Ralph's elliptic curve cryptography

Ralph is exploring elliptic curve cryptography based on discrete log as an alternative to prime factoring (such as employed with RSA cryptographic systems). Elliptical curves are based on finite field congruence such that

$$x^3 + ax + b \equiv y^2 \pmod p$$

A generator $G = \{x, y\}$ which satisfies the above congruence is employed for the encoding. The discrete log problem is utilized as it is easy to compute K from $G^d \equiv K \pmod p$ but very difficult to determine d from K, p, and G. The encryption scheme works as follows.

Encryption scheme

1. The parties select a common encoding scheme and common knowledge parameters: a, b, k, n, p.

2. Find a base solution or generator to the elliptic curve $G = \{x_0, y_0\}$.

3. The receiver randomly generates a private key $d < n$, then distributes a public key $G^d \equiv K \pmod p$.

4. The sender randomly generates a private key $r < n$. For each character, the sender encodes the alpha-numeric character, transcribed to an integer m, as $em = \{m * k + j, y\}$ $(j, = 1, 2, \ldots)$ such that y can be found that creates a point on the elliptic curve for $k \geq 10$.That is, first try $j = 1$ if y can be found stop, if not, try $j = 2$, and so on. The sender finds the smallest j on the elliptic curve and shares its value with the receiver.[1] This step is the most time-intensive (but there are tricks for speeding it up).

5. The sender encrypts the encoded message in two quantities: $G^r \equiv C_1 \pmod p$ and $K^r + em \equiv C_2 \pmod p$.

6. The receiver decrypts the message by $G^{dr} + em - G^{dr}. \equiv C_2 - C_1^d \pmod p$ to recover dm. The integrity of this recovered point is checked to see if it resides on the elliptic curve. If not, the receiver requests the message to be resent. If so, the receiver recovers the message as $(dm_x - j) \times k^{-1} \equiv m \pmod p$ where dm_x refers to the x coordinate of the point dm.

7. The receiver decodes the message converting the received number back to its alpha-numeric character.

Suggested:

Suppose "ok" is encoded as $m_1 = 25, m_2 = 21$, $p = 2531, a = 40, b = 300, k = 10, n = 727$.

1. Verify $G = \{5, 25\}$ satisfies elliptical curve congruence.

2. Encrypt each character using the receiver's public key K where $d = 560$ and $r = 24$. (You might explore other randomly chosen private keys d and r.)

3. Decrypt each character in the message.

[1] Alternatively, the sender picks k so that $j = 1$ and this is understood by the receiver.