ISSN:
Print - 2277 - 0593
Online - 2315 - 7461
© FUNAAB 2016

Journal of Natural Science, Engineering and Technology

A SECURED EMBEDDED SCHEME BASED ON MULTI-CRYPTOGRAPHIC PROTOCOLS

***1S. A. ONASHOGA, 2O. OYELEKE, 1O. A. OJESANMI AND 3A. A. A AGBOOLA**

1 Department of Computer Science, Federal University of Agriculture Abeokuta, Nigeria
2Department of Physical Sciences, Mountain Top University, Ogun State, Nigeria
3 Department of Mathematics, Federal University of Agriculture Abeokuta, Nigeria
***Corresponding author:** bookyy2k@yahoo.com, **Tel:** +2348033537505

ABSTRACT

Transmission of information via unsecured channel requires confidentiality, authorization and authentication as part of major factors that must be put into consideration. This work proposes a scheme that can take care of these major factors at once. The method involves a multi-level approach that comprises of a key exchange protocol, a message authentication code, a key derivation function and a symmetric encryption known as polyalphabetic substitution that makes use of a 94x94 Vigenere table. The four cryptographic techniques are combined to form an encryption protocol, coined MLES for a message to be securely transmitted. In order to test the functionality of MLES scheme, it was implemented using Java 1.8.0 and tested on a text data. The result shows a feasible protocol that can secure a message in steganography.

Keywords: Cryptography, MLES and polyalphabetic substitution

INTRODUCTION

Security of information is now of great concern as all information are sent over an unsecure channel now-a-days. Cryptography is the design and analysis of mathematical techniques that enable secure communications in the presence of malicious adversaries. Aside protecting data from being stolen or altered, cryptography can also be used for authentication. It can further be defined as a method of storing and transmitting data in a form that only the intended party or parties can understand it.

The main objectives of any good cryptographic technique or scheme are:

i. **Confidentiality**: keeping data secret from all unauthorized parties.
ii. **Data integrity**: ensuring that the data has not been tampered with or altered by unauthorized means during transmission.
iii. **Authenticity:** Verifying that the message was truly sent by the claimed sender.

Cryptography is an efficient means of protecting sensitive information as it is transmitted over any untrusted medium which includes any network particularly the internet. (Asafe *et al*., 2014). Producing a cipher text with use of encryption algorithm and Message Authentication Code (MAC) gives the message a more secure platform and also the parties involved would be able to verify the identity of one another.

This leads to the motivation of this work by using a scheme known as Elliptic Curve Integrated Encryption Scheme (ECIES). The ECIES was proposed by Bellare and Rogaway and it is a variation of the El-Gamal public –key encryption scheme (Hankerson *et al*., 2003). The scheme proposed in this work also makes use of Elliptic Curve Diffie-Hellman (EDCH), Key Derivation Function (KDF), polyalphabetic substitution and Message Authentication Code which is HMAC.

This work focuses on design and implementation of an encryption scheme by combining more than two cryptographic techniques together to ensure integrity, authenticity and confidentiality of the message sent over an insecure channel. Combining several cryptographic schemes together in securing data gives more strength to its security.

RELATED WORK

OVERVIEW OF CRYPTOGRAPHY

Cryptography is the practice and study of techniques for secure communication in the presence of third parties otherwise known as adversaries (Yekinni *et al*., 2014). Cryptography is the science and study of secret writing. It is about the design and analysis of mathematical techniques that enable secure communications in the presence of malicious adversaries (Hankerson *et al*., 2003). Cryptography not only protects the data from theft or alteration, but also be used for user authentication. Cryptography is combination of encryption and decryption. Encryption is a process of converting the plain text to cipher text using with some keys, while decryption is process of converting the cipher text to plain text using same key or another key (Reddy *et al*., 2012). Cryptography algorithms are mainly into two ways based on the key distribution. They are Symmetric Key algorithm (Private Key algorithm), Asymmetric key algorithm (Public Key algorithm) (Reddy *et al*., 2012).

