Linear Algebra

Amisha Bansal

November, 2020

1 Introduction

Medical images play a major role in precise and detailed diagnoses because doctors rely on them to figure out patients' illnesses and devise treatment programs. It is required that there is confidentiality of patients' medical records. Medical images are only made available to the doctors involving in the treatment of a patient. Image confidentiality is often achieved by employing cryptographic encryption and data authentication mechanisms which require a large amount of computing power. It is common and convenient for doctors to use mobile devices to access patients' medical information . Therefore, there is a pressing demand for a faster secure mechanism that can ensure confidentiality of medical images and is suitable for mobile devices. One characteristic of medical images is their high level of accuracy and detail. If a medical image is made blurred and loses its details, it will be useless. In this scheme, shadow images are created from medical images and stores them in different databases. The original images are only retrievable if all the shadows are collected. Individually, a shadow would not visually reveal any information regarding a patient's condition. This would satisfy the requirement for medical images. The proposed scheme is based on Hamming code, which has been used extensively in image processing . Its encoding and decoding processes are very efficient, thus, it can help our scheme achieve low computational costs and guarantee a fast response when deployed on mobile devices.

1.1 Hamming Codes

Detecting transmission errors is highly significant in data communication. The simplest technique in detecting an error is to append a parity bit to each byte of the transmitted data. This mechanism detects the occurrence of error correctly if the number of error bits is an odd number. However, it cannot pinpoint the error bit by a byte.

Hamming code is superior to parity check because it can detect and identify the location of a single-bit error in the transmitted data [36]. Hamming codes are available in different sizes. If the number of parity bits is m (3), then the code length is n = 2 m 1. Hamming code has a minimum Hamming distance 3, and thus it can correct one single error. For example, for m = 3, the code length is 7 bits in which there are three parity bits and four data bits; this code is called (7,4) Hamming code.Similarly, (15,11) Hamming code has four parity bits and eleven data bits. The number of data bits increases exponentially as the number of parity bits grows.

The parity-check matrix H of Hamming code consists of all non-zero m-tuple 2 m as its columns (i.e., $2^m - 1$ columns). Suppose that the parity-check matrix is H = [I m Q], where I m is an m × m identity matrix and Q is an m-tuple with weight two or more. Then the generator matrix is $G = [Q^T I_{n-m}]$, where $G \times H^T = [0]_{m\times(n-m)}$. Hamming code. The following two examples show how to encode and decode for these two Hamming codes, respectively for which we use (7,4) Hamming code and (15,11) Hamming code.

Example 1. For (7,4) Hamming code, H is a 3 × 7 matrix and G is a 4 × 7 matrix, as shown below.

Obviously, one can easily verify $G \times H^T = [0]_{4\times 3}$

$$H = [I_3 Q] = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$G = \left[Q^T I_4\right] = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The above code is the so-called systematic code, i.e., the 4 data bits ($\mathrm{d}_1, \mathrm{d}_2, \mathrm{d}_3$ and d_4) are at the most right 4 bits, and others are parity bits. To encode a 4 -bit data $d = [d_1 d_2 d_3 d_4]$ into a 7 -bit codeword c, the following equation is applied:

$$c = d \times G = [p_1 p_2 p_3 d_1 d_2 d_3 d_4]$$

From Equation (1), we have $p_1 = d_1 \oplus d_3 \oplus d_4, p_2 = d_1 \oplus d_3 \oplus d_4$, and $p_3 = d2 \oplus d_3 \oplus d_4$. It is observed that there are at least 2 parity bits covering a data bit, so even a parity bit is flipped we still can detect and correct it.

For example, suppose that the data is $d = [0101]$, then $p1 = 1, p2 = 1$ and $p3 = 0$; the codeword is $c = [1100101]$. For decoding, the syndrome, a 3 -tuple, is computed as $s = c^{'} \times H^T = [s_3 s_2 s_1]$.

$$s = c' \times H^T = [s_3 s_2 s_1]$$

$s = (s_3, s_2, s_1) = c' \times H^T$. If the syndrome is a zero vector [000], the codeword is correct; otherwise, the syndrome $(s_3 s_2 s_1)$ can be used for correcting one error. For the received word c' = characteristic [1110101], the syndrome vector [s3 s2 s1] equals to [001]. The 3-tuple (001) is the third row in HT and thus the bit p_3 (the position is 3 from left) should be flipped. The correct codeword is [1100101].

