Musical Cryptography using Genetic Algorithm

Chandan Kumar
Department of IT, BIT,
Mesra, Ranchi, India.
chandankr@bitmesra.ac.in

Sandip Dutta
Department of IT, BIT,
Mesra, Ranchi, India.
sandipdutta@bitmesra.ac.in

Soubhik Chakborty
Dept. of Applied Mathematics,
BIT Mesra, Ranchi, India
soubhikc@yahoo.co.in

***Abstract*—"Music can be used as a language communicable". Eric Sam's in his article "Musical Cryptography" has annotated various cryptographic techniques using music. Music and its attributes have been used in cryptography from early days. Musical symbols and musical notes have been used as substitution cipher. The algorithms applied to musical cryptography use predefined set of notes and rules for the synthesis of musical patterns. The main task in the musical cryptography is to generate musical cryptograms which in turn are good sequence of musical patterns soothing to ear. The traditional methods used in musical cryptography merely produced optimal musical patterns, to overcome this problem evolutionary computation approaches can be used. In this paper, a genetic algorithm (GA) based symmetric key musical cryptography algorithm is proposed to obtain an optimal sequence of musical patterns as a cipher message. The application of genetic algorithm for producing cryptic message doesn't only hide the message as musical piece but it also reduces the chance of cipher message to be detected as cipher.**

***Keywords*— Musical Cryptography, genetic algorithm, musical notes, algorithmic composition, encryption, decryption**

I. INTRODUCTION

In this era of digital word, the traditional communication has been transformed to digital communication with the use of Internet and its technologies. In the digital communication arena, keeping message private is a serious issue to be addressed. To cope with this challenge cryptography is used in information security. Messages rather the information to be exchanged is generally in a specific language that we people use, anyone who understands the language can get the gist of the message without the prior knowledge of the context. The messages to be transmitted are generally called as plain text message. Cryptography is the art of changing a plain text message into unintelligible or unreadable message [1], [2]. Cryptographic algorithms use a cipher algorithm to convert a plaintext message into cipher text and a decryption algorithm to convert the cipher text back into plaintext message. The cipher algorithm takes the plain text message along with the encryption key as input and converts the message into an unintelligible message known as cipher text as an output. The decryption algorithm takes cipher text message and the key used to decrypt the message. The very basic idea of cryptography is to apply some transformation to the plain text such that the message no longer remains intelligible to an intruder without applying the reverse transformation. These transformations are dependent on some permutation and combinatorial or some mathematical transformation. Transposition and substitution are the simplest transformation that has been used from the very beginning of the cryptography [1], [3]. Transpositions shuffles or jumbles the letters of the plain text message depending on the pre-agreed manner, i.e. the first letter of the plain text message may be placed at the tenth position in the cipher text. Substitution algorithms substitute a particular character of messages with other character or symbols, a particular word may be replaced by another word; a prior mapping scheme is to be agreed in this type of system, this mapping may be encoded as a dictionary of words or table of alphabets or symbols. Substitution algorithms may be simple to complex ones.

Depending on the key used to encrypt and decrypt the cipher algorithms are categorized as symmetric key and asymmetric key algorithms. Symmetric key algorithm uses the same key to encrypt and decrypt the message, while the asymmetric key algorithm uses two different keys for encryption and decryption. In symmetric key also called private key cryptography a secret key is being agreed upon prior to the transfer of the messages, depending on the key and the algorithm the sender side encrypts the message and the receiver decrypts the message. In asymmetric key also called public key cryptography a public key is used to encrypt the message while the private key is used to decrypt the cipher text. In context to the public key cryptography, public key is known to the public while private key is private to the receiver of the message. Depending on the working of cipher algorithms they can be classified as stream ciphers and block cipher. In stream cipher a stream of message is encrypted in a single pass i.e. byte wise while block cipher encrypts the message block wise, a fixed block size of the message at a particular time. In block cipher if the message is not a multiple of block size then the last block is padded with some arbitrary bits.

Apart from cryptographic algorithms Steganography is used in hiding messages. Invisible inks and Microdots are the examples of application of steganography. Steganography is the art of hiding secret messages into another message, by this it also conceals the existence of the message [4]. The message which is to be concealed is known as payload or the secret data, the object which covers the secret message is known as cover object, and the key used to hide the message is secret or public key. The message which contains the secret message in itself is known as the stego-object. Various file type associations are used as a cover file to hide messages. Images,

audio and videos are generally used as cover file to hide secret messages. LSB coding, Parity coding, Phase coding, Spread spectrum, echo hiding are the general techniques used in audio steganography. LSB coding changes the least significant bits of the cover message with the bits of the secret message. Parity coding replaces the parity bits of the cover message with the bits of the secret message. Echo hiding technique is used to hide the message by introducing an echo in the discrete signal.

