Enhanced Bandwidth Utilization in Image Steganography with Enhanced Data Security

Balkrishan
Department of CE, Yadavindra College of Engineering, Punjab University, Guru Kashi Campus, Talwandi Sabo-151302
District: Bathinda Punjab, India

Amar Partap Singh
Department of ECE, Sant Longowal Institute of Engineering and Technology,Longowal-148106
District: Sangrur, Punjab, India

ABSTRACT
In this paper, a new method is proposed for image steganography that involves double encryption as well as compression of a message followed by its subsequent hiding in a digital image. In the first instance, original data is encrypted using flexible matrix. Further, encrypted data is encrypted and compressed using Chinese Remainder Theorem (CRT) for extra layer of security and increased the data hiding capacity. The proposed method enhances bandwidth utilization besides ensuring three layered security to the message. The underlying principle of this method involves decomposition of each image-pixel into two blocks. One block is called Parity Reflecting Block (PRB) whereas other is known as Pixel Adjustment Block (PAB). The information about hidden bit is reflected by parity condition in the Parity Reflecting Block. The Pixel Adjustment Block is used to perform local pixel adjustment in order to reduce the degradation effect in the cover image produced as a result of alteration in the moderate bit. The performance of the proposed algorithm is evaluated in terms of Image Quality Measures (IQM) including Mean Square Error (MSE), Peak-Signal-to-Noise Ratio (PSNR), Entropy, Correlation, Mean Value and Standard Deviation. Security analysis is also carried by comparing the histograms of the cover and stego-images. The results of this study are quite promising.

General Terms
Information Security

Keywords
Data Hiding, Chinese Remainder Theorem, Crypto-Compressed-Data, Flexible Matrix, Parity

1. INTRODUCTION
Data encryption [1-4] & compression [2-4] are pivotal for proper storage and transmission of data. The rapid growth in the internet coupled with high bandwidth requirement has propelled the explosive growth of information communication. This type of advancement in the field of data communication has hiked the fear of getting the data snooped at the time of sending it from sender to the receiver. So, information security has already become an important part of data communication. In order to address this issue of information security, data-encryption as well as its hiding [5-16] plays an important role. Data hiding conceals the existence of secret message while cryptography protects its contents. The word steganography is derived from the Greek words- stegos meaning roof or cover and graphia meaning writing, i.e., it is an art of hiding information in such a way that communication takes place without any failure [5-7]. The objective of modern steganography is to keep the payload (embedded information) undetected, but the steganographic systems, because of their invasive nature, may leave behind somewhat traces in the cover image [8]. Steganography techniques use different carriers (cover medium in digital format) to hide data. These carriers may be network packets, floppy disk, hard drive, amateur radio waves [9] or general computer files such as text, image, audio, video, etc. [5, 9-12]. Data hiding capacity & invisibility are the two important parameters used to evaluate the effectiveness of the data hiding technique [14]. In [5] crypto data was embedded into the moderate significant bit of pixel of the cover image by parity condition of bits. The weakness of this method is that author used only lower bits for pixel adjustment and not used the upper bits for re-adjustment of the pixels of the stego image.

The simultaneous data encryption and compression technique is an attempt to provide a solution for optimal bandwidth utilization, space required for data storage and the encryption problems at the same time [2]. Number theory based data encryption and compression is an algorithm that employs Chinese Remainder Theorem (CRT) in order to generate and solve congruencies. However, several methods are available for data or image compression including JPEG-LS, SPIHT, JPEG2000, CALIC, etc. These are standards that use some kind of transform including Discrete Fourier Transformation (DFT), Discrete Cosine Transformation (DCT) & Discrete Wavelet Transformation (DWT) [3]. There are two main drawbacks of traditional transformation based compression methods: (a) In transform techniques, data is transformed from one domain to another, (b) Moreover, these techniques do not encrypt and compress image and/or data in single step [4]. However, such problems of traditional methods of compression are addressed in the proposed approach in which only basic mathematical operations are employed instead of transformation of the data. It also ensures encryption and compression of data in a single step.

