CSC 404 - Activity/Project 15 - Name: Chris G

Problem 1. Alice wants to use the McEliece Public Key Cryptosystem with a $[n, k]$ Hamming Error Correcting Code, C. To keep things 'small', let's use $[n, k] = [7, 4]$. i.e., $r = 3$.

bruh bruh bruh

a. KEY GEN!

i. Construct a $k \times k$ permutation matrix S and an $n \times n$ permutation matrix P.

$k=4$ $n=7$

confused... is this supposed to be by hand?

$$S = \begin{bmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \end{bmatrix} \qquad P = \begin{bmatrix} 0&0&0&0&0&1&0 \\ 1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1 \\ 0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0 \end{bmatrix}$$

7x4

$$G = \begin{bmatrix} 1&0&0&0&1&1&1 \\ 0&1&0&0&0&1&1 \\ 0&0&1&0&1&0&1 \\ 0&0&0&1&1&1&0 \end{bmatrix} \qquad SG = \begin{bmatrix} 0&1&0&0&0&1&1 \\ 0&0&1&0&1&0&1 \\ 0&0&0&1&1&1&0 \\ 1&0&0&0&1&1&1 \end{bmatrix}$$

ii. Compute and publish $G_1 = SGP$.

$$G_1 = SGP = \begin{bmatrix} 1&1&0&1&0&0&0 \\ 0&0&1&1&0&0&1 \\ 0&1&0&0&1&0&1 \\ 0&1&0&1&0&1&1 \end{bmatrix}$$

b. ENCRYPTION!

i. Let $m = 1001$ be Bob's k-bit message.

1 2 2 1 1

ii. Compute $c = mG_1$ and change one of the bits!

$c =$ 1000011

1001011

c. DECRYPTION!

i. Compute $c_1 = cP^T$ bruh what is P^T... I haven't shuffled anything yet...

$P^T =$ transverse of P above?

$$P^T = \begin{bmatrix} 0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&1&0&0&0 \\ 1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0 \end{bmatrix}$$

1100101

$C_1 = 1100101$

ii. Apply Error Correcting Code Decoder to c_1 to find codeword x_1 that is closest to c_1. Then, let x_0 be the first k bits of x_1.

$S = C_1 H^T$

$$H^T = \begin{bmatrix} 1&1&1 \\ 0&1&1 \\ 1&0&1 \\ 1&1&0 \\ 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix}$$

2 2 3 $= 001$

$x_1 = 1100100$

$x_0 = 1100$

iii. Compute $x_0 S^T$. Did this return m?

yes! pog

1001

Problem 2. Alice wants to use the McEliece Public Key Cryptosystem with a $[n, k]$ Hamming Error Correcting Code, C. Let's up the ante and use $[n, k]$ ... an record all of the values in Replit/Python – really, I just want you to play around w... c

a. KEY GEN!

i. Construct a $k \times k$ permutation ma...

ii. Construct an $n \times n$ permutation ...

iii. Compute and publish $G_1 = SGP$.

```
G1 = SGP =
[0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1]
[0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0]
```

```
P =
[1, 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, 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, 1]
[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, 1, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 1, 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, 1, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 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]
```

```
S =
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
[1, 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, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 1, 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, 1]
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
```

b. ENCRYPTION!

i. Let $m = 10100111001$ (i.e., $m = 1337$) be Bob's k-bit message.

ii. Compute $c = mG_1$ and chang...

```
m = [1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1]
c = [1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1]
```

c. DECRYPTION!

i. Compute $c_1 = cP^T$

```
c1 = [1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0]
```

ii. Apply Error Correcting Code Decoder to c_1 to find codeword x_1 that is closest to c_1. Then, let x_0 be the first k bits of x_1.

iii. Compute $x_0 S^T$. Did this return m?

```
x1 = [1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0]
x0 = [1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0]
m? = x0*St = [1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1]
Did it work? True
```

Problem 3 (Hamming Distance)**.** Given two k-bit strings, x and y, the Hamming Distance, denoted $h(x, y)$ gives the number of bits that differ. For example,

$$h(1101, 1001) = 1 \quad h(1101, 1000) = 2 \quad h(1101, 1110) = 2 \quad h(1101, 0011) = 3 \quad h(1101, 0010) = 4.$$

We can easily compute the Hamming Distance of two k-bit strings by simply adding up the result of 'xoring' each bit pairs. For example,

$$\begin{aligned}
h(1101, 1001) &= (1 \oplus 1) + (1 \oplus 0) + (0 \oplus 0) + (1 \oplus 1) = 0 + 1 + 0 + 0 = 1 \\
h(1101, 1000) &= (1 \oplus 1) + (1 \oplus 0) + (0 \oplus 0) + (1 \oplus 0) = 0 + 1 + 0 + 1 = 2 \\
h(1101, 1110) &= (1 \oplus 1) + (1 \oplus 1) + (0 \oplus 1) + (1 \oplus 0) = 0 + 0 + 1 + 1 = 2 \\
h(1101, 0011) &= (1 \oplus 0) + (1 \oplus 0) + (0 \oplus 1) + (1 \oplus 1) = 1 + 1 + 1 + 0 = 3 \\
h(1101, 0010) &= (1 \oplus 0) + (1 \oplus 0) + (0 \oplus 1) + (1 \oplus 0) = 1 + 1 + 1 + 1 = 4
\end{aligned}$$

a. Determine $h(1011101, 1001111)$ and $h(1011101, 1110100)$.

1001111 1110100
0010010 0101001
2 3

b. Let's play the evil doer, Eve! Suppose you (Eve) intercept the ciphertext $c = 1011101$ that was encrypted with the Public Key

$$G_1 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

shortcut...

(G_1 is the result of scrambling the generating matrix G for the [7, 4] Hamming Code.) Compute xG_1 for all possible 4-bit messages and record those that have a Hamming Distance of 1 to $c = 1011101$ – any x that does this is a possible contender for Bob's (secret) plaintext message, m. What are the possibilities for Bob's plaintext message?

10 111 01 1001101 1011100
00 111 01 1010101
11 111 01 1011001
1011111

15

hmm...

c. Let's play the evil doer, Eve, but Bigger! Suppose you (Eve) intercept the ciphertext $c = 110100101001111$ that was encrypted with the Public Key, G_1, from the [15, 11] Hamming Generating Matrix G (see Replit Link for G_1 – I was too lazy to copy and paste it :-))

Compute xG_1 for all possible 11-bit messages and record those that have a Hamming Distance of 1 to $c = 110100101001111$ – any x that does this is a possible contender for Bob's (secret) plaintext message, m. What are the possibilities for Bob's plaintext message?

1101 0010 0011 111 0010 0011 111
0101 ... 1010 ... 1011 011
1 1001 ... 2 ... 0110 ... 3 ... 0111 ... 4 ...101
1111 ... ... 0000 ... ... 0001 110
1100 ... ... 0011 ... ... 0010 ...

d. (Remark) Based off of parts b and c, the single bit change of the Hamming Error Correcting Codes are clearly not strong enough, but they give us a fun view into how ECCs can be incorporated into cryptosystems. In general, the structure remains the same – we just swap out the Hamming ECCs for something cooler. For (possible) post quantum security, the Classic McEliece round 3 submission makes use of so-called 'Goppa' Error Correcting Codes. For more information on the Post Quantum Standardization process and the 4 finalists see https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions (Also, obligatory plug for my Number Theory and Cryptography course and Cryptography and Codes course – much much more about these worlds)