A Symmetric Key Algorithm for Cryptography using Music

Sandip Dutta[1], Chandan Kumar[2], Soubhik Chakraborty[3]

[1,2] Department of Information Technology, Birla Institute of Technology
Mesra, Ranchi-835215, India
[3] Department of Applied Mathematics, Birla Institute of Technology
Mesra, Ranchi-835215, India
[1]sandipdutta@bitmesra.ac.in, [2]chandankr@bitmesra.ac.in, [3]soubhikc@yahoo.co.in

Abstract: Music and its attributes have been used in cryptography from early days. Today music is vastly used in information hiding with the use of Steganography techniques. This paper proposes an alternative to steganography by designing an algorithm for the encryption of text message into music and its attributes. The proposed algorithm converts the plain text message into a musical piece by replacing the text characters of the message by mathematically generated musical notes. The sequence of musical notes generated for the particular character sequence of plain text message mimic a musical pattern. This musical pattern is sent to the receiver as a music file. The seed value for encryption/decryption key is sent using the asymmetric algorithm RSA, where the key maps the letters corresponding to a musical note. The encryption key used is an n x n matrix and it will be generated using the seed value for the key on both sender and receiver ends.

Keywords: Musical Cryptography, fundamental frequency, musical notes, encryption, decryption.

I. Introduction

With the advancements of information and communication technology, internet and its technologies have replaced the traditional way of information exchange. Undoubtedly, internet is the fastest medium of Information exchange. Every organization whether it belongs to private or public sector relies on these technologies. In the era of electronic communication, the demand of secure information exchange is highly needed and appreciated. There are situations where private personal data must be secured from the untrustworthy parties. This demand introduced the idea of cryptography in modern day communication system. Cryptography is the science of information security [1]. Cryptography has been used in early days for secure communication between kings and queens, in war like conditions and governmental issues, enigma and rotor machines are the examples of early day cryptography [2]. Modern day cryptography deals with the issues like confidentiality, data integrity, authentication, and non-repudiation. There are three types of cryptography, symmetric cryptography, asymmetric cryptography and message digest algorithm. In symmetric cryptography, the same key is used for the purpose of encryption and decryption. In asymmetric cryptography two separate keys i.e. one public key and another private key are used for encryption and decryption respectively. In asymmetric cryptography, the public key of the receiver is known to public, which can be used for encrypting the message which is to be sent to the receiver, while the private key is private to the receiver and is used to decrypt the encrypted message. A message digest algorithm uses a hash function to encrypt and decrypt a message.

Cryptographic algorithms are focused towards scrambling or disguising the message so that the intruder cannot get the message in the original form unless he knows the exact decryption algorithm along with the key used to decrypt the message. Steganography is the science of hiding the existence of the message [3]. Steganography algorithms are focused towards hiding a message into another message so that the intruder cannot guess the existence of the message [4]. Various steganography techniques use a cover file also called a stego object to hide the message in it. Images, audio, video files are generally used as the stego object. “Bit stuffing” is one of the concepts used in steganography. It replaces the predefined bits of cover file with the bits of the message to be transmitted [3], [4]. Nearly all steganography algorithms suffer with the problem of payload, i.e. if the message to be transmitted is larger in size finding a proper stego file is a problem.

As music is a vast arena, mathematically generated digital musical notes can be used for the encryption of plain text message. Music and its attributes can be vastly used in modern day cryptography; this can be used as a replacement of steganography. In digital form even a slightest variation in musical note can be differentiated using computers while a human ear cannot distinguish that. Digital media representation needs the digitization of analog signal into digital signal. The digitization process consists of discretization, sampling, quantization, encoding [5]. While using mathematically generated music, we are using the discrete sample values of a note at continuous discrete time intervals and the sample values to be scaled so that the values do not exceed the limits for sound values.

