EAI Endorsed Transactions
on Security and Safety

Research Article EAI.EU

FPGA Implementation of Elliptic Curve Cryptoprocessor for Perceptual Layer of the Internet of Things

V.Kamalakannan[1],* and S.Tamilselvan[2]

[1]Research Scholar, Department of Electronics and Communication Engineering, Pondicherry Engineering College, Puducherry, INDIA.
[2]Associate Professor, Department of Electronics and Communication Engineering, Pondicherry Engineering College, Puducherry, INDIA.

Abstract

Today's developing era data and information security plays an important role in unsecured communication between Internet of Things (IoT) elements. In IoT, data are transmitted in plaintext for many reasons. One of the most common reason is the availability of hardware. Many IoT products are inexpensive components with limited memory and computational resources. Such devices might be unable to support the computationally intense cryptographic functions of asymmetrical cryptography. If designers considered the privacy implications of unencrypted data, they have limited options for encryption because of the hardware platform. Therefore the designers have to create their own security protocols or implement stripped-down versions of existing security protocols. The second option has a better chances. Evidence recommends such a modified protocol would run efficiently on small devices. Elliptic Curve Cryptography (ECC) is used to ensure complete protection against the security risks such as confidentiality, integrity, privacy and authentication by implementing an Elliptic Curve Cryptoprocessor. The work focuses on high-performance Elliptic Curve Cryptoprocessor design, optimized for Field Programmable Gate Array (FPGA) implementation, using the concept of asymmetric and hash algorithms. A novel cryptographic algorithm consisting of matrix mapping methodology and hidden generator point theory is to be applied for encryption/decryption between the sender and receiver whereas Elliptic Curve Digital Signature Algorithm (ECDSA) designed using Keccak Secured Hash Algorithm (SHA) algorithm is applied for the validation of the encrypted data. The proposed Cryptoprocessor operates at a minimum period of 6.980 ns and maximum frequency of 143.276 MHz. This work focuses on the practicability of public key cryptography implementation for devices connected in the perceptual layer of IoT.

Keywords: Internet of Things, Perceptual layer, Wireless Sensor Network, Elliptical Curve Cryptography, Elliptic Curve Digital Signature Algorithm.

Received on 02 July 2018, accepted on 09 October 2018, published on 15 October 2018

doi: 10.4108/eai.15-10-2018.155739

1. Introduction

Internet of Things (IoT) is a global network of interoperable devices. The physical things generally connected in the perceptual layer are Wireless Sensor Networks (WSNs), Radio Frequency Identification (RFIDs), etc., [1]. According to the IoT concept, these devices meet new challenges related to the amount of data exchange, data transferring like storing and retrieving with each other. Thus the demand increases for securing the information from Men in Middle (MIM) attack. The information transmitted pose serious threats to the privacy as they can be tracked anytime, anyplace and by anyone [2]. As these devices have limited processing and memory capabilities, securing by implementing cryptographic

*Corresponding author. Email:vkamalakannan@pec.edu algorithm becomes difficult. Thus an attacker can exploit a weakness in an IoT devices. Therefore cryptographic techniques are applied to protect the data stored, processed and shared, between the devices [3], [4]. A cryptoprocessor is constructed on FPGA using the concept of asymmetrical cryptography and hash algorithm. Cryptographic algorithm consisting of matrix mapping methodology and hidden generator point concept is applied for encryption/decryption whereas modified ECDSA using Keccak SHA-3 algorithm is applied for validation of the encrypted data. A novel version of data security method based on ECC-SHA combinational is implemented on FPGA for Internet of Things devices [5]. An introduction to IoT is provided in section II and the novel Elliptic Curve Cryptoprocessor implementation in section III, modified Elliptic Curve Digital Signature algorithm in section IV. The Proposed cryptoprocessor is provided in section V and the implementation of proposed cryptoprocessor in performed in section VI. The simulation results of Cryptoprocessor are explained in section VII and result analysis in section VIII. The conclusion is discussed in section IX and is followed by references.

2. Internet of Things

The concept of IoT was explained in 1999 by Kenn Ashton and was defined as a dynamic global network infrastructure by *International Telecommunication Union* (ITU) and International Energy Research Centre (IERC) [6]. The idea of IoT was evolved from the concept of Internet. From the research it is said that by 2020 around 50 billion devices will be interconnected with each other. It is considered as an emerging field and is specified in the Gartner's IT hype cycle shown in Figure 1. IoT are different from the existing distributed system because of limited connectivity.

In the IoT scenarios, constellation of wireless devices having a number of key properties such as mobility, wireless, embedded use, diversity and scale; that yields several vulnerabilities related to security. In the non-secure channel, challenges related to security have to be tackled for devices as well as the information [7]. The devices connected in the IoT have their own security requirements, thus in the IoT to ensure confidentiality, authentication, integrity encryption of the information and authentication has to be provided.