Example 2 . The (15,11) Hamming code is constructed in the same manner as the (7,4) Hamming code. The systematic forms of 4×15 parity-check matrix $\boldsymbol{H}$ and 11×15 generator matrix $\boldsymbol{G}$ are sliown beloto.

$$H = \mid l_4 Q] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$G = \mid Q^T I_1 1\big] = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

An 11-bit data $d = [d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10} d_{11}]$ can be encoded into a systematic 15 -bit codeword $c = d * G = [p_1 p_2 p_3 p_4 d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10} d_{11}]$, where the parity bits are computed as follows:

$$\begin{aligned} p_1 &= d_1 \oplus d_2 \oplus d_4 \oplus d_7 \oplus d_8 \oplus d_9 \oplus d_{11} \\ p_2 &= d_1 \oplus d_3 \oplus d_5 \oplus d_7 \oplus d_8 \oplus d_{10} \oplus d_{11} \\ p_3 &= d_2 \oplus d_3 \oplus d_6 \oplus d_7 \oplus d_9 \oplus d_{10} \oplus d_{11} \\ and p_4 &= d_4 \oplus d_5 \oplus d_6 \oplus d_8 \oplus d_9 \oplus d_{10} \oplus d_{11} \end{aligned}$$

For an 11 -bit data $d = [011010111011$, the parity bits are $p_1 = 1, p_2 = 1, p_3 = 1$ and $p_4 = 0$. Thus, the codeword is $c = [111001101011101]$. Suppose that the received word is $c' = [111001111011101]$. The

syndrome of $c' : s = (s4, s3, s2, s1)$ is (1001). Then, we can locate the error bit which is at the eighth row in the matrix H^T (the error position is 8 from the left).

1.2 Scheme:

Suppose that there are two medical images (I_1 and I_2) for the same patient. The proposed scheme produces two shadows (S_1 and S_2) using the shadows generating algorithm. These two shadows have very little resemblance with the original images. If a malicious entity could, however, acquire one and only one shadow, he/she could not perceive any meaningful information about the patient's condition from it. For diagnosis purpose, having a shadow is as good as none.

When an authorized doctor would like to access this patient's medical images, he/she must obtain both shadows S_1 and S_2. The original image reconstruction algorithm is then used to regenerate the original medical images for diagnosis. Because two shadows can be used to reproduce the two original medical images, we have to put them on two different databases. This is a security measure ensuring that any security breach happened to one server will not compromise the whole system. Without the shadow stored on the other server, the patients' health information does not leak out after the incidence. Using (7,4) Hamming Code This shadows generation algorithm will produce two shadows (S_1 and S_2) from two medical images I_1 and I_2. It manipulates the images at pixel level, therefore, one pixel from each original image will be picked sequentially for creating a new pair of pixels of the two shadows. Suppose that pixel $P_1 = [i_{1,1}i_{1,2}i_{1,3}i_{1,4}i_{1,5}i_{1,6}i_{1,7}i_{1,8}]$ from I_1 and $P_2 = [i_{2,1}, i_{2,2}i_{2,3}i_{2,4}i_{2,5}i_{2,6}i_{2,7}, i_{2,8}]$ from I_2 are selected as shown in Figure 1, where $i_{1,1}, i_{1,2}, \ldots, i_{1,8}$ and $i_{2,1}, i_{2,2}, \ldots, i_{2,8}$ are the bits in those two pixels. Two corresponding pixels P_1' and P_2' on shadows S_1 and S_2 are produced as follows:

Algorithm: Shadow generation algorithm
Input: P_1 and P_2
Output: $\mathrm{P}_1' and P_2'$
*Step*1.Assign $[\mathrm{d_1d_2d_3d_4}] = [i_{1,1}i_{1,2}i_{1,3}i_{1,4}]$, and $[x_1x_2x_3] = [i_{2,1}i_{2,2}i_{2,3}]$.
Step 2. Use (7, 4) Hamming code to compute $c = d * G = [p_1p_2p_3d_1d_2d_3d_4]$.
Step 3 . For the codeword c, flip one bit at the position $(x_3x_2x_1)_2$, where x_1 is the least significant bit, to output a new codeword $c' = [p_1'p_2'p_3'd_1'd_2'd_33'd_4']$

