A Speech Encryption based on Chaotic Maps

Saad Najim Al Saad
Associate professor, Department of Computer Science, University of Al-Mustansiriyah
Baghdad, Iraqi

Eman Hato
Department of Computer Science, University of Al-Mustansiriyah
Baghdad, Iraqi

ABSTRACT

In this paper a new speech encryption system is presented. It is based on permutation and substitution of speech samples using secret keys in time and transform domains. The system is with multilevel to increase the security and to present an encrypted signal with low residual intelligibility. The logistic map is employed in keys generation to generate permutation and mask keys to be used in the permutation and substitution process. In order to maximize the benefits of the permutation process for the system, Arnold cat map is applied to permute the samples in the last level of encryption system. Simulations results are presented in the paper indicate that the encryption system provides encryption speech signal of low residual intelligibility, key sensitivity and high quality recovered signal. Total key space for the proposed encryption system is \(2^{348}\), which is large enough to protect the encryption signal against brute-force attack.

Keywords

Speech encryption, Logistic map, Arnold cat map, Permutation, Substitution, Residual intelligibility.

1. INTRODUCTION

The Speech communication is in close relation with daily life, such as education, commerce, politics, military, e-learning, phone banking and news telecasting. With the advancement of modern telecommunication and multimedia technologies, a huge amount of sensitive speech data travels in a daily routine over the open and shared networks. In order to keep security, sensitive data need to be protected before transmission or distribution. To protect speech when it is transmitted through any insecure channel, certain cryptograph techniques are needed to convert the intelligible speech to unintelligible form before transmitting (encryption). There are two types of speech encryption: digital and analog (scramble) [1].

Digital speech encryption produces digital speech encrypted by digital cryptosystems such as the Advanced Encryption Standard (AES) and Data Encryption Standard (DES). While speech scrambling algorithm involves permutation of the speech segments in time, frequency, time–frequency domain or permutation of transform coefficient of each speech block [1, 2].

Digital encryption is more secure than analog, but it needs a complex implementation and a large bandwidth for transmission. Therefore, in the case of limited bandwidth channels, analog scramblers are better [3].

The conventional cryptographic techniques are efficient for the text data. But they computationally fail in providing ample security due to the bulk data capacity and high redundancy of speech data. Therefore, the design of efficient speech security methods demands new challenges which can provide high security to the speech data. The chaos-based techniques are considered efficient for dealing with bulky, redundant speech data. They provide fast and highly secure encryption methods [4].

The main goal of this paper is to propose a speech encryption system used with low residual intelligibility, key sensitivity, and preserving the good quality of the reconstructed speech signal by chaotic maps.

The remainder of this paper is organized as follows: In the next section a general chaotic system is discussed as an example of existing cryptosystems. In Section 3, the transforms is reviewed. The proposed speech cryptosystem is presented in Section 4. Section 5 presents the test results for the proposed cryptosystem. Finally, the concluding remarks are given in Section 6.

2. CHAOTIC SYSTEM

A chaotic system is a nonlinear deterministic dynamical system which exhibits pseudorandom behavior. The output values of chaotic systems vary depending on specific parameters and initial conditions. Different parameter values yield different periods of oscillations at the output of the systems [5].

The advantage of using chaotic system lies in its random behavior and sensitivity to initial conditions and parameter settings that makes chaotic functions very important for application in cryptography to fulfill the cryptographic properties such that confusion, diffusion and disorder. A tiny difference in the starting state and parameter setting of these systems can lead to enormous differences in the final state of the system over a few iterations. Thus, sensitivity to initial conditions makes chaotic system very attractive for pseudorandom number generators [6].

In mathematics, a function that possesses some kind of chaotic behavior is defined as a chaotic function or map [7]. Brief descriptions for two types of the chaotic maps (logistic map and Arnold cat map) that are used in this paper are introduced in the following section.

2.1 Logistic Map

One of the simplest chaotic functions that have been studied recently for cryptography applications is the logistic map. The logistic map function is expressed as [8]:

\[ x_{n+1} = r \cdot x_n \cdot (1 - x_n) \]  

(1)

Where \(x_n\) takes value in the interval (0, 1), the parameter \(r\) is a positive constant and takes values upto 4. Its value determines and explores the behavior of the logistic map. From \(r=3.57\) the iterations become totally chaotic and begin to lend themselves to the purpose of encryption [8]. Thus a larger value of parameter \(r\) is chosen to obtain a highly chaotic yet deterministic discrete-time signal.