Therefore, in the present work, a new type of image steganography is described that combines three operations—cryptography, data compression and steganography in a simultaneous manner. In the first instance, the message is encrypted using flexible matrix [15] serving the purpose of a new type of symmetric key. The data byte is assigned a row number and column number from the flexible matrix itself. All row and column numbers are combined to form one dimensional (1-D) array. This process implements first layer of security to the data. It provides 256 combinations for data byte instead of single combination (remainder and quotients) when data byte is divided by 16 in the CRT. After this, enciphered data is again encrypted as well as compressed using Chinese Remainder Theorem (CRT). In addition to this, it also provides an additional layer of security to the original message and also increases the data hiding capacity of the
cover image by compressing the secret data. At the end, crypto-compressed-data is embedded into the pixel of cover image using moderate-bit substitution thus providing third layer of security in the form of camouflage. Moderate bit substitution is achieved in such a way that no appreciable distortion is observed in the cover image. To improve the image quality further, a pixel adjustment process is also applied at each pixel of the cover image where alteration has occurred due to moderate significant bit substitution. This is done by slight adjustment of other bits in the image pixel without any damage to the secret data. To evaluate the visual quality of stego-image, Image Quality Measures (IQM) are evaluated including Mean Square Error (MSE), Peak-Signal-to-Noise Ratio (PSNR), Entropy, Correlation, Mean Value and Standard Deviation [17]. The proposed method is applied on different standard test-images of size 256x256 serving the purpose of good cover. Extensive experimental results proved that the hidden data remains invisible and there is no appreciable visual distortion in the image at all.

2. CHINESE REMAINDER THEOREM

The Chinese Remainder Theorem (CRT) is a good application of number theory to other fields. It is based on the algorithm of linear and modular congruencies. Congruence is nothing more than a statement about divisibility [4]. The Chinese Remainder Theorem is mainly based on the system of linear congruencies

\[ a = b \pmod{n} \]

which can be reduced to a set of

\[ a = b \pmod{n_i} \]

where \( n_1, n_2, \ldots, n_k \) are prime factors of \( n \).

(a) Theorem

Let \( n_1, n_2, \ldots, n_k \) denote \( k \) positive integers which are relatively prime numbers, and let \( a_1, a_2, \ldots, a_k \) denote any \( k \) integers. Then the congruencies

\[ X \equiv a \pmod{n_i}, \quad \text{where } i = 1, 2, \ldots, k \]

have common solutions. Any two solutions are congruent Modulo \( n_1 n_2 \cdots n_k \).

\[ Y = X \pmod{P} \]

where \( P = n_1 n_2 \cdots n_k \)

Theorem: Any two solutions of \( X \equiv a \pmod{n_i} \) are congruent, that is, relatively prime numbers \( a_1, a_2, \ldots, a_k \) are integers that are relatively prime numbers (\( n_1 \)).

(b) Numerical Example of data encryption and compression using CRT

Let \( n_1 = 17, n_2 = 18, n_3 = 19 \) & \( n_4 = 23 \)

which are relatively prime

Let \( a_1 = 10, a_2 = 14, a_3 = 9 \) & \( a_4 = 8 \)

Encryption:

\[ P = 133722 \]

\[ N_1 = 7866, N_2 = 7429, N_3 = 7038 \quad \text{&} \quad N_4 = 5814 \]

\[ x_1 = 10, x_2 = 7, x_3 = 12 \quad \text{&} \quad x_4 = 9 \]

Decryption:

\[ a_1 \equiv X \pmod{n_i} \]

\[ a_2 = 10, a_3 = 14, a_3 = 9, a_4 = 8 \]

3. PROPOSED METHOD

The secret data is first encrypted using flexible matrix proposed by the authors of the present paper in their earlier work [15]. This matrix is reproduced in Table-1 with new entries for the sake of illustration [15]. The encrypted data is further re-encrypted & then compressed using CRT. The complete process is illustrated in Fig.1. The number theory based technique is applicable for encryption application by suitable selection of relatively prime numbers \( n_i \). Data compression depends on the block size used in CRT.

Relative prime numbers are chosen in such a way that these are larger than any value in one dimensional (1-D) array. During decoding, the same combination of \( n_i \) that are selected for encoding should be applied correctly. To achieve safe transfer of data and obtain better results of embedding, moderate-bit substitution method is proposed in the present work for hiding crypto-compressed-data in a digital image. In this paper, for compression purpose, the block size (1×K) with \( K = 4 \) is taken. Depending on the amount of compression requirement, a large block size can also be considered.

