TEXT ENCRYPTION WITH IMPROVED ELLIPTIC CURVE CRYPTOGRAPHY

ABSTRACT

The security of data encrypted with an encryption algorithm should be guaranteed such that it is never easy for a third party to recover the message from the encrypted data. To this effect, ECC has been a reliable option. However, the base equation that defines the security of *Elliptic Curve Cryptography* (ECC) is in the form of a linear equation with one unknown which is easy to solve. The ease with which this equation can be solved is a weak point in the algorithm. Thus, the aim of this research work is to improve the security of ECC by improving the nature of its base linear equation.

Elliptic curve arithmetic was used to develop the improved model. The encryption process was specifically targeted and improved from single to double encryption using separate encryption constant for each round of encryption. Simulation was done using a 256 bits key size on selected number of character inputs. Java programming language was used to simulate the model on Net Beans IDE. Results of the research show that despite a longer key size of 256 bits and double encryption process, the improved ECC performed better than the existing ECC model in both encryption and decryption times, but compared to RSA, the encryption is higher while the decryption time is lower. Generally, this shows that the improved ECC out – performed the existing systems and is therefore better.

Keywords – Elliptic Curve Cryptography; Elliptic Curve Discrete Logarithm Problem; Dual Encryption/Decryption; Elliptic Curve Diffie – Hellman

1. INTRODUCTION

Security of information has become more important in this age of the Internet where sensitive and heavy amount of information get transmitted across networks. These information are sometimes very crucial, like financial data, health records, etc and need to be highly protected to prevent breaches in their security, like eavesdropping, signal tapping, interferrence, etc and preserve their integrity. Cryptography is the study of mathematical techniques related to the field of information security which include confidentiality, data integrity, entity authentication and data origin authentication. Cryptography , however, not the only means for information security, it is rather one of the set of techniques used for information security. Cryptography is also a branch of computer science that deals with the

study and analysis of encryption systems. It deals with the study and analysis of algorithms of encryption systems, their underlying principles, strength and weaknesses in relation to other cryptographic systems. Encryption is a process whereby a text, image, audio or video file is converted from its original format to another format called the cipher text that is not readily readable by humans or machines for the preservation of the security of the file so that it could be safely transmitted over computer networks. Decryption is the opposite of encryption, which is the conversion of an encrypted file into its original format. Elliptic Curve Cryptography (ECC) is a public key cryptography that uses two set of keys for its operation, a private key and a public key [1]. The private key is usually an integer while the public key is usually a point on the elliptic curve. ECC's operation is based on elliptic curves. Elliptic curves are equations that assume the general form of a Weistrass equation. The equation is of degree 3 because the highest degree of x in the equation is 3. Elliptic curves operate over both prime field, $\mathbf{F}_p$, and binary field, $\mathbf{F}_2^m$ [2].

All elliptic curve operations are usually in the modulus of the field for which it is defined. In the prime field $\mathbf{F}_p$, all operations are in modulus p, where *p* is a prime number. *2p* is the highest number of points the elliptic curve may have. Point multiplication operation is the most important operation in elliptic curves. This point multiplication operation is performed in repeated addition operation. Point doubling and point subtraction are also elliptic curve operations. The strength of ECC is in the Elliptic Curve Discrete Logarithm Problem (ECDLP) where a point J on an elliptic curve gives another point K on the elliptic curve after point multiplication operation with point multiplication constant c [3].

ECC uses two keys, being a public key cryptography, a private key and a public key. The private key is an integer while the public key is a point on the elliptic curve. The public key is a function of the private key and a base point selected from all the point generated by the elliptic curve. Point multiplication operation is the most important operation of elliptic curves which is done in repeated point addition operation. Operations defined over an elliptic curve include point addition, point subtraction and point doubling operations. Elliptic Curve Cryptography is not as secure as believed because the base equation that defines its strength and security is a linear equation with one unknown variable that is easy to evaluate. This linear equation has mathematical vulnerability because linear equations with one unknown are the most easy to solve. Thus the mathematical vulnerability can be utilized to break the cipher. This linear equation is usually expressed as the Elliptic Curve Discrete Logarithm Problem (ECDLP). The focus of this research work is to improve the security of ECC by improving the strength of the Elliptic Curve Discrete Logarithm Problem (ECDLP) making it less vulnerable to attack. This research work is significant because it details the processes required for the improvement of the security of ECC. This research work only considered improvement in the security of ECC over prime field $\mathbf{F}_p$ and was applied only to text (letters of the English alphabet and some selected special characters) to evaluate its performance.

