TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 5, October 2021, pp. 1636~1647
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i5.19416

Efficient hardware prototype of ECDSA modules for blockchain applications

Devika K. N., Ramesh Bhakthavatchalu
Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India

Article Info

Article history:

Received Jan 08, 2021
Revised Jul 16, 2021
Accepted Jul 31,2021

Keywords:

Blockchain
ECDSA
FPGA
Hardware security

ABSTRACT

This paper concentrates on the hardware implementation of efficient and re-configurable elliptic curve digital signature algorithm (ECDSA) that is suitable for verifying transactions in Blockchain related applications. Despite ECDSA architecture being computationally expensive, the usage of a dedicated stand-alone circuit enables speedy execution of arithmetic operations. The prototype put forth supports N-bit elliptic curve cryptography (ECC) group operations, signature generation and verification over a prime field for any elliptic curve. The research proposes new hardware framework for modular multiplication and modular multiplicative inverse which is adopted for group operations involved in ECDSA. Every hardware design offered are simulated using modelsim register transfer logic (RTL) simulator. Field programmable gate array (FPGA) implementation of var- ious modules within ECDSA circuit is compared with equivalent existing techniques that is both hardware and software based to highlight the superiority of the suggested work. The results showcased prove that the designs implemented are both area and speed efficient with faster execution and less resource utilization while maintaining the same level of security. The suggested ECDSA structure could replace the software equivalent of digital signatures in hardware blockchain to thwart software attacks and to provide better data protection.

Corresponding Author:

Devika K. N.
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
Email: devikanandalal@gmail.com

1. INTRODUCTION

Blockchain that has been foreseen among researchers as one of the promising and trending technology for humanitarian benefits. It have explored its potential in diverse sectors namely healthcare, supply chain, logistics, internet of things (IoTs), Academic Fields and many more in addition to financial domain [1]-[5]. This distributed ledger framework gains its prevalence owing to the transparency, immutability and data integrity that it offers to each transaction involved within the network [6]. Each node verifies the transaction after it gets signed through digital signatures. Legal transactions after verification are stored to form blocks in the blockchain. Since every proceeding are validated through signature process a stand-alone hardware logic design to do the procedure could speed up the entire blockchain system.

Elliptic curve digital signature algorithm (ECDSA) is the most commonly used validation method among all other signature techniques to verify transactions in blockchain. The popularity of ECDSA stems from its powerful arithmetic framework and strong security in comparison to other digital signing schemes.

RSA algorithm and ECDSA forms the two most popular public key cryptosystems presently utilized for cryptographic operations. RSA introduced by Rivest, Shamir and Adleman have its enciphering strength on Integer factorization and on the choice of key-size. Elliptic curve cryptography (ECC) have an upper hand in imparting equivalent level of security as compared to RSA with reasonable data processing cost and smaller key-size. For instance, RSA encryption for 1024-bit and 160-bit sized ECC encryption provides the same level of security [7].

ECC is a robust technique that evolved into an important research domain in public key cryptography [8], [9]. Mathematical operations within elliptic curves are utilized to generate key pairs for ECDSA. Realizing fast arithmetic units, ECC point operations such as modular multiplication, point addition and point doubling can enhance the efficiency of digital signature algorithm over elliptic curves. Since ECC take advantage of smaller parameters in contrast to discrete logarithms and other non-EC cryptography faster computations and shorter key size are guaranteed. Computational insolvability of discrete logarithm problem in elliptic curves forms the mathematical foundation for its unbreakable cryptographic structure. ECC have become the favorite choice for a broad spectrum of applications ranging from the sensors to blockchain due to its strong security with small key size and low power dissipation [10].

ECDSA architecture can be prototyped either in software or hardware domain. This paper solely targets on the hardware implementation of digital signature structure in ECC since hardware design enables speedy execution when compared to its software counterpart. Nevertheless, hardware model of ECDSA with low resource utilization and low latency is quite a demanding task considering the fact that both parameters are contradictory. Thus area-latency trade-off is essential to achieve greater efficiency. The main focus of this research work is to design faster and area efficient cryptographic modules for ECDSA framework. This hardware paradigm can be integrated with hardware blockchain to overcome the security issues brought in by a software based digital signature.

2. LITERATURE REVIEW

The work done in [11] deliberates about the role of elliptic curve digital signature algorithm in blockchain transactions. Data provenance is ensured through this signature scheme and the parties involved stand in need to deal with matter of trust. Drug counterfeiting, its effects on various domains and how blockchain techniques along with ECDSA could be applied to curb drug fabrication is detailed in [12]. Flaws in supply chain is the fundamental cause for drug counterfeiting. This work explains the benefits of blockchain to track, detect and safeguard supply chain of drugs. The public ledger technology records the drug constituents along with its unique serial numbers from producers to consumers. Since entire transactions are digitally signed and amended in the distributed ledger even a slight modification in any phase and fraudulent individuals could be recognized. Nyame *et al.* [13] proposed adopts ECDSA to establish a viable and effective client verification and approval protocol for managing knowledge transmissions in role-based access control (RBAC). Digital signature algorithm contributes to build a trustwor- thy mechanism to verify users prior to share, exchange and storing information in the knowledge repository. System model proposed make use of blockchain to ensure transparency and immutability in all exchanges performed among entities. Therefore, the action once committed can never be divided as they are interlinked. Knowledge management systems (KMS) is a function-oriented data system in which user generates, share and reserve knowledge to enhance performance of an organization [14], [15].

Authenticity of the information transmitted in car to infrastructure system (CITS) is of supreme im-portance to curb the influence of priority vehicles on traffic flow and traffic lights. In the work demonstrated in [16] develops ECDSA signature scheme with low latency and high throughput with an ability to authenticate about 2700 verification per second. Pipeline architecture that makes use of carry look ahead adder and long word summations is adopted in [17] to handle group operations in ECDSA. FPGA based architecture to perform mathematical operations required by ECC primitives is presented in [18] while the work done in [19] elaborates on ECC encryption and suggest a technique to segment original message into digital block codes and to map it sequentially into specific points in elliptic curve. However, details on the implementation of the proposed system is not provided but the working principle regarding the execution of ECC algorithm is illustrated.

