Implementation of ECC on FPGA using Scalable Architecture with equal Data and Key for WSN

Leelavathi.G [#1], Shaila.K [*2], Venugopal K.R [#3]
[#1] Visvesvaraya Technological University, [#3]Bangalore University
[#1] Research Schloar, Vivekananda Institute of Technology
[#3]Principal, University Visvesvaraya College of Engineering
[1] nisargamodini@gmail.com
[3] venugopalkr@gmail.com
[*2] Vivekananda Institute of Technology
Professor and Head, Department of Electronics and Communication
[2] shailak17@gmail.com

Abstract—**Security of data transferred on the Wireless Sensor Network is of vital importance. In public key cryptography RSA algorithm has been used for a long time, but it does not meet the constraints of WSNs. Elliptic Curve Cryptography(ECC) has been employed recently because of its highest security for same length bit. ECC point multiplication operation is time consuming which affects the speed of encryption and decryption of data. Security in WSNs is addressed in our work, where a modified ECC is designed by performing the point multiplication using Montgomery multiplication technique that achieves considerable speed and with reduced area utilization. The ECC is first simulated on different FPGA devices, with key length 11, 112, 131 and 163 bits and the area-speed tradeoff is compared. ECC algorithm is implemented with software and hardware choosing Artix 7 XC7a100t-3csg324 FPGA which supports key lengths of 11, 112, 131 and 163 bits. When implemented on a Artix 7 FPGA, it completes 163 bit data encryption operation over GF(2^{163}) in 1ms with the maximum frequency of 229MHz.**

The ECC algorithm is reconfigurable with low level to high level security with different bit key sizes. The proposed ECC algorithm modeled using VHDL and synthesized on Spartan 3 and 6, Virtex 4, 5 and 6 and Artix7 before the hardware implementation on Atrix 7. The design satisfies the needs of resource constrained devices by decreasing the encryption and decryption time to 1 ms with equal keylength and datasize, while device utilization is within 13%.

Keyword-- Publlic key Cryptography, Elliptic Curve Cryptography, Montgomery, Scalar Multiplica-tion, Wireless Sensor Networks, FPGA.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) comprises of one or more base stations that are of low-cost, low power with multi-functional sensor nodes. WSNs form an adhoc network, which interact with one or more sink nodes with the outside world for communication. Execution of tasks between the sensors is instructed with data aggregation, followed by processing and transfer. The network connectivity and coverage area, always def nes maximum participating nodes in a sensor network. There exist a number of applications such as Industrial control, environment observation, health monitoring and military applications. In these scenarios frequently exchange of data among sensor nodes are done in insecure format. Hence, it is necessary to provide security for the WSNs against the threats in an unsecured channel that leads to achieve an integrity, authenticity and conf dentiality.

In Wireless sensor networks security issues can be categorized as: key management, cryptography, secure routing, secure data aggregation and intrusion detection. Security requirements in WSNs are:

- *Data Confidentiality:* Ensures that authorized sensor nodes, can only access the contents of messages.
- *Data Authentication*: Confirms identity of the sender i.e., data sent is from a known source.
- *Data Integrity*: Information cannot be modified easily. It is performed by encryption and digital signature.
- *Data Freshness*: This ensures that no old data is replayed. This type of security requirement is essential when keys are changed from time to time and when the key distribution occurs.
- *Data Availability*: Ensures that services that are offered by WSNs or by a single node on demand.

In Embedded system market, WSNs applications have greater share with promising future. WSNs are deployed in many real-world applications, including Ambient Intelligence and Ubiquitous Computing. Owing to scarcity of energy, unsecure channel and intensive mathematical operations of asymmetric cryptographic primitives, it is difficult to realize secure WSNs. Due to these exclusive challenges, security of WSNs become a very important

topic in the research area. Many mechanisms are explored to provide security for WSNs. Regardless of the efficiency of these mechanisms, sufficient security cannot be provided if they are implemented on top of unsecure platforms.

In certain application domains, security and privacy issues cause a great challenge for the current and future embracing of WSN technology. WSNs are hard to protect than a conventional network because the sensor nodes are often deployed in unattended environments. This implies that an attacker may be able to directly access individual node. In this scenario, an attacker capturing one or more nodes can extract secret keys by performing physical attacks. The adversary can manipulate the functionality of nodes to compromise the correct operation and/or security of the WSNs. As a result, WSNs necessitate a sophisticated security architecture that takes up these unusual threat scenarios and adversary models into account. Most of the security protocols designed on the basis of computation-demanding asymmetric/public key cryptography are not easily adaptable to WSNs, mainly due to the low processing power of sensor nodes.

WSNs security requirements are similar to Conventional Computer Networks. The parameters for instance, authenticity, confidentiality, integrity and availability are deeply considered in creation of a network environment. All security solutions designed for conventional computer networks cannot be implemented directly on WSNs. Due to high processing power, it was understood that the public key cryptography was not appropriate for WSNs. But through studies of encryption algorithms based on curves proved the feasibility of using public key cryptography in WSNs [1][2]. Some of the public key cryptographic methods include RSA and Rabins scheme. RSA is one of the first practical public key crypotsystems and is generally used for secure data transmission.

Shaila et al., [3] discussed about constraints of WSNs and proposed a scheme called Modified Blooms Scheme (MBS). The asymmetric matrices of keys are used in place of symmetric matrices in order to establish secret keys between node pairs. Implementation emphasizes on improvement in network resilience against node capture attack. In [4] Secured data fusion technique reduces the packet drop that is caused by the malicious nodes and at the same time reduces the traffic load and increase the network life time. The network is secure and reliable. A light weight cryptosystem, using AES and ECC is proposed. ECC is used to verify the authenticity of the data that ensures data integrity and authenticity of node [5].

Specific architecture of sensor networks and resource limitations are identified for custom-made security mechanisms in WSNs. To accomplish the cryptographic computation there exist the following possible solutions: (i) Software (ii) Hardware with Application-Specific Integrat ed Circuits (ASICs) (iii) Hardware with Field-Programmable Gate Arrays (FPGAs). Solutions with software are cheaper and most flexible, but performance is slow. The solutions with ASICs are faster, inflexible performance, very expensive and needs long development time. The FPGAs solution is flexible, fast and the development time required is less [6][7]. The advantages of using Reconfigurable Hardware FPGA for security algorithm implementations are: (i) Algorithm agility (ii) Algorithm upload (iii) Architecture efficienc y (iv) Resource efficiency (v) Algorithm modification, (vi) High Throughput and Cost efficient. Public Key Cryptography algorithms are the most promising schemes with respect to energy and time consumption, which makes it suitable for data encryption in WSNs.

Motivation: Various security mechanisms projected for WSNs in literature avoids Private key/asymmetric primitives, as a result of their high energy and time utilization, in particular for software implementations. The conception of infeasibility of these asymmetric primitives has been changed moderately because of the development of new asymmetric algorithms which are more eff cient than RSA i.e., ECC. Elliptic Curve Cryptosystem with 163 bit is known to provide a security level comparable to that of a 1024-bit RSA-based cryptosystem. A 2048-bit RSA-based cryptosystem is comparable to a 224-bit Elliptic Curve Cryptosystem. The Elliptic Curve Cryptography (ECC) is expected to be the most suitable asymmetric cryptographic primitive for WSNs. In ECC the per-bit security is higher than in other asymmetric algorithms. ECC is derived from algebraic concepts associated with elliptic curves over Galois Fields i.e., binary f elds GF(2^m) or prime f elds GF(P). Binary f elds are more attractive, for low energy hardware implementations. The operations engaged in binary f elds are only shifts and bitwise addition modulo 2 arithmetic [6].

Contribution: To achieve secure communication in WSNs, we have implemented public key cryptographic system based on Elliptic curves with Montgomery multiplication technique that promises faster and memory eff cient encryption and decryption compared to other traditional approaches. This work concentrates on the design and development of the ECC cryptosystem for sensor nodes used in WSNs. Montgomery Multiplier is used in the design for increasing the eff ciency of the system by optimizing the area and throughput parameters based on FPGA. Different FPGA with variable capacity has been chosen for simulation with bit sizes 11, 112, 131, 162 and the parameters are compared with earlier work. Our work concentrates on software implementation, followed by hardware implementations of PKC on FPGA.

Organization: In Section II research works related to security techniques, public key cryptography and elliptic curve cryptography, different multiplication algorithms are explained. Background work is discussed in Section III. In Section IV def nes the Problem and network model is described in Section V. Modular multiplication with Montgomery techniques and Elliptic Curve Cryptography are discussed in Section VI. Algorithm is explained in Section VII. Implementation and Performance Evaluation details are explained in Section VIII and Section IX respectively. Section X contains Conclusions.

II. LITERATURE SURVEY

Neal Koblitz and Victor Miller proposed Elliptic Curve Cryptography (ECC) in 1985. Elliptic Curve Cryptographic schemes provide similar functionality as RSA schemes. Currently, ECC is treated as the most eff cient public key cryptosystem because it utilizes shorter keys. To implement public key cryptography in particular, ECC on Embedded devices, numerous researches were conducted. For computation of ECC scalar multiplication, in literature several hardware implementations have been reported.

Abidalrahman et al., [8] scrutinizes the security problems in WSNs scenario and checks the existing techniques to implement the cryptographic approaches used to provide security services for WSNs. Further, they have proposed a customized hardware platform to f x holes for Wireless Sensor Networks. The cryptoprocessor utilizes AES algorithm along with ECC. The synthesis results on Spartan VI shows utilisation of 4828 slices i.e., 53% of device utilization. Ajay et al., [9] demonstrated the hardware implementation of Elliptic Curve Cryptography on Vertex-Pro XCV1000 FPGA. Conditional successive subtractions used to perform the modular arithmetic division operations with reduction in area, that is suitable for the systems which has limited storage space and computation ability. Arya et al., [10] design and implement high performance cryptosystem on a single FPGA Virtex6 XC6VCX75T based on Elliptic curve cryptosystems with projective co-ordinate representation and normal basis representation over GF(2^m) to reduce hardware complexity. A single module is used, on the basis time sharing and there is increase in hardware for implementing message encryption by 1%.

Bednara et al., [11] analyze how eff ciently mapping of f eld multipliers to lookup table based FPGAs can be done. A LFSR and a word-serial Massey-Omura multipliers are compared with respect to area and performance. The multipliers are analyzed in two different models (i) a classical 2-input gate model (ii) a 4-input LUT model for FPGA. Roy et al., [12] illustrate suitable scheduling for performing point addition and doubling in a pipelined data path of the ECSMA. It is observed high speed design with signif cantly low area due to better utilization of LUTs.

Shylashree et al., [13] introduce Ancient Indian Vedic Mathematics for prime f elds with Point addition, doubling to speed up Scalar Multiplication. Implementation emphasizes on point doubling rather than the complete ECC operation.Houssain et al., [14] provide, study of hardware implementations of ECC in Wireless Sensor Networks. ECC using normal basis representation over GF(2^m) is implemented in this work.