2. MATERIAL AND METHODS

A comparative analysis between RSA and ECC was carried out to determine the better encryption algorithm between the two. The security of the RSA cryptosystem is based on the integer factorization problem (IFP) whereas the security of ECC is based on the elliptic curve discrete logarithm problem (ECDLP). The significant attraction towards ECC is that the best-known algorithm for solving the ECDLP takes full exponential time while for solving IFP of RSA takes sub-exponential time. This analysis suggests that ECC takes less memory than RSA and is better than RSA, especially on memory-constrained devices. Experimentation was carried out using data sizes 8 bits, 64 bits and 256 bits and the corresponding encryption and decryption times for both RSA and ECC were recorded as

shown in the following table. Some of the results of their research work are as shown in Table 1 [4].

Table 1: 256 bits Encryption, Decryption and Total Time (sec)

Input: 256 bits						
Security bit level	Encryption time		Decryption time		Total time	
	ECC	RSA	ECC	RSA	ECC	RSA
80	7.9240	0.5596	22.8851	19.3177	30.8091	19.8772
112	39.7008	0.5815	26.3331	102.0337	66.0339	102.6153
128	58.4386	0.5611	27.4060	209.6086	85.8446	210.1697
144	77.5034	0.5718	32.1522	311.0649	109.6556	311.6368

Source: [4].

A system based on elliptic curve cryptography was developed for the encryption of audio files. The version of the elliptic curve used was based on prime field where the value of p used was 277, which is a prime number. The elliptic curve equation used was $y^2 = x^3 + x + 1$ mod 277. That is E_{277} (1, 1). Generated all the points of the elliptic curve using the conventional elliptic curve arithmetic, and created a mapping table for encryption/decryption operation. The base point used was (0, 256). The conventional elliptic curve encryption/decryption format was used without any improvement. The audio file format used to test this method was the WAVE audio file format. Encryption was done by mapping each byte of the audio file to a point of the elliptic curve. The whole audio file was divided into two sections, the header section and the actual message. The header was not encrypted to enable playing of the unencrypted header, while the actual message was the part encrypted. The message was transmitted using standard network protocols, [5]. A research on the double encryption capacity of ECC was done and analyzed its time complexity. Multiple encryptions were discussed, the computation overhead in the process was analyzed and the feasibility of the practical application studied. The process of multiple encryption increased security and the scheme is preferable when high security requirements are needed excluding the time constraints [6].

A comparative study of symmetric cryptographic mechanism was done on AES, DES and EB64 for information security [7].A comparative analysis between RSA and ECC was done to determine the better encryption algorithm between the two. The prediction of incoming attacks is achieved in a timely manner which enables security professionals to install defense systems in order to reduce the possibility of such attacks in Zero Day attack Prediction [8].Text Encryption with Advanced Encryption Standard (AES) for Near Field Communication (NFC) Using Huffman Compression in [9] .The focus of the study is to mitigate against intrusion in the levels of communication transaction between the card and the reader.

A Secured Text Encryption with Near Field Communication (NFC) using Huffman Compression in [10] .This result shows a decline of the symbol-by-symbol restriction with elapses time which can secure the information of the unique character. This analysis suggests that ECC takes less memory than RSA and is better than RSA, especially on memory-constrained devices [11]. An intelligent spam – scammer filter mechanism was developed using bayesian techniques [12].

2.2: Description of the Existing Method

The plaintext message m to be sent is encoded as $x – y$ point P_m. It is the point P_m that will be encrypted as the cipher text and subsequently decrypted. The sender selects a

private key *a* and generates a public key P_a = a x *G*. To encrypt and send a message P_m to the receiver, the sender chooses a random positive integer *k* and produces the cipher text C_m. The sender uses the receiver's public key P_b. To decrypt the cipher text, the receiver multiplies the first point in the pair by their secret key and subtracts the result from the second point.

