International Journal of Security and Its Applications
Vol.9, No.1 (2015), pp.237-248
http://dx.doi.org/10.14257/ijsia.2015.9.1.23

Hiding Messages using Musical Notes: A Fuzzy Logic Approach

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

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

Abstract

"Music can be used as a communicable language". Musical symbols and musical notes have been used as codes and ciphers from early days. The art of encrypting messages using music is termed as Musical cryptography and it uses predefined set of notes and set of rules for composing musical pieces which in turn are musical cryptograms. Traditional algorithms applied to musical cryptography used simple substitution cipher which merely produced good musical sequences. To overcome this problem a fuzzy logic based algorithm for musical cryptography is proposed in this paper. The paper proposes a symmetric key substitution cipher which uses one of the n *candidate notes to encrypt a particular character. The application of fuzzy logic in musical cryptography produces acceptable musical sequences which are hard to be suspected as cipher.*

Keywords: *Musical Cryptography, musical notes, algorithmic composition, fuzzy logic, encryption, decryption*

1. Introduction

Various techniques have been employed for private communication from the age of Julius Caesar. The process of secure information exchange demands techniques that convert the message into unintelligible form so that the intruder cannot guess the content of the message. For secure information exchange cryptography and steganography have been used. Cryptography is the art of transforming the plain text message into an unintelligible form (cipher text) which cannot be understood by untrustworthy parties [1-2]. The process of converting plain text message into cipher text is known as encryption, while the process of getting plain text back from cipher text is known as decryption. Various mathematical transformations are used to convert the plain text message into cipher text. These transformations can be simple permutation and combination, transposition, substitution, matrix multiplication, use of number theoretic approaches etc. For encryption and decryption process usually a key is used, if the same key is used for both encryption and decryption the process is known as symmetric key cryptography while two different keys, private and public keys are used in asymmetric key cryptography. In asymmetric key cryptography the public key is known to public which can be used to encrypt the message while the private key remains private to the receiver which is used to decrypt the message. Cipher algorithms can also be classified as block ciphers and stream cipher depending on the working. A block cipher algorithm encrypts a particular fixed length sized block while a stream cipher encrypts a stream of plain text and can encrypt a particular character at a time. Steganography is the art of hiding messages into another message; by doing this the

ISSN: 1738-9976 IJSIA

existence of the message is also concealed [1]. Generally a cover file is used to hide the secret message inside it [2]. Examples of early day steganography include secret inks which can glow on applying certain chemicals, Morse code written on knitting yarns, microdots, use of different type faces for cover media and intended message, writing messages on the shaved head of soldiers and letting it grow to hide the message etc. Modern day steganography uses digital media as a cover file to hide the intended message also known as payload data. Images, audio, video and executable files are used as cover media. The bits of the cover media is modified as per the bits of intended message. The techniques used for modifying bit are LSB least significant bit substitution, echo hiding, parity bit modification, DCT discrete cosine transform, wavelet transforms, etc. Both steganography and cryptography have advantages and disadvantages over another. The aim of cryptography is to encrypt the message in such a way that the cipher text should not be decrypted without having the access of the decryption key and guessing and trying all the possible keys should not be feasible. The aim of steganographic algorithms is to hide the message in such a way that guessing the locations for the bits of intended message or the encoding scheme in the cover file is impractical. Nowadays, cryptography and steganography are used together to solve the purpose of information security.

With the enormous advancement of computing power the early day cryptography techniques have succumbed to death. Today cryptographic algorithms are devised keeping consideration in mind that the intruder cannot get the plain text message from the cipher text in polynomial time. Modern day cryptography addresses the issues of confidentiality, integrity, authentication and non-repudiation.

