Chaotic Image Encryption Technique using S-box based on DNA Approach

Anchal Jain  
Inderprastha Engineering College, Ghaziabad, India.

Pooja Agarwal  
Inderprastha Engineering College, Ghaziabad, India.

Rashi Jain  
Inderprastha Engineering College, Ghaziabad, India.

Vyomesh Singh  
Inderprastha Engineering College, Ghaziabad, India.

ABSTRACT

In recent years many DNA approach based encryption algorithms have been suggested but many of them are vulnerable to attacks or have slow execution time. Recently DNA index based techniques have also been suggested. In this paper a new image encryption is proposed which is effective and has improved execution time compared to index based symmetric cryptographic algorithms having DNA approach. The algorithm presented in the paper is an integrated approach coupling the power of S-Box's and the two dimensional logistic map. Theoretical analysis and experimental results show that the proposed algorithm is reasonably secure and faster in run time compared to indexed based approach.

Keywords

Cryptography, DNA, Logistic map, S-box.

1. INTRODUCTION

The rapid advancement in the field of information technology has increased the capability to exchange a large volume data across the widespread network of computers. Security is a prime concern when data is transmitted across the network of computers. Various types of attacks are possible on the data being transmitted so protecting the data against such unauthenticated access and keeping its integrity intact is an important task.

Cryptography is mainly divided in two fields the symmetric key cryptography and asymmetric cryptography depending upon the no of keys used in the process of encryption and decryption. Since traditional cryptographic algorithms are susceptible to various kinds of attack therefore the thrust to explore new ideas to enhance the security of data has led to the emergence of new era of data security based upon the analogy of the biological sciences with the computer sciences called DNA cryptography. Deoxyribose Nucleic Acid (DNA) is the genetic material inside the living creatures responsible for carrying the hereditary features from parents to the offspring. DNA is polymer which is made up of several monomer units called nucleotides. Each nucleotide is further composed of following three units: Deoxyribose sugar, phosphate group and the nitrogenous base. The nitrogenous bases are Adenine, Guanine, Thymine and Cytosine. Watson-Crick proposed the complementary DNA structure and this inherent complementarity is used in DNA computing. The ability to store large amount of data, inherent complementarily and massive parallelism makes it suitable choice towards security. DNA computing does not actually involve the real biological DNA strands rather it uses the central principle behind DNA. Initially work in DNA computing was done by Adleman [1] where Hamiltonian path problem was solved using DNA approach. The principle which Adleman used is the ability to represent information in DNA clusters. Gehani et al. [2] also proposed DNA based cryptography. Since then several algorithms have been proposed by various others which include the DNA based implementation of YAEA algorithm [3]. Various authors have taken the advantage of DNA based operations such as addition, complementation, deletion, insertion, truncation, transformation for proposing encryption scheme. Qiang Zhang et al. [4] proposed an image encryption using DNA addition combining with chaotic maps. But Houcemeddine Hermassi et al. [5] proved that this algorithm was weak against known plain text attacks. The complementary property of DNA nucleotides is also utilized in DNA based cryptography. Hongjun Liu et al. [6] proposed the image encryption scheme based on DNA complementary rule combined with chaotic map. Qiang Zhang [7] also proposed an encryption scheme using DNA operations like as deletion, insertion, truncation, and transformation combined with chaotic maps. Chaotic maps have been widely used by authors to propose effective encryption schemes. In paper [8] also, one dimensional chaotic logistic map is used to propose encryption scheme using neural network.

Recently Grasha Jacob et al. [9, 10] proposed two image encryption schemes using DNA gene bank based indexing. Zhang et al. [11] also proposed the index based symmetric DNA encryption algorithm using chaotic key generator. In DNA based indexing approach the cipher generation is based upon searching the DNA encoded original plain text pattern in gene bank, the search is sequential and execution is slow in generating cipher. All these papers [9, 10, 11], have time complexity issue. In this paper a new effective and comparatively faster image encryption technique based on DNA substitution box is proposed. The proposed algorithm is compared with the techniques [9, 10] and found to be faster in execution time. The paper is organized in total 5 sections. In section 2, the step by step process for encryption is described. In section 3, decryption algorithm is described. In section 4, the algorithm is analyzed against various attacks such as, brute force attack, key space analysis, statistical analysis including correlation and histogram analysis and in section 5 paper is concluded.