Algorithm of the existing model	
Plaintext	$P_m(x, y)$
Base point *G*	$G(x, y)$
Sender private key	a
Receiver private key	b
Sender public key P_a	$P_a = a \times G$
Receiver public key P_b	$P_b = b \times G$
Multiplication constant	k
Cipher text C_m	$\{C_m = P_m + kP_b, kG\}$

2.3: Description of the Improved Method

Encryption was done in this research work thus: plaintext messages P_m were mapped to elliptic curve points to produce there equivalent $x - y$ coordinates$P_m(x_m, y_m)$. The first encryption constant is selected by the sender to encrypt the message. The multiplication constant is multiplied by the receiver's public key to produce the first set of encryption key E_1. The second encryption constant is used to encrypt the message a second time. This multiplication constant is multiplied by the receiver's public key P_r to produce the second set of encryption key E_2. The first encryption is the done by adding the message point P_m to the first encryption key E_1. The second encryption is also done by adding the first encrypted message to the second encryption key E_2 which produces the cipher text *C*. Decryption is done in the reverse order.

Algorithm of the improved model	
Plaintext	$P_m(x, y)$
Base point *G*	$G(x, y)$
Sender private key	a_1
Receiver private key	b_1
Sender public key P_a	$P_a = (a_1)G$
Receiver public key P_b	$P_b = (b_1)G$
First multiplication constant	k_1
Cipher text C_m	$\{C_m = P_m + k_1P_b, k_1G\}$

Fig. 1 shows the complete block diagram of the improved ECC model. It shows how double encryption was carried out on the plaintext to generate the cipher text. It also shows the process of the flow of information from the plaintext to the cipher text. The detailed process of encryption where the first set of elliptic curve points were added to the plaintext to form another set of elliptic curve points referred to as the intermediate cipher text. These were then added to the second set of elliptic curve points to form the real cipher text. All operations were elliptic curve operations.

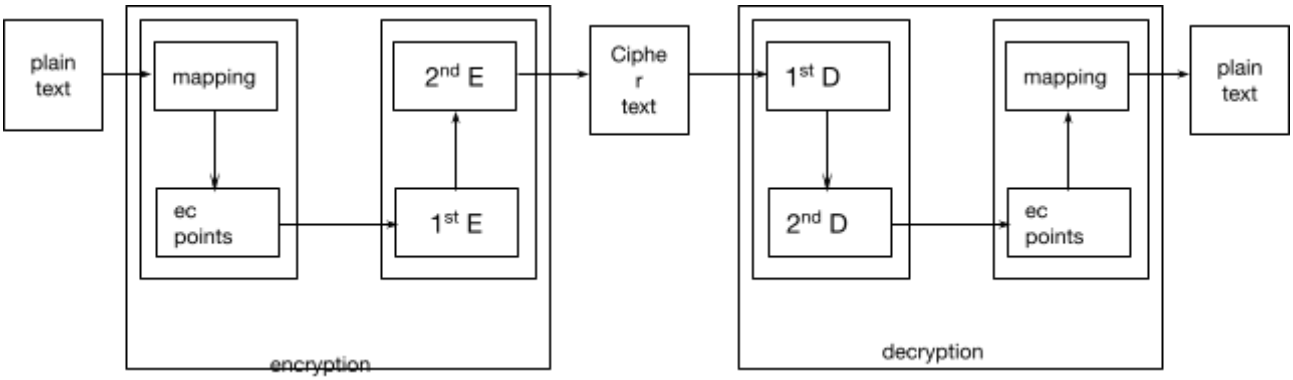

Fig. 1: Improved ECC Model with Double Encryption

The elliptic curve equation used for this research work is

$$y^2 \, mod \, 37 = (x^3 + 2x + 9) \, mod \, 37$$

Elliptic curve points were generated for x values

$$x = 0; y = \pm\sqrt{0^3 + 2(0) + 9} \, mod \, 37 = \pm\sqrt{0 + 0 + 9} \, mod \, 37 = \pm\sqrt{9} mod \, 37 = \pm 3$$

$$elliptic \, curve \, points = (0, 3) and \, (0, 34)$$

The generated points are:

(0, 3)=a (0, 34)=b (1, 7)=c (1, 30)=d (2, 13)=e (2, 24)=f (4, 9)=g (4, 28)=h (5, 12)=i (5, 25)=j (7, 12)=k (7, 25)=l (9, 4)=m (9, 33)=n (10, 17)=o (10, 20)=p (11, 17)=q (11, 20)=r (13, 7)=s (13, 30)=t (15, 11)=u (15, 26)=v (16, 17)=w (16, 20)=x (21, 5)=y (21, 32)=z (23, 7)=1 (23, 30)=2

(25, 12)=3 (25, 25)=4 (26, 5)=5 (26, 32)=6 (27, 5)=7 (27, 32)=8 (29, 6)=9 (29, 31)=0 (31, 15)=@

(31, 22)=& (33, 14)=? (33, 23)=. (35, 16)=" " (35, 21)=!

Elliptic Curve Point Addition is denoted below

$$J = J(x_j, y_j), \; K = K(x_k, y_k), \; L = L(x_l, y_l), \; J \neq K$$

$$if \; L(x_l, y_l) = J(x_j, y_j) + K(x_k, y_k)$$

$$slope \; s = \frac{y_k - y_j}{x_k - x_j} \qquad x_l = s^2 - x_j - x_k \qquad y_l = s(x_j - x_l) - y_j$$

Elliptic Curve Point Doubling is denoted below

$$J = J(x_j, y_j), \; K = K(x_k, y_k), \; L = L(x_l, y_l), \; J = K$$

$$if \; L(x_l, y_l) = 2J(x_j, y_j) = 2K(x_k, y_k)$$

$$slope \; s = \frac{3x_j^2 + a}{2y_j} \qquad x_l = s^2 - 2x_j \qquad y_l = s(x_j - x_l) - y_j$$

Key Generation and Exchange

$a < n$	Private key a
Sender public key P_a	$P_a = a \times G$
$b < n$	Private key b
Receiver public key P_b	$P_b = b \times G$
Sender Secret Key	$k = a \times P_b$
Receiver Secret Key	$k = b \times P_a$

Encryption

Intermediate cipher text = $P_m + k_1P_r, k_1G$

Cipher text = $P_m + k_1P_r + k_2P_r, k_1G, k_2G$

Where P_m = plaintext, k_1&k_2 are encryption constants, P_r = receiver public key, G = base point.

Decryption

Intermediate cipher text = $(P_m + k_1P_r + k_2P_r, k_1G, k_2G) – b_2k_2G$

Plaintext = $(P_m + k_1P_r, k_1G) – b_1k_1G$

Plaintext = P_m.

To encrypt the character "c" = (9, 4)

Receiver public key (9, 4), $k_1 = 2$ a = 2

$E_1 = k_1P_r = (9, 4)\, slope = \frac{3x_1^2+a}{2y_1} = \frac{3(9)^2+2}{2(4)} = \frac{23}{8} mod\ 37 = 26$

$x_3 = \left(26 - (9)\right) mod 37 = 29 \qquad y_3 = (26(9 - 29) - 4) mod 37 = 31$

$E_2 = k_2P_r = 2(9, 4) = (29, 31)$

1st encryption = $P_m + E_1 = (1, 7) + (29, 31)$

$x_3 = \left(22^2 - 1 - 29\right) mod 37 = 10 \qquad y_3 = (22(1 - 10) - 7) mod 37 = 17$

2nd encryption = ic + E_2 = (10, 17) + (29, 31) $\qquad s = \frac{y_2-y_1}{x_2-x_1} = \frac{31-17}{29-10} = \frac{14}{19} mod 37 = 28$

$x_3 = (28^2 - 10) mod 37 = 5 \qquad y_3 = (28(10 - 5) - 17) mod 37 = 12$

cipher text = (9, 4)

To decrypt the cipher text: (5, 12)

C = cipher text – E_2 = (5, 12) - (29, 31) = (5, 12) + (29, -31) = (5, 12) + (29, 6)

$x_3 = \left(9^2 - 4 - 29\right) mod 37 = 10 \qquad y_3 = (9(4 - 10) - 12) mod 37 = 17$

Message = C - E_1 = (10, 17) - (29, 31) = (10, 17) + (29, -31) = (9, 4) + (29, 6)

$s = \frac{y_2-y_1}{x_2-x_1} = \frac{6-10}{29-9} = \frac{-11}{20} mod 37 = \frac{26}{19} mod 37 = 15$