Musical cryptography techniques convert the plain text message into musical notes or musical symbols. Musical cryptography uses music and its features to encode the massage into a cipher message, the cipher message can be symbolic, verbal or instrumental. A care should be taken while differentiating between musical cryptography and steganography.

II. Literature Review

Music and musical attributes have been used in early days of cryptography. Musical composers were employed to write messages in the form of musical notation [5]. The simplest cipher algorithm used to transpose a letter or word into corresponding musical score. Hiding messages in musical scores was nothing but steganography. People used to beat drums percussively to gather the public of certain area for passing messages. The idea of bells was used by the infantry in invading into other's territories in night. They used to ring predefined number of bells together for a particular message [6]. Bells of different sizes were used to produce different sounds that denoted different type of actions to be performed. Eric Sam's [5] in his article has mentioned that many of the cryptologists were notable musicians. Many composers used to hide messages in their musical compositions; Elgar and Schumann are one of them. Schumann used a three lines eight notes cipher system which was derived from Kluber's work. Hooper and Kluber [7] used a cipher wheel to encrypt messages. Elgar [12] used to write names of his friends in musical style. The compositions of Elgar gave a hint of steganography in his work; he used to hide the letters of names in each of the fifth or eleventh place in his musical score. Apart from musical cryptography one of Elgar's letter [12] to Dora Penny is still a cipher. Dora Penny was not a close friend to Elgar rather he had met her at very few occasions. Elgar's letter to Dora comprised of 87 characters, each in a form of one, two or three semicircles, oriented in one of the eight directions. In 2007 and 2008, the Elgar Society offered a prize of £1500 for solving the cipher, but no entry provided sufficient clue for solving the cipher, thus the cipher remained unsolved.

In the 15th century Tractus varii medicinales [13] used five different pitches in five different ways to produce 25 symbols to make an alphabetic cipher. Each of the pitches taken by Tractus had a certain notation and a stem direction. After the end of 16th century there were many variations of cipher algorithms similar to Tractus, a complex system comprising of 9 pitches in 8 different ordering was developed which was capable to encode 72 different symbols. Athanius Kircher [8], a polymath gave an idea of using orchestra in musical cryptography. Kircher used six different instruments with four notes from each to encode 26 letters. Leibniz [9] put forward an idea of musical language which was capable of mapping the plain text messages into an artificial language which contained tones and intervals. Bach [10], [11] used to write names of his friends in musical notations. Dutta [14] *et al.* have used 36 numbers by taking twelve musical notes each from three different octaves and encrypted the plain text message using musical notes. Dutta [14] *et al.* in his work was able to encrypt 26 character literals and 10 numerals. Dutta [15] *et al.* have used Indian raga "Raga Malkhauns", where they have first found the transition probabilities for transition of musical notes and used the transition probability to encrypt the message. Dutta [16] *et al.* has used mathematical function to generate waveforms for the musical notes and used them to encrypt messages. Dutta [16] *et al.* in his work has used the transition of characters to encrypt messages. Yamuna [17] *et al.* has used graph theory for hiding binary string using musical notes. In [18] the authors have used fluctuation of duration for hiding messages in standard MIDI file. In [19] authors have used three different steganography algorithms, namely LSB algorithm, repeated command algorithm and system exclusive command algorithm to hide message in standard MIDI files.

III. MIDI files and Music Attributes

MIDI, the Musical Instruments Digital Interface standard, was established in 1983. MIDI was developed for the communication of digital instruments. MIDI has instructions and commands which are sent to the devices for the playback of musical notes. MIDI files contain midi streams and midi commands. A standard MIDI file can contain more than one MIDI stream along with time information for each of the MIDI events. MIDI file have song, sequence, tempo and time signature information. Each musical sequence contains track and note events that build up together to form a musical piece. Each note in a MIDI contains the following data structure.

- *Onset*: Is the time when a note starts to play.
- *Duration*: Is the time for which a note will play.
- *MIDI channel*: Can take value from 1 to 16.
- *MIDI Pitch*: Can take value from 0 to 127.
- *Velocity*: Can take a value from 0 to 100.

The onset tells the time a particular note starts to play. Onset time is provided in "Beats" and "Seconds". The tempo/rhythm tells how many beats will be played per second. The duration tells us about how long the note will play. Same as onset time the duration is also provided in "Beats" and "Seconds". Midi channel denotes the channel number in which the note will play. It has 16 channels in which musical notes can be played simultaneously. The various instruments can be used together to form a symphony. Frequency/pitch is the standard pitch for a corresponding musical note. Pitch is denoted as the note number in midi. "Sa, re, Re, ga, Ga, ma, Ma, Pa, da, Da, ni, Ni" are the twelve chromatic notes used in

Indian classical music whose corresponding western notation is "C Db D Eb E F F# G Ab A Bb B". The velocity tells that how loud or soft a particular note will play.