A novel mapping method is adopted in [20] to convert unencrypted text into corresponding American Standard Code for Information Interchange (ASCII) values and then translate into its hexadecimal equivalent. This two-phase conversion contributes to secure message encryption using ECC since cipher input considers hexadecimal value in reverse order. Theoretical aspects of ECC are discussed in [21]. This paper converse about the key principles and design strategies to implement arithmetic operations in ECC. Area reduction in ECC architecture is achieved in [22] by means of Montgomery ladder algorithm. Reduced complexity and shift operation in place of division enhances computational speed and add area compactness to the suggested prototype. Novel architectures of elliptic curve operations on twisted Edwards curve for 256-bit key [23] and hybrid pipeline technique along with Montgomery ladder algorithm is adopted in [24] to achieve high speed

binary field point multiplication. Parallelism in ECC group operations is exploited in [25] to bring out high speed and less complex point multiplication hardware.

2.1. Contributions

In this work, hardware implementation of area and speed efficient reconfigurable ECDSA architecture for verification of blockchain transactions is proposed. Main objective of the research is to optimize computationally intricate operations such as multiplicative inverse, modular multiplication and finite field operations to ensure high performing ECDSA framework for applications that utilizes blockchain technology. This paper proposes a novel methodology to unravel the complexity of finite field operations by implementing simple and highly efficient arithmetic units that forms the building blocks of ECDSA.

Significant contributions put forward by this paper can be outlined as given:

- An efficient prototype for elliptic curve cryptography based signature generation and verification has been proposed to attain better performance in terms of speed and area overhead.
- Novel architecture for modular multiplication using counter mechanism is adopted to optimize resource utilization and operational frequency in ECC finite field operations.
- New hardware design is introduced for modular inversion to alleviate the computational complexity of multiplicative inverse algorithm.
- Every arithmetic module including modular multiplication, modular inversion, modular squaring and group operations that forms the digital signature circuit is designed to be reconfigurable and flexible enough to handle any bit-size values.
- Additionally, hardware resource consumption and operating frequency of the proposed ECC architectures are better in comparison to related works which assures improved performance of all the designs proposed.

The rest of the paper is formulated as follows: fundamentals of elliptic curve cryptography and the steps involved in ECDSA algorithm is summarized in section 2. Section 3 discusses about the proposed hardware architectures for modular multiplication, modular inversion and group operations in ECC. Implementation results of the designed ECC prototypes are conferred in section 4. Section 5 highlights the comparison results of the proposed designs with existing works. Section 6 concludes the paper along with future scope of the research work.

3. FUNDAMENTALS OF ELLIPTIC CURVE CRYPTOGRAPHY

3.1. Basic concepts of elliptic curve cryptography

The need for strong security and improved bandwidth in the area of communication urged two scientists named Victor Miller and Koblitz to introduce elliptic curves into the world of cryptography in 1985. The objective of ECC is to increase the mathematical complexity of Discrete Logarithm problem through its translation to elliptic curves. Besides encryption and decryption ECC also realizes key exchange schemes and digital signatures. In ECC implementation simple encryption algorithm is mapped to reliable elliptic curve. Therefore, the elementary operations such as modular addition and modular multiplication as in RSA turns into point addition and point multiplication respectively in this new form of public key cryptography.

Weierstrass elliptic curve equation for a prime field F_p (p>3) is given by the form:

$$E: y^2 = x^3 + ax + b \tag{1}$$

for constant parameters a and b when the specific condition $4a^2+27b^3=0$ is satisfied [26], [27].

If two points on the curve are represented as A (x_1, y_1) and B (x_2, y_2), then ECC operation known as point addition results in another point C (x_3, y_3) on the curve where C=A+B which can be calculated as:

$$x_3 = m^2 - x_1 - x_2 \tag{2}$$

$$y_3 = m\,(x_1 - x_3) - y_1 \tag{3}$$

where $m = \frac{y_2 - y_1}{x_2 - x_1}$ gives the slope of the line between A and B.

Addition of same points A (x_1, y_1) to itself results in C (x_3, y_3) on the curve through another group operation called point doubling where C=A+A=2A is computed as follows:

$$x_3 = m^2 - 2x_1 \tag{4}$$

$$y_3 = m\,(x_1 - x_3) - y_1 \tag{5}$$

where $m = \frac{3x^2 + a}{2}$.

The above equations demand arithmetic operations such as addition, subtraction and modulo multiplication. Computing the slope of the curve 'm' necessitates hardware prototype for modular multiplicative inverse. Since modulo-p multiplication and inversion are executed in the prime field Fp, latter becomes too costly to afford with respect to performance and resources in comparison with the former.

3.2. Steps for ECDSA signature generation and verification

Sender A uses his private key d_A to sign the message M to be sent to the receiver B. Here the message provided can be of any bitsize. The key values are predefined by the sender prior to transmission.

3.2.1. Steps under signature generation

- Hash of the message M is calculated to get Z. Typically the hash function applied is SHA-1. Z = hash(M)
- Select a random value K within the range [1, n-1].
- Compute co-ordinate (x_1, y_1) = kG where G is the base point.
- Determine r = x_1 mod *n*.
- Compute S = K^{-1} (z + d_Ar) mod *n*.
- If r = 0 and s = 0, go for a new random value 'k'.
- Signature pair generated is (r,s).

After ECDSA signature generation, A transmits signed message to B. B possess A's public key Q_A. B validates the authorship of transaction through verification.

3.2.2. Steps for signature verification

- Certifies r and s lies in the range [1, n-1].
- Compute $W = s^{-1}$ mod *n*.
- Calculate $u_1 = Zw$ mod *n* where Z is the hash of message M.
- Calculate $u_2 = rw$ mod *n*.
- Compute co-ordinates $(x_2, y_2) = u_1G + u_2Q_A$.
- If x2 = r signature is verified and validated else it is rejected or considered invalid.

