

Performance Analysis of MANET using Elliptic Curve Cryptosystem.

Adarsh Kumar¹, Alok Aggarwal², Charu³

¹²³Department of CSE/IT, Jaypee Institute of Information Technology, Noida, India

¹adarsh.kumar@jiit.ac.in, ²alok.aggarwal@jiit.ac.in, ³er.charu@gmail.com

Abstract—In this work various elliptic curve are compared and evaluated based on their operations i.e. Addition and Doubling. Out of these elliptic curves classical weierstrass and jacobian elliptic curves are selected to implement in MANETs (Mobile Ad Hoc Networks) and further network performance is analyzed to calculate efficient processing unit in a partially distributed key management environment. To manipulate the efficient processing unit various fast processing algorithms are selected and compared. Performance of these algorithms are evaluated in MANET using AODV (Ad-Hoc On Demand Distance Vector), DSDV (Destination Sequenced Distance Vector) and DSR (Dynamic Source Routing) protocols for small to medium scale networks. Results are compared to find the best routing and computational algorithm.

Keywords - Elliptic curve, processing unit, MANET, fast computational algorithms, partial key distribution, performance

I. INTRODUCTION

Mobile Ad Hoc Network (MANET) is decentralized network and operates without the support of fixed infrastructure. Nodes in MANET moves freely and at random speed. MANET can be implemented using either laptops or sensor nodes [1]. On laptop there is no major concern of memory storage or processing speed but on sensor nodes, MANET has limitations in terms of scarce bandwidth, processing unit, routing, switching, security etc. In last few decades much work has been progressed in different areas of MANET. Out of these areas one of the major concerns is about cryptography and network security.

Cryptography and network security are means of hiding information for secure communication. In traditional scenarios, cryptography is classified as (i) Symmetric key cryptography / Secret key cryptography and (ii) Asymmetric key cryptography / public key cryptography. In symmetric key cryptosystem both sender and recipient party shares the same key. If one end uses this for encryption other end will use it for decryption. Symmetric key can be classified as: (a) transposition cipher, (b) substitution cipher (c) block cipher and shift cipher protocols. Whereas public key cryptography can be classified as: (a) integer factor method, (b) discrete logarithm and (c) elliptic curve cryptosystem. Further, Cryptography is a mode of providing three major services: authentication, encryption/decryption and key management [2]. Majority of security algorithms need large computations thus demand high processing unit. But in sensor based MANETs, there is scarcity of resources. Thus, what

processing unit is required for efficient implementations of cryptography algorithm?, is a major concern.

In [5-8] different methods of key management are proposed. These methods show variation in certificate authority (CA) and/or registration authority (RA) in a network. Network can be know from CA or RA distribution and can have forms like: centralized CA, partial distributed CA, fully distributed CA, identity based, certificate chaining, cluster based, pre-deployment based, mobility based, parallel key, self organized, fully self organized key management techniques. With respect to elliptic curve cryptosystem (ECC), key generation and agreement protocol are: Diffie Hellman key generation and agreement protocol and Menezes Vanstone key generation and agreement protocol [2]. In this work, Diffie Hellman key generation and agreement protocol is selected because of its simplicity in key generation as compare to other methods.

To understand, implement and analyze MANET, Table 1 shows various toolkits and libraries [9-12]. In this work, to analyze the performance of MANET Python language [10], ns3 simulator [11], Shamir's Secret Sharing Scheme [12], Seccure Elliptic Curve Crypto Utility for Reliable Encryption [12] and PyECC[10] are used.

Table 1: TOOLKITS AND LIBRARIES FOR CRYPTOSYSTEM

Toolkits	Avraes,bc,calc,gap,jsaes,jssha256,KANT/KASH, Magma,Maple,Mathematica,MuPAD,SSSS,SECC URE
Libraries	Botan ,Bouncy Castle, borzoi, Catacomb, cryptix , cryptlib,crypto++,Flexiprovider,libgcrypt, LibTomCrypt,LIDIA,MatrixSSL,Mecca,MIRACL ,Mozilla's NSS, NTL , OPENSSL, OpenCDK, PARI-GP, PyCrypto,PyEcc.
Elliptic Curve Libraries	borZai, Cryptlib, Crypto++, LibTomCrypt, Magma, Mecca (viBotan,) MIRACL, PyECC.