2. PROPOSED ENCRYPTION PROCESS

The image is first encoded into DNA sequence and the resulting sequence is substituted using DNA based S-Box which generates the intermediate cipher and finally a chaotic sequence is generated using 2 D logistic mapping which
permute the rows and columns of intermediate cipher to produce final cipher. Encryption process is divided into following steps:

### 2.1 DNA Encoding of Image

A gray scale image has the pixel range from 0 to 255. The input image of size MxN is taken and each pixel value is converted into its 8 bit binary equivalent. Now, the 8 bit binary equivalent is converted into a quadruple nucleotide sequence using the following scheme:

\[
\begin{align*}
C & \rightarrow 00 \\
T & \rightarrow 01 \\
A & \rightarrow 10 \\
G & \rightarrow 11
\end{align*}
\]

The process is repeated for all the pixels in the image and a matrix of size Mx4N is generated.

### 2.2 Generation of DNA based S-Box

The inception behind S-box traces back to cryptographic substitution introduced by Shannon [12]. S-box can be classified in two groups namely static S-box where the input vector does not changes and dynamic S-box where the change input values result in the changes of corresponding output as well [13].

A good design of nonlinear S-box ensures better cipher complexity. The proposed algorithm in this paper uses a DNA based static S-box for the encryption scheme. To create DNA S-box a gene sequence is downloaded from the gene bank. This sequence is a binary file that contains a single DNA strand.

Canis_familiaris chromosomal sequence (genome) from the gene bank is sufficiently large enough to encode image. For sake of simplicity the algorithm has been implemented on gray scale images. Intensity values of gray scale image lies in the range 0 ≤ f(x, y) ≤ 255. Therefore, the static S-box will be of size 16x16 having total 256 entries in it. Each such entry will correspond to a gray scale value in image. Start the search in the downloaded genome from location x which is part of secret key for the quadruple nucleotide sequences and pick 256 quadruple with equivalent distinct decimal values. For example suppose one of the quadruple DNA nucleotide sequence is “ATGC” the equivalent decimal value will be 156. Therefore 256 distinct values are chosen and filled in S-box sequentially in row major fashion. The generated s box is shown in figure 1.

### 2.3 Generation of Chaotic Sequence and Final Cipher

For generating the initial condition method described in [4] is used. Calculate two parameters k1 and k2 as in (1)

\[
k_1 = \frac{1}{256} \mod (\sum_{i=1}^{m} \sum_{j=1}^{n} P_{ij}, 256)
\]

\[
k_2 = \frac{1}{256} \mod (\sum_{i=2}^{m} \sum_{j=1}^{n} P_{ij}, 256)
\]

where \( P_{ij} \) is the value of the image pixel at location (i, j) in the image. Further let \( x_0 = 0.59 \) and \( y_0 = 0.15 \). Calculate initial conditions as in (2)

\[
x_0 = \mod \left( x_0 + k_1, 1 \right)
\]

\[
y_0 = \mod \left( y_0 + k_1, 1 \right)
\]

The proposed algorithm uses 2 D logistic map based on chaotic sequence as defined in (3)

\[
x_{i+1} = \frac{\beta_1 x_i (1 - x_i) + \gamma_1 y_i^2}{x_i^2 + y_i^2}
\]

\[
y_{i+1} = \frac{\beta_2 y_i (1 - y_i) + \gamma_2 x_i^2 + x_i y_i}{x_i^2 + y_i^2}
\]

where \( 2.75 < \mu_1 \leq 3.4, 2.75 < \mu_2 \leq 3.45, 0.15 < \gamma_1 \leq 0.21, 0.13 < \gamma_2 \leq 0.15 \) for example values are set \( \gamma_1 = 0.16 \) and \( \gamma_2 = 0.14 \).

