International Journal of Future Generation Communication and Networking
Vol. 14, No. 1, (2021), pp. 335–346

Elliptic Curve Cryptography Enabled Security For Wireless Communication

1.Dr.K.Ramakrishna Rao,

Professor in CSE, ISTS College for Women ,Rajanagaram, Email Id:abhirkin@gmail.com

2.Mr.V.Ravi Kishore ,

Research Scholar ,Sri Satya Sai University of Technology and Medical Sciences,Sehore, Email Id:ravikishorev1985@gmail.com

3.Mr.Nallamilli .V.K.Reddi,

Assistant Professor in CSE ,Aditya College of Engineering and Technology,Surampalem, Email Id:verendranallamilli88@gmail.com

ABSTRACT

Because of modern technical developments and advancements, computational conditions have evolved dramatically over the past two decades. Also, elliptic curve cryptography is called a fresh method these days, and Elliptic Curve Cryptography is known as a wonderful strategy with a low-key size encryption mechanism for the customer, as well as a task that would be extremely challenging for a talker to hack the wireless framework. The paper addresses the principle of Elliptic Curve Cryptography and the methods for applying it. This paper discusses the problems involved with integrating ECC in sensor network platforms and their numerous applications.

KEYWORDS: Technique, Wireless network, Communication, Cryptography.

I. SECURITY REQUIREMENTS IN WSN

Securing sensor network networks is important in a broad spectrum of device scenarios. WSNs, such as interference, interception, alteration, and fabrication, pose several challenges to the protocol that the system is based upon.

Ideationally, the hazards may be organized according to differing viewpoints. The previous research split down attacks into three separate categories: in terms of the technique's attackers use to conduct their attacks, where these attacks are carried out on the networking stack, and eventually whether or not the malicious node is accepted into the network during the attack. A lot of the problems you can find in WSN come down to issues in the cryptographic domain, especially issues related to key management, safe routing, data aggregation, and intrusion detection, as outlined in Figure 1.

ISSN: 2233-7857 IJFGCN

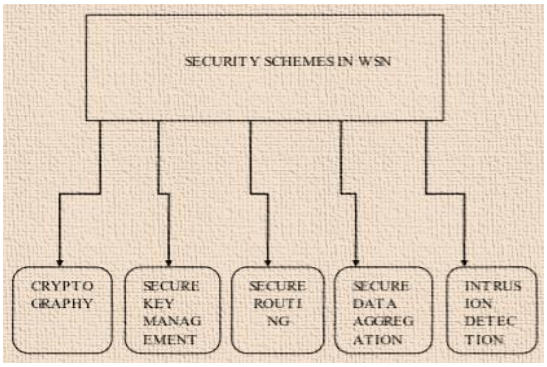

Figure 1: "Various Security schemes in WSN"

With regards to protecting WSNs, there are security priorities which provide security services, such as Secrecy, which is necessary to protect sensitive information from being released [1]. To prevent being impersonated, authenticity is also required for any sensor node and base station. Reasonable security is necessary when coordinating sensor nodes in a WSN, as the confidentiality of information should be maintained to prevent interfering with the details. WSN (Wireless Sensor Networks) may enforce the anonymity of symmetric algorithms while simultaneously providing ample capacity, bandwidth, and memory to satisfy network needs. Though they fall short of providing authenticity and proper key exchange mechanisms that are allowed by public key cryptography, they do enforce some of the requisite security steps.