II. LITERATURE REVIEW

From early days music and musical notation have been used for the cryptographic purposes. The assignment of letters to an individual notes was the simplest cipher algorithm on those days. The musical scores have been used as substitution ciphers. Plain text has been represented by the use of musical scores; an individual musical note was used for a particular letter or word. Sams [6] observed that many cryptologists were notable musicians. In the 15th century Tractus varii medicinales [7] constructed a system comprising of five different pitches and used them in five different ways which yielded 25 symbols to make an alphabetic cipher, each pitch was given a certain notation and the stem directions and the note values were changed in five ways. By the end of 16th century many variations of complex systems were introduced which used 9 pitches and were capable to produce 72 different symbols. Garrison[8] used to send messages using ringing bells in prearranged ways. Athanasius Kircher [9] used the idea of orchestra by allocating four different notes each of six different musical instruments, this yielded 24 different notes. Hooper and Kluber [10], [11] used a cipher wheel, which had notes and corresponding letters written round in two circles. The device used by Hooper and Kluber permitted frequent resetting thus generating different notations at different times. In the late 18th and early 19th century it was an issue of debate for generation musical cipher with real music. Leibniz [12] gave an idea of artificial language containing tones and intervals. Leibniz in 1817 used seven different symbols and combined five at a time, the order and stress was also varied. Bach [13], [14] used musical notation to write names in musical style. Elgar [15] used musical notation to write messages to his friends, one of his message to his wife Dorabella is still unbreakable. Dutta [16] et al used 36 number (twelve musical notes in three different octaves) and encrypted a plain text message in musical notes. Dutta [17] et al also used raga Malkhauns whereby they exploited the transition probabilities of the Malkhauns notes to encrypt a message.

Sudoku or Latin squares have been used for cryptographic, and steganography purposes. Latin squares are n X n square matrix with n different elements, each occurring exactly once in a row and a column. Kadouche [20] et al have proposed an information hiding technique using Sudoku and wet paper codes. Bakhtiari [21]et al have used latin squares for message authentication codes. Kulkarni [19] et al have used fuzzy logic to generate encryption and decryption matrix. Cooper [22] et al used critical set from latin squares for secret sharing schemes.

Random number generation is a big task in any cryptographic algorithm, random number generator function, basically pseudo random generator are more prone to attack. Kelsey [23] has addressed the problems with pseudo random number generators. Gutterman [25] has pointed the problems with linux random no generation technique. Lagarias [24] has addressed the relation between number theory and random number generation. Lagarias also talked about the dependence of cryptographic algorithms on pseudo random number generators.

The rest of the paper focuses on designing a symmetric key cryptography algorithm which uses musical notes to substitute the characters of the plain text message. The input to the algorithm is the message in plaintext along with the key (refer to Fig 1). The key will be generated using the seed value and the predefined note set, which will map the substitution of letters to musical notes. The seed value is a value given to a random function to start with. Every random function repeats a same sequence of random number while generating random number. The output of the encryption algorithm is the sequence of musical notes in a musical file. The input to the decryption algorithm is the key and the musical file (refer to Fig 2). The decryption key will be generated using the seed value and the predefined note set which will be same as the encryption key. The corresponding output is the message in plain text. The main aim of the algorithm is to represent the message as a sequence of musical notes and not to hide the existence of the message, as in done in steganography.

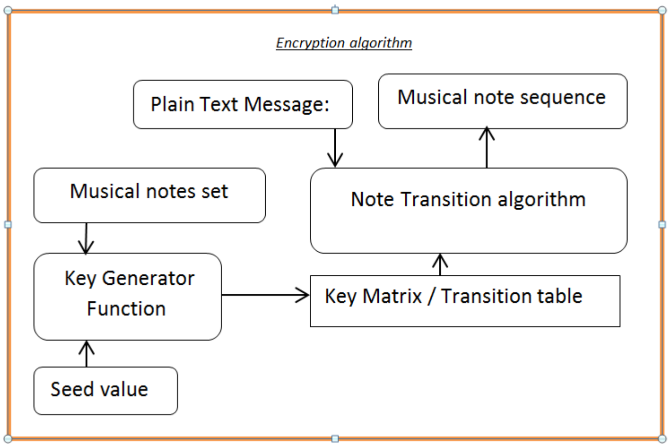


Fig 1: Encryption of plain text message.