4. PROPOSED HARDWARE IMPLEMENTATION OF ECDSA MODULES

4.1. Modular multiplication

Modular multiplication constitutes one among the most time intensive mathematical operations in elliptic curve cryptography over a finite field. Adoption of efficient algorithms such as Montgomery, residue number system (RNS) or interleaved modular multiplication can improve the latency and speed of computational process [28]-[34]. However, Montgomery technique needs extra processing operations to constrain values within the specified finite field. RNS multiplication scheme results in significant area overhead and power consumption to implement its hardware counterpart [35], [36]. Hence to reduce the area overhead as well as to improve the performance in terms of speed, modular multiplier using counter mechanism is designed and adopted.

Modular multiplication of two N-bit values A and B over prime field GF(q) is given by the formula:

$$M = AB \bmod q \tag{6}$$

$$= \{b_{n-1}(2^{n-1}A) + b_{n-2}(2^{n-2}A) + \ldots\ldots + b_1(2^1A) + b_0A\} \tag{7}$$

$$= \{\sum_{i=0}^{N-1} b_i\,(2^i\,A) \bmod q\} \tag{8}$$

Figure 1 illustrates the flowchart of proposed modular multiplication circuit using counter mechanism. In this technique, left shift register shifts the value stored in reg P by one bit for every i^{th} bit of multiplier B. Whenever B_i is one accumulator adds the output value of shift register along with multiplicand A and stores the result in reg F. Value stored in reg F after "N" iterations forms the final product of A and B after multiplication. To perform modulo operation the content of reg G is repeatedly subtracted by prime value 'q' until the difference become less than input 'q'. The module make use of three multiplexers where the first mux checks each bit of B for 1 and if the condition is satisfied it enables register F to store the sum of A and accumulator content or else retains the accumulator value. The other two muxes aids to perform modulo operation. In the proposed framework N-bit input requires 'N' clock cycles to complete modular multiplication. Same structure work as squaring circuit when identical inputs are considered as both multiplier and multiplicand.

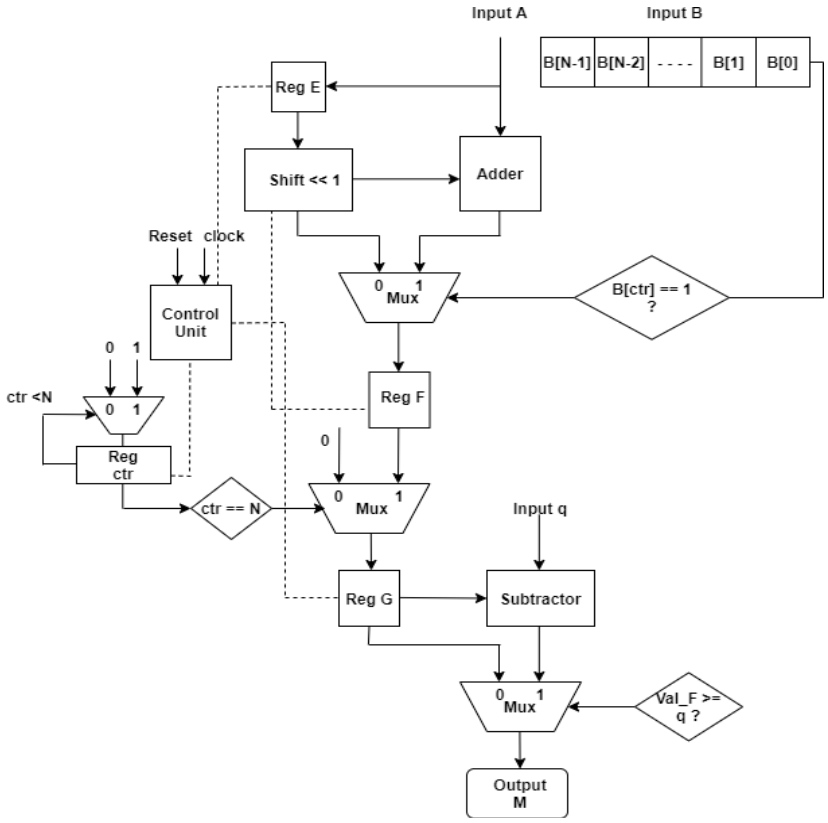

Figure 1. Proposed modular multiplication using counter mechanism

4.2. Modular multiplicative inverse

In number theory multiplicative inverse of an integer A is said to be B if and only if the product of A and B is congruent to 1 in respect of modulus N [37]. As per the conventional annotation, the congruency of inverse function is given by the formula:

$$AB \equiv 1 \bmod N \tag{9}$$

Multiplicative inverse operation constitutes the most resource intensive and time-consuming arithmetic process in ECC. It is a mandatory action undertaken in point addition and point doubling to calculate the slope of the curve. Several algorithms currently exist to aid the mathematical computation of modular inversion [38]-[41]. Nonetheless, the Extended Euclidean method establish the simple and effectual approach to do so.

The steps involved in Extended Euclidean algorithm for inputs A, p and output Z = a^{-1} mod p is summarized below:

Algorithm 1 Extended Euclidean algorithm

Result: A^{-1} mod p
Initialization: $x_1 \leftarrow p$, $x_2 \leftarrow A$, $n_1 \leftarrow 0$, $n_2 \leftarrow 1$
while r ≠ 0
  r = x_1 mod x_2 , q = $x_1 \div x_2$, x = $x_1 - q.x_2$
  If r = 1 then
    Z returns A^{-1} mod p;
  else
    $x_1 \leftarrow x_2$; $x_2 \leftarrow x$;
    $t \leftarrow n_1 - q*n_2$; $n_1 \leftarrow n_2$; $n_2 \leftarrow t$; **end**
  end
  return Z - Multiplicative inverse is Z

Figure 2 details the sequence of arithmetic operations involved in the inversion operation. The pro- posed design employs multiplier using counter method and basic arithmetic circuits to optimize hardware resource utilization. N-bit inputs are initially stored in reg A and reg B respectively. Output of reg B is denoted as val_B while register R delivers val_r. Divider circuit takes in A and B and provide quotient 'q' as the output which is eventually stored in reg q. Similarly, remainder computation during division is executed using modulo circuit to get val_r. Two values within reg 'B' and reg 'r' are compared. If val_B is greater than val_r, content of reg A is replaced with that of reg B and latter is occupied with val_r. Computation of quotient and remainder using above condition continues until val_r is 0. This determines the number of iterations to calculate the inverse of A with respect to B. Reg P is initialized with 1. Previous content within reg P P[ctr] is multiplied with quotient value 'q' for each iteration and subtracted from the previous value stored in P[ctr+1] where 'ctr' indicates the iteration value. The final result is stored in reg Z corresponding to R[ctr] being 1.