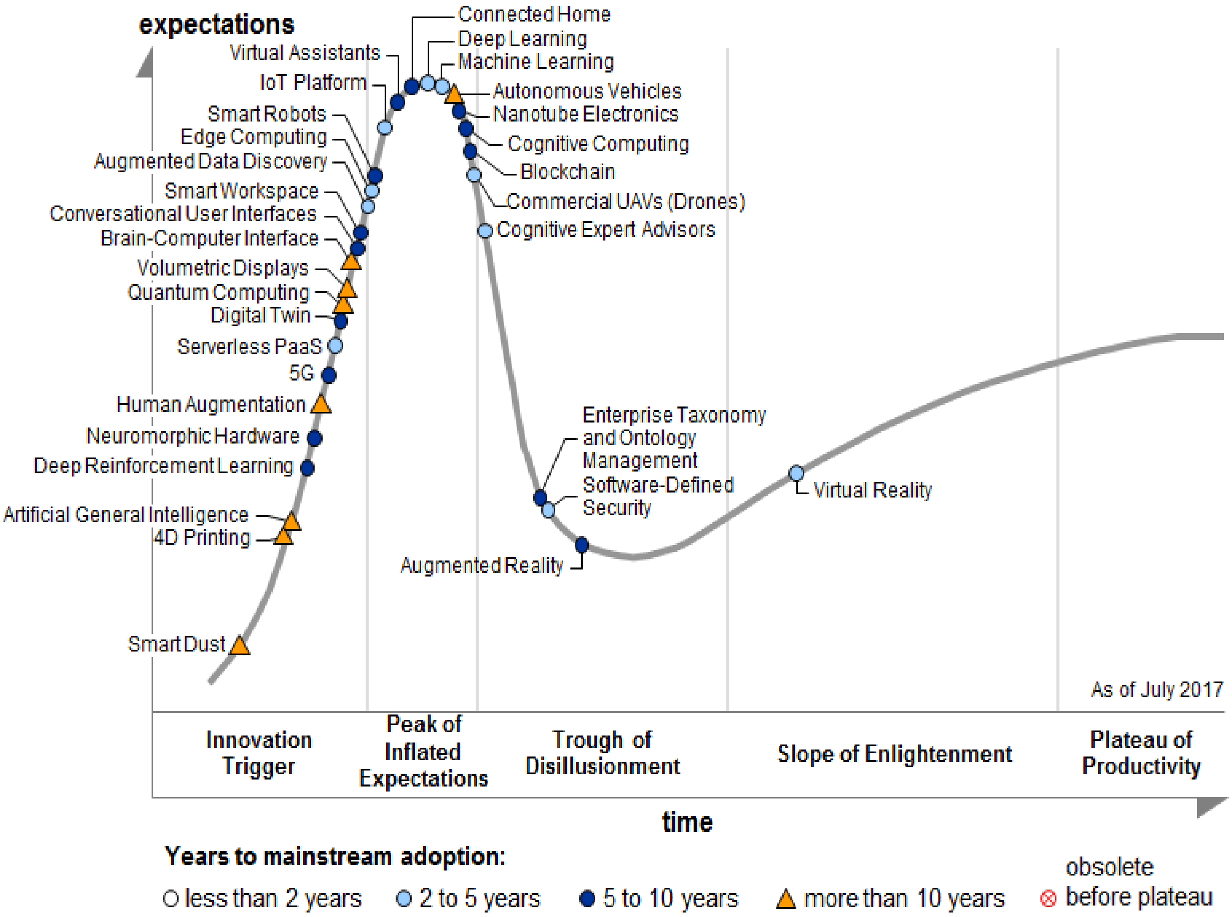

Figure 1. Gartner's Hype cycle

The internal architecture of IoT is built based on the security deficiency. As shown in the Figure 2, IoT is normally considered as three layered architecture consisting of the perception or perceptual layer, the network layer and the application layer [8].

APPLICATION LAYER	APPLICATION SERVICE DATA SECURITY (SMART HOME)
	APPLICATION SUPPORT SECURITY (CLOUD COMPUTING)
NETWORK LAYER	NETWORK TRANSMISSION AND INFORMATION SECURITY (MOBILE COMMUNICATION/INTERNET)
PERCEPTION LAYER	PERCEPTION LAYER LOCAL SECURITY (RFID/SENSORS/GPS/ZIGBEE)

⇧	⇧	⇧
INFORMATION SECURITY	PHYSICAL SECURITY	MANAGEMENT SECURITY

Figure 2. Security Architecture of IoT

The bottom layer is the perception layer or perceptual layer that receives the information through WSNs and RFIDs and passes this information to the middle layer. The middle layer is the network layer which process the information received from the perception layer. The top most layer is the Application layer which generally processes enormous data [9], [10]. This layer segregates the information based on the devices. These enormous data are managed through cloud computing. IoT security is primarily a management issue involving a highly efficient cryptographic algorithms providing confidentiality, integrity, and authenticity [19].

3. Novel Elliptic Curve Cryptography Protocol

The normal ECC primitive protocols are configured with the points on the Elliptic Curve, a generator point '*G*' is selected publicly available and distributed over the network by the Certificate Authority (CA) [11], [12]. In these protocols, the requirement of CA makes it difficult to implement security. The information shared by the CA can be breached by the intruders, making the network susceptible to MIM attack.

Hence to elucidate this exposure and to secure the network against MIM attacks, maintaining the security for each session of communication between the two nodes without a common generator point is suggested. Therefore a generator point is shared only between the devices connected to communicate. This concept is implemented in the ECC primitive protocols [21], [23].

3.1. Initialization Stage

Let us consider two nodes shown in Figure 3 in the WSN. It is assumed that both nodes i.e, sender and receiver select their generator points, G_S and G_R individually apart from the private keys, K_S and K_R. The inverse of the private keys K_S^{-1} and K_R^{-1} are also computed.

Once the keys are initialized, the sender generates public key P_{SA} and using the Eq. (1) and receiver generates public key P_{RA} using the Eq. (2)

$$P_{SA} = K_S^{-1}G_S \tag{1}$$

$$P_{RA} = K_R^{-1}G_R \tag{2}$$

Both the public keys P_{SA} and P_{RA} exchanged after multiplying it with the inverses of private keys. The key transmitted to the receiver is specified in the Eq. (3) and the key received by the sender is specified in the Eq. (4)

$$P_{SB} = P_{RA}K_S^{-1} = K_R^{-1}G_RK_S^{-1} \tag{3}$$

$$P_{RB} = P_{SA}K_R^{-1} = K_S^{-1}G_SK_R^{-1} \tag{4}$$

The keys of the sender and the receiver are multiplied again to generate P_{SC} and P_{RC} given in Eq. (5) and Eq. (6) as

$$P_{SC}=P_{RB}K_S=K_S^{-1}G_SK_R^{-1}K_S =G_SK_R^{-1} \tag{5}$$

$$P_{RC}=P_{SB}K_R=K_R^{-1}G_RK_S^{-1}K_R= G_RK_S^{-1} \tag{6}$$

When P_{SC} and P_{RC} received by the individuals, they are multiplied with K_S and K_R to obtain G_R and G_S.

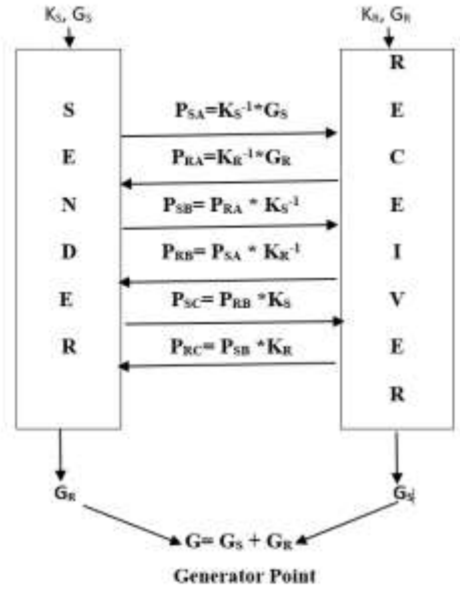

Figure 3. Computational process for Generator point

The sender computes the receiver's generator point in Eq. (7) as

$$P_{RC}*K_S=K_S^{-1}*G_R*K_S=G_R \tag{7}$$

The receiver computes the sender's generator in Eq. (8) as

$$P_{SC}*K_R=K_R^{-1}*G_S*K_R=G_S \tag{8}$$

The generator points G_S and G_R are added to generate a common generator points between the sender and receiver given in Eq. (9) as

$$G = G_S + G_R \tag{9}$$

Hence the sender and receiver exchanges information between them and generated using '*G*' and computing *P*, *2P….. kP*.