2. Musical Cryptography

Musical cryptography uses musical notations and musical notes to encrypt messages. The encrypted messages can be in form of musical symbols, verbal or instrumental musical sequences. Musical cryptography produces musical cryptograms which are hard to be detected as cipher. Music and its attributes have been have used in cryptography from early days. The simplest of the musical ciphers used to replace characters of the plain text message with musical notes. Eric Sams [3] in his article "Musical Cryptography" has pointed that many of the cryptologists were notable musicians. Haydyn and Ravel [3] used musical symbol to hide messages. Elgar [4] used to write the names of his friends in musical notation. Bach [8-9] used musical notation to write names of his friends, he has also hidden names of his friends in several of his compositions. A cipher wheel was used by Hooper and Kluber [5] to encrypt messages. Schumann derived three lines and eight notes cipher system from Kluber's work. Tractus varii medicinales [10] devised a system which used five notes which varied with stem directions in five ways, giving rise to 25 symbols which were further used to encrypt messages. Many complex systems were devised following the work of Tractus, a system comprising of 9 pitches in 8 ordering was devised which was capable to encrypt 72 different symbols. Athanius Kircher [6] a polymath used orchestra in musical cryptography. Kircher used four notes from six different instruments and encoded 26 characters. Lebniz [11] gave an idea of a language comprising of notes and intervals. Olivier Messiaen [12] put forward an idea of language communicable which was based on the ideas of using music for a mystic language. Olivier's language communicable is based on three components: alphabets, case system and leitmotif series. Dutta *et al.* [13] used musical notes to hide messages. Dutta *et al.* in their work used 12 notes from three octaves to encrypt 26 characters and 10 numerals. Dutta *et al.*

[14] in their work used the transition probabilities of raga malkauns to encrypt messages in musical style. Dutta *et al.* [15] generated mathematical waveforms for the musical notes and used them to encrypt message. Dutta *et al.* in their work have also used transition of characters for getting candidate notes. Yamuna *et al.* [16] have used musical notes along with graph theory to encrypt binary messages. Maity [27] used magic squares to permute the number for the Polybius cipher. Glatfelter *et al.* [25] have proposed a framework for encrypting messages using frequencies of musical notes. Glatfelter *et al.* have also used the matrix multiplication for obtaining the cipher message. Lee *et. al.* [26] have proposed a rhythm key based encryption scheme for Ubiquitous Devices where they have used musical rhythms for generating the cryptographic key for encryption process. Yamuna [27] has proposed an insertion technique using musical notes where she has encrypted messages using the codes generated using musical notes.

3. MIDI Files and Algorithmic Composition

3.1. MIDI

MIDI is an abbreviation for Musical Instrument Digital Interface, it is a standard developed for the communication of musical devices. MIDI is an event list representation of musical composition. A general midi file contains sequences of musical notes, the timing intervals, the control messages etc. A simple musical note or musical event can be represented in MIDI as a set of onset, duration, MIDI channel, MIDI note, velocity associated to a particular note in chronological order of occurrence of events (refer Table 1). The onset time specifies when a particular note will start to play, onset time is provided in Beats and Seconds. Duration tells about how long a particular note will play. MIDI channel specifies one of the channels from the 16 channels in which a particular note will play. In polyphonic music all the 16 channels can be used for simultaneous playback of different notes for different instruments. MIDI notes represent the pitch of the notes; generally pitch is also called fundamental frequency and is represented as the note number in MIDI. MIDI note number ranges from 0 to 127. “C Db D Eb E F F# G Ab A Bb B” are the 12 chromatic notes used in western music composition whose Indian equivalent is “Sa, re, Re, ga, Ga, ma, Ma, Pa, da, Da, ni, Ni”. An octave contains the 12 chromatic notes. Same note played on different octave have different frequencies. The relation between the frequencies of notes of two different octaves can be described with the temperament. MIDI is capable of handling 10.7 octaves which is generally beyond the limits of the instrument. Velocity tells how soft or loud a particular note will play. Velocity can take value from 0 to 127. MIDI files are best suited for the purpose of musical representation in Algorithmic Composition.

