In this work, we present a simple algorithm to design \( n \times n \) bits substitution boxes (S-boxes) based on chaotic time series of the logistic map for different carrying capacities. The use of different carrying capacities in the chaotic map leads to low computational complexity, which is desirable to get high-speed communication systems. We generate a main sequence by means of two auxiliary sequences with uniform distribution via the logistic map for different carrying capacities. The elements of the main sequence are useful for generating the elements of an S-box. The auxiliary sequences are generated by considering lag time chaotic series; this helps to hide the chaotic map used. The U-shape distribution of logistic chaotic map is also avoided, in contrast with common chaos-based schemes without considering lag time chaotic series, and uncorrelated S-box elements are obtained. The proposed algorithm guarantees the generation of strong S-boxes that fulfill the following criteria: bijection, nonlinearity, strict avalanche criterion, output bit independence criterion, criterion of equiprobable input/output XOR distribution, and maximum expected linear probability. Finally, an application premised on polyalphabetic ciphers principle is developed to obtain a uniform distribution of the plaintext via dynamical S-boxes.

1. Introduction

Nowadays, we are in the era of informatics, and due to a large number of attacks, it is important to adequately protect the information to be transmitted and avoid the possible misuse of it. The aforementioned comment motivates the generation of different approaches to have secure cryptographic systems. In general, cryptosystems can be divided into two classes: stream cipher and block cipher. A stream cipher takes one bit and transforms it into one output bit. Meanwhile, a block cipher takes \( m \) input bits and transforms them into \( m \) output bits. The core of this transformation in block ciphers is static S-boxes. The S-boxes give cryptosystems the confusion property described by Shannon [1] and are used in conventional block ciphers such as the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). The security in these cryptographic systems depends mainly on the properties of S-boxes used. A strong S-box fulfills the following criteria: bijection, nonlinearity, strict avalanche criterion (SAC), and the output bit independence criterion (BIC) [2]. If an S-box fulfills the above criteria, then it is called “good S-box.” Other desirable characteristics are being resistant to linear and differential cryptanalysis attacks. The construction of cryptographically secure dynamic S-boxes is a field of interest in the area of cryptography.

In recent years, many papers have been reported and focus on studying cryptosystems based on chaos, see ref. [3–16], this is, because of the relationships that exists between the chaotic system properties and the cryptosystem properties. In ref. [17] the relationship between these properties are given; for instance, confusion is related to ergodicity, the diffusion property with sensitivity to initial conditions and the deterministic dynamic with the deterministic pseudorandomness. Taking advantage of the chaotic system’s properties, a strong and dynamic S-Box is proposed that fulfills the criteria of a good S-box.
Regarding the generation of S-box based on chaos, some algorithms have been developed using discrete dynamical systems. For example, in ref. [3, 5, 8–10, 16], the generation of substitution boxes was introduced through using only a time series of a map or by combining two time series of different maps. In the same way, there are algorithms that use continuous chaotic dynamical systems [4, 6, 11, 12]. Nevertheless, these algorithms do not guarantee that the series used have a uniform distribution, in contrast with the approach used in this work. There are also algorithms that consider the mixing of time series of continuous and discrete dynamical systems [7, 13], and in ref. [18] the algorithm is built via time-delay series. Other types of encryption algorithms consider so-called hidden attractors [15]. The rest of the work is organized as follows: in Section 2, an algorithm for generating substitution boxes based on binary sequences of the logistic map is presented. In Section 3, a dynamic analysis of the extended logistic map and the proposed scheme to generate a dynamic S-box based on the map are presented. In Section 4, the criteria for a “good” $n \times n$-bits S-box are described. In Section 5, the performance analysis of the proposed algorithms is provided and compared with other S-boxes reported in the literature. In Section 6, an application to obtain a uniform distribution of the plaintext via dynamical S-boxes is presented. Finally, conclusions are drawn in Section 7.

2. Dynamical S-Boxes Based on CSPRNG

First, we consider an algorithm to generate dynamic S-boxes based on a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG), which was proposed by García-Martínez and Campos-Cantón [20] and tested with the suite of NIST. The CSPRNG is based on two lag time series generated with the logistic map, $f_L$: $I \rightarrow I$, which is defined as

$$f_L(x_i) = \alpha x_i (1 - x_i),$$

where $x$ is the state variable of the logistic map, and $\alpha$ is the parameter of the system.