This paper is organized as follows. Section 2 presents the introduction to elliptic curve cryptosystem and its forms. In Section 3, processing unit for ECC integrated MANET is calculated, designed and mathematically evaluated to improve the performance of two chosen elliptic curves: Weierstrass and Jacobian forms. Section 4 presents the simulation scenario taken and implementation of key management strategies. Finally, Section 5 concludes the work presented.

Table 2: SYMBOLS AND NOTATIONS

Notation	Description
F	Field
R	Ring
G	Group
GF	Galois Field
GF(N ⁿ)	Extension Field. Where, N and n = {3,4,5,.....so on}
Z _p	Prime field, where p is prime number
P	Prime number
GF(2 ⁿ)	Binary Field, where n={0,1,2,3,.....(n-1)}
W	Bit processor (e.g. 8-bits, 16 bits, 32 bits) i.e. multiple of 8.
E	Elliptic Curve
M	Multiplication
S	Squaring
I	Inverse
GF(p)	Prime field, where p is prime number
[]	Greatest integer function
K	Window size

II. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptography was proposed by Victor Miller and Neal Koblitz independently. According to [3] Well known forms of elliptic curves used in cryptosystem are: (i) Simple Elliptic Curve (Point Addition: 4M+1S+1I and Point Doubling: 7M+2S+1I) (ii) Weierstrass Form of elliptic curve (Point Addition: 2M+1S+1I and Point Doubling: 2M+2S+1I) (iii) Projective coordinates form (Point Addition: 12M+2S and Point Doubling: 5M+6S) (iv) Weighted projective coordinates (Point Addition: 11M+5S and Point Doubling: 1M+8S) (v) Montgomery form (Point Addition: 6M+4S [at once] and Point Doubling: 3M+5S) (vi) Hessian form (Point Addition: 12M and Point Doubling: 7M+1S) (vii) Edward form(Point Addition: 11M+1S and Point Doubling: 3M+4S (viii) Other forms: Toric Form.

FIPS PUB 186-3 recommends fifteen forms of elliptic curves for federal government [16]. These are as follows:

- A pseudo random curve over GF(p) denoted as P-XXX. Where E: $y^2 \cong x^3 - 3x + b \pmod{p}$, cofactor (h)=1, base point (G_x, G_y), order (n), prime modulus (p), output c of SHA-1 algorithm, coefficient b (satisfying $b^2 \cdot c \cong -27 \pmod{p}$). The curve s recommended in this category are: (i) P-192 (ii) P-224 (iii) P-256 (iv) P-384 (v) P-521.
- A pseudo random curve over GF(2ⁿ) denotes as B-XXX. Where E: $y^2 + xy \cong x^3 + x^2 + b \pmod{p}$. Where cofactor (h)=2. The curves recommended in this category are: (i) B-163 (ii) B-233 (iii) B-283 (iv) B-409 (v) B-571.
- A special curve over GF(2ⁿ) called a koblitz curve or anomalous binary curve. Where E: $y^2 + xy \cong x^3 + ax^2 + b \pmod{p}$. Where a=0 or 1, h=2 if a=1 and h=4 if a=0. The curves recommended in this category are: (i) K-163 (ii) K-233 (iii) K-283 (iv) K-409 and (v) K-571.

A. Elliptic Curve Forms

Let P₁(x₁, y₁), P₂(x₂, y₂), P₃(x₃, y₃) be points on E given by

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

Satisfying $4a^3 + 27b^2 \neq 0$, where 'a' and 'b' are the coefficients. This form of equation is referred as prime field equation.