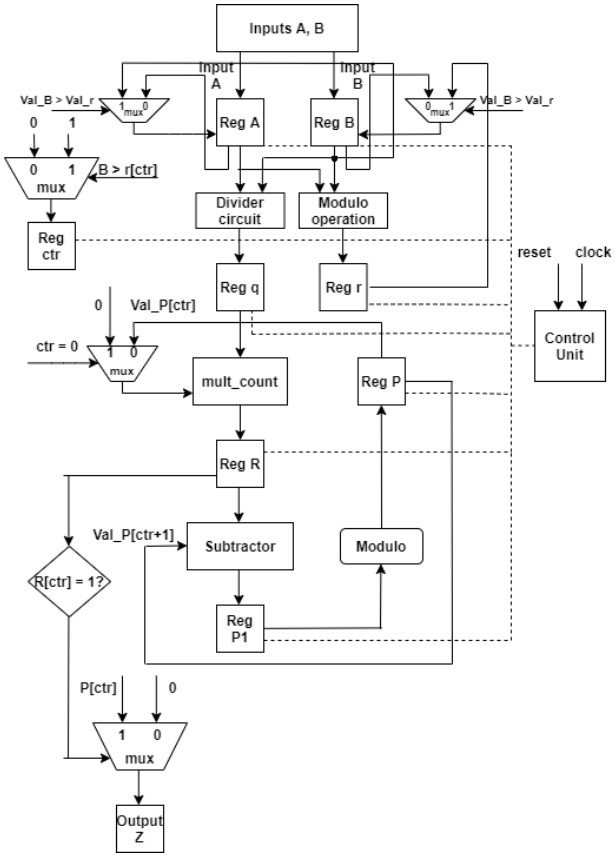

Figure 2. Proposed architecture for multiplicative inverse circuit

4.3. Elliptic curve field operations

Elliptic curve field operations mainly comprise of point addition and point doubling that are per-formed for definite consecutive stages to execute elliptic curve multiplication that forms the critical operation in ECC. If P and Q are two points on a given elliptic curve so that kP=Q, then computation of 'k' is infeasible even if provided with P and Q when k is significantly large. Here 'k' is the scalar value that turns out to be the discrete logarithm of Q for base P. Figures 3 and 4 portrays the series of arithmetic operations involved in proposed architectures of point addition and point doubling respectively.

Proposed architecture for point addition requires 2M+2S+1inversion operations while point doubling needs 3M+1S+1inversion. To achieve optimized performance in terms of area, speed and latency the sub-module structures are efficiently designed and effectively activated at required stages. Multiplication and squaring operations have a latency of N+1 clock cycles.

Point multiplication constitutes a series of field operations as per the binary pattern of scalar 'k' and the proposed structure for ECC multiplication is depicted in Figure 5. Double and add (left to right) algorithm is the simplest method to execute point multiplication wherein point doubling is performed for each bit value of 'k' while point addition functions only when i^{th} bit of 'k' is 1 over a prime field P.

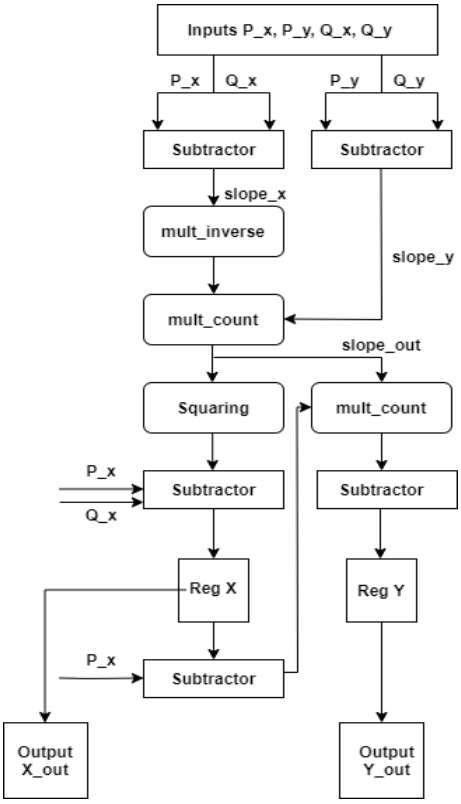

Figure 3. Flowchart of various operations performed in the proposed design of point addition

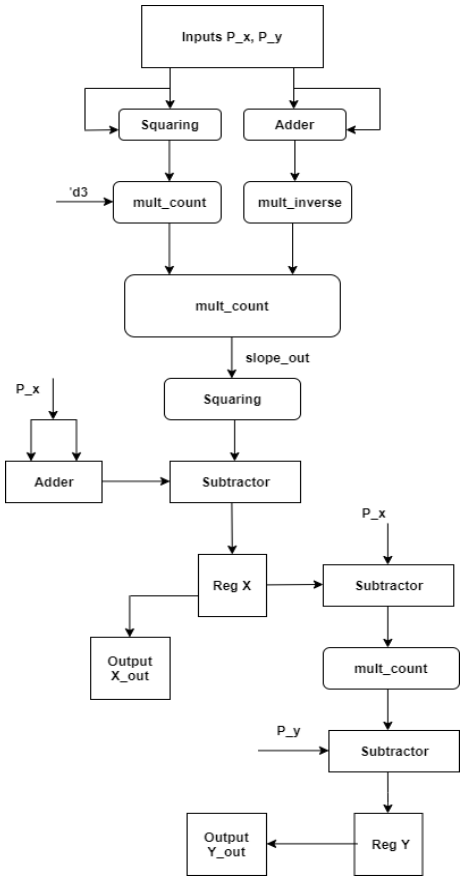

Figure 4. Flowchart of various operations performed in the proposed design of point doubling