3.2. Authorization Stage

The nodes must share a secret key for encryption and decryption. This key must be generated by the process shown in the Figure 4 for every session of transmission between the sender and receiver. Thus authorization has to be provided for transmission.

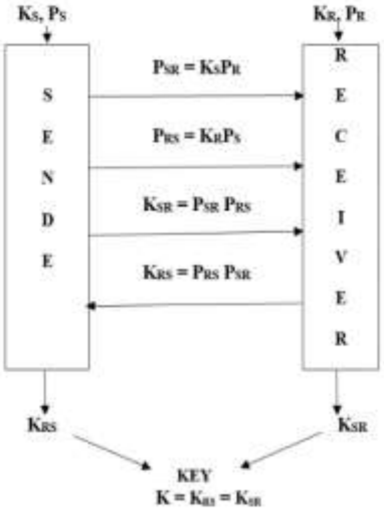

Figure 4. Computational process for Key

The public key and the private keys of the transmitter are P_S and K_S whereas for receiver P_R and K_R. The sender generates P_{SR} and transmits it to receiver, similarly receiver generates P_{RS} and transmits it to sender. The sender then generates K_{SR} in Eq. (10) and the receiver generates K_{RS} in Eq. (11) as

$$K_{SR} = P_{SR}P_{RS} \tag{10}$$

$$K_{RS} = P_{RS}P_{SR} \tag{11}$$

The key for ECC process is specified in Eq. (12) as

$$K = K_{RS} + K_{SR} \tag{12}$$

3.3. Novel Cryptographic Algorithm

The Elliptic Curve Cryptography is implemented using matrix mapping methodology. In matrix mapping method the alphanumeric characters are mapped on to the points of the Elliptic Curve are represented in the form of *m x n* matrix. The points in the matrix form are multiplied with a non-singular matrix and is explained in steps below. It is assumed that both the sender and the receiver of the message to know non-singular matrix A.

Step 1: The original message M of length n is taken for consideration and transformation of the alphanumeric characters into points on Elliptic Curve represented in Eq. (13).

$$[P_1(x_1, y_1), P_2(x_2, y_2) \ldots P_n(x_n, y_n)] \tag{13}$$

Step 2: Matrix of 3 x r is created as shown in Eq. (14)

$$M = \begin{bmatrix} P1 & P2 & P3 & Pr \\ Pr+1 & Pr+2 & Pr+3 & Ps \\ Ps+1 & Ps+2 & Ps+3 & Pn \end{bmatrix} \tag{14}$$

With $r = \frac{n}{3}$ and $s = \frac{2n}{3}$

Step 3: Select a matrix ‘*A*’ of order 3x3 as in Eq. (15), such that ‘A’ is non-singular diagonal.

$$A = \begin{bmatrix} A11 & A12 & A13 \\ A21 & A22 & A23 \\ A31 & A32 & A33 \end{bmatrix} \tag{15}$$

Step 4: Compute Q = AM given in Eq. (16) as

$$Q = \begin{bmatrix} A11 & A12 & A13 \\ A21 & A22 & A23 \\ A31 & A32 & A33 \end{bmatrix} \begin{bmatrix} P1 & P2 & P3 & Pr \\ Pr+1 & Pr+2 & Pr+3 & Ps \\ Ps+1 & Ps+2 & Ps+3 & Pn \end{bmatrix} \tag{16}$$

Step 5: The *Q* matrix is a set of points is given in Eq. (17) as

$$Q = \begin{bmatrix} Q1 & Q2 & Q3 & Qr \\ Qr+1 & Qr+2 & Qr+3 & Qs \\ Qs+1 & Qs+2 & Qs+3 & Qn \end{bmatrix} \tag{17}$$

The message ‘*M*’ to be transmitted after matrix mapping is (Qx_i, Qy_i). For encryption of the message, select the generator point ‘*G*’ and the shared secret key ‘*K*’ compute *KG* in Eq. (18) as

$$KG = (K_1, K_2) = (K_x, K_y) \tag{18}$$

The ciphertext (C_{1i}, C_{2i}) is computed by Eq. (19) and Eq. (20) as

$$C_{1i} = (Qy_i\, K_x + Qx_i) \bmod p \tag{19}$$

$$C_{2i} = (Qy_i + Qy_i\, K_xK_y + Qx_iK_y) \bmod p \tag{20}$$

For decryption, the ciphertext (C_{1i}, C_{2i}) is considered and the message is computed using the Eq. (21) and Eq. (22) as

$$Qy_i = (C_{2i} - K_yC_{1i}) \bmod p \tag{21}$$

$$Qx_i = (C_{1i} - Qy_i\, K_x) \bmod p \tag{22}$$

At the receiver end the decrypted values are represented in the matrix Q and the following step occurs in demapping process

Step 1: The receiver performs inverse of matrix selected and is given in Eq. (23) as

$$A^{-1} = \begin{bmatrix} Aa1 & Aa2 & Aa3 \\ Ab1 & Ab2 & Ab3 \\ Ac1 & Ac2 & Ac3 \end{bmatrix} \tag{23}$$

Step 2: The resultant of the inverse matrix and encrypted data are considered to compute the Elliptic Curve mapped message M = A^{-1}Q represented in Eq. (24).

$$M = \begin{bmatrix} Aa1 & Aa2 & Aa3 \\ Ab1 & Ab2 & Ab3 \\ Ac1 & Ac2 & Ac3 \end{bmatrix} \begin{bmatrix} Q1 & Q2 & Q3 & Qr \\ Qr+1 & Qr+2 & Qr+3 & Qs \\ Qs+1 & Qs+2 & Qs+3 & Qn \end{bmatrix} \tag{24}$$

Step 3: The resulting set of points is given in Eq. (25) as

$$M = \begin{bmatrix} P1 & P2 & P3 & Pr \\ Pr+1 & Pr+2 & Pr+3 & Ps \\ Ps+1 & Ps+2 & Ps+3 & Pn \end{bmatrix} \tag{25}$$

Step 4: This Elliptic Curve point mapped message is remapped with ASCII characters to get back the original information.

