Aluno: João Pedro Adami do Nascimento (19100531)

Learning With Errors

1. Explicação sobre o método

1.1 LWE

Inicialmente é gerada aleatoriamente uma matriz **A**, um vetor secreto **s**, e um vetor de erro **e**. A matriz **A** e o vetor secreto **s** são mod **q** (q será um parâmetro do esquema), e o erro é um vetor com entradas centradas em 0 (isto é, {1,2,-2,-1}).

A partir disto, calcula-se **B** da seguinte forma:
B = (A*s + e) mod q

(A,B) será a chave pública e **s** a chave privada.

Sem o vetor de erro **e**, teríamos o problema da seguinte forma:
(Note que o **x** da imagem abaixo corresponde ao nosso **s**.)

$$AX = B \Rightarrow \begin{cases} x_1 + 2x_2 + x_3 = 8 \\ -2x_1 + x_2 + 3x_3 = 9 \\ 4x_1 - x_2 - x_3 = -1 \end{cases}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ -2 & 1 & 3 \\ 4 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 9 \\ -1 \end{bmatrix}$$

Dado que isto caracteriza um sistema de equações lineares, temos diversos métodos para resolução deste problema, tal como eliminação gaussiana. Portanto um atacante conseguiria facilmente descobrir o vetor secreto **s**.

Disto conclui-se que o elo de segurança deste esquema é baseado no vetor de erro, que é o que torna este problema difícil.

Cifragem

A cifragem é realizada bit a bit.

Após a computação da chave pública, recolhe-se amostras **u** e **v** das chaves públicas **A** e **B**, respectivamente. Faz-se então o somatório destas amostras, sendo que no **v** alguns cálculos a mais são necessários:

$$u = \sum A_{samples}$$

$$v = \sum B_{samples} - \frac{q}{2}.M$$

q é um parâmetro do esquema criptográfico e **M** é o bit (pode assumir valor 0 ou 1).

(u,v) será o texto cifrado.

Decifragem

Para decifrar, será computado o seguinte: dec = v - s*u (mod q)
Feito isto, será feita a análise:

- se dec < q/2 então o bit é 0
- se dec >= q/2 então o bit é 1

1.2 RLWE

RLWE será usado para acordo de chaves.
Inicialmente os participantes do acordo de chaves negociam **n,** de tal forma que o maior expoente dos polinômios que serão gerados é n-1. Além disto teremos **q**, que é um parâmetro do esquema criptográfico.

A partir disto, obtém-se **A'** aleatoriamente. Logo após, divide-se **A'** por (x^n + 1) e atribui o resto da divisão para **A**.
De forma aleatória obtém-se o polinômio secreto **s** (com coeficientes mod q) e o polinômio de erro **e** (com coeficientes centrados em zero).

Novamente calcularemos **B** da seguinte forma: B = (A*s + e)
No entanto as operações de multiplicação e adição serão realizadas entre os coeficientes do polinômio, dentro de um corpo finito **Fq**. Os coeficientes dos polinômios serão menores que **q**.

Com isto, compartilha-se o **A** com o outro participante do acordo de chaves, que já possui seu próprio polinômio secreto **s** e vetor de erro **e** (obtidos aleatoriamente, mod **q**), para que este calcule sua chave **B** da mesma forma que o primeiro participante: B = (A*s + e)

Com o polinômio B computado, ambos participantes trocam entre si os seus **B's** e calculam o segredo compartilhado:

segredo_compartilhado = (B' * s) / (x^n + 1)

(**B'** corresponde ao polinômio **B** do outro participante)

2. Toy Example

Foi feita uma generalização do conceito de criptografia assimétrica com LWE, onde o usuário pode cifrar e decifrar mensagens passadas na linha de comando.

O código foi disponibilizada juntamente com o pdf.

2.1 Função Encrypt

Foi construída uma função de cifragem chamada *encrypt,* à qual recebe como parâmetros uma chave pública e uma mensagem a ser cifrada, sendo que esta mensagem é codificada em ASCII e convertida para binário. Feito isto, manipula-se seus bytes internamente para que a mensagem seja representada como uma lista de bytes.
[byte1, byte2, byte3, ...]

Onde byteN é constituído de 7 itens, que representam 7 bit: [0,0,0,0,0,0,0]

Sabe-se que 1 byte é constituído de 8 bits, no entanto como estamos trabalhando com caracteres ASCII, são necessários apenas 7 bits.

Esta função é constituida principalmente de dois loops, onde o primeiro percorre os bytes da mensagem, e o segundo os bits da mensagem. Isto é necessário pois este esquema criptográfico cifra 1 bit por vez.

Em cada iteração do loop realiza-se a amostragem das chaves públicas **A** e **B** e computa-se **u** e **v**.

