# CS641

Modern Cryptology
Indian Institute of Technology, Kanpur

Group Name: ANV
Dibbu Amar Raja (21111009), Idamakanti Venkata Nagarjun Reddy (21111031), Vikas (21111067)

# End Semester Examination

Submission Deadline:
May 5, 2022, 11:55hrs

---

## Solution 1

### Lattice

Q a):- Given:

$L \epsilon Z^{nxn}$, be the matrix defined as: L = n · I , Here I is an identity matrix of size nxn
Let $U \epsilon Z^{nxn}$ be a unitary matrix with det U = 1.
Let $R \epsilon Q^{nxn}$ be a rigid rotation matrix, that is, $RR^T = I$
$\hat{L} = ULR$.
Public key is the matrix $\hat{L}$. Private key is the matrix R

**We will now prove that the lattice generated by $\hat{L}$ has a basis consisting of n orthogonal vectors, each of length n.**

Let the lattice generated by taking $\hat{L}$ as basis be 'z'.
$U \epsilon Z^{nxn}$ be a unitary matrix, that is, det U = 1.
A unitary matrix with real entries is orthogonal.
So, we can say that $U.U^T = I$
It is given that $RR^T = I$, We can say that R is an orthogonal matrix.
It is given that $\hat{L} = ULR$
$\hat{L} = U.n.I.R$
$\hat{L} = n.U.I.R$
$\hat{L} = n.U.R$

We know that product of two orthogonal matrices is orthogonal.

*Proof.* $\Rightarrow$
Let A, B be 2 orthogonal matrices.
$\Rightarrow A.A^T = I$ and $B.B^T = I$
Product of these two matrices is A.B
Let us take $(A.B).(A.B)^T$
This can be written as
$\Rightarrow (A.B).(B^T.A^T)$
$\Rightarrow (A.B.B^T.A^T)$
$\Rightarrow (A.I.A^T)$
$\Rightarrow (A.A^T)$
$\Rightarrow I$
From this, we can say that product of two orthogonal matrices is orthogonal. $\square$

$\hat{L} = n.U.R$ , here, U is orthogonal and R is orthogonal. So, product of U and R is orthogonal too.
Let U.R be denoted by matrix P, P is an nxn matrix.
$\Rightarrow \hat{L} = n.P$
P is orthogonal
That means, P has n columns which are orthogonal to each other and each of the columns are of length 1.Then, if we multiply P with n, that is nP, we can say that nP has n columns which are orthogonal to each other and the length of the each column will be n.That means, $\hat{L}$ has has n columns which are orthogonal to each other and the length of the each column will be n. This means that the lattice z has a basis consisting of n orthogonal vectors, each of length n. This basis is $\hat{L}$.

(or)

**Another approach** :-

$\hat{L}.\hat{L}^T = (ULR).(ULR)^T$
$\hat{L}.\hat{L}^T = (ULR).(R^T L^T U^T)$
$\hat{L}.\hat{L}^T = (UL.I.L^T U^T) \qquad (\because R.R^T = I)$
$\hat{L}.\hat{L}^T = (ULL^T U^T)$

$\hat{L}.\hat{L}^T = (U.n.I.n.I.U^T)$ (given $L = n.I$)
$\hat{L}.\hat{L}^T = (U.n^2.I.U^T)$
$\hat{L}.\hat{L}^T = (n^2.U.U^T)$
$\hat{L}.\hat{L}^T = (n^2.I)$ ($\because U.U^T = I$)

$\hat{L}/n \,.\, \hat{L}^T/n = I \implies$ Now we can say that $\hat{L}/n$ is an orthogonal matrix each column of length "1".
($\because$ if A is a square matrix and $A.A^T = I$ then A is Orthogonal matrix )

$\hat{L}/n$ is an orthogonal martix, this means that $\hat{L}$ has has n columns which are orthogonal to each other and the length of the each column will be n. This means that the lattice z has a basis consisting of n orthogonal vectors, each of length n. This basis is $\hat{L}$.