$x_3 = (15^2 - 10 - 19) mod 37 = 1 \qquad y_3 = (15(10 - 10) - 17) mod 37 = 7$

decrypted message = (1, 7) = c

3. RESULTS AND DISCUSSION

The system was simulated using Java programming language on Net Beans IDE 8.2. The analysis of the performance of this improved system includes key performance test in terms of key size, strength test (entropy, vulnerability of the cipher to brute force attack) and speed test (encryption and decryption speed of the cipher and comparison with those of other ciphers). The key size used for this research work is 256 bit in length, as shown in Fig. 3.0, Fig 3.1., Fig. 3.2 and Fig. 3.3. Simulation was done using the 256 bits key size on 8, 64, 100, 200, 256, 300, 400 and 500 bits of character inputs as shown below.

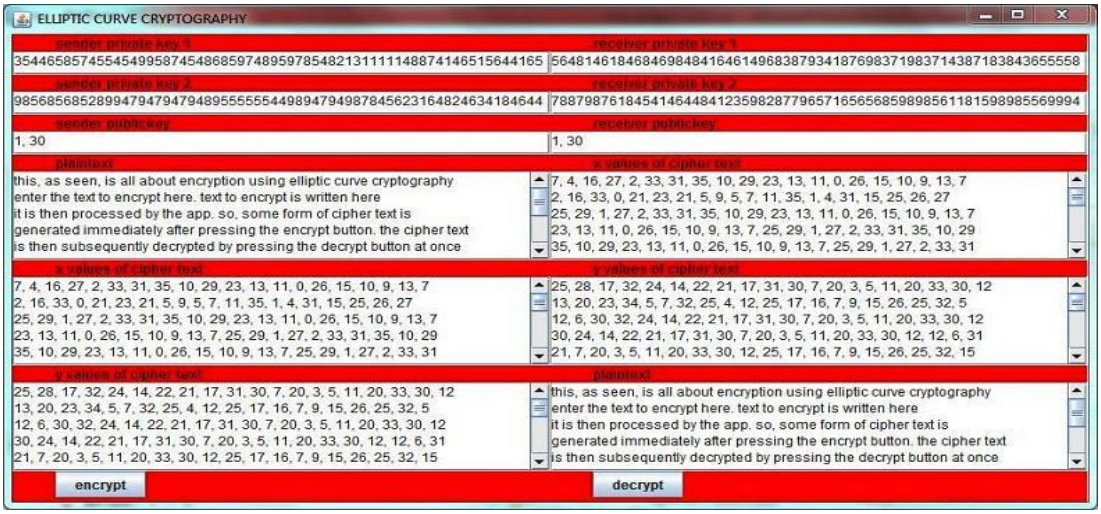

Fig. 3.0: Encryption/Decryption Interface 1.

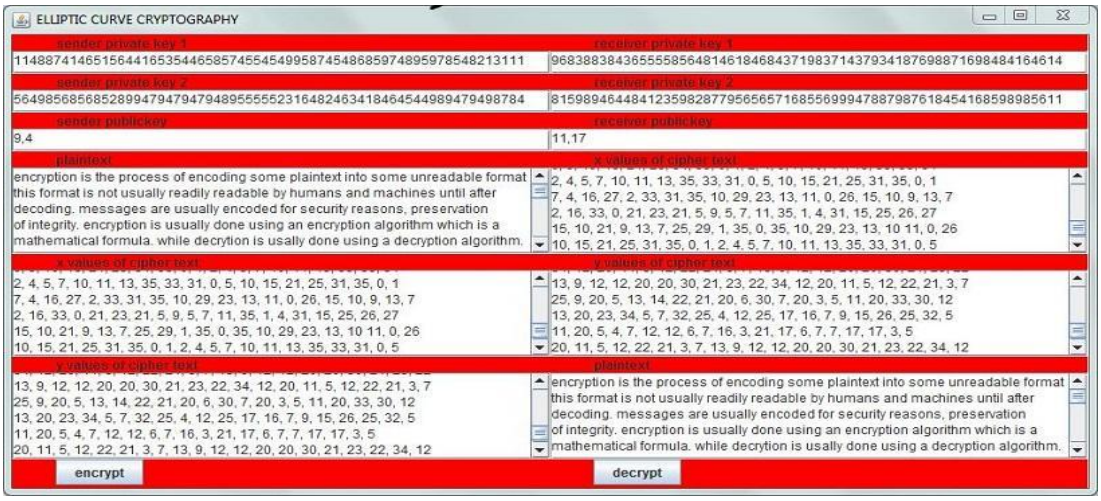

Fig. 3.1: Encryption/Decryption Interface 2.