Uma vez que **(u,v)** é usado para cifrar 1 bit, teremos que **u** e **v** serão uma lista de mesma estrutura que *message.*

```
for i in range(len(message)):
      for j in range(7):
          # Recolhendo amostras de valores aleatórios para usar como índice
          # Para cada bit da mensagem, devemos realizar a amostragem
          sample = random.sample(range(nvals-1), nvals//4)


          for x in range(0,len(sample)):
              u[i][j] = u[i][j] + A[sample[x]]
              v[i][j] = v[i][j] + B[sample[x]]
```

```
            v[i][j] = v[i][j] + math.floor(q/2) * int(message[i][j])
            v[i][j] = v[i][j] % q
            u[i][j] = u[i][j] % q

    # (u,v) é o nosso texto cifrado
    return u,v
```

2.2 Função Decrypt

Esta função recebe como parâmetros uma chave secreta **s** e um texto cifrado, que consiste de uma tupla com duas listas, u e v:

ciphertext = (u,v)

u = [u1, u2, u3...]
  u1 = [u11, u12, u13, u14, u15, u16, u17]

(o mesmo vale para o v)

Esta função também será constituída de dois loops, um para percorrer os 'bytes' de **u** e **v**, e outro para percorrer os 'bits' de **u** e **v**.

Em cada iteração realiza-se o cálculo de *dec*:
dec[i][j] = (v[i][j] - s*u[i][j]) % q

E julga-se o valor encontrado, para formar a variável de retorno *ret*:

```
for i in range(len(u)):
        for j in range(7):
            dec[i][j] = (v[i][j] - s*u[i][j]) % q
            if (dec[i][j] > q/2):
                ret[i][j] = '1'
            else:
                ret[i][j] = '0'
```

2.3 Função Main

Calcula-se **A**, **B**, **s** e **e** por meio do código disponibilizado pelo professor Bill Buchanan (tentei exaustivamente re-implementar o código utilizando a biblioteca numpy, mas sem sucesso, uma vez que a decifragem não era precisa):

```
A = random.sample(range(q), nvals)
```

```
for x in range(0,len(A)):
   e.append(random.randint(1,4))
   B.append((A[x]*s+e[x])%q)
```

Após isto, o programa recebe uma string passada pelo usuário, que será codificada em ASCII e convertida para binário.

```
text_bin = []
text = str(input("Digite a mensagem a ser cifrada: "))
text_bytes = text.encode('ascii')

for i in range(len(text_bytes)):
   a = bin(text_bytes[i])
   text_bin.append(a[2:])
```

E logo após, chama as funções de cifragem e decifragem.

```
ciphertext = encrypt(public_key, text_bin)
response = decrypt(s, ciphertext)
```

2.4 Output

O programa irá printar no terminal os vetores criptográficos, a mensagem cifrada e a mensagem decifrada em binário.

```
jpadn@jpadn:~/projects/ufsc/post-quantum/lwe_python/lwe$ python3 medium.py
Digite a mensagem a ser cifrada: joao
Chave pública A:
[21, 71, 38, 54, 6, 43, 47, 81, 29, 35, 96, 59, 23, 45, 67, 85, 51, 31, 42, 25]

Vetor de erros e:
[4, 2, 2, 1, 2, 3, 1, 4, 4, 2, 2, 2, 4, 1, 3, 3, 3, 4, 3, 3]

Chave secreta:
20

Chave pública B:
[36, 64, 83, 14, 25, 87, 68, 72, 2, 23, 79, 18, 76, 28, 82, 54, 53, 42, 67, 18]

Texto cifrado:
u: [[8, 76, 14, 8, 67, 87, 74], [15, 72, 27, 28, 41, 57, 76], [78, 80, 25, 50, 61, 26, 84], [6, 63, 40, 68, 14, 4, 0]]
v: [[27, 29, 2, 26, 93, 55, 41], [69, 45, 66, 36, 7, 37, 29], [66, 8, 25, 42, 68, 47, 90], [85, 63, 37, 64, 48, 45, 62]]

Texto decifrado (binário): 1101010110111111000011101111
```

3. Como e de que forma esse problema computacional pode ser usado em algoritmos de criptografia pós-quântica?

Os problemas da família LWE são usados para criptografia assimétrica, onde são utilizadas duas chaves públicas **A** e **B** (podem ser vetores, matrizes ou estruturas n-dimensionais) para cifrar 1 bit através de amostras **u** e **v** obtidas de **A** e **B**, respectivamente.

Já os problemas da família RLWE são usados para acordo de chaves, onde as chaves públicas **B** de cada participante são trocadas e submetidas à uma computação juntamente com a chave secreta **s** correspondente ao participante, para que assim obtenha-se um segredo compartilhado. Todos os participantes do acordo de chaves chegam ao mesmo segredo.