Various security principles in WSN can be categorizedas:

- Data Confidentiality: This preserves the contents of communications from unwanted sensor nodes.
- data authentication: Confirms the sender's identity by validating that the data they are transmitting was originally created by the proper source.
- Data Integrity: Data is not manipulated without authorization. Digital signature and coding are required to do that.
- Data Freshness: Means the data from previously played doesn't repeat. This protection criteria are required where essential network architecture techniques are applied. Keys must be traded regularly because there is a pause owing to the shipment.
- Availability: Ensure that WSN or a single node delivers resources at all times. Denial of Service attacks and sensor node capture impair overall sensor network availability. By utilizing the center point scheme, a single point loss would be implemented. If this happens, the availability of the network may be seriously threatened [2]. The criteria for protection expand to only security concerns for wireless sensor networks. affects the network service, but is often essential to network stability.
- Sensor network applications typically rely on some sort of time synchronization arrangement. In order to conserve battery life, an individual sensor's radio can be switched off at times. Second, sensors may even like to measure the end-to-end latency of a packet when it passes between two pairwise sensors. The usage of a more collaborative sensor network can necessitate collaboration of sensor networks to keep track of individual applications.
- Seamless, scalable, agile, self-regulating, and corrective community management requires usage of sensor nodes. Solid localization: To protect the integrity of WSN facilities, make sure to share position details regarding sensor nodes safely when talking to authenticated neighbours.

II. FRAMEWORK OF ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

The words elliptic curve cryptography (ECC) and elliptic curve algorithms (ECAlgorithms) relate to the same subject. Although it has some benefits, it has no downsides. Other systems, including those

ISSN: 2233-7857 IJFGCN

with integrated ECC (ECC devices), need more room, more control, more memory, and more bandwidth. Through integrating cryptography on restricted systems, including cellular smartphones, mobile computers, smart cards, and thin clients, you are able to apply it in certain contexts. In conditions where productivity is essential, it provides a major win. Legacy public systems, including the type actually used in the United States, can use a main size of 2048 bits [3].

Although elliptic curves are more inclined than the form of an ellipse or a normal curve, they are not formed like an ellipse or a regular curve. In the looks department, these doughnuts are really close to each other. To use the example of a torus, which is made up of two circles while projected in three-dimensional coordinates, these artifacts somehow mimic the form of the shape [4].

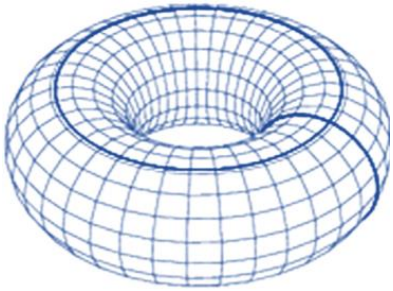

Figure 2: A Torus

It is possible to use a much smaller 224-bit ECC key to achieve the same degree of protection. "If security level changes, an older key is often deemed equal to a newer one—a legacy key with a 3072-bit key length and an ECC key that is 256 bits long have the same level of security.

An elliptic curve is a set of points in the equation.

$$Y^2 = x^3 + ax + b$$

where a&b are real numbers and x and y takeon values in real numbers. Such equations are said to cubicbecause the highest exponent they contain is 3. Using theabove equation, we have to plot the elliptic curve. If threepoints in an elliptic curve lie in a straight line then their sumis O. The negative of point p is the point with the same xcoordinate but the negative of the y coordinate; that is, ifp=(x,y),then –p=(x,-y).note that these two points can bejoined by a vertical line [5]. Note that $p + (-p) = p - p = O$.

To add two points P and Q with different x coordinates, astraight line is drawn between them and find the third pointof intersection R. To form a group structure, we need todefine addition on these three points as follows:

$$P + Q = -R$$

we define P+Q to be the mirror image of the third point ofintersection.

Let $P(x_1, y_1)$and $Q(x_2, y_2)$ be elements of the ECG.

Then $P + Q = (x_3, y_3)$, where

$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$x_3 = \lambda^2 x_1 - x_2$$

ISSN: 2233-7857 IJFGCN

And

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & if \quad P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & if \quad P = Q \end{cases}$$

The points on the elliptic curve should satisfy the equation $Y^2 \bmod p = (x^3 + ax + b) \bmod p$

III. ECC IN WIRELESS SENSOR NETWORK

Sensor networks in the form of wireless sensors (WSNs) are fast gaining acceptance and interest to the science community and the general public. It is important for a range of sensor network applications to have stable protection. Many kinds of attacks may be performed against WSNs due to the vast amount of protection bugs that occur [6].