If points P₁(x₁, y₁), P₂(x₂, y₂), P₃(x₃, y₃) are points on E given by

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

in GF(2ⁿ). This form of equation is referred as binary field equation. In following sections, doubling and addition on two different forms of equations one with minimum processing and another with maximum processing are presented as shown in Table 3.

a. Classical Weierstrass form in affine coordinates

1. Prime Field

1. Doubling

$$P_3(x_3, y_3) = 2P_1(x_1, y_1) = \begin{cases} \infty & \text{if } y_1 = 0 \\ \lambda = (3x_1^2 + a) / 2y_1 \\ x_3 = \lambda^2 - 2x_1 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases} \quad \text{if } y_1 \neq 0 \quad (3)$$

2. Addition

$$P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2) = \begin{cases} 2P_1(x_1, y_1) & \text{if } y_1 = y_2 \\ \infty & \text{if } y_1 \neq y_2 \end{cases} \quad \text{if } x_1 = x_2$$

$$P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2) = \begin{cases} \lambda = (y_2 - y_1) / (x_2 - x_1) \\ x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases} \quad \text{if } x_1 \neq x_2 \quad (4)$$

II. Binary Field

Let P₁(x₁, y₁), P₂(x₂, y₂), P₃(x₃, y₃) be points on E given by E: $y^2 + xy = x^3 + ax^2 + b$ in GF(2ⁿ).

1. Doubling

$$P_3(x_3, y_3) = 2P_1(x_1, y_1) = \begin{cases} \infty & \text{if } x_1 = 0 \\ \lambda = x_1 + y_1 / x_1 \\ x_3 = \lambda^2 + \lambda + a \\ y_3 = x^2 + \lambda x + x \end{cases} \quad \text{if } x_1 \neq 0 \quad (5)$$

2. Addition

$$P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2) = \begin{cases} \infty & \text{if } y_1 \neq y_2 \\ 2P_1(x_1, y_1) & \text{if } y_1 = y_2 \end{cases} \quad \text{if } x_1 = x_2$$

$$P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2) = \begin{cases} \lambda = (y_2 + y_1) / (x_2 + x_1) \\ x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \\ y_3 = \lambda(x_1 + x_3) + x_3 + y_1 \end{cases} \quad \text{if } x_1 \neq x_2 \quad (6)$$

Table 3. REPRESENTING COST ESTIMATION OF DIFFERENT CURVES IN PRIME AND BINARY FIELDS.

Elliptic Curves	Prime Field									Binary Field								
	Doubling			Addition			Doubling / Addition			Doubling			Addition			Doubling / Addition		
	M	S	I	M	S	I	(1,1,1)	(1,0,5,0,5)	(1,0,0)	M	S	I	M	S	I	(1,1,1)	(1,0,5,0,5)	(1,0,0)
Classical Weierstrass form in affine coordinates	2	2	1	2	1	1	5 / 4 M	3.5 / 3 M	2 / 2 M	1	2	1	1	1	1	4 / 3 M	2.5 / 2 M	1 / 1 M
Jacobian Coordinates	4	6 *	0	12	4	0	10 / 16 M	7M	12 M	5	5	0	15 **	5 **	0	10 / 20 M	7.5 / 17.5 M	15 / 15 M

*S=4 if a=-3

**M=14, S=4 if a=0 or 1

b. Jacobian Coordinates

I. Prime Field

1. Doubling

Let $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, $P_3(x_3, y_3, z_3)$ are three points represented in jacobian coordinates. The point (x, y, z) in jacobian coordinates form can be represented in affine coordinates as: $(x/z^2, y/z^3)$. Thus, doubling and addition on points: $P_1(x_1/z_1^2, y_1/z_1^3)$, $P_2(x_2/z_2^2, y_2/z_2^3)$, $P_3(x_3/z_3^2, y_3/z_3^3)$ can be represented as:

$$P_3(x_3, y_3, z_3) = 2P_1(x_1, y_1, z_1) = \begin{cases} \left(\begin{array}{l} \infty \\ A = 4x_1^2 y_1^2 \\ B = 3x_1^2 + az_1^4 \\ x_3 = B^2 - 2A \\ y_3 = B(A - x_3) - 8y_1^4 \\ z_3 = 2y_1 z_1 \end{array} \right) & \text{if } y_1 = 0 \\ \left(\begin{array}{l} \infty \\ A = 4x_1^2 y_1^2 \\ B = 3x_1^2 + az_1^4 \\ x_3 = B^2 - 2A \\ y_3 = B(A - x_3) - 8y_1^4 \\ z_3 = 2y_1 z_1 \end{array} \right) & \text{if } y_1 \neq 0 \end{cases} \quad (7)$$

