Encryption Schemes Based on Solution of Linear Diophantine Equation and System of Linear Diophantine Equations

P. Anuradha Kameswari¹*, Aweke Belay²

* Department of Mathematics, Andhra University, Visakhapatnam - 530003, Andhra Pradesh, India.

*Corresponding Author (E-mail: panuradhakameswari@yahoo.in).

Abstract
In this paper we propose encryption schemes, whose security relies on the hardness of finding particular solutions of linear Diophantine equation and system of linear Diophantine equations that have infinitely many solutions. In the proposed schemes, the public keys generated are the coefficients of the linear Diophantine equation and system of linear Diophantine equations and the private keys are used to deduce the particular solution of linear Diophantine equation and system of linear Diophantine equations which is the message that is encrypted.

Keywords: Encryption keys, Decryption keys, Diophantine Equations, parametric solutions

AMS Subject Classification: 94A60, 11T71

1 INTRODUCTION
Cryptography is the study of maintaining the security of data communications between legal parties from outsiders, using cryptosystems. In cryptosystem there are two types of data transformation. The text to be sent is called plain text and the text to be received is called cipher text. The process of changing a plain text to a cipher text is called encryption and the process of changing a cipher text to a plain text is called decryption [6],[8].

The encryption and decryption depends on keys that are either private or public. In a public key system, parties use the encryption algorithm \( E(P_e, M) \) and the decryption algorithm \( D(P_d, C) \), where \( P_e \) is the public key, \( P_d \) is the private key, \( M \) is a plain text to be encrypted, and \( C \) is the cipher text to be decrypted. The parties publish their own encryption key by keeping the decryption key as secret. If \( A \) and \( B \) are parties that wish to communicate and suppose that \( B \) wants to send a message to \( A \), \( B \) finds the public encryption key from the public directory, and then encrypts the message \( M \) as \( C = E(P_e, M) \). On receiving \( C \), \( A \) decrypts it as \( M = D(P_d, C) \) [10]. In the public key cryptosystem decryption is based on the difficulty of computation of private key \( P_d \). There are public key cryptosystems with algorithms for decryption based on hard computational problems, like integer factorization, discrete logarithm based on modular integers, elliptic curves, Lucas sequences, short vector problem in lattices in higher dimensions. There are also several encryptions based on Diophantine equations. In the protocols based on Diophantine equations the evaluation of solutions plays a vital role. In this paper we propose an encryption scheme, whose security relies on the hardness of finding a particular solution of system of linear Diophantine equations that have infinitely many solutions.

II PRELIMINARIES
A linear Diophantine equation is of the form \( a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \) for \( a_1,b \in \mathbb{Z} \). A general, system of linear Diophantine equations is of the form

\[
\begin{align*}
  &a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\
  &a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\
  &\vdots
  \\
  &a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots + a_{mn}x_n = b_m,
\end{align*}
\]

where, \( b_1, \ldots, b_m \), and \( a_{ij} \) for all \( 1 \leq i \leq m, 1 \leq j \leq n \) are given integers [11].

In general, an \( n \) variable linear Diophantine equation, may have no solution, may have unique solution or infinitely many solutions. A common solution of all linear Diophantine equation in the system is a solution of system of linear Diophantine equations. There are several studies on system of linear Diophantine equations with infinitely many solutions as
given in [15]. In this section we recall the theorem as in [7] on existence of infinitely many solutions for linear Diophantine equations, and system of linear Diophantine equations. The following theorem gives the condition for the existence of solutions of the linear Diophantine equation with \( n \) variables for \( n \geq 2 \) basing on the \( \text{gcd} \) of the coefficients.

**Theorem 2.1** The Linear Diophantine equation \( a_1x_1 + a_2x_2 + \ldots + a_nx_n = b \) has infinitely many integer solutions if and only if \( \text{gcd}(a_1, a_2, \ldots, a_n) \) divides \( b \).

**Proof.** follows from [7], [14].

A solution of a system of linear Diophantine equations can be tackled initially using similar techniques to those found in linear equations over the real numbers in linear algebra, by elimination, and substitution methods, or Gaussian elimination, and Gauss-Jordan methods, and Cramer’s Rule using matrices. There are studies on different methods of finding the general parametric solutions of system of Linear Diophantine equations such as Hermite Normal Form, Smith Normal Form, and using integer row reduction in [2], [4], [9]. The following theorem gives the parametric solutions for system of linear Diophantine equation by row modular reductions.