The receiver performs inverse of matrix selected and the resultant of the inverse matrix and decrypted data are considered to compute the Elliptic Curve mapped message M. This Elliptic Curve point mapped message is remapped with ASCII characters to get back the original information The decryption process is verified as

$$\begin{aligned} Qy_i &= C_{2i} - K_yC_{1i} \\ &= (Qy_i + Qy_iK_xK_y + Qx_iK_y) - K_y\,(Q_{yi}K_x + Qx_i)\ (\mathrm{mod}\ p) \\ &= Qy_i + Qy_iK_xK_y + Qx_iK_y - Q_{yi}K_{xi}K_{yi} - Qx_iK_y \\ &= Qy_i \end{aligned}$$

$$\begin{aligned} Qx_i &= C_{1i} - Q_{yi}K_x \\ &= Q_{yi}K_x + Q_{xi} - Q_{yi}K_x = Qx_i \end{aligned}$$

4. Modified Elliptic Curve Digital Signature Algorithm

.In the proposed ECDSA model hidden generator point concept is applied to authenticate the encrypted message communicated between the devices connected in the perceptual layer of IoT [13], [14].

Considering two nodes as sender and receiver in the WSN. These nodes have individual generator points, G_S and G_R with their unique private keys, K_S and K_R. After initializing the keys generation process, both devices exchange the generator points G_S and G_R and generate a common generator point by the initialization process explained in sub section 3.1. Hence the sender and receiver exchanges information between them by considering common generator point *'G'* and computing *P, 2P, 3P….. kP*[17]. [18].

The sensor nodes must securely share a key before encryption. The shared secret key is generated and refreshed between the sender and receiver. The public key of sender and receiver are P_S and P_R. are exchanged using DHKE process and a key is generated by the method explained in the section 3.

Considering the generator point *'G'* and key *'K'*, scalar multiplication is performed to compute *KG* provided in Eq. (26), to be applied for signature generation and signature verification process.

$$KG = (K_1, K_2) = (K_X, K_Y) \tag{26}$$

From the initialization and authorization stage, the values of K and G are known. This scheme processes are discussed as following steps

To generates the signature for message *M* the signer using the values of *K* and *G* by performing the following steps:

Step 1: Calculate h = HASH (M) = SHA-3 (M)
Step 2: Compute $KG = (x_1, y_1)$
Step 4: Compute $r = x_1$ (mod p)
Step 5: Compute $s = (K + (r\ xnor\ h))\ G$ (mod p).
The signature pair thus generated is (r, s)

The verifier verifies the signature using K and G from the initialization and authorization stage for message M by performing the following steps:

Step 1: Verify that s is integers in $[1, p-1]$. If not, the signature is invalid
Step 2: Compute $KG = (x_1, y_1)$
Step 3: Compute $r = x_1$ (mod p)
Step 4: Compute $u = (r\ xnor\ h)$ mod (mod p)
Step 5: $(x_2, y_2) = (s - uG)$
Step 6: The signature is valid if $v = x_2$ (mod p) $= r$, invalid otherwise.

4.1 Proof of Proposed ECDSA Scheme

Signature send by sender to receiver is (r, s) and s can be generated only by Sender because of its private key

Step 1: Compute $s = wG = (K + (r\ xnor\ h)\ G$
Step 2: Compute $s = (K + u)\ G$ where $u = (r$ xnor h)
Step 3: Compute $s = KG + uG$
Step 4: Compute $KG = s - uG = (s - uG) = (x_2, y_2)$
Therefore
LHS = KG = (x_1, y_1) and $r = x_1$ (mod p)
RHS = $(s - uG) = (x_2, y_2)$ and v = x_2(mod p)
Hence v=r

The improved ECDSA scheme reduce the computational cost while keeping the same security as original ECDSA. They are suitable for the users who have limited compute capacity [15], [16].

5. Proposed Elliptic Curve Cryptoprocessor

With the development of Integrated Circuits technology, WSN is growing strong and is widely used in special fields. As it has limited energy, communication and storage resources, encryption and authentication protocols must be improved [20], [22]. In the proposed cryptoprocessor the security of information transmitted by devices in the IoT is implemented. This cryptoprocessor plays a major part by applying methods such as key generation using Diffie Hellman Key Exchange process and hidden generator concept requiring exchange of messages during the transmission of data.

Considering two nodes as sender and receiver in the WSN. These nodes have individual generator points, G_S and G_R with their unique private keys, K_S and K_R. After initializing the keys generation process, both devices exchange the generator points G_S and G_R and generate a common generator point by the initialization process explained in sub section 3.1. Hence the sender and receiver exchanges information between them by considering common generator point *'G'* and computing *P, 2P, 3P….. kP*.

The sensor nodes must securely share a key before encryption. The shared secret key is generated and refreshed between the sender and receiver. The public key of sender and receiver are P_S and P_R. are exchanged using DHKE process and a key is generated by the method explained in the sub section 3.2.

Considering the generator point *'G'* and key *'K'*, scalar multiplication is performed to compute *KG* provided in Eq. (27), to be applied for encryption/decryption and signature generation and verification process.

$$KG = (K_1, K_2) = (K_X, K_Y) \tag{27}$$

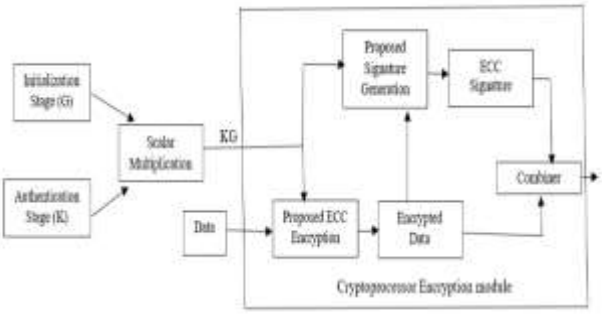

Figure 5. Cryptoprocessor Transmitter

In the proposed Cryptoprocessor transmitter model represented in Figure 5, by considering the value of KG, the plain data is encrypted by applying proposed ECC encryption process to generate cipher data. The cipher data is applied to proposed ECDSA protocol for signature generation. The hash function SHA-3 Keccak algorithm is considered for generating signature. The two ECC primitive protocols outputs i.e., encrypted data and signature are mutually combined in transmitter before transmission.

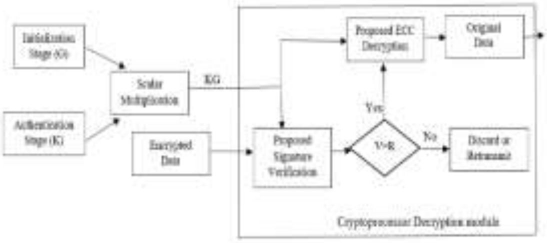

Figure 6. Cryptoprocessor Receiver

In the proposed Cryptoprocessor receiver model represented in Figure 6. The combined data i.e., the encrypted data and signature are separated in the receiver verification and decryption process. Considering the value of KG, the cipher data is applied to proposed ECDSA protocol for signature verification. If the signature is verified the cipher data is decrypted by applying proposed ECC decryption process to generate plain data else the cipher data is discarded. The hash function SHA-3 Keccak algorithm is considered for verifying the signature.