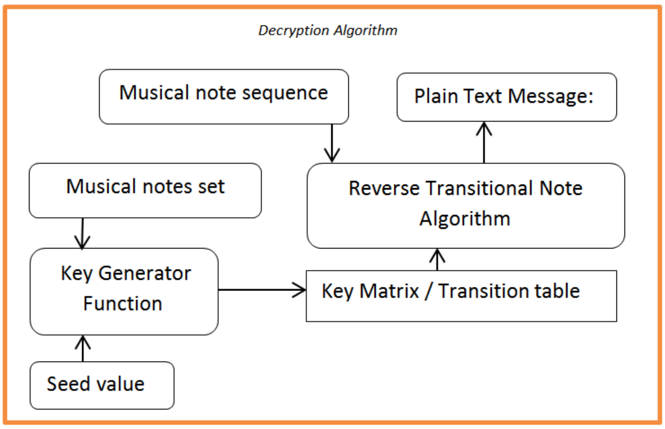


Fig 2: Decryption of Musical Note sequence.

III. PROPOSED ALGORITHM

A musical note consists of fundamental frequency (pitch is the perceived fundamental frequency), its amplitude (volume) and its shape (or its character). For a simple sound we can use a simple sine wave. A single note could be represented by the function:

$$A\sin(2\pi ft)$$

Where: f is the frequency (cycles per second)
t is the elapsed time or the duration of note
A is the amplitude

The function $A\sin(2\pi ft)$ generates a sinusoidal waveform with frequency f as the time t progresses. The waveform is the nature or shape of the wave. Here the time t goes from 0 to the specified time interval for the note which is set to be half a second for our case that means a particular note will play for half a second. The amplitude which is generally the measure of height of wave is set to be one unit for our case. Various mathematical functions can be used to generate musical notes. Some of the mathematical functions can be used to modulate the basic notes to generate the notes of different musical instruments. These different mathematical functions can mimic any available digital sound for e.g. Electronic guitar, violin etc. A symphony can be produced mixing different musical instruments together for encrypting messages, making the decryption more complex to decode by intruders.

Table 1. Frequencies for mean tone [18]