**Theorem 2.2** The system \( AX = B \) has integer solutions if and only if the system \( R^tK = B \) has integer solutions for \( K \), where \( R \) is a row echelon form obtained from uni-modular row reductions of \( [A^tI] \) to \( [R^tT] \) and all the solutions of \( AX = B \) are of the form \( X = T^tK \).

**Proof.** Follows from [4]

Parametric solutions of system of linear Diophantine equation are generated by implementing row modular reductions in above theorem by a crushing method given by the theorem in the following [15].

**Theorem 2.3** Let \( a_1, a_2, \ldots, a_n \) be \( n \) integers and if \( d = \text{gcd}(a_1, a_2, \ldots, a_n) \) then the column matrix \( \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \) be reduced to \( \begin{pmatrix} d \\ 0 \\ \vdots \\ 0 \end{pmatrix} \) by uni modular reductions using gcd by crushing.

**Proof.** follows from [15]

The parametric representation of solutions emphasises the difficulty of computing a particular solution from an infinite class of solutions. In sections 3 and 4 we propose encryption schemes with the message as particular solution of a Linear Diophantine Equation and a system of linear Diophantine equations respectively.

### III ENCRYPTION SCHEME BASED ON SOLUTION OF LINEAR DIOPHANTINE EQUATION

This encryption demonstrates the potentiality of using linear Diophantine equation. In this scheme, two parties are involved in the transformation of messages. First the receiver generates private and public keys, then the sender can send the message using the public key of receiver and the receiver can decrypt the message using his private key. The schemes is as follows;

**Public Key and Private Key generation**

**Step 1** : Private key generation by receiver \( A \)
- \( A \) chooses large numbers \( q_1, q_2, \ldots, q_r \) such that \( \text{gcd}(q_i, q_j) = 1 \) for all \( i \neq j \).
- \( A \) keeps \( (q_1, q_2, \ldots, q_r) \) as the private key.

**Step 2** : Public key generation by receiver \( A \)
- \( A \) defines \( Q_i = \prod_{j=1}^{n} q_j \) for all \( j \neq i \) and finds the reciprocal \( Q_i^{-1} \) of \( Q_i \).
- \( A \) defines \( s_i = Q_iQ_i^{-1} \) for all \( i = 1, 2, \ldots, r \).
- \( A \) publishes \( S = (s_1, s_2, \ldots, s_r) \) as the public key.

**Encryption by Sender \( B \)**

**Step 1** : If the message to be communicated is a number \( M \), then
- \( B \) chooses his message \( M \) and fragments the message as \( M = (m_1, m_2, \ldots, m_r) \), for \( m_i \) small numbers, so that number \( M \) is a concatenation of \( m_i \)'s, i.e. \( M = m_1m_2 \ldots m_r \).

**Step 2** :
- \( B \) uses the public key \( S = (s_1, s_2, \ldots, s_r) \) of \( A \) and encrypts his message as
  \[ C = E(S, M) = S \ast M = \sum_{i=1}^{r} s_im_i = s_1m_1 + s_2m_2 + \ldots + s_rm_r \]
- \( B \) sends cipher text \( C \) to \( A \).
Decryption by Receiver A

Step 1:

A decrypts the original message using his private key \((q_1, q_2, ..., q_r)\) as

\[
(C \mod q_1, C \mod q_2, ..., C \mod q_r) = (m_1, m_2, ..., m_r)
\]

### Proposition 3.1

If \(m_i < q_i\), then \(C \mod q_i = m_i\) for all \(i = 1, 2, ..., r\)

**Proof.** First note for all \(i = 1, 2, ..., r\) as \(gcd(Q_i, q_i) = 1\), there exists \(Q_i^{-1}\), the reciprocal of \(Q_i\) such that \(Q_i Q_i^{-1} \equiv 1 \mod q_i\), and also as \(q_j\) divides \(Q_i\), note \(Q_i Q_i^{-1} \equiv 0 \mod q_j\) for all \(i \neq j\) [1], [5]. Therefore we have for all \(i = 1, 2, ..., r\) \(C \mod q_i = (s_1 m_1 + s_2 m_2 + \ldots + s_r m_r) \mod q_i\) which results in:

