


# Recent Trends in Audio Encryption

By

Miroslav Milenov Cholakov




The recent decade has witnessed a rise in digital content, especially multimedia and multimedia applications. The security requirements of these platforms need advanced encryption schemes. In this presentation some of the novelties of audio encryption are reviewed.

# New Audio Encryption Algorithm Based on Chaotic Block Cipher

This algorithm begins with reading audio file chunks in the preprocessing stage, beginning with the header of the WAV file till reaching the audio data. The audio data is left untouched so that the user can hear the scrambled audio data. It is then encrypted and decrypted in block sizes of 25x25 bytes.

Each of the blocks is input into three stages:

- Permutation stage-permits the audio block using the tent map
- Xor adding stage-Xores the resulted block with the key block
- Substitution stage-substitutes the audio block into a new block



using a substitution method based on the multiplication inverse.

The chaotic map used for the permutation of the audio blocks is the Tent map where  $\alpha \in [0,1]$  is the control parameter and  $x_i$  is the current state of the system. Tent map has uniform invariant probability density in  $[0, 1]$  interval.

$$x_{i+1} = \begin{cases} \frac{x_i}{\alpha} & \text{if } x \in [0, \alpha], \\ \frac{(1 - x_i)}{(1 - \alpha)} & \text{if } x \in (\alpha, 1] \end{cases}$$

The multiplication inverse method used in the Substitution stage is the following

**Input:**  $e \in \mathbb{Z}_n$  such that  $\gcd(e,n) = 1$   
**Output:**  $e^{-1} \pmod n$ , where  $e^{-1} = t_1$  provided that it exists

1. Set  $r_1=n, r_2=e, t_1=0, t_2=1$ .
2. While ( $r_2 > 0$ )
  - $r = r_1 - q \times r_2$ .
  - $r_1 = r_2, r_2 = r$ .
  - $t = t_1 - q \times t_2$ .
  - $t_1 = t_2, t_2 = t$ .End while
3. if  $r_1 = 1$  then  $e^{-1} = t_1$

The decryption algorithm is reverse of each stage of the audio encryption algorithm.

In reverse permutation stage the Tent map will iterate in the same way as in the encryption process, where each position determined by the Tent map will be used as an index to return the 4 bit block back to its position.

Reverse Xor stage is achieved by reverse XOR-ing the



encrypted audio block to the same key.

Finally the Reverse Substitution stage is the same as in the encryption process.

# Noise-Tolerant Audio Encryption Framework

Designed by the

## Application of $S_8$ Symmetric Group and Chaotic Systems

This algorithm is based on frequency hopping-a military technique where the transmitter switches the carrier wave frequency. Similarly this algorithm switches between different chaotic maps.

# Chaotic maps used for the cryptosystem

Chebybyshev chaotic map, where  $n$  is an integer and  $x$  has a range  $[-1, 1]$

$$x(i+1) = \text{mod}(\text{floor}(x(i) \times 1000 \times n))256 + 1.$$




TD-ERCS chaotic map is consisted of two chaotic sequences .

$$\begin{cases} x_n = -\frac{2k_{n-1}y_{n-1} + x_{n-1}(\mu^2 - k_{n-1}^2)}{\mu^2 + k_{n-1}^2} \\ k_n = \frac{2k'_{n-m} - k_{n-1} + k_{n-1}k'^2_{n-m}}{1 + 2k_{n-1}k'_{n-m} - k'^2_{n-m}} \end{cases}$$

$$k'_n = \frac{x_n}{x_n} \mu^2,$$

$$y_n = k_{n-1}(x_n - x_{n-1}) + y_{n-1},$$


$$k_{n-m} = \begin{cases} \frac{x_{n-1}}{y_{n-1}} \mu^2, & \text{if } n < m, \\ \frac{x_{n-m}}{y_{n-m}} \mu^2, & \text{if } n \geq m, \end{cases}$$