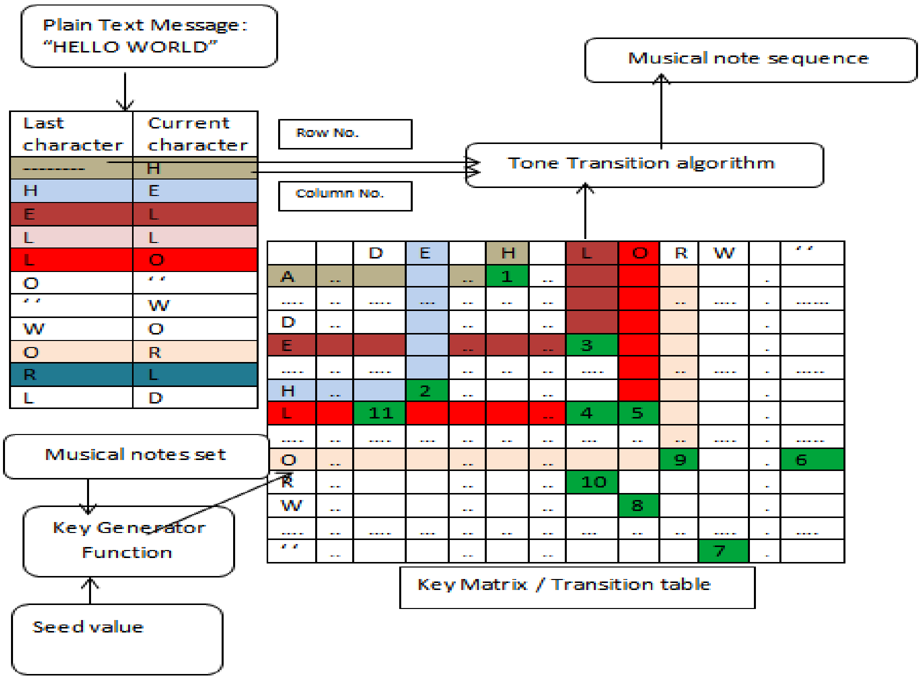
Note/Octave	1	2	3	4	5	6
C	65.07	130.14	260.29	520.58	1041.16	2082.31
C#/Db	69.46	138.92	277.84	555.69	1111.37	2222.75
D	73.27	146.54	293.08	586.17	1172.34	2344.68
D#/Eb	77.42	154.83	309.66	619.33	1238.65	2477.30
E	82.57	165.14	330.28	660.56	1321.12	2642.24
F	86.72	173.45	346.90	693.80	1387.60	2775.19
F#/Gb	93.05	186.10	372.19	744.39	1488.78	2977.56
G	97.65	195.30	390.61	781.21	1562.43	3124.86
G#/Ab	103.70	207.41	414.82	829.64	1659.28	3318.56
A	110.00	220.00	440.00	880.00	1760.00	3520.00
A#/Bb	115.58	231.16	462.33	924.65	1849.31	3698.61
B	123.96	247.92	495.84	991.68	1983.36	3966.72

We have taken 12 notes and 6 octaves into consideration (refer table 1). This gives rise to 72 notes which will be used to encrypt 72 characters. We have used two functions to generate the musical notes. The frequencies for the used notes are given in the frequency set which has been taken from the third octave of Table 1, the third octave is the mean and the multiples of the values of third octave are also being considered. The generated notes are shown in the note set. The generated 72 musical notes are all different. The characters to be encrypted are been shown in the character set. Capital letters are converted to small letters. A transition table is generated having 72 rows and 72 columns where, a unique note is assigned to every cell in a corresponding row. The transition table will have a random permutation of the 72 notes in each row. The transition table serves as the encryption and decryption key. The Transition table will be generated using the seed value for the key on sender and receiver side, this table will also be called as key matrix/ transition matrix. The transition table will be used to find the note for a particular character as per the occurrence of character in the plain text. The transitional note algorithm finds the particular note sequence for the plain text message. The generated musical sequence is then saved as a (.wave) file. This wave file along with the seed value for the key is send to the receiver as an encrypted message. The receiver after getting the musical file i.e. the wave file provides the file along with the key to the decryption algorithm and gets the message in plain text.

A. *Note transition algorithm*

- The mapping of the note for first character will be done by finding the note corresponding to the letter in the respective column of the first row from the transition table.
- The second note will be generated by searching in the column for the character in the row number of the last character from the transition table (that is the transition of current letter from the last letter corresponds to the musical note).
- The third and so on notes will be generated as per the second step while the end of character has not been reached.
- The generated note sequence is saved into a wave file.

The note is generated according to the transition of the character from the last character (refer to Fig 3).



Last character	Current character
--------	H
H	E
E	L
L	L
L	O
O	' '
' '	W
W	O
O	R
R	L
L	D

Fig 3: Encryption using note transition algorithm

B. *Reverse transitional note algorithm.*

- The first note is searched in the first row of the transition table and the corresponding character is read, the character read is the first character of plain text message. This character is saved and will be used in the next stage.
- The previous character is the row in which the next note is to be mapped from the transition table; the mapped column number will correspond to the second letter.
- The whole sequence is repeated until the end of file is reached.
- The character sequence read is saved into a text file.

