International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

The Arithmetic Of elliptic Curve for Prime Curve Secp-384r1 using One Variable Polynomial Division for Security of Transport Layer Protocol

Santoshi Pote, B.K. Lande

Abstract: ***In this paper, we present a new method for solving multivariate polynomial elliptic curve equations over a finite field. The arithmetic of elliptic curve is implemented using the mathematical function trace of finite fields. We explain the approach which is based on one variable polynomial division. This is achieved by identifying the plane $F_p \times F_{pp}$ with the extension of F_{p^2} and transforming elliptic curve equations as well as line equations arising in point addition or point doubling into one variable polynomial. Hence the intersection of the line with the curve is analogous to the roots of the division between these polynomials. Hence this is the different way of computing arithmetic of elliptic curve.Transport layer security provides end-to-end security services for applications that use a reliable transport layer protocol such as TCP. Two Protocols are dominant today for providing security at the transport layer, the secure socket layer (SSL) protocol and transport layer security (TLS) protocol. One of the goals of these protocols is to provide server and client authentication, data confidentiality and data integrity. The above goals are achieved by establishing the keys between server and client, the algorithm is called elliptic curve digital signature algorithm (ECDSA) and elliptic curve Diffie-Hellman (ECDH). These algorithms are implemented using standard for efficient cryptography(SEC) prime field elliptic curve secp-384r1 currently specified in NSA Suite B Cryptography. The algorithm is verified on elliptic curve secp-384r1and is shown to be adaptable to perform computation.***

Key words: ***Transport layer, Elliptic curve arithmetic, Polynomial division.***

I. INTRODUCTION

Cryptography is an essential component of modern electronic commerce. With the explosion of transactions being conducted over the Internet, ensuring the security of data transfer is critically important. Considerable amounts of money are being exchanged over the network, either through e-commerce sites, auction sites, online banking, stock threading, and even government.[9]

Communication with these sites is secured by the Secure Socket Layer (SSL) or its variant, Transport Layer Security (TLS), which are used to provide authentication, privacy, and integrity[9].

Revised Manuscript Received on 30 July 2019.
* Correspondence Author
Santoshi Pote*, Electronics Engineering, Ramrao Adik Institute of Technology, Mumbai University, Mumbai, India
B.K.Lande, Electronics and Telecommunication, Datta Meghe College of Engineering, Mumai University, Mumbai,India.

A key component of the security of SSL/TLS is the cryptographic strength of the underlying algorithm used by the protocol. In this paper, we have implemented one variable based elliptic curve algorithms, a unique way of solving computation of elliptic curve which is distinct from that given in [4].

A. Elliptic Curve

Elliptic curves are defined over prime field F_p where p is a prime number. The general form of the elliptic curve which is used in most of the elliptic curve cryptographic application is Weierstrass curve[1]. The general equation form of this curve is

$$y^2 = x^3 + ax + b \pmod{F_p} \qquad (1)$$

where $a, b \in F_p$

Each value of a,b gives a different elliptic curve. All points(x,y) corresponds to (h,k) which satisfies the above equation pulse point at infinity lies on the elliptic curve. Elliptic curve cryptography is asymmetric/public key cryptosystem which is based on two keys public key a private key. The public key is a point on the elliptic curve and the private key is a random number from the field [1][2].

B. Elliptic Curve Arithmetic Operation

The existing approach of elliptic curve arithmetic used in public key cryptography is based on addition and doubling of elliptic curve points over prime field[1][2]. It is represented by following simplified form

- Two points $P=(h_1,k_1)$ and $Q=(h_2,k_2)$ located on Elliptic curve E over F_p. When $p \neq Q$the addition of two points generate the third point by computing equations (2),(3).

$$h_3 = (m^2 - h_1 - h_2)(\text{mod } F_p) \; (2)$$
$$k_3 = (m(h_1 - h_3) - k_1)(\text{mod } F_p)(3)$$

Where $m = \frac{k_2 - k_1}{h_2 - h_1}$

- When P=Q doubling of point generate third point from equation (2) and (3). Where $m = \frac{3h_1^2 + a}{2k_1}$.
- The above approach is based on two variables h and k.The operations involved in the above computations are addition, additive inverse, multiplication, squaring and inversion[1][2]. The proposed approach is simply based on the division of polynomials over finite field.One of the major time consuming finite field operationinversion is not computed in our approach.