2. Addition

$$P_3(x_3, y_3, z_3) = P_1(x_1, y_1, z_3) + P_2(x_2, y_2, z_3) =$$

$$\left| \begin{array}{l} A_1 = x_1^2 z_2^2 \\ A_2 = x_2^2 z_1^3 \\ B_1 = y_1^2 z_2^3 \\ B_2 = y_2^2 z_1^3 \\ \left(\begin{array}{l} \infty \\ B_1 = B_2 \end{array} \right) 2P_1(x_1, y_1, z_1) \left| \begin{array}{l} B_1 \neq B_2 \\ A_1 = A_2 \end{array} \right. \\ C = A_2 - A_1 \\ D = B_2 - B_1 \\ x_3 = D^2 - C^3 - 2A_1 C^2 \\ y_3 = D(A_1 C^2 - x_3) - B_1 C^3 \\ z_3 = C z_1 z_2 \end{array} \right| \begin{array}{l} A_1 \neq A_2 \end{array}$$

(8)

II. Binary Field

1. Doubling

$$P_3(x_3, y_3, z_3) = 2P_1(x_1, y_1, z_1) =$$

$$\left| \begin{array}{l} \left| \begin{array}{l} \infty \\ z_3 = x_1^2 z_1^2 \\ x_3 = (x_1 + b^{2(m-2)} z_1^2)^4 \\ \lambda = x_3 + x_1^2 + y_1 z_1^4 \\ y_3 = x_1^4 z_3^2 + \lambda x_3 \end{array} \right. \\ x = 0 \\ x \neq 0 \end{array} \right|$$

(9)

2. Addition

$$P_3(x_3, y_3, z_3) = P_1(x_1, y_1, z_3) + P_2(x_2, y_2, z_3) =$$

$$\left| \begin{array}{l} A_1 = x_1^2 z_2^2 \\ A_2 = x_2^2 z_1^3 \\ B_1 = y_1^2 z_2^3 \\ B_2 = y_2^2 z_1^3 \\ \left(\begin{array}{l} \infty \\ B_1 = B_2 \end{array} \right) \left| \begin{array}{l} B_1 \neq B_2 \\ A_1 = A_2 \end{array} \right. \\ (2P_1(x_1, y_1, z_1)) \left| \begin{array}{l} A = A_1 + A_2 \\ B = B_1 + B_2 \\ C = z_1 A \\ D = B x_2 + C y_2 \\ E = B + z_3 \\ z_3 = C z_2^2 \\ x_3 = a z_3^2 + E B + A^3 \\ y_3 = E x_3 + D C^2 \end{array} \right. \\ A_1 \neq A_2 \end{array} \right|$$

(10)