3.1 Data Encryption and Compression using CRT

Consider a data of \( n \) characters and assign the row number and column number of \( n \) characters using flexible matrix [15]. The Row number and column number are combined to form 1-D array. Pad the 1-D array with zeros to make a block size of 10 or 8 or 4. Now, 1-D array is solved using the Chinese Remainder Theorem for block of 4. In equation (2) \( ni \) and \( xi \) are pre-calculated as coefficients and these values need not be calculated for every \( X \). All \( ai \) (\( i = 1, \ldots, m \), where \( m \) is the elements in 1-D array) are the 1-D array values after encryption using flexible matrix. The reason for using Chinese Remainder Theorem for solving the linear congruencies is to reduce a bigger number to a smaller representation. For data of size 1x3000 bytes and block size 4, all 1500 or less \( X \) are computed. After computing all \( X \), the frequency of each distinct \( X \) and their counts are determined. All \( X \) are sorted in descending order of their count. A table of unique \( X \) and an equivalent smaller code is generated. Using this table, each \( X \) obtained is encoded into this smaller code. Data compression ratio was 1.35 for 3000 bytes. For the sake of clarification, the complete process is illustrated here.

3.2 Data Decryption and Decompression

Data decoding is performed at the receiving end. At the receiver, \( Ai \) is found for each \( X \) using the equation (4). Single 1-D array of all \( Ai \) is divided into two equal size arrays (Row number & Column number). The original pixel values are then reconstructed using the flexible matrix.
3.3 Illustration

<table>
<thead>
<tr>
<th>Original message to be embedded</th>
<th>Image steganography</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data byte/ASCII value of character</td>
<td>105 109 97 103 101 32</td>
</tr>
<tr>
<td>Encrypt the message using flexible matrix</td>
<td>115 116 101 103 97 110</td>
</tr>
<tr>
<td></td>
<td>111 103 114 97 112 104</td>
</tr>
<tr>
<td></td>
<td>121</td>
</tr>
<tr>
<td>Compressed crypto data using flexible matrix</td>
<td>Column array = 9 9 9 9 9 9</td>
</tr>
<tr>
<td></td>
<td>9 9 9 9 9 9 9 9 9 9 9 9 9 10 9 10 10</td>
</tr>
<tr>
<td></td>
<td>Row array = 8 9 1 6 4 0</td>
</tr>
<tr>
<td></td>
<td>13 14 4 6 1 10 11 6</td>
</tr>
<tr>
<td></td>
<td>12 1 6 7 7</td>
</tr>
<tr>
<td>Further encrypt and compress crypto data using Chinese Remainder Theorem</td>
<td>1-D array = 9 9 9 9 9 9 9 9 9</td>
</tr>
<tr>
<td></td>
<td>9 9 9 9 9 9 9 9 9 9 9 9 9 10 9 10 8 9 1 6</td>
</tr>
<tr>
<td></td>
<td>4 0 13 14 4 6 1 10</td>
</tr>
<tr>
<td></td>
<td>11 6 12 1 6 7 7 0 0</td>
</tr>
<tr>
<td>Data embedding into cover image to obtain stego-image</td>
<td>Calculate the values of X and assign the smaller code to each X</td>
</tr>
<tr>
<td></td>
<td>Data embedding at Moderate Significant Bit position</td>
</tr>
</tbody>
</table>

3.4 Crypto-compressed-data hiding

To understand the process of crypto-compressed-data hiding, it is assumed that a cover image is composed of pixels with odd or even number of one’s. In the pixel of cover image, bit is altered at moderate position depending upon the parity condition of the bits counted from moderate position to the most significant bit (MSB). All the lower bits from moderate position to Least Significant Bit (LSB) are used for local pixel adjustment. The procedure used for moderate bit alteration followed by local pixel adjustment is narrated below:

I. If the PRB is of odd parity and message bit is 1, then there is no change in both the pixel blocks of the image pixel. The odd parity condition in the PRB reflects that the stored bit is 1 [16].

II. If the PRB is of odd parity and the message bit is 0, then complement the 4th moderate significant bit. Convert all the lower bits to 1 if moderate bit is changed to 0 or vice-versa otherwise [16].

III. If the PRB is of even parity and the message bit is also 0, then there is no change in both the pixel blocks of the image pixel. The even parity condition of the PRB reflects that image pixel stores 0 as the message bit [16].

IV. If the PRB is of even parity and the message bit is 1, then complement the 4th moderate significant bit. Convert all the lower bits to 1 if moderate bit is changed to 0 or vice-versa otherwise [16].

