellent platform for big data analytics for SNM.

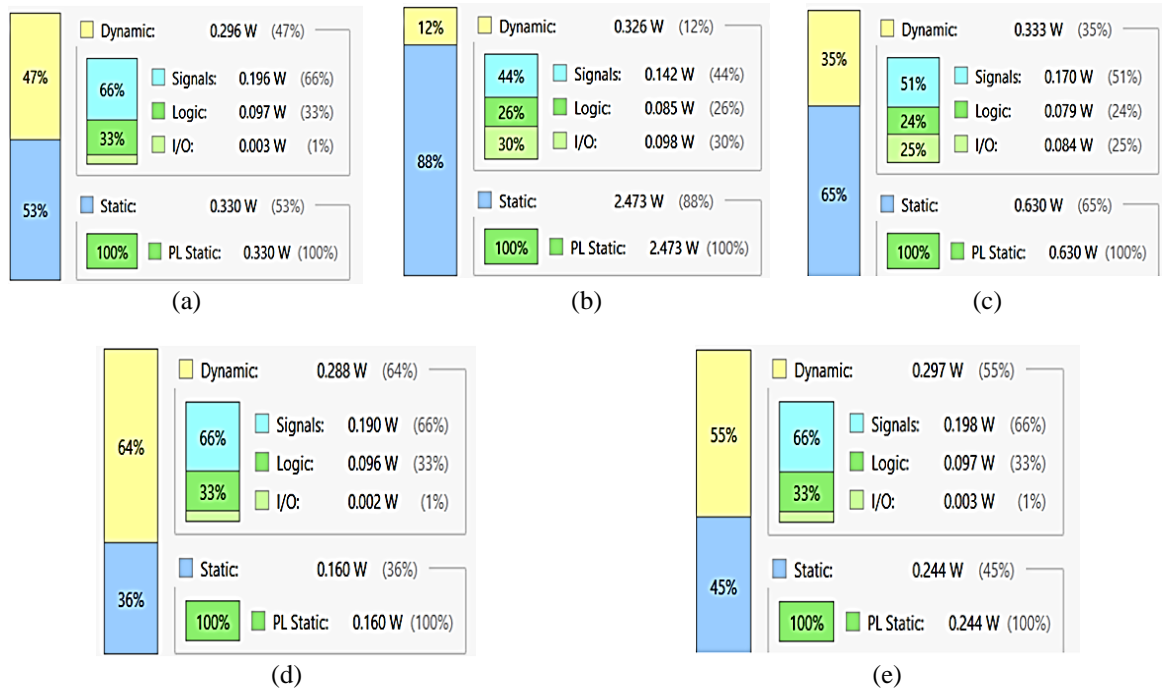


Figure 9. Power report of FPGA; (a) Alpha, (b) Alveo, (c) Kintex, (d) Pico, and (e) Virtex

### 3.7. Assessing data rate

In the context of big data analytics, where large volumes of data from multiple meters are aggregated and analyzed, accurate data rate calculations are even more critical. They ensure that:

- Data aggregation is seamless and timely.
- Analytical models receive up-to-date and continuous data streams for accurate forecasting and analysis.
- System performance remains robust under high data throughput conditions.

Overall, data rate calculation [28] for SNM is fundamental to ensuring efficient, reliable, and scalable operation, which is vital for modern smart grid infrastructures.

$$\text{Throughput per Core} = \text{Clock Frequency} \times \text{Operations per Clock Cycle} \quad (1)$$

$$\text{Total Throughput} = \text{Throughput per Core} \times \text{Number of Cores} \quad (2)$$

From (1) and (2) as in [29], [30], for each bit size (163, 233, 283, 409, and 571 bits) with a 100 MHz clock and assuming one operation per clock cycle, the theoretical throughput for your 3-core design is 300 mega operations per second (MOPS). Therefore, the data rate calculation is depicted in Table 2.

Table 2. Data rate calculation for different curve

Koblitz curve (bits)	Data rate=throughput×bits per operation (Gbps)	FPGA board
163	48.9	Virtex-7 family board
233	69.9	
283	84.9	
409	122.7	
571	171.3	

### 3.8. Comparison of results

Overall, the comparison between Arduino Uno and FPGA computing for SNM is important for making informed decisions about performance, scalability, power consumption, cost, flexibility, real-time processing, reliability, and ease of development, ensuring the selection of the best platform for specific needs and constraints. Table 3 shows the performance comparison for the computing capability. The comparison indicates that the Virtex-7 is the best choice for a big data analytics environment for SNM applications across a wide range of use cases.