Rahuman et al., [15][16]propose an architecture with Lopez-Dahab Elliptic Curve Point Multiplication algorithm that is based on an eff cient Montgomery add and double algorithm using Xilinx XCSVLX200 FPGA device. GF(2^{193}) with Gaussian normal basis for f eld arithmetic, KaratsubaOfman multiplier and Itoh-Tsujii algorithm are used as the inverse component. The design lacks the storage and computational power.

Hassan et al., [17] investigate the strength of a hardware/software co-design techniques to realize a scalable Elliptic Curve Cryptography (ECC) processor with low resources. The processor designed with binary f nite f elds GF(2^m) on an platform of FPGA. The software is run on a free-soft-core processor from Xilinx FPGA i.e., PicoBlaze. The two novel arithmetic circuits serve as the hardware platform to perform multi-precision arithmetic and scalable reduction. Cui et al., [18] introduced the concepts of Finite binary f eld and Public Key cryptography i.e., Elliptic Curve Cryptography. Some of the operations are optimized by mapping the ECC algorithm to hardware. Authors achieve a good speed up with their processor for ECC computing.

Thapiyal et al., [19] utilizes Vedic Mathematics to implement Elliptic Curve Encryption that speeds up the task of multiplication process. Houssain et al., [20] describe an implementation of an ECC crypto processor on a Nano Field Programmable Gate Array. Nano FPGAs provide groundbreaking possibilities with essential parameter i.e., power, size, cost and performance. The ECC crypto processor is coded with VHDL and synthesized on Nano FPGA Actel IGLOO AGLN250V2VQFP100 and satisfy the requirements of nodes in Wireless Sensor Network (WSN).

Hassan et al., [21] present, a scalable Elliptic Curve Point Multiplication on FPGA. Out of the two new low area designs, f rst design is a pure software implementation and second is novel hardware architecture. The designs are implemented on a PicoBlaze from Xilinx 8-bit embedded microcontroller which is a compact ECC processor. The device utilization is 23% and 61% with Spartan-III XC3S50.

Talapatra et al., [22] propose multiplication architecture based on Montgomery multiplication (MM) algorithm, for ECC. In order to simplify the multiplications over GF(p), the polynomial coeff cients are represented in Montgomery residue format. Carry-Save-Adder (CSA) based implementation is utilized to reduce the complexity of the MM architecture over GF(p). In the proposed architecture area complexity is signif cantly

less. Leboeuf et al., [23] explicitly present a GPU implementation of the MM algorithm which is optimized for the GPUs SEVID architecture. The f eld sizes and parameter constraints required for Elliptic Curve Cryptography is also optimized.

Orlando et al., [24] present an optimized high performance elliptic curve co-processor that uses Lopez and Dahabs projective coordinate system specif cally for Koblitz curves, where a f eld multiplier performs a f eld multiplication over the extension f eld with degree of 163. A prototype processor runs at 66 MHz for both generic and Koblitz curve. On Xilinx XCV2000E FPGA prototype performs an elliptic curve scalar multiplication in 0.233 msec and 0.075 msec respectively. Sutter et al., [25] discuss the implementation of a new highspeed point multiplier used for Elliptic Curve Cryptography on ASIC. This works only for point multiplication than complete ECC operation. Inversion is performed using repeated multiplications and squaring requires extra time for computation.

Lee et al ., [26] propose a novel diverse Dual-Processing Element (Dual-PE) architecture based on priority-oriented scheduling of right-to-left double-and-add Elliptic Curve Scalar Multiplication (ECSM). Pontie et al., [27] designed a coprocessor that supports critical operations on ECC cryptosystem. It utilizes a window method that scans leftto-right the scalar value. The main drawback while using window method in scalar multiplication results in large area consumption because of variable window size and is not suitable for WSNs applications.

Munoz et al., [28] demonstrated the cryptoprocessors with two multipliers over f nite f eld GF(2^{163}) using digit-level processing. The Scalar Multiplication kP was performed using window-NAF algorithm with good computation time and area trade-off. It uses approximately 17% of the Arithmetic LUTs of FPGA. The architecture designed for generic curves and cannot be applied to accelerate scalar multiplication.

Table 1 depicts the survey on different methods of implementations of scalar multiplication and cryptographic process using various techniques. Most of the works are implemented using FPGAs, few are with ASIC design. In [10]-[13] Elliptic Curve Scalar Multiplication architectures are implemented using different multipliers. In [16], the design is implemented on Xilinx XC4VLX200 device. In this FPGA 16.209 slices are utilized, which shows more hardware usage compared to our design. In [19] Vedic mathematics squaring architecture helps to accelerate the process of point multiplication. The complete encryption and decryption process using different FPGA devices are concentrated in [18][20]. In [21], Montgomery multiplication algorithm with Lopex Dahab coordinates are utilized for the Hardware-software co-design with Picoblaze microcontroller with data width 8-32 bits and the design is implemented upto f eld width of 571 bits. But it leads to hardware complexity in implementation. ASIC implementation using MM architecture with primary f eld is discussed in [22]. The design uses Carry save representation of data for point multiplication. Further in [23] MM algorithm for the GPU with SIMD architecture, is heavily optimized for 112-521 bits. In [25][28] point multiplication is implemented using digit serial and window techniques. When these techniques are utilized for complete cryptographic process, mapping of area, energy consumption and computational time becomes very tedious to provide security for wireless sensor nodes. Our design involves the Montogomery Multiplication algorithm for the complete cryptographic process i.e., encryption and decryption with equal key and data sizes that achieve considerable area and time tradeoff.

A. Background

Xining et al., [18] propose an ECC based on FPGA. The computation process of ECC at bit level is parallelized. This modif cation achieves a considerable speed-up. The key lengths of 113-bit, 163-bit and 193-bits are supported by this ECC processor which is implemented in hardware. Multiplication is performed according to Karatsuba algorithm that consumes area and time.

Houssain et al., [20] utilize nanoFPGA for the implementation of ECC for resource constrained devices, providing low level security. Double and Add algorithm is utilized for scalar multiplication that consumes more time to perform the operation in ECC. The synthesis results reveals the information that the bit size cannot be more than 11 bits as they have chosen nano FPGA. This criteria affects power, size and cost of the processor designed. These limitations are addressed in our work based on the selection of FPGA and algorithm for implementation [20].

B. Problem Definition

Cryptographic algorithms on general purpose processors are not suitable for WSNs. To achieve high performance, hardware implementations are of quite important to provide security against system intruders. In some asymmetric algorithms, a cryptographic task is time consuming and key lengths no longer provide suff cient protection from contemporary threats. The main problem in asymmetric algorithms is poor utilization of chip area, low performance and low throughput architectures. Further, many of the architectures do not utilize the area eff ciently resulting in high cost when implementation is done using silicon.

C. Objectives

The main objectives are :

• Implement Elliptic Curve Cryptography on FPGA to increase the area utilization and speed. The area and throughput are traded that makes it suitable for Wireless Sensor Node Communication.

• Provide required security level, through selection of appropriate Elliptic Curve parameters set.

D. Assumptions

• Encode the character set to the points for elliptic curve with equation 1, by choosing p1(x,y), p2(x,y), p3(x,y) .,

• The public key Kb (computed using Montgomery multiplication) of the sender is published and is made available to the receiver.

III. PRELIMINARIES

The Public Key Cryptography (PKC) involves cryptographic operations which are computationally intensive and are incompatible for providing security services in WSN. Recently it has been proved that PKC is in fact possible to be realized in sensor networks. Even though its realization on WSNs is diff cult, Public Key Cryptography brings immense simplicity and effectiveness in providing security services.

Due to its less signif cant key size, Elliptic Curve Cryptography is having good potential for Wireless Sensor Network security, as there is a possibility to reduce key calculation time. It uses Scalar Multiplication that consumes around 80% of key computation time on WSNs motes. Elliptic Curve PKC Cryptosystems are up-and-coming as a new creation of cryptosystems. They offer the smallest key size and maximum strength per bit in comparison to other public key cryptosystem. Smaller key sizes make them highly suitable for hardware implementation on FPGAs.

The earlier versions of PKC was very expensive but it is partially changed for WSN as a result of the existence of innovative hardware and software prototypes which are based on Elliptic Curve Cryptography and new Public Key Cryptography primitives.

E. Elliptic Curve Cryptography

A number of cryptographic schemes exists, based on elliptic curves that exert on a subgroup of points of an Elliptic Curve over a f nite f eld. Variables and coeff cients are constrained to elements of a f nite f eld. ECC relies on a group structure on an elliptic curve. Arbitrary f nite f elds are accepted to be appropriate for ECC. Two families of elliptic curves are def ned for use in cryptography,

• Binary f elds denoted by GF (2^m), where m is big integer.

• Prime f elds denoted by GF (p), where p is the big prime number.

In IEEE P1363 Public Key Cryptography are used for successful implementation of the Elliptic Curve Cryptography. The binary f eld uses only two coeff cients 0 and 1, hence coding can be done very easily and is well suited for hardware implementations. The prime f eld GF (p)) requires the modular operations performing coprocessor that increases the speed of operation.

TABLE I. COMPARISION OF OUR WORK WITH PREVIOUS WORKS

Author	Algorithm	Performance	Advantages	Disadvantages
Abidalrahman et al.,[8] 2011	Redesign of the WSN node, with AES,ECC and SHA	More security with three Algorithms	Different approach for WSN node	more hardware required
Arya et al.,[10] 2014	Binary left to right algorithm	Computational delay time reduced	Word level Multiplier reduces hardware complexity	Only scalar multiplication, no encryption and Decryption process
Roy et al., [12] 2013	ECSMA with Karatsuba multiplier	high-speed architecture small area	Discussion of Frameworks MM based scalar multiplier	Only scalar multiplication, Process
Shylashree et	Ancient Indian	Compare to Thapiyal	Implemented for 8/16	Only Point Multiplication

al.,[13] 2012	Vedic Mathematics	[19] delay is reduced	bits, performance is good	is carried out
Khaleel et al.,[16] 2010	Lopez-Dahab elliptic curve PM algorithm, GNB basis for arithmetic Karatsuba-Ofman multiplier and Itoh-Tsujii algorithm	Used where Security is needed but lacks the power storage, computational that is necessary for WSNs.	propose and compare three levels of GF(2^{163}) ,GF(2^{193}) and GF(2^{256})	Montgomery Ladder algorithm, Matlab scripts are utilized
Xining Cui, Jing ei Yang[18] 2012	Karatsuba Algorithm	Longer the keylength, better speedup ratio.	Application Specific Circuit	LUT Utilization is about 54% of the device
Thapliyal et al.,[19] 2005	Ancient indain vedic mathematics	Good device utilization	Only squaring architecture	Implemented for very less number of bits
Hilal Houssain, Turki.F. Al-Somani[20] 2012	Double and Add Algorithm	Good speed and moderate area usage is achieved.	Suitable for resource limited devices, with very low level security	- Implementation on Nano FPGA, results in limited number of resources available. - Nano FPGA cannot exceed the values of bitsize 11. - Low level of security
Hassan et al., [21] 2010	Montgomery Lopez Dahab	Implemented on picoblaze MC Highest field m=571	H/W co-design not complete hardware, Data width is 8/16/32 bits	Scalable with m=163,191 for different datawidth Is implemented
Talapatra et al., [22] 2010	PB LSE first MM architecture for ECC over GF(p)	Area complexity is significantly low	Only multiplication with CSR in Prime field,	Implementation concentrates on ASIC design
Leboeuf et al., [23] 2013	MM algorithm for the GPU's SIMD architecture,	Attained very high throughput	heavily optimized,Field Field size 112 to 521 bits, Prime field implementation	could be used as component of an EC cryptography acceleration appliance or cryptanalysis
Sutter et al., [25] 2013	Digit-serial approach in GF multiplication	High-performance architecture	shows the scalability of architecture using less area and division, projective	Only point multiplication is carried out

			coordinates	
Pontie et al.,[28] 2014	Window method Window-NAF method	good computation time and area tradeoff	Different approach with windowing techniques	Only PM implementation, more hardware
Our work 2015	Montgomery algorithm	Encryption and Decryption is 1.091ms which speed up the process and avoids attacks.	(i). Selection of FPGAs (ii). Improvement in (iii). Less device utilization (iv). Speed up during encryption and decryption process	FPGAs selected are expensive

