



Paper Type: Original Article

Elliptic Curve Cryptography for Bitcoin Security

Bindu V A^{1*} , Manju Somanath²

¹ Rajagiri School of Engineering & Technology, Cochin-682039, Kerala, India; binduva@rajagiritech.edu.in.

² National College (Affiliated to Bharathidasan University), Tiruchirappalli – 620002, Tamil Nadu, India; manjusomanath@nct.ac.in.

Citation:

Received: 24 February 2025

Revised: 16 April 2025

Accepted: 07 May 2025

V A, B. & Somanath, M. (2025). Elliptic curve cryptography for Bitcoin security. *Risk Assessment and Management Decisions*, 2(2), 131-143.

Abstract


Mathematics has addressed many complex problems in the past and is consistently providing avenues for present-day security problems, too. This time, we understand the interesting Mathematics powering the security structure for Bitcoin. The complex Mathematics that goes on provides much more and ensures a critical component in the security apparatus implemented in the Bitcoin protocol. The solution is addressed by the Elliptic Curve representation and the science of cryptography fortified by Elliptic Curve Cryptography (ECC). We attempt to present the Elliptic curve secp256k1 $y^2 = x^3 + b$ and its relation to security enforced for the Bitcoin network. For easier understanding, the order 113 is identified to present this integration.


Keywords: Elliptic curve, Bitcoin security, Elliptic curve cryptography, Elliptic curve cryptography secp256k1 curve, Elliptic curve cryptography mathematics.


1 | Introduction

In this paper, we are extending the know-how of Elliptic Curve Cryptography (ECC) to one of the most prolific implementations of cryptography, i.e., Bitcoin. Cryptography depends on a certain approach to generate public and private key value pairs over a finite field. It is noted that ECC provides a robust solution to these specialized needs of Bitcoin security [1].

Bitcoin is facilitated through the usage of Bitcoin wallets, which will take many forms, like online wallets, hardware wallets, or mobile wallets. Once the Bitcoin wallet is installed on our system, we are ready to make transactions through Bitcoin. We can view it like the email address, which can be shared with friends and others so that they can pay you and vice versa. When the Bitcoin wallet is installed, it generates a unique address for us. ECC is the use of elliptic curves to generate public and private key pairs over a finite field. An elliptic curve is of the form shown [2].

 Corresponding Author: binduva@rajagiritech.edu.in

 <https://doi.org/10.48314/ramd.vi.62>

 Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

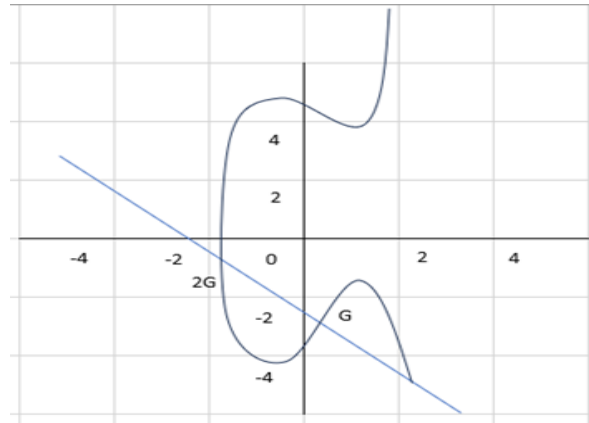


Fig. 1. The Elliptic curve that represents points over a finite field.

As we learn more about ECC, we find that it is derived over a finite field, hence the values of the x and y axes will always have a limit. The concept behind using the ECC is that of the complexity involved, which arises from the following facts.

- I. It is comparatively very easy to plot the two points and is computationally feasible.
- II. It is virtually impossible to compute the other point due to the implementation of asymmetric cryptography over this curve.

The point that can be easily plotted is the public key, and the other point, which is attributable to the Elliptic Curve computation, is designated as the private key [1], [3].

2 | Definitions

Bitcoin private key

It is a large (Typically 256 bits) number which is also called a secret number. It is like a KEY to a secret door. It is used for authorizing and sending bitcoins [4].

Bitcoin public key

It is another large number that can be shared with others because we need to receive bitcoins.

Bitcoin address

It is a hashed, smaller version of the private key.

Examples

A typical real-world representation of the above-defined terms is illustrated below.