Table 1. General Midi Data Structure

Onset (Beats)	Duration (Beats)	MIDI channel	MIDI Note	Velocity	Onset (Sec)	Duration (Sec)
0.00	1.48	1	50	127	0.00	0.89
1.50	0.98	1	09	127	0.90	0.59
2.50	1.00	1	85	127	1.50	0.60
3.50	0.50	1	64	127	2.10	0.30
4.00	0.98	1	29	127	2.40	0.59
5.00	0.48	1	33	127	3.00	0.29

Algorithmic composition refers to the art of using computers and some deterministic or stochastic rules along with soft computing tools for generating good piece of musical sequences in automatic and semi-automatic fashion [17-18, 20]. Recurrent neural networks, cellular automata, rule based grammars, genetic algorithms, fuzzy logic and many others have been employed in musical compositions [17-19]. These techniques not only help the composers in composing good musical piece but it also helps in reducing the manual efforts in composition and reduces the time required. In algorithmic composition the composer is seen as the searcher and all possible musical compositions are considered as search space [18]. The composer tries to reach at optimum or nearly optimum solution which is in terms of quality musical sequence.

Any permutation and combination of musical notes does not produce music. Music composition consists of set of rules and grammar [23-24]. Algorithmic composition should take care of harmonic and melodic relation between consecutive and concurrent notes. Melody refers to the playback of musical notes in sequence (one after another) in a way which is pleasant to ear. Harmony refers to the concurrent playback of notes which are in consonance to each other. For better understanding harmony and melody (refer Figure 1 and 2). Harmonic and melodic relations are hard to quantify, so use of strict rules and grammars does not provide flexibility in algorithmic composition. Chord progression also plays a vital role in music composition. Random occurrence of musical notes is prohibited in music; we can predict the next note to occur provided we have knowledge of the last notes and the transition probabilities of all the other notes. Transition probability is defined as the probability of occurrence of next note provided the previous note was fixed. Transition probability matrix for a particular genre can be deduced from observation of several musical pieces of that genre (refer Table 2). Markov chains have also been used to predict notes to occur. A first order Markov chain considers previous outcomes to predict next outcome, while second order Markov chain uses last two outcomes to predict the next outcome.

Table 2. Transition Probability Matrix for Raga.Bilawal

	Sa	**Re**	**Ga**	**ma**	**Pa**	**Da**	**Ni**
Sa	0.075	0.313	0.161	0	0.048	0.130	0.273
Re	0.475	0	0.460	0.024	0.024	0.017	0
Ga	0	0.267	0.013	0.358	0.362	0	0
Ma	0	0.512	0.464	0	0.024	0	0
Pa	0.010	0	0.010	0.495	0.067	0.410	0.010
Da	0.010	0	0.010	0.052	0.448	0	0.479
Ni	0.343	0.015	0	0	0.015	0.612	0.015

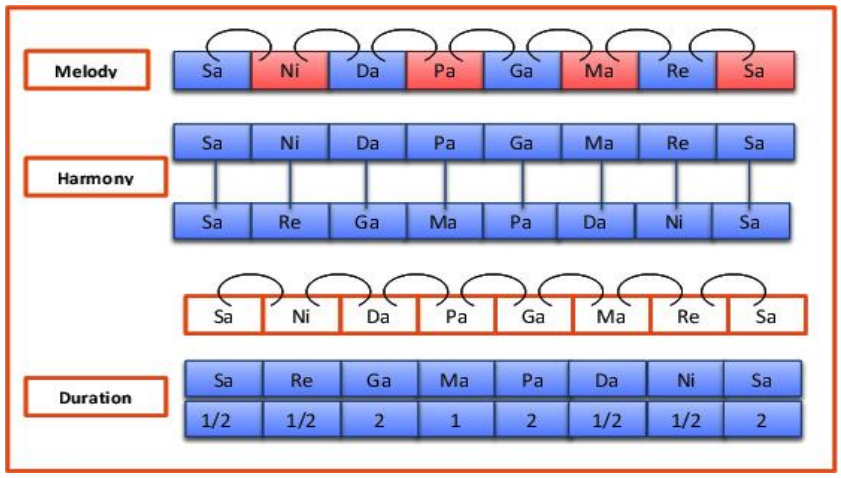

Figure 1. Representation of Melody and Harmony

Sa	Re	Ga	Ma	Pa	Da	Ni	Sa	Harmonium
1/2	1/2	2	1	2	1/2	1/2	2	Duration
Sa	Re	Ga	Ma	Pa	Da	Ni	Sa	Tabla
1/2	1/2	2	1	2	1/2	1/2	2	Duration
Sa	Re	Ga	Ma	Pa	Da	Ni	Sa	Sitar
1/2	1/2	2	1	2	1/2	1/2	2	Duration