Wireless sensor networks (WSN) use these compact and low-power sensor devices as building blocks to construct the network. A sensor network involves two connectivity methods: one between sensor nodes, and the other between the base station and the sensor nodes (BS).” Because both the resource limitation and security-sensitive applications in traditional wireless networks, along with critical protection capabilities, are important topics in WSN, security is highly important in WSN [7].

The primary objective of protection in the WSN is not to hide the sender, but to avoid information tampering. Mutual identification is the most essential in order to protect against identity theft.

Owing to energy limitations, a large amount of research activities has been undertaken to allow the effective use of networks. “However, the algorithms will only have anonymity if they are symmetric. The crucial distinction between private key cryptosystems and public key cryptosystems is that the former is highly resource hungry, yet the latter are able to offer a lot more than secrecy. Due to its smaller main scale, ECC drew the interest of researchers [8]. Resource-constrained devices may reap the benefits that come with providing an easy-to-implement interface. Public key algorithms are suggested for wireless sensor networking because of their reliability in wireless sensor networking and because of the beneficial impact that smaller ECC keys and certificates would have on energy saving. authentication and main control are usually accomplished by the usage of ECC.

IV. SECURITY MECHANISM USING ECC

Safe communication between nodes is expected due to the transparency of wireless sensor networks. It utilizes algebraic principles such as elliptic curves over finite fields, Fp or F2m, which is an application of the elliptic curve cryptography (ECC). It takes two parameters (G and an elliptic group) to use elliptic curve encryption and decryption.

For purposes of encryption and transmission, A selects a random positive integer k and computes the cipher text Cm as defined by the following equation: where the pairs of points are used as coordinates.

$$Cm = [k * G, Pm + k * P_B]$$

Here A has used B's public key P_B.

Decryption using ECC

To decrypt the cipher text, B multiples the first point in thepair by B's private key n_B and subtracts the result from thesecond point as shown by equation.

ISSN: 2233-7857 IJFGCN

$$Pm + k * PB - nB(k * G) = Pm + k(n_B * G) - n_B(k * G) = Pm$$

A key exchange between users A and B can be explained as following steps: -

- A select an integer $n_{A<}n$ as A"s private key.
- A generates a public key $P_A = n_{A*}G$ which is a point in E_q (a, b).
- B select an integer $n_{B<}n$ as B"s private key.
- B generates a public key $P_B = n_{B*}G$ which is a point in E_q(a, b).
- Public keys are exchanged between A and B.Agenerates the secret key $K = n_{A*} P_B$ and B generatesthe secret key $K = n_{B*}P_A$

Advantages of ECC

It is a public key cryptographic strategy that is based on elliptic curve theory that uses ECC. While most devices need a 1024-bit key to obtain a comparable degree of protection, with the aid of ECC, you can achieve a security level that needs fewer computational power and battery power.

ECC reduces the CPU overhead, so updating the hardware is not necessary. Newer and older firmware were worked on concurrently by ECC.

V. ALGORITHMS FOR ECC PERFORMANCE OPTIMIZATION

Various algorithms to optimise the performance of elliptic curve multiplication, squaring, point multiplication etc are in use today some of the famous algorithms are given below;

- **Karatsuba Multiplication**

By taking advantage of Karatsuba and Ofman's invention, the number of multiplications can be exchanged for further additions by way of a special algorithm. This approach is effective as long as the time ratio for executing a multiplication is greater than the time ratio for executing an addition.

Consider the example of two degree-1 polynomials, $A(x) = a_1x + a_0$ and $B(x) = b_1x + b_0$

For the traditional method, we must calculate the product of each possible pair of coefficients.