2.2 Arnold Cat Map

Arnold cat map is a two-dimensional chaotic map described by equation [9]:
The operation of the proposed decryption system can be summarized as follows:

1. Segmentation input speech signal.
2. First processing level:
   - Generation of permutation key1.
   - Permutation the samples with the permutation key1.
3. Second processing level:
   - Generation of permutation key2.
   - Application of the DWT or DCT.
   - Permutation the coefficient of DWT with the permutation key2.
   - Application of the inverse DWT or DCT.
4. Third processing level:
   - Generation of mask key.
   - Application of XOR operation between mask key and encrypted samples.
5. Fourth processing level:
   - Convert into 2-D format.
   - Application inverse Arnold cat map on the samples in time domain.
   - Convert into 1-D format.

3. Second processing level:
   - Generation of mask key.
   - Application of XOR operation between mask key and encrypted samples.
4. Third processing level:
   - Generation of permutation key2.
   - Application of the DWT or DCT.
   - Inverse permutation the coefficient of DWT with the permutation key2.
   - Application of the Inverse DWT or DCT.
5. Fourth processing level:
   - Generation of permutation key1.
   - Inverse permutation the samples with the key permutation1.

4.1 The Permutation Keys Generation

The main requirement of any permutation key is to minimize the residual intelligibility as much as possible. So the problem of key generation is therefore an important issue in the design of a speech encryption system. The first key is generated by choosing two logistic maps and cross-coupled as shown in Figure 2. The output generated by the first logistic map is fed to the second logistic map as the input (initial condition) and vice versa.

The procedure of the generation first permutation key is provided in pseudo code 1:

**Pseudo code 1: Generation of Permutation Key**

Input:
- x: initial condition.
- r1: control parameter to first logistic map.
- r2: control parameter to second logistic map.

Output:
- key1: permutation key1.

Begin
For (i = 0 , i < keylength, i++) // put the values of logistic

\[
\begin{align*}
\text{chaotic}[i] & \leftarrow x \\
\text{chaotic}[i + 1] & \leftarrow y_2
\end{align*}
\]

For (i = 0 , i < keylength, i++)

\[
\begin{align*}
x & \leftarrow r_1 \times x \times (1 - x) \\
y & \leftarrow r_2 \times x \times (1 - x)
\end{align*}
\]

End
key1 [ i] ← j
chaotic[j] ← 1 //replace the smallest value with large
                      // value (1) to find the 2nd smallest
                      // value in chaotic array in the next iteration
End

The second key is generated from the first key by permutation the first key value according to the first key. An example of second key generation is shown in Figure 3.

The first key is used to permute the sample in time domain while the second key is used to permute the coefficients resulting from DWT or DCT in the frequency domain.

Fig 1: Structure of proposed system
4.2 Generation of Mask Key

Because the permutation process only changes the original sample’s position and the sample’s value has not been changed. Substitution process is added to change the amplitudes of samples in each block. Each sample value is changed by XOR operation with mask key value. The mask is generated from six logistic maps:

\[ x_{n+1} = r_n \cdot x_n \cdot (1 - x_n) \quad \text{for } 1 \leq n \leq 6 \]  

The output values of these logistic maps are real values in interval (0, 1). This real values sequence converted into bits (0 or 1) according the following equations:

\[ A_i = \begin{cases} 0 & \text{if } x_i < 0.5 \\ 1 & \text{if } x_i \geq 0.5 \end{cases} \quad \text{for } 1 \leq i \leq 3 \]  

\[ A_i = \begin{cases} 0 & \text{if } x_i > 0.5 \\ 1 & \text{if } x_i \leq 0.5 \end{cases} \quad \text{for } 4 \leq i \leq 6 \]  