Figure 2. Concurrent Playback for Chords of Different Instruments

3.2. Fuzzy Logic

Fuzzy logic is used to formalize the human capacity of approximate reasoning by using fuzzy sets and fuzzy relations. The classical set theory defines the behavior of an object as either belonging or not in a particular set. In classical set theory the boundaries of the set are real valued. The classical definition of the sets fails in linguistic classification, e.g. hot, cold, short, medium and tall. We cannot define a real valued boundary for the class short or tall, neither it fits for the case of hot or cold. In these cases fuzzy sets are used, in fuzzy sets every member has a grade of membership in the set. The membership value ranges from 0 to 1, where zero means no participation and 1 means full participation, the values between 0 and 1 means partial participation. Classical set operations such as 'and' and 'or' between two fuzzy sets are realized using the minimum and maximum of the membership values of the two sets respectively. Fuzzy rules are used for approximate reasoning.

Fuzzy rules are used in musical composition for the harmonization and orchestration of musical sequences. A particular chord can be harmonized using the fuzzy rules. In this proposed work a particular window size (say of ***n*** notes) is taken which will be harmonized depending on the melodic structures of the chord and candidate notes. The fitness value of the generated sequence can be calculated using the weighted sum of the transition probabilities of the notes in the particular window and the fuzzified

membership for the melodic and harmonic aesthetic of the sequence. The genesis rules will be defined depending on the genre of musical composition chosen. These genesis rules will synthesize and sequence the musical notes to yield an optimal musical pattern. Rules for the note density, pitch range and repeated rhythmic values will be used to make the musical composition as realistic as possible. The relative note density is the proportion of notes and rests and the repeated rhythmic value is the proportion of melodic interval in which both notes have the same rhythmic value. Some of the general rules derived from the musical theory and compositions of various performers can be listed as below [21-22]:

Melodic Rule:

- A Chord should start and end with a tonic.
- Leaps along the notes should not be higher than two octaves.
- A step wise motion of at least half of the melodic interval should be present.
- Melodic consonance should be present and dissonant notes should be avoided.

Harmonic Rules:

- Parallel octaves should be avoided.
- Parallel fifths should be avoided.
- Consonance and dissonance should be considered.

Rhythm rules:

- Note duration of each type should be present from eighth note to full note.
- Same note should not repeat more than thrice.

Harmonic intervals:

Perfect consonance:	1,5,8
Imperfect consonance:	3,6
Dissonance:	2,4,7

Melodic Intervals:

Perfect Consonance:	1,5,8
Imperfect consonance:	3,4,6
Dissonance:	2,7

The fuzzy rules for the melody, harmony, step size can be defined with the membership $m_{harmony}$, $m_{melody,}$ $m_{stepsize}$.

The overall system function of the fuzzy logic block can be defined as

$$M = m_{harmony(interval)} . m_{melody(interval)} . m_{stepsize(interval)} * transition\_prob_{(notes)}$$

$m_{harmony(interval)}$ is the membership function for the Harmonyof the chord. (refer Figure 5)

$m_{melody(interval)}$ is the membership function for melody of the chord. (refer Figure 6)

$m_{stepsize(interval)}$ is the membership function for the step size rules. The membership value for stepsize is either 1 or 0.

Transition probability of the chord is weighted sum of transition probabilities of notes in the chord.