2.1. The First Algorithm for S-Box Design via CSPRNG

The steps of the algorithm are simple as shown below:

Step 1. Select initial conditions $x_{0i}$ and $x_{0j}$ for CSPRNG in order to generate the stream of bits $s_{0i}, s_{1}, s_{2}, \ldots$

Step 2. Generate the block sequence of $n$-bits each, $C_0 = (s_{00}, s_{1}, \ldots, s_{n-1}), C_t = (s_{tn+1}, s_{tn+2}, \ldots, s_{2tn-1}), C_2 = (s_{2n+1}, s_{2n+2}, \ldots, s_{3n-1}), \ldots$

Step 3. Convert the blocks $C_0, C_1, C_2, \ldots$ of $n$-bits to integer numbers $D_0, D_1, D_2, \ldots$

Step 4. Discard the repeated elements $D_i$’s to select $2^n$ different values. The rule to discard an element is as follows: if $D_i = D_j$ with $i < j$ then discard $D_j$.

Step 5. Create the S-Box with the $2^n$ different elements of $D_i$’s.

Once the procedure is over, the proposed algorithm returns a $n \times n$-bits S-box with distinct $2^n$ values. Note that $D_0$ is the first element of the S-box, but the second element could be not $D_1$ if $D_0 = D_1$. However, enough $2^n$ elements have been generated to build the S-box. Each block $C$’s is comprised by $n$-bits, $s_i, s_{i+1}, \ldots, s_{i+n-1}$, which are related with the functions $f_0$ with $i = 1, \ldots, n$. 

where $x$ is the state variable of the logistic map, and $\alpha$ is the parameter of the system.
To exemplify the generation of dynamic S-boxes, consider \( n = 8 \), this allows us to build a 8×8 bits S-Box. The other parameters are set as follows: \( a_1 = 4 \), \( a_2 = -2 \) and two arbitrary initial conditions \( x_{01} \) and \( x_{02} \); then the S-Box obtained is shown in Table 1 for \( x_{01} = 0.8147 \), and \( x_{02} = 0.9058 \). This proposed substitution box has the properties of confusion and diffusion, which are of vital importance for the block ciphers. In this case, the time needed to calculate an S-box was 1.096 seconds.

3. Analysis of the Extended Logistic Map

In the context of mathematics, it is possible to consider the logistic map at different intervals, \([0, \kappa]\), with \( 0 < \kappa \in \mathbb{R} \), and the parameter \( \alpha \) positive and restricted to an interval such that the orbit of an initial condition \( x_0 \in [0, \kappa] \) does not escape to infinity. In this context, the logistic map for \( 1 < \kappa \) is called extended logistic map. For our interest, we consider the extended logistic map as \( f_{LE}(\alpha, x) : [0, 2^2] \rightarrow [0, 2^2] \), so that it is defined as follows:

\[
f_{LE}(\alpha, x) = \alpha x(2^2 - x), \quad n \in \mathbb{Z}^+, \tag{2}
\]

for the bifurcation parameter \( \alpha \in [0, (4/2^2)] \) and \( x_0 \in [0, 2^2] \).

Nevertheless, as it is explained in [21], mathematically, it is possible to consider negative values. As mentioned above, the extended logistic map is now studied with \( \alpha \) in the interval \((-2/2^2, 0)\). The dynamical system (2) has one or two fixed points located at \( x_1^* = 0 \) and \( x_2^* = 2^2 \alpha - 1/\alpha \), for \( \alpha \neq 0 \).

For cryptographic purposes, the value of \( n = 8 \) is chosen because a pixel, specifically, is represented by 8 bits (2\(^8\) colors), and 1 byte = 8 bits. For \( \alpha \in \{-(2/2^2), (4/2^2)\} \), the orbits do not escape to infinity for any initial conditions in an appropriate interval determined by the value and sign of the parameter \( \alpha \). Figure 1 shows the shape of the extended logistic map for \( \alpha = -2/2^2 \) in blue triangles and for \( \alpha = 4/2^2 \) in black crosses. It can be seen that the system has one or two fixed points depending on the value of \( \alpha \) which are located at \( x_1^* = 0 \) and \( x_2^* = (2^2 \alpha - 1)/\alpha \) for \( \alpha \neq 0 \).

The stability of the fixed points is displayed in Figure 2 where a circle denotes a stable or attractive fixed point, while a cross denotes that the fixed point is unstable or repulsive. The stability of these fixed points change according to the parameter \( \alpha \), i.e., when \( |f_{LE}'(x_1^*)| < 1 \) and \( |f_{LE}'(x_2^*)| < 1 \); then the fixed points \( x_1^* \) and \( x_2^* \) are stable, respectively, and they are unstable when \( |f_{LE}'(x_1^*)| > 1 \) and \( |f_{LE}'(x_2^*)| > 1 \). The case of interest is the last, because the system presents complex behavior; this is, both fixed points are repulsive, \( |f_{LE}'(x_1^*)| = |2^2 \alpha| > 1 \) and \( |f_{LE}'(x_2^*)| = | -2^2 \alpha - 2 | > 1 \). The \( x_1^* \) fixed point is repulsive for \(-2/2^2 < \alpha < -1/2^2 \) or \(1/2^2 < \alpha \leq 4/2^2 \). On the other hand, the \( x_2^* \) fixed point is repulsive for \(-2/2^2 \leq \alpha < 1/2^2 \) but \( \alpha \neq 0 \), or \(3/2^2 < \alpha \leq 4/2^2 \). So the interesting values are \( \alpha \in \{-2/2^2, -1/2^2 \} \cup \{3/2^2, 4/2^2\}\); the fixed points are of interest because they are unstable.