\[
\begin{align*}
(C \mod q_1) &= (Q_1 Q_1^{-1} m_1 + Q_2 Q_1^{-1} m_2 + \ldots + Q_r Q_1^{-1} m_r) \mod q_1 \\
(C \mod q_2) &= (Q_2 Q_2^{-1} m_1 + Q_2 Q_2^{-1} m_2 + \ldots + Q_r Q_2^{-1} m_r) \mod q_2 \\
&= \ldots \\
(C \mod q_r) &= (Q_r Q_r^{-1} m_1 + Q_r Q_r^{-1} m_2 + \ldots + Q_r Q_r^{-1} m_r) \mod q_r
\end{align*}
\]

\[
= 0 m_1 + 0 m_2 + \ldots + 1 m_i + 0 m_{i+1} + \ldots + 0 m_r = m_i
\]

for \(m_i < q_i\).

### Remark 3.2

If the message is in number in decimal system then for the message \(M\) fragmented as an \(r\) tuple with \(m_i\)'s as ordinates, the \(m_i\)'s are restricted to single digit numbers, so that for \(q_i\)'s as numbers with more than one digit, we always have \(m_i < q_i\).

### 3.1 Encryption of the message \(M\) with respect to the public key \(S\)

If the fragmentation of message \(M\) is as a \(t\) tuple \((m_1, m_2, ..., m_t)\) and if the public key \(S\) is an \(r\) tuple \((s_1, s_2, ..., s_r)\) note that \(t\) need not equal to \(r\). In the following we discuss the encryption with respect to message tuple length and public key tuple length. If \(M = (m_1, m_2, ..., m_t)\) is the message by the sender and \(S = (s_1, s_2, ..., s_r)\) is the public key by the receiver, then we have the three cases \(r = t, r < t\) and \(r > t\). Now we describe encryption and decryption in each of the three cases in the following sections.

#### 3.1.1 Encryption and decryption in the Case \(t = r\)

In this case the message \(M = (m_1, m_2, ..., m_t) = (m_1, m_2, ..., m_r)\) then for \(S = (s_1, s_2, ..., s_r)\) the encrypted message will be

\[
C = m_1 s_1 + m_2 s_2 + \ldots + m_r s_r
\]

After receiving the encrypted message, the receiver decrypts it using his private keys \((q_1, q_2, ..., q_r)\) as

\[
(C \mod q_1, C \mod q_2, ..., C \mod q_r) = (m_1, m_2, ..., m_r)
\]

#### Example 3.3

A and B communicate the message \(M = 331\).

### Public Key and Private Key generation by receiver A

#### Step 1: Private key generation by receiver A

- **A** chooses large numbers \((q_1, q_2, q_3) = (104, 147, 121)\) such that \(gcd(q_i, q_j) = 1\) for all \(i \neq j\).
- **A** keeps \((104, 147, 121)\) as the private key.

#### Step 2: Public key generation by receiver A

- **A** defines \(Q_i = \prod_{j=1}^{i-1} q_j = \prod_{j=1}^{i} q_j\) for all \(j \neq i\) and finds the reciprocal \(Q_i^{-1}\) of \(Q_i\) for \(i = 1, 2, 3\) for \(Q_i = q_2q_3 = 147 \times 121 = 17787\), \(Q_i^{-1} \equiv 35 \mod 104\) for \(Q_2 = q_1q_3 = 104 \times 121 = 12584\), \(Q_2^{-1} \equiv 38 \mod 147\) for \(Q_3 = q_1q_2 = 104 \times 147 = 15288\), \(Q_3^{-1} \equiv 49 \mod 147\).
- **A** defines \(s_i = Q_i Q_i^{-1}\) for all \(i\). Then for the above choices of \(q_i\) we have

\[
\begin{align*}
s_1 &= Q_1 Q_1^{-1} = 17787 \times 35 = 622545 \\
s_2 &= Q_2 Q_2^{-1} = 12584 \times 38 = 478192 \\
s_3 &= Q_3 Q_3^{-1} = 15288 \times 49 = 749112
\end{align*}
\]

- **A** publishes \(S = (s_1, s_2, s_3) = (622545, 478192, 749112)\) as the public key.

### Encryption by Sender B

#### Step 1:

- **B** chooses the message \(M = 331\) and fragments as \(M = (m_1, m_2, m_3) = (3, 3, 1)\).