Character set = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', ' ', '!', '"', '#', '$', '%', '&', '"', '(', ')', '*', '+', ',' , '-' , '.', '/', ':', ';', '<', '=', '>', '?', '@', '[', '\', ']', '^', '_', '{', '|', '}', '~', ' ', '`' ,'∑','©']

Freq. set = [260.29, 277.84, 293.08, 309.66, 330.28, 346.90, 372.19, 390.61, 414.82, 440, 462.33, 495.84] note set = [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20 a21 a22 a23 a24 a25 a26 a27 a28 a29 a30 a31 a32 a33 a34 a35 a36 a37 a38 a39 a40 a41 a42 a43 a44 a45 a46 a47 a48 a49 a50 a51 a52 a53 a54 a55 a56 a57 a58 a59 a60 a61 a62 a63 a64 a65 a66 a67 a68 a69 a70 a71 a72]

The generation of transition matrix will be done by random permutation of the whole note set for each rows. This transition matrix is the key for the encryption and decryption algorithm.

C. Functions used for generating notes.

Notes	**Functions used**	**Values of the variables**
For tones a1 through a12	A(i) = $\sin(2 \times \pi \times f(\alpha) \times 2^{\beta} \times t)$;	Where α varies from 1 to 12 $\beta = 0$, i varies from 1 to 12, and t [0,T], T=0.5
For notes a13 through a24	A(i) = $\sin(2 \times \pi \times f(\alpha) \times 2^{\beta} \times t)$;	Where α varies from 1 to 12 $\beta = 1$, i varies from 13 to 24, and t [0,T], T=0.5
For notes a25 through a36	A(i) = $\sin(2 \times \pi \times f(\alpha) \times 2^{\beta} \times t)$;	Where α varies from 1 to 12 $\beta = 2$, i varies from 25 to 36, and t [0,T], T=0.5
For notes a37 through a48	A(i) = $(\sin(2 \times \pi \times f(\alpha) \times 2^{\beta} \times t)-$ $\cos(2 \times \pi \times f(\alpha) \times 64 \times t))/3$	Where α varies from 1 to 12 $\beta = 0$, i varies from 37 to 48, and t [0,T], T=0.5
For notes a49 through a60	A(i) = $(\sin(2 \times \pi \times f(\alpha) \times 2^{\beta} \times t) -$ $\cos(2 \times \pi \times f(\alpha) \times t))/3$;	Where α varies from 1 to 12 $\beta = 2$, i varies from 49 to 60, and t [0,T], T=0.5
For notes a61 through a72	A(i) = $(\sin(2 \times \pi \times f(\alpha) \times 2^{\beta} \times t) -$ $\cos(2 \times \pi \times f(\alpha) \times t))/12$;	Where α varies from 1 to 12 $\beta = 3$, i varies from 61 to 72, and t [0,T], T=0.5

Example: - Consider a 10X10 transition table for numerals 0 through 9 (refer table 2). For a particular message say a phone no: 09433938113 the notes will be generated corresponding to the no's with notes as:

(1) a9 for 0 {going in the first row and column for 0 }
(2) a7 for 9 {going in the row for 0 and column for 9}
(3) a1 for 4 {going in the row for 9 and column for 4}
(4) a3 for 3 {going in the row for 4 and column for 3}
(5) a4 for 3 {going in the row for 3 and column for 3}
(6) a2 for 9 {going in the row for 3 and column for 9}
(7) a5 for 3 {going in the row for 9 and column for 3}
(8) a7 for 8 {going in the row for 3 and column for 8}
(9) a8 for 1 {going in the row for 8 and column for 1}
(10) a5 for 1 {going in the row for 1 and column for 1}
(11) a2 for 3 {going in the row for 1 and column for 3}

The generated musical sequence is [a9 a7 a1 a3 a4 a2 a5 a7 a8 a5 a2]

The output of the encrypted message is a series of musical notes mimicking a musical pattern. The intruder on getting this musical file cannot guess the existence of encrypted message in the form of music. For, the decoding of the note into message we will check the first note in the first row and find the column and, this column will specify the first letter we will also save the current column no as say current. For the second letter we will find the second note in the row no with the value current and find the column no and update the value of column with the new one. This process will run until we have processed the last note.