Account number/Bitcoin address

1Eeq4DALDGsdgBZerddGoYpCohdrej

Private key

TPPeReG684+tpB/e/JdhUKQQXQBfNN6r3FjQ8Yyi6xvWrvt+zR6UOVSmMrzuGm+4
EKAHNru83HM90AFIm/I0RUWNeZXFtT+0aBJWWuGrsykEVcarXfLSiKtuzAW3SyuR

Public key

p/FzszkemsIhjXh9knIwPnbTErInzTQesOhtMeMT+6y9qLyzmWGRyZONpJcjr70
JPCIO1oVTNM8dHAstN5oePa0QhIdLrMuJlz8CfSsEHYblOQkcu3Sx94vEyWSCt+8
GPMakFrvFAfia56RsWq2YummmFxrTQ6xULnP0cf5Zcv1eadBAgMBAAGgZTBjBgkq

To understand the concept of “public” and “private” keys, we will visualize a money transaction between “User A” and “User Z” [5].

2.1| Send/Receive Bitcoin

- I. “User A” want to send Bitcoin to “user Z”.
- II. “User A” will share their public key with “user Z”.
- III. “User A” opens their wallet and indicate to send 1 BC to “user Z”.
- IV. “User A’s” wallet will generate a transaction message with details as TX_MSG (Sender info., recipient Info., amount).
- V. “User A’s” wallet uses the private key and groups the TX_MSG to generate a digital signature that is Unique to this transaction.
- VI. It also creates a file called the TX_FILE.
- VII. This TX_FILE is pushed as a broadcast to the Bitcoin network to various NODES.
- VIII. Each node will broadcast again to its nodes until the recipient, i.e., “User Z,” uses their public key and decrypts the file to receive the Bitcoin amount [1].
- IX. During this whole time, the file is retained in the MEMPOOL, i.e., the memory pool.

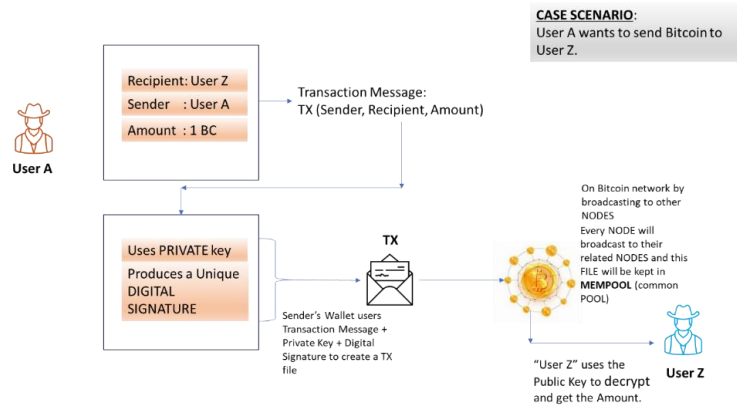


Fig. 2. A simple flow showing the steps involved in Bitcoin security.

It is important to note that in instances like this, as applicable to Bitcoin security, it is virtually impossible to generate a private key from the public key, i.e., reverse engineer. Not possible to reverse engineer means that the process holds good and is easy to implement in one direction, but difficult to do the reverse process.

3| Preliminaries

An elliptic curve for ECC purposes is a plane curve over a finite field that is made up of the points satisfying the equation $E_p(a, b): y^2 = (x^3 + ax + b)$, where the discriminant $\Delta = -16(4a^3 + 27b^2) \neq 0$. If P and Q are two points on the elliptic curve $E_p(a, b)$ then, point addition can be done as follows [6]. Assume $P + Q = (x_3, y_3)$ then:

$$x_3 = (\lambda^2 - x_1 - x_2)(\text{mod } p).$$

$$y_3 = (\lambda(x_1 - x_3) - y_1)(\text{mod } p).$$

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

4 | Method of Analysis

4.1 | Section A: Mathematical Interpretation

Bitcoin implements a possible form of an Elliptic curve as in *Fig. 3*.

The equation for an elliptic curve has the form of $y^2 = x^3 + ax + b$, where a and b are arbitrary constants. It can be noticed that these constants play a great role in shaping the elliptic curve and providing the much-needed complexity [7].