To analyze the behavior of the system by means of a bifurcation diagram, which is shown in Figure 3. This diagram shows orbit’s values as a function of a parameter \( \alpha \) and the route to chaos are period-doubling bifurcations at \( \alpha = 3/2^2 \) and period-halving bifurcations at \( \alpha = -1/2^2 \). There are intervals for the parameter \( \alpha \) near to \(-2/2^2 \) and \(4/2^2 \) where the extended logistic map \( f_{LE}(\alpha, x) \) behaves chaotically.

There are several approaches to demonstrate that a system is chaotic; one of them is to prove that the dynamical system fulfills the definition given by Devaney [22] and another approach is based on the Lyapunov exponent [23, 24]. We use this last concept to prove chaos and the Lyapunov exponent of Equation (2) is shown in Figure 4. The graph of Lyapunov exponents is symmetric with respect to \( \alpha = 1/2^8 = 0.0039 \), the chaotic behavior of the extended logistic map appears for values of the parameter \( \alpha \) near \(-2/2^8 \approx -0.0078 \) and \(4/2^8 \approx 0.0156 \). The local stability of the fixed points are in accordance with the Lyapunov exponent values, for example, when \( \alpha \in (-1/2^2 \approx -0.0039, 3/2^2 \approx 0.0117) \), the orbits of the system converge at a fixed point, and when the bifurcations occur, the orbits converge at periodic orbits up to chaos appears.

The aim is to use the extended logistic map to generate a time series with uniform distribution and without evidencing the mapping used. To achieve this, there is an approach based on two chaotic time series of the extended logistic map given in [20]. Following Lyapunov exponent analysis, the values of \( \alpha \) are arbitrarily selected within the chaos region, so it is considered \( \alpha = -2/2^2 \) and \(4/2^8 \).

The extended logistic map for these parameter values is invariant in different intervals as follows:

\[
\begin{align*}
f_{-2/2^2} : [-128, 384] & \rightarrow [-128, 384], \\
f_{4/2^8} : [0, 256] & \rightarrow [0, 256].
\end{align*}
\tag{3}
\]

It is worth mentioning that the time series generated with both parameter values have a U-shape distribution.

3.1. S-Box Construction via Integer Chaotic Lag Time Series

The main idea of the proposed algorithm for the generation of dynamic S-boxes is to mix two lag time series based on the extended logistic map \( f_{LE} : I \rightarrow I \). The interval \( I \) is determined by the parameter \( \alpha \). Let \( M1 \) and \( M2 \) be two time series generated with the extended logistic map by means of the following considerations: (i) given two arbitrary initial conditions \( x_{01}, x_{02} \); such that, \( x_{01} \neq x_{02} \); (ii) two different bifurcation parameter values \( \alpha_1 \) and \( \alpha_2 \); and (iii) \( l \)-units of memory for each time series \( x_{(i-k_1)}, \ldots, x_{(i-k_1)} \), \( x_{(i-k_1)}, \ldots, x_{(i-k_2)} \), \( x_{(i-k_1)} \), \( x_{(i-k_2)} \), \( x_{i-k_1} \), \( x_{i-k_2} \). So the orbits have uniform distribution independent of the U-shape distribution of the extended logistic map. In order to illustrate the algorithm, we have chosen the bifurcation parameter values \( \alpha_1 = -2/2^8 \) and \( \alpha_2 = 4/2^8 \) for the time series \( M1 \) and \( M2 \), respectively. These parameter values ensure that system (2) has chaotic behavior in both cases; see Figure 4.

To guarantee that the generator presents good statistical properties, it is necessary to generate time series with uniform distribution and also it is desirable to eliminate the extended logistic map shape in these new time series. This is achieved by means of the number of lags involved.
Table 1: The S-box generated by proposed algorithm.

