Digital Sound Encryption with Logistic Map and Number Theoretic Transform

Yudi Satria, Gabe Rizky P.H. and Suryadi MT
Department of Mathematica, Faculty of Mathematica and Natural Sciences, Universitas Indonesia, Kampus UI, Depok, 16424

Email: ysatria@sci.ui.ac.id, gabe.rizky@sci.ui.ac.id, and yadi.mt@sci.ui.ac.id

Abstract. Digital sound security has limits on encrypting in Frequency Domain. Number Theoretic Transform based on field (GF $2^{521} - 1$) improve and solve that problem. The algorithm for this sound encryption is based on combination of Chaos function and Number Theoretic Transform. The Chaos function that used in this paper is Logistic Map. The trials and the simulations are conducted by using 5 different digital sound files data tester in Wave File Extension Format and simulated at least 100 times each. The key stream resulted is random with verified by 15 NIST’s randomness test. The key space formed is very big which more than $10^{465}$. The processing speed of algorithm for encryption is slightly affected by Number Theoretic Transform.

1. Introduction

Various efforts to secure digital data has been done using cryptograph application. There are two main process in cryptograph, namely encryption and decryption process. The main reliability measure of the Encryption algorithm is the durability against various attack and the processing speed.

The encryption algorithm commonly used is DES, AES, and RSA. However, because relating to their durability and or processing speed, currently many of developing encryption are based on chaos function. This is based on the fact that the usage of chaos function in encrypting has more durability against attack and or better processing speed [1, 2].

Several studies related to encryption algorithm based on chaos function in digital data security are encryption algorithm using Logistic Map [3-8], other chaos function [9-14] and non-chaotic mode [15-17]. Each of this studies has its own performance associated with the level of resulted key stream, the key space and the processing time (processing speed).

Digital data security done in previous studies are mostly done to secure digital image while security related to digital sound has not many yet explored. In this study, the improvement for algorithm encryption to secure digital sound will be developed. The developed algorithm is using chaos function, in this case Logistic Map, and combined Number Theoretic Transform (NTT). The development is conducted to provide alternative digital sound encryption algorithm which has better durability against attack and faster processing speed.

2. Research Method

Chaos function in this study will act as the random number generator which will generate a random key stream sequence. This study using Logistic Map as chaos function and combined
by NTT. The sequence of key stream will generated by the Logistic Map in this paper is using the recursive form in Equation (1) as follow [2]:

\[ x_{i+1} = rx_i(1 - x_i) \]  

Where \( x_0 \in (0,1) \), \( r \in (0,4) \) for \( i = 0,1,2,3, ... \)

Sound wave not only can be interpreted by wave super position based on the primitive wave frequency (Frequency Domain) but also can be interpreted by wave amplitude sequence (Time Domain). Randomization at Frequency Domain has limitation in discrete form due to loss of information as the consequence of transformation from continuous form to discrete form and inversely.

The transformation, which to be the bridged both Time Domain and Frequency Domain, is using Fourier Transformation variation called Number Theoretic Transform. It was based on the Fourier Transformation’s formulation [18].

\[ w = e^{-\frac{2\pi i \theta}{N}} \]  

\[ f_k = \sum_{n=0}^{N-1} t_n * w^{-k*n} \]  

\[ t_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n * w^{k*n} \]

But the different about the variation is about using the field where \( w, f, t_k \in f(\mathbb{Z}^{'}, \mathbb{Z}^{''}) \) and \( w^k \neq 1 \) and \( w^N = 1, \forall k = 1 \ldots N - 1 \).

The design of developed improvement encryption algorithm in this study is shown in Figure 1. Furthermore, the viewed performance of developed algorithm is based on the randomness analysis of randomness through the 15 NIST’s test [19], the key space and the measurement of processing speed.

**Figure 1. Encryption Algorithm**
3. Result and Analysis

Encryption algorithm developed in this study will be implemented using C and java programing language and tested using trials data as showed in Table 1. The performance of the trials and simulations result is explained in next section.

| Name of File | Size of File (MB) | Duration (Seconds) | Bit Rate (KB/s) |
|--------------|------------------|--------------------|-----------------|
| Lagu1.wav    | 2.66             | 31                 | 705             |
| Lagu2.wav    | 20.00            | 123                | 1411            |
| Lagu3.wav    | 37.90            | 225                | 1411            |
| Lagu4.wav    | 44.50            | 264                | 1411            |
| Lagu5.wav    | 798.00           | 4746               | 1411            |

3.1. Key Randomness Analysis (by 15 NIST’s test)

The result with significant level are \( \alpha = 0.01 \) on generated key stream was showed on Table 2.