F. Elliptic Curve Arithmetic

The Elliptic Curve Arithmetic is based on a Finite Field of characteristic 2 and extension degree m: GF(2^m), There are a number of bases known for GF(2^m). The most common bases, permitted by the leading standards (IEEE 1363 and ANSI X9.62) concerning ECC are polynomial bases and normal bases. The representation considered in this paper is a normal base, that is most appropriate for hardware implementation. An elliptic curve over GF(2^m) is def ned as the cubic equation E:

$$y^2 + xy = x^3 + ax^2 + b \quad (1)$$

Where a,b,x,y ∈ GF(2^m) and b ≠ 0. The Points of the Elliptic Curve (E) are set of solutions obtained with equation $(x, y)|y^2 + xy = x^3 + ax^2 + b$. Points can be generated by def ning an suitable addition operation and the Point at Inf nity, which is an extra point O, an additive, abelian group with O the neutral element. Points on an elliptic curve are represented in terms of their coordinates P(x, y). Elliptic curve arithmetic involves point addition and point doubling. Addition of two points P (x1, y1) and P (x2, y2) on a curve yield another point P (x3, y3) on the curve, which are obtained through Equation (2) and Equation (3).

$$x3 = \left(\frac{(y2-y1)}{(x2-x1)}\right)^2 + \left(\frac{(y2-y1)}{(x2-x1)}\right) + (x1 + x2) \quad (2)$$

$$y3 = \left(\frac{(y2-y1)}{(x2-x1)}\right) + (x1 + x2) - y1 \quad (3)$$

The point doubling P(x1, y1) yield another point on the curve P(x3, y3), which are computed through Equation (4) and Equation (5).

$$x3 = \left(x1 + \frac{x1}{y1}\right)^2 + \left(x1 + \frac{y1}{x1}\right) \quad + a2 \quad (4)$$

$$y3 = x3 \left(x1 + \frac{y1}{x1} + 1\right) + x1^2 \quad (5)$$

The Elliptic curve point multiplication is computed by repeated point addition such as P + P +.... + P = k.P = R with k ∈ N and P, R ∈N.

The elliptic curve points can be countable and can be added together. Addition of any two points on the elliptic curve results in a third point on the same curve. This process is called as point addition. Point doubling is def ned as adding a point with itself. The points along with the addition operation form a elliptic curve group.Group members are responsible for selecting the points for Elliptic Curve Cryptographic Implementations, which results in the formation of a f nite f eld. So groups and f nite f elds are two vital concepts required for the Elliptic Curve Cryptographic implementation.

IV. NETWORK MODEL

Consider a WSN Cryptography task consisting of a base station and limited mobile nodes which are stable in network. A typical sensor node with an 8 bit processor with frequency between 4-12 MHz, ROM, Flash memory, limited RAM, two AA batteries, an RF module complaint to the IEEE 802.15.4. Power limitation of sensor nodes constrains the 8-bit or 16-bit microcontrollers that consume low power but have restricted memory resources and low processing capabilities. The probability of communication failures in nodes are more in WSNs, than traditional networks as nodes are often located in unattended places and they use a limited power supply. When they are using cryptographic communication, they consume much more energy. In our proposed approach, the computation time, energy and hardware implementation of cryptographic applications are considered, to increase the average network lifetime.

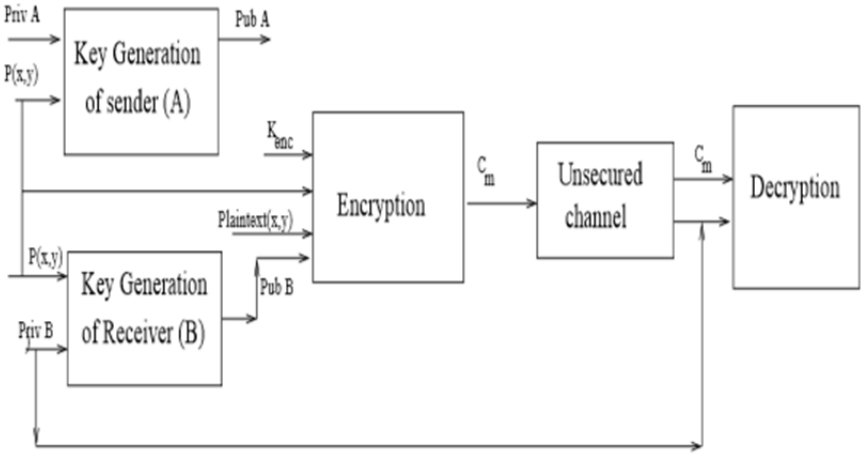

Fig1. Block diagram of Network model

The proposed scheme is implemented in four phases.(i) Elliptic Curve Parameter Selection (ii) Key Generation (iii) Encryption (iv) Decryption Figure 1 shows the complete model in which the first step is to generate the keys. Both public key and private keys are generated by the sender and the receiver. The public key of the sender is published over the network, which is used for the decryption process. The encryption process utilizes the private key of the sender, provides enhanced security for data through digital signature and authentication.

With the aim of achieving essential requirements of security area and speed constraints, ECC algorithm is implemented using hardware. A high degree of flexibility with respect to the cryptographic algorithms is desirable in WSNs. Best solutions can be obtained by combining high flexibility with speed for different data widths and traditional hardware for physical security. It is implemented with cryptographic algorithms for FPGAs which are reconfigurable devices. FPGAs are hardware devices that can be programmed systematically. FPGA implementation can be simply upgraded to integrate a few protocol changes. Whereas ASIC, necessitates for expensive and time consuming physical design, fabrication and testing.

V. ALGORITHM

A. *Montogomery Algorithm*

In our design, point multiplication is implemented using Montgomery point multiplication algorithm. Scalar or point multiplication over an equation (1) elliptic curve is carried out in three steps.

- Finite Field Arithmetic
- Point Addition and Point Doubling
- Scalar or Point Multiplication

In this algorithm, the point multiplier is combination of point adder, doubler, squarer and inverter as shown in Figure 2. Pseudo code is given in Algorithm 1. In Figure 2, Point multiplier consists of point adder, point doubler and conversion module. The point multiplier diagram is shown in Figure 3. Advantages of Montgomery algorithm are: (i) on an average involves less number of field multiplications compared to the traditional method. (ii) since projective coordinates are utilized instead of affine coordinates, inversion is performed at coordinate transformation step. (iii) it is secure against side channel attack. Therefore, Montgomery scalar multiplication algorithm is employed in this work for performing multiplication.

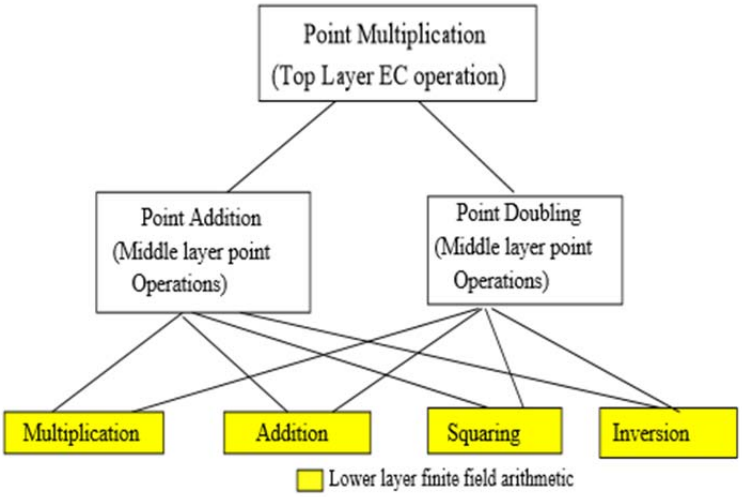

Fig.2 ECC Hierarchy

B. *Montogomery Multiplication*

Montgomery Multiplication algorithm is a very elegant and competent algorithm for manipulating the modular multiplication. A shift and modulus-addition operation, replaces the division operation, which are faster. The implementation of Montgomery multiplication involves the tradeoff between chip area and computational speed and appropriate for hardware implementation. Montgomery has a regular behaviour in execution of sequence of operations, irrespective of the value of bit of key. Performing the same operations over and over again is one of the advantages and this customary behaviour adopts it to be more resilient to attacks like side channel attacks, time and power analysis attacks. It is used in computation of the private keys by point multiplication and a scalable architecture that can be conf gured to attain the design tradeoff i.e., area-time. Xining et al.,[18] along with double and add algorithm utilized Montgomery algorithm, their implementation is compared with our work in Table I and Table VI.

C. *Montogomery Point Multiplication*

There exists several algorithms for point multiplication over elliptic curve. The Montgomery point multiplication algorithm is used in our work. Point multiplication is all about the computation of kP, where k is an integer and P i.e., P(x,y) is a point on an elliptic curve E given by equation (1) and def ned over a f eld GF(2m). The execution time of Elliptic Curve Cryptographic Schemes is dominated by Point multiplication, which is also called Scalar Multiplication. The security of ECC is completely based on the stability of Elliptic Curve Discrete Logarithmic problem (ECDLP). The ECDLP f nds the value of k, given (P, Q) points on the def ned curve, k is scalar value, where Q=kP, and P is a p(x, y), a point on def ned Elliptic Curve.

D. *Point Multiplier*

The point addition and doubling are the two important operations of the Elliptic Curve Cryptography that decides its performance. This point addition requires mainly two points (i) one is in projective coordinate and (ii) other point aff ne coordinates. Finally, the results are in the projective coordinates.