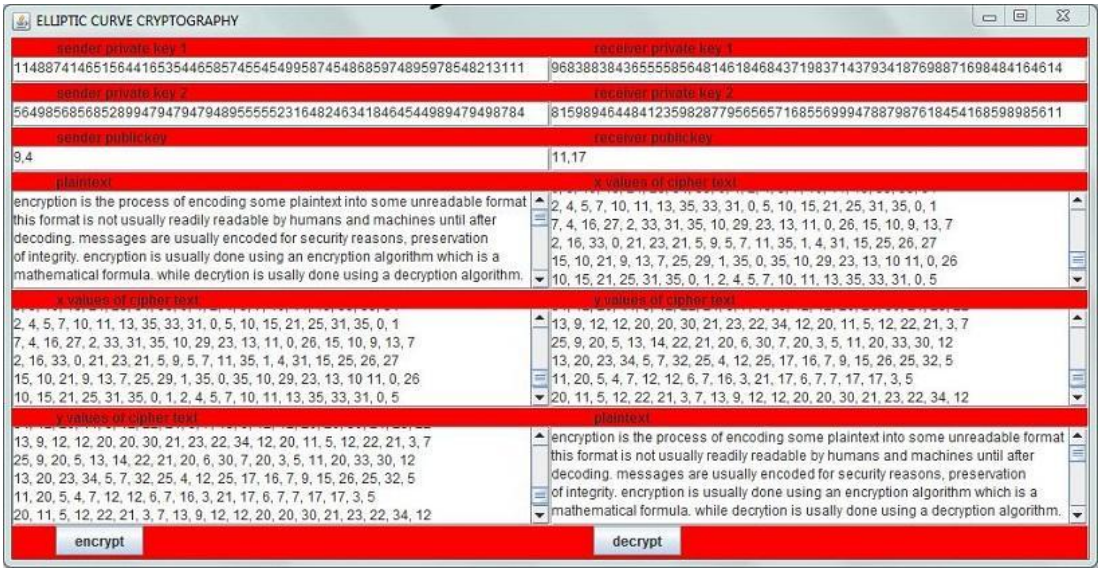

Fig. 3.2: Encryption/Decryption Interface 3

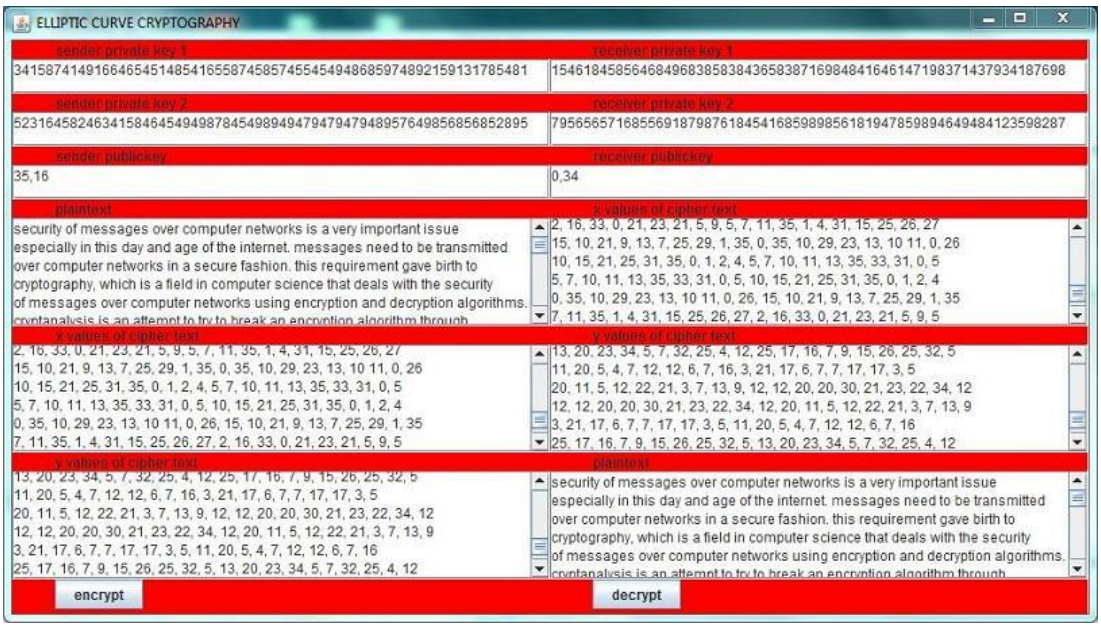

Fig. 3.3: Encryption/Decryption Interface 4

Entropy value for 100 bits characters was 155, for 300 bits characters was 106 and for 500 bits characters was 34. The high entropy values show high randomness which is a security advantage. The improved cipher had encryption and decryption time 1.65s and 0.4s respectively for 8 bits of input characters, encryption and decryption times 47.07s and 11.75s respectively for 256 bits of input characters and 90.2s and 26.99s for 500 bits of input characters. The improved cipher is better than the existing systems since its double encryption times compare favorably with the single encryption times of the existing systems.

Table 2: Entropy information after simulation with character size

Number of Characters (bits)	Entropy
100	155
200	130
300	106
400	71
500	34

$$H_b(S) = n\left(\left(\frac{1}{n}\right)(n)\right)$$
$$H_b(S) = (n)$$
$$therefore$$
$$H(K) = (10^{64}) = 213$$
$$H\left(P^n\right) = nH(P) = n(128) = 500\times7 = 3500$$
$$H\left(C^n\right) = nH(C) = n(10) = 1000\times3.3219 = 3322$$
$$total\ entropy\ H(S) = 3322 + 213 - 3500 = 34$$