The six \( A(i) \) are combined and mixed using XOR operations to produce eight bits according to the following equations:

\[ \text{Bit}_1 = A_1 \oplus A_5 \oplus A_3 \]  

\[ \text{Bit}_2 = A_2 \oplus A_6 \oplus A_3 \]  

\[ \text{Bit}_3 = A_3 \oplus A_2 \oplus A_6 \]  

\[ \text{Bit}_4 = A_4 \oplus A_2 \oplus A_3 \]  

\[ \text{Bit}_5 = A_5 \oplus A_1 \oplus A_3 \]  

\[ \text{Bit}_6 = A_6 \oplus A_5 \oplus A_3 \]  

\[ \text{Bit}_7 = A_4 \oplus A_1 \oplus A_3 \]  

\[ \text{Bit}_8 = A_1 \oplus A_5 \oplus A_6 \]  

The 8 Bits represented the final output from one iteration for six logistic maps. These 8-Bits are appended with the other 8 Bits are generated from next iteration. The 16 Bits are converted to integer number and XOR with the one sample in the block.

4.3 Application Arnold Cat Map

Another permutation step is performed on the samples in time domain to increase the security and prevent a cryptanalyst from discovering the plain speech. Each substituted block is permuted in the last level of the transmitter side by the Arnold cat map.

For applying Arnold cat map, the block of samples in time domain must convert from 1-D vector to 2-D matrix as illustrated in Figure 4, and then permuted by multiplying the position of each sample by Arnold cat map matrix. This process is repeated for number of iteration where the final result is a new position to sample so the samples of the block appear to be randomly rearranged. The output is resized to 1-D vector again.

5. SIMULATION RESULTS

The following sub sections are dedicated to present and discuss the results of the conducted tests to evaluate the performance of the proposed systems.

5.1 The Test Measures

The objective measures used in this work to assess the performance of the proposed system are the following: Segmental Signal-to-Noise Ratio (SNRseg), Frequency-Weighed Signal-to-Noise Ratio (fWNRseg), Log-Likelihood Ratio (LLR) and Correlation \( (r_{xy}) \) \cite{3, 12}. Tow speech signals with sampling frequency of 8 kHz and 16 bits per sample have been selected as test material.
Fig 2: Generation of permutation key 1

Fig 3: Generation of permutation key 2

Fig 4: Simulation example for application of Arnold cat map (a=1, b=1)
5.2 Key Space and Sensitivity Analysis

Key space analysis is one of the important criteria of the performance analysis of encryption system. A good encryption algorithm should have a large key space, and also should be sensitive to the key value.

5.2.1 Key Space

The key space of the encryption system should be large enough to resist the brute-force attack. It is generally accepted that a key space of size larger than $2^{128}$ is computationally secure against such attack.

In the proposed system, all initial conditions and control parameters constitute the secret key of encryption algorithm. For a $10^7$ floating point precision, all keys parameters (one initial condition, two control parameters to permutation key and six initial condition, six control parameters to mask key) can take $10^7$ possible values. Therefore, the key space comes out as $(10^7)^{35} \approx 2^{448}$, which is large enough to resist the brute-force-attack.

These results suppose a known secret control parameters of Arnold cat map ($a$ and $b$) and the number of iteration by the attacker, but really the control parameters is unknown making the search infeasible.

5.2.2 Key Sensitivity

Key sensitivity means that the encrypted signal cannot be decrypted correctly if there is any change between encryption and decryption keys. Large key sensitivity is required by all secure cryptosystems.

For testing the key sensitivity of the proposed algorithms, only one parameter of keys is changed at a time by a tiny amount, keeping all other parameters of keys unchanged. The encrypted signal is decrypted with five different keys are generated by changing only one parameter in the original secret keys. The $r_{xy}$ and LLR are estimated between each decrypted signal and the signal decrypted with the original key, and the results are tabulated in Table 1. The low $r_{xy}$ value (closer the correlation coefficient to zero) and large LLR value show the large key sensitivity of the proposed system.

It is clear from the Table 1 that very low correlation exists among the signals decrypted with tiny changed key and they are totally different from the decrypted signal with the original key.