III. PROCESSING UNIT FOR A MOBILE NODE IN MANET

Cryptography algorithms and protocols are implemented [2] in two different domains: (i) over fields and (ii) over real numbers. Fields are further classified as: prime field, binary field, galois field and extension fields. Let 'W'-bit processor is the minimum requirement of a mobile node in MANET to efficiently do the computation for a secure environment. Since MANET is scarce of resources thus it will thwart for large value of 'W'. What could be the minimum and maximum value possible for a sensor based MANET? To answer this question, Let 'a' be any number such that $a \in F_p$. To compute 'a', 'm' is divided by 'W'. Where, 'm' is bit length of 'a'. This will generate possible word length for W-bit processor. Let $t = \text{ceiling}(m/W)$. Now, one single packet is factorized into $B[t-1]$ to $B[0]$ units such that $a = 2^{(t-1)W} B[t-1] + 2^{(t-2)W} B[t-2] + \dots + B[1] + B[0]$. An integer satisfying $2^{(t-1)} \leq n < 2^t$ has 'k' digits to the base 2. According to logarithm [15], the number of digits to manipulate can be calculated using formula $\lceil \log_2 n \rceil + 1$. Now it is clear that amount of time required to do any operation is directly proportionate to number of bit operations. Since cryptography and network security computations are mainly performed in integer, binary and extension fields. Thus the operation required to do computation in these fields are: addition, multiplication, subtraction, division, inverse (additive and multiplicative), modulus and squaring. All of these operations can further be performed using addition and shifting. Thus, suppose a 'k' digit number 'n' need to be multiplied with 'l' digit number 'm'. Then:

$$\text{Time}_{\text{required to multiply n by m}} < \text{Time}_{\text{to manipulate (k)}} \cdot \text{Time}_{\text{to manipulate (l)}} \quad (11)$$

ECC deals with polynomials thus bit operations required to multiply two polynomials $(\sum a_i x^i \text{ and } \sum b_j x^j)$ of degree n_1 and n_2 where $n_2 \leq n_1$ are:

$$W \geq (n_2 + 1)((\log_2 m + 1)^2 + n_2 (\log_2 (n_2 m^2) + 1)) \quad (12)$$

Time required to do these bit operations is:

$$(n_1 + n_2 + 1)(n_2 + 1)((\log_2 m + 1)^2 + n_2 (\log_2 (n_2 m^2) + 1)) \quad (13)$$

The above equations give the minimum value of W per calculated time. To figure out the efficient value of 'W' the field operations can be made faster using computational algorithms. These computational algorithms can faster the process of multiplication and addition in finite field. If 'b' is used to represent the number of exponentiation bits and $l = \lceil \log_2 n \rceil + 1$ denote the bit length of the ECC modulus n. The complexity of one modular multiplication or squaring is equivalent to $a + c \cdot l^2$. Where 'a' and 'c' are constants and $(l-m) \leq K$. Table 4 is showing the Average number of multiplications and complexity calculation for ECC using various efficient processing algorithms.

Table 4. AVERAGE MULTIPLICATION AND COMPLEXITY CALCULATION FOR GENERAL ECC USING FAST COMPUTATIONAL ALGORITHMS.

Algorithms	Average-M	Complexity-ECC
Square & Multiply Algorithm	$(b-1)/2$	$1.5(l-1) \cdot (a+c \cdot l^2)$
K-ary method	$(2^{K-3} + (b-1) / 2^K)$	$(a+c \cdot (l/K)^2) \cdot ((2^{K-3}) + (l/K-1) (2^{K-1}) / 2^K)$
Improved K-ary method	$(2^{K-1} + (b-1) / 2^K)$	$(a+c \cdot (l/K)^2) \cdot (K(l/K-1) + 1 + (2^{K-1}-1) / (l/K-1) (2^{K-1}) / 2^K)$
Sliding window Method	$(2^{(l-m)-1} + (b-1) / 2^{(l-m)-1}) / 2^{(l-m)}$	$(a+c \cdot (l/(l-m))^2) \cdot ((2^{(l-m)-1} - 1) + (2^{(l-m)-1}) / 2^{(l-m)}) + (b-1)(2^{(l-m)-1}) / 2^{(l-m)}$
Iterative Exponentiation	$(b-1)$	$(b-1)$

From Table 3 and Table 4, the minimum and maximum value for processing in ECC can be measured. It is minimum for square and multiply algorithm using classical weierstrass form in affine coordinates and maximum for sliding window method for classical weierstrass form in affine coordinates. However these values varies in between these numbers depending upon number of 1's in bit length.