<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>46</td>
<td>150</td>
<td>174</td>
<td>220</td>
<td>26</td>
<td>233</td>
<td>224</td>
<td>148</td>
<td>170</td>
<td>143</td>
<td>247</td>
<td>225</td>
<td>212</td>
<td>90</td>
</tr>
<tr>
<td>44</td>
<td>204</td>
<td>59</td>
<td>61</td>
<td>43</td>
<td>121</td>
<td>28</td>
<td>233</td>
<td>224</td>
<td>148</td>
<td>170</td>
<td>143</td>
<td>247</td>
<td>225</td>
<td>212</td>
</tr>
<tr>
<td>216</td>
<td>184</td>
<td>81</td>
<td>213</td>
<td>161</td>
<td>69</td>
<td>89</td>
<td>199</td>
<td>140</td>
<td>38</td>
<td>239</td>
<td>48</td>
<td>163</td>
<td>193</td>
<td>147</td>
</tr>
<tr>
<td>222</td>
<td>217</td>
<td>70</td>
<td>196</td>
<td>195</td>
<td>192</td>
<td>234</td>
<td>41</td>
<td>47</td>
<td>15</td>
<td>14</td>
<td>29</td>
<td>98</td>
<td>190</td>
<td>186</td>
</tr>
<tr>
<td>242</td>
<td>51</td>
<td>60</td>
<td>87</td>
<td>24</td>
<td>104</td>
<td>189</td>
<td>55</td>
<td>118</td>
<td>111</td>
<td>231</td>
<td>120</td>
<td>226</td>
<td>7</td>
<td>141</td>
</tr>
<tr>
<td>85</td>
<td>9</td>
<td>73</td>
<td>101</td>
<td>3</td>
<td>197</td>
<td>12</td>
<td>66</td>
<td>82</td>
<td>110</td>
<td>65</td>
<td>25</td>
<td>165</td>
<td>176</td>
<td>80</td>
</tr>
<tr>
<td>125</td>
<td>31</td>
<td>218</td>
<td>74</td>
<td>68</td>
<td>52</td>
<td>149</td>
<td>95</td>
<td>182</td>
<td>19</td>
<td>112</td>
<td>5</td>
<td>136</td>
<td>79</td>
<td>214</td>
</tr>
<tr>
<td>158</td>
<td>50</td>
<td>188</td>
<td>137</td>
<td>28</td>
<td>191</td>
<td>155</td>
<td>84</td>
<td>105</td>
<td>126</td>
<td>92</td>
<td>179</td>
<td>162</td>
<td>152</td>
<td>200</td>
</tr>
<tr>
<td>171</td>
<td>142</td>
<td>240</td>
<td>203</td>
<td>88</td>
<td>160</td>
<td>32</td>
<td>202</td>
<td>99</td>
<td>18</td>
<td>100</td>
<td>97</td>
<td>145</td>
<td>53</td>
<td>194</td>
</tr>
<tr>
<td>245</td>
<td>119</td>
<td>185</td>
<td>50</td>
<td>20</td>
<td>235</td>
<td>123</td>
<td>134</td>
<td>139</td>
<td>128</td>
<td>116</td>
<td>173</td>
<td>76</td>
<td>17</td>
<td>132</td>
</tr>
<tr>
<td>83</td>
<td>168</td>
<td>57</td>
<td>56</td>
<td>223</td>
<td>30</td>
<td>91</td>
<td>4</td>
<td>22</td>
<td>122</td>
<td>102</td>
<td>221</td>
<td>208</td>
<td>131</td>
<td>71</td>
</tr>
<tr>
<td>39</td>
<td>114</td>
<td>252</td>
<td>10</td>
<td>172</td>
<td>201</td>
<td>177</td>
<td>77</td>
<td>94</td>
<td>246</td>
<td>54</td>
<td>175</td>
<td>183</td>
<td>108</td>
<td>156</td>
</tr>
<tr>
<td>219</td>
<td>210</td>
<td>40</td>
<td>130</td>
<td>113</td>
<td>153</td>
<td>13</td>
<td>166</td>
<td>58</td>
<td>23</td>
<td>253</td>
<td>215</td>
<td>238</td>
<td>33</td>
<td>198</td>
</tr>
<tr>
<td>229</td>
<td>227</td>
<td>96</td>
<td>206</td>
<td>107</td>
<td>144</td>
<td>67</td>
<td>254</td>
<td>115</td>
<td>167</td>
<td>244</td>
<td>106</td>
<td>180</td>
<td>157</td>
<td>255</td>
</tr>
<tr>
<td>307</td>
<td>243</td>
<td>228</td>
<td>187</td>
<td>49</td>
<td>78</td>
<td>251</td>
<td>37</td>
<td>62</td>
<td>1</td>
<td>205</td>
<td>117</td>
<td>29</td>
<td>178</td>
<td>75</td>
</tr>
<tr>
<td>11</td>
<td>250</td>
<td>146</td>
<td>6</td>
<td>151</td>
<td>69</td>
<td>138</td>
<td>133</td>
<td>72</td>
<td>232</td>
<td>211</td>
<td>127</td>
<td>159</td>
<td>63</td>
<td>154</td>
</tr>
</tbody>
</table>

Figure 1: The extended logistic map for different $\alpha$ values: $4/2^8$ (black crosses) and $-2/2^8$ (blue triangles).