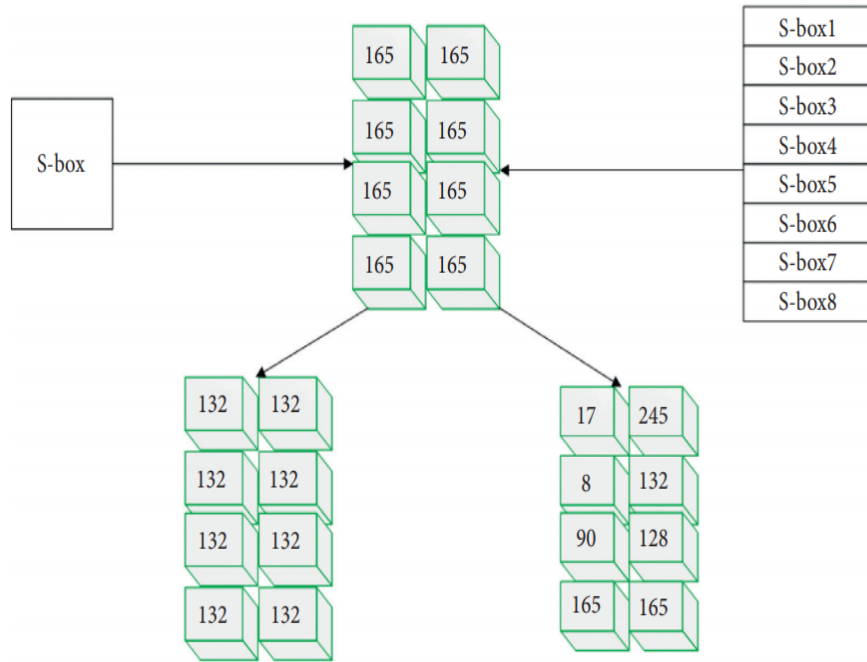
The third map used in the scheme is a one-dimensional circle chaotic map. It has a good chaotic behaviour when used on any data. It is represented with the following formula

$$x_{n+1} = \text{mod} \left( x_n + \Omega - \frac{k}{2\Pi} 2\Pi (x_n), 1 \right)$$

where  $x_n = 0.4$ ,  $\Omega = 0.5$ ,  $x_{n+1}$  is computed as mod 1, and  $k$  is constant.



In addition to the presented chaotic systems S8(S8 MEC S-Box) permutation is used to construct 40320 unique MEC S-boxes with similar strength and features. This highly improves the security of the scheme.



The table shows the transformation effect achieved with S-box.



# Audio Shuffle-Encryption Algorithm


The next algorithm uses an input audio file and a key and performs byte shuffling of the data. The file is streamed and the encryption is data and key dependent

The encryption algorithm is the following:

```
For i = 1 to k
  fixBit = Hash(Key,i)
  D = Vector where D[j] is the value of bit (fixBit) of
    the jth byte of the current stream
  S0 = Vector containing numbers of current stream
    bytes (j) that have (D[j] == 0)
  S1 = Vector containing numbers of current stream
    bytes (j) that have (D[j] == 1)
  Shuffle = Concatenation of S0 with S1
  Substitute the bytes of the current stream so that
    the new location of byte (j) is byte (Shuffle[j])
End For
```

The decryption algorithm resembles the encryption one and is performed with equal number of iterations with the difference that each one is inverted.

```
For i = k to 1 step -1
  fixBit = Hash(Key,i)
  D = Vector where D[j] is the value of bit (fixBit) of
    the jth byte of the current stream
  S0 = Vector containing numbers of current stream
    bytes (j) that have (D[j] = 0)
  S1 = Vector containing numbers of current stream
    bytes (j) that have (D[j] = 1)
  Shuffle = Concatenation of S0 with S1
  Substitute the bytes of the current stream so that
    the new location of byte number (Shuffle[j]) is
    byte number (j)
End For
```



This algorithm is less secure when the input audio quality requires one byte for each file due to the fact that mixing up the single stream preserves the values. This results in the necessity of a separate algorithm that is used in conjunction with this one.



