Journal of Iraqi Al-Khwarizmi (JIKh) Volume:8 Issue:2 Year: 2024 pages: 1-17

Secure Knapsack Cryptosystem Against LLL Algorithm Based on Continued Fraction

Authors Names	**ABSTRACT**
Abdulsalam A khaleell ***Article History*** Publication data: 4 / 4 /2024 ***Keywords:*** *Cryptosystem, LLL algorithm, Continued fraction, Knapsack Problem.*	The Merkel-Hellman Knapsack cryptosystem is classified as a public key cryptosystem that uses two dissimilar keys to encrypt and decrypt, providing a high level of security to these systems. It is understood that the key for decryption cannot be derived from the key for encryption. Unfortunately, it is not immune to cryptosystem attacks, such as the Lenstra, Lenstra, and Lovasz (LLL Algorithm) which can break it. In this paper, we proposed a method to enhance the security of the Knapsack Problem using continued fractions. This method aims to prevent the problem from being compromised by the LLL algorithm and reduce the size of encrypted data, thereby facilitating faster transmission of data. The proposed method demonstrates that the Merkel-Hellman knapsack cryptosystem enhances security as the LLL algorithm is unbreakable, and reduces the size of encrypted data. Moreover, the process of sending and receiving encrypted data is faster because the size of the encrypted data becomes smaller.

1.Introduction

Large data transmission processes are a result of the tremendous advancements in technology and the Internet in the modern world. Additionally, the increase in the number of devices that send and receive data, along with the fact that many of these devices carry out multiple tasks autonomously, contribute to this phenomenon. This data must be protected against attacks immediately as it contains sensitive and private information. Information may be protected in a variety of methods. This issue can be prevented by studying disciplines related to cryptography, which limit access to sensitive or essential data to employees with the appropriate authorization. Cryptography can essentially secure information by providing robust protection for confidentiality and ensuring the authenticity and integrity of data.

The term "cryptography" comes from the Greek language. It is a compound word with two meanings: "crypto" meaning hidden and "grafia" meaning writing. Most importantly, several types of these systems have been invented. The most widely-known cryptosystem and the earliest public key cryptographic (PKC) system in the world was proposed in 1976 by Martin E. Hellman, an American cryptologist, and Ralph C. Merkle, the computer scientist. Cryptosystems are called Public Key Cryptography (PKC) because two different keys are required for both decryption and encryption processes. The first key, known as the public key, is used to encrypt data, while the second key, known as the private key, is used to decrypt data. It is impossible to derive the decryption key from the encryption key.

Most importantly, the knapsack cryptosystem represents one of the earliest public key cryptosystems (PKCs). It is based on additive number theory [4]. Since the technique was first proposed in the 1970s, several versions of this system have been developed, including the multiplicative knapsack cryptosystem [5]. Chor-Rivest knapsack cryptosystem [6].

Islamic Azad University, Iran, Email: salamaljubory9495@gmail.com

The Graham-Shamir knapsack [7], the Naccache-Stern Knapsack [8], and the Super-Pascal Triangle Knapsack [9]. Unfortunately, most knapsack cryptosystems proposed by technicians

so far are not sufficiently reliable in terms of security to withstand cryptanalysis attacks. This type of attack can exploit vulnerabilities in cryptosystem designs, allowing it to overcome these systems. During the early 1980s, Adi Shamir presented an approach to overcome the Merkel-Hellman Knapsack Cryptosystem (MHKC). This approach successfully decrypted encoded text in polynomial time without the need for the private key, he was able to break the MHKP, marking it as the first cryptanalytic approach of this kind [10].

Furthermore, Andrew Michael Odlyzko and Jeffrey Clark Lagarias proposed another approach using a lattice reduction algorithm. Lovász László, Hendrik Lenstra, and Arjen Lenstra originally invented this algorithm, which is also known as the LLL Algorithm. The purpose of the algorithm is to find the shortest vector in a lattice [9].

The Merkel-Hellman knapsack problem (MHKP) system is characterized by having some weaknesses, including susceptibility to attacks using the LLL algorithm [4]. To avoid the LLL algorithm attack, this paper introduces the concept of utilizing continued fractions.

2. The LLL Basic Reduction algorithm

This algorithm aims to find a lattice's new basis with vectors of minimized length. Additionally, the new basis vectors are closer to orthogonality, and this algorithm can achieve that.

It is possible to apply the Gram-Schmidt Orthogonalization (GSO) process to a given set of basis vectors (BVs) when the process occurs in R. Although the entire processes related to a lattice should stay in Z, therefore, the algorithm cannot be applied to GSO simply, but it must be modified to keep the process in the lattice framework [11].

The algorithm starts by computing the Gram-Schmidt orthogonalization (GSO) and then uses this information to analyze the initial set of base vectors still present in the lattice.

Assume that there were n original Basis Vectors (BVs) called y_1 through y_n. when the computing the GSO of the original vectors $y_1, y_2, \dots\dots\ y_n$ a series of projections of the form;

$$y_i^* = b_i - i \overset{n}{=} 1^{u_{i,j} y_j^*} \qquad (1)$$

Where $u_{i,j} = \frac{b_i\, y_j^*}{y_j^*\, y_j^*}$ (2) are employed in computing process of new base vectors. Assume the original basis vector was put in the matrix columns labelled Y. Then, we could put the orthogonal vectors, which Gram-Schmidt compute, in matrix Y^* columns.

It could refer to the matrix as the matrix product $Y^* = Y\,.\,U$ where U is an Upper Triangle Matrix (UTM) of the formula.

$$u = \begin{bmatrix} 1 & -u_{1,2} & -u_{1,3} & \dots & -u_{1,n} \\ 0 & 1 & -u_{2,3} & \dots & -u_{2,n} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -u_{n-1,n} \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \qquad (3)$$

Naturally, if every $u_{i,j}$ was an integer, then U would be an unimodular matrix, and Y^* would be a collection of orthogonal vectors that formed the lattice's basis; however, this is not usually the case.

The real algorithm begins with a basis $b_1, b_2, \dots \dots\ b_n$ of a lattice L in R^n.

The algorithm copies the BVs b_i which it changes because of generating the reduced basis. These copied BVs are labelled $y_1, y_2, \dots \dots\ y_n$, the algorithm first computes the GSO of the BVs for $i = 1, 2, \dots \dots, n$, using orthogonal vectors $y_1^*, y_2^*, \dots \dots, y_n^*$

Where $y_i^* = b_i - i \overset{n}{=} 1^{u_{i,j} y_j^*}$ (3) ,for all $i > j$ and $u_{i,j}$ is $\frac{b_i\, y_j^*}{y_j^*\, y_j^*}$.

During the execution of the LLL algorithm, the vectors $y_1, y_2, \dots \dots\ y_n$ are repeatedly modified to consistently provide a basis for the lattice L composed of shorter vectors. The changes in the basis vectors would need a continuous update of the $u_{i,j}$ values, which are the new entries in the lattice's collection of basis vectors.

The procedure exchanges the lowest vector obtained to maintain a collection of lattice basis vectors organized from shortest to longest.

The algorithm starts by vector two, since GSO does not change vector one, and the value of $u_{2,1} = \frac{y_2 y_1}{y_1 y_1}$ has already been calculated Gram- Schmidt.

The 2nd vector that Gram-Schmidt calculates might or might not be in the lattice. A new vector in the lattice was computed to find the closest integer to $u_{2,1}$ and to calculate the vector $y_2 = y_2 - u_{2,1}\, y_i$.

This is the same computational process that GSO performs except GSO does not perform rounding. This is a new vector in the lattice since $u_{2,1}$ is an integer. When $|u_{2,1}| < \frac{1}{2}$, $u_{2,1} = 0$ and y_2 remains with no change.

When the algorithm is not calculating a new y_2, the value of $u_{2,1}$,1 from GSO will be updated.

Here, it is necessary to make sure by checking if or if not y_2 must should be exchanged with y_1, that can be conducted when y_2 was shorter than y_1 and when the changes happen, the $u_{i,j}$ are updated accordingly (i.e, $u_{i,j}$, values requiring the exchanged vectors).

Currently, vector y_3 is considered $u_{3,2}$ and is checked, when it $< \frac{1}{2}$, no steps are taken. when it is $\geq \frac{1}{2}$, the closest integer will replace it and y_3 is calculated again as $y_3 = y_3 - u_{3,2} y_2$. Because of using the closest integer, the computed vector would be in the lattice.

3. Finite Simple Continued Fractions

What is a Continued Fraction (C.F.)? [12-15].

Definition 1

CF is an expression of formula[16]

$$q_0 + \cfrac{d_0}{q_1 + \cfrac{d_1}{q_2 + \cfrac{d_2}{q_3 + \cfrac{d_3}{q + \ddots}}}} \qquad (4)$$

Where $q_0, q_1, q_2, \dots \dots\ q_n\ \dots., d_1, d_2, d_3, \dots$ are numbers (either complex or real).

Definition 2

A simple (regular) CF is the formula's CF[16].

$$q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{q_4 + \ddots}}}} \qquad (5)$$

Where is q_i an integer for all i with $q_0, q_1, q_2, \dots . > o$.

The integers $q_i, i = 0,1,2 \dots ..$ termed partial quotients of the C.F. A simple CF has either infinite or finite forms.

Definition 3

A simple CF with a finite number of terms is called a finite simple CF[16].

In symbols:

$$q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots + \cfrac{1}{q_{n-1} + \cfrac{1}{q_n}}}}} \qquad (6)$$

It can be named an n^th^-order CF and possesses (n+1) elements (partial quotients). It is widely used for expressing the finite simple CF as

$q_0 + \frac{1}{q_1} \frac{1}{q_2} \frac{1}{q_3} \dots \frac{1}{q_n}$ or simply as $[q_0,\ q_1, q_2 \dots\ q_n]$. .

4.The Merkle-Hellman knapsack cryptosystem

One of the earliest public-key cryptosystems suggested was MHKC [5]. This cipher makes use of a few simple yet effective mathematical concepts. This text involves some elementary mathematical concepts, but they are quite clever. It is possible to explain the knapsack problem as follows:

Given a set of n-weights,

$$R = (r_1, r_2, \dots \dots, r_n)$$

and a sum Z, find $z_1, z_2, \dots \dots, z_n$, where each $z_i \in \{0,1\}$, so that

$$Z = z_1 r_1 + z_2 r_2 + \cdots + z_n r_n \qquad (7)$$

Given that, this can happen. Notice, that the z_i selects simply the weights' subset, i.e., assume that the weights are R = (4,3,9,1,12,17,19,23) and the given sum is Z = 35. Then, there is the solution to the sub-set problem and it can be obtained by Z = (01011010), since $4 * 0 + 3 * 1 + 9 * 0 + 1 * 1 + 12 * 1 + 17 * 0 + 19 * 1 + 23 * 0 = 35$

Regarding to the set of weights, when Z = 8, that means that there is no solution to the problem.

5. Proposed algorithm

Although the LLL work method is known to be able to break a knapsack cryptosystem [11, 17–18], in this work, we introduce a unique knapsack cryptosystem technique based on continued fractions. The suggested process makes the knapsack cryptosystem more vulnerable to attacks using the LLL algorithm. The suggested algorithm may be outlined in the following steps:

Step 1: Assume Alice builds her super growing knapsack with modulus n and q^{-1} as $R = \left(r_1, r_2, \dots\dots, r_j\right)$, where $j \in N$(natural number).

Step 2: By computing $p_i = q\, r_i (mod\, n),\ for\ i = 1,2,3, \dots, j$ (8). Then, $P = \left\{p_1, p_2, p_3, \dots., p_j\right\}$, Alice gets the general knapsack P, P is the public key, whereas the private key is R and $q^{-1}(mod\, n)$.

Step 3: Bob encrypts the message and transmits it to Alice, by putting the message "M" in the form of a continuing fraction and converting it to binary to encode it as "m" Using the table of Corresponding Integers to Letters (Table 1).

Step 4: He calculate cipher text C as $C = mP$ (9), which is then transmitted to Alice as seen in figure 1.

Step 5: Alice calculates q^{-1} using Euclidean Algorithm, then, she calculates $w = C * q^{-1}(mod\, n)$, where W ∈ N (N: natural number) (10) and using the private key, after Alice receives the encrypted message, she decodes it further to obtain the original text, as seen in figure 1.

5.1 Particular Example

To illustrate this algorithm, we will take the same message in the previous example as shown in the following example:

Step 1: Let's say Alice builds her incredibly expanding satchel as R= (2, 3, 7, 13, 27, 53, 107, 213, 427, 853, 1707, 3415, 6819, 13636, 27275, 54549, 109097, 218195, 436389, 872778) with q = 1000003 and modulus n = 1745557.

Step 2: Alice computes p_i to obtain the generic knapsack P.

$Where\ p_i = 1000003\ R_i\ (mod\ 1745557),\ for\ i = 1,2, \dots, 20$. Then,

$$P = \begin{Bmatrix} 254449, 1254452, 17793, 781140, 816726, 633449, 521344 \\ 42685, 1085373, 1170743, 1595932, 700753, 874815, 1495181, \\ 753700, 507397, 14791, 1029585, 1059167, 372777 \end{Bmatrix}.$$

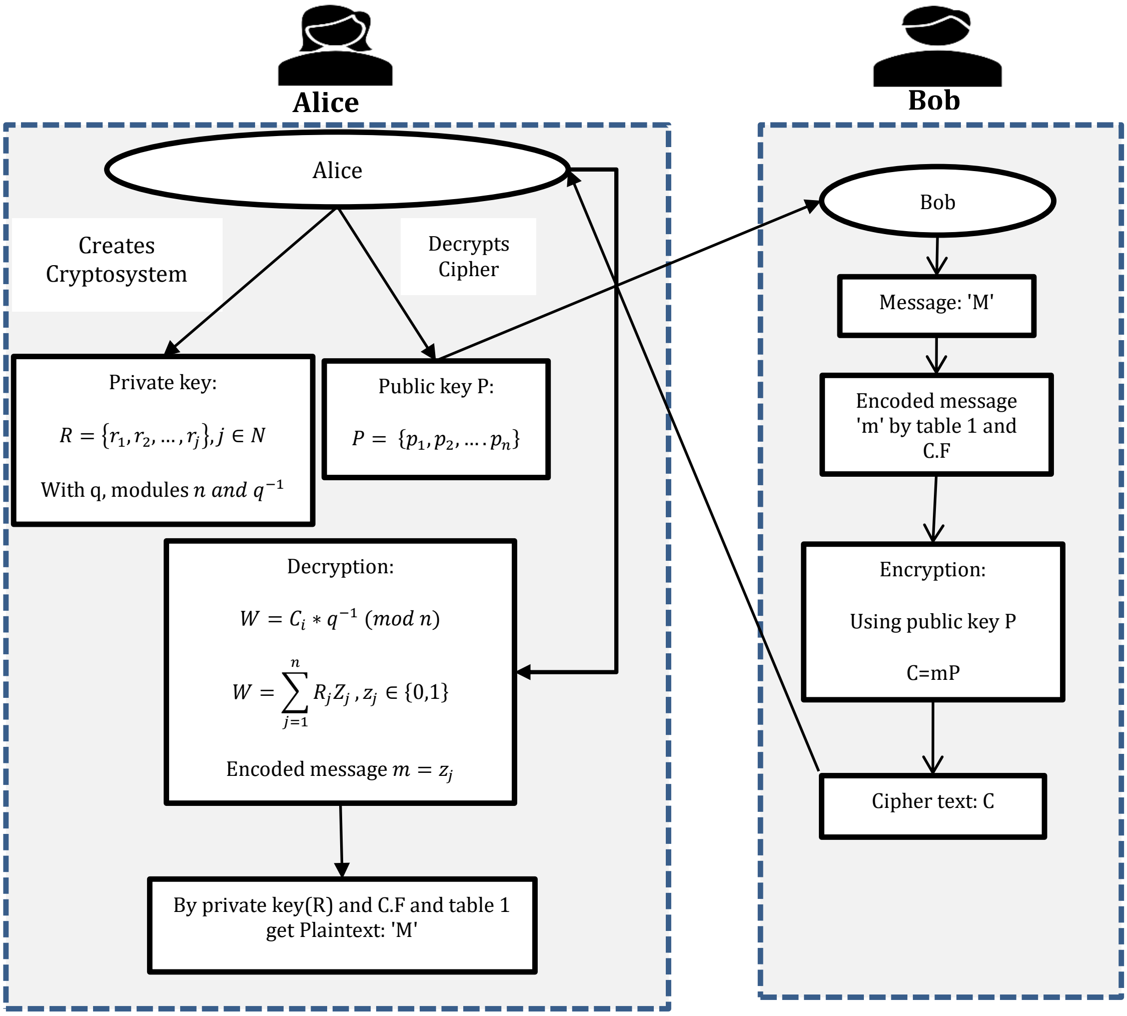
Alice
Bob
Alice
Creates Cryptosystem
Decrypts Cipher
Private key:
$R = \{r_1, r_2, \dots, r_j\}, j \in N$
With q, modules n and q^{-1}
Public key P:
$P = \{p_1, p_2, \dots . p_n\}$
Decryption:
$W = C_i * q^{-1}\ (mod\ n)$
$W = \sum_{j=1}^{n} R_j Z_j, z_j \in \{0,1\}$
Encoded message $m = z_j$
By private key(R) and C.F and table 1 get Plaintext: 'M'
Bob
Message: 'M'
Encoded message 'm' by table 1 and C.F
Encryption:
Using public key P
C=mP
Cipher text: C

Fig. 1 - Encryption and decryption.

Step 3: Suppose Bob wishes to transmit Alice the encrypted message "Ahmed Abdalrhman Mohsn" in four blocks. Bob uses table 1 to obtain the integers that correspond to the characters in this message, which is table 2.

Step 4: As can be seen below, he works in four blocks and writes them as a continuous fraction.

First block:

$$1 + \cfrac{1}{8 + \cfrac{1}{13 + \cfrac{1}{5 + \cfrac{1}{4}}}}$$

Which will be equivalent to $\frac{2514}{2237}$, then he transforms it to binary as:

Decimal	Binary
2514	100111010010
2237	100010111101

Table 1 - Corresponding integers to letters.

Letters	**the Integers Corresponding to letters**
A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
I	9
J	10
K	11
L	12
M	13
N	14
O	15
P	16
Q	17
R	18
S	19
T	20
U	21
V	22
W	23
X	24
Y	25
Z	26

Table 2 - Corresponding integers to letters of the message.

Message	**"Ahmed Abdalrhman Mohsn"**																			
blocks	**First block**					**Second block**					**Third Block**					**Fourth block**				
	AHMED					**ABDAL**					**RHMAN**					**MUHSN**				
	A	H	M	E	D	A	B	D	A	L	R	H	M	A	N	M	U	H	S	N
Integers Corresponding to letters	1	8	13	5	4	1	2	4	1	12	18	8	13	1	14	13	21	8	19	14

Second block:

$$1+\frac{1}{2+\frac{1}{4+\frac{1}{1+\frac{1}{12}}}}$$

Which will be equivalent to $\frac{205}{141}$, then he transforms it to binary as:

Decimal	**Binary**
205	11001101
141	10001101

Third block:

$$18+\frac{1}{8+\frac{1}{13+\frac{1}{1+\frac{1}{14}}}}$$

Which will be equivalent to $\frac{30575}{1687}$, then he transforms it to binary as:

Decimal	**Binary**
30575	111011101101111
1687	11010010111

Fourth block:

$$13+\frac{1}{21+\frac{1}{8+\frac{1}{19+\frac{1}{14}}}}$$

Which will be equivalent to $\frac{592571}{45417}$, then he transforms it to binary as:

Decimal	**Binary**
592571	10010000101010111011
45417	1011000101101001

Step 5: Bob now calculates the cipher text C to the fourth blocks as $C = mP,$

where m is Message, P is public key, and we get two cipher text C_1, C_2 for any block from the fourth blocks, that mean we get eight cipher texts in the total in this example as show in below.

Encryption first block:

$$C_1 = 0 * p_1 + 0 * p_2 + 0 * p_3 + 0 * p_4 + 0 * p_5 + 0 * p_6 + 0 * p_7 + 0 * p_8 + 1 * p_9 + 0 * p_{10} + 0 * p_{11} + 1 * p_{12} + 1 * p_{13} + 1 * p_{14} + 0 * p_{15} + 1 * p_{16} + 0 * p_{17} + 0 * p_{18} + 1 * p_{19} + 0 * p_{20}$$

Then

$$C_1 = 0 * 254449 + 0 * 1254452 + 0 * 17793 + 0 * 781140 + 0 * 816726 + 0 * 633449 + 0 * 521344 + 0 * 42685 + 1 * 1085373 + 0 * 1170743 + 0 * 1595932 + 1 * 700753 + 1 * 874815 + 1 * 1495181 + 0 * 753700 + 1 * 507397 + 0 * 14791 + 0 * 1029585 + 1 * 1059167 + 0 * 372777 = 5722686$$

And,

$$C_2 = 0 * p_1 + 0 * p_2 + 0 * p_3 + 0 * p_4 + 0 * p_5 + 0 * p_6 + 0 * p_7 + 0 * p_8 + 1 * p_9 + 0 * p_{10} + 0 * p_{11} + 0 * p_{12} + 1 * p_{13} + 0 * p_{14} + 1 * p_{15} + 1 * p_{16} + 1 * p_{17} + 1 * p_{18} + 0 * p_{19} + 1 * p_{20}$$

Then,

$$C_2 = 0 * 254449 + 0 * 1254452 + 0 * 17793 + 0 * 781140 + 0 * 816726 + 0 * 633449 + 0 * 521344 + 0 * 42685 + 1 * 1085373 + 0 * 1170743 + 0 * 1595932 + 0 * 700753 + 1 * 874815 + 0 * 1495181 + 1 * 753700 + 1 * 507397 + 1 * 14791 + 1 * 1029585 + 0 * 1059167 + 1 * 372777 = 4638438$$

Encryption Second Block:

$$C_1 = 0 * p_1 + 0 * p_2 + 0 * p_3 + 0 * p_4 + 0 * p_5 + 0 * p_6 + 0 * p_7 + 0 * p_8 + 0 * p_9 + 0 * p_{10} + 0 * p_{11} + 0 * p_{12} + 1 * p_{13} + 1 * p_{14} + 0 * p_{15} + 0 * p_{16} + 1 * p_{17} + 1 * p_{18} + 0 * p_{19} + 1 * p_{20}$$

Then

$$C_1 = 0 * 254449 + 0 * 1254452 + 0 * 17793 + 0 * 781140 + 0 * 816726 + 0 * 633449 + 0 * 521344 + 0 * 42685 + 0 * 1085373 + 0 * 1170743 + 0 * 1595932 + 0 * 700753 + 1 * 874815 + 1 * 1495181 + 0 * 753700 + 0 * 507397 + 1 * 14791 + 1 * 1029585 + 0 * 1059167 + 1 * 372777 = 3787149$$

And,

$$C_2 = 0 * p_1 + 0 * p_2 + 0 * p_3 + 0 * p_4 + 0 * p_5 + 0 * p_6 + 0 * p_7 + 0 * p_8 + 0 * p_9 + 0 * p_{10} + 0 * p_{11} + 0 * p_{12} + 1 * p_{13} + 0 * p_{14} + 0 * p_{15} + 0 * p_{16} + 1 * p_{17} + 1 * p_{18} + 0 * p_{19} + 1 * p_{20}$$

Then,

$$C_2 = 0*254449 + 0*1254452 + 0*17793 + 0*781140 + 0*816726 + 0*633449 + 0*521344 + 0*42685 + 0*1085373 + 0*1170743 + 0*1595932 + 0*700753 + 1*874815 + 0*1495181 + 0*753700 + 0*507397 + 1*14791 + 1*1029585 + 0*1059167 + 1*372777 = 2291968$$

Encryption third Block:

$$C_1 = 0*p_1 + 0*p_2 + 0*p_3 + 0*p_4 + 0*p_5 + 1*p_6 + 1*p_7 + 1*p_8 + 0*p_9 + 1*p_{10} + 1*p_{11} + 1*p_{12} + 0*p_{13} + 1*p_{14} + 1*p_{15} + 0*p_{16} + 1*p_{17} + 1*p_{18} + 1*p_{19} + 1*p_{20}$$

Then

$$C_1 = 0*254449 + 0*1254452 + 0*17793 + 0*781140 + 0*816726 + 1*633449 + 1*521344 + 1*42685 + 0*1085373 + 1*1170743 + 1*1595932 + 1*700753 + 0*874815 + 1*1495181 + 1*753700 + 0*507397 + 1*14791 + 1*1029585 + 1*1059167 + 1*372777 = 9390107$$

And,

$$C_2 = 0*p_1 + 0*p_2 + 0*p_3 + 0*p_4 + 0*p_5 + 0*p_6 + 0*p_7 + 0*p_8 + 0*p_9 + 1*p_{10} + 1*p_{11} + 0*p_{12} + 1*p_{13} + 0*p_{14} + 0*p_{15} + 1*p_{16} + 0*p_{17} + 1*p_{18} + 1*p_{19} + 1*p_{20}$$

Then,

$$C_2 = 0*254449 + 0*1254452 + 0*17793 + 0*781140 + 0*816726 + 0*633449 + 0*521344 + 0*42685 + 0*1085373 + 1*1170743 + 1*1595932 + 0*700753 + 1*874815 + 0*1495181 + 0*753700 + 1*507397 + 0*14791 + 1*1029585 + 1*1059167 + 1*372777 = 6610416$$

Encryption fourth Block:

$$C_1 = 1*p_1 + 0*p_2 + 1*p_3 + 0*p_4 + 0*p_5 + 0*p_6 + 0*p_7 + 0*p_8 + 1*p_9 + 0*p_{10} + 1*p_{11} + 0*p_{12} + 1*p_{13} + 0*p_{14} + 1*p_{15} + 1*p_{16} + 1*p_{17} + 0*p_{18} + 1*p_{19} + 1*p_{20}$$

Then

$$C_1 = 1*254449 + 0*1254452 + 0*17793 + 1*781140 + 0*816726 + 0*633449 + 0*521344 + 0*42685 + 1*1085373 + 0*1170743 + 1*1595932 + 0*700753 + 1*874815 + 0*1495181 + 1*753700 + 1*507397 + 1*14791 + 0*1029585 + 1*1059167 + 1*372777 = 7299541$$

And,

$$C_2 = 0*p_1 + 0*p_2 + 0*p_3 + 0*p_4 + 1*p_5 + 0*p_6 + 1*p_7 + 1*p_8 + 0*p_9 + 0*p_{10} + 0*p_{11} + 1*p_{12} + 0*p_{13} + 1*p_{14} + 1*p_{15} + 0*p_{16} + 1*p_{17} + 0*p_{18} + 0*p_{19} + 1*p_{20}$$

Then,

$$C_2 = 0*254449 + 0*1254452 + 0*17793 + 0*781140 + 1*816726 + 0*633449 + 1*521344 + 1*42685 + 0*1085373 + 0*1170743 + 0*1595932 + 1*700753 + 0*874815 + 1*1495181 + 1*753700 + 0*507397 + 1*14791 + 0*1029585 + 0*1059167 + 1*372777 = 4717957$$

Then, the cipher texts are:

$$C = \begin{Bmatrix} 5722686, 4638438, 3787149, 2291968, 9390107, 6610416, 7299541, \\ 4717957 \end{Bmatrix}$$

and it will be sent to Alice.

step 6: Assume that Rayan tries to recover the plaintext, As Rayan knows the public key P

$$P = \begin{Bmatrix} 254449, & 1254452, & 17793, 781140, & 816726, 633449, & 521344, \\ 42685, & 1085373, & 1170743, 1595932, & 700753, 874815, 1495181, 753700, \\ & 507397, & 14791, & 1029585, 1059167, 372777 \end{Bmatrix}$$

and cipher text C,

$$C = \begin{Bmatrix} 5722686, 4638438, 3787149, 2291968, 9390107, 6610416, 7299541, \\ 4717957 \end{Bmatrix}$$

She must determine a collection of u$_i$ for $i = 1, 2, \dots, 20$ with the restriction that each u$_i$ ∈ {0, 1}. Then,

254449 u_1 + 1254452 u_2 + 17793 u_3 + 781140 u_4 + 816726 u_5 +633449 u_6 + 521344 u_7 + 42685 u_8 + 1085373 u_9 + 1170743 u_{10} +1595932 u_{11} + 700753 u_{12} + 874815 u_{13} + 1495181 u_{14} + 753700 u_{15} +507397 u_{16} + 14791 u_{17} + 1029585 u_{18} + 1059167 u_{19} + 372777 u_{20} = Ci

where C_i cipher text, for all $i = 1, 2, 3, 4, 5, 6, 7, 8$.

The matrix equation may be expressed as follows: $P.U = C$

Then, Rayan rewrites the matrix equation as:

$$M * V = \begin{bmatrix} In*n & 0n*1 \\ Am*n & -Bm*1 \end{bmatrix} \begin{bmatrix} Un*1 \\ 1(1*1) \end{bmatrix} = \begin{bmatrix} Un*1 \\ 0(1*1) \end{bmatrix} = \mathrm{W} \qquad (11)$$

Applying the LLL algorithm to M. Hence, Rayan detects.

In case 1: $C_1 = 5722686,$ Rayan Obtained the following matrix.

$$M = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
254449 & 1254452 & 177093 & 781140 & 816726 & 633449 & 521344 & 42685 & 1085373 & 1170743 & 1595932 & 700753 & 874815 & 1495181 & 753700 & 507397 & 14791 & 1029585 & 1059167 & 372777 & -5722686
\end{bmatrix}$$

The LLL technique is used to create a matrix M' consisting of short vectors inside the lattice formed by the columns of matrix M.

$$M' = \begin{bmatrix}
1 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 2 & 3 & 0 & 0 & 0 \\
0 & -2 & -1 & -2 & 0 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & -2 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & -1 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\
1 & 0 & -1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -2 & -2 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

We note that she failed to get the plain text of first block, since the shortest vector is the First column and is not as the same original plain text.

We continue the same process for all the following cases,

C_1 = 5722686,

C_2 = 4638438,

C_3 = 3787149,

C_4 = 2291968,

C_5 = 9390107,

C_6 = 6610416,

C_7 = 7299541,

C_8 = 4717957

We notice the failure to obtain the plaintext as well.

6. Security analysis

- We demonstrated that the suggested approach is impervious to attacks by the LLL algorithm using the aforementioned scenario.
- We want to determine if there were any other criticisms of the proposed method in this paragraph. An attacker attempting to access the encrypted text "CUP" will be unable to retrieve the original message since it is a coded message. The attacker will be unable to decipher the plaintext of the

message by determining the value of C = U P, as it represents a coded message rather than the actual content. The attacker will not be able to decrypt the message's plaintext if they try to identify the value of C = U.P. or C = U.M.R. due to the unknown values (M, R, and U).

- The effectiveness of the attack should be somewhat reduced by recurring fractions. The examination of frequency, a well-known cryptanalysis technique, relies on finding repeated data. Wanton force attacks function by attempting to access multiple keys, decrypt the data, and ascertain the significance of the resulting data. An attacker must first decrypt the data using CF, then decode it, and finally verify that the resulting data makes sense. He will have to go through a more drawn-out and difficult procedure. If he has any knowledge of the data's coding at all, he will probably never be able to crack the encryption.
- Our proposed solution is impervious to a quantum assault for three reasons [20-21].
  First of all, even with quantum computers, the communication is encrypted and then further secured with encryption. Therefore, even if an adversary manages to access the ciphertext, they will be unable to obtain the plaintext.
  Second, the ciphertext produced after the encoding process is akin to data compression. For example, if the plaintext message contains 60 letters and the block size is 5, the output of the ciphertext will be approximately 20 letters. As a result, one-third of the message's information is concealed from the attacker. The larger the block size, the more hidden information there is.
  Third: There is no obvious padding; the block's length (60) must be split, resulting in the following letter counts for each block: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60.

7. Conclusions

The concept of continued fractions was utilized in our proposed method to enhance the security of the Merkle-Hellman Knapsack cryptosystem, making it resistant to attacks by the LLL algorithm from being able to attack it.

Using continued fractions also has the advantage of providing shorter plaintext and ciphertext, which reduces the time needed for data transmission, encryption, and decryption. Certain cryptanalysis attempts may be hindered by the reduced redundancy of the plaintext. The model that was suggested resulted in heightened security.

References

[1] A. G. Konheim, “Mathematical cryptology for computer scientists and mathematicians. By Wayne Patterson: A course in number theory and cryptography. By Neal koblitz,” *Am. Math. Mon.*, vol. 96, no. 4, pp. 374–375, 1989.

[2] D. A. Charles and Assistant Professor Government College of Engineering, Bargur, India., “Merkle Hellman Knapsack Cryptosystem based Cloud Auditing Protocol,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 2168–2174, 2019.

[3] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[4] W. Diffie and M. Hellman, “New Directions in Cryptography (1976),” in *Ideas That Created the Future*, The MIT Press, 2021, pp. 421–440.

[5] R. Merkle and M. Hellman, “Hiding information and signatures in trapdoor knapsacks,” *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 525–530, 1978.

[6] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[7] B. Chor and R. L. Rivest, “A knapsack-type public key cryptosystem based on arithmetic in finite fields,” *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 901–909, 1988.

[8] S. Vaudenay, “Cryptanalysis of the chor—Rivest cryptosystem,” *J. Cryptology*, vol. 14, no. 2, pp. 87–100, 2001.
[9] Carleton.ca. [Online]. Available: http://people.scs.carleton.ca/~maheshwa/courses/4109/Seminar11/knapsack. [Accessed: 14-Feb-2024].
[10] D. Naccache and J. Stern, “A New Public-Key Cryptosystem,” in *Advances in Cryptology — EUROCRYPT ’97*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 27–36.
[11] J. Divasón, S. Joosten, R. Thiemann, and A. Yamada, “A formalization of the LLL basis reduction algorithm,” in *Interactive Theorem Proving*, Cham: Springer International Publishing, 2018, pp. 160–177.
[12] C. D. Olds, *Continued Fractions*. Washington DC: The Mathematical Association of America, 1963.
[13] J. Samuel Wayne, *Patterns in Continued Fraction Expansions*. 2013.
[14] H. Davenport, *The Higher Arithmetic: An Introduction to the Theory of Numbers, Seventh Edition*. Cambridge University Press, 1999.
[15] A. Y. Khinchin, *Continued Fractions*. Mineola, NY: Dover Publications, 1997.
[16] A. Bassam, *Invitation to Number Theory, First Edition*. Al-Huda, 2003.
[17] P. Q. Nguyen and D. Stehlé, “An LLL algorithm with quadratic complexity,” *SIAM J. Comput.*, vol. 39, no. 3, pp. 874–903, 2009.
[18] O. Regev, *Lattices in Computer Science: LLL Algorithm*. 2019.
[19] Z. Rifaat, *Quantum enecryption algorithim based on modified BB84 and authentication DH algorithm*. 2015.
[20] A. A. Abdullah, *Modified Quantum Three Pass Protocol Based on Hybrid Cryptosystem*. 2015.