3.5 Pixel Adjustment Process

Following are the two steps used in pixel adjustment process.

Pixel Adjustment [16]

If crypto-bit to be embedded is equal to the 4th LSB, then there is no need for doing any pixel adjustment. However, if crypto-bit to be embedded is not equal to 4th LSB, pixel adjustment is performed by using first three LSBs. The underlying logic for the same is detailed below:

(a) If crypto-bit to be embedded is one, then pixel is adjusted by changing first three LSBs to zero.

(b) If crypto-bit to be embedded is zero, then pixel is adjusted by changing first three LSBs to one.

Post pixel adjustment

In this step, post pixel adjustment is applied further improve the visual quality of stego image obtained after pixel adjustment. This process involves only two bits, i.e., 5th and 6th bits. Let P, P’ and P" are the values of ith pixel in the cover-image, modified pixel of stego-image after pixel adjustment and original pixel of cover image, i.e., D1 = abs (P-P’). If D1 > 4, then post pixel adjustment is required otherwise there is no need of doing post pixel adjustment. The value of D1 > 4 is chosen in such a way that it gives better visual quality as compared to its other values including D = 1, 2, 3, 5, 6, 7 or 8. The underlying logic for the same is detailed below:

(a) If after the process of embedding, 4th & 5th LSBs are unequal and 4th & 6th LSBS are equal, then post pixel adjustment is performed by complementing 1st, 2nd, 3rd, 5th and 6th bits of P’. In fact, the proposed method provides a three layered security to protect the hidden data. First, cryptography technique is used to protect the information. Secondly, data hiding capacity of cover image is increased by encrypting and compressing the crypto data using CRT. Then, the cipher information so obtained is embedded in the cover image. The data hiding procedure is illustrated in Fig.2.
4. PROPOSED ALGORITHM FOR HIDING CRYPTO COMPRESSED DATA

4.1 Message Ciphering & Embedding

Step-1: Save secret message as a text file.

Step-2: Commencing with first character, read secret message character-wise from saved text file.

Step-3: Encrypt each character into row and column numbers using flexible matrix.

Step-4: Repeat Step-3 for all characters in the saved text file to obtain a row and column numbers.

Step-5: Combine all the row and column numbers into 1-D array.

Step-6: Pad the 1-D array with zeros such that total values in array must be divided by 4 for a block size of 4.

Step-7: Further encrypt and compress the ciphered data by applying the CRT using block of 4 values of 1-D array to calculate the value of X.

Step-8: Repeat step 7 for all values in 1-D array to obtain the values of X.

Step-9: After computing all X, the frequency of each distinct X and their counts are determined. All X are sorted in descending order of their count. A table of unique X and an equivalent smaller code is generated. Using this table, each X obtained is encoded into this smaller code.

Step-10: All value of X and smaller codes are converted into equivalent binary number (all X and smaller codes have equal number of bits by padding with zeros from left side) and make a binary string.

Step-11: Read each pixel of the cover image commencing with first pixel.

Step-12: Convert each pixel into equivalent eight-bit binary number called image byte.

Table 1. Proposed 16x16 Matrix for Enciphering Data [15]