Table 2 shows the encryption and decryption times for characters ranging from 8 to 500 bits in the whole process. As illustrated in the table, encryption/decryption time increased for each range of character. The increase in encryption/decryption time was as result of the increased processing overhead. Table 3 shows a comparison of the encryption and decryption time for the improved ECC model, RSA and the existing ECC.

Table 3: Comparison of Encryption Times for RSA, Existing and Improved ECC

	Existing ECC		RSA		Improved ECC	
Input characters (bits)	**Single Enc. Time (s)**	**Single Dec. time (s)**	**Single Enc. Time (s)**	**Single Dec. time (s)**	**Double Enc. Time (s)**	**Double Dec. time (s)**
8	4.73	2.00	0.05	13.65	1.64	0.40
64	20.23	8.48	0.14	77.76	13.09	3.20
256	77.50	32.15	0.57	311.07	47.07	11.75

The improvement made in the research work was based on an improved encryption/decryption process for the existing ECC. The strength of the existing ECC is in the form of a linear equation with a single unknown which was improved to a linear equation with two unknowns by the adoption of two private keys and double encryption/decryption process with the use of separate encryption constant for each round of encryption as shown in figure 4 below.

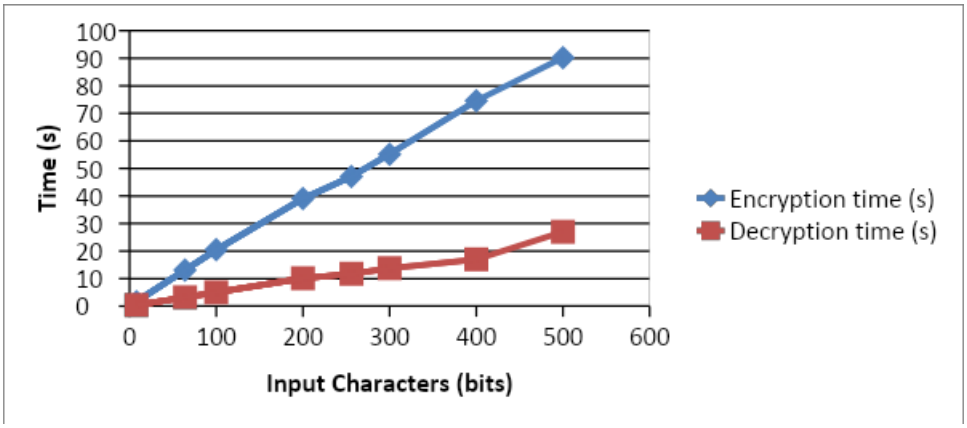

Fig. 4: Encryption/Decryption Times.

