Using audio noise for generating random key stream

Auda Raheemah Odhaib¹, *

¹ Al-Muthanna University, College of Engineering
*Corresponding author’s e-mail address: auda@mu.edu.iq

Abstract. Stream ciphers systems are widely used in cryptography applications due to low possibility of propagating errors, ease of implementation and high processing speed. A stream cipher is an encryption process where random binary stream keys combine with the plaintext to produce cipher text. The stream cipher systems encrypt bits of plaintext individually by adding bits from a key stream to plaintext bits. The security of a cipher text fully depends on the key stream. Therefore, generating a long unpredictable sequence of binary bits used as the encryption key is the main topic in stream cipher systems. This paper focuses on generating a stream of binary keys using audio noise and subjected the generated keys to several local randomness tests such as; frequency test, serial test, Poker test and Autocorrelation test. The randomness tests show that the generated keys contain good statistical characteristics and high security makes them suitable to be used in stream cipher systems.

1. Introduction
The Encryption is the process of converting clear and understandable text into incomprehensible codes to prevent unauthorized people from accessing or understanding this information. On the other hand, the decryption is the process of converting the encrypted text to its original form using the appropriate keys [1]. In today's era, the world is connected to each other through open networks where these networks are used to transmit information electronically to individuals or organizations (military or civilian). To keep this information confidential against the manipulation of hackers and thieves, sensitive information must be transmitted in encrypted form [1]. The Encryption keys used for encryption and decryption messages are based on complex algorithms. The encryption algorithm is a mathematical formula used to encrypt the plaintext or decrypt the encrypted text to get the original plain text. The strength and effectiveness of the encryption process depends on two basic factors; the algorithm and the length of the encryption key [1]. The encryption algorithm works in conjunction with a secret key that is used to encrypt the plaintext [1]. Figure (1) illustrates the operation of the encryption algorithm. The stream cipher systems encrypt bits of plaintext individually by adding bits from a key stream to plaintext bits. The security of a cipher text fully depends on the key stream. Therefore, generating a long unpredictable sequence of binary bits used as the encryption key is the main topic in stream cipher systems [2].

Several works have been suggested to design effective random generator. For example, the authors introduced a new method for generating key stream. The new method based on combined the AES algorithm with variable feedback generator. The proposed system improves the statistical properties of the resulting key stream [3]. In [2] the authors designed Pseudo-random generator (PRG) using Artificial Neural Network for generating a sequence of random binary bits. The design of PRG
includes two stages. In the first stage, the PRG has been implementing using FPGA technology. The randomness of resulting stream key has been checking in the second stage using the MatLab software. To give the required nonlinearity for the generated random binary key, the authors in [4] constructed a generator of the binary key stream based on the reverse the permutation. The empirical test was performed on the output of new generator show that the generated keys have good cryptographic properties and can be used in cryptographic systems. Meanwhile, [5] designed a new random generator with novel PS-LFSR substructure that not only has good random properties but also appears the same speed in data processing that has shown by Lee and Moon. The authors in [6] suggested a new random generator. The new generator has been built by mixing a number of shift registers with a mutual clock control structure to increase the nonlinearity of the resulting binary bit sequence. The researchers in [7] introduced a new random generator (NRG) for producing 32-bit or 64-bit in each iteration. The new NRG is appropriate for software encryption operations that need high-speed. Li and Moon [8] introduced a parallel shift register structure for generating random numbers suitable for a higher speed than the traditional encryption. Although the new design is convenient for encrypted the high-speed communication, the resultant key stream lacks good randomization. This research aims to generate a sequence of random binary bits by using the audio noise and study the complexity of resulting binary sequences by exposing them to a series of statistical tests to ensure that they contain good statistical characteristics. The statistical tests used in this study are frequency test, serial test, Poker test, and autocorrelation test.