## Decryption

Q b):- Given:
L = n.I
Let $U \epsilon Z$
n×n be a unitary matrix, that is, det U = 1. Let $R \epsilon Q^{nxn}$ be a rigid rotation matrix, that is, $RR^T = I$. Define $\hat{L} = ULR$. Public key is the matrix $\hat{L}$ and private key is the matrix R.

Encryption: Given an n-bit long message m, view it as a vector in $Z^n$ with binary entries. Pick a random vector $v \epsilon Z^n$ and compute the vector $c = v\hat{L} + m$. Output c.

Decryption: Given a vector $c \epsilon Q^n$, compute vector $d = cR^T$. Reduce every entry of d modulo n so that the entry becomes < n/2 in absolute value. Let the resulting vector be $\hat{d}$. Compute $m = \hat{d}R$.
Now, we will show that the decryption works.
$c = v\hat{L} + m$
We multiply c with $R^T$ and call it d.
$d = c.R^T$
$\Rightarrow d = (v.\hat{L} + m).R^T$
$\Rightarrow d = (v.\hat{L}.R^T + m.R^T)$
$\Rightarrow d = (v.ULR.R^T + m.R^T)$
$\Rightarrow d = (v.U.L.I + m.R^T)$
$\Rightarrow d = (v.U.n.I.I + m.R^T)$
$\Rightarrow d = (n.v.U + m.R^T)$
$d = (n.v.U + m.R^T)$
this is a cryptosystem. If n is small (< 10), security is very weak. So n should be assumed to be large. (As mentioned by Sir in Forum). So, we will assume that $n \geq 10$
According to decryption, Reduce every entry of d modulo n so that the entry becomes $< n/2$ in absolute value. Let the resulting vector be $\hat{d}$. We will do this.
Now, we will perform (mod n) on d. We will consider floating point modulus operation as R contains rational numbers
d modulo n = $(n.v.U + m.R^T)$ mod n
d modulo n =$(((n.v.U) mod n) + ((m.R^T) mod n))$
The matrix ((n. v. U) mod n) will become a zero matrix because v has integers and U has integers and we are multiplying v.U with the scalar 'n' and each value of n.v.U will be a

multiple of n. So, ((n.v.U) mod n) will result in a zero matrix of dimensions 1xn.
$\Rightarrow$ d modulo n = $((m.R^T)modn)$
Now, we will discuss about the entries of the matrix $m.R^T$
m is a 1xn matrix consisting of entries say $m_1, m_2, m_3..............m_n$
$m_1, m_2, m_3..............m_n$ take values 0, 1
Let us consider R has row vectors $r_1, r_2, ...............r_n$
Let us consider $r_1$ has elements $r_{11}, r_{12}, r_{13}, ..........................r_{1n}$
Let us consider r2 has elements $r_{21}, r_{22}, r_{23}, ..........................r_{2n}$ .
.
.
.
Let us consider $r_n$ has elements $r_{n1}, r_{n2}, r_{n3}, ..........................r_{nn}$
Let us consider $r_1$ has elements $r_{11}, r_{12}, r_{13}......r_{1n}$. Total n elements in $r_1$. As R is an orthogonal matrix, we can say that length of each column vector and each row vector in R is 1.
So, length of $r_1$ is 1, $r_2$ is 1,............... $r_n$ is 1.
This means $\sqrt{(r_{i1}^2 + r_{i2}^2 + r_{i3}^2..............r_{in}^2)} = 1$ (i takes values from 1 to n).
$\Rightarrow r_{i1}^2 + r_{i2}^2 + r_{i3}^2..............r_{in}^2 = 1$
If we transpose R, the row vectors become column vectors. That means, $r_1, r_2, ...............r_n$ are column vectors in $R^T$
If we multiply m by $R^T$, then $i^{th}$ element of $m.R^T$ would be $\sum_{j=1}^{n}$ (m$_j * r_{ij}$)
*For example , 1st element of* m.R$^T$ would be $m_1 * r_{11} + m_2 * r_{12} + m_3 * r_{13} + ...................m_n * r_{1n}$
We will try to find the maximum value and minimum value of the elements of $m.R^T$
Let us take the 1st element of $m.R^T$ i.e $m_1 * r_{11} + m_2 * r_{12} + m_3 * r_{13} + ...................m_n * r_{1n}$
we know that $r_{11}^2 + r_{12}^2 + r_{13}^2..............r_{1n}^2 = 1$
$m_1 * r_{11} + m_2 * r_{12} + m_3 * r_{13} + ...................m_n * r_{1n}$
The maximum of this element is achieved when $m_1, m_2, ..............m_n$ all of them are equal to 1 and $r_{11}, r_{12}, .............rr_{1n}$ are positive. So, we should maximize $r_{11} + r_{12} + r_{13}................r_{1n}$
The maximum value of $r_{11} + r_{12} + r_{13}................r_{1n}$ is achieved when all these values are equal.
$\Rightarrow r_{11} = r_{12}....... = rr_{1n} = z$
$\Rightarrow z^2 + z^2 + .........................ntimes = 1$
$\Rightarrow n * z^2 = 1$