IV. NETWORK SIMULATION AND ANALYSIS

In order to measure cost which involves processing, encryption/decryption, authentication and key management, small to medium scale network scenario is taken into consideration [18][19]. In this work, Diffie Hellman key generation and agreement protocol is used for key generation and management [17-20]. According to algorithm $O_A = a_A$ and $O_B = a_B$ are the private keys for source and destination respectively. Where, 'a' is large random prime integer. $O_{PA} = \alpha \cdot (x_A, y_A)$ and $O_{PB} = \alpha \cdot (x_B, y_B)$ are the public keys calculated using addition or doubling formulas as discussed in section II. From these keys a shared key is generated i.e. $O_{AB} = a_A \cdot \alpha \cdot (x_A, y_A) = a_B \cdot \alpha \cdot (x_B, y_B)$. In a Mobile Network, Public key is assumed to be known to everyone and private key is kept secret. Shared key's shares are collected from one hop distance nodes and are based on threshold scheme [13]. As proposed in [5], in order to construct a dynamic demilitarized zone, a set of CA and RA is selected. Figure 1 explains the construction of 20 nodes partial distributed key management scenario.

In partial distributed certificate authority scheme, A region is divided into sectors with basis as $[\Pi, \Pi/2, 3\Pi/4, \Pi/4]$. Node coming nearest to origin is considered for CA. Each request for a certificate to CA is passed through RA. More than one registration authorities are selected from different regions to distribute overall load. In [5] a selection criteria function is used to select registration authority. This selection criteria function is based on (i) trust level (ii) stability of node (iii) residual energy and (iv) connectivity degree. Once RA is decided, Nodes given extra responsibility can behave selfishly. Thus to get truthful information, incentive based mechanism is used. This incentive based

mechanism will pay the node for its services. Payment can be in form of energy level, speed, slave agents etc. This type of strategy will attract more nodes to participate in the network and eager to show interest as registration authority. In Mobile Ad Hoc Network, all nodes are mobile thus certificate authority used to provide certificates at central point is also mobile. Partially distributed certificate authority approach uses following steps [1][14][20]:

- (i) Offline Trusted Third Party (TTP) uses a threshold secret sharing scheme to generate shares ($s_1, s_2, \dots, s_n$) of the Distributed Certificate Authority (DCA)'s shared key ' O_{AB} '.
- (ii) Now, it is the responsibility of TTP to distribute shares to ' r ' arbitrary nodes (servers) which will collectively form the DCA. These nodes can generate the public key from shared key using 2-party Diffie-Hellman Key exchange mechanism.
- (iii) Certification generation process starts from offline authority for each node with a private key signature of the DCA. Certificate of each node is also stored on the DCA servers for established security association.
- (iv) Nodes that require a certificate will have to contact atleast ' $t+1$ ' out of ' s ' DCA servers. Any one of ' $t+1$ ' can act as combiner node ' c ' to combine the partial digital signatures from ' $t+1$ ' servers. These partials are calculated using addition and doubling formulas.
- (v) All the nodes in the network including the combiner can verify the validity of signature by using CA public key ' O_B ' that can be easily calculated from shared key.

To efficiently implement this Mobility culture, Mobile Certificate Authority (MOCA) framework is used. MOCA [14] framework is based on threshold cryptosystem [13]. Thus to generate a certificate, shares from ' s ' servers are collected and if a threshold ' t ' out of ' s ' are found correct then certificate is issued. This process of requesting and generating a certificate is processed through CREQ (Certificate Request), CREP (Certificate Reply) and CERR (Certificate Error). These packets are same as used in Mobile Ad Hoc Routing Protocol's route request (RREQ), route Reply (RREP) and route error (RERR). Packets are exchanged either through (i) flooding or (ii) β -unicast. If M, C, S and R stand for MOCA, Combiner, Server and Region respectively then combiner function can be calculates as: $C = \sum M^{S(i)}_{R(i)}$. where, i varies from 1 to s . $M^{S(i)}_{R(i)}$ is decided using selection criteria function. Certificate Retrieval Mechanism in this strategy follows the following steps [1][14][20].