Figure 2: Fixed points stability where an asterisk and a circle denote repulsive and attracting fixed points, respectively. The extended logistic map have both repulsive fixed points at $\alpha \in [-2/2^8 \approx -0.0078, -1/2^8 \approx -0.0039] \cup [3/2^8 \approx 0.0117, 4/2^8 \approx 0.0156].$

Figure 3: Bifurcation diagram for the extended logistic map given by (2). The bifurcations occur for $x_1^*$ at $\alpha = \pm 1/2^8 \approx \pm 0.0039$, and for $x_2^*$, the bifurcations occur at $\alpha = 1/2^8 \approx 0.0039$ and at $3/2^8 \approx 0.0117$.

Figure 4: Lyapunov exponent as a function of parameter $\alpha$. 
For example, for the time series $M2 = m_{02}, m_{12}, m_{22}, \ldots$, if we consider two memory units and $\alpha_2 = 4/2^8$, the elements $m_{12}$ are given as follows:

$$m_{12} = M2(x_{(i-k1)2}, x_{i2}) = x_{(i-k1)2} + x_{i2}, \mod 2^8,$$

(4)

where $k_1 = 5$. Figure 5 shows the plot of $m_{(n-1)2}$ against $m_{n2}$, now many points are outside the curve of the extended logistic map. But it is still possible to distinguish the shape of the extended logistic map. The number of iterations considered in the delay does not matter to observe the shape of the extended logistic map and the points never spread enough on the plane $(m_{(n-1)2}, m_{n2})$, so it is necessary to consider more memory units.

Therefore, we consider three memory units, the elements $m_{12}$ of the time series $M2$ are given as follows:

$$m_{12} = M2(x_{(i-k2)2}, x_{(i-k1)2}, x_{i2})$$

(5)

where $k_1 = 10$ and $k_2 = 5$. Now, there are too many points outside the curve of the extended logistic map that look like a cloud of points on the plane $(m_{(n-1)2}, m_{n2})$; see Figure 6. Also the shape of the extended logistic map almost disappears, so three memory units are enough. The problem of considering more memory units has a computational price of information storage. For this reason, two delays, $k_1, k_2$, and the present state of the time series of the extended logistic map are used. Also the lags must not be contiguous in order to avoid regular patterns which directly affect the data distribution.

As a summary, for two given orbits $x_{01}, x_{11}, x_{21}, \ldots$, and $x_{02}, x_{12}, x_{22}, \ldots$, we consider different delays, $k_2 = k_2 = 10, k_1 = 5$ and $k_3 = 6$, to generate both time series $M1$ and $M2$. Then, the series $M1$ is conformed by the sum of two delay states $x_{i-101}, x_{i-51}$, and the actual state $x_i$. In the same way for $M2$, $x_{(i-102)}, x_{(i-62)}$ and $x_i$. The values of the time series are limited by the operation $\mod 2^8$, this guarantees that $M1, M2 \in [0, 2^8) \subset \mathbb{R}$. Explicitly $M1(x_{(i-101)}, x_{(i-51)}, x_{i1})$ and $M2(x_{(i-102)}, x_{(i-62)}, x_{i2})$ are expressed in the following way:

$$m_{11} = M1(x_{(i-101)}, x_{(i-51)}, x_{i1})$$

(6)

$$= x_{(i-101)} + x_{(i-51)} + x_{i1}, \mod 2^8,$$

$$m_{12} = M2(x_{(i-102)}, x_{(i-62)}, x_{i2})$$

(7)

$$= x_{(i-102)} + x_{(i-62)} + x_{i2}, \mod 2^8.$$

Finally, these time series $M1 = m_{01}, m_{11}, m_{21}, \ldots$ and $M2 = m_{02}, m_{12}, m_{22}, \ldots$ given by (6) and (7), respectively, are mixed and the operation mod $2^8$ is applied again, this process generates a new time series $Z_i$ given as follows:

$$Z_i = m_{i1} + m_{i2}, \mod 2^8.$$

(8)

Note that $Z_i \in [0, 2^8) \subset \mathbb{R}$. Equations (2), (6)–(8) define a delayed map that hides the structure of the chaotic map used.

For instance, the plot of the time series $x_n$ onto the plane $(x_{n-1}, x_n)$ reveals the map used; see Figure 7(a), i.e., the extended logistic map is shown. In contrast, the time series $z_n$ does not reveal the extended logistic map onto the plane $(z_{n-1}, z_n)$; see Figure 7(b). In addition, the delayed map has a uniform probability distribution instead of a “U-shaped” probability distribution [25] that the extended logistic map has. Figures 7(b) and 7(d) show the probability distribution of the time series $x_n$ and $z_n$, respectively. This is an important characteristic that makes easier the construction of S-box since all values have the same probability of occurrence in contrast to the use of a single time series.

Because the elements of the time series $Z_i$ are real numbers, they are discretized to obtain a time series that is useful for cryptosystems. The symbolic dynamics of $Z_i$ time series is given by using the floor function to obtain the series $s_i$. So the elements of $s$ are integer numbers, $s_i(Z_i) \in \{0, 1, 2, \ldots, 2^8\}$; thus, the process for getting the integer number series is as follows:

$$s_i = \lfloor Z_i \rfloor.$$

(9)
A CSPRNG based on a discrete dynamical system is given from equations (2) to (9).

### 3.2. The Second Algorithm for S-Box Design via Extended Logistic Map

In this subsection, we introduce an algorithm to generate \( n \times n \) S-boxes based on the extended logistic map. The algorithm steps are simple as shown below.

**Step 1.** Select initial conditions \( x_{01}, x_{02} \), and use equations (2) to (9) to generate the stream of byte sequence \( s_0, s_1, s_2, \ldots \).

**Step 2.** Discard the repeated elements \( D \)'s to select \( 2^n \) different values. The rule to discard an element is as follows: if \( s_i = s_j \) with \( i < j \) then discard \( s_j \).

**Step 3.** Create the S-Box with the \( 2^n \) different elements of \( s_i \)’s.

Once the procedure is finished, the proposed algorithm returns a \( n \times n \) S-box with \( 2^n \) different values. Note that \( s_0 \) is the first element of the S-box, but the second element could be not \( s_1 \) if \( s_0 = s_1 \). However, enough \( 2^n \) elements have been generated to build the S-box.

For example, if \( n = 8, x_{01} = 191, x_{02} = 209, a_1 = 4/2^8 \), and \( a_2 = -2/2^8 \), then the \( 8 \times 8 \) S-Box is obtained and shown in Table 2. This proposed substitution box has the confusion and diffusion properties, which are vital for the block ciphers. In this case, the time needed to calculate an S-box is 0.014 seconds.

### 4. Criteria for a Good \( n \times n \) S-Box

A compilation of six important and well-known criteria reported in the literature to generate cryptographically good S-boxes is presented. These criteria are bijection; nonlinearity, strict avalanche criterion, independence criterion of output bits, XOR distribution of equiprobable input/output, and maximum expected linear probability.

#### 4.1. Bijective Criterion

Let \( S(x) \) be an S-box, which is bijective if and only if their Boolean functions \( f_i \) satisfy the following condition:

\[
wt\left(a_1 \cdot f_1 \oplus a_2 \cdot f_2 \oplus \cdots \oplus a_n \cdot f_n\right) = 2^{n-1},
\]

Figure 7: (a) The extended logistic map given by \( x_n \) against \( x_{n-1} \); (b) “U-shaped” probability distribution of the extended logistic map; (c) Delayed map given by \( z_n \) against \( z_{n-1} \); (d) Uniform probability distribution of the delayed map.
where $a_i \in \mathbb{F}$, $(a_1, a_2, \ldots, a_n) \neq (0, 0, \ldots, 0)$ and $wt(\cdot)$ is the Hamming weight [2, 26], the corresponding S-box is guaranteed to be bijective.

### 4.2. Nonlinearity Criterion

**Definition 1** (see [27]). The nonlinearity of a Boolean function $f : \mathbb{F}^n \rightarrow \mathbb{F}$ is denoted by

$$N_f = \min_{l \in A_{w,c}} d_H(f, l),$$

where $A_{w,c}(x)$ is an affine function set and $d_H(f, l)$ is the Hamming distance between $f$ and $l$.

The minimum distance between two Boolean functions can be described by means of the Walsh spectrum [28]:

$$\min_{l \in A_{w,c}} d_H(f, l) = 2^{n-1} \left(1 - 2^{-n} \max_{\omega \in \mathbb{F}^n} \left| \hat{S}_f(\omega) \right| \right),$$

where the Walsh spectrum of $f(x)$ is defined as follows:

$$\hat{S}_f(\omega) = \langle F(f), \omega \rangle = \sum_{x \in \mathbb{F}^n} (-1)^{f(x) \cdot x \cdot \omega},$$

with $\omega \in \mathbb{F}^n$, and $x \cdot \omega$ is the dot product between $x$ and $\omega$ as $x \cdot \omega = x_1 \cdot \omega_1 + \cdots + x_n \cdot \omega_n$.

### 4.3. Strict Avalanche Criterion (SAC).

This criterion was first introduced by Webster and Tavares [29]. A Boolean function $f$ satisfies SAC if complementing any single input bit changes the output bit with the probability of half. So, a Boolean function $f$ satisfies SAC, if and only if

$$\sum_{x \in \mathbb{F}^n} f(x) \oplus f(x \oplus e_i) = 2^{n-1}, \quad \forall i: 1 \leq i \leq n,$$

where $e_i \in \mathbb{F}^n$ such that $wt(e_i) = 1$.