Table 1. Test for key sensitivity

<table>
<thead>
<tr>
<th>Decryption key</th>
<th>$r_{xy}$</th>
<th>LLR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DCT</td>
<td>DWT</td>
</tr>
<tr>
<td>Key1</td>
<td>-0.01275</td>
<td>-0.002018</td>
</tr>
<tr>
<td>Key2</td>
<td>-0.002485</td>
<td>-0.001361</td>
</tr>
<tr>
<td>Key3</td>
<td>-0.003945</td>
<td>-0.011975</td>
</tr>
<tr>
<td>Key4</td>
<td>-0.08623</td>
<td>-0.092813</td>
</tr>
<tr>
<td>Key5</td>
<td>-0.06247</td>
<td>-0.07895</td>
</tr>
</tbody>
</table>

5.3 Quality of Encrypted Signal

The residual intelligibility is the accuracy with which we can hear what is being said. To measure the residual intelligibility of the encrypted signal four quality measures are used: SNRseg, fwNRseg, $r_{xy}$, and LLR measure. As the value of the LLR is increased, and the values of SNRseg, SSNRseg and $r_{xy}$, are decreased, the higher is the quality of the encrypted signal. Table 2 explains the result for proposed system (for simplicity, DCT and DWT will be used to refer to proposed cryptosystem used DCT and DWT respectively).

From Table 2 the $r_{xy}$ measure has low value that indicates low correlation between original and the encrypted signals. The LLR measure for encrypted signals is high while SNRseg and fwNRseg measures are very low (negative value) which means that no residual intelligibility and encrypted signals are very noisy.

It is worth to mention that different permutation keys generated. The results are tabulated in Tables 3 and 4 for DCT and DWT respectively and show that all keys produce encrypted signal with very low residual intelligibility, which means that all keys give good encryption results. Therefore, there is no need to search for good permutation keys and memorize them.

Figure 5 show the waveform of original signal and their encryption signal in level 2 and level 4 while Figure 6 show the spectrogram of original signal and their encryption signal in level 2 and level 4. In Figure 5 and 6 it is important to note that the first two levels in proposed system (permutation in time and transform domain) show the considerable residual intelligibility will remain. The information has not been obviously destroyed in the encrypted signal. Therefore, the substitution process is an important part in the proposed system.

5.4 Quality of Decrypted Signal

To measure the quality of decrypted signal the same four quality measures are used: SNRseg, SSNRseg, $r_{xy}$ and LLR measures. As the value of the LLR is decreased, and the values of SNRseg, fwNRseg and $r_{xy}$, are increased, the higher is the quality of the decrypted signal. Table 5 illustrates the result.

One can observe that SNRseg and SSNRseg measures in Tables 5 are very high (positive values) for all the decrypted signals while the LLR measure has a small value that indicates high precision data and very good quality of the recovered speech signals. Correlation coefficients $r_{xy}$ indicate a perfect positive correlation (closed to +1) that means high correlation between original and the decrypted signals.

Figure 7 shows the waveform and spectrogram plotting for original and the recovered signal that resulted from applying proposed system.

5.5 Effect of Noise

One believes that the effect of noise on the efficiency of the proposed algorithms is an important issue, which deserves consideration; the evaluation of the proposed encryption system with different power levels of white Gaussian noise has been tested. The SNR, SNRseg, SSNRseg and LLR measures are calculated for decrypted signal in the presence of noise at different signal to noise ratios from (5 dB up to 50 dB) values.

The relationship between SNR, SNRseg, SSNRseg and LLR values and SNR value are illustrated in Figure 8 and 9 for proposed system. From results it is clear that the quality measures values are better at high SNR values.
Table 2. Result of residual intelligibility

<table>
<thead>
<tr>
<th>Measures</th>
<th>Signal 1</th>
<th>Signal 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DCT</td>
<td>DWT</td>
</tr>
<tr>
<td>$r_{xy}$</td>
<td>-0.009221</td>
<td>-0.010604</td>
</tr>
<tr>
<td>SNRseg</td>
<td>-17.70727</td>
<td>-17.70198</td>
</tr>
<tr>
<td>fwSNRseg</td>
<td>-16.9303</td>
<td>-16.9115</td>
</tr>
<tr>
<td>LLR</td>
<td>2.9336</td>
<td>2.9365</td>
</tr>
</tbody>
</table>

Table 3. Result of residual intelligibility for (DCT)