The reverse transitional note algorithm on musical sequence [a9 a7 a1 a3 a4 a2 a5 a7 a8 a5 a2] will produce the result as:

(1) 0 for a9 {going in first row & column for a9}
(2) 9 for a7 {going in row for 0 and column for a7}
(3) 4 for a1 {going in row for 9 and column for a1 }
(4) 3 for a3 {going in row for 4 and column for a3 }
(5) 3 for a4 {going in row for 3 and column for a4 }
(6) 9 for a2 {going in row for 3 and column for a2 }
(7) 3 for a5 {going in row for 9 and column for a5 }
(8) 8 for a7 {going in row for 3 and column for a7}
(9) 1 for a8 {going in row for 8 and column for a8}

(10) 1 for a5 {going in row for 1 and column for a5}
(11) 3 for a2 {going in row for 1 and column for a2}

Deciphered Result: 09433938113 as string.

Table 2: Transition table for numerals 0 through 9 having a 10X10 matrix of musical notes.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	a9	a10	a1	a5	a4	a6	a8	a3	a2	a7
1	a9	a5	a3	a2	a6	a10	a7	a4	a1	a8
2	a2	a9	a6	a5	a4	a3	a7	a10	a1	a8
3	a9	a1	a8	a4	a6	a10	a5	a3	a7	a2
4	a5	a9	a2	a3	a10	a8	a4	a7	a1	a6
5	a9	a10	a6	a8	a1	a5	a3	a7	a4	a2
6	a9	a8	a10	a6	a2	a1	a3	a5	a7	a4
7	a8	a10	a5	a2	a4	a6	a1	a7	a9	a3
8	a5	a8	a7	a2	a6	a9	a3	a1	a4	a10
9	a3	a8	a2	a5	a1	a7	a10	a6	a9	a4

IV. Implementation, Result and Discussion

The proposed algorithm is implemented in Matlab, it uses sine and cosine functions to generate musical notes. 72 different musical notes are generated using these function and these musical notes are assigned for the characters of the message using the transitional note algorithm. The run time of the algorithm is found to be linear. The file size of the generated musical sequence is also linear. The proposed algorithm takes nearly 7.82 Kb to represent a particular note. As the proposed algorithm uses transition of the text literals to find the particular note, the algorithm is better than simple substitution ciphers.

The proposed algorithm gives different result for the same set of letters e.g. for ‘at’ in “cat” and “bat” Cat will correspond to say “a1, a17, a29”.While Bat will correspond to say “a3, a42, a29”.

This is because the “at” in “cat” and “bat” will have different transitions, as in cat the note for ‘a’ will be found in the row for ‘c’ and column for ‘a’, while in bat it will be in row for ‘b’ and column for ‘a’. While for “t” the note will be same as the transition for “t” in both the case is from “a”.

The transition tables for different set of sender and receiver will have a key that is a matrix of 72x72 where each row has random permutation of musical notes 1 through 72. Which makes the probability of guessing the right key to be $1/(72!)^{72}$ which is nearly impossible to guess. The random permutation for the transition table is fixed by setting the seed value for the random function, which will help us reducing the effort of sending the key; it will further enhance the security of key. The proposed algorithm can be further extended to any no of character set with the use of different mathematically generated musical notes.

V. Conclusion and Future Work

Various musical attributes can be used in modern day cryptography. A better cryptic algorithm is demanded in future, which may generate musical sequence as real world music. Indian classical music and raga can be used in the transitional algorithm, which may mimic a real musical pattern or raga. The generation of random transition table in Sudoku form is left as future work. We are also exploring the possibility of using the duration of the musical notes for encryption.