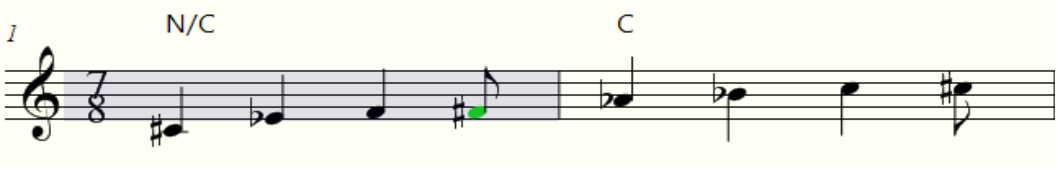

Figure 3. Notation of Chord "Sa re ga ma pa da ni sa"

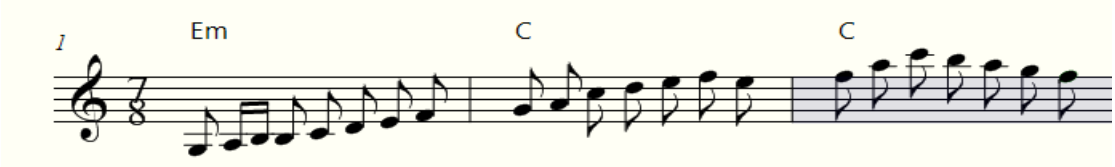

Figure 4. Relative Step Size of One for Two Consecutive Notes

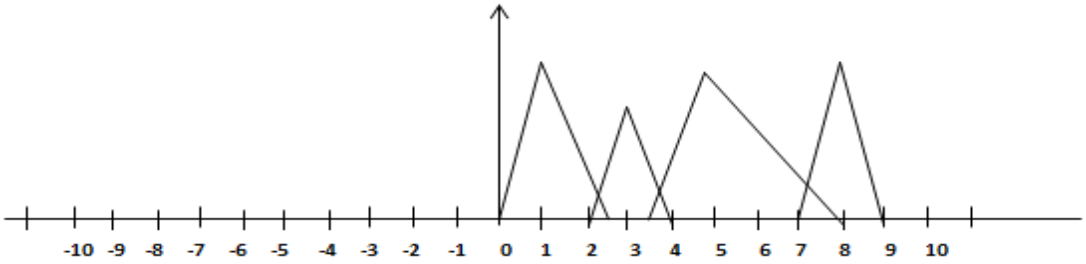

Figure 5. $m_{harmony}$

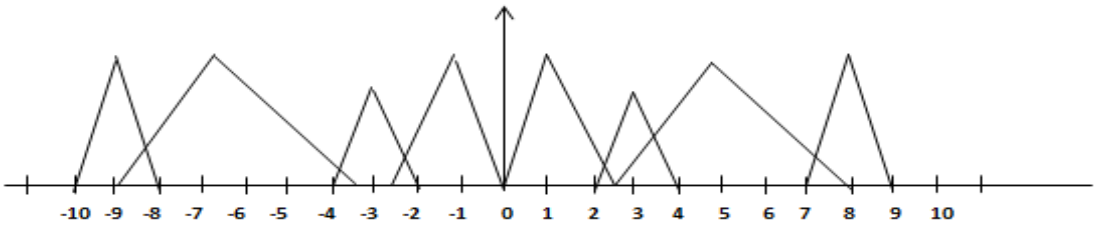

Figure 6. m_{melody}

4. Key Matrix and Genesis Rules

Let ***C={c1, c2, ……., cn}*** be the character set which is used to write the plaintext message and ***N={n1,n2,n3,……..nn}*** be the note set, where the features of ***ni*** can be denoted as ***fni={MIDI pitch, channel number, velocity, tempo}.*** The MIDI pitch and channels together will produce a note set of length ***128x16***, which can be used for encoding ***2048*** different letters into musical notes, provided we are using a simple replacement/ substitution of particular letter with single note.