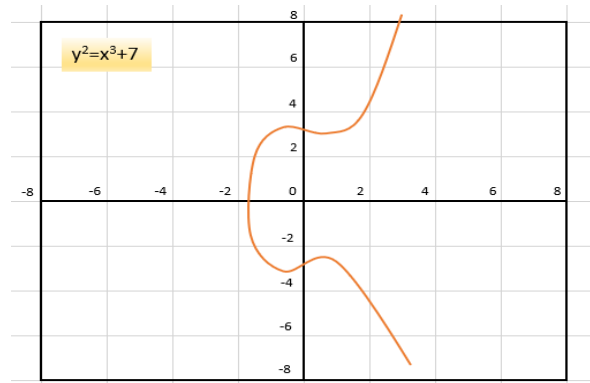


Fig. 3. The secp256k1 ECC curve represented by the appropriate equation.

The elliptic curve equation $y^2 = x^3 + ax + b$ is reduced to $y^2 = x^3 + b$ by having the values $a = 0$, $b = 7$ as in the bitcoin curve representation. This representation of the Elliptic curve, also known as Secp256k1, is used to generate the key pairs (Public and private keys).

This elliptic curve representation for Bitcoin has three essential features.

- I. Every point along the curve has a mirror coordinate, the y coordinate.
- II. If we plot/draw a line through the curve line, we will get two points of intersection, P and Q . The third intermediate point that is intersected is called point Z [8].
- III. This point Z is mirrored to get Z' .

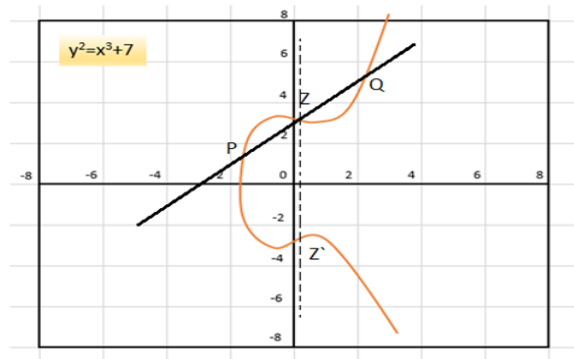


Fig. 4. The figure shows the method for inverse mirroring.

This has a special representation as illustrated in *Fig. 5*. You will notice that the points A , $2A$, and $3A$ plotted there have their mirror co-ordinates, which are extensively used in the ECC security feature.

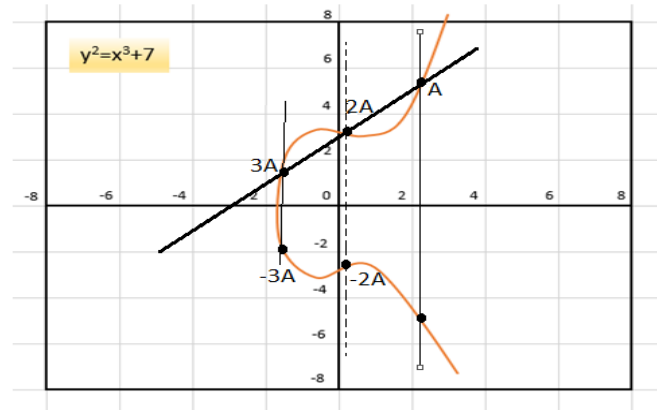


Fig. 5. The secp256k1 curve showing representations of adding and doubling.

These point representations are therefore called the algebra of elliptic curves. Let's see what we mean by the algebra of elliptic curves. The elliptic curve represents the following algebraic operations:

- I. Addition of points along ECC represented by $(A+B)$, i.e., $A+A = 2A$.
- II. Subtraction of points represented by $(A-B) = (A+(-A))$.
- III. Doubling of the points is called multiplication by 2, i.e., $2*A$ [9].

Illustration of elliptic curve algebra

- I. Get the point representation for $8A$.

- First calculate $2*A = 2A$
- Calculate $2*2A = 4A$
- Calculate $2*4A = 8A$

- II. Get the point representation for $10A$.

- First calculate $2*A = 2A$
- Calculate $2*2A = 4A$
- Calculate $2*4A = 8A$
- Add $8A+2A$ to get $10A$

Note: You can multiply, double a point by any integer, but it is difficult to get the original integer value. This is the crux of having Elliptic curves for cryptography purposes. The best part is that it works for exceptionally large numbers, too.