6. Implementation of Proposed Elliptic Curve Cryptoprocessor

The hardware implementation proposed Cryptoprocessor model was realized in Verilog HDL and simulation carried out using ISim simulation tool in XILINX for verifying its functional correctness. For validating the functional correctness Xilinx 14.3 synthesis tool used to do synthesis. The wireless sensor nodes in the network transfer or exchange information between the nodes. The input data to be transmitted in hexadecimal form DA2A806A1F4A1F4A1F4A1F4A1F0652D9. The 128 bits of data divided into 16 blocks and is encrypted using modified ECC algorithm with matrix mapping methodology.

In the initial stage, a generator point 'G' is computed individually by the sender and receiver. The sender has the choice to choose an Elliptic Curve, represented as $Y_1^2=(X_1^3+3X_1+4) \ mod \ 43$, where a=3, b=4 and p=43. To find out the points on an Elliptic Curve we compute $Y_1^2 \ mod \ 43$ for $Y_1= 0 \ to \ 43$ and compute $Y_1^2=(X_1^3+3X_1+4) \ mod \ 43$ for $X_1= \ 0 \ to \ 43$. Considering the values the elliptic points available on the curve a generation point G_A = (30, 41) was selected.

The Elliptic Curve individually chosen by the receiver is $Y_2^2=(X_2^3+9X_2+9) \ mod \ 29$, where a=9, b=9 and p=29. To find out the points on an Elliptic Curve we compute $Y_2^2 \ mod \ 29$ for $Y_2= 0 \ to \ 29$ and compute $Y_2^2=(X_2^3+9X_2+9) \ mod \ 29$ for $X_2= \ 0 \ to \ 29$. Considering these values the elliptic points available on the curve a generation point G_B = (22, 26) was selected.

The shared generator point is calculated by generating the sender's secret key K_S as 13 and receiver's secret key K_R as 25. Generator point between two nodes for each communication session is generated and common generator point is calculated based on the proposed method and is generated as

$$G_C = G_A + G_B = (14, 34) + (21, 18) = (12, 1) = G$$

Considering $G = P = (12, 1)$, scalar point multiplication is performed to calculate $2P, 3P, \ldots\ldots\ldots up \ to \ 256P$, and are mapped with the hexadecimal equivalent representation, 00 to FF, where each block is of 8 bits. The scalar multiplication is performed by point addition and point doubling process. The hexadecimal characters are than mapped to the scalar multiplied points.

The hexadecimal equivalent of the individual block is mapped to a scalar multiple of the generator point and is represented in a matrix form. Let M be the matrix form of key of length n =16 and is given in Eq. (28) as

$$\mathrm{M}=\begin{vmatrix} DA & 2A & 80 & 6A \\ 1F & 4A & 1F & 4A \\ 1F & 4A & 1F & 4A \\ 1F & 06 & 52 & D9 \end{vmatrix}= \begin{vmatrix} (127,148) & (115,109) & (072,104) & (191,100) \\ (225,082) & (108,173) & (225,082) & (108,173) \\ (225,082) & (108,173) & (225,082) & (108,173) \\ (225,082) & (228,042) & (113,215) & (201,135) \end{vmatrix} \qquad (28)$$

A 4*4 matrix is selected in Eq. (29) such that |A| ±1

$$A=\begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 1 & 1 & 1 & 0 \\ 1 & 4 & 2 & 3 \end{vmatrix} \qquad (29)$$

The mapping is accomplished by multiplying the non-singular matrix with the message matrix to get Q matrix represented in the Eq. (30) as

$$Q=AM=\begin{vmatrix} (115,101)(\ 94,229)(153, \ 81)(\ 55,135) \\ (\ 30, \ 90)(216,213)(\ 88, \ 22)(181,88) \\ (176,203)(110,68)(110, \ 68)(\ 42,187) \\ (185, \ 22)(229,169)(\ 31,210)(\ 52,135) \end{vmatrix} \qquad (30)$$

In modified Diffie Hellman key exchange process a random number K_S is generated by sender is 13 and a random number K_R generated by receiver is 25. The public key shared by sender and receiver are $P_S = P_R = 45$.

Hence $P_{SR} = K_S P_S$ = 13 x 45 mod 233 = 119

Hence $P_{RS} = K_R P_R$ = 25 x 45 mod 233 = 193

$K = K_{SR} = K_{RS} = P_{SR} P_{RS}$ mod p = 119 x 193 mod 233 = 133

This is the number with generator point applied to Eq. (6.1) to get

KG= 133 (12, 1) mod 233 = (131, 133)

The elements of the Q matrix are encrypted using modified ECC algorithm represented in Eq. (16) and Eq. (17)

$$C_{1i} = (Qy_i \ K_x + Qx_i) \ mod \ p$$

$$C_{2i} = (Qy_i + Qy_i \ K_xK_y + Qx_iK_y) \ mod \ p$$

The ciphertext (C_{1i}, C_{2i}) is computed to get C Matrix specified in Eq. (31)

$$C=\begin{vmatrix}(223,29)(182,188)(2,142)(114,117)\\(147,30)(196,183)(70,60)(109,34)\\(137,71)(12,201)(12,201)(122,180)\\(167,106)(6,119)(71,162)(111,142)\end{vmatrix} \qquad (31)$$

The signature of A for message M is the pair (r, s) = (d3, 565d)

The Receiver receives the cipher text and the ciphertext *(C_{1i}, C_{2i})* is decrypted using Eq. (16) and Eq. (17) to get Q matrix specified in Eq. (32) as

$$Qy_i = (C_{2i} - K_y C_{1i}) \text{ md } p$$

$$Qx_i = (C_{1i} - Qy_i\, K_x) \bmod p$$

$$Q=\begin{vmatrix}(115,101)(\ 94,229)(153,\ 81)(\ 55,135)\\(\ 30,\ 90)(216,213)(\ 88,\ 22)(181,88)\\(176,203)(110,68)(110,\ 68)(\ 42,187)\\(185,\ 22)(229,169)(\ 31,210)(\ 52,135)\end{vmatrix} \qquad (32)$$

Consider the inverse of the non-singular matrix specified in Eq. (33)

$$A^{-1} = \begin{vmatrix}-1 & 2 & 1 & -1\\-2 & 1 & 1 & 0\\3 & -3 & -1 & 1\\1 & 0 & -1 & 0\end{vmatrix} \qquad (33)$$

In the receiver the message in the form of Elliptic Curve points are calculate by multiplying the inverse of non-singular matrix with the decoded Q matrix. The message is de-mapped to get back the information in the Hexadecimal formatshown in Eq. (34).

$$M=A^{-1}Q$$

$$M=\begin{vmatrix}(127,148) & (115,109) & (072,104) & (191,100)\\(225,082) & (108,173) & (225,082) & (108,173)\\(225,082) & (108,173) & (225,082) & (108,173)\\(225,082) & (228,042) & (113,215) & (201,135)\end{vmatrix}$$

$$M=\begin{vmatrix}DA & 2A & 80 & 6A\\1F & 4A & 1F & 4A\\1F & 4A & 1F & 4A\\1F & 06 & 52 & D9\end{vmatrix} \qquad (34)$$

7. Simulation Results of Cryptoprocessor

The proposed cryptoprocessor transmitter and receiver was realized in Verilog HDL and simulation carried out using ISim simulation tool in XILINX for verifying its functional correctness.