<table>
<thead>
<tr>
<th>Column Number Row Number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>192</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
<td>199</td>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
<td>204</td>
<td>205</td>
<td>206</td>
<td>207</td>
</tr>
<tr>
<td>1</td>
<td>208</td>
<td>209</td>
<td>210</td>
<td>211</td>
<td>212</td>
<td>213</td>
<td>214</td>
<td>215</td>
<td>216</td>
<td>217</td>
<td>218</td>
<td>219</td>
<td>220</td>
<td>221</td>
<td>222</td>
<td>223</td>
</tr>
<tr>
<td>2</td>
<td>224</td>
<td>225</td>
<td>226</td>
<td>227</td>
<td>228</td>
<td>229</td>
<td>230</td>
<td>231</td>
<td>232</td>
<td>233</td>
<td>234</td>
<td>235</td>
<td>236</td>
<td>237</td>
<td>238</td>
<td>239</td>
</tr>
<tr>
<td>3</td>
<td>240</td>
<td>241</td>
<td>242</td>
<td>243</td>
<td>244</td>
<td>245</td>
<td>246</td>
<td>247</td>
<td>248</td>
<td>249</td>
<td>250</td>
<td>251</td>
<td>252</td>
<td>253</td>
<td>254</td>
<td>255</td>
</tr>
<tr>
<td>4</td>
<td>143</td>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
<td>150</td>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
<td>156</td>
<td>157</td>
<td>158</td>
<td>159</td>
</tr>
<tr>
<td>5</td>
<td>160</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
<td>168</td>
<td>169</td>
<td>170</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
</tr>
<tr>
<td>6</td>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
<td>180</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
<td>191</td>
</tr>
<tr>
<td>7</td>
<td>64</td>
<td>55</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>8</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td>9</td>
<td>32</td>
<td>97</td>
<td>99</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>109</td>
<td>110</td>
<td>111</td>
<td>112</td>
</tr>
<tr>
<td>10</td>
<td>46</td>
<td>98</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>113</td>
<td>112</td>
<td>111</td>
<td>112</td>
<td>113</td>
<td>114</td>
<td>119</td>
<td>118</td>
<td>120</td>
<td>122</td>
<td>124</td>
</tr>
<tr>
<td>11</td>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
<td>132</td>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
<td>139</td>
<td>140</td>
<td>141</td>
<td>142</td>
<td>144</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>13</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>14</td>
<td>123</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>15</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td></td>
</tr>
</tbody>
</table>
Step-13: Convert image byte into two blocks-Pixel Adjustment Block and Parity Reflecting Block.

Step-14: Determine the parity of the Parity Reflecting Block (PRB) & read first crypto-compressed-bit.
   a) If the PRB is of odd parity and message bit is 1, then there is no change in both the pixel blocks of the image pixel. The odd parity condition in the PRB reflects that the stored bit is 1.
   b) If the PRB is of odd parity and the message bit is 0, then complement the 4th moderate significant bit. Convert all the lower bits to 1 if moderate bit is changed to 0 or vice-versa otherwise.
   c) If the PRB is of even parity and the message bit is also 0, then there is no change in both the pixel blocks of the image pixel. The even parity condition of the PRB reflects that image pixel stores 0 as the message bit.
   d) If the PRB is of even parity and the secret message bit is 1, then complement the 4th moderate significant bit. Convert all the lower bits to 1 if moderate bit is changed to 0 or vice-versa otherwise.

Step-15: Go to next image byte and next crypto-compressed-bit and repeat the Steps-12 to 14 until all the crypto-compressed bits of the secret message are embedded into the PRBs of the cover image.

4.2 Message Extraction & Decryption

Step-1: Read the pixel of stego-image starting from first pixel.

Step-2: Convert each pixel value into equivalent binary number called image byte.

Step-3: Extract first crypto-compressed-bit by determining the parity condition of the Parity Reflecting Block. If it is odd parity then embedded crypto-bit is 1 otherwise it is 0.

Step-4: Repeat steps 2 to 3 until all the crypto-compressed-bits of the secret message are extracted.

Step-5: Divide the crypto-compressed-bits for values of X and smaller codes.

Step-6: Smaller codes are also replaced with X values.

Step-7: Apply the equation (4) and A_i of each X using the equation.

Step-8: Repeat step 7 for all values of X.

Step-9: Combine all A_i to form a single 1-D array. Divide the 1-D array into two equal sized arrays (Row & Column). The original pixel values are then reconstructed using the flexible matrix.

Step-10: Save all characters in the form of text file.

5. RESULTS AND DISCUSSION

In this section, simulation results of the proposed algorithm are presented by embedding 3000 bytes of crypto-compressed data at moderate significant bit positions of the pixels in the cover image. For experiments, four standard gray scale test images with 256x256 pixel size are employed as cover images (Lena, Mandrill, Pepper and Cameraman). MATLAB software is used to implement the algorithm and validate the results. Original cover images are shown in Fig. 3(a) along with their corresponding stego-images are shown in Fig. 3 (b) & (c) that are generated as a result of proposed algorithm. The proposed technique embeds crypto-compressed-data into moderate-significant-bit positions in the image-pixels without causing any appreciable distortion in the cover image. Visual quality of a stego image is an important parameter in evaluating the performance of the proposed algorithm. It is expressed in terms of Image Quality Measure (IQM) parameters including Mean Square Error (MSE), Peak-Signal-to-Noise Ratio (PSNR), Entropy, Correlation, Mean Value and Standard Deviation. The performance of the proposed algorithm is summarized in terms of IQM parameters. These results are tabulated in Table-2, Table-3 and Table-4. The results in these tables indicate that PSNR, Entropy, correlation, Mean value and Standard Deviation are decreasing whereas the MSE is increasing as the hidden message bit moves from LSB towards moderate significant bit position. The performance of the proposed method is better than [16] in terms of IQM parameters shown in Tables 2, 3 and 4.