From the figures on the right we can see that the sample audio file encrypted with the algorithm has nothing in common with its former self after the encryption

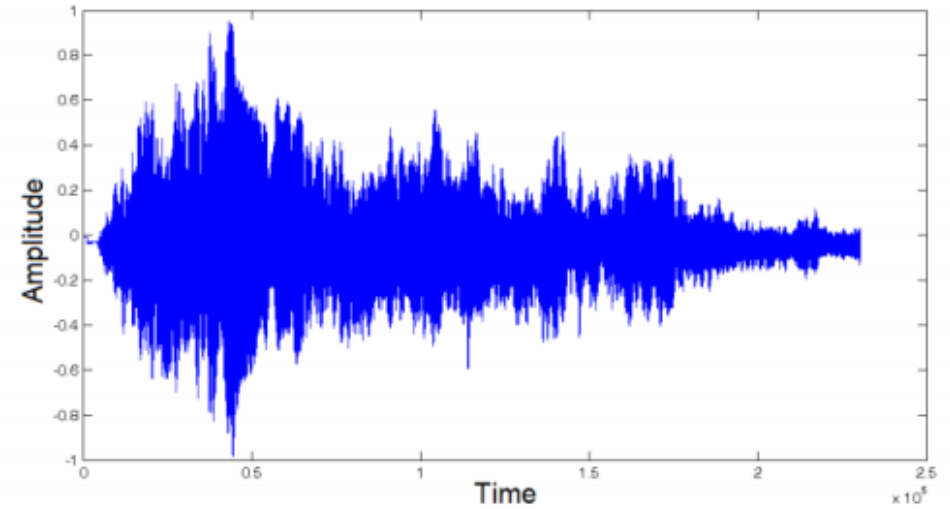
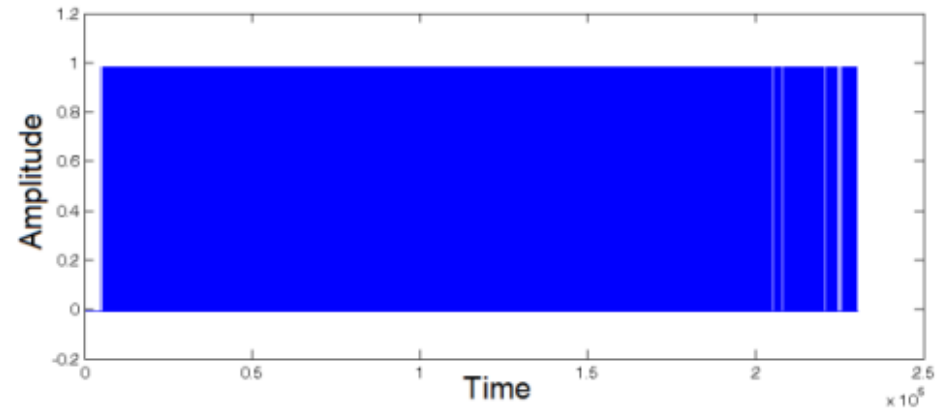


Fig. 1. Original audio for Sample.wav.

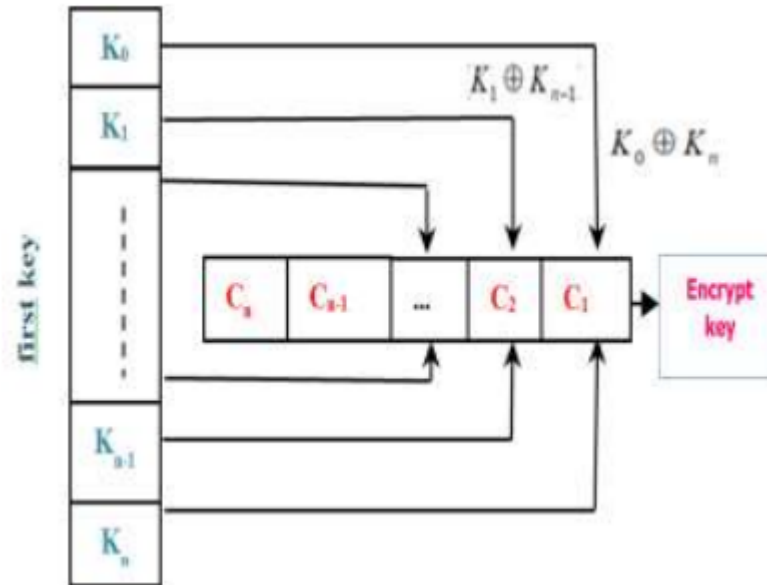



# Encryption algorithm using Speech Audio File as a key

The algorithm uses the Rijndael algorithm in addition to two password keys .It is relatively simple and safe and can be separated in the following stages:

- Generating password keys,which is consisted of the following steps:
  - entering the speech audio file
  - converting the already input file to text and saving the data in a binary file
- Creating two pass keys from a binary file produced by the first stage using a hash function

Encrypting the first key by a conversion byte to binary and creating a new key. The figure below shows the encryption diagram.




$$\text{first\_key} = C_1, C_2, C_3, \dots, C_{n-1}, C_n \dots\dots(1)$$

$$C_1 = (K_0 \oplus K_n)$$

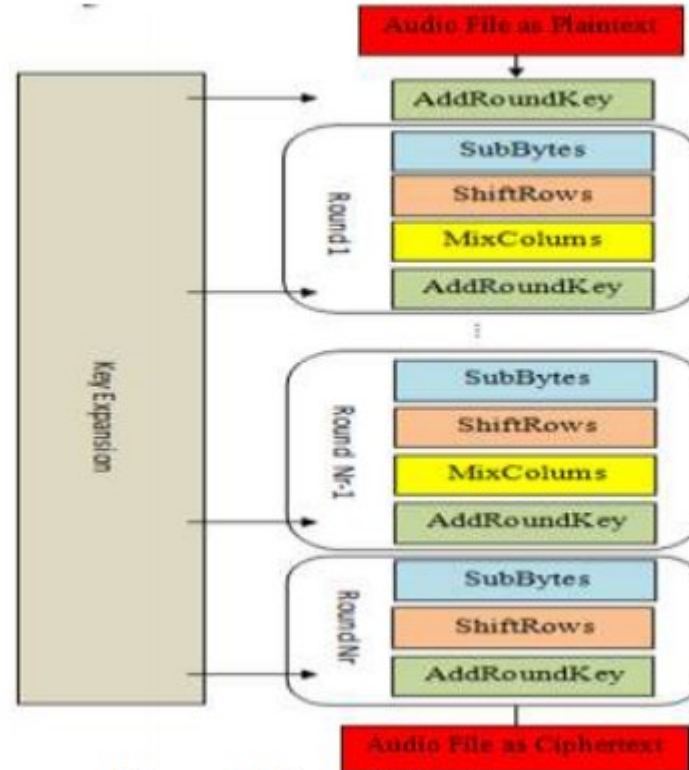
$$C_2 = (K_1 \oplus K_{n-1})$$

$$\vdots$$

$$C_{\frac{n}{2}} = (K_{\frac{n}{2}} \oplus K_{\frac{n}{2}+2})$$

Encryption scheme.

The Rijndael algorithm is shown.





The encryption and decryption processes are relatively simple and are divided into the following stages:

Encryption:

1. Entering audio file as input to the Rijandean function
2. Entering two pass keys
3. First key-bytKey- is used for the symmetric algorithm and has a size of 256 bits.
4. Second key-bytIV- has a size of 128 bits.
5. Rijandean encryption.
6. Saving the output as a .WAV file



## Decryption:

1. Inputting the .WAV file for decryption
2. Inputting the two pass keys.(bytKey and bytIV)
3. Decryption with the Rijanddeal function.
4. Saving the decrypted file which is identical to the original.