4.2|Section B: Implementing the Bitcoin Elliptic Curve

As a first step, let us consider the following parameters for the curve $y^2 = x^3 + b$. We will consider the following scenarios:

Case 1. The ECC points are spread over a prime value of 11.

Case 2. The ECC points are spread over a prime value of 113.

Case 3. The ECC points are spread over the actual prime value of

115792089237316195423570985008687907853269984665640564039457584007908834671663 as used in the Elliptic curve for Bitcoin security. This is one such value. Let us get into the details of the various cases indicated above [10].

The ECC points are spread over a prime value of 11 (Case 1). The values of the secp256k1 curve are $a = 0$, $b = 7$, and $p = 11$.

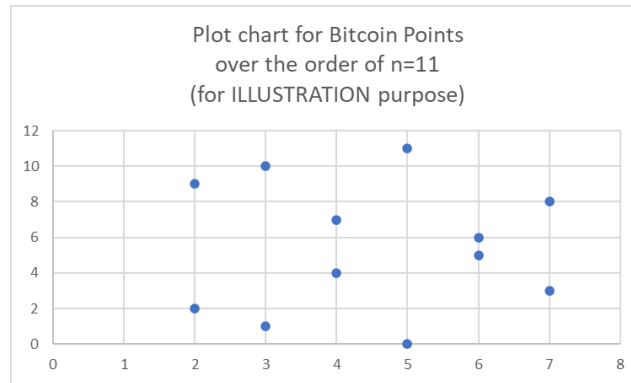


Fig. 6. The ECC representation curve for the order $n=11$.

Note the points that are plotted and fall well within the allocated prime value over which it has been calculated and plotted. We took this prime number to showcase the possible range of values that are calculated and plotted, which will increase the complexity of the ECC security representation. The ECC points are spread over a prime value of 113 (Case 2). The values of the secp256k1 curve are $a = 0$, $b = 7$, and $p = 113$.

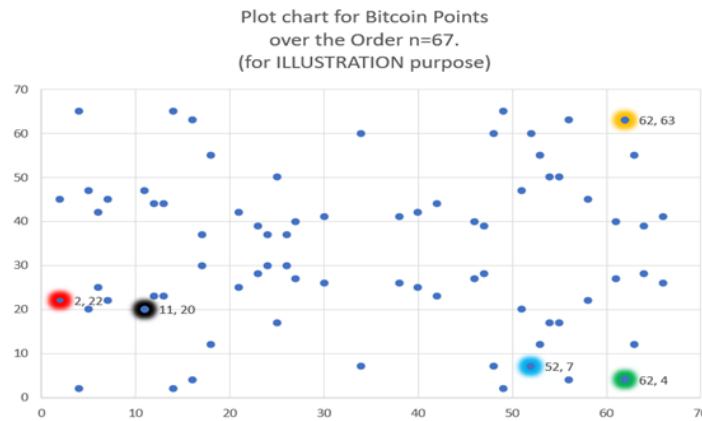


Fig. 7. The ECC secp256k1 curve for the order $n=79$.

The ECC points are spread over the actual prime value of (Case 3). 115792089237316195423570985008687907853269984665640564039457584007908834671663 as used in the Elliptic curve for Bitcoin security. The values of the secp256k1 curve are $a = 0$, $b = 7$, and $p = 115792089237316195423570985008687907853269984665640564039457584007908834671663$.

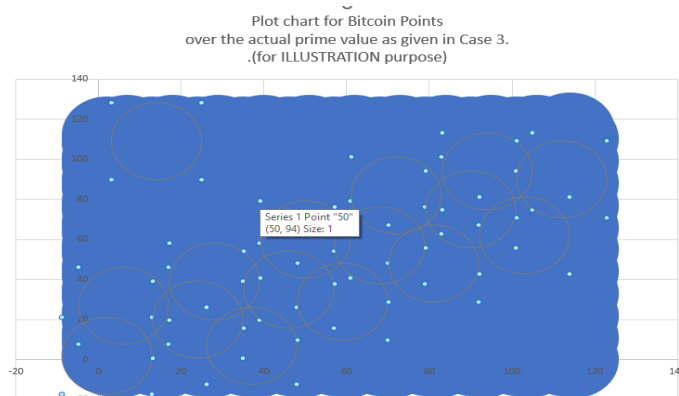


Fig. 8. The ECC curve is simulated for an exceptionally large order.