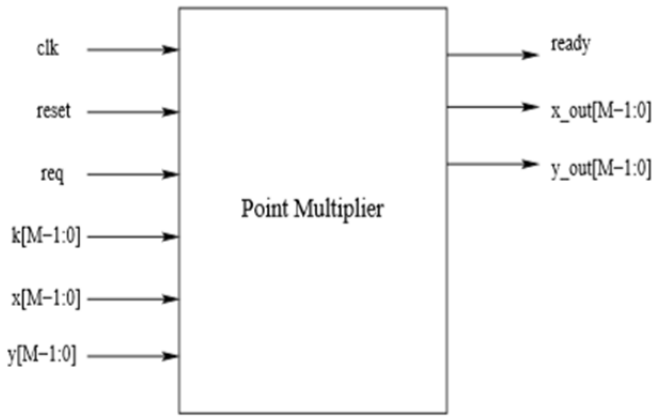

Fig.3 Point Multiplier

E. *Projective Coordinate System*

An elliptic curve def ned by equation (1) comprises of the inf nity point O and a set of points in the aff ne coordinates (x, y) for x, y a f nite f eld GF(2^m).A point on an elliptic curve can be represented in a projective coordinate system in the form (x, y, z). The Point addition in which p1=(x1, y1, z1), p2=(x2, y2, z2) and the resultant point p3 =p1 + p2. Point doubling is 2p1=(x3 , y3, z3). Point addition and doubling are signif cant operations in ECC algorithm. They are the construction blocks for scalar multiplications that are required for all ECC schemes. These addition and doubling operations in aff ne coordinate system demand modular inversion operations. The modular inversions are much more luxurious than other operations such as modular multiplications, which is not appropriate for wireless sensor nodes and resource forced devices.

By means of projective coordinate system, modular inversions can be replaced with a small number of modular multiplications and squares operations. As a result of this, the execution times of point addition and doubling based on projective coordinate system are faster. Use of Optimized ECC involves mixed point addition algorithm, which purely adds a point in projective coordinate and a second point in aff ne coordinate. To further reduce the number of modular multiplications and squaring operation this algorithm can be used in scalar multiplications, resulting in smaller and faster code.

The main purpose of our design is to optimize the parallel processing of Montgomery point multiplication and reduce chip area by using (i) Point doubler (ii) Point Adder and (iii) co-ordinate converter. The block diagram of a Point Multiplier is shown in Figure 3. Derived from Montgomery's point multiplication algorithm, the point multiplier is made up of point adder, point doubler, coordinates converter, squarer and XORs. The data f ow of point adder, doubler and coordinate converter are shown in Figure 3, Figure 4 and Figure 5 respectively, where T1, T2, T3 and T4 are registers to store intermediate results.

1) Point Doubler: The inputs of point doubler are X1, Z1 and X, Z is the obtained output after a series of f eld arithmetic operations. Figure 4 depicts four f eld squaring, two f eld multiplications and one simple f eld addition. The pseudocode is given in Function 1.

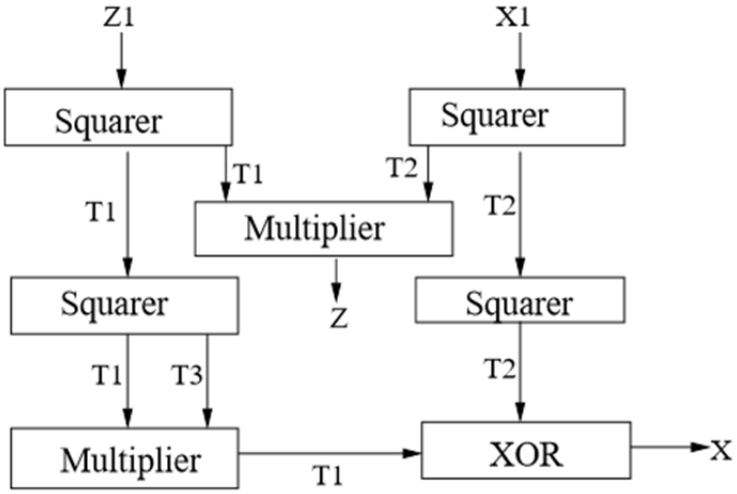

Fig.4 Point Doubler

Function 1 : Point Doubling Pseudocode

function{**MPdouble**}{**X1, Z1**}
{
$T1 = Z1^2$
$T2 = X1^2$
$Z = T1.T2$
$T1 = T1^2$
$T1 = b.T1$
$T1 = T1^2$
return (X, Z)

}
end function

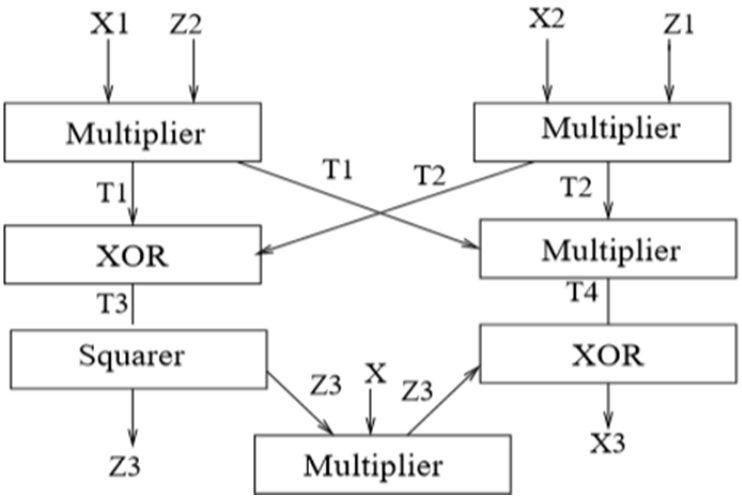

Fig.5 Point Addition

2) Point Adder: The inputs to point adder in projective coordinate are (X1, Z2), (X2, Z1) and (X3, Z3) are the obtained outputs after a series of f eld arithmetic operations. Figure 5 depicts four multiplications, two additions and one squaring. The pseudo code is shown in Function 2.

Function 2 : Point Addition

---

function{**MPadd**}{**X1, Z1, X2, Z2**}
{
T 1 = X1.Z2
T 2 = Z1.X2
T3 = T 1 + T 2
$Z3 = T\ 3^2$
T 3 = Z3.x
T 4 = T 1.T 2
X3 = T 3.T 4
return (X3, Z3)

}
end function

---

3) Coordinate Converter: The coordinate converter maps the four outputs in projective coordinates (X1, Z1, X2, Z2) into aff ne coordinate (xk, yk). The previous resources used for point addition and doubling, are utilized eff ciently to calculate conversions. Note that there is only one inversion in realization of coordinate converter as shown in Figure 6. The pseudocode is given in Function 3.

Function 3 : Coordinate Converter

---

function{**MPxy**}{**X1, Z1, X2, Z2, y**}
{
Invert(Z1, Z2, x)
Multiplication(X1, Z1, X2, Z2, x, y)
Square(x)
Addition(X1, Z1, X2, Z2, x, y)
return (xk, yk)
}
end function

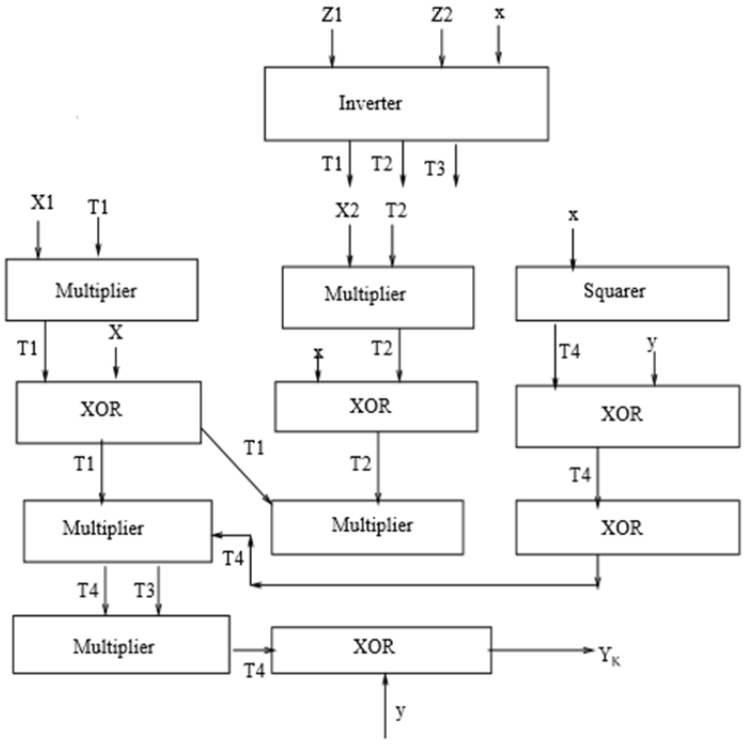

Fig.6 Coordinate Converter

Algorithm 1 Montgomery Algorithm for Scalar Multiplication

Input : A point $P(x, y) \epsilon E(GF(2^m)$,Integer $k = (kl - 1, kl - 2,..., k1, k0)$
Output : Point $Q = kP$
coordinates $X1 = x,\ Z1 = 1,\ X2 = x4 + b,\ Z2 = x2$

if $k = 0$ or $x = 0$ **then**
$x = 0, y = 0$
end if

for $i = l - 2$ to 0 **do**

if $ki = 1$ **then**
(X1,Z1) = MPadd (X1, Z1, X2, Z2), (X2, Z2),
MPdouble(X2, Z2)
else
(X2, Z2)= MPadd(X2, Z2, X1, Z1), (X1, Z1),
MPdouble(X1, Z1)
end if
end for
Q = MPxy(X1, Z1, X2, Z2)
return Q =0

Algorithm 1 depicts Montgomery Scalar Multiplication for elliptic curves over binary f elds. In this algorithm MPadd (X1, Z1, X2, Z2) is function for point addition, MPdouble (X1, Z1) is a function for point doubling and MPxy(X1, Z1, X2, Z2) is a function for conversion of projective coordinates to aff ne coordinates. The algorithm 1 is used to perform point addition over the binary f led using Lopez Dahab mixed coordinates, since it uses two points one is of aff ne coordinates and second point is of projective coordinates.

VI. IMPLEMENTATION

This section presents the design and key parameters considered for the hardware realization using Montgomery approach.

A. *Design of Main Unit*

The ECC algorithm was coded and modeled using VHDL and synthesized on different FPGA devices. As an accelerator/ application specif c processor, the main unit of ECC works like application software controlling data f ow, computation and the state of the processor. ECC algorithm works faster in FPGA than in software implementation. The design of main unit is followed using Finite State Machine (FSM).