II. ONE VARIABLE CONVERSION

A polynomial function $\boldsymbol{f(h,k)}$ over $\boldsymbol{F_p}$ is a polynomial with co-efficients in F_p where the variables h, $\boldsymbol{k}$ take values in $\boldsymbol{F_p}$. Hence such a function is a map from $\boldsymbol{F_p^2}$ to $\boldsymbol{F_{p^2}}$. To convert any equation $\boldsymbol{f(h,k)=0}$ to one variable we can identify $\boldsymbol{F_p^2}$ with the field $\boldsymbol{F_{p^2}}$ by defining a variable $\boldsymbol{z=h+k\theta}$ and treat the equation as an equation over $\boldsymbol{F_{p^2}}$. where $\boldsymbol{\theta}$ is a root of a second degree irreducible polynomial over $\boldsymbol{F_p}$. A system of equations in variables h, $\boldsymbol{k}$ can also be treated as a single variable polynomial system over this extended field. Then the computation of solutions of this system can be performed by one variable polynomial arithmetic over $\boldsymbol{F_{p^2}}$ using Euclidean division. This is in short one variable approach to explained systems of polynomials in multiple variables as developed in this paper[3].

We now describe the above conversion to one variable polynomial. Lets $\varphi(x)$ be a second-degree irreducible polynomial over F_p and $\varphi(\theta)=0$. Define $z=h+k\theta$ then h *and* k are linear function of z which is in F_{p^2}. Hence there exist α_1,α_2 in F_{p^2} such that h= $Tr(\alpha_1 z), k=Tr(\alpha_2 z)$. Then substituting for h , k in equations $f(h,k)=0$ we get $F(z)=f(Tr(\alpha_1 z),Tr(\alpha_2 z)=0$.This way an equation in two variable is treated as an equation in one variable over F_{p^2} . The roots z in F_{p^2} of this equation give the solutions of the original equation, where h, k components give solution of both variables over F_p[5][8].

A. Elliptic Curve Arithmetic in One Variable

Consider an equation $f_E(h,k)=0$ of an EC over $\boldsymbol{F_p}$denoted E and let $F_E(z)$ be the one variable polynomial corresponding to the above one variable conversion over elliptic curve E. Let P = (c, d) be a point on E. Then P also corresponds to $t=c+d\theta$ in $\boldsymbol{F_{p^2}}$and since $f_E(h,k)=0$ is an equation of the elliptic curve over F_p then t is a root of $F_E(z)$ after converting the EC equation to one variable as above. Hence entire E is the set of roots of this polynomial function $F_E(z)$. Now if points P,Q are in E (which corresponds to t, s in one variable), $l(h,k)=0$ is the equation of the line through these points , then t, s are roots of L(z) =0 after one variable conversion of $l(h,k)$. Hence$(z-t)(z-s)$divides gcd $(F_E(z),L(z))$. Since P+Q or [2]P is the reflection of the third point common to E and the line equation, the gcd is exactly of degree at most three and the third root of this gcd gives point $R=(r_h,r_k)$ corresponding to the third root $r=r_h+r_k\theta$. Hence the point addition (respectively doubling) is -R which for non binary F is (r_h, r_k). In this way the EC arithmetic can be completely achieved by polynomial division and gcd computation over F_{p^2}[5][8].

However, the field F_{p^2} is of squared size of F_p, the degree of the EC polynomial $F_E(z)$ is 3p and the degree of the line polynomial is p. Hence the Euclidean division required to compute gcd $(F_E(z),L(z))$ is not likely to be scalable for practically large size of q which is roughly 160 bits or more in size. This is where a following surprising observation comes into picture. Due to lack of a mathematical proof to justify this observation we made this is as conjecture

Conjecture 1. Let $F_E(z)$ be a one variable polynomial corresponding to E over and L(z) be a polynomial corresponding to an equation of a line passing through points P, Q on E (respectively tangent at P) then

$$\gcd(F_E(z),L(z))=\bmod\left(F_E(z),L(z)\right) \qquad (1)$$

A curiosity about this conclusion is that although degree of $F_E(z)$ is 3p and that of L is p the gcd get computed in a single shot by just one division as shown above and no successive calculations of remainders are needed. Therefore, the EC arithmetic by polynomial division using the single variable approach becomes scalable. The above approach is verified on large prime field elliptic curve Secp384r1 which is used for ECDSA and ECDH algorithm. The parameters for secp384r1 recommended by the standard of efficient cryptography (SEC)[3].

III. PROPOSED WORK

The algorithm for point addition and doubling in this approach can be split intothe offline and online computation. The offline computation corresponds to the generation of one variable polynomial elliptic curve equation $F_E(z)=0$. This subsumes computation of constants α_1,α_2. Online computation then corresponds to formation of the line equation L(z)=0 in one variable and computation of the residue as in (1) which returns the gcd according to the observation in the conjecture [7].

A. One Variable Polynomial Approach

1) Find a second degree irreducible polynomial $\varphi(X)$over $F_p$and denotes its roots byθ. points (h, k) *in* F_p^2correspond to z= h + kθin F_{p^2} .

2) **Offline Computation**

- Compute α_1,α_2 in $F_{p^2}$from values of Tr $(\alpha_i z)$ for z=h+kθ.
- The expressions of h, k co-ordinates in z requires computation of constants α_1,α_2 such that

$$h=T_r(\alpha_1 z),k=T_r(\alpha_2 z).(4)$$

where $\alpha_1=a_1+b_1\theta$, $\alpha_2=c_1+d_1\theta$

- Due to linearity of of the trace on F_{p^2}.[6][7] we get

$$T_r(\alpha_1 z)=xT_r(\alpha_1 1)+yT_r(\alpha_1\theta) \qquad (5)$$

$$T_r(\alpha_2 z)=xT_r(\alpha_2 1)+yT_r(\alpha_2\theta)(6)$$

- After substituting for z equal to 1and θin equation (5) and (6) it follows that $\alpha_1$and α_2 satisfy following equations.

$$T_r(\alpha_1 1)=1,T_r(\alpha_1\theta)=0$$

$$T_r(\alpha_2 1)=0,T_r(\alpha_2\theta)=1$$

- Transform equation $f_E(h,k)=0$ by substituting h= $T_r(\alpha_1 z),k=T_r(\alpha_2 z)$.

3) **Online computation**

- Point addition. Let $P=(h_1,k_1),Q=(h_2,k_2)$ be points on E correspondes to t,s in F_p, let $l(h,k)=0$ be the equation of the line through P,Q. Transform $l(h,k)$ to $L(z)$ substituting h, k in terms of z as above.

- Point doubling. For a point $P = (h_1, k_1)$ let $l(h,k) = 0$ denote the tangent to E through P. Translate $l(h,k)$ to $L(z)$.
- Compute $H(z) = mod((F_E(z), L(z))$.

By the above conjecture $H(z)$ is at most a third degree polynomial.

- $\mathcal{M}(z) = H(z)/\chi(z)$ for point addition $\chi(z) = (z-t)(z-s)$ and for point doubling $\chi(z) = (z-t)^2$.
- $\mathcal{M}(z)$ $=(z + h_3 + k_3\theta)$. This gives the third point of intersection between the line and E as $R^* = (h_3, k_3)$.
- Compute $-R^*$ and get point addition or doubling as $(h_3, -k_3)$. The following section described the steps of algorithm on prime curve secp384r1.

IV. ALGORITHM IMPLEMENTATION OVER CURVE secp-384r1

Algorithm1: Point addition and doubling

Input: Ellipticcurve domain parameter [3]

Points: $P=(h_1,k_1)$,$Q=(h_2,k_2)$ which corresponds to t, s ϵ F_p.

Output: $R^*=(h_3, k_3) = P\oplus Q$ or R=2P

- Prime Field (p) = $2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$
- A=(0XFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFF0000000000000000FFFFFFFC)
- B=(0XB3312FA7E23EE7E4988E056BE3F82D19181D9C6EFE8141120314088F5013875AC656398D8A2ED19D2A85C8EDD3EC2AEF)
- Elliptic curve equation

y^2 = x^3 +
394020061963944792122790401001436138050797392704654466679482934042457217714968703290472660882589380018616069731123 16x +
2758019355995970587784901184038904809305690585636156852142870730198868924130986086513626076488374510776543976123 0575

- Base point (G) = 0XAA87CA22BE8B05378EB1C71EF320AD746E1D3B628BA79B9859F741E082542A385502F25DBF55296C3A545E3872760AB7,0X3617DE4A96262C6F5D9E98BF9292DC29F8F41DBD289A147CE9DA3113B5F0B8C00A60B1CE1D7E819D7A431D7C90EA0E5F
- Order (n) = 0XFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC7634D81F4372DDF48B0A77AECEC196ACCC52973

Offline Computation

1) Find second degree irreducible polynomial

$\varphi(x)$ =x^2 + x + 1 Over F_{384}[x].

2) Compute α_1 *and* α_2 to evaluate parameter h,k.

$$h = Tr(\alpha_1 z),\quad k = Tr(\alpha_2 z)$$

- α_1 = 26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958667907737982074879 θ + 13134002065464826404093013366714537935026579756821815555982764468081907257165623443015755362752979333953868991037440
- α_2 =13134002065464826404093013366714537935026579756821815555982764468081907257165623443015755362752979333953868991037439 θ + 26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958667907737982074879
- h = 13134002065464826404093013366714537935026579756821815555982764468081907257165623443015755362752979333953868991037440 θ + 26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958667907737982074880)$z^{3940200619639447921227904010014361380507973927046544666794829340424572177149687032904726608825893800186160697311231 9}$ + (26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958667907737982074879 θ + 13134002065464826404093013366714537935026579756821815555982764468081907257165623443015755362752979333953868991037440)z
- k = 26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958667907737982074880 θ + 13134002065464826404093013366714537935026579756821815555982764468081907257165623443015755362752979333953868991037440) $z^{3940200619639447921227904010014361380507973927046544666794829340424572177149687032904726608825893800186160697311231 19}$ + (13134002065464826404093013366714537935026579756821815555982764468081907257165623443015755362752979333953868991037439 θ + 26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958667907737982074879)z

3) Replace elliptic curve equation in one variable form.

$$F_E(z) = Tr(\alpha_2 z)^2) - Tr(\alpha_1 z)^3 - aTr(\alpha_1 z) - b$$

F_E(z)= 30646004819417928276217031189000588515062019432584236297293117092191116933386454700370095846423618445892360979087359 θ + 35024005507906203744248035644572101160070879351524841482620705248218419352441662514708680967341278223876983976099839)$z^{1182060185891834376368371203004308414152392178113963400038448802127371653144906109871417982647768140055848209193369 57}$ + (26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958667907737982074879 θ + 13134002065464826404093013366714537935026579756821815555982764468081907257165623443015755362752979333953868991037439)$z^{7880401239278895842455808020028722761015947854093089333589658680849144354299374065809453217651787600372321394622463 9}$ + 26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958...

$5866790773798207 4879z^{788040123927889584245580802002872276101594785409308933358965868084914435429937406580945321765178760037232139462246381}$ +

(13134002065464826404093013366714537935026579756821815555982764468081907257165623443015755362752979333953868991037440 θ+ 262680041309296528081860267334290758700531595136436311119655289361638145143312468860315107255059586679077379820748 79)$z^{394020061963944792122790401001436138050797392704654466679482934042457217714968703290472660882589380018616069731123 21}$ +

$26268004130929652808186026733429075870053159513643631111965528936163814514331246886031510725505958667907737982074880z^{3940200619639447921227904010014361380507973927046544666794829340424572177149687032904726608825893800186160697311232 0}$

\+ (θ + 2) $z^{394020061963944792122790401001436138050797392704654466679482934042457217714968703290472660882589380018616069731123 19}$ +

(87560013769765509360620089111430252900177198378812103706551763120546048381104156286771702418353195555969245994024960 θ+ 43780006884882754680310044555715126450088599189406051853275881560273024190552078143385851209176597779846229970 12480)z^3 + 262680041309296528081860267334290758700531595136436311119655289361638145143312468860315107255059586679077379820748 79z^2 + (394020061963944792122790401001436138050797392704654466679482934042457217714968703290472660882589380018616069731123 18 θ + 1)z + 1182181263643477333443002825975456571202283341410387814651958610225703253018700946391100532337519289409616721188 1744

Online Computation

1. Line equation L(z) obtained from random points by substituting x,y in one variable form for point addition
2. P = (h_1, k_1) and Q = (h_2, k_2)
3. P= (5189526572741699611751410419603195384402795421023007858022892058967985911955018528642628133908191522536121516815196, 3883791216729376001794572946586559985998942732851877320035354465645047773772796893580977215638676199981931667734963 9)
4. Q = (1235907809180578983623100984321971513363601388036861122203114097476935575732529807260911254876553716346385499099346 7, 2804372731788538536540906255196042074440664046948870197116205255982340739895102545212370694077398387672954720944591 9)
5. Line equation in one variable form

$L(z) = \big((h_2 - h_1)Tr(\alpha_2 z)\big) - \big((h_2 - h_1)Tr\alpha_1 z\big) - k_1h_2 - h_2h_1)$

L(z)= 8377762629178851700498621920379406204683074592573759319069329976076603343172500857206344681775823134981745472086754 θ + 958597373929361317651764441714229266013293072580191527413041103635183684097472217044620494869430062903575746999523 7)$z^{394020061963944792122790401001436138050797392704654466679482934042457217714968703290472660882589380018616069731123 19}$ + (3102424356721562751178041817976420760039666467789168734887896342816911842832436947184092140648311486687986150102556 5 θ + 120821111011476147601902249676288645544985613322815595506108106027523349780222131323986026691847749405401199790848 3)z + 157734454566503787526385244507433640124553959222545335847847503038621790087078758815113126988571619211750886042939 69

6. Compute H(z) and $\mathcal{M}(z)$

$$H(z) = mod\big((F_E(z), L(z)\big)$$

H(z)= z^3 + (112930496496687251684691863972253008018277723998277153980447581924355323858098525033083987759082125041446279825844 20 θ + 370970262382779890430736943776973792041023063531434264518152107765202892640448358811944767570296766902005694785130 12) z^2 + (748483229460533631199538643307518137410272415229052981521508142277505264944778165800653934493517232010699216100090 8 θ + 190421200167601194349442574151860344642800289600511533184811937840936138973828039144464058931228906214340097833819 8)z + 341717958234176741636241778072640763644617588486913397908430731566647621432887666838533577985990631811350307099591 78 θ + 238584441022451658367102802986360686808773611716730081014226361536594404481406862016656534228692212039774469776715 05

7. and $\mathcal{M}(z) = H(z)/\chi(z)$,

where $\chi(z) = (z - t)(z - s)$

$\chi(z)$ =z^2 + (119223729076098130412032881824612070057634107429234181643809895922175584063147462701610530793571301271743500594290 80 θ + 218534015318469897642966198373207032870409299690738275878942603705083801022165537277955254055852093158616304653036 56)z + 386979079703710723330426317667792412188690104339684429741694620239338567487869755611814462711446382782765188191609 5 θ + 345821439890146566413887040295481126930825500154147813815603647296347354861818039187717500359880923138463639937715 57

8. $\mathcal{M}(z) = z + h_3 + k_3\theta$, h_3, k_3 are root of $\mathcal{M}(z)$
9. $\mathcal{M}(z)$ =z + 387726829384533913395449383149077076011441009273697439016120620044636957509919765621946117848100203788318848962676 59 θ + 152436247064309992787770745403766759170613763840695988639209504060119091618282821533989513514444673743389390132093 56
10. R* = (h_3, k_3), -R*=$(h_3, -k_3)$= (2415838148996347993350196555976693788801836288

6395847804027342998233812609668588175648314736
8144706275226679599 02963,
3877268293845339133954493831490770760114410092
7369743901612062004463695750991976562194611784
8100203788318848962 67659)

11. For point doubling

$$L(z) = (2k_1Tr(\alpha_2 z) - (3h_1^2 + a)(Tr(\alpha_1 z) - (-h_1^3 + (ah_1 + 2b)$$

12. Repeat the above steps with $\chi(z) = (z - t)^2$.

The R* is the point obtained by addition of two points P and Q. The algorithm was also verified for point doubling and scalar multiplication.

Thus, above example give us expansive idea of one variable polynomial division approach. The following section give the overview of how this approach is implemented in elliptic curve digital signature(ECDSA) or Digital signatur algorithm(DSA)[1][2].The above arithmetic is implemented using Sagemath open source software[11].

The computation time required for the above computation in second is given in the following table I.

Elliptic Curve	Point Addition	Point Doubling	Scalar Multiplication
Secp384r1	0.0051	0.0038	2.27

Table I. Computation time in second

The computation time as mentioned in the table I can be optimized by decomposition and parallel computation

V. ECDSA IN TRANSPORT LAYER PROTOCOL

Transport layer protocol provides authentication mechanism, encryption algorithms that used during the secure session. The implementation of ECDSA in TLS security should follow the processes of keygeneration, signing and verification algorithm. In ECDSA the key generation is based on ECC algorithm. Following section gives the implementation of key generation process on prime curve secp384r1.

A. Key Pair Generation

The key pair in ECDSA is generated based on the domain parameters, the domain parameters are listed in section 4 for curve secp384r1.

1. Choose a point $P(h_{p,}k_p)$ on the curve and a random integer $s \in [1, n-1]$.
2. Compute $Q(h_{q,}k_q)$= sP, the point Q is also on the curve.
3. Public key is Q and private key is s.

- Point: $P(h_{p,}k_p)$=G(base point)=
(2624703509579968926862315674456698189185292349
1109213387815615900925518854738050089022388053
975719786650872476732087,
8325710961489029985546751289520108179287853048
8613155947092059024805031998844192244386437603
929473330780865116 27871)
- Random integer : S=(Private key)9173994463960286046443283581208347763186259
9566731244949500321595993962602487865564680326
86736042971441523
- Public key : $Q(h_{q,}k_q)$= S*P
(1691986347862417604007362673301723731418968148
0315808721710466215365596137125833822987836866
307383605967147189561714,
1348729823180250329990779267419277125356054270
5659570467965315331578671894647986626664373641
603174348410513645495100)
- The public key is known to everyone and the private key is a secret key which is difficult to hack to the cryptanalysis.

VI. CONCULSION AND DISCUSSION

What we proposed here is not just a new algorithm, but a new way to look at the problem of solving a set of multivariate polynomial equations over finite field. Our goal in this paper is to examine a different way of solving arithmetic of elliptic curve secp384r1. The scalability of this approach proves in the conjecture (1) due to which this approach is practicable and is beneficial to improve the strength of the cryptographic algorithm which is used for authentication, data confidentiality and data integrity.

ACKNOWLEDGEMENT

Theauthor duly acknowledges with gratitude the guidance of Prof. Virendra Sule, Professor IIT Mumbai, for sharing his valuable knowledge related to the area during the preparation of this manuscript.

REFEREENCES

1. Lawrence C. Washington. Elliptic Curves Number Theory and Cryptography. Champman & Hall CRC press, 2008.
2. J.H. Silverman. The arithmetic of Elliptic Curves, Graduate tex in Mathematics , vol. 106, Springer 1986
3. Standards for Efficient CryptographySEC2: Recommended Elliptic Curve Domain Parameters January 27,2010
4. Ding, Jintai and Gower, Jason E and Dieter S. Schmidt, Zhuang-Zi: A New Algorithm for Solving Multivariate Polynomial Equations over Finite Field, IACR Cryptology
5. Santoshi Pote, Virendra Sule, B.K.Lande. One Variable Polynomial Division Approach for Elliptic Curve over Prime Fields. Computing, Analytics and Security Trends (CAST), IEEE conference, December 2016.
6. Robert J, McEliece. Finite Field for Computer Scientists and Engineer.
7. Rudolf Lidl, Harald Niederreiter. Introduction to finite fields and their applications, Cambridge University press, Melbourne Sydney, 1986.
8. Santoshi Pote, Virendra Sule, B.K.Lande. Journal paper submitted.
9. Behrouz A. Forouzan, Data Communications and Networking, Fifth Edition, Mc Grawhill.
10. Computational Mathematics with SageMath Paul ZimmermannAlexandre CasamayouNathann CohenGuillaume ConnanThierry DumontLaurent FousseFrançois MalteyMatthias MeulienMarc MezzarobbaClément PernetNicolas M. ThiéryErik BrayJohn CremonaMarcelo ForetsAlexandru GhitzaHugh Thomas

AUTHORS PROFILE

Santoshi Poteworking as Associate professor at SNDT University. She received the B.E. degree in electronics engineering from the Mumbai University, India, in 1997, and the M.E. degrees in electronics and telecommunication engineering from the Mumbai University in2005. She is pursuing Ph.D. in the area of finite field algebra and cryptography at RamraoAdik Institute of Technology.

Dr.B.K.Lande working as Professor at DattaMeghe college of engineering.He Received the B.E degree from Amravati engineering College, M.E from Walchand College of engineering. He did his Ph.D. in control system from IIT Mumbai. His current interest in the area of control and communication engineering