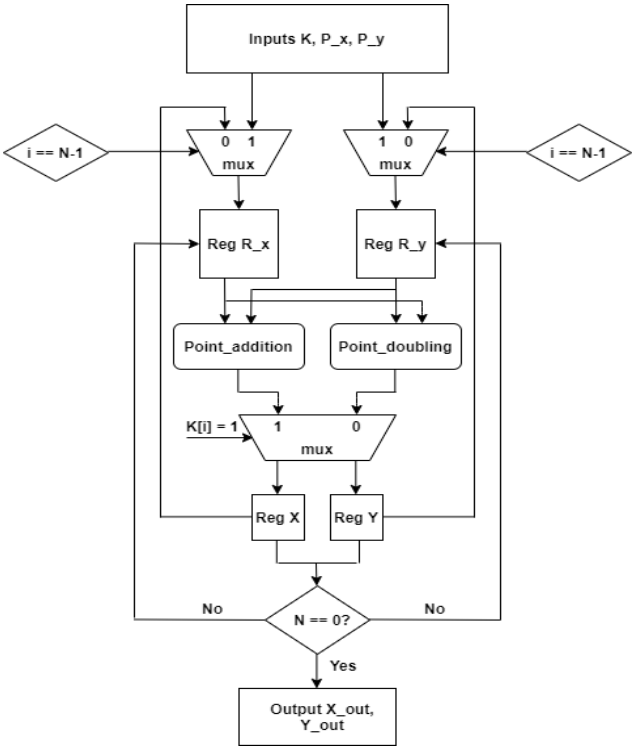

Figure 5. Proposed architecture of point multiplication

Algorithm 2 Double and Add algorithm

Result: Q = kP

Inputs: k = $(\sum_{i=0}^{N-1} k_i 2^i)$, P, N

Q = P, i = N-2

while $i <= 0$ **do**

$Q \leftarrow 2Q$; Point Doubling

if $k_i = 1$ **then**

$Q \leftarrow Q + P$; Point Addition

else

$Q \leftarrow Q$;

end

i = i - 1

end

return Q

5. IMPLEMENTATION RESULTS

Efficient Hardware architectures suggested for every mathematical process in ECDSA have been modelled using hardware programming language known as verilog. Modelsim simulator was utilized to verify the functionalities of the designed structures. Every module proposed are synthesized and implemented in Xilinx Virtex 7 FPGA and the device chosen is xc7v2000t-2flg1925 in Xilinx ISE 14.7 design suite. The results obtained after implementation of building blocks modular multiplication, modular multiplicative inverse proto- types for various bit-size are encapsulated in Tables 1 and 2 respectively and other designs results are represented graphically.

Using the mentioned Virtex-7 FPGA, the results listed shows that all structures suggested works at an average frequency of 137 MHz for a bit-length of 256 bits. If the size of N is taken to be 256, modular multiplication consumed only 141 slices on the other hand mathematically intricate multiplicative inverse ex-pended about 900 slices. ECC Group operations such as point addition made use of 1321 at the same time 1385 slices were taken over for point doubling. Area consumption for point multiplication, signature generation and verification were found to be 6030, 6924 and 21055 slices respectively.

Table 1. Synthesis results of modular multiplication using Xilinx ISE design suite

Bit-size	#Registers	#LUTs	#LUT-FF pairs	#IOBs	Area (Slices)	Speed (Mhz)
24	101	140	79	98	13	384.20
32	134	183	104	130	17	366.77
64	263	357	200	258	33	295.29
86	351	478	266	346	44	260.47
103	419	570	318	414	53	238.24
142	576	770	450	570	72	200.20
160	648	867	491	642	81	186.35
163	660	880	492	654	83	184
192	776	1035	541	770	97	165.94
224	904	1198	549	898	113	149.577
256	1123	1305	592	1014	141	136.59

Table 2. Synthesis results of modular multiplicative inverse in Xilinx ISE design suite

Bit-size	#Registers	#LUTs	#LUT-FF pairs	#IOBs	Area (Slices)	Speed (Mhz)
24	689	1743	589	74	87	264.53
32	911	2318	779	98	114	250.33
64	1809	4500	1593	194	226	231.30
86	2426	6024	2077	260	304	214.97
103	2897	7335	2523	311	362	205.62
142	3995	10060	3422	428	500	185.62
160	4500	11220	3908	482	563	177.44
163	4584	11417	3925	491	573	176.18
192	5395	13327	4622	578	675	164.49
224	6305	14349	5619	674	788	145.87
256	7201	16388	6365	770	900	137.22

6. PERFORMANCE COMPARISON WITH EXISTING WORKS

Various works have been published and is still ongoing with regard to the hardware implementation of different ECDSA modules such as modular multiplication, Modular multiplicative inverse, Point multiplication and ECDSA signature generation and verification. Table 3 displays the performance comparison results of various proposed architectures with the similar existing works over GF (256). It is indeed a challenging effort to optimize the hardware resource usage and at the same time to attain high frequency of operation since both performance parameters are contradicting when it comes to FPGA based implementations. In this work, a trade-off is maintained between area and speed in comparison to other related designs.

Table 3. Performance comparison of the various proposed architectures with existing works over GF (256)

Work	Platform	Reported Area (slices)	Frequency (Mhz)
		Modular Multiplication	
This work	Virtex-7	1305	136.59
[30]	Virtex-7	1400(LUTs)	134.7
[30]	Virtex-5	2259(LUTs)	110.5
[31]	Virtex-6	23977(LUTs)	40.1
[32]	Virtex-6	3877(LUTs)	95.2
[33]	Virtex-4	31946(LUTs)	60
		Modular multiplicative inverse	
This work	Virtex-7	900	137.22
[37]	Virtex-7	1480	146.38
[39]	Virtex-7	1577	138.3
[41]	Virtex-5	592	129
[40]	Virtex-2	5863	55.7
[38]	Virtex-2	2085	68.17
		Elliptic Curve Point Multiplication	
This work	Virtex-7	6030	136.77
[23]	Virtex-7	8900	177.7
[35]	Virtex-7	24200	72.9
[26]	Kintex-7	11300	121.5
[27]	Virtex-5	10200	66.7
[25]	Virtex-4	35700	70
		ECDSA Verification	
This work	Virtex-7	12055	136.81
[16]	Nand gate 45nm CMOS	1034kGE	295
[36]	Virtex-5	36000	39.7
[18]	Virtex-2	15755	39.5
[17]	Virtex-2	3529	67

Modified Interleaved modular multiplication is proposed in [30] that has been realized in various FPGA platforms. In virtex-7, it had a usage of 1400 LUTs with a speed of 134.7 MHz while in virtex-5 the same prototype demanded 2259 LUTs functioning at a reduced frequency of 110 MHz. Yang *et al.* [31] proposed Montgomery algorithm is adopted to build modular multiplication over F_{256} with reduced latency. The design on virtex-6 platform consumed about 23,977 LUTs in terms of area with an operational frequency of 40.1 MHz. Radix-4 booth encoded algorithm was utilized by [32] to implement modular multiplier. Three consecutive digits of multiplier from LSB to MSB was checked to get the product of multiplicand and {0,-1 or +1,-2 or +2 } and to add it along with the 2-bit shifted version of partial product. This design expended 3877 LUTs on virtex-6 FPGA for a speed of 95.2 MHz. Ananyi *et al.* [33] studied make use of normal multiplier that executes modular reduction after the completion of multiplication rather than performing bitwise multiplication proceeded by modular reduction. As a result, this multiplier demands huge amount of area in virtex-4 device with the utilization of about 31,946 LUTs in addition to 32 DSP slices with a speed performance of 60 MHz. In comparison to the above discussed architectures, this work spends only 1305 LUTs in virtex-7 FPGA while operating at a higher frequency of 136.59 MHz which assures better performance of the proposed design.

The technique proposed to implement multiplicative inverse design attained an operating frequency of 137.22 MHz with resource utilization of only 900 slices over GF (256). In [37], [39] works implemented using virtex-7 device showcased higher speed but with increased area consumption. Other designs [38], [40], [41] synthesized using virtex-5 and virtex-2 FPGA had lower performance in terms of both slices and frequency of execution. Performance analysis with other works highlighted in the table shows that a trade-off is achieved between area and speed of execution by the method adopted in this research for inversion.

Highly efficient ECPM architecture realized in both ASIC and FPGA is introduced in [26]. However, in kintex-7 platform the design utilized about 1.9 times extra slices and is 1.1 times slower than proposed elliptic curve point multiplier put forward in this paper. Asif *et al.* [35] based on RNS related point multiplication consumes 24.2 k slices to achieve a speed of 72.9 MHz. While [27] occupied 10.2 k slices to execute ECPM, [25] had to use 35.7 k slices for the same to attain a speed of about 70 MHz. Even though [23] has higher operational speed of 177.7 MHz it was found to absorb more slices compared to the work discussed in this paper that employed only 6030 slices to work at a frequency of 136.77 MHz.

Analysis of performance parameters for ECDSA signature verification indicates that the suggested design for ECDSA exhibited better performance in all parameters examined. Knezevic *et al.* [16] considered NAND gate complementary metal oxide semiconductors (CMOS) technology for 45nm for its ASIC implementation but demanded huge logic consumption. All other works [17], [18], [36] had lower execution speed when compared to the design proposed in this work. Figure 6 illustrates the area utilization and speed of operation attained by various ECDSA components implemented in FPGA for different bit-sizes. The graphical result reveals that all the elements of ECDSA were able to attain consistency in terms of operational speed among themselves. However, it displayed a linear increase in area consumption with the bit-size value.

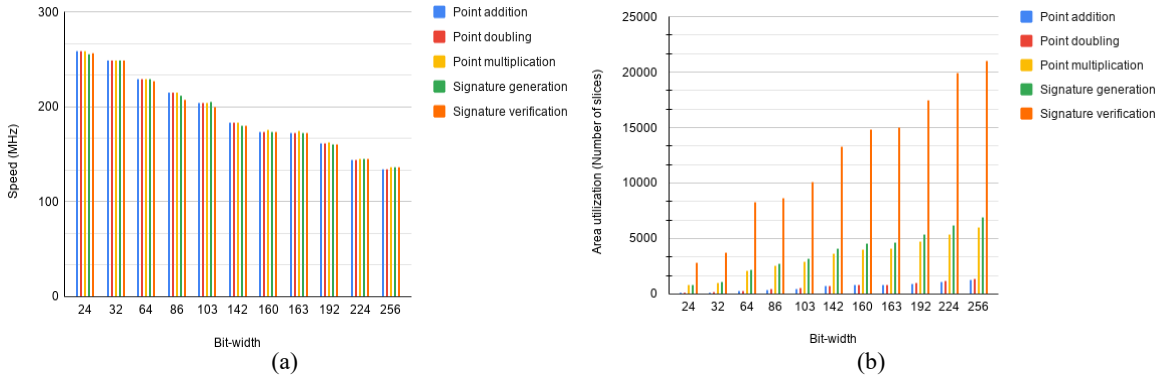

Figure 6. Comparison of performance parameters among various ECDSA components for varying bit-size, (a) operational frequency, (b) area utilization

7. CONCLUSION AND FUTURE SCOPE

In this work, highly efficient re-configurable hardware prototypes are introduced to assemble elliptic curve digital signature architecture for any N-bit value. Novel architectures are suggested for modular multiplication and multiplicative inverse circuitry to optimize resource utilization and improve the operational speed of the entire ECDSA framework. Xilinx Virtex 7 is the FPGA device utilized to implement all designs proposed up to bit size of 256 bits. The results shows that 256-bit ECC signature generation make use of 4956 slices running with a frequency of about 136.81 MHz. While the same size ECC verification runs with equivalent speed but consuming about 21055 slices owing to its higher complexity. Performance analysis with similar works prove that all designs suggested were found to manifest better performance with respect to area and speed. Comparison results with existing works clearly highlights the trade-off between frequency and area consumption for the modules designed with other equivalent structures. Future scope of this work is to verify blockchain transactions using the above realized re-configurable structures to enable application of blockchain technology in hardware security.