# Sample file histograms before and after encryption:

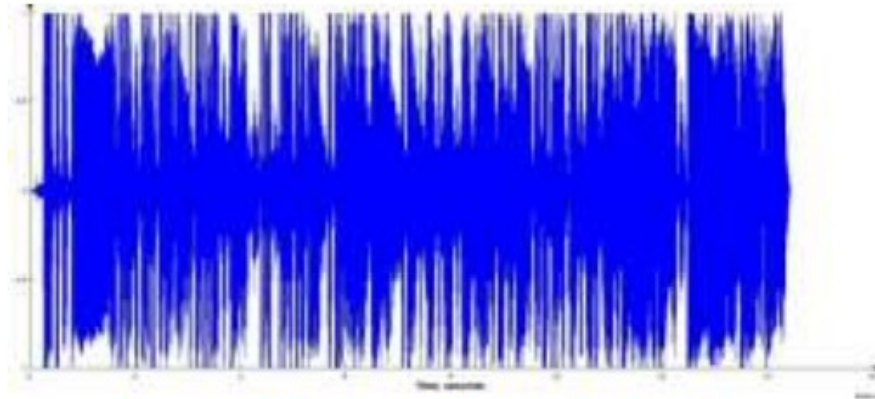


Figure (5) original tested audio file.

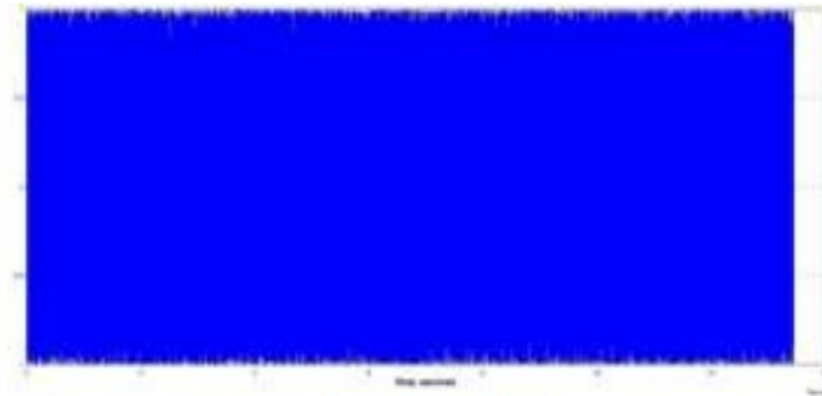


Figure (6) tested audio file after encryption.





# Conclusion

In recent years there has been a rise in digital devices usage, which has led to the need of better data protection. This leads to constant data protection evolution. Recently we see a rise in the complexity of the algorithms by incorporating of chaotic maps in audio encryption algorithms.

# References:

1. N. Hassan, F. Al-Mukhtar, E. Ali, Encrypt Audio File using Speech Audio File As a key, 2nd International Scientific Conference of Al-Ayen University (ISCAU-2020) *IOP Conf. Series: Materials Science and Engineering* 928 (2020) 032066.
2. H. Aziz, S. Gilani, I. Hussain, A. Janjua, S. Khurram, A Noise-Tolerant Audio Encryption Framework Designed by the Application of S8 Symmetric Group and Chaotic Systems. *Mathematical Problems in Engineering*, 2021, Article ID 5554707.
3. A. Tamimi, A. Abdalla, An Audio Shuffle-Encryption Algorithm, *Proceedings of the World Congress on Engineering and Computer Science 2014 Vol I WCECS 2014*, 22-24 October, 2014, San Francisco, USA.



## References continued:

4.E. A. Albahrani, "A new audio encryption algorithm based on chaotic block cipher," 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), 2017, pp. 22-27, doi: 10.1109/NTICT.2017.7976129.