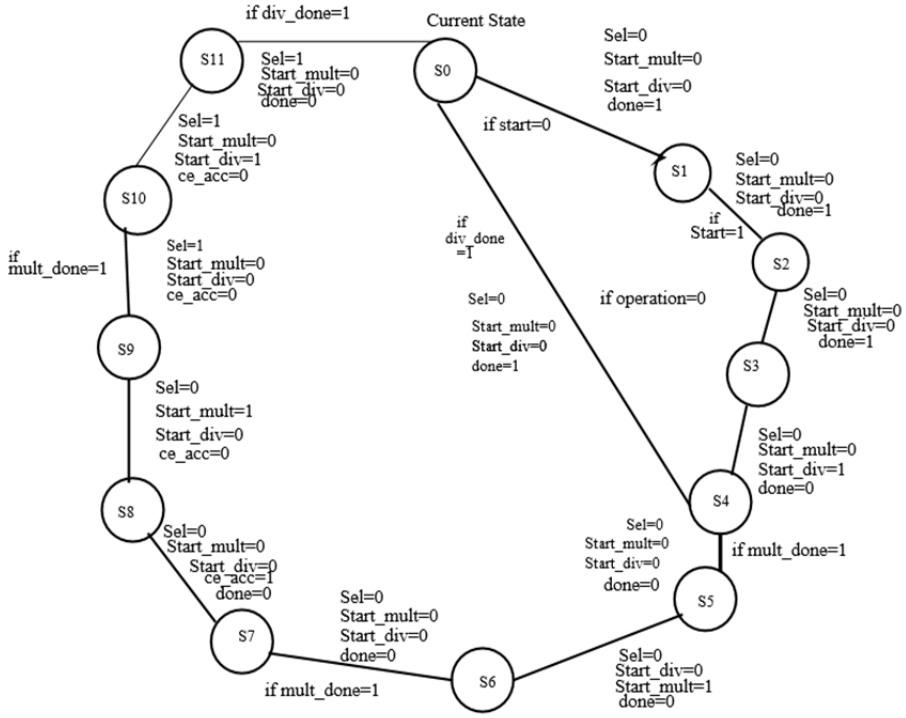

Fig. 7. FSM Flow Diagram

B. *Encryption Unit*

The sender chooses a scalar value which is a prime number as his private key Kenc, for public key generation shown in network model in Figure 1. Using the Equation (1), the elliptic curve points are generated, out of those one point will be selected as generator point p(x,y) and this point is multiplied with the Kenc to generate pubic key of the sender i.e., Ka. This calculation is called point multiplication and given in Equation(6). In the same way the public key Kb of the Receiver is calculated in our implementation.

The encryption unit receives inputs plaintext (x, y), a generator point on the curve P(x, y), key for encryption Kenc and a public key of a receiver. This unit ouputs Cm, ciphertext (x, y) and public key of sender Ka.

In order to perform encryption of the message to obtain cipher text Cm, the private key of sender is multiplied with the public key of the receiver and the product is added to the message M. For example, in our implementation for 11bit data and key, the plaintext i.e., point on curve p(x,y) is choosen as p(ode, ode), and Kenc is 6ad. The cipher text obtained is Cm=(0336,0336). The above procedure is repeated for 112, 131, 163 bits.

$$P(x,y)Kenc = Ka \quad (6)$$

$$M + Kenc * Kb = Cm \quad (7)$$

C. *Decryption Unit*

The decryption unit receives inputs ciphertext (x, y) public key of the sender, a generator point on the curve P(x, y), key for decryption Kdec and a public key of a receiver. This unit outputs plaintext (x, y). For example : the cipher text is Cm=(0336, 0336), the key for decryption Kdec is 6ad, the plaintext obtained is p(ode,ode) which is calculated using Equation (8). The above procedure is repeated for 112, 131, 163 bits.

$$Cm - Kdec * Ka = M \quad (8)$$

VII. PERFORMANCE EVALUATION AND ANALYSIS

The FPGA approach is measured as one of the widely used prototyping environments. The FPGA solutions are considered more than a validation step with their adaptability to the re-use concept.

Figure 8 shows Schematic Diagram of Main Unit which consists of both encryption and decryption of implemented VHDL code using Xilinx on FPGA. The Sections B and C describe the operation of modules.

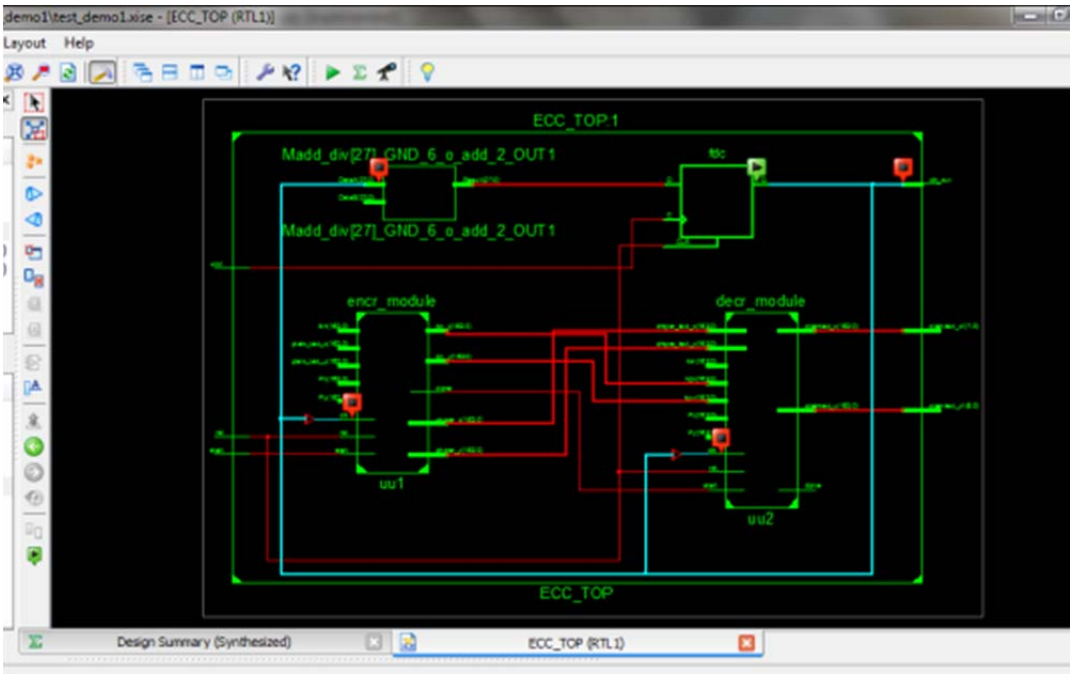

Fig 8. Schematic Diagram of Main Unit

A. *Simulation Setup*

The FPGA devices used in this work are Spartan 3 XC3S400-5pq208, Spartan 6 XC6s1X150t-3csg484, Virtex 4 xc4vlx200-11ff11513, Virtex 5 xc5vlx330t-2ft1738, Virtex-6 XC6vlx760-2ff1760, Artix 7 xc7a100t-3csg324. The Artix 7 xc7a100t-3csg324 is chosen for hardware implementation with Nexys4 board. The Nexys4 can host designs ranging from introductory combinational circuits to powerful embedded processors with its large, high-capacity FPGA, Xilinx part number XC7A100T-1CSG324C, liberal external memories, Ethernet, collection of USB and other ports. It consists of several built-in peripherals, including temperature sensor, an accelerometer, MEMs digital microphone, a speaker amplif er. A number of I/O devices allow the Nexys4 to be utilized for a broad range of designs without the requirement of any other components. The Artix-7 FPGA that is considered for this work is optimized for high performance logic, higher performance and more resources than earlier designs. Artix-7 100T features comprise 15,850 logic slices, each logic cell with four 6-input LUTs and eight FFs, 4,860 Kbits of fast block RAM, six clock management tiles with Phase-Locked Loop (PLL), internal clock speeds exceeding 450MHz, On-chip analog-to-digital converter, 240 DSP slices.

B. *Simulation Results*

Figure 9, 10, 11 and 12 show the simulation results with Atrix 7-Xc7a100t-3csf324. The parameters observed in the results are plain-text with 11 bits in Figure 9, 112 bits in Figure 10, 131 bits in Figure 11, 163 bits in Figure 12 and the key for encryption kenc and decryption kdec along with cipher text and plain-text. The timing diagram for corresponding data and key sizes are shown in Figure 9, 10, 11 and Figure 12 with same size input of plain-text and key size. The minimum input arrival time before clock and maximum output required time after clock for 11, 112, 131 and 163 bits are 2.286η seconds and 3.307η seconds, 2.354η seconds and 3.065η, 2.364η seconds and 3.083η seconds, 3.967η seconds and 4.797η seconds respectively. The synthesis results of 163 bits are shown in Figure 13.