VI. References

[1] AbuTaha, Mohammed, et al. "Survey Paper: Cryptography Is The Science Of Information Security." *International Journal of Computer Science and Security (IJCSS)* 5.3 (2011): 298..

[2] Davies, Donald. "A brief history of cryptography." Information Security Technical Report 2.2 (1997): 14-17.

[3] Gopalan, Kaliappan. "Audio steganography using bit modification." *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*. Vol. 1. IEEE, 2003..

[4] Rhoads, Geoffrey B. "Audio steganography." U.S. Patent No. 6,330,335. 11 Dec. 2001.

[5] Burg, Jennifer. *The science of digital media*. Prentice Hall/Pearson Education, 2009.

[6] Sams, Eric. "Musical cryptography." CRYPTOLOGIA 3.4 (1979): 193-201.

[7] Sadie, Stanley E. "The new Grove dictionary of music and musicians." (1980).

[8] William Chambers, Robert Chambers, “A chapter on bells” , Chambers's Journal, Volume 24, page-78.

[9] Kircher, Athanasius. Musurgia universalis.: 1650. 1988.

[10] Davies, H. Neville. "The History of a Cipher, 1602-1772." *Music & Letters* (1967): 325-329.

[11] Klüber, Johann Ludwig. *Kryptographik*. 1809.

[12] Coudert, Allison P., Richard Henry Popkin, and Gordon M. Weiner, eds. *Leibniz, mysticism and religion*. Vol. 158. Springer, 1998.

[13] Bourne, Joyce. The concise Oxford dictionary of music. OUP Oxford, 2004.

[14] Tatlow, Ruth. “Bach and the Riddle of the Number Alphabet”. Cambridge University Press, 1991.

[15] Sams, Eric. "Elgar's Cipher Letter to Dorabella.", *The Musical Times* 111.1524 (1970): 151-154.

[16] Dutta S, Chakraborty S, Mahanti N.C., "A novel Method of Hiding Message Using Musical Notes", International Journal of Computer Application (0975-8887) volume1-No.16, 2010.

[17] Dutta S, Chakraborty S, Mahanti N.C., "Using Raga as a Cryptographic Tool", Advances in Network Security and Applications, Communications in Computer and Information Science, 2011, Volume 196, Part 1, 178-183, DOI: 10.1007/978-3-642-22540-6_18, D. C. Wyld et. al. (Eds.), CNSA 2011 (Springer).

[18] http://www.phy.mtu.edu/~suits/etvsmean.html

[19] Kulkarni, S. S., Rai, H. M., & Singla, S. Design of an Effective Substitution Cipher Algorithm for Information Security using Fuzzy Logic.

[20] Kadouche, R., Abdulrazak, B., Giroux, S., Mokhtari, M., Mou, T. Y., Jeng, T. S., ... & Matías, I. R. A Sudoku Based Wet Paper Hiding Scheme.

[21] Bakhtiari, S., Safavi-Naini, R., & Pieprzyk, J. (1997, January). A message authentication code based on latin squares. In *Information Security and Privacy* (pp. 194-203). Springer Berlin Heidelberg.

[22] Cooper, J., Donovan, D., & Seberry, J. (1994). Secret sharing schemes arising from Latin squares.

[23] Kelsey, J., Schneier, B., Wagner, D., & Hall, C. (1998, January). Cryptanalytic attacks on pseudorandom number generators. In Fast Software Encryption (pp. 168-188). Springer Berlin Heidelberg.

[24] Lagarias, J. C. (1990). Pseudorandom number generators in cryptography and number theory. *Cryptology and computational number theory*, *42*, 115-143.

[25] Gutterman, Z., Pinkas, B., & Reinman, T. (2006, May). Analysis of the linux random number generator. In *Security and Privacy, 2006 IEEE Symposium on* (pp. 15-pp). IEEE.