CHAPTER 4

ENCRYPTED MEDICAL IMAGE COMPRESSION USING PIXEL BLOCK SHORT ALGORITHM

4.1 INTRODUCTION

The aim of the proposed system is to compress encrypted medical images in an efficient manner by proposing efficient compression schemes. In order to reduce the encrypted image size, many algorithms have been proposed for the compression of images because during TVCE process the encrypted image size is increased compared to the original image size. However, these compression techniques have limitations in compressing medical images. These limitations are:

1. Loss during the compression
2. Less compression ratio
3. High compression time
4. High Compression size as compared to the original image
5. Quality is not up to the satisfaction

This chapter describes how the encrypted medical image is compressed by one of the three proposed compression techniques. One of the compression techniques is Pixel Block Short Algorithm (PBSA), the PBSA can be divided into two levels of processing as follows:
1. Compressing the encrypted medical image using PBSA Encoder.

2. Decompressing the compressed encrypted medical image using PBSA Decoder.

These levels are discussed in a detailed manner in Sections 4.2 and 4.3. In chapter 3 the grayscale medical image encryption and decryption process are described. Figure 4.1 shows, how the original grayscale medical image is encrypted by the proposed TVC Process. The encrypted medical image is compressed by the Pixel Block Short Encoding (PBSE) process and this image is decompressed by the Pixel Block Short Decoding (PBSD) process.

**Figure 4.1 Overview of the PBSA**
4.2 COMPRESSION OF ENCRYPTED MEDICAL IMAGES

In Figure 4.1, the original grayscale image is encrypted by the proposed TVCE technique. The encrypted medical image is the input for the compression process. The encrypted secret image $E_{i2}(;)$ from Equation 3.23 is compressed by the PBSE. This encoding process is classified into different stages which are:

Step 1  The original encrypted medical image is accepted as an input image.

Step 2  The image is divided into four identical blocks, similar to 2x2 matrix.

Step 3  The minimum occurrence pixel(s) are found out from every block and the positions of the minimum occurrence pixel(s) are found using the Verdict Occurrence Process.

Step 4  The pixel positions are shortened with the help of a shortening process.

Step 5  The features (Symbols and shortened pixel positions) are extracted from each block. The extracted features are stored in a particular place, and the values of these features put together are called the compressed medical image.

The entire system diagram for PBSE process is shown in Figure 4.2.
4.2.1 Getting the Input Image for PBSE Process

The encrypted grayscale medical image $E_{12}();$ from Equation 3.23 is the input binary image for the compression process. This input image is used for the divided process as shown in Figure 4.2.

4.2.2 Dividing the Input Image into Blocks

The input binary encrypted medical image $E_{12}();$ is divided into four equal blocks similar to a 2x2 matrix. Finally, the splitted blocks are named as $B_1$, $B_2$, $B_3$, $B_4$ in Equation 4.1.

$$E_{12}(); = B_1 \oplus B_2 \oplus B_3 \oplus B_4 \quad (4.1)$$

4.2.3 Verdict Occurrence Process

In this process, each of the blocks $B_1$, $B_2$, $B_3$ and $B_4$ are taken in a separate manner. From each block, the minimum occurrence pixel(s) and its positions are also calculated.

For instance, if the minimum occurrence pixel value is ‘0’, the position of ‘0’ is found from the particular block and it is stored in a place. The same process is applied within every block, one symbol (the minimum occurrence pixel) and its corresponding position is found out. From the Equations 4.2 to 4.5 are used for calculating the minimum occurrence of the pixel from each and every block. In case, the occurrence is equal, the minimum occurrence pixel value is automatically considered as ‘1’.
Figure 4.2 Pixel Block Short Encoding Process
\[ F_{\text{Min}}(B_1) = S_1 \]  
\[ F_{\text{Min}}(B_2) = S_2 \]  
\[ F_{\text{Min}}(B_3) = S_3 \]  
\[ F_{\text{Min}}(B_4) = S_4 \]  