4. Conclusion

The security of encrypted data is very important. An improved encryption/decryption process for the existing ECC was presented in the study with secure of Elliptic Curve Cryptography which is embedded in the Elliptic Curve Discrete Logarithm Problem (ECDLP) The strength of the existing ECC is in the form of a linear equation with a single unknown which was improved to a linear equation with two unknowns by the adoption of two private keys and double encryption/decryption process with the use of separate encryption constant for each round of encryption. The multiplication constant was multiplied by the receiver's public key, using elliptic curve point addition operation, to produce the first set of encryption key. Second encryption constant was selected to encrypt the message a second time. This multiplication constant was multiplied by the receiver's public key P_r, using elliptic curve point addition operation, to produce the second set of encryption key E_1. The first encryption was done by adding the message point P_m to the first encryption key E_1. The second encryption was done by adding the first encrypted message to the second encryption key E_1 which produced the cipher text C.

.

Acknowledgements

The research was conducted at the Department of Computer Science, University of Ibadan, Nigeria. The authors thank the department for their support in this research work.

References

1. Abdullah, K. E. & Ali, N. H. M. (2018). Security Improvement in Elliptic Curve Cryptography. *International Journal of Advanced Computer Science and Applications,* Vol. 9, No. 5

2. Kumar, V., Kumar, R., Barbhuiya, M. A. & Saikia, M. (2016). Multiple Encryption using ECC and Its Time Complexity Analysis. *International Journal of Computer Engineering In Research Trends, 3(11):* 568 – 572.
3. Balamurugan, R., Kamalakannan, V., Rahul Ganth, D., & Tamilselvan, S. Enhancing Security in Text Messages Using Matrix based Mapping and El Gamal Method in Elliptic Curve Cryptography. *International Conference on Contemporary Computing and Informatics, IEEE*, pp. 103–106. (2014).
4. Koblitz, A. H., Koblitz, N. & Menezes, A. (2011). Elliptic Curve Cryptography: The Serpentine Course of a Paradigm Shift. *In Journal of Number Theory, Elsevier,* vol. 131, pp. 781–814
5. . Mahto, D. & Yadav, D. K. (2017). RSA and ECC: A Comparative Analysis. *International Journal of Applied Engineering Research ISSN 0973-4562* Volume 12, Number 19, pp. 9053-9061.

6. Luma, A. Selimi, B, & Ameti, L. (2015). Using Elliptic Curve Encryption and Decryption for Securing Audio Messages. https://www.researchgate.net/publication/284344817.

7. Singh, K. J. & Manimegalai, R. (2015). Evolution of Encryption Techniques and Data Security Mechanisms. *World Applied Sciences Journal 33 (10):* 1597 – 1613.

8. Adeniji O.d., Olatunji O.O 2020. Zero Day Attack Prediction with Parameter Setting Using Bi Direction Recurrent Neural Network in Cyber Security.International Journal of Computer Science and Information Security (IJCSIS), Vol. 18, No. 3,pp 111-118.

9. Adeniji Oluwashola David, Akinola Olaniyan Eliais . (2022). A Secured Text Encryption with Near Field Communication (NFC) using Huffman Compression. international Journal of Engineering and Applied Computer Science/IJEACS Volume 4,issue 2,pp14-18

10. **Adeniji, O.D., Akinola, O.E., Adesina, A.O., Afolabi, O. (2022). Text Encryption with Advanced Encryption Standard (AES) for Near Field Communication (NFC) Using Huffman Compression. In: Florez, H., Gomez, H. (eds) Applied Informatics. ICAI 2022. Communications in Computer and Information Science, vol 1643. Springer, Cham. https://doi.org/10.1007/978-3-031-19647-8_12**

11. Lundgren, B. & Moller, N. (2017). Defining Information Security. *Sci. Eng. Ethics, Springer*
12. Mahto, D. & Yadav, D. K. RSA and ECC: A Comparative Analysis. *International Journal of Applied Engineering Research ISSN 0973-4562* Volume 12, Number 19, pp. 9053-9061. (2017).