$\Rightarrow z^2 = 1/n$
$\Rightarrow$ z = +1/ $\sqrt[2]{n}$ or -1/ $\sqrt[2]{n}$
As we want to maximize $r_{11} + r_{12} + r_{13}$................$r_{1n}$, we take z as +1/ $\sqrt[2]{n}$
So, $r_{11} = r_{12}$........... $= r_{1n}$ = 1/ $\sqrt[2]{n}$
So, max value of $m_1 * r_{11} + m_2 * r_{12} + m_3 * r_{13} +$ ..................$m_n * r_{1n}$ is 1/ $\sqrt[2]{n}$ + 1/ $\sqrt[2]{n}$ + .............n times
$\Rightarrow$ max value of $m_1 * r_{11} + m_2 * r_{12} + m_3 * r_{13} +$ ..................$m_n * r_{1n}$ is n/ $\sqrt[2]{n}$
$\Rightarrow$ max value of $m_1 * r_{11} + m2 * r_{12} + m3 * r_{13} +$ ..................$m_n * r_{1n}$ is $\sqrt[2]{n}$
We can also prove that maximum value of $r_{11} + r_{12} +$ .............$r_{1n}$ is $\sqrt[2]{n}$ using L1-L2 norm inequality.
Proof: According to L1-L2 norm inequality,
L1-norm $\leq$ $\sqrt[2]{n}$ * L2-norm
$\Rightarrow |r_{11}| + |r_{12}| +$ ..............$|r_{1n}| <=$ $\sqrt[2]{n}$* $\sqrt[2]{r_{11}^2 + r_{12}^2 + r_{13}^2 .............. r_{1n}^2}$
we take r$_{11}, r_{12}$.... *as positive values*
*we know* $r_{11}^2 + r_{12}^2 + r_{13}^2$..............$r_{1n}^2 = 1$
so, r$_{11} + r_{12} +$ ..............$r_{1n} \leq$ $\sqrt[2]{n}$
$\Rightarrow$ max value of r$_{11} + r_{12}$.................$r_{1n} is \sqrt[2]{n}$
So, max value of m1 * r$_{11} + m2 * r_{12} + m3 * r_{13} +$ ..................$mn * r_{1n} is \sqrt[2]{n}$ ( because to achieve max value , we take m1, m2, .............. as 1)

Similarly, we can show that minimum value of $m1 * r_{11} + m2 * r_{12} + m3 * r_{13} +$ ..................$mn * r_{1n}$ is -$\sqrt[2]{n}$
So, we can say that each element of $m.R^T$ lies in range [-$\sqrt[2]{n}$, $\sqrt[2]{n}$] inclusive
Also, We can say that each element of (($m.R^T$) mod n) lies in range [-$\sqrt[2]{n}$, $\sqrt[2]{n}$] inclusive ( since ($\sqrt[2]{n}$ mod n) is equal to $\sqrt[2]{n}$ for every $n > 1$ ) As we have assumed that $n \geq 10$, we can say that absolute value of (($m.R^T$) mod n) is less than n/2.
Proof $\Rightarrow$
Let us assume ($\sqrt[2]{n}$) $< n/2$
$\Rightarrow$ n/2> $\sqrt[2]{n}$
$\Rightarrow n >$ $\sqrt[2]{n}$*2
applying square on both sides
$\Rightarrow n^2 >$ n*4
$\Rightarrow n > 4$
So, for $\sqrt[2]{n}$ to be less than n/2, n should be greater than 4. As our assumption is $n \geq 10$, we can say that absolute value of (($m.R^T$) mod n) is less than n/2.