REFERENCES

[1] J. Al-Jaroodi and N. Mohamed, "Blockchain in Industries: A Survey," in *IEEE Access*, vol. 7, pp. 36500-36515, 2019, doi: 10.1109/ACCESS.2019.2903554.

[2] T. Ali Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem and T. Alghamdi, "A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations," in *IEEE Access*, vol. 7, pp. 176838-176869, 2019, doi: 10.1109/ACCESS.2019.2957660.

[3] B. Shahzad and J. Crowcroft, "Trustworthy Electronic Voting Using Adjusted Blockchain Technology," in *IEEE Access*, vol. 7, pp. 24477-24488, 2019, doi: 10.1109/ACCESS.2019.2895670.

[4] S. Mondal, K. P. Wijewardena, S. Karuppuswami, N. Kriti, D. Kumar and P. Chahal, "Blockchain Inspired RFID-Based Information Architecture for Food Supply Chain," in *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5803-5813, June 2019, doi: 10.1109/JIOT.2019.2907658.

[5] J. Liu and Z. Liu, "A Survey on Security Verification of Blockchain Smart Contracts," in *IEEE Access*, vol. 7, pp. 77894-77904, 2019, doi: 10.1109/ACCESS.2019.2921624.

[6] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 858-880, 2019, doi: 10.1109/COMST.2018.2863956
[7] D. Toradmalle, R. Singh, H. Shastri, N. Naik and V. Panchidi, "Prominence of ECDSA Over RSA Digital Signature Algorithm," *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 2018, pp. 253-257, doi: 10.1109/I-SMAC.2018.8653689.
[8] A. Rezai and P. Keshavarzi, "A New Finite Field Multiplication Algorithm to Improve Elliptic Curve Cryptosystem Implementations," *Journal of Information Systems and Telecommunication*, vol. 1, no. 2, pp. 119-129, 2013, doi: 10.7508/jist.2013.02.006
[9] A. Rezai and P. Keshavarzi, "High-performance implementation approach of elliptic curve cryptosystem for wireless network applications," *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet),* 2011, pp. 1323-1327, doi: 10.1109/CECNET.2011.5768248.
[10] A. Rezai and P. Keshavarzi, "CCS Representation: A New Non-Adjacent Form and its Ap- plication in ECC,"*Journal of basic and applied scientific research*, vol. 2, no. 5, pp. 4577-4586, 2012, doi: 53341775.
[11] X. Wang, "Research on ECDSA-Based Signature Algorithm in Blockchain," *Universe Scientific Pub- lishing*, vol. 4, no. 2, pp. 55-58, 2019, doi: 10.18686/fm.v4i2.1600.
[12] M. Sahoo, S. S. Singhar, B. Nayak and B. K. Mohanta, "A Blockchain Based Framework Secured by ECDSA to Curb Drug Counterfeiting," *2019 10th International Conference on Computing, Communica- tion and Networking Technologies (ICCCNT)*, Kanpur, India, 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944772.
[13] G. Nyame, Z. Qin, K. O. B. O. Agyekum, and E. B. Sifah, "An ECDSA Approach to Access Control in Knowledge Management Systems Using Blockchain," *Informa- tion 2020*, no. 2, pp. 1-15, 2020, doi: 10.3390/info11020111.
[14] M. Alavi, "Knowledge management and knowledge management systems: Conceptual foundations and research issues," *Management Information Systems Quarterly*, vol. 25, pp. 107-136, 2001, doi: 10.2307/3250961
[15] K. Feng, E. T. Chen, and W. Liou, "Implementation of knowledge management systems and firm per- formance: An empirical investigation," *Journal of Computer Information Systems*, vol. 45, pp. 92-104, 2004, doi: 10.1080/08874417.2005.11645835.
[16] M. Knezevic, V. Nikov and P. Rombouts, "Low-Latency ECDSA Signature Verification-A Road Toward Safer Traffic," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 24, no. 11, pp. 3257-3267, Nov. 2016, doi: 10.1109/TVLSI.2016.2557965.
[17] N. Mentens, "Secure and efficient coprocessor design for cryptographic applications on FPGAs," Ph.D. dissertation, Dept. ESAT/COSIC, Katholieke Univ. Leuven, Leuven, Belgium, 2007, doi: LIRIAS1662239.
[18] C. McIvor, M. McLoone, and J. V. McCanny, "An FPGA elliptic curve cryptographic accelerator over GF (p)," *Proc. Irish Signals and Systems Conference (ISSC)*, 2004, pp. 589-594, doi: 10.1049/cp:20040606.
[19] X. Fang and Y. Wu, "Investigation into the elliptic curve cryptography," *2017 3rd International Conference on Information Management (ICIM)*, Chengdu, 2017, pp. 412-415, doi: 10.1109/INFOMAN.2017.7950418.
[20] K. Keerthi and B. Surendiran, "Elliptic curve cryptography for secured text encryption," *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Kollam, India*, 2017, pp. 1-5, doi: 10.1109/ICCPCT.2017.8074210.
[21] B. Qing-Hai, Z. Wen-Bo, J. Peng and L. Xu, "Research on Design Principles of Elliptic Curve Public Key Cryptography and Its Implementation," *2012 International Conference on Computer Science and Service System, Nanjing, China*, 2012, pp. 1224-1227, doi: 10.1109/CSSS.2012.310.
[22] M. Janagan and M. Devanathan, "Area compactness architecture for elliptic curve cryptography," *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012), Salem, India*, 2012, pp. 131-134, doi: 10.1109/ICPRIME.2012.6208300.
[23] M. M. Islam, M. S. Hossain, M. K. Hasan, M. Shahjalal and Y. M. Jang, "FPGA Implementation of High- Speed Area-Efficient Processor for Elliptic Curve Point Multiplication Over Prime Field," *IEEE Access*, vol. 7, pp. 178811-178826, 2019, doi: 10.1109/ACCESS.2019.2958491.
[24] J. Li, S. Zhong, Z. Li, S. Cao, J. Zhang and W. Wang, "Speed-Oriented Architecture for Binary Field Point Multiplication on Elliptic Curves," *IEEE Access*, vol. 7, pp. 32048-32060, 2019, doi: 10.1109/ACCESS.2019.2903170.
[25] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "FPGA based high-speed SPA-resistant elliptic curve scalar multiplier architecture," *International Journal of Reconfigurable Computing*, no. 5, pp. 1-10, 2016, doi: 10.1155/2016/6371403.
[26] M. S. Hossain, Y. Kong, E. Saeedi, and N. C. Vayalil, "High-performance elliptic curve cryptography processor over NIST prime fields," *IET Computers and Digital Techniques*, vol. 11, no. 1, pp. 33-42, 2017, doi: 10.1049/iet-cdt.2016.0033.
[27] H. Marzouqi, M. Al-Qutayri, and K. Salah, "An FPGA implementation of NIST 256 prime field ECC processor," *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), Abu Dhabi, United Arab Emirates*, 2013, pp. 493-496, doi: 10.1109/ICECS.2013.6815461.
[28] J. Li, S. Zhong, Z. Li, S. Cao, Zhang, and W. Wang, "Modular multiplication without trial division," *Mathematics of Computations*, vol. 44, no. 170, pp. 519-521, 1985, doi: 10.1090/S0025-5718-1985-0777282-X.
[29] J. Bajard, L. Didier, and P. Kornerup, "An RNS Montgomery modular multiplication algorithm," *IEEE Transactions on Computers*, vol. 47, no. 7, pp. 766-776, July 1998, doi: 10.1109/12.709376.
[30] M. M. Islam, M. S. Hossain, M. Shahjalal, M. K. Hasan and Y. M. Jang, "Area-Time Efficient Hardware Implementation of Modular Multiplication for Elliptic Curve Cryptography," *IEEE Access*, vol. 8, pp. 73898-73906, 2020, doi: 10.1109/ACCESS.2020.2988379.

