

CSC 404 - ACTIVITY/PROJECT 15 - NAME:

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., $r = 3$.

a. KEY GEN!

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

- ii. Compute and publish $G_1 = SGP$.

b. ENCRYPTION!

- i. Let $m = 1001$ be Bob’s k -bit message.
- ii. Compute $c = mG_1$ and change one of the bits!

c. DECRYPTION!

- i. Compute $c_1 = cP^T$
- 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_0S^T . Did this return m ?

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 $[n, k] = [15, 11]$. i.e., $r = 4$. You can record all of the values in Replit/Python – really, I just want you to play around with the system and see how cool it is!

a. KEY GEN!

- i. Construct a $k \times k$ permutation matrix S .
- ii. Construct an $n \times n$ permutation matrix P .
- iii. Compute and publish $G_1 = SGP$.

b. ENCRYPTION!

- i. Let $m = 10100111001$ (i.e., $m = 1337$) be Bob’s k -bit message.
- ii. Compute $c = mG_1$ and change one of the bits!

c. DECRYPTION!

- i. Compute $c_1 = cP^T$
- 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_0S^T . Did this return m ?

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,

$$h(1101, 1001) = (1 \oplus 1) + (1 \oplus 0) + (0 \oplus 0) + (1 \oplus 1) = 0 + 1 + 0 + 0 = 1$$

$$h(1101, 1000) = (1 \oplus 1) + (1 \oplus 0) + (0 \oplus 0) + (1 \oplus 0) = 0 + 1 + 0 + 1 = 2$$

$$h(1101, 1110) = (1 \oplus 1) + (1 \oplus 1) + (0 \oplus 1) + (1 \oplus 0) = 0 + 0 + 1 + 1 = 2$$

$$h(1101, 0011) = (1 \oplus 0) + (1 \oplus 0) + (0 \oplus 1) + (1 \oplus 1) = 1 + 1 + 1 + 0 = 3$$

$$h(1101, 0010) = (1 \oplus 0) + (1 \oplus 0) + (0 \oplus 1) + (1 \oplus 0) = 1 + 1 + 1 + 1 = 4$$

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

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

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

(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?

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?

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)