From the Equations 4.6 to 4.9, the minimum occurrence of the pixel position is calculated by using \( F_{\text{pos}} \), where \( F \) denotes the found block and \( F_{\text{pos}} \) denotes the position of the particular block based on \( S \). \( S \) is the minimum occurrence of the pixel value in a particular block, it is also called symbols. \( S_1 \) is the first block minimum occurrence pixel value or symbol. \( P_1, P_2, P_3 \) and \( P_4 \) are the minimum occurrence pixel positions.

\[ F_{\text{pos}}(B_1(S_1)) = P_1 \]  
\[ F_{\text{pos}}(B_2(S_2)) = P_2 \]  
\[ F_{\text{pos}}(B_3(S_3)) = P_3 \]  
\[ F_{\text{pos}}(B_4(S_4)) = P_4 \]  

### 4.2.4 Petite Process for Finding the Minimum Occurrence Pixel Position

The Petite does the vital role of compressing the encrypted medical image. In this process the minimum occurrence of the pixel position is compressed based on the following steps:
Step 1 The beginning of the pixel position is considered as the starting point and the difference between first and second position is calculated. If the difference is ‘1’ the same process is done between the next two positions, till the difference is not equal to ‘1’.

Step 2 Suppose, the process is stopped because the difference is not equal to ‘1’, take the starting position and the end position whose difference is ‘1’. In between start and end position allocate the symbol ‘.’. This symbol ‘.’ denotes continuous occurrence.

Step 3 If the start and end positions are in between any symbols, the end position will become the start position.

Step 4 If the difference between the two positions is ‘2’, the same process is done till the difference is not equal to ‘2’, as in step-1. If the process is stopped due to the difference being not equal to ‘2’, the symbol ‘-’ is given in between the start and end positions. This symbol ‘-’ denotes the alternate occurrence.

Step 5 If the difference is not equal to ‘1’ or ‘2’, the position is taken directly.

\[ P_1 = S_{(p_1)} \quad (4.10) \]

\[ P_2 = S_{(p_2)} \quad (4.11) \]

\[ P_3 = S_{(p_3)} \quad (4.12) \]

\[ P_4 = S_{(p_4)} \quad (4.13) \]
From the Equation 4.10 to 4.13 the block positions \((P_1, P_2, P_3, P_4)\) are shortened by \(S_0\).

The example below shows the shortening process.

\[
\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
33 & 34 & 35 & 36 & 37 & 40 & 42 & 44 & 46 & 50 & 55 \\
\end{array}
\]

The above values are considered as the minimum occurrence pixel position. Based on step 1, the difference between first and second values is calculated. The difference ranges are from ‘1’ to ‘11’. So, in this process the starting value is 1 and the end value is 11. After 11, the difference is not equal to ‘1’ so the process will stop.

\[
\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
33 & 34 & 35 & 36 & 37 & 40 & 42 & 44 & 46 & 50 & 55 \\
\end{array}
\]

\[
\begin{array}{cccccccccccc}
29 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 40 & 42 & 44 \\
46 & 50 & 55 \\
\end{array}
\]

In between the start and the end, the ‘.’ symbol is allotted if the difference is ‘1’. Similarly, all near positions are compared by the process. From the comparison, it can be seen that from position 31 to 37, the difference is ‘1’. The difference of the remaining positions’ is not equal to ‘1’. In this calculation, the starting value is 31 and end value
is 37. In between 31 and 37, the continuous occurrence symbol is allotted. This can be seen in the third line.

```
1   .  11 13 15 17 19 21 23 25 27
29  31 32 33 34 35 36 37 40 42 44
46  50  55
```

Similarly, step 3 is considered for the next process. In the above line, the starting position is 1 and the next position is ‘.’. If the start and end positions are in between any symbol, the end position becomes the start position as mentioned in step 3. In step 3, the ending position 11 becomes the starting position. Since 11 onwards, the position difference is to be calculated. In the calculation from 11 to 31 the difference is ‘2’, so step 4 is done here. From step 4, if the difference is ‘2’ in between two positions, the one after another occurrence symbol ‘-’ is allotted as shown in the next example.

```
1   .  11 13 15 17 19 21 23 25 27
29  31   . 37 40 42 44 46 50  55
```