<table>
<thead>
<tr>
<th>Key name</th>
<th>DCT</th>
<th>DWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{xy}$</td>
<td>-0.003644</td>
<td>-17.703509</td>
</tr>
<tr>
<td>Key1</td>
<td>0.010093</td>
<td>-17.654174</td>
</tr>
<tr>
<td>Key2</td>
<td>0.004625</td>
<td>-17.685894</td>
</tr>
<tr>
<td>Key3</td>
<td>0.005663</td>
<td>-17.676131</td>
</tr>
<tr>
<td>Key4</td>
<td>0.004354</td>
<td>-17.680125</td>
</tr>
<tr>
<td>Key5</td>
<td>0.000324</td>
<td>-17.680125</td>
</tr>
</tbody>
</table>

Table 4. Result of residual intelligibility for (DWT)

<table>
<thead>
<tr>
<th>Key name</th>
<th>DWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{xy}$</td>
<td>-0.004734</td>
</tr>
<tr>
<td>Key1</td>
<td>0.010093</td>
</tr>
<tr>
<td>Key2</td>
<td>0.004625</td>
</tr>
<tr>
<td>Key3</td>
<td>0.005663</td>
</tr>
<tr>
<td>Key4</td>
<td>0.004354</td>
</tr>
<tr>
<td>Key5</td>
<td>0.000324</td>
</tr>
</tbody>
</table>

Table 5. Result of recovered signals quality

<table>
<thead>
<tr>
<th>Measures</th>
<th>Signal 1</th>
<th>Signal 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DCT</td>
<td>DWT</td>
</tr>
<tr>
<td>$r_{xy}$</td>
<td>-0.003644</td>
<td>-17.703509</td>
</tr>
<tr>
<td>Key1</td>
<td>0.010093</td>
<td>-17.654174</td>
</tr>
<tr>
<td>Key2</td>
<td>0.004625</td>
<td>-17.685894</td>
</tr>
<tr>
<td>Key3</td>
<td>0.005663</td>
<td>-17.676131</td>
</tr>
<tr>
<td>Key4</td>
<td>0.004354</td>
<td>-17.680125</td>
</tr>
<tr>
<td>Key5</td>
<td>0.000324</td>
<td>-17.680125</td>
</tr>
</tbody>
</table>

Fig 5: Waveform for encrypted signal
(a) Original signal  (b) Encrypted signal after level 2  (c) Encrypted signal after level 4

Fig 6: Spectrogram for encrypted signal
(a) Original signal  (b) Encrypted signal after level 2  (c) Encrypted signal after level 4
Fig 7: Waveform and Spectrogram for decrypted signals
(a). Waveform for original signal  
(b). Waveform for decrypted signal  
(c). Spectrogram for original signal  
(d). Spectrogram for decrypted signal

Fig 8: Quality measures for (DCT)
(a). Output SNR vs. input SNR  
(b). SNRseg vs. input SNR  
(c). SSNRseg vs. input SNR  
(d). LLR vs. input SNR

Output SNR vs. input SNR

SNRseg vs. input SNR

SSNRseg vs. input SNR

LLR vs. input SNR
6. CONCLUSIONS

Some important conclusions can be drawn from this work:

1. The proposed algorithms in this work encrypt the original signal with multilevel which makes the cryptanalysis a difficult task and increases the security of the speech signal.

2. The proposed algorithms are very sensitive to the initial condition and control parameters that means the encrypted signal cannot be decrypted correctly, if there is tiny change between encryption and decryption keys. Key sensitivity indicates a high security and suitability of the proposed algorithms.

3. The key space is very large which makes a brute-force attack impracticable.

4. The results show that the proposed system returns a very good level of security and high quality of recovered speech quality.

5. The permutation process is not sufficient to remove speech silence patterns. Therefore, a masking step is very necessary in order to change the remaining non-permutated and silence portions of speech signals and that increases the security of the proposed algorithm.

6. An evaluation of the proposed algorithms with different power levels of the white Gaussian noise was tested. The results proved that as the signal to noise ratio increases, the SNR, SNRseg and SSNRseg increases and LLR decrease. From the previous tests, it can be concluded that the proposed algorithm can tolerate noise with high SNR and it can be implemented to decrypt speech signal with high efficiency.

7. REFERENCES