With the prime value being an extremely large representation, we created this chart as a simulation. It indicates the number of possible values that could be used to map to the public and private key pairs.

This is one of the reasons being that the unearthing of the private key from the public key is virtually impossible, i.e., reverse engineering is complex and time-consuming.

5 | Elliptic Curve Cryptography Encryption and Decryption for Bitcoin Security: An Implementation Perspective

The elliptic curve digital signature algorithm, also called ECDSA, employs the algebra that we have seen earlier for the sum of points, doubling of points, and so on. The important features are:

- *A message can be signed using the private key. The concept behind this is to know that the Digital Signature belongs to User A, since it is associated with the public key.*
- *Let us assume that the message is TX_MSG= "Sending 1 BC "bc1qj89046x7zv6pm4n00qgqp505nvljnf6xfznyw" (Which is the Wallet address of the recipient user Z).*

The following steps illustrate, on a high level, the working of the Bitcoin security implemented through the Elliptic curve.

- I. First, the point (G) is identified, which lies on the Elliptic curve. Point (G) is called the "generator point".
 - *Generator point is a standard value on the Elliptic Curve.*
- II. Generate a random integer which is user A's private key.
 - *The private key is a secret value kept by user A.*
- III. Generate a digital signature by using user A's private key.
 - *The digital signature is a secret value, and nobody can find it.*
 - *The important feature about this digital signature is that, on being computed and being valid, it does not show anywhere on the Elliptic curve.*
- IV. Now, the private key is multiplied by point (G) to obtain the public key.
 - *Public key = private key * G*
 - *public key is shared with everyone on the Bitcoin network.*
 - *There is no restriction on the public key in sharing with others.*
- V. Note that this public key also lies on the Elliptic curve.

6 | Steps in Bitcoin Security Implementation

The figure shows the illustrative steps that are involved in the process of Bitcoin security implementation.

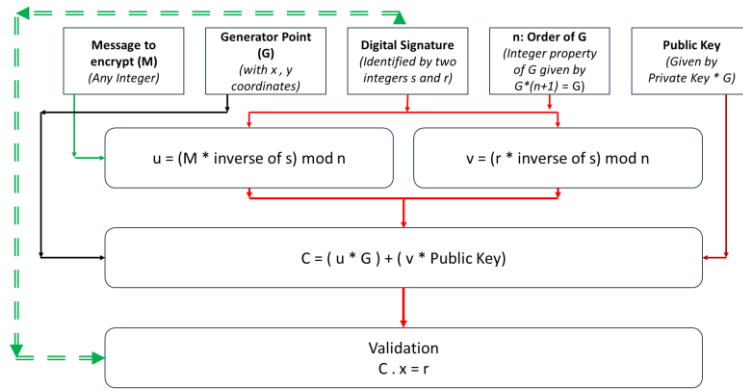


Fig. 9. Steps to implementing Bitcoin security.

The explanation for the components, as shown in the figure, is given below.

- I. The first-level components belong to the owner (i.e., user A) and contain the knowns.
- II. The second-level components are derived from the:
 - Message (M),
 - Digital signature (r, s), and
 - Order of G given as “ n ”.
- I. The third level component is the encryption process.
- II. The fourth level component resembles the decryption once the recipient (i.e., user Z) gets the message.

7 | Bitcoin Implementation – Illustrative Step

Example references www.coindesk.com.

Consider the secp256k1 – Elliptic curve given by $y^2 = x^3 + b$.

Here we have $a = 0, b = 7$.

Let us have the order of G given by n as 79 and the prime modulus as 67.

Base point is $G = (2, 22)$. Shown in Fig. 7 in RED HUE.

Let us also consider a private key value between 1 and $(n-1)$, i.e., $d = 2$.