[31] Y. Yang, C. Wu, Z. Li, and J. Yang, "Efficient FPGA implementation of modular multiplication based on Montgomery algorithm," *Microprocessors and Microsystems*, vol. 47, pp. 209-215, 2017, doi: 10.1007/978-3-642-31494-0_47.
[32] K. Javeed, X. Wang, and M. Scott, "High performance hardware support for elliptic curve cryptography over general prime field," *Microprocessors and Microsystems*, vol. 51, pp. 331-342, 2017, doi: 10.1016/j.micpro.2016.12.005.
[33] K. Ananyi, H. Alrimeih and D. Rakhmatov, "Flexible Hardware Processor for Elliptic Curve Cryptogra- phy Over NIST Prime Fields," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 17, no. 8, pp. 1099-1112, Aug. 2009, doi: 10.1109/TVLSI.2009.2019415.
[34] A. Rezai and P. Keshavarzi, "High-Throughput Modular Multiplication and Exponentiation Algorithms Using Multibit-Scan-Multibit-Shift Technique," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1710-1719, Sept. 2015, doi: 10.1109/TVLSI.2014.2355854.
[35] S. Asif, M. S. Hossain, and Y. Kong, "High-throughput multi-key elliptic curve cryptosystem based on residue number system," *IET Computers and Digital Techniques*, vol. 11, no. 5, pp. 165-172, 2017, doi: 10.1049/iet-cdt.2016.0141.
[36] K. Sakiyama, E. De Mulder, B. Preneel and I. Verbauwhede, "A Parallel Processing Hardware Architec- ture for Elliptic Curve Cryptosystems," *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, France*, 2006, pp. 904-907, doi: 10.1109/ICASSP.2006.1660801.
[37] M. S. Hossain, and Y. Kong, "High-performance FPGA implementation of modular inversion over F_{256} for elliptic curve cryptography," *IEEE International Conference on Data Science and Data Intensive Sys- tems (DSDIS)*, 2015, pp. 169-174, doi: 10.1109/DSDIS.2015.47.
[38] J. Lee, S. Chung, H. Chang and C. Lee, "Efficient Power-Analysis-Resistant Dual-Field Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture," *IEEE Transac- tions on Very Large-Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 49-61, Jan. 2014, doi: 10.1109/TVLSI.2013.2237930.
[39] T. Kudithi, and R. Sakthivel, "High-performance ECC processor architecture design for IoT security applications," *The Journal of Supercomputing*, vol. 75, no. 1, pp. 447-474, 2019, doi: 10.1007/s11227-018-02740-2.
[40] Z. Liu, D. Liu and X. Zou, "An Efficient and Flexible Hardware Implementation of the Dual-Field Elliptic Curve Cryptographic Processor," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2353- 2362, March 2017, doi: 10.1109/TIE.2016.2625241.
[41] A. Mrabet, N. El-Mrabet, B. Bouallegue, S. Mesnager, and M. Machhout, "An efficient and scalable modular inversion/division for public key cryptosystems," *International Conference on Engineering MIS (ICEMIS)*, 2017, pp. 1-6, doi: 10.1109/ICEMIS.2017.8272995.

BIOGRAPHIES OF AUTHORS

Devika K. N. is a research scholar in Electronics and Communication department at Amrita school of engineering with Master of Engineering degree in VLSI Design from Amrita university, Kollam, India (2017). She received her B. Tech degree in Electronics and Communication Engineering from MG University in 2015.Her research interests include Blockchain concepts, Hardware security, Cryptography, VLSI testing, Field Programmable Gate Arrays (FPGA), Digital system design and Logic synthesis.

Ramesh Bhakthavatchalu is currently working as Associate Professor in the Department of Electronics and Communication Engineering at Amrita Vishwa Vidhyapeetham since 2005. He completed his M.E in Applied Electronics from College of Engineering, Guindy. In 2016, he received his Ph.D. degree in Electronics and Communication Engineering from Amrita University. His current research interests include hardware security, Automatic test pattern generation (ATPG), LBIST, Cryptography, Logic testing, VLSI testing, Formal verification, Signal processing, Logic design and Field programmable gate arrays. He has more than 10 years of experience in core VLSI industries across the world. He worked as Design Engineer in Arasan Chip Systems Inc. for 4 years and application engineer in SynTest Technologies, Inc. for 2 years. He is an active reviewer of IEEE access journals and other scientific publications.