In the proposed work the algorithm encodes a particular character into one of the ***m*** candidate notes. Which can be written as ***candidate_notes(Ci)={nm1, nm2,…, nmn}.*** This representation will mean that a particular character can be encoded by one of the several candidate notes, while one particular note can represents only one character. If ***n*** is the length of character set, then ***m*** should be chosen such that ***nxm*** <= ***2048***. The value ***2048*** represents the maximum possible length of the note set. If the length of character set is higher, then the number of candidate notes will be lesser. In that case

we will encrypt each byte of data individually, that means character set is reduced to a manageable length of ***256*** and the number of candidate notes will be ***8***.

The encryption and decryption key consists of an ***nxm*** matrix where ***n*** rows will correspond to ***n*** characters of the character set and each row will have ***m*** columns which will hold candidate notes for each character. A pseudorandom generator function is used to generate the key matrix. The pseudo random generator function uses a seed value to permute the note set to generate the key matrix. The same seed value will be used on both encryption and decryption side to generate the key matrix.

Table 3. Candidate Notes along with Channel for Each Character

Character	Candidate Notes (MIDI Note No, Channel No.)						
	1	**2**	**3**	**4**	**5**	**6**	**7**
..	……	……	…….	……	……..	…….	…….
E	54,7	12,6	24,3	18,9	16,7	22,5	22,3
H	111,5	22,4	16,3	12,2	11,7	17,12	27,4
L	11,6	11,2	24,8	17,6	18,3	12,3	11,1
..	..,..	..,..	..,..	..,..	..,..	..,.	..,..
O	65,13	13,7	7,13	22,6	17,9	2,8	18,6
..	….	….	…..	…..	…..	…..	…..

5. Encryption and Decryption Algorithm

5.1. Encryption Algorithm

The encryption algorithm takes key matrix and the plaintext message and produces the sequence of musical notes as a midi file, which can be transmitted securely over the wired network (refer Figure 7). The key matrix is generated using a seed value which initializes the random function for the random permutation of notes for each character (A part of key matrix is depicted by Table 3). The steps involved in the encryption process are:

- Generate the key matrix.
- Find candidate notes for each character of message.
- Feed candidate notes to the melodic composer.
- Apply fuzzy rules for the music composition and select the best plausible musical sequence generated.
- Generate the midi file as the encrypted message.

The melodic composer tries various permutations and combination and selects the best combination of notes where notes come sequentially one after another from one of the candidate notes for each character. The selection of the notes depends on the transition probabilities of the notes and fuzzy rules.

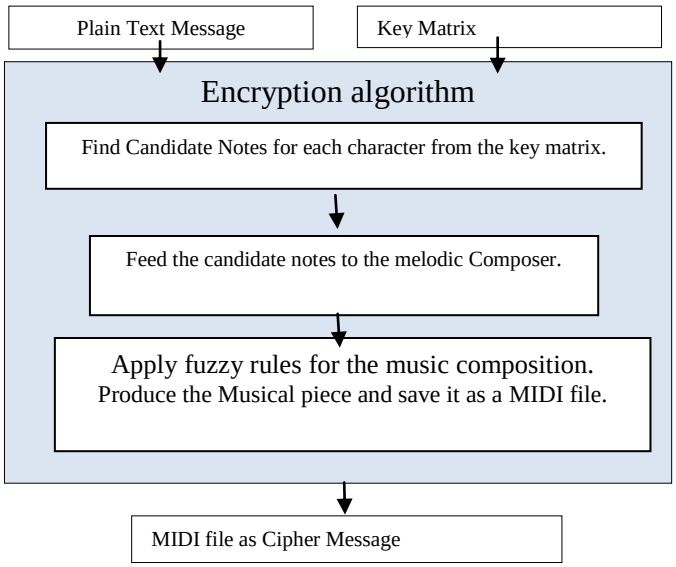

Figure 7. Encryption Algorithm

5.2. Decryption Algorithm

The decryption of the musical notes as a cipher message is simply done by mapping the characters for each note from the key matrix (refer Figure 8). The steps involved in decryption process are:

- Generate the key matrix using the seed value.
- Find the plain text character for each musical note by mapping notes to character form the key matrix sequentially.
- Save the plain text sequence.

The same seed value is used for generating the exactly same key matrix used on the encryption side.