The same process is done in a continuous manner, between 11 and 31. Subsequently, based on step 3, the end position becomes the starting position. This means that the end position 31 becomes the starting position. From 31 to 37, one continuous occurrence symbol occurs and here also the ending position 37 becomes the starting position. The difference in between 37 to 40 is not equal to ‘1’ or ‘2’.
The next position if 40 starting position, from 40 to 46 the difference is 2. The shortening process is shown below

\[ 1 \quad 11 \quad - \quad 31 \quad . \quad 37 \quad 40 \quad - \quad 46 \quad 50 \quad 55 \]

In the next positions 46, 50 and 55 the difference is not equal to ‘1’ or ‘2’, so step 5 is considered.

\[ 1 \quad . \quad 11 \quad - \quad 31 \quad . \quad 37 \quad 40 \quad - \quad 46 \quad 50 \quad 55 \]

\[ \begin{array}{ccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 13 & 15 \\
17 & 19 & 21 & 23 & 25 & 27 & 29 & 31 & 32 & 33 & 34 & 35 & 36 \\
37 & 40 & 42 & 44 & 46 & 50 & 55 \\
\end{array} \]

**Figure 4.3 Minimum occurrence pixel positions**

In the Figure 4.3, 33 positions are shortened into 8 positions and three symbols in Figure 4.4. Equally position shortening is done in every block in an individual manner.

In the proposed compression PSBE technique, the size of encrypted image can be reduced in an efficient manner. These experimental results have been discussed in Section 4.4.

**Figure 4.4 Result of the petite process from minimum occurrences pixel position**
In every block, the symbols and their shortened positions are provided. The first symbol and its position are taken one by one, and the block information is added in a particular table similar to RLE (Equation 4.14). CMI is the compressed medical image.

\[
\text{CMI} = S_1 \oplus S_{(p_1)} \oplus S_2 \oplus S_{(p_2)} \oplus S_3 \oplus S_{(p_3)} \oplus S_4 \oplus S_{(p_4)}
\]  

(4.14)

4.3 DECOMPRESSION PROCESS FOR COMPRESSED ENCRYPTED MEDICAL IMAGE

The next process of PBSE is Pixel Block Short Decoding (PBSD) process. In the decoding process, there are eight steps involved while decompressing the compressed encrypted medical image (Figure 4.5). The steps are as follows:

**Step 1** The feature extraction value of compressed information is found out.

**Step 2** From the feature extraction, the symbols are split and the positions are shortened in a separate manner.

**Step 3** The position is retrieved from the rescheduled process.

**Step 4** The symbols and reconstructed positions of the minimum occurrence pixels are taken block wise.

**Step 5** Every symbol is placed based on the position in each block.

**Step 6** If the minimum occurrence pixel is ‘0’, rest of the places are automatically allocated ‘1’ or if the
minimum occurrence pixel is ‘1’ the remaining place is automatically allocated ‘0’.

**Step 7** Both the blocks are merged as per order 2x2.

**Step 8** The final output is the reconstructed encrypted medical image.

### 4.3.1 Input Feature Extraction Process

In this process, as per the instructions or steps which are given for decompressing the compressed encrypted medical image, the features are extracted in a separate manner. Initially, the entire block’s information is split block-wise. CMI Information is given in Equation 4.14. This feature information is split block-wise as in Equation 4.15, 4.16, 4.17 and 4.18. In Equation 4.14, the following symbols and their positions are split:

\[
S_1 \oplus S_{(R_1)} = B_1 \quad (4.15)
\]

\[
S_2 \oplus S_{(R_2)} = B_2 \quad (4.16)
\]

\[
S_3 \oplus S_{(R_3)} = B_3 \quad (4.17)
\]

\[
S_4 \oplus S_{(R_4)} = B_4 \quad (4.18)
\]
4.3.2 Petite Position Rearranging Process

The splitted symbols and their shortened positions are taken as input for this process. In the rearrangement process, two types of position symbols occur. These symbols are ‘.’ for continuous occurrences (Ex. 1 2 3 4 5…) and ‘-’ for ‘alternate’ occurrences (Ex. 1 3 5 7 9…). In between the position symbols, there is a starting position and
an ending position. The process will start from the starting position and if any position symbols are available before the end position, based on the pixel description, the position will be replaced. The example of rearrangement process steps as given below.