2. Encryption systems
The Encryption system divided into two parts; the cryptosystem and crypto-analysis. The cryptosystem deals with design and construction of cryptographic systems (the ways in which plaintext is converted to encrypted text and vice versa with the secret key). The crypto-analysis studies the cipher text, ciphers, and cryptosystems with the intent of recognition how they work and finding and developing the methods for defeating or weakening them. The cryptosystems are divided into two categories: the public key and secret key encryption systems, which are then divided into stream and block cipher systems [1]. Figure (2) illustrates the classification of Encryption systems.
2.1. Stream Cipher System
Stream cipher system is one of the encryption systems that shared the same secret key in encryption and decryption operations. For example to encrypt the message $m := m_1 m_2 m_3 ... m_i$ where $m_i \in [1]$ with stream key $k := k_1 k_2 k_3 ... k_i$ where $K_i \in [9]$ This achieved by adding (XOR operation) bits from the stream keys to plaintext bits individually. Figure (3) illustrates the encryption and decryption operation.

$$E^*(k, m) = k \oplus m$$ (ciphertext)
$$D^*(k, c) = k \oplus c$$ (plaintext)

Where

$E^*$: represents the encrypted operation
$D^*$: represents the decrypted operation
Stream ciphers are sometimes referred to as one-time pads ciphers where every key stream bit is only used once [10]. Designing of encryption systems should be taken into consideration the following criteria [11]:

- The ease of implementation.
- The Speed of generating encryption keys.
- Complexity (high security).

The stream cipher system has many different properties made it be ideal and highly confidential. These include the ease of use in practical applications, high speed of implementation and lack of errors [10]. For the high complexity of the encryption system, the sequence of encryption keys used in encryption must be characterized by high randomness [4, 12].

2.2. Randomization of key stream

One aspect of measuring the quality of a cryptographic algorithm is based on its randomization. Randomization means not to predict the sequence of keys if it is possible to know part of them (it achieves a non-predictive property because there is no relationship between successive elements) [13]. There is no complete random sequence (infinite random) but there are some conditions that must be provided to be sequential keys are acceptable and then called pseudo random-sequence or pseudo-noise sequence [14].

2.3. Randomness tests

The generated stream key must contain good statistical properties to eliminate the statistical properties in the plaintext so that it is difficult to obtain plaintext from the encrypted text [15]. There are several statistical tests used to analyze a set of distribution data to measure the randomness of them. These tests called local statistical tests such as; frequency test, serial test, poker test, run test and Autocorrelation test [15, 16].

2.3.1 Frequency Test

The frequency test based on determines the number of ones and zeros that occurrence in the generated binary bit sequence. This test considers acceptable if the value of \( X^2 \) does not exceed the threshold (with freedom degree equal to 1, the value of \( X^2 \) not exceed 3.84) [2, 17]. The equation used in this test is:

\[
X^2 = \frac{(n_0 - n_1)^2}{n}
\]

Where

- \( n \) represents the length of generated binary sequence (numbers of bits).
- \( n_0 \) represents the numbers of zeros in the n-bit sequence.
- \( n_1 \) represents the numbers of ones in the n-bit sequence.

2.3.2. Serial Test

The serial test determines the number of occurrences of two bits at a time (01, 10, 00, and 11). This test gives indications if binary numbers do not depend on their predecessor. For a stream key with length equal to n bits, the result of this test must not exceed the threshold (with freedom degree equal to 1, the value of \( X^2 \) not exceed 5.99 (\( X^2 \leq 5.99 \)) [2, 17]. The equation that used in this test is:
Where

\[ n \] represents the length of generated binary sequence (numbers of bits).

\[ n_{00} \] represents the number of occurrence of (00) bits in the n-bit sequence.

\[ n_{01} \] represents the number of occurrence of (01) bits in the n-bit sequence.

\[ n_{10} \] represents the number of occurrence of (10) bits in the n-bit sequence.

\[ n_{11} \] represents the number of occurrence of (11) bits in the n-bit sequence.

2.3.3. Poker Test

The poker test divides the generated binary sequence into bit patterns (with length equal to m) and checks the occurrence of each \(2^m\) and \(m\) bit patterns for different values of \(m\). The generated stream key with n bits divided into number of blocks (\(F\)), each block has a length equal to \(m\) bits and then calculates the \(X^2\) of the generated stream key. The threshold in the Poker test is \((X^2 \leq 14.0671)\) [14, 17]. The equation that used in this test is:

\[
X^2 = \frac{4}{n-1} \sum_{i=0}^{1} \sum_{j=0}^{1} \left( n_{ij} \right)^2 - \frac{2}{n} \sum_{i=0}^{1} \left( n_{i} \right)^2 + 1
\]  

(2)

Where

\( n \) : represents the length of generated binary sequence (numbers of bits).