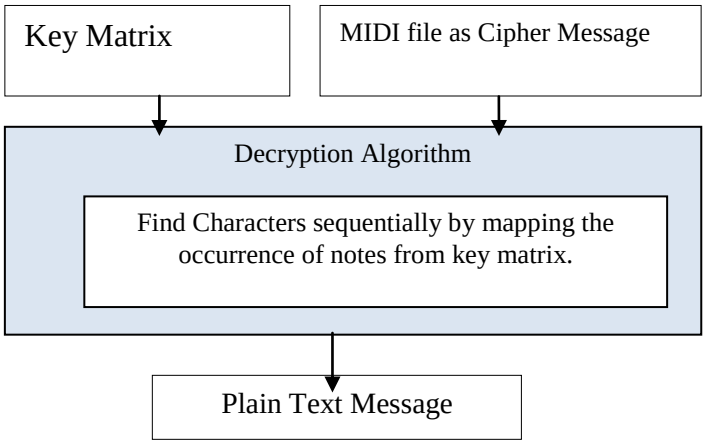

Figure 8. Decryption Algorithm

6. Implementation, Results and Discussion

The proposed algorithm was implemented in MATLAB®. MIDI library functions were used for representation of musical notes. Fuzzy genesis rules were defined depending on the genre used for composition. A pseudo random generator function was used to generate the key matrix. Depending on the characters of the plain text message the candidate notes were fed to the composer. The results of the encryption process were found to be quite satisfactory in terms of aesthetic appeal. The encrypted message in the form of musical piece was found to be more realistic so even after intercepting the communication, the intruder cannot guess the musical piece as a cipher message.

The same characters were encrypted into different musical notes depending on the occurrence of the characters in the plain text because of the fact that same character could have been encrypted using one of the several candidate notes. The same text encrypted with the same key produced different musical patterns which was an added feature of the proposed algorithm. The key used in the encryption and decryption is not a simple one-to-one substitution so guessing the key is very hard in practice.

The primary goal of enciphering plain text message using musical notes was achieved along with satisfying the second goal as an aesthetic appeal. Complex rules for note density, note duration, note intensity, rhythm and harmony are subject to a specific composition style, which can be adopted by following the styles of well-known composers. These rules then can easily be quantified using fuzzy rules and can be used to encipher messages. Musical cryptography can be seen as a counterpart for audio steganography.

References

[1] D. Kahn, “The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet”, (**1996).**
[2] D. Davies, “A brief history of cryptography”, Information Security Technical Report, vol. 2, no. 2, (**1997**), pp. 14-17.
[3] E. Sams, “Musical cryptography”, CRYPTOLOGIA, vol. 3, no. 4, (**1979**), pp. 193-201.
[4] E. Sams, “Elgar's Cipher Letter to Dorabella”, The Musical Times, vol. 111, no. 1524, (**1970**), pp. 151-154.
[5] J. L. Klüber, “Kryptographik”, (**1809**).
[6] A. Kircher, “Musurgia universalis”, 1650, (**1988).**
[7] A. P. Coudert, R. H. Popkin, and G. M. Weiner, “eds. Leibniz, mysticism and religion”, International Archives of the History of Ideas, vol. 158, Springer, (**1998**).
[8] J. Bourne, “The concise Oxford dictionary of music”, OUP Oxford, (**2004**).
[9] R. Tatlow, “Bach and the Riddle of the Number Alphabet”, Cambridge University Press, (**1991**).
[10] S. E. Sadie, “The new Grove dictionary of music and musicians”, (**1980**).
[11] A. P. Coudert, R. H. Popkin, and G. M. Weiner, “eds. Leibniz, mysticism and religion”, International Archives of the History of Ideas, vol. 158, Springer, (**1998**).
[12] A. Shenton, “Olivier Messiaen's system of signs: notes towards understanding his music”, Ashgate Publishing, Ltd., (**2008**).
[13] S. Dutta, S. Chakraborty and N. C. Mahanti, “A novel Method of Hiding Message Using Musical Notes”, International Journal of Computer Application, vol. 1, no. 16, (**2010**).
[14] S. Dutta, S. Chakraborty and N. C. Mahanti, “Using Raga as a Cryptographic Tool”, Advances in Network Security and Applications, Communications in Computer and Information Science, vol. 196, CNSA (**2011**) (Springer).
[15] S. Dutta, C. Kumar and S. Chakraborty, “A Symmetric Key Algorithm for Cryptography using Music”, International Journal of Engineering and Technology, vol. 5, no. 3, (**2013**), pp. 3109-3115.
[16] M. Yamuna, A. Sankar, S. Ravichandran and V. Harish, “Encryption of a Binary String Using Music Notes and Graph theory”, International Journal of Engineering and Technology, vol. 5, no. 3, (**2013**) June-July, pp. 2920-2925.
[17] D. Whitley, “A genetic algorithm tutorial”, Statistics and computing, vol. 4, no. 2, (**1994**), pp. 65-85.
[18] B. Jacob, “Composing with genetic algorithms”, (**1995**).