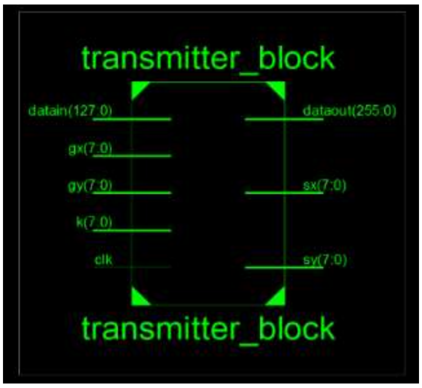

Figure 7. Top module of Cryptoprocessor Transmitter

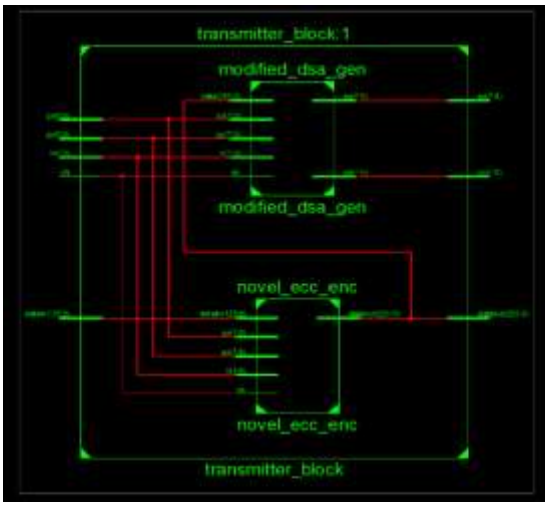

Figure 8. Block Diagram of Cryptoprocessor Transmitter

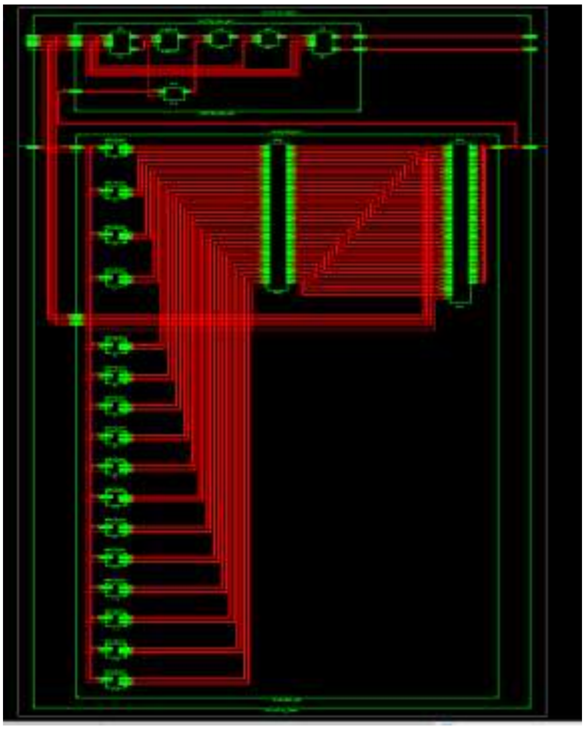

Figure 9. Expanded Block Diagram of Cryptoprocessor Transmitter

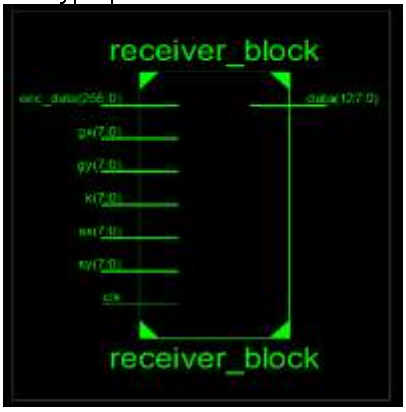

Figure 10. Top Module of Cryptoprocessor Receiver

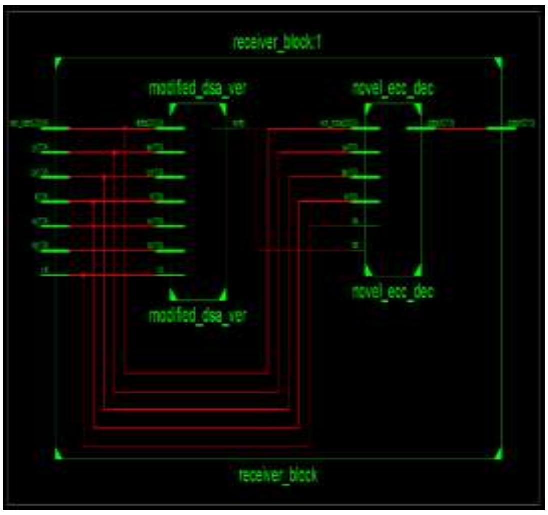

Figure 11. Block Diagram of Cryptoprocessor Receiver

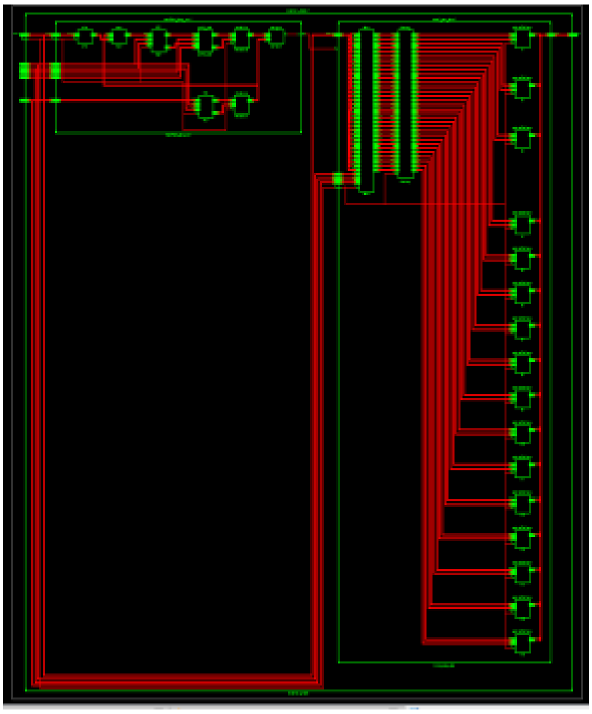

Figure 12. Expanded Block Diagram of Cryptoprocessor Receiver

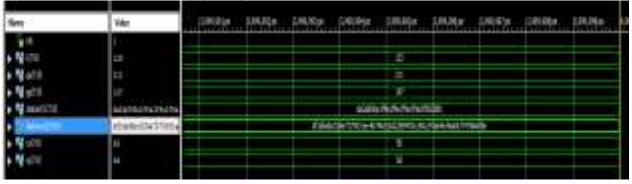

Figure 13. Simulation Results of Cryptoprocessor Transmitter

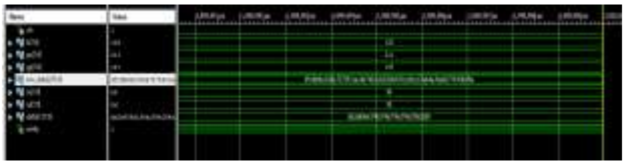

Figure 14. Simulation Result of Cryptoprocessor Receiver

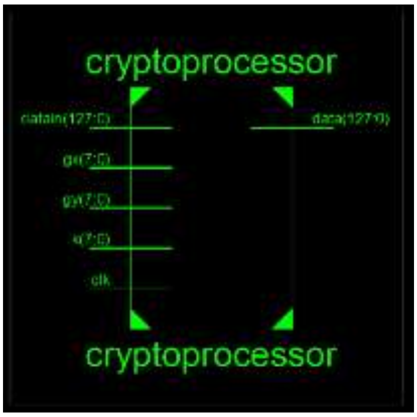

Figure 15. Top Module of Proposed Cryptoprocessor

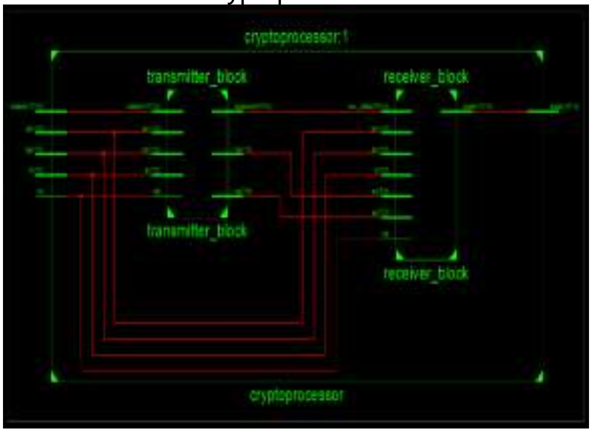

Figure 16. Block Diagram of Proposed Cryptoprocessor

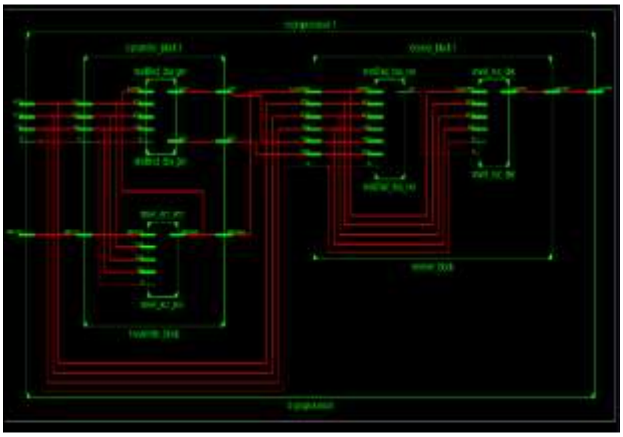

Figure 17. Expanded Block Diagram of Proposed Cryptoprocessor

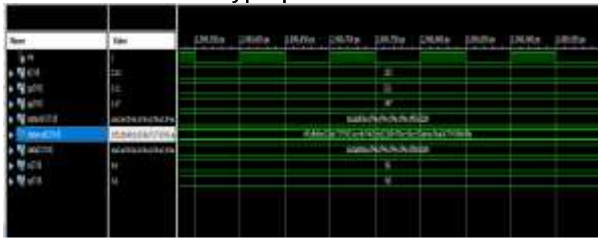

Figure 18. Simulation Result of Proposed Cryptoprocessor

For validating the functional correctness Xilinx 14.3 synthesis tool used to do synthesis. The top module, block diagram and the expanded block diagram of the proposed cryptoprocessor transmitter generated are shown in the Figure 7 to Figure 9 and proposed cryptoprocessor receiver generated are shown in the Figure 10 to Figure 12. The simulation result of the transmitter block is presented in Figure 13 and receiver block is presented in Figure 14. The proposed cryptoprocessor was realized in Verilog HDL and simulation carried out using ISim simulation tool in XILINX for verifying its functional correctness. For validating the functional correctness Xilinx 14.3 synthesis tool used to do synthesis. The top module, block diagram and expanded block diagram of the proposed cryptoprocessor is generated and is as shown in the Figure 15 to Figure 17. The simulation result of the transmitter block is presented in Figure 18.