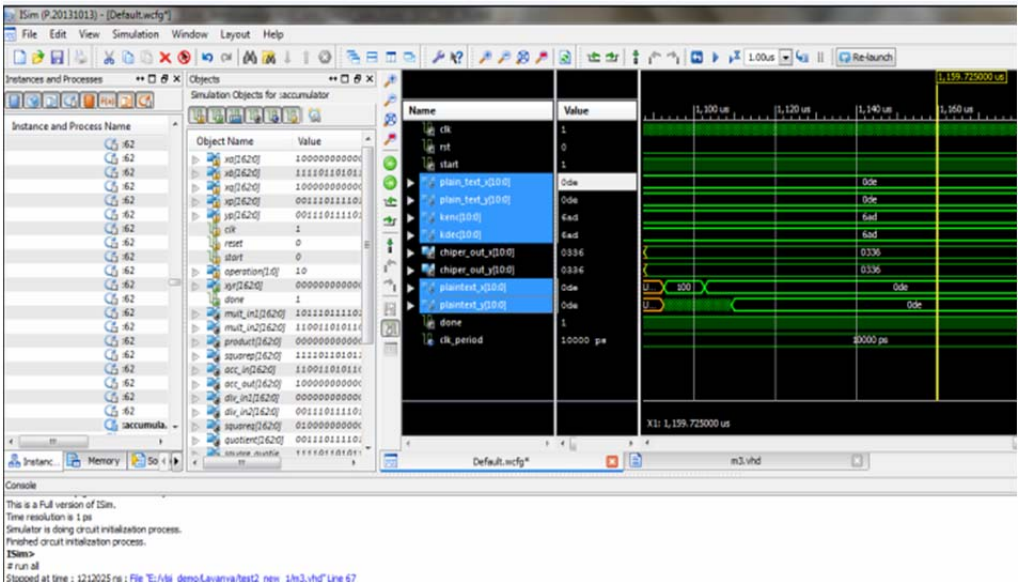

Fig 9. Timing Diagram for Data and Key of 11bits

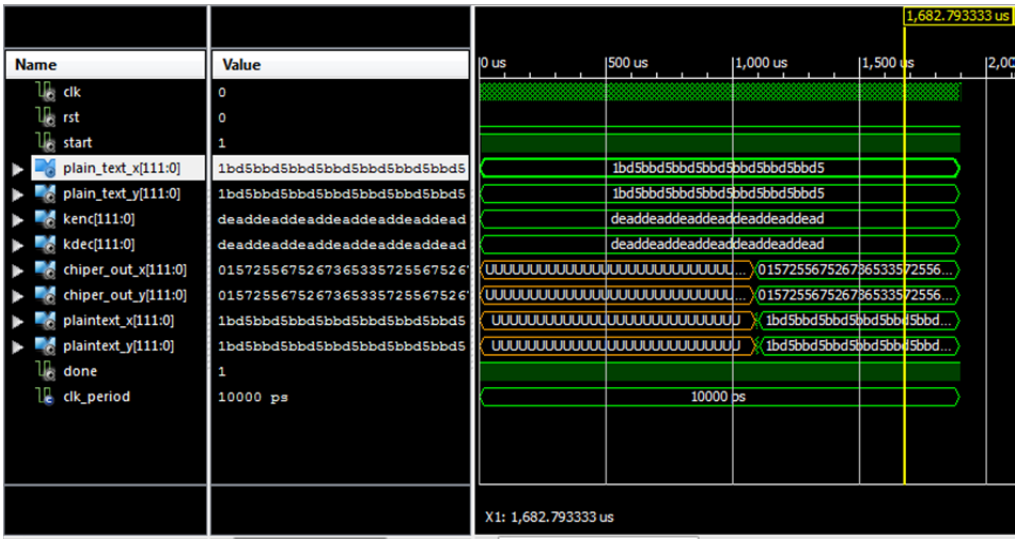

Fig 10. Timing Diagram for Data and Key of 112bits

Fig 11. Timing Diagram for Data and Key of 131bits

Fig 12. Timing Diagram for Data and Key of 163 bits

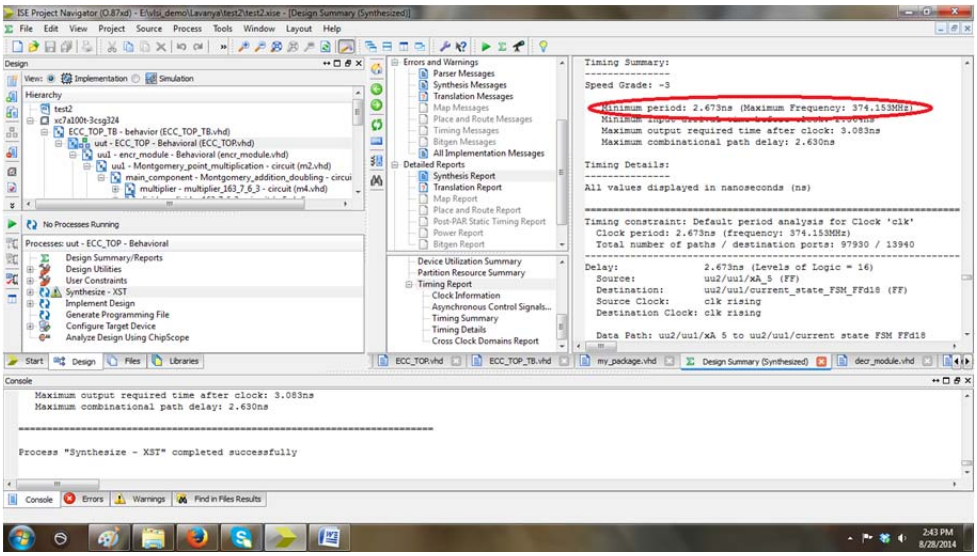

Fig 13. Synthesis Results of 162 bits

C. *Computational time*

Figure 14 gives the details of computational time with Atrix 7-Xc7a100t-3csf324 device. The encryption and decryption time with software and hardware implementation are compared. From graph as shown in Figure 14, it is experiential that the time required with software implementation is more than 2.5 milliseconds and it increases as the the number of input bit size increases. Whereas with hardware implementation the computational time is around 1.091 ms, for all input data size considered.

From Table II, Figure 14 and Figure 15, it is observed that the time required for the software implementation is more than 75% compared to the Hardware implementation. As the input data and key sizes increases i.e., from 11 bits to 163 bits the time increases from 2.5 milli seconds to 4.5 milliseconds. In comparison with previous results of [18][19][20] [21], our implementation shows around 70% of improved software and hardware results with device utilization and computational time, maintaining input data and key sizes same.

TABLE II. COMPUTATIONAL TIME

Bit Size	Software		Hardware	
	Encryption (ms)	Decryption (ms)	Encryption (ms)	Decryption (ms)
11	4.37	4.4	1.083	1.091
112	3.94	3.96	1.085	1.091
131	4.06	4.08	1.085	1.09
163	2.493	2.5	1.086	1.089

TABLE III. COMPARISON VALUES OF DEVICE UTILIZATION WITH DIFFERENT FPGA

Bit Size	Spartan-6	Virtex-4	Virtex-5	Virtex-6	Atrix-7
11	0%	1%	0%	0%	0%
112	3%	7%	2%	0%	4%
131	3%	8%	3%	0%	5%
163	4%	12%	4%	1%	6%

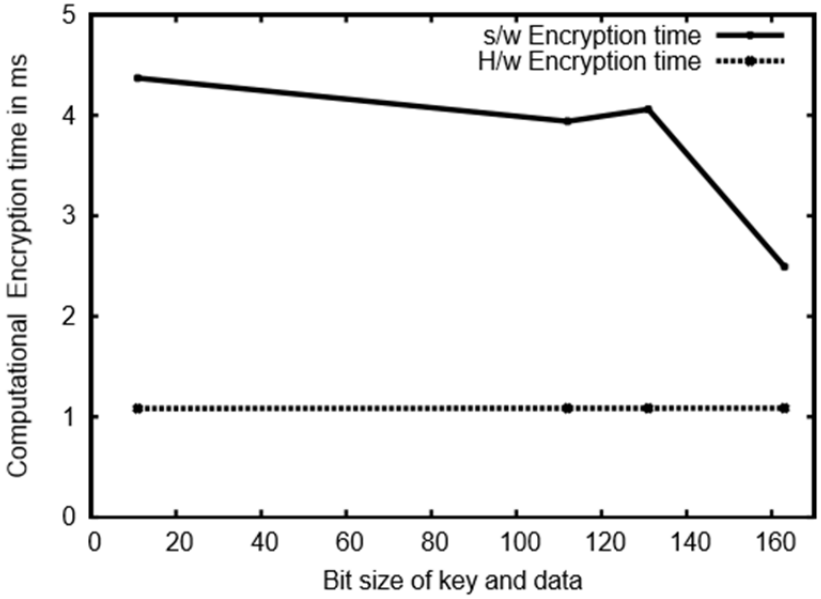

Fig 14. Computational Encryption Time

Table III and Figure 16 shows the number of slices utilized for encryption and decryption operation with different key and data sizes. There is variation in device utilization with respect to the devices chosen i.e., Artix 7, Virtex 7, Virtex 5,Virtex 4 and Spartan 6. The overall percentage utilization of area is from 0% to 13%, i.e., is the improved results compared with previous implementations of [18][19][20] [21]. Here, 0% indicates very low amount usage of devices. From the results, Table III and Table IV, the designer can select the desired device for implementation. The simulation results shown in Table IV and V gives detailed information about Utilization of Slices for different bits of data and key, along with encryption and decryption times.

D. *Performance Analysis*

From the table VI, it is observed that with software implementation, time consumption for encryption and decryption is more in [18] compared to our work. In Xining Cui et al., [18], as the number of data input bit size increases, the computation time increases, whereas in our implementation, it varies slightly with software implementation and almost remains constant with hardware implementation. It also shows that energy consumption required is less in our work in comparison with [18]. Level of security provided is good as data and key size is same. The work designed is most suitable for constrained environments such as Wireless Sensor Networks.

It is observed that from the graphs shown in Figure 17 and Figure 18 there is a great improvement in-terms of computational time both in hardware and software in comparison with [18]. The data-size is varied from 16 bits to 512 bits, with key size of 163 bits. As the data size increases, the software encryption time and decryption time also increases, whereas in our work, as the data and key size increases, there is a slight variation in the encryption time and it is repeated with software decryption time. The Hardware implementation shows same variation with bit-size maintaining constant key-size. The computational time remains constant for both encryption and decryption that avoids the attacks.

TABLE IV. Comparison Values of Frequency, Device Utilization and Computational Time for Different Bit Sizes

Device	Bit Size	Clock(MHz)	Time(ηsec)	Available slices	Used	Usage(%)	Encryption time(ms)	Decryption time(ms)
Artix7	163	229.537	4.35	126800	8847	6%	2.493	2.50
	131	374.153	2.673	126800	6654	5%	4.06	4.08
	112	362.884	2.480	126800	5722	4%	3.94	3.96
	11	403.177	2.480	126800	846	0%	4.37	4.40
V irtex6	163	232.769	4.296	948480	8858	1%	2.52	2.54
	131	357.932	2.94673	948480	6668	0%	3.88	3.91
	112	369.297	2.708	948480	5722	0%	4.01	4.03
	11	411.074	2.433	948480	846	0%	4.46	4.49
V irtex5	163	183.293	5.456	207360	8849	4%	1.99	2.01
	131	294.825	3.392	207360	6664	3%	3.20	3.22
	112	300.879	3.324	207360	5732	2%	3.26	3.28
	11	356.494	2.805	207360	857	0%	3.87	3.89
V irtex4	163	124.953	8.003	89088	11259	12%	1.35	1.36
	131	184.142	5.431	89088	7921	8%	1.99	2.01
	112	205.508	4.866	89088	6510	7%	2.23	2.24
	11	248.201	4.029	89088	969	1%	2.69	2.71
Spartan6	163	127.476	7.845	184304	8941	4%	1.38	1.40
	131	201.118	4.972	184304	6726	3%	2.18	2.20
	112	201.351	4.966	184304	5816	3%	2.18	2.20
	11	212.555	4.705	184304	864	0%	2.30	2.32

TABLE V. Comparison Values of Frequency, Device Utilization and Computational Time for Different Bit Sizes

Device	Bit Size	Clock(MHz)	Time(ηsec)	Available slice LUTs	Used	Usage(%)
Artix7	163	229.537	4.35	63400	15114	23%
	131	374.153	2.673	63400	12461	19%
	112	362.884	2.480	63400	8564	4%
	11	403.177	2.480	63400	1295	0%
V irtex6	163	232.769	4.296	474240	15162	3%
	131	357.932	2.94673	474240	9715	2%
	112	369.297	2.708	474240	11380	2%
	11	411.074	2.433	474240	1427	9%
V irtex5	163	183.293	5.456	207360	16245	7%
	131	294.825	3.392	207360	11004	5%
	112	300.879	3.324	207360	9450	4%
	11	356.494	2.805	207360	1419	0%
V irtex4	163	124.953	8.003	171816	20691	11%
	131	184.142	5.431	171816	14862	8%
	112	205.508	4.866	171816	12137	6%
	11	248.201	4.029	171816	1809	1%
Spartan6	163	127.476	7.845	92152	15728	17%
	131	201.118	4.972	92152	1269	13%
	112	201.351	4.966	92152	12417	13%
	11	212.555	4.705	92152	1433	1%