So, we can say that (($m.R^T$) mod n) is equal to ($m.R^T$)
We can now say that the resulting vector $\hat{d} = (m.R^T)$
We now compute $\hat{d}.R$
$\hat{d}.R = m.R^T.R$
As R is orthogonal, $R^T.R = R.R^T = I$
$\Rightarrow \hat{d}.R = m.I$
$\Rightarrow \hat{d}.R = m$
In this way, we have shown that the decryption works correctly and it gives us the message vector m.

## Cryptosystem Security

Q c):-

**Analysis**:

It is a public key cryptography system based on lattices , where public key is the matrix $\hat{L}$ and private key is the matrix $R$. Differential Cryptanalysis might not be possible as v is random in c = v$\hat{L}$ + m

**Breaking the crypto system:**

- Breaking the security of the system , by bruteforcing on all the possibilites of "U" takes very large amount of time and is not practically feasible.
- Breaking the security of the system , with a bruteforce approach through "m" message makes us go through $2^n$ possibilities of the message. ( We have described these two solutions in "Q d)")
- We found out that the optimal approach to break this crypto system is using the orthogonal basis of lattice generated by $\hat{L}$.

We will now analyze the security of the cryptosystem. In particular, we will show that if any orthogonal basis of $\hat{L}$ can be found, then the security is broken.

Let us now assume that an orthogonal basis of $\hat{L}$ has been found. Let us call this orthogonal basis as 'Q'.Given an orthogonal basis, we can break the security ( i.e decrypt the cipher ) by solving CVP (Closest Vector Problem).Given a lattice L and a target point x, CVP asks to find the lattice point closest to the target.

We can use Babai's algorithm to solve CVP.

Babai's algorithm takes a point 'r' and a set of basis vectors $[g_1, ..., g_n]$ as input. The algorithm then solves $r = a_1 * g_1 + ... + a_n * g_n$ where $[a_1, ..., a_n]$ are real number coefficients. Babai then approximates a solution to CVP by rounding all coefficients $a_1, ..., a_n$ to their nearest integer.For orthogonal bases, Babai works well and will return the closest lattice point to 'r'.If we have a target vector c, let c' be the closest vector found out using Babai's Algorithm and 'Q' be the orthogonal basis of the lattice.

For finding c', we do the following:

1) First we create an equation c = A.Q where c is the target vector i.e encrypted message, A is a matrix of dimensions 1xn, Q is the orthogonal basis.

2) On multiplying the above equation with $Q^{-1}$ on both sides, we get $A.Q.Q^{-1} = c.Q^{-1}$.

3) Now, $A.I = c.Q^{-1}$.
4) Now, $A = c.Q^{-1}$
5) After finding $Q^{-1}$,we do $c.Q^{-1}$ and from this, we will get the matrix A.
6) The values in A will be rational. We will round off the values in A to their nearest integers. Let us call the matrix after rounding off the values as A'.
7) Now, the closest vector c' will be c' = A'. Q .

In this way, we have found the closest vector c'. Now, let us calculate a vector
Z = c - c' . Now, we will create a vector by taking absolute values of the entries in Z and round them off to their nearest integers. The vector thus obtained is the message vector m. In this way, we have shown that if any orthogonal basis of $\hat{L}$ can be found, then the security is broken and we can find the message m. We can also break this cryptosystem with $\hat{L}$ itself as we have proved $\hat{L}$ to be an orthogonal basis.