### Table 2. 15 NIST’s Randomness Test Result

| STATISTICAL TEST                        | PROPORTION | P-VALUE |
|-----------------------------------------|------------|---------|
| Frequency                               | 32/32      | 0.739918|
| Block Frequency                         | 32/32      | 0.671779|
| Cumulative Sums                         | 32/32      | 0.573505|
| Runs                                    | 32/32      | 0.862344|
| Longest Run                            | 32/32      | 0.178278|
| Rank                                    | 32/32      | 0.534146|
| FFT                                     | 32/32      | 0.739918|
| Non Overlapping Template                | 31/32      | 0.436885|
| Overlapping Template                    | 32/32      | 0.468595|
| Universal                              | 31/32      | 0.862344|
| Approximate Entropy                     | 32/32      | 0.862344|
| Random Excursions                       | 23/24      | 0.267745|
| Random Excursions Variant               | 23/24      | 0.301477|
| Serial                                  | 32/32      | 0.762807|
| Linear Complexity                       | 29/32      | 0.299251|

In the table 2, all Resulting P-Value values for all 15 NIST’s tests are greater than significant level (0.01) which means passing the NIST’s test. It can be deducted the random stream which was generated by Logistic map, was random and verified by 15 NIST’s test.

3.2. Analyst key space

There are two core parameters in Logistic map function as initial value in Real domain but with modified in the field \( \text{GF}(2^{521} - 1) \) then the key space in every parameter will be same as the norm set of the field. In other word, the key space in a Logistic map which based on the field will be \((2^{521} - 1)^2 > 2^{1040} \approx 10^{313}\).

There are one core parameter in Fourier Transformation as initial value in Real Domain and it just constant and the value was depending on the size (or order or N). By using NTT over the field, the parameter still will be one core parameter but there has restriction on the order but the good point is all element can be used as the key which is better than just only using NTT over a ring like normally use in integer 8-bit, 32-bit, 64-bit or may be 1024-bit. In the other word, the key space in a NTT over the field was \((2^{521} - 1) > 2^{520} \approx 10^{156}\).
In the summary, with the combining Logistic Map and NTT, it can make the key space larger and help to reach \((2^{521} - 1)^3 > 2^{1560} \equiv 10^{469}\). The result is better than only using Logistic Map as algorithm with IEEE Floating Point Format which only can reach \(10^{30}\) [7]. It can improve the Logistic Map 15.6 time in exponential to fight against Brute Force Attack.

### 3.3. Processing time

The simulation process of encryption and decryption process to get the average time process was tested with 4 different key and at least 100 times for every key for every data tester. Table 3 shows the average time of encryption and decryption process.

| Data tester | Logistic map Encryption(s) | Logistic map + NTT Encryption(s) | Logistic map Decryption(s) | Logistic map + NTT Decryption(s) |
|-------------|---------------------------|---------------------------------|---------------------------|---------------------------------|
| Lagu1.wav   | 4.711863                  | 4.90875                         | 5.640585                  | 5.938621                        |
| Lagu2.wav   | 12.84444                  | 15.32002                        | 13.78578                  | 16.47847                        |
| Lagu3.wav   | 20.38666                  | 25.28059                        | 21.43264                  | 26.38297                        |
| Lagu4.wav   | 23.51062                  | 29.08023                        | 24.40229                  | 30.02419                        |
| Lagu5.wav   | 358.6138                  | 468.8384                        | 358.0492                  | 462.2297                        |

The average time for encryption and decryption process is relatively similar. The format of relative key streams also does not have any significant impact on the speed of processing time. The file size is highly impacting the speed both encrypting and decrypting. Other than those, The NTT has slightly affected.

### 4. Conclusion

Based on the discussion, trials and simulation used in previous segment. We can conclude the following:

A. The key streams generated by Logistic Map are random which was verified by NIST’s tests.
B. The key space reach \(10^{469}\) which has high durability against Brute Force Attack.
C. The average encryption time is relatively faster than decryption time.
D. The usage key in Logistic map will not affected the time.
E. The NTT has slightly affected the processing time.