VIGENERE CIPHER

The Vigenere cipher is a method of encrypting alphabetic text by using series of different Ceaser ciphers based on the letters of a keyword. It is a simple form of polyalphabetic substitution. What is now known as the Vigenère cipher was originally described by Giovan Battista Bellaso in his 1553 book *La cifra del. Sig. Giovan Battista Bellaso.* He built upon the tabula recta of Trithemius, but added a repeating "countersign" (a key) to switch cipher alphabets every letter (wiki, 2015).

SYMMETRIC KEY CRYPTOGRAPHY

This type of cryptography uses the same key (or keys that are trivially related) for both encryption and decryption of data. Symmetric algorithm is also called secrete key algorithm private key, or single key. In secret key algorithm, both the sender and receiver will use the same key to encrypt and decrypt the data. Symmetric schemes are generally faster as compared to asymmetric keys and used to establish session keys since it involves only a key. Examples of symmetric key algorithms are data encryption standard (DES), Triple DES, Cipher Block Chaining (CBC) and Blowfish (Reddy *et al.,* 2012).

Symmetric key ciphers are implemented as either block ciphers or stream ciphers. A block cipher enciphers input in blocks of plaintext that is work on blocks of plaintext and cipher text, as opposed to individual characters, the input form used by a stream cipher.

Block Cipher: When a block cipher algorithm is used for encryption and decryption purposes, the message is divided into blocks of bits. These blocks are then put through substitution, transposition, and other mathematical functions. The algorithm dictates all the possible functions available to be used on the message, and it is the key that will determine what order these functions will take place. Strong algorithms make reengineering, or trying to figure out all the functions that took place on the message, basically impossible.

Stream Cipher: A stream cipher does not divide a message up into blocks; instead, a stream cipher treats the message as a stream of bits or bytes and performs mathematical functions on them individually When using a stream cipher, the same plaintext bit or byte will be transformed into a different cipher text bit or byte each time it is encrypted. Some stream ciphers use a key stream generator, which produces a stream of bits that is XORed with the plaintext bits to produce cipher text. Exclusive OR (XOR) is an operation that is applied to two bits. It is a function in binary mathematics. If both bits are the same, the result is zero (1 + 1= 0). If the bits are different than each other, the result is one (1 + 0= 1) (Asafe *et al*., 2014)**.**

Cipher Block Chaining (CBC) Mode: Cipher Block Chaining (CBC) does not reveal a pattern because each block of text, the key, and the value based on the previous block is processed in the algorithm and applied to the next block of text**.** This gives a more random resulting cipher text. A value is extracted and used from the previous block of text. This provides dependence between the blocks and in a sense they are chained together. This is where the title of Cipher Block Chaining (CBC) comes from, and it is this chaining effect that hides any repeated patterns (Asafe *et al*., 2014).

Data Encryption Standard (DES): is a symmetric algorithm based on block encryption algorithm (that is, message is divided into blocks of bits. These blocks are then put through substitution, transposition and other mathematical functions). DES uses a 64-bit key in which 56 bits makes up the main key and 8 bits are used for parity. When DES is applied to data, it splits the message up into blocks and performs operation on them one at a time (Asafe*et al*., 2014).

Advantages of symmetric-key cryptography

1. Symmetric-key ciphers can be designed to have high rates of data throughput.
2. Keys for symmetric-key ciphers are relatively short.
3. Symmetric-key ciphers can be employed as primitives to construct various cryptographic mechanisms including pseudorandom number generators, hash functions and computationally efficient digital signature schemes.
4. Symmetric-key ciphers can be composed to produce stronger ciphers. Simple transformations which are easy to analyze, but on their own weak, can be used to construct strong product ciphers (Menezes *et al.,* 1997).

Disadvantages of symmetric-key cryptography

1. In a two-party communication, the key must remain secret at both ends.
2. In a large network, there are many key pairs to be managed. Consequently, effective key management requires the use of an unconditionally trusted third party
3. In a two-party communication between entities A and B, sound cryptographic practice dictates that the key be changed fre-

quently and perhaps for each communication session.

4. Digital signature mechanisms arising from symmetric-key encryption typically require either large keys for the public verification function.

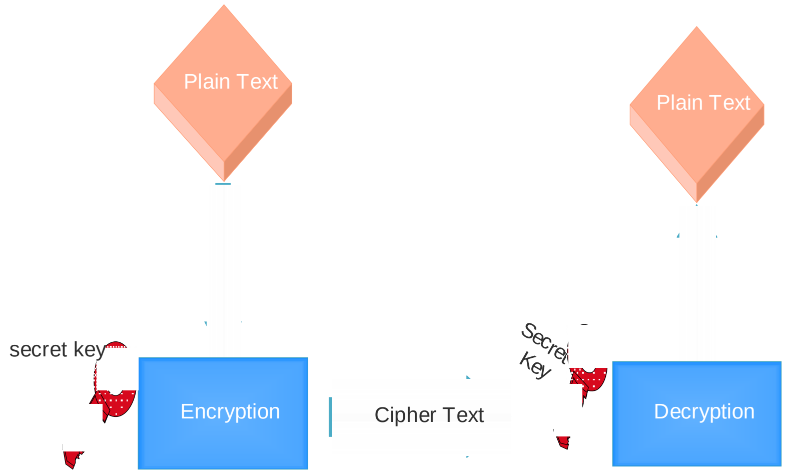


Figure 1: Secret key algorithm

ASYMMETRIC CRYPTOGRAPHY

Asymmetric key algorithms are also called public key algorithms. In public key algorithm both parties (sender and receiver) have their own different keys. In public key cryptography each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. The sender encrypts the data with his own key, and the receiver decrypts the data with his own key. First they will do the encryption or decryption with the same key, and again do the encryption or decryption with their own key. (Reddy *et al.,* 2012). Commonly used public key schemes are:

i. RSA public key encryption and signature schemes.

ii. The discrete logarithm (DL) problems whose hardness us essential for the security of the ElGamal public key encryption and signature schemes and their variants such as the Digital Signature Algorithm (DSA). (whitepaper, 2012)

iii. The elliptic curve discrete logarithm problem whose hardness is essential for the security of all elliptic curve cryptographic schemes (Hankerson *et al*. 2004).

Kester, (2012) presented a cryptosystem based on Vigenere Cipher with varying key. Vigenere cipher encryption algorithm automatically changes the cipher key after each encryption step. The successive keys will be dependent on the initial key value during the encryption process. The algorithm ultimately makes it possible for encryption and decryption of the text and also makes the Vigenère cryptosystem more difficult against frequency attack using varying keys.

Asafe *et al*., (2014) designed a crypto system for preventing unauthorized access using stand-alone software that implements cryptography using polyalphabetic substitution. The algorithm is a simple method of encrypting alphabetic text using a series of different shift ciphers based on the letters of a keyword. The system does have a high confidentiality rating in order to defend against sniffing and man-in-the-middle attacks.

MATERIALS AND METHODS

PROPOSED SCHEME

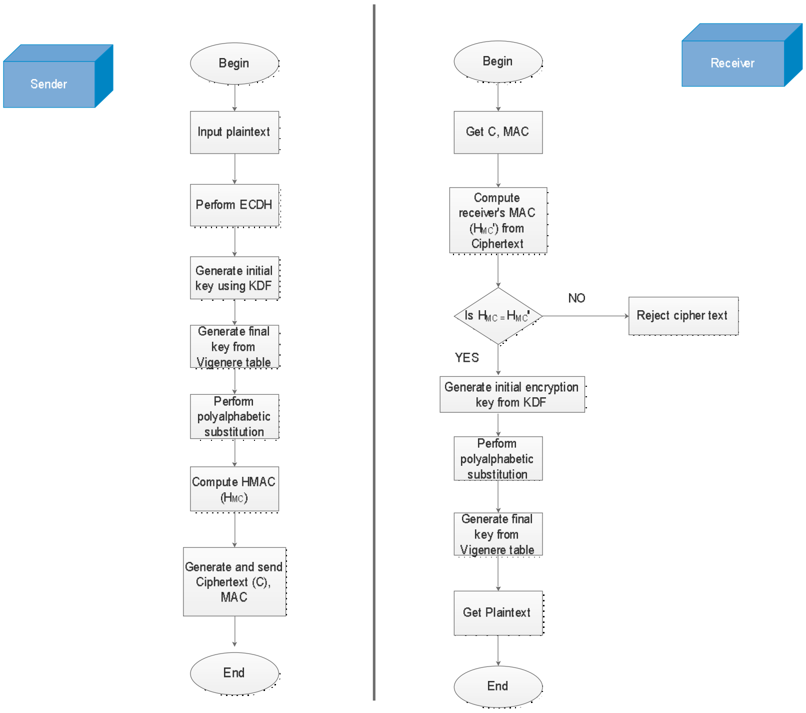

Sender
Begin
Input plaintext
Perform ECDH
Generate initial key using KDF
Generate final key from Vigenere table
Perform polyalphabetic substitution
Compute HMAC (H_{MC})
Generate and send Ciphertext (C), MAC
End
Begin
Receiver
Get C, MAC
Compute receiver's MAC (H_{MC}') from Ciphertext
Is $H_{MC} = H_{MC}'$
NO
Reject cipher text
YES
Generate initial encryption key from KDF
Perform polyalphabetic substitution
Generate final key from Vigenere table
Get Plaintext
End

Figure 2: Multi-level Encryption Scheme (MLES) flow diagram

This work makes use of a scheme known as Multi-level Encryption Scheme (MLES) which is based on the idea of the Elliptic Curve Integrated Encryption Scheme (ECIES).

Elliptic Curve Integrated Encryption Scheme

The Elliptic Curve Integrated Encryption Scheme (ECIES) was proposed by Bellare and Rogaway, and is a variant of the ElGamal public-key encryption scheme. It has been standardized in ANSI X9.63 and ISO/IEC 15946-3, and is in the IEEE P1363a draft standard. ECC is a good choice for low power environments. ECC has got applications as a public key sharing scheme and as digital signature authentication scheme. The applications of ECC are:

A. Elliptic Curve Diffie-Hellman (ECDH) Key Exchange

B. Elliptic Curve Menezes-Qu-Vanstone (ECMQV) Key Exchange and Verification
C. Elliptic Curve Digital Signature Algorithm (ECDSA). (Malik, 2010).

The best known ECC schemes and protocols are ECDH (Elliptic Curve Diffie Hellman), a key agreement protocol (Martinez and Encinas 2013).

In the proposed scheme, an Elliptic Curve Diffie-Hellman (ECDH) key exchange is used to derive two symmetric keys k_1 and k_2. Key k_1 is used to encrypt the plaintext using a symmetric-key cipher, while key k_2 is used to authenticate the resulting cipher text. Intuitively, the authentication guards against chosen cipher text attacks since the adversary cannot generate valid cipher texts on her own. The following cryptographic primitives us are employed:

Definition: Domain Parameters:
i. The *field order q*. which indicates number of elements in $\mathbf{F}_p$
ii. Two *coefficients* $a, b \in \boldsymbol{F}_p$ that define the equation of the elliptic curve *E* over $\boldsymbol{F}_p(i.e\ y^2 = x^3 + ax + b)$ in a prime field.
iii. Two field elements $xp\ and\ yp\ in\ \boldsymbol{F}_p$ that define a finite point $P = (xp, yp) \in E(\boldsymbol{F}_p)$ in affine coordinates. *P* has prime order and is called the *base point*.
iv. The *order n* of *P*.
v. The *cofactor* $h = \#E(\boldsymbol{F}_p)/n$. (It is necessary $E(\boldsymbol{F}_p)$ that be divisible by a sufficiently large prime *n*. At a minimum, one should have $n > 2^{160}$ *n*.

MULTI-LEVEL ENCRYPTION SCHEME (MLES)

The proposed scheme consists of the following sub modules

(i) Elliptic Curve Diffie-Hellman would be used for key exchange protocol in the proposed scheme. Public keys would be exchanged between the sender and the receiver prior to encryption and decryption.
(ii) A Key Derivation Function (KDF) would be used and it would be constructed from a hash function (SHA-256). SHA-256 (secure hash algorithm, FIPS 182-2) is a cryptographic hash function with digest length of 256 bits. It is a keyless hash function; that is, an MDC (Manipulation Detection Code). The KDF would be used in generating encryption key and Hash based Message Authentication Code (HMAC) key for the Message Authentication.
(iii) HMAC algorithm which would be used for the signature of the message to verify if it is truly from the claimed sender. HMAC is a hash function based message authentication code. The algorithm is given in algorithm 4 H denotes the SHA-256 hash function, C denotes the ciphertext to be transmitted and K_D is the unique $HMAC_{key}$ generated by the KDF.
(iv) A symmetric encryption technique would be combined with the above to get the cipher text. Polyalphabetic Substitution is a simple method of encrypting alphabetic text by using a series of different shift ciphers based on the letters of a keyword. For this work, a polyalphabetic substitution using one-time pad key system including a Vigenere cryptosystem would be employed. The polyalphabetic encryption and decryption procedures involved in this work are outlined in figures 3 and 4 respectively.

The proposed scheme would be implemented using Java object oriented programming language and Netbeans Integrated Development Environment (IDE) with Java 1.8.0

Algorithm 1: Polyalphabetic Encryption

```
Input: Plain text M, Encryption Key K.
Output:   Ciphertext C.
Process:  Step 1: Generate a 94 X 94 vigenere table consisting of
            lowercase alphabets, uppercase alphabets, and special
          characters.
          Step 2: Convert derived key from integer to ASCII characters.
          Step 3: Convert integer k to string S.
          Step 4: If S.length> 1
                      Sum up all elements of S to get int R
                          $R \leftarrow R \times 2$
                      Convert R to ASCII character
          Else,
                      Convert integer k to ASCII character
          Step 5: Return initial encryption key
          Step 6: While (Encryption key < plaintext)
          Step 7: Get the character at plaintext [Encryption key.length] = x
          Step 8: Get the character at plaintext [Encryption key.length-1] = y
          Step 9: Convert x and y to binary.
          Step 10: Get the complement of the binary of x and y
          Step 11: XOR the complements of x and y to get Z
          Step 12: Add Z to each initial encryption key
          Step 13: NewEncKey = Number of characters in InitialEncKey
                      equivalent to plaintext length.
          Step 14: For int i = 1 to plaintext length N
                      introwpos = row position of EncKey: in vigenere table
                      intcolpos = column position of plaintext: in vegenere table
                      ciphertext i= vigeneretable[rowpos][colpos]
                    return ciphertext.
          Step 15: End
```

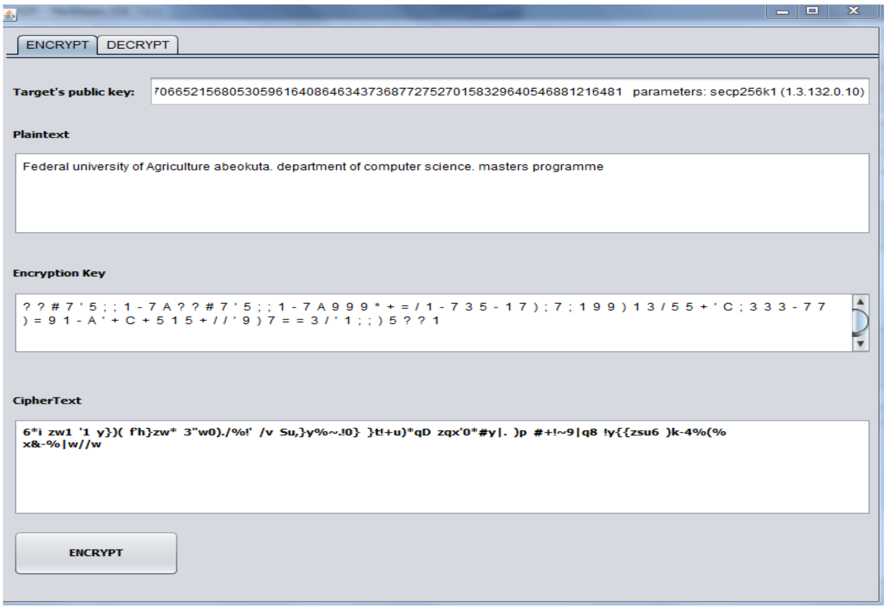
Algorithm 2: Polyalphabetic Decryption

```
Input:  Ciphertext C
Output: Plaintext
Process:  Step1:  Generate a 94 X 94 vigenere table
          Step 2: Convert derived key from integer to ASCII characters.
          Step 3: Convert integer k to string S.
          Step 4: If S.length> 1
                      Sum up all elements of S to get int R
```

```
                $R \leftarrow R \times 2$
                Convert R to ASCII character
                Else,
                Convert integer k to ASCII character
Step 5: Return initial encryption key
Step 6: While (decryption <ciphertext)
                introwpos = Get last character in decryption key from table
                intcipherpos = Get position of pos2 on rowpos in table
                plaintext2 = table [0] [cipherpos]
Step 7: XOR plaintext1 and plaintext2 to get int
Step 8: Add Z to each initial decryption key
        Return decryptionkey;
Step 9: For int i = 1 to ciphertext length N
                introwpos = get row position of decryption key [i]
                intcipherpos = get pos of ciphertext [i] in table
                plaintext [i] = table [0] [cipherpos]
                Return plaintext.
Step 10:        End
```

RESULTS

The scheme had been tested with small text data to know its feasibility. Work is in progress in making it more efficient by accepting larger size of messages or word documents. Figures 3 and 4 show the encryption and decryption outputs of the proposed Multi-level encryption scheme (MLES) respectively.



Target's public key: 7066521568053059616408646343736877275270158329640546881216481 parameters: secp256k1 (1.3.132.0.10)

Plaintext

Federal university of Agriculture abeokuta. department of computer science. masters programme

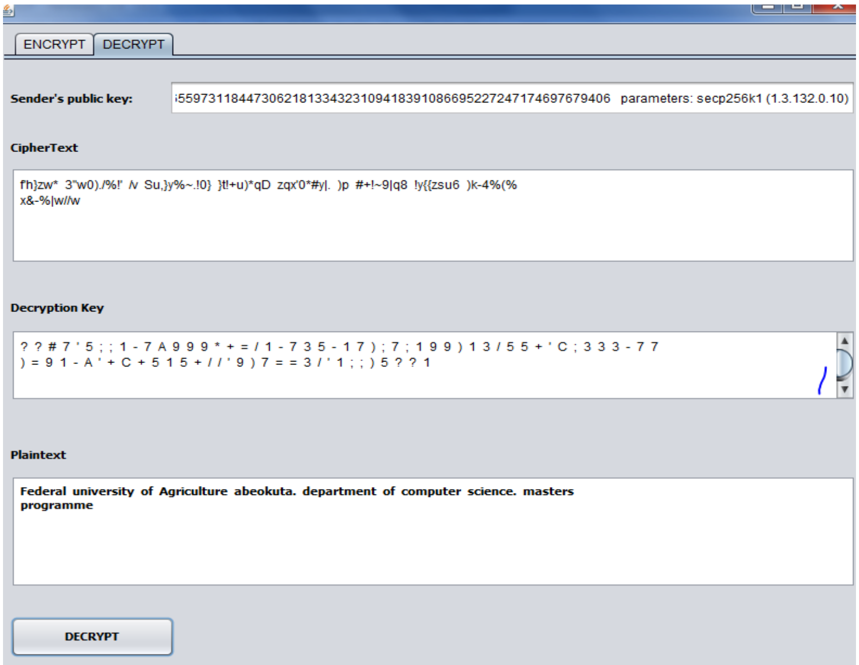
Encryption Key

```
? ? # 7 ' 5 ; ; 1 - 7 A ? ? # 7 ' 5 ; ; 1 - 7 A 9 9 9 * + = / 1 - 7 3 5 - 1 7 ) ; 7 ; 1 9 9 ) 1 3 / 5 5 + ' C ; 3 3 3 - 7 7
) = 9 1 - A ' + C + 5 1 5 + / / ' 9 ) 7 = = 3 / ' 1 ; ; ) 5 ? ? 1
```

CipherText

```
6*i zw1 '1 y})( f'h}zw* 3"w0)./%!' /v Su,}y%~.!0} }t!+u)*qD zqx'0*#y|. )p #+!~9|q8 !y{{zsu6 )k-4%(%
x&-%|w//w
```

Figure 3: MLES Encryption output



Sender's public key: ¡5597311844730621813343231094183910866952272471746976794O6 parameters: secp256k1 (1.3.132.0.10)

CipherText

```
f'h}zw*  3"w0)./%!'  /v  Su,}y%~.!0}  }t!+u)*qD  zqx'0*#y|.  )p  #+!~9|q8  !y{{zsu6  )k-4%(%
x&-%|w//w
```

Decryption Key

```
? ? # 7 ' 5 ; ; 1 - 7 A 9 9 9 * + = / 1 - 7 3 5 - 1 7 ) ; 7 ; 1 9 9 ) 1 3 / 5 5 + ' C ; 3 3 3 - 7 7
) = 9 1 - A ' + C + 5 1 5 + / / ' 9 ) 7 = = 3 / ' 1 ; ; ) 5 ? ? 1
```

Plaintext

Federal university of Agriculture abeokuta. department of computer science. masters programme

Figure 4: MLES Decryption output

CONCLUSION

The work presents combination of different cryptographic schemes for securing messages to be sent across unsecure channel. The proposed approach can be deployed in military sector and other sectors that require high security of messages. In this paper, a new encryption scheme known as Multi level Embedded Encryption Scheme had been implemented using polyalphabetic substitution that involves a 94 x 94 Vigenere cipher. The Vigenere Cipher comprises of upper case and lower case characters, special characters, symbols which is a novel approach. The combination of four cryptographic techniques makes it more secured against attacks such as man in the middle attack.

REFERENCES

Asafe, Y.N., Aigbokhan E. E., Okiki F. M. 2014. Cryptography System for Online Communication Using Polyalphabetic Substitution Method. *International Journal of Advanced Networking and Applications* 6: 2151-2157.

Hankerson, D., Menezes A.J., Vanstone S 2004. Guide to Elliptic Curve Cryptography *Springer-VL-rlag New York, Incorporated.* Pp. 1-311.

Martinez V.G., Encinas L.H. 2013. Implementing ECC with Java Standard Edition 7". *International Journal of Computer Science and Artificial Intelligence.* Vol. 3, Issue 4, pp. 134-142.

Reddy, S., Sowjanya, P., Praveena P., Shalini L. 2012. Poly-alphabetic Symmetric Key Algorithm Using Randomized Prime Numbers. *International Journal of Scientific and Research Publications* 2 (9): 1-3.

Bundesamt fur Sicherheit in der Informationstechnik. 2012. Technical Guideline Elliptic Curve Cryptography. *http://*

www.bsi.bund.de/cae/servlet/contentblob/471398/publicationFile/3061/BSI-TR-03111_pdf.pdf.

Malik M.Y. 2010. Efficient Implementation of Elliptic Curve Cryptography Using Low-power Digital Signal Processor" *12th International Conference on Advanced Communications Technology ISBN 978-89-5519-146.* 2: 1464-1468.

Kester Q. A. 2012. "A cryptosystem Based on Vigenère cipher with varying key" *International Journal of Advanced Research in Computer Engineering & Technology*. 1 (10): 108-113.

(Manuscript received: 04th July,2014; accepted: 15th June, 2016).