- (i) Any node can initiate the process using broadcast certification request packets to receive a certificate.
- (ii) Any MOCA node (server) from the set of MOCA nodes receives the CREQ packet and answer with a certificate reply containing its partial signature on the certificate.
- (iii) A node ' n ' that receives successfully ' $t+1$ ' valid CREPs in a fixed period. From these replies it reconstructs the full certificate in that fixed period.
- (iv) If number of CREPs is insufficient to generate a valid certificate within expiry of the node's CREQ timer, the process will be considered as failure and the node can initiate another new request.

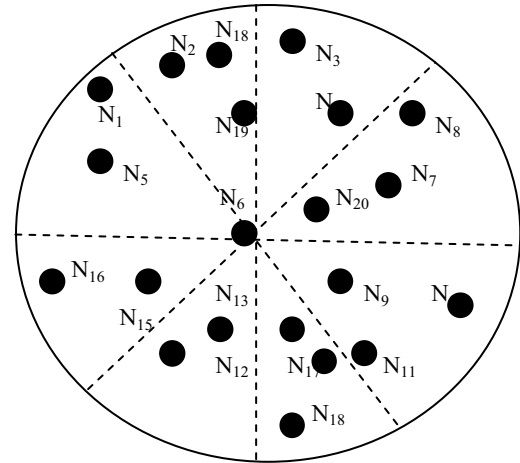


Figure 1: Partial Distributed Network Scenario

Table 5 shows QoS analysis. Throughput is maximum for jacobian coordinates with DSDV protocol and minimum for jacobian coordinates with AODV protocol. Because the complexity of jacobian coordinates is much higher than weierstrass form thus this form is more preferred from of elliptic curve.

Table 5. QoS ANALYSIS OF MANET USING ECC.

Parameters	(Weierstrass form)			(Jacobian coordinates)		
	AODV	DSR	DSDV	AODV	DSR	DSDV
Packets Drop	96	77	65	88	48	32
Throughput	12Mbps	15Mbps	13Mbps	11Mbps	17Mbps	22Mbps
End to End Delay	0.201	0.345	0.561	0.442	0.134	0.348
Jitter	0.0341	0.055	0.103	0.048	0.051	0.036

Figure 2 shows packet drop for jacobian coordinates and weierstrass form of elliptic curve. Packet drop is constantly low for DSDV protocol with jacobian as well as weierstrass form of elliptic curve. Figure 3 shows end to end delay for both forms. End to End delay counts propagation delay, transmission delay and processing delay. End to End delay gradually increases for weierstrass form of elliptic curve and it is minimum for DSR protocol for jacobian coordinates. Because of jacobian complexity AODV is considered to be the lighter protocol with minimum delay for weierstrass form of elliptic curve.