$D_0 = a_0b_0,\ D_1 = a_0b_1,\ D_2 = a_1b_0,\ D_3 = a_1b_1$. And then the product: $C(x) = A(x).B(x)$is:

$$C(x) = D_3x^2 + (D_2 + D_1)x + D_0$$

The Karatsuba method begins by taking the same two polynomials, and calculating the three products

These are then used to assemble the result

$$C(x) = A(x).B(x);$$

$$C(x) = E_1x^2 + (E_2 - E_1 - E_2)x + E_0$$

It is easy to verify that results are equal.

In the conventional method, you need to do four multiplications and one addition, while the Karatsuba approach needs to perform three multiplications and four additions. For that function, the karatsuba

ISSN: 2233-7857 IJFGCN

system is able to execute a single multiplication accompanied by three additions. The strategy is successful if the multiplicity on the target platform is at least three times the sum needed to incorporate." The original paper provides a simple procedure for Karatsuba, but this method can be extended to tackle polynomials of greater degree [9].

- **Itoh-Tsujii Inversion**

Extension field inversion is normally a costly operation, "but the nature of OEFs (Optimal Extension Fields, explained below;)allow the reduction of the extension field inversion to a subfield inversion. The Itoh-Tsujii algorithm which was originally developed for use with composite fields $GF(2^{n^m})$ in a normal basis representation can be applied to extension fields $GF(q)^m$ in polynomial representation. It is assumed that the subfield inverse can be calculated by efficient means, such as table-lookup or the Euclidean algorithm, given a small order of the subfield. To perform the OEF inversion, we use the following expression:

$A^{-1} = (A^r)^{-1}A^{r-1}$, where $r = \frac{q^m-1}{q-1}$. This equation shows the general case for inversion.

- **de Rooij Point Multiplication**

for Q=k.P, well-studied techniques used for ordinary integer exponentiation can be advantageously adapted. The most basic of these algorithms is the binary-double-and-add algorithm It has a complexity of log2(k) + WH(k) group operations, where WH is the Hamming weight of the multiplier k. On average, we can expect this algorithm to require 1.5 log2(k) group operations. Using a method devised by de Rooij, we are able to reduce the number of group operations necessary by a factor of four over the binary-double-and-add algorithm.

- **The Montgomery Modular Multiplication Algorithm**

This is a very famous algorithm for modular multiplication. Its multiple implementations are available both in hardware and software as it is capable to speed up the modular multiplication process by five times." The basic idea behind Montgomery multiplication is the fact that one can add a multiple of the modulus M to the product A·B to yield a result that is at most 2n+1 bits wide. Adding, instead of subtracting, "a multiple of the modulus M does not affect the computation, since the result will be congruent to A·B modulo M. Two numbers are said to be congruent if their remainder when divided by the modulus is the same. Thus, $A \cdot B, A \cdot B + M, A \cdot B + 2M \ldots A \cdot B + kM$ are all congruent modulo M. This implies: $A \cdot B \equiv A \cdot B + M \equiv A \cdot B + 2M \equiv \ldots (A \cdot B + kM)$ mod M. In the Montgomery algorithm, the multiple of the modulus M that is added to A·B is chosen in such a way that the lower n-bits of the 2n+1-bit result are all zeroes. The least significant half of the 2n+1-bit result that are all zeroes is then discarded [10]. This way, the result would have been reduced to at most n+ 1 bit in width. A single subtraction of the modulus M can then be performed to further reduce the result to at most n- bits and make it less than M if required. It has been shown by Walter that the extra subtraction may not be necessary under certain conditions.

It's easy to explain in decimal, but Montgomery multiplication is easier to implement in binary. The place of 10n is taken by some suitable power of 2, but the key simplification is that the adding of the multiple of the modulus becomes much easier. The rule is this: if the number you are looking at is odd (a 1 in LSB), add R before you halve it; if it's even (a 0 in LSB), just halve it. Halving a number in binary is simply discarding its lower significant bit. Eg: (1100 is binary 12 if we discard LSB zero, we get 110, which is binary 6)." Binary shift and rotate instructions are available in every machine level instruction set and can be used for this purpose.