Step 1. Find the public key.

$$\lambda = \frac{(3 * 2^2 + 0)}{(2 * 22)} \text{ mod } 67.$$

$$\lambda = \frac{(3 * 4)}{(44)} \text{ mod } 67.$$

$$\lambda = \frac{12}{44} \text{ mod } 67.$$

By multiplicative inverse $44^{-1} = 32$

$$\lambda = 12 * 32 \text{ mod } 67 = 49.$$

Step 2. Calculate the public key

Compute $2G$ as

$$x_2 = (49^2 - 2 * 2), \text{ mod } 67 = 2397, \text{ mod } 67 = 52.$$

$$y_2 = (49 * (2 - 52) - 22), \text{ mod } 67 = (-2472), \text{ mod } 67 = 7.$$

This gives us the public key as (52, 7). Shown in *Fig. 7* in blue HUE.

Step 3. Digitally sign the data with the private key from the earlier steps. We have

I. Order (n) = 79

II. Generator point (G) = (2, 22)

– Is referred to as the base point.

III. Private key (d) = 2

IV. For ease of understanding, we will set our data to 17.

V. The next step is to select a random number (k) between 1 and (n-1), i.e., between 1 and 78.

– Let us select this number $k = 3$.

VI. We will need to calculate the point (x, y).

(x, y) given as 3G.

Therefore (x, y) = 3G

Which gives (x, y) = G + 2G.

$$(x, y) = (2, 22) + (52, 7),$$

$$(x, y) = (62, 63),$$

$$x = 62, \text{ and } y = 63.$$

(x, y) is shown in *Fig. 7* in the orange HUE.

Step 4. Find the values of “r” and “s”

First, find “r”

$$r = x \text{ mod } n = 62 \text{ mod } 79 = 62.$$

Next, find “s”

$$s = \frac{(z + r * d)}{k} \text{ mod } n,$$

$$s = \frac{(17 + 62 * 2)}{3} \text{ mod } 79,$$

$$s = 47 \text{ mod } 79 = 47.$$

Hence, the value of (r, s), which is our digital signature, is given by (62, 47). This will never be shown in the curve because it is a secret key that is computed.

Step 5. Verifying the digital signature with the public key. The detailed process in concept is given in the shaded box.

Algorithm 1. Verifying a digital signature using the public key—a step-by-step conceptual guide.

Verify the values of “r” and “s” are between 1 and (n-1).
 Calculate $w = s^{-1} \bmod n$.
 Calculate $u = z * w \bmod n$.
 Calculate $v = r * w \bmod n$.
 Calculate point $(x,y) = uG + vQ$.
 Verify the validity of the signature
 $r = x \bmod n$ (from previous calculation step)
 If the calculated value of “r” does not match the original value of “r”, then the
 Signature does not match, hence NO ACCESS will be granted.

According to the outlined steps, we have

I. $z = 17$ (This is the data we identified for our case scenario)

II. $(r,s) = (62,47)$

- This computed value is the digital signature.
- It does not appear as any point on the Elliptic curve.
- This value is always kept a secret.

III. $n = 79$.

- The complexity of the Bitcoin network arises from the value of the order n .
- The larger the value, the larger the complexity.
- This value also imparts an unknown uncertainty when an attacker attempts to break through the security.

IV. $G = (2,22)$.

- This is our identified base point.

V. $Q = (52,7)$.

- This is the computed public key.

We find, $(r,s) = (62,47)$ is between 1 and 79 (Hence, it is considered a valid step), now, to calculate “w”.

$$w = s^{-1} \bmod n,$$

$$w = 47^{-1} \bmod 79,$$

$$w = 37.$$

To calculate “u”

$$u = zw \bmod n,$$

$$17 * 37 \bmod 79,$$

$$u = 629 \bmod 79 = 76.$$

To calculate “v”

$$v = rw \bmod n,$$

$$62 * 37 \bmod 79 = 3.$$

Now, calculate point (x,y) which is given by

$$(x,y) = uG + vQ.$$

To do this manually is very difficult and complex. We will use our illustrative example of using a lower order to break up the steps. In the original sequence using the order 115792089237316195423570985008687907853269984665640564039457584007908834671663 will take billions of steps to identify the value.

$$uG = 76G,$$

$$uG = 2(38G),$$

$$uG = 2(2(19G)),$$

$$uG = 2(2(G + 18G)),$$

$$uG = 2(2(G + 2((9G))),$$

$$uG = 2(2(G + 2(G + 8G))),$$

$$uG = 2(2(G + 2(G + 2(4G))),$$

$$uG = 2(2(G + 2(G + 2(2(2G)))).$$

Substituting the value of G will give

$$uG = 2(2(G + 2(G + 2(2(2(2,22))))),$$

$$uG = 2(2(38,26)),$$

$$uG = 2(27,40) = (64,4).$$

We can see that “uG” is shown in *Fig. 7* in the green HUE.