TABLE VI. Comparison of Computational Time values of Xiniing Cui et al.,[18] with Our Work

Device	data Size	key size	s/w Enc Time(ms)	s/w Dec Time(ms)	H/w Enc time(ms)	H/w Dec time(ms)
	Xiniing Cui et al.,[15]					
XC4V LX60	16	163	757.385	764.530	1.370	1.378
	32	163	1484.888	1144.982	1.776	1.574
	64	163	3019.725	1916.224	2.526	2.004
	128	163	5345.684	3032.338	3.769	2.615
	256	163	9922.824	5334.276	6.189	3.881
	512	163	19654.690	10234.911	11.363	6.5581
	Our Implementation					
Artix7	11	11	4.37	4.4	1.083	1.091
Xc7a100t − 3csf 324	112	112	3.94	3.96	1.085	1.091
	131	131	4.06	4.08	1.085	1.09
	163	163	2.493	2.5	1.086	1.089

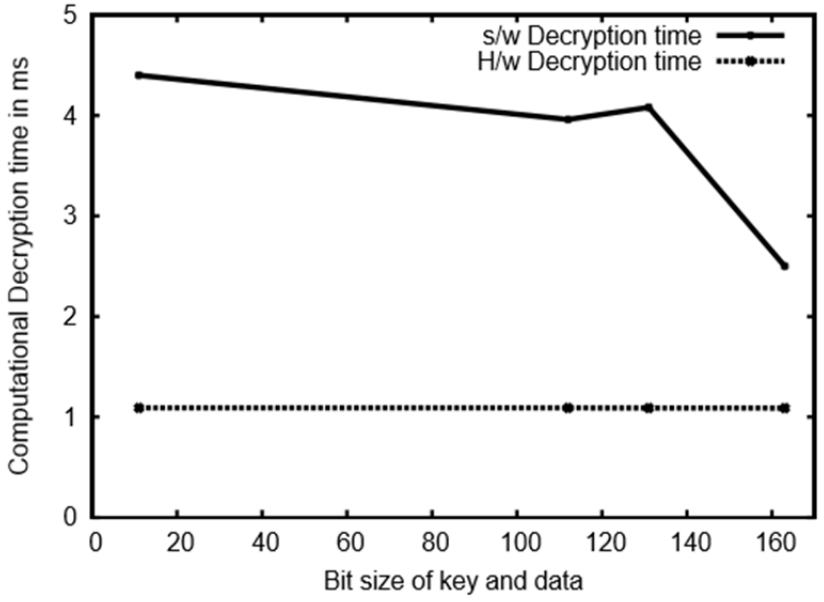

Fig.15 Computational Decryption Time

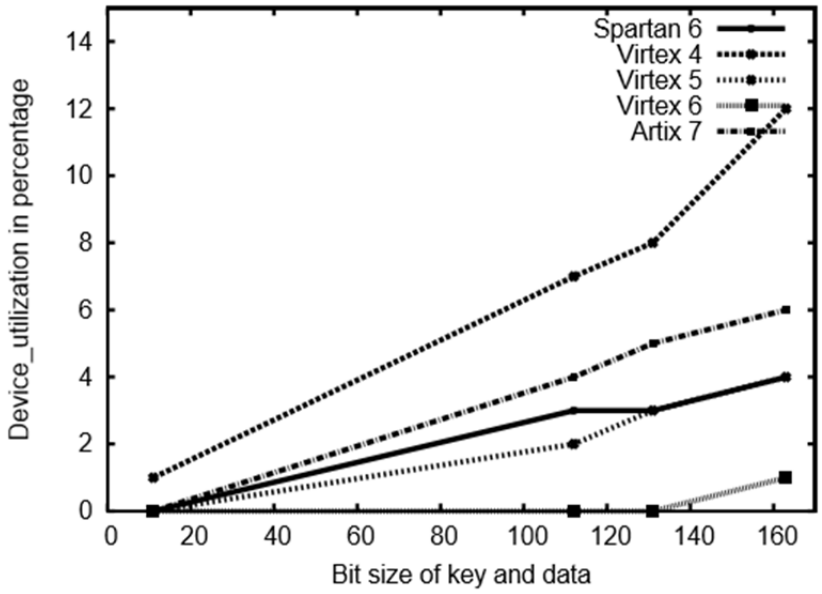

Fig. 16 Device Utilization

The proposed design gives a extensive reduction of around 70% computational delay time in comparison with earlier designs [18][19][20][21]. Enhancing the speed of most time critical part of Elliptic Curve (EC) Crypto schemes i.e., Point multiplication, enables the use of this method within combined Hardware/Software systems with reasonably 75% low computing time and 13% area utilization. Consecutively, the ECC algorithms in software facilitate algorithmic f exibility and at the same time the requisite performance is contributed by dedicated coprocessors. Whatever the value of scalar k in binary (0,1) point addition and doubling is performed in both hardware and software implementation. In all the iterations, that depends on the key size and key value in binary, the process of operation involved for binary 0 and binary 1, is similarly visible to an attacker who performs a side channel attack. On analysis of power and time consumption by the adversary, useful information about k cannot be obtained. Brute force attack is not possible as the key size used is large i.e., 11 bits to 163 bits. In Brute force attack, the attacker keeps checking bits patterns of particular key size in trial and error basis. If the bit size is n there are 2n patterns needed to be checked. Moreover, in our implementation we have used both data and same key size same which puts attacker in more effort to decrypt and long time investment. The above results reveals that the speed and area tradeoff can be obtained by selecting suitable FPGA device. An extensive speed-up of Elliptic Curve cryptosystems can be achieved for the nodes in Wireless Sensor Networks by exploiting the offered coprocessor architectures. The designed cryptosystems utilized 2%, 13%, 19%, 23% of arithmetic Look Up Tables (LUT) of the Artix 7-Xc7a100t-3csf324for 11, 112, 131 and 163 bits respectively.

The software and hardware results reveals that the computational time for both encryption and decryption is almost the same for different bit sizes of both data and key. This constant time puts an attacker in confusion about the bit sizes used for data and key as shown in Table II. Thus, using of same data and key sizes avoids many of the attacks including power analysis attack. Since, the energy consumed is same as that of the time required to perform both encryption and decryption process of 0 and 1 bits. The attacker is unable to know whether 0 or 1 is encrypted or decrypted in that duration of time. The computational time for encryption and decryption in hardware is in the order of 1.091ms.

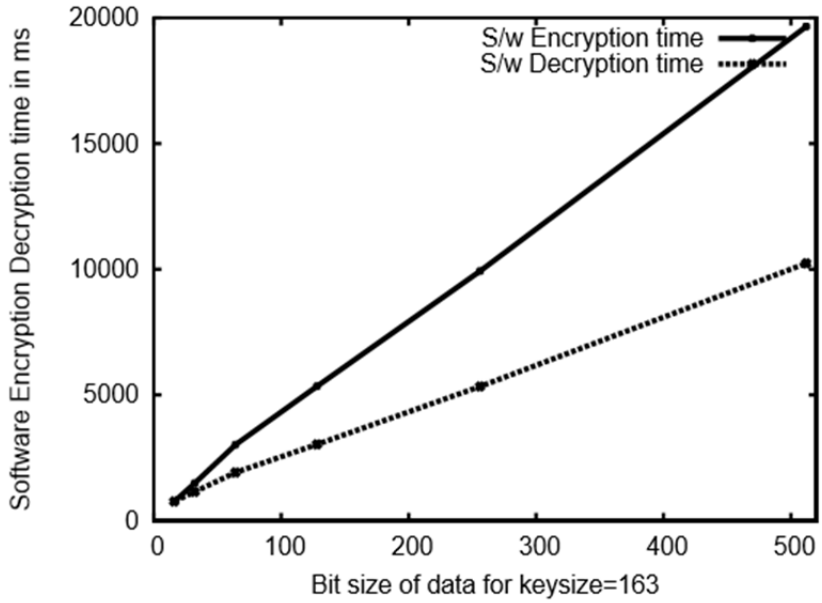

Fig 17. Xining Software Computation Time

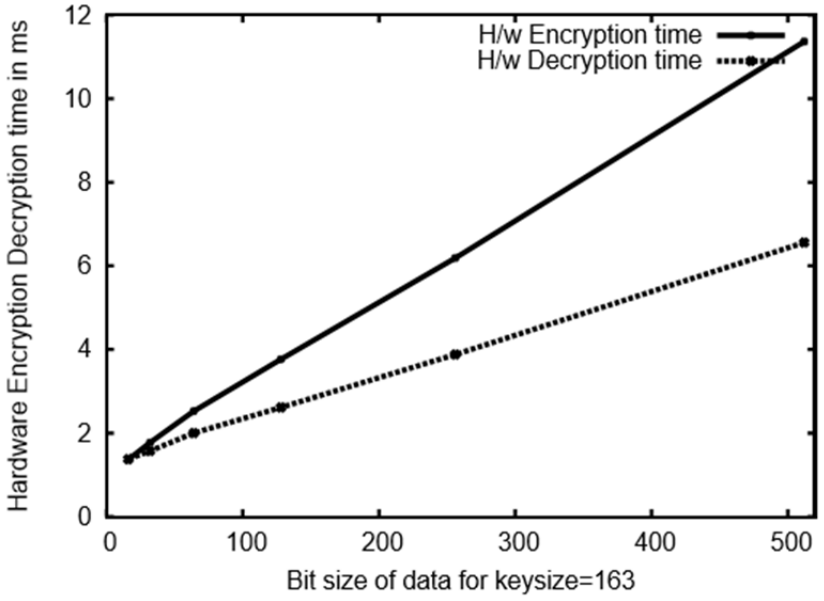

Fig 18. Xining Hardware Computation Time

The wireless sensor nodes based on off-the-shelf general purpose microcontrollers, have utilized by most deployments of wireless sensor networks over the past years. The hardware accelerators for custom system architecture are required to characterize WSNs applications to achieve long deployment lifetime and bursts of computation. The work and design adopts the accelerator-based computing paradigm, together with acceleration for the process of Encryption and decryption of data in WSNs.

VIII. CONCLUSIONS

Although realization of Public Key Cryptography (PKC) is challenging on WSN, PKC brings enormous simplicity and effectiveness in providing a number of necessary security services. Public Key Cryptography algorithms are the most promising schemes with respect to energy and time consumption, which makes it suitable for data encryption in WSN. For WSNs, we have selected Elliptic Curve Cryptography to achieve the security in spite of constraints of the WSN node. The proposed work deals with the different level of security with different FPGA devices to make it suitable for WSNs. In order to evaluate, analysis was performed on different key sizes in $F(2^m)$ and implemented on different FPGAs, with the aim to recognize the one that is most appropriate for WSNs applications. The key size with 11 bits to 163 bits, were generated to provide low level to elevated level of security that can be used for various applications in Wireless Sensor Networks. Both software

and hardware results shows minimum of 13% utilization of the area and device. The time required for encryption and decryption is very low i.e., 50% less compared to the previous implementations in [18][19][20] [21]. Further, we have planned to focus on different techniques for multiplication using Elliptic Curve Cryptography.