---

**Table 1:** Static S box generated from the gene bank having 256 distinct values

<table>
<thead>
<tr>
<th>INDEX</th>
<th>CC</th>
<th>CT</th>
<th>CA</th>
<th>CG</th>
<th>TC</th>
<th>TT</th>
<th>TA</th>
<th>TG</th>
<th>AC</th>
<th>AT</th>
<th>AA</th>
<th>AG</th>
<th>GC</th>
<th>GT</th>
<th>GA</th>
<th>GG</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>131</td>
<td>208</td>
<td>50</td>
<td>13</td>
<td>64</td>
<td>216</td>
<td>15</td>
<td>129</td>
<td>32</td>
<td>211</td>
<td>0</td>
<td>67</td>
<td>96</td>
<td>248</td>
<td>38</td>
<td>14</td>
</tr>
<tr>
<td>CT</td>
<td>108</td>
<td>54</td>
<td>128</td>
<td>48</td>
<td>52</td>
<td>2</td>
<td>99</td>
<td>40</td>
<td>97</td>
<td>11</td>
<td>56</td>
<td>220</td>
<td>62</td>
<td>12</td>
<td>224</td>
<td>66</td>
</tr>
<tr>
<td>CA</td>
<td>8</td>
<td>205</td>
<td>136</td>
<td>24</td>
<td>20</td>
<td>100</td>
<td>6</td>
<td>1</td>
<td>202</td>
<td>3</td>
<td>212</td>
<td>180</td>
<td>244</td>
<td>112</td>
<td>140</td>
<td>178</td>
</tr>
<tr>
<td>CG</td>
<td>139</td>
<td>195</td>
<td>160</td>
<td>61</td>
<td>192</td>
<td>176</td>
<td>58</td>
<td>34</td>
<td>179</td>
<td>227</td>
<td>53</td>
<td>254</td>
<td>232</td>
<td>246</td>
<td>35</td>
<td>63</td>
</tr>
<tr>
<td>TC</td>
<td>190</td>
<td>23</td>
<td>76</td>
<td>200</td>
<td>55</td>
<td>252</td>
<td>80</td>
<td>240</td>
<td>222</td>
<td>127</td>
<td>207</td>
<td>141</td>
<td>184</td>
<td>29</td>
<td>206</td>
<td>134</td>
</tr>
<tr>
<td>TT</td>
<td>217</td>
<td>47</td>
<td>143</td>
<td>51</td>
<td>59</td>
<td>162</td>
<td>4</td>
<td>104</td>
<td>16</td>
<td>218</td>
<td>72</td>
<td>22</td>
<td>135</td>
<td>45</td>
<td>98</td>
<td>172</td>
</tr>
<tr>
<td>TA</td>
<td>132</td>
<td>44</td>
<td>120</td>
<td>204</td>
<td>28</td>
<td>247</td>
<td>253</td>
<td>88</td>
<td>163</td>
<td>255</td>
<td>115</td>
<td>189</td>
<td>223</td>
<td>17</td>
<td>199</td>
<td>230</td>
</tr>
<tr>
<td>TG</td>
<td>237</td>
<td>126</td>
<td>241</td>
<td>251</td>
<td>60</td>
<td>119</td>
<td>245</td>
<td>147</td>
<td>65</td>
<td>243</td>
<td>137</td>
<td>138</td>
<td>130</td>
<td>74</td>
<td>117</td>
<td>57</td>
</tr>
<tr>
<td>AC</td>
<td>94</td>
<td>168</td>
<td>221</td>
<td>101</td>
<td>122</td>
<td>148</td>
<td>10</td>
<td>103</td>
<td>153</td>
<td>69</td>
<td>175</td>
<td>79</td>
<td>242</td>
<td>239</td>
<td>191</td>
<td>111</td>
</tr>
<tr>
<td>AT</td>
<td>219</td>
<td>125</td>
<td>95</td>
<td>249</td>
<td>114</td>
<td>49</td>
<td>7</td>
<td>43</td>
<td>209</td>
<td>146</td>
<td>33</td>
<td>18</td>
<td>42</td>
<td>157</td>
<td>78</td>
<td>151</td>
</tr>
<tr>
<td>AA</td>
<td>234</td>
<td>90</td>
<td>183</td>
<td>82</td>
<td>165</td>
<td>89</td>
<td>81</td>
<td>87</td>
<td>235</td>
<td>152</td>
<td>250</td>
<td>75</td>
<td>118</td>
<td>124</td>
<td>113</td>
<td>226</td>
</tr>
<tr>
<td>AG</td>
<td>92</td>
<td>68</td>
<td>164</td>
<td>36</td>
<td>167</td>
<td>83</td>
<td>145</td>
<td>106</td>
<td>37</td>
<td>86</td>
<td>173</td>
<td>70</td>
<td>169</td>
<td>85</td>
<td>142</td>
<td>166</td>
</tr>
<tr>
<td>GC</td>
<td>236</td>
<td>109</td>
<td>203</td>
<td>174</td>
<td>19</td>
<td>21</td>
<td>229</td>
<td>213</td>
<td>197</td>
<td>84</td>
<td>210</td>
<td>93</td>
<td>228</td>
<td>171</td>
<td>39</td>
<td>5</td>
</tr>
<tr>
<td>GT</td>
<td>156</td>
<td>121</td>
<td>185</td>
<td>158</td>
<td>30</td>
<td>46</td>
<td>187</td>
<td>215</td>
<td>186</td>
<td>116</td>
<td>283</td>
<td>149</td>
<td>27</td>
<td>123</td>
<td>225</td>
<td>144</td>
</tr>
<tr>
<td>GA</td>
<td>150</td>
<td>177</td>
<td>233</td>
<td>170</td>
<td>231</td>
<td>196</td>
<td>159</td>
<td>188</td>
<td>25</td>
<td>161</td>
<td>41</td>
<td>26</td>
<td>31</td>
<td>77</td>
<td>194</td>
<td>9</td>
</tr>
<tr>
<td>GG</td>
<td>155</td>
<td>201</td>
<td>107</td>
<td>133</td>
<td>181</td>
<td>71</td>
<td>91</td>
<td>73</td>
<td>214</td>
<td>110</td>
<td>102</td>
<td>105</td>
<td>154</td>
<td>198</td>
<td>182</td>
<td>193</td>
</tr>
</tbody>
</table>