Similarly, do the calculation for “vQ”

$$vQ = 3Q,$$

$$vQ = Q + 2Q,$$

$$vQ = Q + 2(52,7),$$

$$vQ = (52,7) + (25,17),$$

$$vQ = (11,20).$$

The value “vQ” is shown in *Fig. 7* in the black HUE. Putting these individual values together, we have

$$C = (x,y) = uG + vQ,$$

$$(x,y) = (62,4) + (11,20),$$

$$(x, y) = (62, 63).$$

With this step, we will be able to verify if the stipulated condition is met.

$$r = x \bmod n,$$

$$62 = 62 \bmod 79,$$

$$62 = 62.$$

Since the computed value “ $x \bmod n$ ” matches the value of “ r ”, we prove that the signature is valid. The signature being valid ensures that the “user Z” gets the Bitcoin.

7 | Conclusion

We have seen the evolution of security constructively involving cryptography. It should be noted that the most prominent security implementation was that of RSA, which is hugely dependent on large prime numbers. The ECC emphasises the mathematical concept of elliptic curves to achieve the security implementation. Additionally, the keys associated with ECC are not as large as those of the RSA security implementation. With the advent of ECC, the biggest gain has been that of Bitcoin. ECC has given the edge and teeth to implement security and taken it to a new level. The combination of encryption with digital signatures has made it more relevant. As Bitcoin gains more importance and with more virtual currencies making their foray, the focus is more on the following aspects:

- I. Wallets: Wallets to be more heavily encrypted and to minimize the vulnerable points.
- II. Minimizing attacks: Bitcoin has seen more coordinated attacks, and with more complex implementations of ECC, it should help in mitigating the risks in dealing with Bitcoin.
- III. ECC has changed the way security is implemented. The implementation has become easier and has resulted in improved security.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

All data are included in the text.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

- [1] Trappe, W. (2006). *Introduction to cryptography with coding theory*. Pearson Education India. <https://www.amazon.com/Introduction-Cryptography-Coding-Theory-2nd/dp/0131862391>
- [2] Stallings, W. (2017). *Cryptography and network security*. Pearson Education India. <https://www.amazon.com/Cryptography-Network-Security-7Th-Stallings/dp/9332585229>
- [3] Bindu, V. A. (2024). Elliptic curve cryptography using a diophantine triple, constructed through tetradecagonal numbers. *International journal of intelligent systems and applications in engineering*, 12(22), 1524. <https://www.ijisae.org/index.php/IJISAE/article/view/6695>
- [4] Manju Somanath, J. K. and K. R. (2021). Cryptographic algorithm based on prime assignment. *International journal for research in applied science engineering technology (IJRASET)*, 10(1). <https://B2n.ir/qk4267>

- [5] Kannan, J., Somanath, M., Mahalakshmi, M., & Raja, K. (2022). Encryption decryption algorithm using solutions of Pell equation. *International journal of mathematics and its applications*, 10(1), 1–8. <http://ijmaa.in/>
- [6] Bashmakova, I. G. (1974). *Diophantus of Alexandria, arithmetics and the book of polygonal numbers*. Nauka, Moscow. <https://www.sciepub.com/reference/301065>
- [7] L, D. (1952). *History of the theory of numbers, volume II: Diophantine analysis*. Chelsea Publishing Company, New York. <https://www.scirp.org/reference/referencespapers?referenceid=3721276>
- [8] Dujella, A., & Petričević, V. (2008). Strong Diophantine triples. *Experimental mathematics*, 17(1), 83–89. <https://B2n.ir/hk6479>
- [9] Gopalan, M. V., & Srividhya, G. (2012). Two special Diophantine triples. *Diophantus journal of mathematics*, 1(1), 23–27. <https://www.saspublishers.com/article/2532/download>
- [10] Gopalan, M. A., Sangeetha, V., & Somanath, M. (2014). Construction of the Diophantine triple involving polygonal numbers. *Scholars journal of engineering and technology*, 2(1), 19–22. <https://saspublishers.com/article/2436/10.36347/sjet>