REFERENCES

[1] Ian F Akylidiz, Weilian Su, Yogesh Sankarasubramaniam and E Cayirci, "Wireless Sensor Network : A Survey on Sensor Networks," in IEEE Communication Magazine, ISSN:0163-6804, vol. 40, no. 8, pp. 102-114, 2002.

[2] F Amin, A Jahangir and H Rasifard, "Analysis of Public-Ke y Cryptog-raphy for Wireless Sensor Networks Security," in Proceedings of the Conference World Academy of Science, Engineering and Technology, ISSN:1307-6884, vol. 41, pp. 396-402, 2008.

[3] Shaila K, S H Manjula, Thriveni J, Venugopal K R and L M Patnaik, "Resilience Against Node Capture Attack using Asymmetric M atrices in Key Predistribution Scheme in Wireless Sensor Networks ," in Interna-tional Journal on Computer Science and Engineering, ISSN:0975-3397, vol. 3, no. 10, pp. 3490-3502, 2011.

[4] Shaila K, Nalini L, Tejaswi V, Thriveni J, Venugopal K R, L M Patnaik, "Secure QoS-Aware Data Fusion to Prevent Node Misb ehavior in Wireless Sensor Networks," in International Journal of Computer Science and Network Security, ISSN:0975-3397, vol. 11, no. 3, pp. 31-41, 2011.

[5] Lata B T, Vidya Rao, Sivasankari H, Tejaswi V, Shaila K, Venugopal K R, L M Patnaik, "SEAD: Source Encrypted Authentic Data for Wi reless Sensor Networks," in International Journal of Engineering Research and Development, e-ISSN: 2278-067X, p-ISSN: 2278-800X, vol. 11, no. 3, pp. 01-16, 2015.

[6] Ming Lu, "Study on Secret Key Management Project of WSN ba sed on ECC," Journal of Networks, ISSN:1796-2056, vol. 7, no. 4, pp. 652-659, April 2012.

[7] Wang Wei-hong, Lin Yu-bing and Chen Tie-ming, "The Study and Application of Elliptic Curve Cryptography Libray on Wireless Sensor Networks," in Proceedings of Eleventh IEEE International Conference on Communication Technology, e-ISBN:978-1-4244-2251-7, p-ISBN:978-1-4244-2250-0, pp. 785-788, 2008.

[8] Abidalrahman Mohd, HoseinMarzi, NaumanAslam, William Phillips and William Robertson, " A Secure Platform of Wireless Sensor Ne tworks," in Proceedings of the 2nd International Conference on Ambient Systems, Networks and Technologies, ISSN:1877-0509, vol. 5, pp. 115-122, 2011.

[9] Ajay S, Kotresh H, Shruti B S, Swetha G S and Srividya B V, " L ow Power FPGA Based Elliptical Curve Cryptography," in IOSR Journal of Electronics and Communication Engineering, e-ISSN:2278-2834, p-ISSN:2278-8735, vol. 6, no. 2, pp. 11-14, May-June 2013.

[10] S P Arya, A Muruganandhan, " Design and Implementation o f EC based Cryptosystem on FPGA," in International Journal of Advanced Information and Communication Technology, ISSN:2348-9928, vol. 1, no. 2, pp. 235-240, June 2014.

[11] M Bednara, M Daldrup, J Shorkollahi, J Teich and J Von ZurGathen, "Tradeoff Analysis of FPGA Based Elliptic Curve Cryptography," In proceedings of of the IEEE International Symposium on Circuits and Systems (ISCAS-02), ISSN:0-7803-7448-7 vol. 5, pp. 797-800, 2002.

[12] Sjay Sinha Roy and Chester Rebeiro, " Theoretical Model ing of Elliptic Curve Scalar Multiplier on LUT-Based FPGAs for Area and Speed," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, ISSN:1063-8210, vol. 21, no. 5, pp. 901-909, May 2013.

[13] Shylashree N and V Sridhar, " Efficient Implementation o f Scalar Multiplication for Elliptic Curve Cryptography using Ancient Indian Vedic Mathematics over GF(p)," in International Journal of Applied Information Systems, ISSN:0975-8887, vol. 49, no. 7, pp. 46-50, July 2012.

[14] Hilal Houssain, Mohamad Badra and Turki F Al-Somani, " C omparative Study of Elliptic Curve Cryptography Hardware Implementations in Wireless Sensor Networks," in International Journal of RFID Security and Cryptography (IJRFIDSC), vol. 1, no. 1/2, pp. 67-74, March/June 2012.

[15] Kaleel Rahuman and G Athisha, " Reconfigurable Architec ture for Elliptic Curve Cryptography," in Proceedings of the IEEE Interna-tional Conference on Communication and Computational Intelligence, INSPEC Accession number 1188746, pp. 461-466, 2010.

[16] A Kaleel Rahuman and G Athisha, "Reconfigurable Archite cture for Elliptic Curve Cryptography using FPGA," in Mathematical Problems in Engineering, Hindawi Publishing Corporation, Article ID 675161, 8 pages, July 2013.

[17] Hassan M N and Benaissa M, "A Scalable Hardware/Softwar e Co-design for Elliptic Curve Cryptography on PicoBlaze Microcontroller," in Proceedings of 2010 Symposium on IEEE Circuits and Systems (ISCAS), p-ISBN:978-1-4244-5308-5, pp.2111-2114, Paris, 2010.
Xining Cui and Jingwei Yang , "An FPGA Based Processor fo r Elliptic Curve Cryptography," in Proceedings of International Conference on Computer Science and Information Processing (CSIP), p-ISBN:978-1-46733-1410-7, pp. 343-350, Xian, China 2012.

[18] Himanshu Thapliyal and M B Srinivas, "An Efficient Metho d of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics," in Proceedings of the 48th IEEE Midwest Symposium on Circuit and Systems, ISBN:0-7803-9197-7, vol. 1, pp. 826-828, 2005.

[19] Hilal Houssain and Turki F Al-Somani, "Elliptic Curve C ryptoprocessor Implementation on a Nano FPGA:Interesting for Resource-Constrained Devices," in International Journal of RFID Security and Cryptography (IJRFIDSC), ISSN:2046-3715, vol. 1, no. 1/2, pp. 45-51, March/June 2012.

[20] Hassan M N and Benaissa M, "Small Footprint Implementat ions of Scalable ECC Point Multiplication on FPGA," in Proceedings of the IEEE International Conference on Communications (ICC), ISSN:1550-3607, pp. 1-4, Capetown, South Africa, 2010.

[21] Talapatra S and Rahaman H, "Low Complexity Montgomery M ul-tiplication Architecture for Elliptic Curve Cryptography over GF(p)," in Proceedings of 18th IEEE/IFIP Conference on VLSI System on Chip Conference (VLSI-SoC), p-ISBN:978-1-4244-6469-2, pp. 219-224, Howrah, India, 2010.

[22] Leboeuf K, Muscedere R and Ahmadi M, "A GPU Implementati on of the Montgomery Multiplication Algorithm for Elliptic Curve Cryptog-raphy," in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), ISSN:0271-4302, ISBN:978-1-4673-5760-9, pp. 2593-2596, Windsor, Canada, 2013.

[23] Gerardo Orlando and ChristofPaar, "A High-Performanc e Reconfig-urable Elliptic Curve Processor for GF(2^M)," in Proceedings of the Sec-ond International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2000, p-ISBN:978-3-540-41455-1, e-ISBN:978-3-540-44499-2, pp. 41-56, 2000.

[24] Gustavo D Sutter, Jean-Pierre Deschamps and Jos Luis Imaa, "Efficient Elliptic Curve Point Multiplication Using Digit-Serial Binary Field Op-erations," in IEEE Transactions on Industrial Electronics, ISSN:0278-0046, vol. 60, no.1, pp. 217-225, 2013.

[25] J W Lee and H C Chang, "Efficient Power-Analysis-Resista nt Dual-Field Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture," in IEEE Transactions on Very Large Scale Integration Systems, ISSN:1063-8210, vol. 22, no. 1, pp. 49-61, 2014.

[26] S Pontie and P Maistri, "Design of a Secure Architecture for Scalar Multiplication on Elliptic Curves," in Proceedings.of the 10th Conference on Ph.D. Research in Microelectronics and Electronics, INSPEC Accession Number:14516154, pp. 1-4, July 2014.
[27] P R Munoz, V T Olaya and J V Medina, "Design of Elliptic Cur ve Cryptoprocessors over GF(2^{163}) on Koblitz Curves," in IEEE Latin American Symposium on Circuits and Systems, p-ISBN:978-1-4799-2506-3, pp. 1-4, 2014.

AUTHOR PROFILE

Leelavathi G is presently a Research Scholar at Visvesvaraya Technological University-Research Center, Department of Electronics and Communication, Vivekananda Institute of Technology, Bengaluru. She is working as Assistant Professor in the Department of Electronics and Communication Engineering at Govt. SKSJTI, Bengaluru, India. She received her B.E and M.E degrees in Electronics and Communication Engineering from Bangalore University and Visvevaraya Technological University respectively. She has published around 25 papers in National and International conferences. She has received Best paper Awards. Her research interest includes Wireless Sensor Networks and Reconfigurable Embedded systems.

Shaila K is currently Professor and Head in the Department of Electronics and Communication Engineering at Vivekananda Institute of Technology, Bengaluru. She received her B.E in Electronics and M.E in Electronics and Communication Engineering from Bangalore University. She was awarded Ph. D. in Computer Science in the area of Wireless Sensor Networks from Bangalore University, Bengaluru. She has authored and edited three books and published more than 80 papers in refereed International Journals and International Conferences. Her name is listed in Marquis Who's Who in Science and Engineering and has received Best paper Awards. She is selected for the Marquis Who's Who 2017 Albert Nelson Marquis Lifetime Achievement Award. Her research interests include Wireless Sensor Networks, Cloud Computing, Internet of Things and Image Processing.

Venugopal K R is currently the Principal, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science, Bengaluru. He was awarded Ph. D. in Economics from Bangalore University and Ph. D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Law, Mass Communication, Electronics, Economics, Business Finance, Computer Science, Public Relations and Industrial Relations. He has authored and edited 64 books and published more than 600 papers in refereed International Journals and International Conferences. He has supervised 630 M.E. dissertations, 21 Ph.Ds and filed 101 Patents. He has been conferred Fellow of IEEE, USA and ACM Distinguished Educator for contributions to Computer Science Engineering and Electrical Engineering Education. His research interests include Computer Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining.