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 $F_p \times F_{pq}$ with the extension of $F_p$ 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-384r1 and 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] 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

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

where $a,b \in F_p$

Each value of $a,b$ gives a different elliptic curve. All points $(x,y)$ correspond 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 and 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 $(2),(3)$.

$$h_3 = (m^2 - h_1 - h_2) \pmod{F_p} \quad (2)$$

$$k_3 = (m(h_1 - h_3) - k_1) \pmod{F_p} \quad (3)$$

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

- When $P=Q$ doubling of point generate third point from equation $(2)$ and $(3)$. Where $m = \frac{3h^2 + a}{2k} \pmod{F_p}$.

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 operation inversion is not computed in our approach.

II. ONE VARIABLE CONVERSION

A polynomial function $f(h,k)$ over $F_p$ is a polynomial with co-efficients in $F_p$ where the variables $h,k$ take values in $F_p$. Hence such a function is a map from $F_p^2$ to $F_p$. To convert any equation $f(h,k)=0$ to one variable we can identify $F_p^2$ with the field $F_{pq}$ by defining a variable $z = h + k\theta$ and treat the equation as an
equation over \( F_{p^2} \). where \( \theta \) is a root of a second degree irreducible polynomial over \( F_p \). A system of equations in variables \( h, k \) 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 \( F_{p^2} \) 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. Let \( \varphi(x) \) 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 \( \alpha_1, \alpha_2 \) in \( F_{p^2} \) such that \( h = Tr(\alpha_1z), k = Tr(\alpha_2z) \). Then substituting for \( h, k \) in equations \( f(h,k) = 0 \) we get \( F(z) = f(Tr(\alpha_1z), Tr(\alpha_2z)) = 0 \). This way an equation in two variables is treated as an equation in one variable over \( F_{p^2} \). The roots \( z \) in \( F_{p^2} \) of this equation gives 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 \( F_p \) 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 \( t = c + dh \) in \( F_p \) 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 \( z \), \( h(k,k) = 0 \) is the equation of the line through these points, then \( t, s \) are roots of \( L(z) = 0 \) after one variable conversion of \( h(k,k) \). Hence \( z - t \) divides \( gcd(F_E(z), L(z)) \). Since \( P+Q \) or \( 2P \) 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 \( R = (r_3, r_3) \) corresponding to the third root \( r = r_3 + r_3\theta \). Hence the point addition (respectively doubling) is \( -R \) which for non binary \( F \) is \( (r_3, r_3) \). In this way the EC arithmetic can be completely achieved by polynomial division and gcd computation over \( F_p[5][8] \).

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 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

\[
\text{gcd}(F_E(z), L(z)) = \text{mod} \left( F_E(z), L(z) \right) \tag{1}
\]

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 into 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 \( \alpha_1, \alpha_2 \). 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 denote its roots by \( h, k \) in \( F_p \) correspond to \( z = h + k\theta \). (4)

2) Offline Computation

- Compute \( \alpha_1, \alpha_2 \) in \( F_{p^2} \) from values of \( Tr(\alpha_1z), \alpha_2z \) for \( z = h + k\theta \).
- The expressions of \( h, k \) co-ordinates in \( z \) requires computation of constants \( \alpha_1, \alpha_2, c_1, \theta \) such that

\[
h = Tr(\alpha_1z), k = Tr(\alpha_2z), \tag{4}
\]

where \( \alpha_1 = a_1 + b_1\theta, \alpha_2 = c_1 + d_1\theta \).
- Due to linearity of the trace on \( F_{p^2} \) [6][7] we get

\[
T_r(\alpha_1z) = xT_r(a_1) + yT_r(\alpha_1\theta) \tag{5}
\]

\[
T_r(\alpha_2z) = xT_r(a_2) + yT_r(\alpha_2\theta) \tag{6}
\]

- After substituting for \( z \) equal to \( x \) and \( \theta \) in equation (5) and (6) it follows that \( \alpha_1, \alpha_2 \) satisfy following equations.\[
T_r(\alpha_1) = 1, T_r(\alpha_1\theta) = 0
\]

\[
T_r(\alpha_2) = 0, T_r(\alpha_2\theta) = 1
\]

- Transform equation \( F_E(h,k) = 0 \) by substituting \( h = T_r(\alpha_1z), k = T_r(\alpha_2z) \).

3) Online computation

- Point addition. Let \( P = (h_1, k_1), Q = (h_2, k_2) \) be points on \( E \) corresponding to \( t, s \) in \( F_p \) and 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 \( h(l(h,k)) = 0 \) denote the tangent to \( E \) through \( P \) Translate \( l(h,k) \) to \( L(z) \).
- Compute \( H(z) = \text{mod}(F_E(z), L(z)) \).

By the above conjecture \( H(z) \) is at most a third degree polynomial.
- \( M(z) = H(z)/\chi(z) \) for point addition \( \chi(z) \)
(z - t)(z - s) and for point doubling $\alpha(z) = (z - t)^2$.

- $\mathcal{M}(z) = z + h_3 + k_3 \theta$. 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

Algorithm 1: Point addition and doubling

Input: Elliptic curve domain parameter $E$

Points: $P = (h_1, k_1), Q = (h_2, k_2)$ which corresponds to $t, s \in \mathbb{F}_p$.

Output: $R = (h_3, k_3) \in \mathbb{F}_p \times \mathbb{F}_p$ or $R = 2P$.

1. **Prime Field** ($p = 2^{384} - 2^{256} + 2^{192} + 1$)

2. **Algorithm Implementation**

3. **Elliptic curve equation** $y^2 = x^3 + ax + b$

4. **Point addition**

   - $P + Q = R$.

5. **Point doubling**

   - $2P = R$.

6. **Base point** $G = 57592029525361638141533124686803151072550958667907737982074880x^{394000619947221796807302947622688025893800181606973123}$

7. **Elliptic curve equation $y^2 = x^3 + x + 1$**

8. ** Retrieve $s$ or $R = 2P$**

   - $s = \frac{y_2 - y_1}{x_2 - x_1}$.

9. **Compute** $a_1$ and $a_2$ to evaluate parameter $h, k$.

   - $h = Tr(\alpha_2), k = Tr(\alpha_3)$.

10. **Offline Computation**

    1. Find second degree irreducible polynomial $\phi(x) = x^2 + x + 1$ Over $\mathbb{F}_{2^6}$

    2. Compute $a_1$, $a_2$ to evaluate parameter $h, k$.

11. **Conclusion**

    - $s = \frac{\phi(x)}{x^2}$.

12. **IJE RTE**

    - Published by: Blue Eyes Intelligence Engineering & Sciences Publication

---

**IV. ALGORITHM IMPLEMENTATION OVER CURVE secp-384r1**

**Algorithm 1:** Point addition and doubling

**Input:** Elliptic curve domain parameter $E$

Points: $P = (h_1, k_1), Q = (h_2, k_2)$ which corresponds to $t, s \in \mathbb{F}_p$.

**Output:** $R = (h_3, k_3) \in \mathbb{F}_p \times \mathbb{F}_p$ or $R = 2P$.

1. **Prime Field** ($p = 2^{384} - 2^{256} + 2^{192} + 1$)

2. **Algorithm Implementation**

3. **Elliptic curve equation** $y^2 = x^3 + ax + b$

4. **Point addition**

   - $P + Q = R$.

5. **Point doubling**

   - $2P = R$.

6. **Base point** $G = 57592029525361638141533124686803151072550958667907737982074880x^{394000619947221796807302947622688025893800181606973123}$

7. **Elliptic curve equation $y^2 = x^3 + x + 1$**

8. ** Retrieve $s$ or $R = 2P$**

   - $s = \frac{y_2 - y_1}{x_2 - x_1}$.

9. **Compute** $a_1$ and $a_2$ to evaluate parameter $h, k$.

   - $h = Tr(\alpha_2), k = Tr(\alpha_3)$.

10. **Offline Computation**

    1. Find second degree irreducible polynomial $\phi(x) = x^2 + x + 1$ Over $\mathbb{F}_{2^6}$

    2. Compute $a_1$, $a_2$ to evaluate parameter $h, k$.
The Arithmetic Of Three Variable Polynomial Division for Security

7. and \( M(z) \) is \( H(z) / \chi(z) \), where \( \chi(z) = (z - t) / (z-s) \)

8. \( M(z) = z + h_3 + k_3 \theta \) and \( h_3, k_3 \) are root of \( M(z) \)

9. \( M(z) = z + 38776289384533913954493831407076011440902 73697439016120620446369570099176562194611784 80102378831848896267659 \) + 3524213667463099927877704537667591706137638 40695988632905040601190161828282153398951351 4444673743389390132390356

6. Compute \( H(z) \) and \( M(z) \)

\[
H(z) = \text{mod}((F_z, E_z), L(z))
\]

\[
H(z) = z^2 + (1129039469646678251664961893792253008182777239 8287715398044758719243553238580985250303893775 908812504416627982584420 \theta + 370970262837798904307369437767973290210236035 31434264518521076752029824048438581911447675 29676690200659478513012 \zeta + (74843329460533611995983643075181374120274215 290592815251081422775052644947786508663539449 35172315069926160100908 \theta + 9041220167660194349442574158603446428002896 0051513318481193784903613897 3828039144464058936122861434009783338198z + 341795822341761462477087260746376461578845 8619937979084303715666647674213283876668835357798 59960318113530709959178 + 23858444102245165836710280296366688077367116 1360081014226361365944044184068620165653422 896221203977446977671505

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 = (518952657247169691175141049630195834402795421 023007858002282920589679870855001852462681339 081915225621216185196, 388397121672937600119745729465865599898492732 058717720305354465604577737772968935809772156 387661999811967789463969496)

4. \( Q = (1235907809180578983623100984321971513363601388 03686112220311409479673955573525928076290112548 765573616385934999499799, 2804372731788583536609265196042047446004064 948870197116205529822403749398951025452123706940 7379386776925472094459) \)

5. Line equation in one variable form \( L(z) = \)

\[
(h_2 - h_1 \boldsymbol{\alpha} + \theta (a_2z) - (h_2 - h_1 \boldsymbol{\alpha} + k_3 \beta z - h_3, k_3) \)
\]

\( L(z) = \)

\( 8377762629178851700498621920379406246830754925 57357959106993299976076603343172525908572634648167 5782813498718457426078429 + 801951257401103635369386409497722170462409486 94304629035757499952372z \) + (340200161963449721227904010143618050979 39270544456764829824304257711946703920476688259930816469731231239 +

\( 39402006196344972122790400100413618050797937206446646674829042454721771496703920476688259930816469731231239 +

\( 262680041309296528081860267334290758700535195136 43631111965528936163814513342166860315107255059 586677907737920748797\)

\( 394020061963449721227904001004136180507979372064466674829042454721771496703920476688259930816469731231239 +

\( 07821255303886217900870787588151132698 85761192117508866042939693

\[\text{DOI: 10.35940/ijrte.}\]

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

Retrieval Number: B1875078219

DOI: 10.35940/ijrte.B1875.078219

4773
For point doubling
\[ L(z) = (2k_1 Tr(a_2 z) - (3h_1^2 + a) Tr(a_1 z) - (-h_1^3 + (a h_2 + 2b)) \]

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

The \( \text{R}^* \) 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 signature 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 key generation, 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 keys is \( Q \) and private key is \( s \).

- Point: \( P(h_q, k_q) = G(\text{base point}) = (2624703509579968926862315674456981891858292349\) 11092133878156159090255188547380500809202388053 975719786650872476732087, 8325109614989985546751289520108179287853048 8613159947092059204805031998844192244386437603 9924973307086511627871)

- Random integer : \( S = (\text{Private key}) 91733994463960286604644323851208347763186259 956673124494590032159599362602487865564680326 8673602971441523 \)

- Public key : \( Q(h_q, k_q) = sP \) = (16919863478624176040473276303172731418968148 03158087217010466215365596131725833822987386866 307386306967174189561174, 134872982318052302990779267419277125356054270 5659504769653153315786718946798666664373641 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. CONCLUSION 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

The author 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.

REFERENCES

1. Lawrence C. Washington, Elliptic Curves Number Theory and Cryptography, Chapman & 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 Cryptography SEC2: 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 MalteyMathias MeulienMarc MezzarobbaClément PernetNicolas M. ThiéryErik BrayJohn CremonaMarcello ForetsAlexandru GhitzaHugh Thomas

AUTHORS PROFILE

Santoshi Potaworking 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 in 2005. 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.