VI. IMPLEMENTATION APPROACHES

ISSN: 2233-7857 IJFGCN

There are specifically software and hardware implementation of an algorithm and it is performed on different platform likely software, ASIC or FPGA. In any implementation process the major aspects to be focused are the memory requirements and the clock cycles in software implementation and the chip size and the clock cycles as well in the hardware implementation [11].

Matlab implementation of ECC

For the implementation of ECC algorithm using MATLAB programming few necessary elements are to be developed for the various required operations namely:

- Modular exponential
- Multiplicative Inverse
- Modular Square roots
- Addition over an elliptic curve
- Multiplication over an Elliptic Curve
- Elliptic curve populates and
- Elliptic curve point verification.

From the figure below shown below the points plotted on the graph denote the (x, y) points which denote the solutions for the curve y2 =x3 + x over F23. The curve point verification factor mentioned above can be checked for the equation thatsatisfies 32=153+15 (mod23)over F23. The interceptpointsobtained(19,22),(15,20),(20,19),(13,18),(9,18),(1,18),(21,17),(16,15), (17,13), (17,10), (16,8), (21,6), (13,5), (9,5), (1,5), (21,4), (15,3), (19,2), (18,13), (18,10) all these 20 points are taken on the graph x varying from 0 to 22 and y varying from 0 to 22.

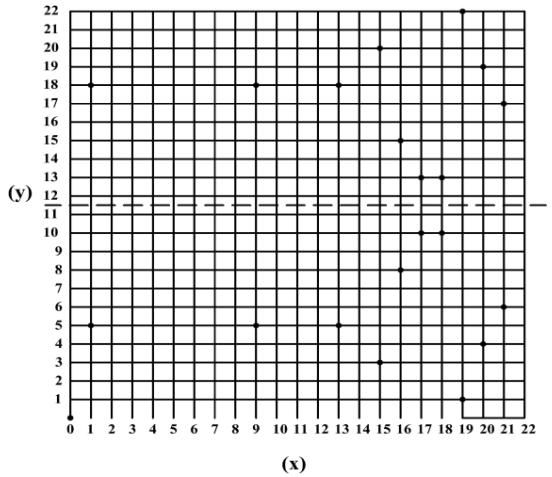

Figure 3:The Elliptic Curve $y^2 = x^3 + x$ over F_{23}

- **Image Encryption Approach**

The block diagram in the Figure 4 below shows an approach of elliptic curve cryptographic algorithm implementation for secured applications for a use case as image as input data. The main criteria in any implementation process are the input in the form of plain text or image is to be converted in to the binary form to continue the further process. In the process of conversion ASCII approach can also be used which is to be stored in the bits form and given as inputs for next block of implementation. The

ISSN: 2233-7857 IJFGCN

block diagram consists of input original image as first block where the input is read and then matrix operation is performed which stores the pixel values and then the major key role of key generation process by ECC which further gives the encrypted data using the elliptic curve discrete logarithm problem (ECDLP). The obtained cipher text is then processed step wise to extract the original image is termed as the decryption process [6].

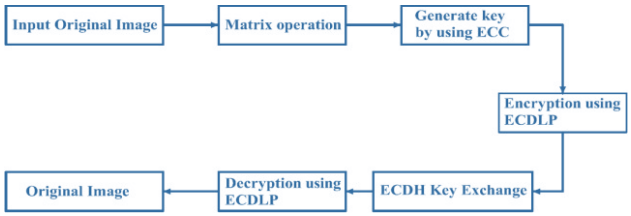

Figure 4: Block Diagram of Image Encryption Method using ECC.

