Exercise sheet 3, 13 April 2023

1. The binary Hamming code $\mathcal{H}_4(2)$ has parity check matrix

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

   and parameters $[n, k, d] = [15, 11, 3]$.
   Correct the word $(0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1)$.

2. This exercise is about attacks on code-based cryptography. Let G be the generator matrix of an $[n, k, d]$ code with $d = 2t + 1$. In the basic schoolbook-version of McEliece encryption, a message $m \in \mathbb{F}_2^k$ is encrypted by computing $y = mG + e$, where $e \in \mathbb{F}_2^n$ is randomly chosen of weight t.

   Alice and Bob use this method to send m but Eve intercepts $y_1 = mG + e_1$ and stops the transmission. After a while, Alice resends an encryption of m, using a different error vector e_2, so $y_2 = mG + e_2$, where both e_i have weight t.

   (a) Compute the average weight of $e_1 + e_2$, where $+$ denotes addition in $\mathbb{F}_2^n$, and the average weight of $e_1 \cdot e_2$, where $\cdot$ denotes componentwise multiplication in $\mathbb{F}_2^n$.

   (b) Show how Eve can recover the message m.
   Hint 1: Eve's task should be stated as a decoding problem of a code of length less than n.
   Hint 2: First solve the problem assuming that e_1 and e_2 have no overlap in their non-zero positions.
   Hint 3: Figure out how to retrieve m from y_1 if you know $k(1+\epsilon)$ positions that are error free, for some positive ϵ.

3. Let K be the public parity-check matrix of a code of length n, dimension k, and minimum distance $d = 2t + 1$. The school-book version of the Niederreiter system encrypts a message $m \in \mathbb{F}_2^n$ of Hamming weight t by computing the syndrome $s = K \cdot m$.

   You are given access to a decryption oracle. In the following two situations, show how to recover m and compute how many calls to the oracle are required.

(a) The oracle decrypts any ciphertext $s' \neq s$ provided that $s' = K \cdot m'$ with m' of Hamming weight less than or equal to t.

(b) The oracle decrypts any ciphertext $s' \neq s$ provided that $s' = K \cdot m'$ with m' of Hamming weight exactly equal to t.

4. RaCoSS is a signature system submitted to NIST's post-quantum competition. The system is specified via two parameters n and $k < n$ and the general system setup publishes an $(n-k) \times n$ matrix H over $\mathbb{F}_2$.

   Alice picks an $n \times n$ matrix over $\mathbb{F}_2$ in which most entries are zero. This matrix S is her secret key. Her public key is $T = H \cdot S$.

   RaCoSS uses a special hash function h which maps to very sparse strings of length n, where very sparse means just 3 non-zero entries for the suggested parameters of $n = 2400$ and $k = 2060$. You may assume that h reaches all possible bitstrings with exactly 3 entries and that they are attained roughly equally often.

   To sign a message m, Alice first picks a vector $y \in \mathbb{F}_2^n$ which has most of its values equal to zero. Then she computes $v = Hy$. She uses the special hash function to hash v and m to a very sparse $c \in \mathbb{F}_2^n$. Finally she computes $z = Sc + y$ and outputs (z, c) as signature on m.

   To verify (z, c) on m under public key T, Bob does the following. He checks that z does not have too many nonzero entries. The threshold here is chosen so that properly computed $z = Sc + y$ pass this test. For numerical values see below. Then Bob computes $v_1 = Hz, v_2 = Tc$ and puts $v' = v_1 + v_2$. He accepts the signature if the hash of v' and m produces the c in the signature.

   (a) Verify that $v' = v$, i.e. that properly formed signatures pass verification. As above, you should assume that the other test on z succeeds.
   Note: All computations take place over $\mathbb{F}_2$.

   (b) The concrete parameters in the NIST submission specify that $n = 2400$, and that the output of h has exactly 3 entries equal to 1 and the remaining 2397 entries equal to 0.

   Compute the size of the image of h, i.e., the number of bitstrings of length n that can be reached by h.

(c) Based on your result under b) compute the costs of finding collisions and the costs of finding a second preimage.

(d) For the proposed parameters the threshold for the number of nonzero entries in z is larger than 1000.

Break the scheme without using any properties of the hash function, i.e. find a way to compute a valid signature (z, c) for any message m and public key T. You have access to the matrix H and can call h.

Hint: You can construct a vector z of weight no larger than $n-k$ that passes all the tests.