8. Result Analysis of Cryptoprocessor

The cryptoprocessor transmitter and receiver are synthesized and the device utilization summary, timings summary and memory utilization is tabulated in Table 1. The hardware implementation of Elliptic Curve Cryptoprocessor transmitter and receiver has been implemented. The transmitter and receiver are designed using Verilog HDL, and implemented on a Genesys Virtex-5 5XC5VLX50T-1FF1136 FPGA Development board by XILINX to evaluate the area and speed. The transmitter operates at a frequency of 151.366 MHz and performs the encryption operation in 6.606 ns. The receiver operates at a frequency of 140.089 MHz and performs the encryption operation in 7.138 ns.

Table 1 Synthesis Summary of Cryptoprocessor Transmitter and Receiver

Cryptoprocessor	Transmitter	Receiver
Parameters	**Used**	**Used**
Number of Slice Registers	13407	5628
Number of Slice LUTs	24452	5058
Number of LUT flip flop pairs used	8892	2349
Total Real Time to Xst Completion	1345.00 secs	339.00 secs
Total CPU Time to Xst Completion	1344.87 secs	338.21 secs
Total Memory usage	1482412 KB	591916 KB
Minimum period	6.606 ns	7.138 ns
Maximum Frequency	151.366 MHz	140.089 MHz

The proposed cryptoprocessor is synthesized and the device utilization summary, timings summary and memory utilization is tabulated in Table 2. The hardware implementation of proposed Cryptoprocessor has been implemented in this chapter. The Cryptoprocessor is designed using Verilog HDL, and implemented on a Genesys Virtex-5 5XC5VLX50T-1FF1136 FPGA Development board by XILINX to evaluate the area and speed. The system operates at a frequency of 143.276 MHz and performs the operation in 6.980 ns.