- **VHDL Implementation Approach**

This approach is hardware implementation of ECC algorithm using VHDL as a programming language. The basic operation of the elliptic curves is calculated using the concepts described in the above sections out of all the above mentioned approaches this VHDL approaches gives a scope to understand the operation and the logic operation calculations required which will be further helpful in designing the reconfigurable architecture for ECC using FPGA as the target device. The algorithm is implemented with a key size 233 shown in Figure 5 in the RTL diagram of the ECS_multiplier diagram having key, clock, reset pins as inputs. Figure 6 shows the device utilization summary. Figures 7, 8 and 9 show the simulations of the encryption, decryption and point multiplication respectively. Figure 10 gives the timing summary and Figure 11 gives the power values of the implementation of ECC algorithm on FPGA using Xilinx 14.0 version for synthesis and model sim for simulation on the target device 3s1200ef320-4. The logic blocks, flip flops, LUT's and IOB blocks required in hardware point of view can be seen in device utilization summary. The path delay in terms of critical path delay which is 9.651ns for this implementation plays a vital role in attaining the latency which is to be as low as possible for 5G and IoT era application [8].

VII. SIMULATION RESULT

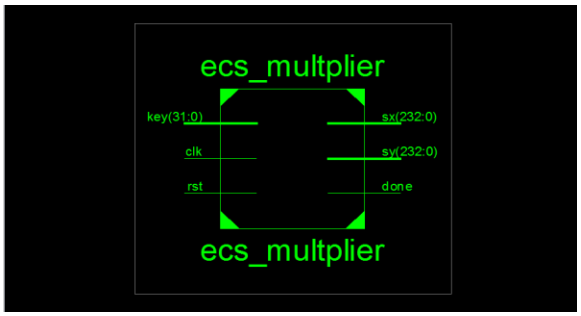

Figure 5: RTL of ecs_multiplier

```
Device utilization summary:
---------------------------

Selected Device : 3s1200efg320-4

 Number of Slices:                 16626  out of   8672   191% (*)
 Number of Slice Flip Flops:          76  out of  17344     0%
 Number of 4 input LUTs:           33447  out of  17344   192% (*)
    Number used as logic:          32049
    Number used as RAMs:            1398
 Number of IOs:                      501
 Number of bonded IOBs:              501  out of    250   200% (*)
 Number of GCLKs:                      1  out of     24     4%
```