\( m \) : represents the number of bits in each block.

\( F \) : represents the number of blocks.

\( X_i \) : represents the number of blocks that contain number of ones = \(i\) and number of zeros = \(m-i\).

\[ F = \frac{n}{m} \]

2.3.4. Auto Correlation Test

This test uses to measure the randomness of the generated stream key with following binary sequence \((a_1a_2a_3\ldots a_n)\). The threshold of this test is when \((X^2 \leq 3.841)\) for each value of \(d\) [2, 17]. The equations used in this test are:

\[
A(d) = \sum_{i=1}^{n-d} a_i \cdot a_{i+d} \quad 0 < d < n - 1
\]

(4)

\[
X^2 = \frac{(A(d) - \mu)^2}{\mu}
\]

The value of \(\mu\) can be calculated as:

\[
\mu = \frac{n^2(n-d)}{n^2}
\]

Where

\( n \) : represents the length of generated binary sequence (numbers of bits).

\( a_i \) : represents the bits of the generated stream key.

\( d \) : represents a value between 0 and \(n-1\) \((0 \leq d \leq n-1)\)

3. Methodology

The experimental results involve two stages. The first stage is to record the noise audio and then converted it to the sequence of the binary bits. The second stage is to investigate the
randomness of the resulting keys using several statistical tests to ensure if they have good cryptographic characteristics

4. Results and discussion

4.1 Generating the key stream

The audio noise is recorded as (.wav) file format and then converted to a series of binary bits using Matlab programming. Figure (4) shows the audio file and converting operation it to sequence of binary bits. Figure (5) shows the resulting sequence bits.

Figure 4. audio file and matlab codes.

Figure 5. the sequence of binary bits.

4.2 Statistical tests

In order to decide the generated sequence binary key has good statistical properties and it's appropriate to use in cryptographic systems, the generated stream key must be tested by a set of statistical tests to
measure the strength of the key in terms of its randomness. The more randomness of generated stream key means the more difficult it is to know the plaintext of the attacker or the enemy. To check the randomness of the generating stream key, a portion of the resulting binary sequence bits is taken and subjected to a series of statistical tests, for example when taken the following binary sequence:

\[
0010011101001000110100111011000
\]

This binary sequence will be subjected to the following statistical tests to measure its randomness. The randomized of generated stream key are tested by subjecting the resulting keys to set of statistical tests using Matlab programming.

To determine the value of \( (X^2\text{-distribution}) \) for the **Frequency test**:

Length of this binary sequence is 31 bits
\[
n = 31, \quad n_0 = 14, \quad n_1 = 17
\]

\[
x^2 = \frac{(n_0 - n_1)^2}{n} = \frac{(17 - 14)^2}{31} = 0.2903
\]

To determine the value of \( (X^2\text{-distribution}) \) for the **Serial Test**

\[
n = 31, \quad n_0 = 14, \quad n_1 = 17
\]
\[
n_00 = 8, \quad n_01 = 8, \quad n_10 = 8 and \quad n_11 = 6
\]

\[
x^2 = \frac{4}{30} \left| n_00^2 + n_01^2 + n_10^2 + n_11^2 - \frac{2}{31} (n_0^2 + n_1^2) \right| + 1
\]

\[
x^2 = 0.109
\]

To determine the value of \( (X^2\text{-distribution}) \) for the **Poker test**

\[
F = \frac{n}{m} = \frac{31}{5} = 6
\]
\[
x_0 = 0, \quad x_1 = 2, \quad x_2 = 1, \quad x_3 = 2, \quad x_4 = 1, \quad x_5 = 0
\]
\[
\nu = \frac{\sum (n - d)}{n^2} = \frac{14^2(31 - 1)}{31^2} = 6.111 \quad for \ d = 1
\]
\[
\nu = \frac{\sum (n - d)}{n^2} = \frac{14^2(31 - 2)}{31^2} = 4.942 \quad for \ a = 2
\]

\[
x^2 = \frac{(A(d) - \nu)^2}{\nu} = 0.00227 \quad for \ d = 1
\]