#### Step 2:

- **B** considers the public key \(S = (622545, 478192, 749112)\) of \(A\) and encrypts the message as

\[
C = E(S, M) = S \ast M = \sum_{i=1}^{3} s_i m_i = s_1 m_1 + s_2 m_2 + s_3 m_3
\]
That is, \( C = 622545 \times 3 + 478192 \times 3 + 749112 \times 1 = 4051323 \) is the encrypted message.

- B sends cipher text \( C = 4051323 \) to the receiver A.

**Decryption by receiver A**

**Step 1:**
- A decrypts the original message using the private key \((q_1, q_2, q_3) = (104, 147, 121)\) as follows

\[
\begin{align*}
    m_1 & \equiv C \mod q_1 = 4051323 \mod 104 = 3 \\
    m_2 & \equiv C \mod q_2 = 4051323 \mod 147 = 3 \\
    m_3 & \equiv C \mod q_3 = 4051323 \mod 121 = 1
\end{align*}
\]

A concatenates the message \( M = (m_1, m_2, m_3) = (3, 3, 1) \) and obtains the original message as \( M = m_1 m_2 m_3 = 331 \).

**3.1.2 Encryption and decryption in the Case t < r**

In this case the sender fragments the message \( M = m_1 m_2 \ldots m_t \) as an \( r \)-tuple by taking 0 in the first \( r - t \) entries as 0 on the left has no place value, i.e., the message is fragmented as \( M = (p_1, p_2, \ldots , p_{r-t}, p_{r-t+1}, \ldots , p_r) \) as

\[
(0, 0, \ldots , 0, m_1, m_2, \ldots , m_t),
\]

then encrypts the message as

\[
C = 0 \times s_1 + 0 \times s_2 + \ldots + 0 \times s_{r-t} + m_1 \times s_{r-t+1} + \ldots + m_t \times s_r,
\]

After receiving the encrypted message, the receiver decrypts it using his private keys \((q_1, q_2, \ldots , q_r)\) as \( M = 0 \times m_1 m_2 \ldots m_t \) as

\[
(C \mod q_1, C \mod q_2, \ldots , C \mod q_r) = (0, 0, m_1, m_2, \ldots , m_t)
\]

is 0 \( \ldots 0, m_1, m_2 \ldots m_t \).

**3.1.3 Encryption and decryption in the Case t > r**

By division algorithm for \( t \) and \( r \) there exists integers \( 0 > a > 0 \) and \( 0 \leq b < r \) such that \( t = ar + b \). Now using \( a \) and \( b \) the sender considers the message \( M \) as \( M = (0, 0, m_1, m_2, \ldots , m_b, m_{b+1}, \ldots , m_{b+r}, \ldots , m_{b+ar}) \) and encrypts the message \( M \) as \( (C_0, C_1, \ldots , C_{a-1}, C_a) \) where:

\[
\begin{align*}
    C_0 & = 0s_1 + 0s_2 + \ldots + 0s_{r-b} + m_1 s_{r-b+1} + \ldots + m_b s_r \\
    C_1 & = m_{(b+1)} s_1 + \ldots + m_{b+r} s_r \\
    C_2 & = m_{(b+r+1)} s_1 + \ldots + m_{b+2r} s_r \\
    \vdots \\
    C_{a-1} & = m_{(b+(a-1)+1)} s_1 + \ldots + m_{b+(a-1)r} s_r \\
    C_a & = m_{(b+(a-1)+1)} s_1 + \ldots + m_{b+ar} s_r
\end{align*}
\]

Now, the sender sends the encrypted message \( M \) as \((C_0, C_1, \ldots , C_{a-1}, C_a)\) to the receiver. After receiving the encrypted message, the receiver decrypts it using his private key \((q_1, q_2, \ldots , q_r)\) as

\[
(C \mod q_1, C \mod q_2, C \mod q_3, C \mod q_4, C \mod q_5, \ldots , C \mod q_r)
\]

as \( (0, 0, m_1, m_2, \ldots , m_t) \) and obtains the orginal message as \( M = m_1 m_2 \ldots m_t \).

**Example 3.4** A and B communicate the message \( M = 32123113 \)

**Public Key and private Key generation**

**Step 1:** Private key generation by receiver A
- A chooses large numbers \((q_1, q_2, q_3, q_4, q_5) = (41, 25, 63, 52, 121)\)
- A keeps \((41, 25, 63, 52, 121)\) as the private key.

**Step 2:** Public key generation by receiver A
- A defines \( Q_i = \prod_{j \neq i}^{5} q_j \) for all \( j \neq i \) and finds the reciprocal \( Q_i^{-1} \) of \( Q_i \) for \( i = 1, 2, 3, 4, 5 \) as \( Q_1 = q_2 q_3 q_4 q_5 = 25 \times 63 \times 52 \times 121 = 9909900, Q_1^{-1} \equiv 8 \mod 41 \)
- \( Q_2 = q_1 q_3 q_4 q_5 = 41 \times 63 \times 52 \times 121 = 16252236, Q_2^{-1} \equiv 16 \mod 25 \)
- \( Q_3 = q_1 q_2 q_4 q_5 = 41 \times 25 \times 52 \times 121 = 6449300, Q_3^{-1} \equiv 44 \mod 63 \)
- \( Q_4 = q_1 q_2 q_3 q_5 = 41 \times 25 \times 63 \times 121 = 7813575, Q_4^{-1} \equiv 35 \mod 52 \)
- \( Q_5 = q_1 q_2 q_3 q_4 = 41 \times 25 \times 63 \times 52 = 3357900, Q_5^{-1} \equiv 96 \mod 121 \)
A defines \( s_i = Q_iQ_i^{-1} \) for all \( i \), then for the above choices of \( q_i \) we have
\[
\begin{align*}
\ s_1 &= Q_1Q_1^{-1} = 9909900 \times 8 = 79279200 \\
\ s_2 &= Q_2Q_2^{-1} = 16252236 \times 16 = 260035776 \\
\ s_3 &= Q_3Q_3^{-1} = 6449300 \times 44 = 283769200 \\
\ s_4 &= Q_4Q_4^{-1} = 7813575 \times 35 = 273475125 \\
\ s_5 &= Q_5Q_5^{-1} = 3357900 \times 96 = 322358400.
\end{align*}
\]

A publishes \((s_1, s_2, s_3, s_4, s_5) = (79279200, 260035776, 283769200, 273475125, 322358400)\) as the public key.

### Encryption by Sender B

#### Step 1:
- B chooses his message
  \[ M = (m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8) = (3, 2, 1, 2, 3, 1, 1, 3). \]

#### Step 2:
- The public key of \( A \) is 
  \((79279200, 260035776, 283769200, 273475125, 322358400)\) with \( t= 8 \) and \( r=5 \).
  
  For \( 8 = 5.1 + 3 \),
  
  - B considers the message as \((3, 2, 1, 2, 3, 1, 1, 3)\) and encrypts the message as \((C_1, C_2)\) for
  \[
  \begin{align*}
  C_1 &= s_1m_1 + s_2m_2 + s_3m_3 + s_4m_4 + s_5m_5 \\
  &= 79279200 \times 0 + 260035776 \times 0 + 283769200 \times 3 \\
  &+ 273475125 \times 2 + 322358400 \times 1 \\
  &= 1720616250.
  \end{align*}
  \]

  \[
  \begin{align*}
  C_2 &= s_1m_6 + s_2m_7 + s_3m_8 + s_4m_9 + s_5m_{10} \times 79279200 \times \\
  &+ 260035776 \times 3 + 283769200 \times 1 + 273475125 \times \\
  &+ 322358400 \times 3 \\
  &= 2462985253.
  \end{align*}
  \]

  - B sends the encrypted message \((C_1, C_2) = (1720616250, 2462985253)\)
    
    to the receiver \( A \).

### Decryption by receiver A

#### Step 1:
- A decrypts the message using his private key
  \[(q_1, q_2, q_3, q_4, q_5) = (41, 25, 63, 52, 121)\]
  as
  \[
  (C_1 \mod q_1, C_1 \mod q_2, \ldots, C_1 \mod q_5, C_2 \mod q_1, \\
  C_2 \mod q_2, \ldots, C_2 \mod q_5)
  \]
  
  \[= (1720616250 \mod 41, 1720616250 \mod 25, 1720616250 \mod 63, \\
  1720616250 \mod 52, 1720616250 \mod 121, 2462985253 \mod 41, \\
  2462985253 \mod 25, 2462985253 \mod 63, \\
  2462985253 \mod 52, 2462985253 \mod 121)
  \]
  
  \[= (0, 0, 3, 2, 1, 2, 3, 1, 1, 3)\]

- A obtains the original message as \( M = 0032123113 = 32123113 \).

#### Note 3.5.
\((0, 0, 3, 2, 1)\) and \((2, 3, 1, 1, 3)\) as sub messages are particular solutions of the linear Diophantine equations

\[79279200x_1 + 260035776x_2 + 283769200x_3 \\
+ 273475125x_4 + 322358400x_5 = 1720616250 \]

\[79279200x_6 + 260035776x_7 + 283769200x_8 \\
+ 273475125x_9 + 322358400x_{10} = 2462985253 \]

### IV ENCRYPTION SCHEME BASED ON SOLUTION OF SYSTEM OF LINEAR DIOPHANTINE EQUATION

In this section we construct an encryption scheme based on difficulty of evaluation of a particular solution of System of linear Diophantine equation with infinitely many solutions. This encryption is used when a sender wishes to send some messages to more than one receiver. All the receivers first generate their respective public keys and make them public, then using each of the public keys sender encrypts the message and sends the respective encrypted message to the corresponding receiver.

Public key and private key generation by receivers \(A_1, A_2, \ldots, A_m\)
Step 1: Private key generation by receivers $A_1, A_2, \ldots, A_m$.

- $A_i$ choose large numbers $\{q_{i1}, q_{i2}, \ldots, q_{ir_i}\}$ such that $(q_{il}, q_{is}) = 1$ for all $l \neq s$
- $A_i$ keeps $\{q_{i1}, q_{i2}, \ldots, q_{ir_i}\}$ as the private key.

Step 2 : Public key generation by receivers $A_i$

- $A_i$ defines $Q_{il} = \prod_{j=1}^{l} q_{ij}$ for all $j \neq l$ and finds the reciprocal $Q_{il}^{-1}$ of $Q_{il}$.
- $A_i$ defines $s_{il} = Q_{il}Q_{il}^{-1}$ for all $l = 1, 2, \ldots, r_i$ and $A_i$ publish $S_i = (s_{i1}, s_{i2}, \ldots, s_{ir_i})$ as the public key.

Encryption by sender $B$

step 1 :
- $B$ considers the public key $S_i$ of $A_i$ takes $r = \max\{r_i\}_{i=1}^{n}$ and considers $S_i$ as an $r$-tuple given as $(k_{i1}, k_{i2}, \ldots, k_{ir}) = (0, 0, \ldots, 0, s_{i1}, \ldots, s_{ir})$ and encrypts the message $C_{A_i}$ as

$$C_{A_i} = \sum_{l=1}^{r} k_{il}m_l = k_{i1}m_1 + k_{i2}m_2 + \ldots + k_{ir}m_r.$$  

Step 2 :
- $B$ chooses the message $M = (m_1, m_2, \ldots, m_r)$ for small $m_i$.
- $B$ sends the cipher text integer $C_{A_i}$ to $A_i$.

Decryption by receivers $A_i$

Step 1 :
- $A_i$ decrypts the original message $M = m_1m_2 \ldots m_r$ using the private key $\{q_{i1}, q_{i2}, \ldots, q_{ir_i}\}$

Note 4.1. The encryption and decryption with respect to message component and public key component can be considered as in the case of linear Diophantine equation in section 3.1. according to the cases $t = r$, $t < r$, $t > r$.

Example 4.2 $B$ communicates the message $M = 321$ to two receivers $A_1$ and $A_2$.

Public key and private key generation by receivers $A_1$ and $A_2$

Step 1: Private key generation by the receivers $A_1$ and $A_2$

- $A_1$ chooses numbers $(q_{11}, q_{12}, q_{13}) = (10, 21, 11)$
- $A_1$ keeps $(10, 21, 11)$ as the private keys.

And,

- $A_2$ chooses numbers $(q_{21}, q_{22}, q_{23}, q_{24}) = (13, 14, 15, 121)$
- $A_2$ keeps $(13, 14, 15, 121)$ as the private keys.

Step 2 : Public key generation by receivers $A_1$ and $A_2$

- $A_1$ defines $Q_{i1}$ as $Q_{i1} = \prod_{j=1}^{l} q_{ij}$ for all $j \neq l$ and finds the reciprocal $Q_{i1}^{-1}$ of $Q_{i1}$ for $l = 1, 2, 3$ as

$$Q_{11} = q_{12}q_{13} = 21 \times 11 = 231, \ Q_{11}^{-1} \equiv 1 \mod 10$$

$$Q_{12} = q_{11}q_{13} = 10 \times 11 = 110, \ Q_{12}^{-1} \equiv 17 \mod 21$$

$$Q_{13} = q_{11}q_{12} = 10 \times 21 = 210, \ Q_{13}^{-1} \equiv 1 \mod 11$$

- $A_2$ defines $Q_{2l}$ as $Q_{2l} = \prod_{j=1}^{l} q_{2j}$ for all $j \neq l$ and finds the reciprocal $Q_{2l}^{-1} \mod q_{2l}$ for $l = 1, 2, 3, 4$ as

$$Q_{21} = q_{22}q_{23}q_{24} = 14 \times 15 \times 121 = 25410, \ Q_{21}^{-1} \equiv 5 \mod 13$$

$$Q_{22} = q_{21}q_{23}q_{24} = 13 \times 15 \times 121 = 23595, \ Q_{22}^{-1} \equiv 3 \mod 121$$

$$Q_{23} = q_{21}q_{22}q_{24} = 13 \times 14 \times 121 = 22022, \ Q_{23}^{-1} \equiv 8 \mod 15$$

$$Q_{24} = q_{21}q_{22}q_{23} = 13 \times 14 \times 15 = 2730, \ Q_{24}^{-1} \equiv 105 \mod 121$$

- $A_1$ defines $s_{i1} = Q_{i1}Q_{i1}^{-1}$ for $l = 1, 2, 3$ as

$$s_{11} = Q_{11}Q_{11}^{-1} = 231 \times 1 = 231$$

$$s_{12} = Q_{12}Q_{12}^{-1} = 110 \times 17 = 1870$$

$$s_{13} = Q_{13}Q_{13}^{-1} = 210 \times 1 = 210$$

- $A_2$ defines $s_{i2} = Q_{i2}Q_{i2}^{-1}$ for $l = 1, 2, 3, 4$ as

$$s_{21} = Q_{21}Q_{21}^{-1} = 25410 \times 5 = 127050$$

$$s_{22} = Q_{22}Q_{22}^{-1} = 23595 \times 3 = 70785$$

$$s_{23} = Q_{23}Q_{23}^{-1} = 22022 \times 8 = 176176$$

79
\( s_{24} = Q_{24}Q_{24}^{-1} = 2730 \times 105 = 286650. \)

- \( A_1 \) publishes \( s_1 = (s_{11}, s_{12}, s_{13}) = (231, 1870, 210) \) as public key.
- \( A_2 \) publishes \( s_2 = (s_{21}, s_{22}, s_{23}, s_{24}) = (127050, 70785, 176176, 286650) \) as public key.

**Encryption by Sender \( B \)**

**Step 1 :**
- \( B \) chooses the message \( M = (m_1, m_2, m_3) = (3, 2, 1) \).

**Step 2 :**
- \( B \) takes \( r = \max\{r_i\}_{i=1}^2 = \max\{3, 4\} = 4 \) and considers the message \( M = P = (p_1, p_2, p_3, p_4) = (0, m_1, m_2, m_3) \), then encrypts as follows

\[
\begin{align*}
\text{Encryption of the message for the receiver} & \text{ } A_1 \\
& \text{B considers the public key of} \text{ } A_1 \text{ as } \text{ } S_1 = (k_{11}, k_{12}, k_{13}, k_{14}) = (0, s_{11}, s_{12}, s_{13}) = (0, 231, 1870, 210) \text{ and encrypts the message as} \\
& C_{A1} = k_{11}p_1 + k_{12}p_2 + k_{13}p_3 + k_{14}p_4 = 0.p_1 + s_{11}p_2 + s_{12}p_2 + s_{13}p_3 \\
& = 0 \times 0 + 231 \times 3 + 1870 \times 2 + 210 \times 1 = 4643.
\end{align*}
\]

**Encryption of the message for the receiver \( A_2 \)**

- \( B \) considers the public key of \( A_2 \) as \( S_2 = (k_{21}, k_{22}, k_{23}, k_{24}) = (s_{21}, s_{22}, s_{23}, s_{24}) = (127050, 70785, 176176, 286650) \) and encrypts the message as

\[
\begin{align*}
\text{Encryption of the message for the receiver} & \text{ } A_2 \\
& \text{B considers the public key of} \text{ } A_2 \text{ as } \text{ } S_2 = (k_{21}, k_{22}, k_{23}, k_{24}) = (s_{21}, s_{22}, s_{23}, s_{24}) = (127050, 70785, 176176, 286650) \text{ and encrypts the message as} \\
& C_{A2} = k_{21}p_1 + k_{22}p_2 + k_{23}p_3 + k_{24}p_4 = s_{21}p_1 + s_{22}p_2 + s_{23}p_3 + s_{24}p_4 \\
& = 127050 \times 0 + 70785 \times 3 + 176176 \times 2 + 286650 \times 1 \\
& = 851357.
\end{align*}
\]

- \( B \) sends the encrypted messages \( C_{A1} \) and \( C_{A2} \) to the receivers \( A_1 \) and \( A_2 \) respectively.

**Decryption by the receivers \( A_1 \) and \( A_2 \)**

**Step 1:**
- \( A_1 \) and \( A_2 \) decrypt the original message \( M = (m_1, m_2, m_3) = (3, 2, 1) \) using their private keys \( (q_{11}, q_{12}, q_{13}) \) and \( (q_{21}, q_{22}, q_{23}, q_{24}) \) respectively.

- \( A_1 \) decrypts \( M \) as follows

\[
\begin{align*}
& m_1 \pmod{q_{21}} = 4643 \mod 10 = 3 \\
& M_2 \equiv C_{A1} \mod q_{22} = 4643 \mod 21 = 2 \\
& m_3 \equiv C_{A1} \mod q_{23} = 4643 \mod 11 = 1
\end{align*}
\]

- \( A_1 \) decrypts the message as

\[
M = (0, 3, 2, 1) \text{ and obtains the original message as } M = 32
\]

- \( A_2 \) decrypts the message \( M \) as

\[
\begin{align*}
& m_1 \equiv C_{A2} \mod q_{21} = 851357 \mod 13 = 0 \\
& m_2 \equiv C_{A2} \mod q_{22} = 851357 \mod 14 = 1 \\
& m_3 \equiv C_{A2} \mod q_{23} = 851357 \mod 15 = 2 \\
& m_4 \equiv C_{A2} \mod q_{24} = 851357 \mod 121 = 1
\end{align*}
\]

- \( A_2 \) decrypts the message as \( M = (0, 3, 2, 1) \) and obtains the original message as \( M = 321 \).

The two receivers decrypt the original message \( M = 321 \) written in different forms using their respective public keys.

**Note 4.3.** The message \( M = (0, 3, 2) \) is the particular solution of the following system of Linear Diophantine Equation;

\[
\begin{align*}
231x_2 + 1870x_3 + 210x_4 &= 4643 \\
(127050) x_1 + 70785x_2 + 176176x_3 + 286650x_4 &= 851357.
\end{align*}
\]

**Remark 4.4.** In this scheme on system of linear Diophantine equations as the number of equations in the system is equal to the number of receivers and the number of message components is the number of variables, note the number of receivers at a time cannot be more than the number of message components.

**V SECURITY OF THE CRYPTOSYSTEM**

The complexity of the encryption is based on finding the message as a particular solution of the linear Diophantine equation. The time complexity for parametric form of solution by algorithm in [15] of crushing method is \( \frac{n \log_2 n}{3} \).

Obtaining the message from this infinitely many possibilities is difficult, as this exhaustive search is based on the size of the
component $m_i$ and the number $t$, note if each message component $m_i$ is a decimal number of $d$ digits, then the size is $< 10^d$ and if $t$ is the number of message components then $(10^d)^t$ is the number of possible searches for message tuple $(m_1, m_2, \ldots, m_t)$. so increasing $d$ and $t$ as per the requirement improves the efficiency.

VI CONCLUSION

In this paper, we proposed two encryption schemes based on linear Diophantine equations and system of linear Diophantine equations. The message in each scheme are encrypted as solutions of linear Diophantine equations and system of linear Diophantine equations respectively, there by the security depends on computing the particular solution from infinitely many possibilities with the time complexity based on the size and the number of message components, and the number of public key components. The advantage of the proposed scheme is that the computations of encryptions for sender and decryption for reciever are efficient. For encryption, it requires $r$ multiplication operations and $r$ addition operations. And for decryption $r$ modulus operations are required, with public and private keys of $r$ tuple. Thus, from the computation time view of point, this encryption is efficient. The security can be increased by controlling the message components size and number.

REFERENCES