The histograms of cover and stego-images give clear idea of the security of the transmitted data, i.e., if change in the cover is minimal, then stego system is considered secure. In this context, the effectiveness of the proposed algorithm is also determined in terms of having a comparison of the histograms of cover and stego images as show in Fig.4. The histogram analysis of the stego-images proves that these images look exactly similar to that of the original image. Further, this process provides two additional layers of security to the hidden message. First, cryptography technique is used to protect the information and it requires less computational overhead as compared to other steganographic techniques which transforms cover images into frequency domain. Secondly data hiding capacity of cover image is increased by encrypting and compressing the ciphered data using CRT. It also provides 256 combinations for data byte instead of single combination (remainder and quotients) when data byte is divided by 16 in CRT.

### Table 2. Mean Square Error and Peak Signal-to-Noise Ratio resulted from hiding 3000 bytes of Crypto-Compressed data

<table>
<thead>
<tr>
<th>Test Image (256x256)</th>
<th>Peak Signal-to-Noise Ratio (PSNR)</th>
<th>Mean Square Error (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>42.7959</td>
<td>43.6624</td>
</tr>
<tr>
<td>Goldhill</td>
<td>42.7764</td>
<td>43.3040</td>
</tr>
<tr>
<td>Mandrill</td>
<td>42.5498</td>
<td>43.2155</td>
</tr>
<tr>
<td>Camera</td>
<td>41.7003</td>
<td>42.2161</td>
</tr>
</tbody>
</table>

### Table 3. Entropy and Correlation resulted from hiding 3000 bytes of Crypto-Compressed data

<table>
<thead>
<tr>
<th>Test Image (256x256)</th>
<th>Entropy</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>5.1151</td>
<td>5.1239</td>
</tr>
<tr>
<td>Goldhill</td>
<td>5.1016</td>
<td>5.1090</td>
</tr>
<tr>
<td>Mandrill</td>
<td>5.1695</td>
<td>5.1748</td>
</tr>
<tr>
<td>Camera</td>
<td>4.8165</td>
<td>4.8258</td>
</tr>
</tbody>
</table>
CONCLUSION

In this study, an attempt is made to develop a new approach for data hiding in grey scale images in which data encryption, compression and hiding are combined to enhance bandwidth utilization and achieve secure communication. First layer of security is achieved by using a symmetric key based on the concept of a flexible matrix. The data hiding capacity of the cover image is enhanced by encrypting as well as compressing the ciphered data using CRT. In this algorithm, upper 5th & 6th bits of the image pixel are also used for post pixel adjustment to further improve the visual perception of the stego image. In fact, by doing so, an additional layer of security to the original message is also provided besides increasing data hiding capacity of the cover image. The proposed algorithm requires less computational overhead as compared to other steganographic techniques which transform the cover image into frequency domain. Experiment results have demonstrated that the proposed scheme for data hiding works satisfactorily for different gray level digital images. Moreover, the proposed method provides acceptable image quality with very little distortion in the cover image.

REFERENCES


<table>
<thead>
<tr>
<th>Test Image (256×256)</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cover image</td>
<td>With [16]</td>
</tr>
<tr>
<td>Lena</td>
<td>124.0923</td>
<td>124.1127</td>
</tr>
<tr>
<td>Goldhill</td>
<td>112.0349</td>
<td>112.2391</td>
</tr>
<tr>
<td>Mandrill</td>
<td>102.7616</td>
<td>102.7898</td>
</tr>
<tr>
<td>Camera</td>
<td>118.7228</td>
<td>118.6388</td>
</tr>
</tbody>
</table>

Fig 3: Results of Experimental Validation

(a) Original cover images
(b) Stego images obtained with the proposed method in [16]
(c) Stego images obtained with the proposed method

Fig 4: Histograms of cover & stego-images

(a) Histograms of original cover images
(b) Histograms of stego images obtained with the proposed method [16]
(c) Histograms of stego images obtained with the proposed method