**Other Ways to break the security** :-

Q d):-
**Approach 1** :- We can break the security of this cryptosystem by finding out the matrix R (i.e the private key). Finding out R is a difficult task.
Let us assume that we have a known message 'm' and its corresponding cipher i.e 'c'. Here, m is the n-bit long message m, it is a vector in $Z^n$ with binary entries. A random vector $v \epsilon Z^n$ was picked and the vector $c = v\hat{L} + m$ was computed.
We know a message 'm' and its corresponding cipher 'c'. We also know the public key $\hat{L}$
Now, we will try to find out the Matrix R.
We know that $\hat{L} = ULR$
Here, $U \epsilon Z^{nxn}$ be a unitary matrix, with det U = 1
$R \epsilon Qnxn$ be a rigid rotation matrix, that is, $RR^T = I$
L = n.I
I is identity matrix of dimensions nxn
U has integer entries and U is unitary matrix with det 1. So, we can say that U is an orthogonal matrix. U is of dimensions nxn. The number of possibilities for U is finite. 'U' can only contain entries from 0,1,-1 because the length of each row and column is 1 (because U is an orthogonal matrix). Also, The det of U should be 1.

The nxn matrices which satisfy the above criteria are finite in number. We can use this to our advantage and we can bruteforce all possibilities U and find out the correct R.
we know that $\hat{L} = U.L.R$
$\Rightarrow \hat{L} = U.n.I.R$
$\Rightarrow \hat{L} = n.U.R$
$\Rightarrow \hat{L}/n = U.R$
$\Rightarrow U^{-1}(\hat{L}/n) = R$
$\Rightarrow R = U^{-1}(\hat{L}/n)$
For each possible U with the above constraints, we will find R in the following way.Now, for each R, we will find out if this R is correct or not using 'm' and 'c'. We have our known $c \epsilon Qn$. We compute vector $d = cR^T$.
We Reduce every entry of d modulo n so that the entry becomes $< n/2$ in absolute value.
Let the resulting vector be $\hat{d}$.We now compute $m' = \hat{d}R$
If m' == m, we can say that we have found out the correct R matrix.
In this way, we can break the security of the crypto system using Bruteforce.

**Approach 2** :-

There is another approach to decrypt the message without figuring out 'R'. In this approach, we will take all $2^n$ possiblilities of messages( because each position takes either 0 or 1. The length of message is n. So total $2^n$ possibilities).we will try each of them and find out whether it is the correct message or not. Let c be the encrypted message. $\hat{L}$ be public key of dimensions nxn.

For every message m' in the $2^n$ possibilities, do the following:
1) $Calculate\ c - m'$
2) Now, calculate $(\hat{L})^{-1}$ .
3) Now, multiply $(\hat{L})^{-1}$ with (c - m'). That is $(c - m').(\hat{L})^{-1}$. let us call this v'.
$v' = (c - m').(\hat{L})^{-1}$.
4) Now, if all of the entries in the vector v' are integers, we can confidently say that we have found the correct message.That is m' is the correct decrypted message if if all of the entries in the vector v' are integers. We print this message m'.
In this way, we can decrypt the message without figuring out 'R'.

The python code for this approach is given below:

```
import numpy as np
from random import randint, uniform
#number of dimensions. replace it with your own dimension value. n>=1
n = 13
# replace this with your own invertible l^
l= np.array([[uniform(1,100) for i in range(n)] for j in range(n)])
#replace this with your own v
v=np.array([randint(1,100) for i in range(n)])
#this is the message. replace it with your own binary message
m=np.array([randint(0,1) for i in range(n)])
c = np.matmul(v,l) + m
inv_l= np.linalg.inv(l)

def decrypt_message(c,l,n):
    def check(z,mm):
        z=[round(i) for i in z]
        return (c==(np.matmul(z,l)+mm)).all()
    ans=[]
    for i in range(2**n):
        mm=np.array([int(j) for j in bin(i)[2:].zfill(n)])
        z=np.matmul((c - mm ),inv_l)
        if check(z,mm):
            ans.append(mm)
    return ans
ans=decrypt_message(c,l,n)

if len(ans)==1:
    print("Message has been decrypted and it is: ",ans[0])
    print((ans[0]==m).all())
    #verifying whether the message is correct or not
else:
    print("Failed to decrypt the message")
```

RESOURCES

https://kel.bz/post/lattices/

https://myweb.uiowa.edu/pbreheny/7110/wiki/l1-l2-inequality.html

http://www.noahsd.com/mini_lattices/05__babai.pdf

https://mathworld.wolfram.com/UnitaryMatrix.html

Code for Q d) Approach 2