Table 2 Synthesis Summary of Proposed Cryptoprocessor

Parameters	Available	Used	Utilization
Number of Slice Registers	28,800	11,124	38 %
Number of Slice LUTs	28,800	9,957	34 %
Number used as Logic	28,800	9,955	34 %
Number of occupied Slices	7,200	4,353	60 %
Number of LUT flip flop pairs used	15,489	5,592	36 %
Number of bonded IOBs	480	389	81 %
Number of Block RAM/FIFO	60	19	31 %
Total Memory usage (KB)	2160	576	26 %
Average Fan-out of Non-Clock Nets		4.23	
Minimum period		6.980 ns	
Maximum Frequency		143.276 MHz	

9. Conclusion

Elliptic Curve Cryptoprocessor and the parameters used for devices connected in the perceptual layer of IoT are briefly reviewed and the structure combining ECC and ECDSA has been presented. To overcome MIM attack and secure the unsecured channel, ECC primitive protocols were modified using a matrix mapping methodology and hidden generator point concept. It also focused on generating a unique shared key for each session of communication between the two nodes in unsecured channel. The information transmitted was validated by adding a signature to the data. The two ECC primitive protocols (a novel Elliptic Curve Cryptosystem and modified ECDSA) mutually enhance the performance of Cryptoprocessor. These ECC primitive protocols are suitable for the devices having limited compute capability. The proposed Elliptic Curve Cryptoprocessor was implemented on a Genesys Virtex-5 5XC5VLX50T-1FF1136 FPGA Development board by XILINX. It operates at a minimum period of 6.980 ns and maximum frequency of 143.276 MHz. Thus the proposed hardware assisted approach makes asymmetrical cryptography practicable on resource constrained devices connected in the perceptual layer of IoT.

Reference

[1] Ashton, K (2009). That “Internet of Things” Thing: In the Real World Things Matter More than Ideas. RFID Journal, pp. 97-114.

[2] Manoj Kumar S (2015). An Analysis of Authentication Schemes for Internet of Things. International Journal of Engineering Sciences & Research Technology, vol. 4, no. 6, pp. 978-984.

[3] Weizhe Zhang, Baosheng Qu (2013). Security Architecture of the Internet of Things Oriented to Perceptual Layer. International Journal on Computer, Consumer and Control (IJ3C), vol. 2, no.2, pp. 37-45.

[4] Sundmaeker H, Guillemin P, Friess P, Woelfflé S (2010). Vision and challenges for realizing the IoT. Cluster of European Research Projects on the IoT—CERP-IoT.

[5] William Stallings (2011). Cryptography and Network Security-Principles and Practice. Pearson Education.

[6] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswami (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future generation computer system (Elsevier),vol.29, pp. 1645-1660.

[7] Manoj Kumar S (2015). An Analysis of Authentication Schemes for Internet of Things. International Journal of Engineering Sciences & Research Technology, vol. 4, no. 6, pp. 978-984.

[8] Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva (2015). Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. IEEE Communication Surveys & Tutorials, vol. 17, no. 3, pp. 1294-1312.

[9] Sheetal Kalra and Sandeep K. Sood (2015). Secure Authentication Scheme for IoT and Cloud Servers. Journal of Pervasive and Mobile Computing, vol. 24, issue C, pp. 210-223.

[10] HuiDan Gao, YaJun Guo, JianQun Cui, HengGeng Hao and Hui Shi (2012). A Communication Protocol of RFID Systems in Internet of Things. International Journal of Security and Its Applications, vol. 6, no.2, pp. 91-102.

[11] Rajam STR, Kumar SBR (2015). Enhanced Elliptic Curve Cryptography. Indian Journal of Science and Technology, vol. 8, no. 26, pp. 1-6.

[12] Chelton WN, Benaissa M (2008). Fast Elliptic Curve Cryptography on FPGA. IEEE Transactions on Very Large Scale Integration (VLSI) System, vol. 16, no. 2, pp. 198-205.

[13] G. Provelengios, P. Kitsos, N. Sklavos, and C. Koulamas (2012). FPGA-Based Design Approaches of Keccak Hash Function. Proceedings of 15th Euromicro Conference, pp. 648–653.

[14] Elhadjyoussef Wajih, Benhadjyoussef Noura, Machhout Mohsen & Tourki Rached (2012). Low Power Elliptic Curve Digital Signature Design for Constrained Devices. International Journal of Security (IJS), vol. 6, no.2, pp. 1-14.

[15] Greeshma Sarath, Devesh C Jinwala and Sankita Patel (2014). A Survey on Elliptic Curve Digital Signature Algorithm and its Variants. Computer Science & Information Technology (CS & IT) –CSCP, pp. 121–136.

[16] Al Imem Ali, Prince Isitc, H (2015). Comparison and Evaluation of Digital Signature Schemes Employed in NDN Network. International Journal of Embedded systems and Applications (IJESA), vol.5, no.2, pp. 15-29.

[17] Debiao He and Sherali Zeadally (2015). An Analysis of RFID Authentication Schemes for Internet of things in Healthcare Environment Using Elliptic Curve Cryptography. IEEE Internet of Things Journal, vol. 2, no. 1, pp. 241-247.

[18] Farooq, M.U, at.el. (2015). A Critical Analysis on the Security Concerns of Internet of Things (IoT). International Journal of Computer Applications, vol. 11, pp. 1–6.

[19] Farheen Fatima, at el. (2015). , Internet of things: A Survey on Architecture, Applications, Security, Enabling Technologies, Advantages & Disadvantages. International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 12, pp. 498-504.

[20] Dindayal Mahto and Dilip Kumar Yadav (2017). RSA and ECC: A Comparative Analysis. International Journal of Applied Engineering Research, vol. 12, no. 19, pp. 9053-9061.

[21] Swapnoneel Roy and Chanchal Khatwani (2017). Cryptanalysis and Improvement of ECC Based Authentication and Key Exchanging Protocols. MDPIi Journal of Cryptography, vol. 1, no. 9, pp. 1-25.

[22] Dr.S.Vasundhara (2017). Elliptic curve Cryptography and Diffie- Hellman Key exchange. IOSR Journal of Mathematics (IOSR-JM), vol. 13, no. 1, pp. 56-61.

[23] Muneer Ahmad Dar and Javed Parvez (2017). Security Enhancement in Android using Ellipic Curve Cryptography. International Journal of Security and Its Applications, vol. 11, no. 6, pp.27-34.