---

**Fig. 1:** Static S box generated from the gene bank having 256 distinct values
\( \mu_1=3.2 \) and \( \mu_2=3.3 \) these values are part of the secret key. Use \( x_0 \) and \( y_0 \) as initial conditions for (3) and obtain two matrices of size \( 1 \times 256 \) as in (4).

\[
X_m = (x_1, x_2, x_3, \ldots, x_m) \\
Y_m = (y_1, y_2, y_3, \ldots, y_m)
\]

Using (4) permute the rows and columns as in (5):

\[
R(i) = R((x_i \times m) \mod i) \\
C(j) = C((y_j \times n) \mod j)
\]

**The pseudo code for encryption**

**Input:** X (image) to be encrypted, G (Binary file which contains the nucleotide sequences) and Secret Key (start location \( x \) for search in gene bank, end location \( y \) in gene bank, \( x_0, y_0, \mu_1, \mu_2, \gamma_1, \gamma_2 \))

**Output:** Encrypted image Y.

**Step 1:** DNA encoding of input image resulting in a matrix P of size \( (m \times 4n) \) as described in subsection 2.1.

**Step 2:** Create a DNA based S-box as described in subsection 2.2.

**Step 3:** Now substitute the pixel values of image with the values from S-box as:

For example: If \( X(i, j) = 109 \)

Then equivalent DNA encoded quadruple is \( M = \text{TAGT} \) Now this \( M \) can be split into 2 parts, \( m = \text{TA} \) and \( n = \text{GT} \).

The new substituted values \( X(i, j) \) are calculated using \( m \) and \( n \) as index in substitution box. Where \( m \) is the row index and \( n \) is the column index.

\( X(i, j) = \text{S_box} (\text{decimal}(m), \text{decimal}(n)) \).

This generates an intermediate cipher image of the same size as that of the original image.

**Step 4:** Now permute the rows and columns as described in (5). This step generates the final cipher image Y.

**3. DECRIPTION**

**Input:** Encrypted image Y, G (Binary file which contains the nucleotide sequences) and Secret Key (start location \( x \) for search in gene bank, end location \( y \) in gene bank, \( x_0, y_0, \mu_1, \mu_2, \gamma_1, \gamma_2 \)).

**Output:** Original image X.

**Step 1:** Perform the reverse permutation operation on the rows and columns of the cipher image Y using the reversible mapping of (5).

**Step 2:** Create an inverse S-box and perform reverse substitution to obtain the original image.

**4. SECURITY ANALYSIS OF THE ALGORITHM**

The algorithm was simulated on Matlab R2012a to determine the efficiency and security against various kinds of attacks. Standard \( 256 \times 256 \) gray image Lena.tif and Peppers.png are used and the initial parameters as \( x_0 = 0.59, y_0 = 0.15, \mu_1 = 3.2, \mu_2 = 3.3, \gamma_1 = 0.16 \) and \( \gamma_2 = 0.14 \). The encrypted image is shown in figure (2).

**4.1 Efficiency of the Algorithm**

The algorithm was tested on different images and the run time for the encryption scheme was noted down and tabulated along with the run time of the previous algorithm that used indexed based sequential search for cipher generation. The results in table 1 shows that proposed algorithm has better run time than the run time of encryption scheme in [9, 10]. The run time result has been calculated by finding the average of the values obtained by applying the algorithm on different images.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Algorithm</td>
<td>Generation of S Box: 4950.408</td>
</tr>
<tr>
<td>Indexed Based Algorithm</td>
<td>36078.380</td>
</tr>
</tbody>
</table>

**4.2 Security Analysis**

The resistance of the algorithm against various categories of attacks defines its ability to protect the data in transit. The process of cryptanalysis includes various cryptographic attacks and the immunity of the cryptographic algorithm against such attacks should be at par without the access of keys. An encryption technique must be able to resist against cryptanalytic and brute force attack.