[19] A. Gartland-Jones and P. Copley, "The suitability of genetic algorithms for musical composition", Contemporary Music Review, vol. 22, no. 3, (**2003**), pp. 43-55.
[20] N. Fortier and M. V. Dyne, "A Genetic Algorithm Approach to Improve Automated Music Composition", INTERNATIONAL JOURNAL OF COMPUTERS, vol. 5, no. 4, (**2011**), pp. 525-532.
[21] A. E. Yilmaz and Z. Telatar, "Note-against-note two-voice counterpoint by means of fuzzy logic", Knowledge-Based Systems, vol. 23, no. 3, (**2010**), pp. 256-266.
[22] A. E. Yilmaz and Z. Telatar, "Fuzzy logic based four-voice choral harmonization in traditional style", *Journal of Intelligent and Fuzzy Systems*, vol. 21, no. 5, (**2010**), pp. 289-301.
[23] G. Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: A survey, a critical view and future prospects", *AISB Symposium on Musical Creativity*. Edinburgh, UK, (**1999**).
[24] M. Edwards, "Algorithmic composition: computational thinking in music", *Communications of the ACM* 54.7 (**2011**), pp. 58-67.
[25] J. W. Glatfelter and C. W. Raab, "Cryptography using a symmetric frequency-based encryption algorithm", U.S. Patent No. 8, vol. 855, no. 303, (**2014**) October 7.
[26] J. D. Lee, H. J. Im, W. M. Kang and J. H. Park, "Ubi-RKE: A Rhythm Key Based Encryption Scheme for Ubiquitous Devices", *Mathematical Problems in Engineering*, (**2014**).
[27] Y. Manimuthu, "INSERTION METHOD USING MUSIC NOTES", *Innovare Journal of Engineering & Technology* (**2014**).
[28] M. Maity, "A MODIFIED VERSION OF POLYBIUS CIPHER USING MAGIC SQUARE AND WESTERN MUSIC NOTES", International Journal For Technological Research In Engineering, vol. 1, no. 10, (**2014)** June.

Authors

Chandan Kumar, he is a research scholar in the Department of Computer Science and Engineering Birla Institute of Technology, Mesra, Ranchi. His areas of interest are Cryptography and network Security and Biometrics.

Sandip Dutta, he is a PhD in Computer Science, is Head of Department Computer Science and Engineering, BIT Mesra, Ranchi, India. His areas of interest are Cryptography and Network Security, Biometrics, Software Engineering. He has been guiding PhD scholars in the areas of cryptography and Software engineering.

Soubhik Chakraborty, a PhD in Statistics, is an associate professor in the department of Applied Mathematics, BIT Mesra, Ranchi, India. He has published several papers in algorithm and music analysis and is guiding research scholars in both the areas. He is a reviewer of prestigious journals like Mathematical Reviews (American Mathematical Society), Computing Reviews (ACM) and IEEE Transactions on Computers etc. Besides being the Principal Investigator of a UGC major research project on music analysis in his department he is also an amateur harmonium player.