round P_1 was selected from I_1, then it will be obtained from I_2 in the next round to make sure that the shadows are very much different from the original images. Example 3. Suppose that two pixels selected from two medical images are $P_1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$ and $P_2 = [\ 1 \quad 0 \quad 1 \quad 00101]$. To generate two shadow pixels P_1' and P_2', the following steps are executed: Step 1. Determine d and x from P_1 and $P_2, d = [0110]$ and $x = [x_1x_2x_3] = [101]$ Step 2. Encode d using (7,4) Hamming code, we get

$$c = d * G = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

can be constructed. We can see that the first four bits in P_1 have been changed from 0110 to 0010 in P_1'. The pixel P_2' has also changed since the first three bits have been changed from (101) to (110). As we mentioned earlier, P_1 and P_2 are selected alternately from the medical images I_1 and I_2 to ensure that changes are spread evenly between the two shadows images.

The above algorithm demonstrates how to create two shadow images from two medical images. The original image reconstruction algorithm shows how to reproduce the original images from the two shadows.

Suppose that pixels P_1' and P_2' are selected from shadows S_1 and S_2 in turn. The following ster are performed to reconstruct the original images.

Algorithm: Original images reconstruction algorithm
Input: P_1' and P_2'
Output: P_1 and P_2
Step1. Assign $\left[d_1^{'}d_2^{'}d_3^{'}d_4^{'}\right] = [o_{1,1}o_{1,2}o_{1,3}o_{1,4}]$ from first four bits of $P^{'}{}_1$ and $\left[p_1^{'}p_2^{'}p_3^{'}\right] = [o_{2,1}o_{2,2}o_{2,3}]$ from first three bits of P_2'
Step2. Construct the codeword $c' = [p_1'p_2'p_3'd_1'd_2'd_3'd_4']$.
Step3.Compute the syndrome $s = [s_3s_2s_1] = c' * H^{\mathrm{T}}$.
Step4. If s is not equal to [000] , go to Step 5; or else, stop the algorithm and return $P_1 = P_1'$ and $P_2 = P_2'$
Step5. Find the position of $(s_3s_2s_1)$ in H^{T}, i.e., $(x_3'3x_2'2x_1')_2$ and then obtain the correct codeword $[p_1p_2p_3d_1d_2d_3d_4]$
Step 6. Output the pixels for the original medical images as shown in Figure 2 . The pixels P_1 and P_2 are reconstructed as follows:

$$P_1 = [i_{1,1}i_{1,2}i_{1,3}i_{1,4}i_{1,5}i_{1,6}i_{1,7}i_{1,8}] = [d_1d_2d_3d_40_{1,5}o_{1,6}0_{1,7}o_{1,8}]$$

$$and\ P_2 = [i_{2,1}i_{2,2}i_{2,3}i_{2,4}i_{2,5}i_{2,6}i_{2,7}i_{2,8}] = [x'{}_1x_2'x'{}_3\sigma_{2,5}o_{2,6}o_{2,7}o_{2,8}]$$

Example 4. In Example 3, it outputs two shadow pixels P' 1 = [0 0 1 0 1 1 0 1] and P' 2 = [1 1 0 0 0 1 0 1]. Now we apply the original image reconstruction algorithm to get back the two original pixels P 1 and P 2 as follows.
Step 1. Determine $d' = [0010]$ and $p' = [110]$
Step 2 . The codeword is $c' = [1100010]$.
Step 3. Compute the syndrome of c', we have

$$s = [s_3s_2s_1] = c' * H^T = [1\ 1\ 0\ 0\ 0\ 1\ 0\ ] * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$$

Step 4. since $s = [001]$ is not equal to [000], go to Step 5.
Step 5. Flip the bit located at the position (101) in H^{T} (the fifth position from the left in c', i.e. $(x_3'x_2'x_1') = (101)$, and we have $c = [1\ 1\ 0\ 0\ 1\ 1\ 0]$
Step 6. Obtain the original pixels $P_1 = [0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ ]$ and $P_2 = [1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ ]$.
It shows that the recovered pixels are correct; thus, the algorithms function correctly as expected. So far, we have only demonstrated how to generate shadows and reconstruct original images, but we have not mentioned exactly which bits are selected to go through the processes. In a pixel, there are three groups of bits that we can select from (a) the least significant bits, (b) the most significant bits, and (c) those bits in the middle.The selection of the group of bits for this algorithms affects the outcomes significantly.