**4.2.1 Key Space Analysis**

In proposed algorithm, location of gene bank, the initial value and the system parameter of the chaotic maps can be seemed as secret key. Thus, there are eight secret keys. (Start location \( x \), end location \( y \), \( x_0, y_0, \mu_1, \mu_2, \gamma_1, \gamma_2 \)) If the precision is \( 10^{-12} \), the secret key’s space is \( (10^{-12} \times 10^{-12} \times 10^{-12} \times 10^{-12} \times 10^{-12} \times 10^{-12} \times 10^{-12} \times 10^{-12}) \). The secret key’s space is large enough to resist exhaustive attack.

**4.2.2 Statistical Analysis**

The cipher generated by the algorithm must not show any predictable statistical relation to the original image in order to prevent the attacker to use that information and to use it to decipher the information being transmitted. Statistical attacks include histogram analysis and correlation coefficient analysis.

**4.2.2.1 Histogram Analysis**

The histogram of an image is the profile of the image describing the distribution of the image pixels against the various intensity levels. A good encryption algorithm is the one which generates a cipher image whose histogram is
completely different from that of an original image. Thus it will be difficult for attackers to identify the pixels from the encrypted image having similar nature to that of the original image. The histogram for the cipher and the original image has been shown in figure 2.

4.2.2.2 Correlation Coefficient Analysis
There exists a high correlation between the pixel values of image. The adjacent pixels are highly correlated. A good encryption algorithm must produce an encrypted image where there exists a very little correlation between the existing pixels. Pearson’s correlation coefficient is given by the formula:

\[ r = \frac{n \sum xy - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2} \sqrt{n(\sum y^2) - (\sum y)^2}} \]  

\[ (6) \]

Table 2: Comparison of Correlation Coefficients

<table>
<thead>
<tr>
<th>File Name</th>
<th>Peppers.png</th>
<th>Lena.tif</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y_ref</td>
<td>0.9478</td>
<td>0.9201</td>
</tr>
<tr>
<td>Encrypted image</td>
<td>0.0271</td>
<td>0.0200</td>
</tr>
<tr>
<td>Y_row</td>
<td>0.9482</td>
<td>0.9560</td>
</tr>
<tr>
<td>Encrypted image</td>
<td>0.0330</td>
<td>0.0267</td>
</tr>
</tbody>
</table>

Where \( x \) and \( y \) are the grayscale values of adjacent pixels in the image and \( n \) is the number of image pixels chosen for calculation of correlation coefficient. A value near to zero for the correlation coefficient shows a little or no correlation. The correlation coefficient for cipher image and original image has been shown in table 2. And the corresponding scatter diagrams are shown in fig. 2.

4.2.3 Differential Attacks
The sensitivity of the algorithm against slight changes is a key issue that should be kept in mind while designing the encryption algorithm. For a slight change in the original image the extent to which the encrypted image changes represents the strength of the encryption algorithm against differential attacks. Pixel change rate measures the change in the entire encrypted image corresponding to the change in the single pixel value in the original image. Let \( E_1 \) and \( E_2 \) be the two encrypted images of the original images which vary only by difference of one pixel value. Then define Number of Pixel Change Rate (NPCR) as in (7)

\[ NPCR = \frac{\sum_{i,j} S(i,j) \times 100}{W \times H} \]  

\[ (7) \]

Where \( W \)=Width of the image, \( H \)=Height of the image

\[ S (i, j) = \begin{cases} 1 & E_1 (i, j) \neq E_2 (i, j) \\ 0 & \text{otherwise} \end{cases} \]

NPCR values of Lena.tif and Pepper.png is found out to be 99.03% and 99.01% respectively, which is consistent with the results of the algorithm.

5. CONCLUSION
A symmetric image encryption algorithm based upon DNA S-Box and chaotic sequence is proposed in this paper. First, the unique DNA based s-box performs the substitution on the DNA encoded image and then confusion is achieved by shuffling rows and columns of cipher. The simulation of the algorithm shows that it is has faster execution time than the algorithms proposed in [9, 10]. The proposed technique is analyzed in terms of brute-force attack, key space analysis, statistical analysis and satisfactory results have been found.
Further, in future the technique can be tested on various other attacks. It can also explore the use of dynamic S-box for better computing security.

6. REFERENCES