ISSN: 2233-7857 IJFGCN

Figure 6: Device Utilization Summary

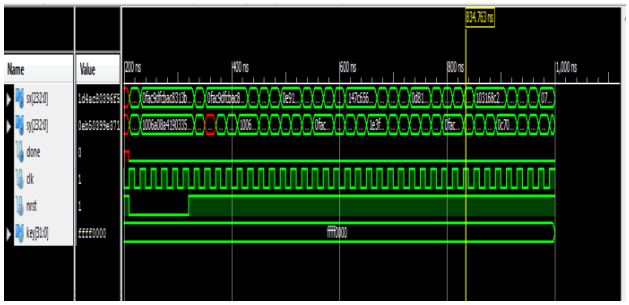

Figure 7: Simulation diagram of Encryption process

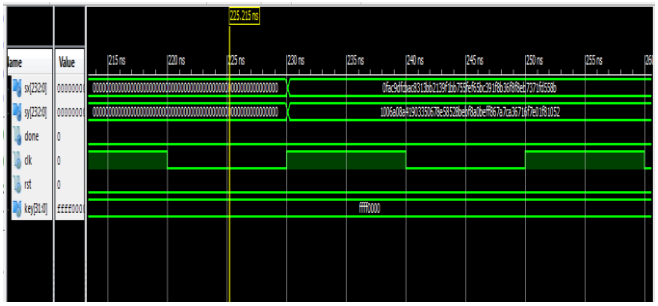

Figure 8: Simulation of Decryption Process

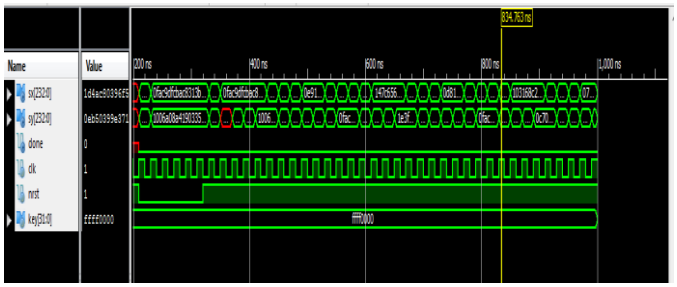

Figure 9: Simulation diagram of Point Multiplication

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 30.683ns (Maximum Frequency: 32.591MHz)
   Minimum input arrival time before clock: 5.189ns
   Maximum output required time after clock: 12.211ns
   Maximum combinational path delay: 9.651ns

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
```

Figure 10: Timing Summary diagram

ISSN: 2233-7857 IJFGCN

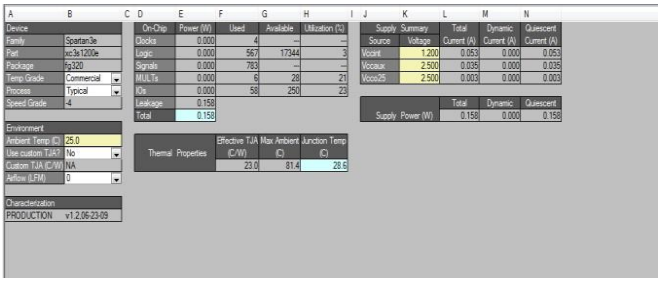

Figure 11: Power values diagram.

Out of all the mentioned approaches of ECC algorithm implementation the VHDL approach is "the basic contribution made so as to understand the algorithm in terms of security application implementation process.

VIII. IMPLEMENTATION ISSUES FOR ELLIPTIC CURVE CRYPTOGRAPHY IN WSN:

The indicative list of relevant issues in implementing ECC is as per below

1. Seven Tuples: In order to set up ECDH scheme all users chooses one group of points. This group of points can be specified by an Elliptic Curve parameter which can be defined as seven Tuples

(a, b, q, G, n, h, Fr)

q (the order of the finite field used) specifies the particular elliptic curve, and a and b specify the specific elliptic curve. Furthermore, q (the order of the finite field used) specifies the specific elliptic curve, and a and b identify the specific elliptic curve. Respectively, q (the order of the finite field used) specifies the specific elliptic curve, and a and b identify the specific elliptic curve. When Fr is used, it gives an example of the underlying field element representation to use.

2. Two separate choices are involved when selecting the parameters: Fixed-point selection or Random-point selection. The first approach consists of selecting a finite field, creating a curve, and selecting a suitable subgroup. Next, a random number generator is used to choose a curve. The seed string is chosen using a finite field with an arbitrary seed." The selection of the proper coordinate system influences the overall output of the ECC system, particularly in relation to computation costs. Affine coordinates indicate inclusion is faster, "while updated Jacobian coordinates show doubling is faster [5].

Table 1: Computational Cost Of Addition And Doubling

Coordinate System	Addition	Doubling
Doubling	S+2M+I	2S+2M+1
Jacobian	4S+12M	6S+4M
Modified Jacobian	6S+13M	4S+4M

* S=Squaring=Multiplication, I=Inversion.

Required level of Security: The Elliptic curve cryptography is based on difficulty of solving Elliptic Curve Discrete Logarithmic Problem. The difficulty of solving ECDLP depends on the size of n which gives the number of points in the specified group. Reasonable size of n give very long time

ISSN: 2233-7857 IJFGCN

periods for solving ECDLP. The following table gives idea about time required to solve ECDLP problem for given size of n

Table 2: Computing Time Estimate for Solving the ECDLP For Various Values Of N

Bit size of n	$\sqrt{\pi/4}$	**MIPS years**
160	2^{80}	$8.5*10^{11}$
180	2^{90}	$8.5*10^{15}$
234	2^{117}	$8.5*10^{23}$
354	2^{177}	$8.5*10^{41}$
426	2^{12}	$9.2*10^{51}$

4. Interoperability: The interoperability aim was to make sure that all of the nodes in the sensor network shared the same EC parameters, and hence had similar keys. Additionally, structured ECC schemes and protocols must be used during correspondence.

5. Performance: Selecting the ECC parameters has a significant effect on the amount of underlying mathematical operations that algorithms will introduce. By using powerful algorithms, the primary computation period would often be decreased." The bandwidth, memory, and computing expense of a wireless sensor network is directly influenced by the number of sensors in the network.

IX. CONCLUSION

After reviewing the above, it is evident that elliptic curve cryptography is very complicated, and incorporating it in wireless sensor networks for sensor network security is fraught with challenges. For device architecture, "the time, memory, and bandwidth specifications for applying authentication and encryption/decryption should be addressed. A rough description of elliptic curve cryptography has been painted in this article.

Application-specific encryption criteria have propelled elliptic curve cryptography into the spotlight. The origins of elliptic curve cryptography can be traced back to the number principle, which was used in cryptographic implementations before ECC. Researchers in the area of Elliptic Curve Cryptography have developed different methods to evaluate the best way of implementing Elliptic Curve cryptography in both hardware and software systems."

REFERENCES: -

[1] Joseph, M &Lokeshwari, Gandrakoti& Kumar, Susarla&gira, aparna. (2020). Implementation of Elliptic Curve Cryptographic Algorithmic Approach for Secured Wireless Communication Applications.

[2] "Anuj Kumar Singh, Arun Solanki, Anand Nayyar and Basit Qureshi (2020) Elliptic Curve Signcryption-Based Mutual Authentication Protocol for Smart Cards Appl. Sci. 2020, 10, 8291; doi:10.3390/app10228291"

[3] E.Pavithra, F.Anishya and M.Nivetha Kumari (2017) "Elliptic Curve Cryptography Based Security Enhancement for Wireless Body Area Network System Asian Journal of Applied Science and Technology (AJAST) Volume 1, Issue 5, Pages 142-146, June 2017."

ISSN: 2233-7857 IJFGCN

[4] Mohamed Elhoseny, HamdyElminir, "Alaa Riad, Xiaohui Yuan (2016) A secure data routing schema for WSN using Elliptic Curve Cryptography and homomorphic encryption, Journal of King Saud University - Computer and Information Sciences, Volume 28, Issue 3, Pages 262-275, ISSN 1319-1578,

[5] Asha Rani Mishra & Mahesh Singh (2012) "Elliptic Curve Cryptography (ECC) for Security in wireless Sensor Network" International Journal of Engineering Research &Technology (IJERT) Vol. 1 Issue 3, May - 2012

[6] Rahat Afreen and S.C. Mehrotra (2011) "A REVIEW ON ELLIPTIC CURVE CRYPTOGRAPHY FOR EMBEDDED SYSTEMS" International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 3, June 2011

[7] SAJEEV, & Jose, G.."(2010). Elliptic Curve Cryptography Enabled Security for Wireless Communication. International Journal on Computer Science and Engineering. 2."

[8] Pritam Gajkumar Shah, Xu Huang, Dharmendra Sharma (2010) "Analytical study of implementation issues of Elliptical Curve Cryptography for Wireless Sensor networks 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops"

[9] J. Dams "Portable Elliptic Curve Cryptography For Medium-Sized Embedded Systems University of Vaasa, Faculty of Technology Department of Computer Science, Vaasa, 2008."

[10] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography, in 2nd IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004) pp. 71-80."

ISSN: 2233-7857 IJFGCN