### Acknowledgment

This study is supported by Kementerian Ristek Dikti RI through Direktorat Riset dan Pengabdian Masyarakat Universitas Indonesia, in Hibah Penelitian Terapan Unggulan Perguruan Tinggi Tahun 2017 scheme.

### References

[1] Devaney, R.L. 2.5(1989). *An introduction to chaotic dynamical systems (2nd ed.)*. Addison-Wesley Publishing company, Inc.

[2] Kocarev, L., & Lian, S. (2011). *Chaos-based cryptography*. Berlin Heidelberg : Springer-Verlag.

[3] Pareek, N.K., Patidar, V., Sud, K.K. (2006). Image encryption using chaotic logistic map. *Journal of Image and Vision Computing*, 24, 926-934.

[4] Patidar, V., Pareek, N.K., Sud, K.K. (2009). A new substitution-diffusion based image cipher using chaotic standard and logistic maps. *Journal of Commun Nonlinear Sci Numer Simulat*, 14, 3056-3075.

[5] Miki Yasushi, (2013). Using the Logistic Map to Generate Scratching Sound. *The Mathematica Journal*, Wolfram Media, Inc., 8, 1-27
[6] Suryadi M.T., (2013). New Chaotic Algorithm for Video Encryption, 4th the International Symposium on Chaos Revolution in Science, Technology and Society 2013, Jakarta, 28-29, August, 2013.

[7] Suryadi M.T., Eva Nurpeti, and Dhian Widya. (2014), Performance of Chaos-Based Encryption Algorithm for Digital Image, Journal Telecommunication Computing Electronics and Control, 12(3), 675-682.

[8] Sonis, M., (1996). Once more on Hénon map: Analysis of bifurcations, Journal of Chaos, Solitons & Fractals, 7(12), 2215–2234.

[9] Zhang, W., Wong, K., Yu, H., Zhu, Z. (2013). An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion. Journal of Commun Nonlinear Sci Numer Simulat, 18, 2066-2080.

[10] Ye, Ruisong and Zhao, Haiying,, (2012), An Efficient Chaos-based Image Encryption Scheme using Affine Modular Maps, International Journal Computer Network and Information Security, 7, 41-50.

[11] Ansari, S., Gupta, N and Agrawal, S., (2012), An Image Encryption Approach Using Chaotic Map in Frequency Domain, International Journal of Emerging Technology and Advanced Engineering, 2(8), 287 – 291.

[12] Liu, S., Sun, J. and Xu, Z., (2009), An Improved Image Encryption Algorithm based on Chaotic System, Journal of Computer, Academy Publisher, 4(11), 1091-1100.

[13] Abu Zaid, Osama M., El-Fishaway, Nawal A., Nigm, E.M., (2013). Cryptosystem Algorithm Based on Chaotic System for Encrypting Colored Image, International Journal of Computer Science Issues, 10, Issue 4, No 2, 215-224.

[14] Gao, H., Zhang, Y., Liang, S., Li, D. (2006). A new chaotic algorithm for image encryption. Journal of Chaos, Solutions and Fractals, 29, 393-399.

[15] Liu, L., Zhang, Q., & Wei, X. (2012). A RGB image encryption algorithm based on DNA encoding and Chaos Map. Computers and Electrical Engineering design, 38(5), 1240-1248.

[16] Yohan Suryanto, Suryadi M.T., and Kalamullah Ramli. (2016), A Secure and Robust Image Encryption Based on Chaotic Permutation Multiple Circular Shrinking and Expanding, Journal of Information Hiding and Multimedia Signal Processing, ISSN 2073-4212, 7(4), 697-713.

[17] Yohan Suryanto, Suryadi M.T., and Kalamullah Ramli. (2017), A new image encryption using color scrambling based on chaotic permutation multiple circular shrinking and expanding, Journal Multimedia Tools and Application, Springer, 76, 16831–16854

[18] Makwana, Vinita and Parmar, Neha (2013). Analysis of Performance of Fast Fourier Transformation of an Audio File, International Journal of Application or Innovation in Engineering and Management, (IIAIEEM), 2, (11), 68-71.

[19] Rukhin A., et.all, (2010). A Statistical Test Suite for Random and Pseudorandom Number Generator for Cryptographic Application. National Institute of Standards and Technology (Revision 1a), Technology Administration U.S. Departament of Commerce.