Table 3. Computing capability of Arduino Uno vs. FPGA

Arduino Uno [1]	FPGA for SNM (proposed)
Security feature: not available	Security feature: AEPIC multi-core HSM
Clock frequency: 16 MHz	Clock frequency: 100 MHz
On-chip power consumption is around 0.3 watts under typical conditions (5 V and 60 mA)	On-chip power consumption 0.542 watts
Data rate: 0.3 kbps to 27 kbps	Data rate: 48.9 kbps to 171.3 kbps at Virtex-7

## 4. CONCLUSION

This work demonstrated the successful integration of innovative technologies to support SDG 9 through the development of an HSM for an SNM application within CPS. By employing an AEPIC approach and utilizing a multi-core HSM based on ECC, the proposed solution effectively meets the demands of large-scale data analytics. The use of Alpha, Alveo, Kintex, Pico, and Virtex FPGAs ensures adaptability and scalability, enabling optimized performance regarding big data aggregation for SNM applications. Among that, Virtex-7 has the best aptness for a 100 MHz clock frequency and 48.9 kbps to 171.3 kbps data rate. Future work will focus on validating the HSM against various side-channel and cybersecurity attacks to ensure robust protection. Additionally, addressing the real-world impact of the algorithm includes potential energy savings and enhancements in security.

## FUNDING INFORMATION

Authors state no funding involved.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
B. Muthu Nisha	✓	✓	✓		✓	✓	✓	✓	✓	✓			✓	
J. Selvakumar		✓		✓		✓		✓		✓	✓	✓	✓	

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.




## DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.




## REFERENCES

- [1] V. Okulich-Kazarin, "Statistics Using Neural Networks in the Context of Sustainable Development Goal 9.5," *Sustainability*, vol. 16, no. 19, pp. 1-17, 2024, doi: 10.3390/su16198395.
- [2] H. Ajra, M. A. Majid, and M. S. Islam, "IoT-enabled smart cities towards green energy systems: a review," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 13, no. 3, pp. 708–723, 2024, doi: 10.11591/ijres.v13.i3.pp708-723.
- [3] R. Raman, A. Iyer, and P. Nedungadi, "Forecasting artificial general intelligence for sustainable development goals: a data-driven analysis of research trends," *Sustainability*, vol. 17, no. 16, pp. 1-29, 2025, doi: 10.3390/su17167347.
- [4] A. Y. Chaudhari, P. Mulay, and S. Chavan, "The role of smart electricity meter data analysis in driving sustainable development," *MethodsX*, vol. 14, pp. 1-10, 2025, doi: 10.1016/j.mex.2025.103196.
- [5] M. G. M. Almihat, M. T. E. Kahn, K. Aboalez, and A. M. Almaktoof, "Energy and sustainable development in smart cities: an overview," *Smart Cities (MDPI)*, vol. 5, no. 4, pp. 1389–1408, 2022, doi: 10.3390/smartcities5040071.
- [6] A. Rizwan, R. Rasheed, H. Javed, Q. Farid, and S. R. Ahmad, "Environmental sustainability and life cycle cost analysis of smart versus conventional energy meters in developing countries," *Sustainable Materials and Technologie*, vol. 33, p. e00464, 2022, doi: 10.1016/j.susmat.2022.e00464.
- [7] P. Ezhilarasi, L. Ramesh, P. Sanjeevikumar, and B. Khan, "A cost-effective smart metering approach towards affordable deployment strategy," *Scientific Reports*, vol. 13, no. 1, Art. no. 19452, Nov. 9, 2023, doi: 10.1038/s41598-023-44149-9.
- [8] I. Laassar and M. Y. Hadi, "Intrusion detection systems for internet of thing based big data: a review," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 12, no. 1, pp. 87–96, 2023, doi: 10.11591/ijres.v12.i1.pp87-96.
- [9] M. A. S. Mohammad and M. R. Pradhan, "Machine learning with big data analytics for cloud security," *Computers & Electrical Engineering*, vol. 96, part A, Dec. 1, 2021, doi: 10.1016/j.compeleceng.2021.107527.
- [10] J. W. Bos, C. Costello, P. Longa, and M. Naehrig, "Selecting elliptic curves for cryptography: an efficiency and security analysis," *Journal of Cryptographic Engineering*, vol. 6, pp. 259–286, 2016, doi: 10.1007/s13389-015-0097-y.
- [11] S. Di Matteo, L. Baldanzi, L. Crocetti, P. Nannipieri, L. Fanucci, and S. Saponara, "Secure elliptic curve crypto-processor for real-time IoT applications," *Energies*, vol. 14, no. 15, Art. no. 4676, Aug. 1, 2021, doi: 10.3390/en14154676.
- [12] M. S. Sami *et al.*, "Advancing trustworthiness in system-in-package: A novel root-of-trust hardware security module for heterogeneous integration," *IEEE Access*, vol. 12, pp. 48081–48107, Mar. 14, 2024, doi: 10.1109/ACCESS.2024.3375874.
- [13] Y. Xie *et al.*, "Security-related hardware cost optimization for CAN FD-based automotive cyber-physical systems," *Sensors*, vol. 21, no. 20, pp. 1-17, Oct. 2021, doi: 10.3390/s21206807.
- [14] A. J. Cabrera-Gutiérrez *et al.*, "Integration of Hardware Security Modules and Permissioned Blockchain in Industrial IoT Networks," *IEEE Access*, vol. 10, pp. 114331–114345, 2022, doi: 10.1109/ACCESS.2022.3217815.
- [15] M. H. Murtaza *et al.*, "A portable hardware security module and cryptographic key generator," *Journal of Information Security and Applications*, vol. 70, Nov. 2022, doi: 10.1016/j.jisa.2022.103332.
- [16] B. Lin *et al.*, "A Unified Memory and Hardware Security Module Based on the Adjustable Switching Window of Resistive Memory," *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 1257–1265, 2020, doi: 10.1109/JEDS.2020.3019266.
- [17] R. Höglund and M. Tiloca, "Current state of the art in smart metering security," *SICS Swedish ICT AB, Security Lab, Technical Report T2015:03*, Kista, Sweden, Apr. 2015.
- [18] M. A. Mehrabi, C. Doche, and A. Jolfaei, "Elliptic curve cryptography point multiplication core for hardware security module," *IEEE Transactions on Computers*, vol. 69, no. 11, pp. 1707–1718, Nov. 2020, doi: 10.1109/TC.2020.3013266.
- [19] N. P. Kumar and C. Shirisha, "An area-efficient ECC architecture over GF(2m) for resource-constrained applications," *AEU - International Journal of Electronics and Communications*, vol. 125, p. 153383, 2020, doi: 10.1016/j.aeu.2020.153383.
- [20] Y. Zhang, D. Chen, Y. Choi, L. Chen, and S.-B. Ko, "A high performance ECC hardware implementation with instruction-level parallelism over GF(2<sup>163</sup>)," *Microprocessors and Microsystems*, vol. 34, no. 6, pp. 228–236, 2010, doi: 10.1016/j.micpro.2010.04.006.
- [21] Y. Zhang *et al.*, "A high performance pseudo-multi-core ECC processor over GF(2<sup>163</sup>)," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2010)*, Paris, France, 2010, pp. 701–704, doi: 10.1109/ISCAS.2010.5537251.
- [22] L. Li and S. Li, "High-Performance Pipelined Architecture of Elliptic Curve Scalar Multiplication Over GF(2m)," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 4, pp. 1223–1232, Apr. 2016, doi: 10.1109/TVLSI.2015.2453360.
- [23] J. Fan, K. Sakiyama, and I. Verbauwhede, "Elliptic curve cryptography on embedded multicore systems," *Design Automation for Embedded Systems*, vol. 12, no. 3, pp. 231–242, Sep. 2008, doi: 10.1007/s10617-008-9022-6.
- [24] N. S. Kim *et al.*, "Leakage current: Moore's law meets static power," *IEEE Computer*, vol. 36, no. 12, pp. 68–75, 2003, doi: 10.1109/MC.2003.1250885.
- [25] K. S. Khouri and N. K. Jha, "Leakage power analysis and reduction during behavioral synthesis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 6, pp. 876–885, Dec. 2002, doi: 10.1109/TVLSI.2002.808436.
- [26] H. F. Blanchette, T. Ould-Bachir, and J. P. David, "A State-Space Modeling Approach for the FPGA-Based Real-Time Simulation of High Switching Frequency Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4555–4567, Dec. 2012, doi: 10.1109/TIE.2011.2182021.
- [27] L. Idkhajine and E. Monmasson, "Embedded Fully FPGA-Based Real-Time Simulators for Static Power Converters With Power Switch Characteristics Approximated by Identification," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 9, pp. 9624–9633, Sept. 2022, doi: 10.1109/TIE.2021.3112999.
- [28] J. Constantin *et al.*, "An FPGA-based 4 Mbps secret key distillation engine for quantum key distribution systems," *Journal of Signal Processing Systems*, vol. 86, no. 1, pp. 1–15, 2017, doi: 10.1007/s11265-015-1086-1.
- [29] V. S. Sathe *et al.*, "Resonant-Clock Design for a Power-Efficient, High-Volume x86-64 Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 140–149, Jan. 2013, doi: 10.1109/JSSC.2012.2218068.
- [30] S. Chaudhry, P. Caprioli, S. Yip, and M. Tremblay, "High-performance throughput computing," *IEEE Micro*, vol. 25, no. 3, pp. 32–45, May–Jun. 2005, doi: 10.1109/MM.2005.49.

**BIOGRAPHIES OF AUTHORS**

**B. Muthu Nisha**    pursuing her Ph.D. degree from the Department of Electronics and Communication Engineering, SRMIST, Kattankulathur. She has worked her Post Graduate project at National Small Industries Corporation (NSIC) in the research area of Embedded systems. Her research interests include hardware security, artificial intelligence, and cryptography. She can be contacted at email: mb1850@srmist.edu.in.



**J. Selvakumar**    received the Ph.D. degree in 2013 from the Department of Electronics and Communication Engineering, SRMIST (formerly known as SRM University) Kattankulathur. Currently, he is a professor, with the Electronics and Communication Engineering Department, SRMIST, Kattankulathur. His research interests include low power VLSI design, reconfigurable VLSI architecture design, and VLSI cyber security. He can be contacted at email: selvakuj@srmist.edu.in.