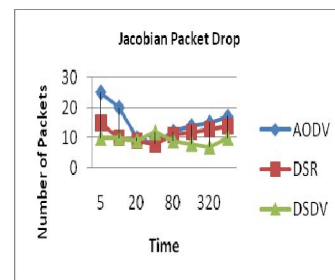


Figure 2a: Jacobian Coordinates

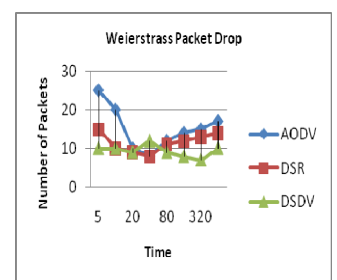


Figure 2b: Weierstrass Curve

Figure 2: Packet Drop for Jacobian Coordinates and Weierstrass Curve

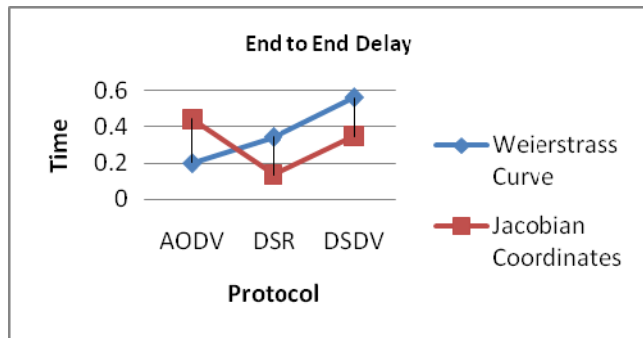


Figure 3: End to End Delay

V. CONCLUSION

In this work, MANET performance is analyzed with ECC using partially distributed key management technique with Diffie Hellman key generation mechanism over different routing protocols. It is found that DSR is the lightest protocol using weierstrass elliptic curve and square and multiply protocol. Whereas, AODV will generate heavy traffic and maximum packet loss using sliding window protocol in jacobian curves.

REFERENCES

- [1] Johann van der Merwe, Dawoud Dawoud and Stephen McDonald, "A Survey on peer to peer key management for Mobile Ad Hoc Networks", *ACM Computing Surveys*, vol. 39, no. 1, Article 1, April 2007.
- [2] William Stallings, "Cryptography and Network Security: Principles and Practices", Prentice Hall, 4th edition.
- [3] Forms of elliptic curves:: <https://www.cosic.esat.kuleuven.be/index.html>
- [4] NIST, "Recommended elliptic curves for federal government use," <http://csrc.nist.gov/>, June 2009.
- [5] A. Rachedi, A. Benslimane, H. Otrak, N. Mohammed and M. Debbabi, "A Secure Mechanism Design-Based and Game Theoretical Model for MANETs", *Journal on Mobile Netw. Appl.*, vol 15, pp.191-204, 2010.
- [6] Srdjan capkun, levente Buttyan and Jean Pierre Hubaux, "Self Organized Public key Management for Mobile Ad Hoc Networks", *IEEE Transaction on Mobile Computing*, vol.2 No.1, 2003.
- [7] Y. Dong, H.W. Go, A.F. Sui, Victor O.K. Li, Lucas C.k. Hui, S.M. Yiu, "Providing Distributed Certificate Authority service in Mobile Ad Hoc Network", *Proceeding of the First international conference on Security and privacy for emerging areas in communication networks*, 2005.
- [8] Yair Amir, Yongdae Kim, Cristina Nita-Rotaru, and Gene Tsudik, "On the Performance of Group Key Agreement Protocols", *ACM Transactions on Information and System Security*, Vol. 7, No. 3, Pages 457-488, August 2004.
- [9] Darrel Hankerson, Alfred Menezes, Scott Vanstone, "Guide to Elliptic Curve Cryptography", Springer, July 2003.
- [10] Python Platform: <http://www.python.org/>
- [11] Ns3 Simulator: <http://www.nsnam.org/>
- [12] B. Poettering project site: <http://point-at-infinity.org>
- [13] Adi Shamir, "How to Share a Secret", *Communication of the ACM*, vol. 22, No. 11, November 1979.
- [14] Seung Yi, Robin Kravets, "MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks" *In Proceedings of the 2nd Annual PKI Research Workshop*, 2003.
- [15] Neal Koblitz, "A course in Number Theory and cryptography", Springer-verlag, 1994.
- [16] FIPS PUB 186-3: <http://csrc.nist.gov/publications/PubsFIPS.html>
- [17] Whit Diffie and Martin Hellman, "New Directions In Cryptography", *IEEE Transactions on Information Theory*, IT-22(6):644-654, November 1976.
- [18] Michael Steiner, Gene Tsudik, and Michael Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication", *ACM Conference on Computer and Communication Security*, Pages 31-37, 1996.
- [19] Le-Hung Vu, Karl Aberer, "Effective Usage of Computational Trust Models in Rational Environments", *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 6, No. 4, Article 24, Pages 24:1-25 2011.
- [20] Frank Lehrieder, Simon Oechsner, Tobias HoBfeld, Dirk Staehle, Zoran Despotovic, Wolfgang Kellerer, Maximilian Michel, "Mitigating unfairness in locality-aware peer-to-peer networks", *International Journal of Network Management*, Volume 21 Issue 1, Pages 21:3-20, 2011.