### 4.4. Output Bits Independence Criterion (BIC).

Output Bit Independence Criterion was also introduced by Webster and Tavares [29]. It means that all the avalanche variables should be pairwise independent for a given set of avalanche vectors generated by complementing a single plaintext bit.

Adam and Tavares introduced another method to measure the BIC, for Boolean functions, $f_i$ and $f_j$ ($i \neq j$) of two output bits in a S-box, if $f_i \oplus f_j$ is highly nonlinear and comes as close as possible to satisfy SAC [2]. Additionally, $f_i \oplus f_j$ can be tested with a Dynamic Distance (DD). The DD of a function $f$ can be defined as

$$DD(f) = \max_{d \in \mathbb{F}^n} \frac{1}{2} \left| 2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus d) \right|.$$  \hspace{1cm} (16)

If the value of DD is a small integer and close to zero, the function $f$ satisfies the SAC.

### 4.5. Criterion of Equiprobable Input/Output XOR Distribution.

Biham and Shamir [30] introduced differential cryptanalysis which attacks S-boxes faster than brute-force attack. It is desirable for an S-box to have differential uniformity. This can be measured by the maximum expected differential probability (MEDP). Differential probability for a given map $S$ can be calculated by measuring differential resistance and is defined as follows:

$$DP_f = \max_{\Delta x \neq 0, \Delta y} \left( \frac{\#\{x \in \mathbb{F}^n | S(x \oplus \Delta x) = \Delta y\}}{2^m} \right).$$

where $2^n$ is the cardinality of all the possible input values $(x)$, $\Delta x$ and $\Delta y$ are called input and output differences, respectively, for the $S$. Thus, the smaller value of $DP_f$ gives better cryptographic property, i.e., better resistance to differential cryptanalysis.

### 4.6. Maximum Expected Linear Probability.

The Maximum Expected Linear Probability (MELP) is the maximum value of the unbalance of an event. Two randomly selected masks $a$ and $b$ are given, where $a$ is used to calculate the mask of all possible values of an input $x$, and $b$ is used to calculate the mask of the output values of the corresponding S-box. The parity of the input bits mask $a$ is equal to the parity of the

<table>
<thead>
<tr>
<th>Table 2: The S-box generated by proposed algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
<tr>
<td>85</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>78</td>
</tr>
<tr>
<td>51</td>
</tr>
<tr>
<td>106</td>
</tr>
<tr>
<td>36</td>
</tr>
<tr>
<td>192</td>
</tr>
<tr>
<td>168</td>
</tr>
<tr>
<td>108</td>
</tr>
<tr>
<td>143</td>
</tr>
<tr>
<td>161</td>
</tr>
<tr>
<td>150</td>
</tr>
<tr>
<td>86</td>
</tr>
<tr>
<td>179</td>
</tr>
</tbody>
</table>
output bits the mask $b$. MELP of a given S-box can be computed by the following equation:

$$LP_f = \max_{a \in F_2^n \setminus \{0\}} \left( 2^{-n} \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x + f(x)} \right)^2,$$

(18)

The closer the MELP is to zero, the greater the resistance against linear cryptanalysis.

5. Performance Analysis of the Generated S-Box

In this section, we analyze the proposed S-boxes based on the second algorithm with the six important cryptographic criteria. Furthermore, our results are compared with results reported in the literature using other approaches.

5.1. Bijective Criterion. The computed value of proposed S-box is one-to-one, surjective, and balanced. It indicates the randomness of an S-box when an input has a change. The generated S-box is given in Table 3. We obtain a maximum SAC equal to 0.6094, the minimum is 0.4219, and its average value 0.5059 is close to the desired value 0.5. With these results, we conclude that the S-box generated by the proposed method fulfills the property of SAC.

5.2. Nonlinearity Criterion. Nonlinearity criterion ensures that an S-box is not a linear function between input vectors and output vectors. The nonlinearity gives the degree of dissimilarity between the Boolean function $f$ and n-bit linear function $l$. If the function has high minimum Hamming distance is said to have high nonlinearity, i.e., by reducing the Walsh spectrum in (12). An S-box contains $2^n$ functions and the nonlinearity of each Boolean function must be calculated. The nonlinearity of the proposed S-box are 96, 104, 106, 102, 104, 102, 108, and 96. High nonlinearity ensures the strongest ability to resist powerful modern attacks such as linear cryptanalysis.

5.3. Strict Avalanche Criterion (SAC). The avalanche effect indicates the randomness of an S-box when an input has a change. The generated S-box is given in Table 3. We obtain a maximum SAC equal to 0.6094, the minimum is 0.4219, and its average value 0.5059 is close to the desired value 0.5. With these results, we conclude that the S-box generated by the proposed method fulfills the property of SAC.

5.4. Output Bits Independence Criterion (BIC). The BIC criterion guarantees that there is no statistic pattern or dependency between output vectors. The obtained BIC results are shown in Tables 4, 5, and 6.