To determine the value of \( (X^2\text{-distribution}) \) for the **Auto correlation Test**

\[
n = 31, \quad n_1 = 14, \quad n_0 = 17
\]
Table 1. shows the value of $X^2$ – distribution for each local randomness test (with a 0.05 significant level and degree of freedom equal to 1). In each test, the value of $X^2$ – distribution does not exceed the allowable threshold.

| Test           | Frequency | Serial | Poker | Autocorrelation |
|----------------|-----------|--------|-------|-----------------|
| $X^2$          | 0.2903    | 0.109  | 2     | 0.00277         |

5. Conclusion
The main aim of this work is to presents a new method for generating a sequence of random binary keys that are used for encrypting sensitive information. The stream key was generated by recording the audio noise as (.wav) file format and converting the recorded file into a sequence of binary bits using MatLab codes. The resulting binary bits are tested by several local statistical tests to measure their randomness and strength. The resulting randomness tests show that the value of $X^2$ – distribution are (0.2903, 0.109, 2.0 and 0.00277) for Frequency, Serial, poker and Autocorrelation tests respectively. All the results randomness tests are not exceeding the value of the threshold. Thus, the generated stream key has met the required conditions of randomness and valid for encrypting the messages in stream cipher systems.

6. References
[1] A J Menezes P C Van Oorschot and S A Vanstone 2018 Handbook of applied cryptography: CRC press.
[2] K M Othman and M Jammas 2011 Implementation of neural-cryptographic system using fpga journal of engineering science and technology 6 pp. 411-428
[3] H Shi X D Qian J W Lu J Gong and Y Q Deng 2014 The Design and Analysis of the Key-Stream Generator Based on the AES in Applied Mechanics and Materials pp. 430-433.
[4] V Scholtz and J Scholtzová 2010 The Stream Ciphers and the Keystream Generator Based on Inverse Permutation IFAC Proceedings 43, pp. 7-10.
[5] I Z Moghadam A S Rostami and M R Tanhatalab 2010 Designing a random number generator with novel parallel LFSR substructure for key stream ciphers in Computer Design and Applications (ICCDA) International Conference pp. 5 598-601.

[6] H Lee and K Chen 2007 Pingpong-128 a new stream cipher for ubiquitous application," in Convergence Information Technology International Conference pp. 1893-1899.

[7] Y Nawaz K C Gupta and G Gong 2005 A 32-bit RC4-like Keystream Generator IACR Cryptology ePrint Archive 2005 p. 175

[8] H Lee and S Moon 2002 Parallel stream cipher for secure high-speed communications Signal Processing 82 pp. 259-265

[9] A Z Abbasi N Islam and Z A Shaikh 2014 A review of wireless sensors and networks’ applications in agriculture Computer Standards & Interfaces 36 pp. 263-270

[10] P Gleichauf 2003 Method and system for securely storing and trasmitting data by applying a one-time pad ed: Google Patents

[11] A Klein 2013 Stream ciphers: Springer

[12] A Smolyakov A Sulimov A Karpov and O Sherstyukov 2013 Experimental verification of possibility of secret encryption keys distribution with a phase method in a multipath environment in Control and Communications (SIBCON) International Siberian Conference pp. 1-5.

[13] X Y Wang and L Yang 2012 Design of pseudo-random bit generator based on chaotic maps International Journal of Modern Physics B 26 p. 1250208, 2012.

[14] W M Abdel-Rehim I A Ismail and E Morsy 2015 Testing randomness: the original poker approach acceleration using parallel MATLAB with OpenMP Computer Science and Engineering 5 pp. 25-29

[15] Y Wang 2014 On the Design of LIL Tests for (Pseudo) Random Generators and Some Experimental Results 1401.3307

[16] Y Wang and T Nicol 2014 Statistical properties of pseudo random sequences and experiments with PHP and Debian OpenSSL in European Symposium on Research in Computer Security pp. 454-471.

[17] A Rukhin J Soto J Nechvatal M Smid and E Barker 2001 A statistical test suite for random and pseudorandom number generators for cryptographic applications Booz-Allen and Hamilton Inc Mclean