i. 1 . 11 - 31 . 37 40 . - 46 50 55

In step i, some position symbols are taken in between the starting and ending positions.

ii. 1 2 3 4 5 6 7 8 9 10
11 - 31 . 37 40 - 46 50 55

Step ii involves taking the symbols in between 1 and 11. It means that the starting position is 1 and ending position is 11. Thus the position will appear in a continuous manner between 1 and 11.

In the same step, the end position 11 becomes the starting position of another process. In this case also, the end position for other process is 31. So, in between 11 and 31, there occurs one ’alternate’ symbol. It denotes alternate occurrences.

iii. 1 2 3 4 5 6 7 8 9 10
11 13 15 17 19 21 23 25 27 29
31 . 37 40 - 46 50 55

In the step iii, ‘alternate’ occurrence, occurs from 11 to 31. Once this process is completed, the end position 31 becomes the starting position of another process. There is a symbol which defines continuous occurrence after 31 and before 37.
In the above step iv, the values occur in a continuous manner from starting position 31 to ending position 37. After the end position 37, the difference between the two positions 37 and 40 is not equal to ‘1’ or ‘2’. So the value 40 is taken directly and then 40 is taken as the starting position of another process. In between 40 and 46, there is a symbol which denotes alternate substitution. From 40 onwards, the value occurs alternatively in step v. Finally, the ending position 46 becomes the starting position of another process. In this process, in between 46, 50, and 55, the difference is not equal to ‘1’ or ‘2’, so those values appear directly. Similar processes are applied in each block $R_{s(P_1)}$, $R_{s(P_2)}$, $R_{s(P_3)}$, $R_{s(P_4)}$. R is the rearrangement process of $S_{(P)}$ from Figure 4.6. The example of overview of the position after rearranging is given in a detailed manner below.

\[
\begin{array}{cccccccccccc}
iv. & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
   & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 40 & - & 46 \\
   & 50 & 55 \\
\end{array}
\]

\[
\begin{array}{cccccccccccc}
v. & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
   & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 40 & 42 & 44 \\
   & 46 & 50 & 55 \\
\end{array}
\]

4.3.3 Block Process for Finding the Symbol and Reconstructing the Positions

Every block is considered in a separate manner. For example, in a particular block the minimum occurrence pixel is replaced based on
the position which is reconstructed for the corresponding blocks (Equation 4.19, 4.20, 4.21 and 4.22).

\[
\begin{align*}
S_1 \oplus R_{S(p_1)} &= B_1 \\
S_2 \oplus R_{S(p_2)} &= B_2 \\
S_3 \oplus R_{S(p_3)} &= B_3 \\
S_4 \oplus R_{S(p_4)} &= B_4
\end{align*}
\]

\[B_1 \oplus B_2 \oplus B_3 \oplus B_4 = \text{EMI} \]  
\[(4.23)\]

**Figure 4.6 Pixel Rearrangement Process**

The opponent pixel value is placed automatically. It means that, if the minimum occurrence is ‘0’, ‘1’ is placed in the rest of the positions or if the minimum occurrence is ‘1’, ‘0’ is automatically placed
in the rest of the positions. Likewise, each block is regenerated from this block process. Once the block process for each block is completed, the blocks are taken as per the sequence order \( B_1, B_2, B_3 \) and \( B_4 \). These blocks are merged as a 2x2 matrix in Equation 4.23. The final output of the Equation 4.23 is the reconstructed Encrypted Medical Image (EMI).

### 4.4 RESULT ANALYSIS AND DISCUSSION

In this Section 4.4, the combination of the PBSA and TVCS is considered. It means the original grayscale medical image is encrypted by the proposed TVCE system which is discussed in chapter 3, from the encryption process output is compressed by the PBSE Algorithm. Once the compression is completed the compressed encrypted medical image is decompressed by PBSD Algorithm, and this output is decrypted by proposed TVCD system. This Section discusses the proposed TVCS, PBSA combinations results and performance alone based on the limitation of the secure compression, to measure the algorithm efficiency.

Based on the limitations several of the parameters are measured for check algorithm efficiency. The parameters are as follows:

1. To check if there is any loss during the process, MSE and CC are calculated by using standard formula. When the CC values are near to one, exact replica image is reconstructed. Based on this point the integrity is also justified. In the same way, in finding the error of reconstructed image, MSE is doing vital role.

2. Compression Ratio (CR)
3. Execution Time
4. Size
5. PSNR

![Image 1 (CAG)](image1.png)

Before encryption the size is 127.30kb
After decryption the size is 127.34kb

![Image 2 (CT)](image2.png)

Before encryption the size is 234.12 kb
After decryption the size is 234.12 kb

**Figure 4.7** Input images 1 & 2 size before and after process
Before encryption the size is 102.8 kb  
After decryption the size is 102.9 kb

Before encryption the size is 158.4 kb  
After decryption the size is 158.4 kb

Figure 4.8  Input images 3& 4 size before and after process
Before encryption the size is 96.26 kb
After decryption the size is 96.26 kb

Before encryption the size is 239.3 kb
After decryption the size is 239.3 kb

Figure 4.9  Input images 5 & 6 size before and after process

In chapter 3, eighteen medical images are discussed and demonstrated. In this chapter for every process, nearly 200 images from various area like CT, US and X-ray etc., are tested but in this
documentation only six images are discussed and demonstrated. Those images are given in Figures 4.7, 4.8 and 4.9. Each image is named as from 1 to 6. The image-1 size is 127.30 kb in Figure 4.7, once this image is encrypted by proposed TVCE system the size of encrypted image is 3502.53 kb in Table-4.1, the proposed compression technique is reduced from original image 1/7 and from the encrypted medical image nearly 1/205 times. After the compression, the compressed image size is 16.97 kb. Once the encrypted image size is reduced, automatically information is also reduced, so the confidentiality of the particular encrypted image is improved.

Table 4.1 Performance measurement based on the size (kb)

<table>
<thead>
<tr>
<th>Original Image Name</th>
<th>Original Size (kb)</th>
<th>Encrypted Size (kb)</th>
<th>Compressed Size (kb)</th>
<th>Decompressed Size (kb)</th>
<th>Decrypted Size (kb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(CAG)</td>
<td>127.30</td>
<td>3502.53</td>
<td>16.97</td>
<td>3501.98</td>
<td>127.34</td>
</tr>
<tr>
<td>2(CT)</td>
<td>234.12</td>
<td>6225.25</td>
<td>30.44</td>
<td>6225.25</td>
<td>234.12</td>
</tr>
<tr>
<td>3(Echo)</td>
<td>102.80</td>
<td>2831.11</td>
<td>13.46</td>
<td>2831.12</td>
<td>102.90</td>
</tr>
<tr>
<td>4(MRI)</td>
<td>158.46</td>
<td>4303.21</td>
<td>21.89</td>
<td>4303.21</td>
<td>158.46</td>
</tr>
<tr>
<td>5(US)</td>
<td>96.26</td>
<td>2613.81</td>
<td>13.80</td>
<td>2613.81</td>
<td>96.26</td>
</tr>
<tr>
<td>6(X-Ray)</td>
<td>239.34</td>
<td>6499.39</td>
<td>29.26</td>
<td>6499.39</td>
<td>239.34</td>
</tr>
</tbody>
</table>

Figure 4.10 Encrypted Image
Table 4.1 gives the information of proposed TVCS and PBSA based on the size for six input grayscale medical images. In Figure 4.10 it is the encrypted image.

![Figure 4.10: Encrypted Image](image)

**Figure 4.11 Proposed TVCS with PBSA performance based on size**

![Figure 4.11: Proposed TVCS with PBSA performance based on size](image)

**Figure 4.12 PBSA compression size performances**

![Figure 4.12: PBSA compression size performances](image)
The size comparison graph for all six images has been given in Figure 4.11. From the graph, it is understood that the encrypted image size is increased to nearly 27 times when compared to the input image size. This is clearly stated in the Figures 4.11 and 4.12 shows how the PBSA compression is performed.

![PSNR graph](image)

**Figure 4.13 Proposed combination (TVCS and PBSA) PSNR performance**

**Table 4.2 Proposed combination (TVCS and PBSA) performance based on the parameters**

<table>
<thead>
<tr>
<th>Original Image Name</th>
<th>Compression Ratio (%)</th>
<th>MSE</th>
<th>PSNR (dB)</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(CAG)</td>
<td>86.60</td>
<td>0.930</td>
<td>48.44</td>
<td>0.993</td>
</tr>
<tr>
<td>2(CT)</td>
<td>86.99</td>
<td>0.683</td>
<td>49.78</td>
<td>0.998</td>
</tr>
<tr>
<td>3(Echo)</td>
<td>86.90</td>
<td>0.734</td>
<td>49.47</td>
<td>0.952</td>
</tr>
<tr>
<td>4(MRI)</td>
<td>86.18</td>
<td>0.632</td>
<td>50.12</td>
<td>0.992</td>
</tr>
<tr>
<td>5(US)</td>
<td>85.97</td>
<td>0.864</td>
<td>48.76</td>
<td>0.932</td>
</tr>
<tr>
<td>6(X-Ray)</td>
<td>87.73</td>
<td>0.654</td>
<td>49.97</td>
<td>0.963</td>
</tr>
</tbody>
</table>
To measure the quality which is given in the limitation of the secure compression, it has been proved in the graph as shown in Figure 4.13. From this graph the proposed combination of algorithms are providing the Peak Signal to Noise Ratio from 48.4 dB to near 50 dB. The minimum value of PSNR is 48.4 provided by proposed combination, it means TVCS with PBSA. In the Table 4.2 is given the entire parameter values which are taken from the experimental, only six input image which is taken for the documentation. Every parameter is calculated by using standard formulae.

![Graph showing CC results for six images](attachment:image.png)

**Figure 4.14 Proposed combination (TVCS and PBSA) CC results**

To check if the proposed algorithms (TVCS and PBSA) have produced correct output or correct reconstructed image, the CC and MSE are doing the vital role. These error matrixes are calculated by the standard formulae. The first point is the system which is produced after the process, output has been the same compared with original information that time CC value was near to 1. From the Figure 4.14 it
clearly stated that all input images (six) CC values are between 0.93 to 1. In this parameter it is proved that, the proposed combination algorithms are reconstructed as same replica of the original image. So, one of the CIA property and lossless has been verified. The second point, when the MSE is near to ‘0’ the same replica of the original image is reconstructed. From this point CIA properties of integrity can be predicted.

![MSE Graph](image)

**Figure 4.15 Proposed combination (TVCS and PBSA) MSE results**

The MSE value for image 1 to image 6 is varied from 0.63 to 0.93 as given in the Table 4.2. Hence, the error occurrence is very minimal as represented in Figure 4.15. Based on the compression ratio values which are provided by the proposed system, the Figure 4.16 is drawn. In the Figure 4.17 the original image size is compared with after reconstructed image size based on the Table 4.1.
Figure 4.16 Proposed combination (TVCS and PBSA) CR results

Figure 4.17 Proposed combination (TVCS and PBSA) reconstruction performance based on the size
4.5 SUMMARY

The chapter 4 discussed the entire process of Pixel Block Short Algorithm and the combination of the proposed TVC and PBSA. The results are taken by this combination which is mentioned in before line. To measure the algorithm efficiency CR, CC, MSE, PSNR, size in all aspects like, before encryption and after decryption, before compression and after decryption are clearly stated in the above Section. The Time is not considered in this chapter. Because, in the chapter 6 while compared with the proposed algorithms, the time is considered as an important factor. In this chapter, the time is measured during the experiment but it is considered in the result and discussion of chapter 6. The chapter 5 discusses another combination of compression as follows,

1. Proposed TVCS with 4-Bit RLE
2. Proposed TVCS with 8-Bit RLE

These two combinations of compression processes with detailed description and results are given in the next chapter.