The mean value of BIC-nonlinearity is 103.50, the mean value of BIC-SAC is 0.5050, and maximum value of DD is 12 which indicates that S-box satisfies the BIC criterion.

5.5. Criterion of Equiprobable Input/Output XOR Distribution. The equiprobable input/output XOR Distribution analyzes the effect in particular differences of input pairs of the resultant output pairs to discover the key bits. The idea is to find the high probability difference pairs for an S-Box under attack. In Table 7, the maximal value of the
The computational time depends on the algorithm and the computer used to simulate the algorithm. For example, algorithms based on chaos need more steps for generating one iteration if the chaotic systems are based on ordinary differential equations (ODEs) instead of mappings. References [4, 6, 15, 36, 39] are based on ODEs which imply more computational operations to solve the systems than references [2, 3, 9, 10, 35, 38, 40] based on iterated maps. It would be unfair to compare the computational time between these algorithms that all of them generate good S-boxes. Our two proposals are based on iterated logistic maps, the first proposal maps the interval $[0, 1)$ and second proposal maps $[0, 2^8]$. One of the characteristics of the second proposal (extended logistic map) is the use of fewer iterations to generate S-boxes than the first proposal (logistic map). Fewer iterations are possible because the extended logistic map employs the interval $[0, 2^8]$ instead of the logistic map $[0, 1)$. The use of the interval $[0, 2^8]$ allows to glimpse that the algorithms reported in the literature can be scaled to operate in the interval $[0, 2^8]$ to generate less iterations to build good S-boxes.

### 6. Dynamical Generation of S-Boxes and its Application

The Alberti cipher was one of the first polyalphabetic ciphers where the principle is substitution, using multiple substitution alphabets such that the output has a uniform distribution. Nowadays, the Alberti cipher is considering a codification instead of a cipher. Taking this idea of
“polyalphabetic ciphers,” we present an application of dynamical S-boxes, where a particular intensity of a pixel is substituted by different intensities in the same round. Usually, an S-box is used to substitute all the pixels of an image of size $p \times q$ in the same way. The idea of “polyalphabetic ciphers” is to use a dynamical S-box to achieve uniform distribution in the encoded image.

Our encoded approach, to get a uniform distribution, is given by applying dynamical S-box which changes in each pixel row. The codification input is the grayscale Lenna image (Figure 8(a)), and the process is to substitute pixels of a row according to an S-box, but different rows use different S-boxes. The codification output is shown in Figure 8(c), and Figures 8(b) and 8(d) show the image histograms of plain image of Lenna and encoded Lenna image, respectively.

Furthermore, in cryptography, a uniform distribution is always desired; since this property was achieved by simple substitution with the S-boxes, a good result is expected for a full cryptographic algorithm based on these S-boxes.

It is well known that many ciphers have been successfully analyzed with the help of statistical analysis and several statistical attacks have been devised on them. Therefore, an effective cipher should be robust against any statistical attack, for instance, the information entropy, the correlation of two adjacent pixels, Peak Signal to Noise Ratio (PSNR), Unified Average Changing Intensity (UACI), and others, which it is not the purpose of this article.

It is important to point out that this substitution is a simple and useful approach intended to catch a glimpse of possible applications of dynamical S-boxes presented in this assignment.

### 7. Concluding Remarks

In this work, simple algorithms to design $n \times n$-bits substitution boxes are presented. The algorithms are based on two lag time chaotic series of the logistic map and the extended logistic map. In both approaches, two lag time series are generated by considering different carrying capacity parameter values. The mixing of these lag time series favors two things: a uniform distribution, and the concealment of the chaotic map used.

Two proposals were presented, the former generates bits and the latter generates bytes. The generation of bytes instead of bits helps to generate S-boxes with less iterations. Although, for this work, we use the extended logistic map, it is possible to employ the logistic map or different chaotic maps to generate bytes.

To evaluate the performance of the proposed S-boxes, several statistical tests were carried out. The numerical
analysis results for both proposed algorithms show that all good S-box criteria were fulfilled with high immunity to resist differential cryptanalysis and linear cryptanalysis. The number of operations is considerably reduced when we generate bytes instead of bits. The Lyapunov exponent, the bifurcations, and the local stability of logistic map are preserved in the extended logistic map. We contrast the performance of our S-boxes with other S-boxes reported in the literature and our results are in the average of good S-boxes. Finally, a simple and useful S-box application approach for coding that provides a uniform distribution was presented.

Data Availability
The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
B.B.C.Q. received a CONACYT scholarship under registration 262247 through the project supported by the Sectoral Fund for Research for Education, CONACYT-SEP, under grant number A1-S-30433. E.C.C. acknowledges CONACYT for the financial support through Project No. A1-S-30433.

References