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	9	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
5.50	0.98	1	42	127	3.30	0.59
6.50	1.00	1	63	127	3.90	0.60

IV. GENETIC ALGORITHM FOR COMPOSITION

Genetic algorithms are evolutionary algorithm based on the principle of genesis of life. Genetic algorithms have been used in art and music from a long time. The idea of genetic algorithm in music is to compose musical note sequence in automated or semi-automated fashion. The music composition can be seen as a search problem [20]. When we see musical composition as a search problem, all the possible composition can be said as a search space and the artist rather the composer can be viewed as a searcher, where the composer tries to reach at an optimal or reasonably satisfactory solution, which in turn is in terms of better composition results [21]. Many attempts have been made to represent musical notes with MIDI and form some rules to generate a composition [21][22][23].

Genetic algorithm starts with an initial population. The initial population holds the chromosomes, where each chromosome represents a possible solution. Each chromosome is associated with a fitness value, which represents the quality of the solution. Then chromosomes from the initial population are selected for crossover and mutation. After the crossover and mutation new chromosomes are generated which are now subjected to a fitness function. If the newly generated chromosomes pass the fitness test the chromosomes are added to the population and the old chromosomes are discarded [20]. After every generation the number of chromosomes in the population remains same, to accomplish this chromosomes with least fitness value are killed. The stopping criteria of Genetic Algorithm are generally either the maximum no of iterations reached or the satisfactory fitness value is reached. The chromosome with the maximum fitness value from the population is taken as the plausible solution. The basic elements of genetic algorithm for algorithmic composition are as follows

- Rhythm elements: Rhythm elements are described basic elements of musical notes. Mathematically rhythm element can be defined as
  RE= { *pitch, channel, velocity, duration, tempo* }
  A= $\{f_1, f2, \ldots\ldots, fn\}$ where n is the number of feature in a rhythm element and *f* is a feature
  Hence, Rhythm elements are set of features and information's used to represent a particular note for the initial population for genetic algorithm.
- Chromosome: Chromosome is the set of initial population which contains the rhythm elements. Chromosomes are the representation of individual solution; generally they are represented as binary strings. At the initial stage, the population is not optimized.
- Multi chromosome: Multi chromosome is a combination of candidate chromosomes together. In case of musical composition we use multi-chromosomes in spite of chromosomes. In case of musical composition multi-chromosomes represents a particular musical chord. Here, the population will contain multi-chromosomes.
- Selection Operator: The selection of multi chromosome is based on the roulette wheel method. In the roulette wheel method, multi chromosomes share segment of different sizes, based on their individual fitness. Hence, the chromosome with best fitness value has greater probability for being selected as parental chromosome. The selected multi-chromosomes undergo through crossover and mutation phase to produce offspring's.
- Crossover Operator: In crossover phase, successor of parental chromosome is generated using crossover operator. The output of the crossover phase is a pair of child chromosome. To generate the good pair of child chromosome multi crossover technique is used, if a pair of child chromosome is generated with greater fitness value then parental chromosome will be discarded and the child chromosome will be added to the population otherwise the parental chromosomes will be put back in the population. A straight crossover technique will be used to generate new multi-chromosomes.
- Mutation Operator: In mutation phase, multi-bit mutation is performed. The mutation operator selects bits randomly and changes the value 1 to 0 or otherwise. The reason to use multi-bit mutation rather than single bit is to generate the good piece of multi-chromosome. The example of multi-bit mutation is given as

  1011111 011 1011 000 010 011100 → Parent Chromosome

  1011011 010 1011 001 010 011101 → Mutated Chromosome
- Fitness Operator: The fitness function can be defined as the sum of all transition probabilities of all notes in a multi-chromosome.

V. KEY MATRIX AND GENESIS RULES

Consider a character set C={c1,c2,…..,cN }, where N is the maximum no of characters in the character set. Taking pitch and the channels together can give us a note set of length 127x16, which means we are capable of encoding 2032 different letters into musical notes provided we are using a simple replacement of letters with notes. The key matrix NMAT is an NxNxsp matrix in which NxN 2-D matrix denotes the transition of characters while the 3-rd dimension

hold the sparse list of musical notes for each transition. At a maximum of sp notes can be assigned for a particular transition. Nxsp selective notes can be used as the domain of the notes, which will further represent a particular letter in musical language. The notes for the cipher message will be generated using the transition of character literals. The same transition matrix will be used as the encryption and decryption key. Transition matrix will not be shared rather it will be created using the seed value or the initialization string. Pseudo random generator function will be used to generate the transition matrix/ key matrix. This pseudo random number generator function will be used on both encryption and decryption side, which will further help the cryptographic process in key exchange by reducing the effort of sending the bulky key and security of the key itself (see fig. 1, fig. 2).

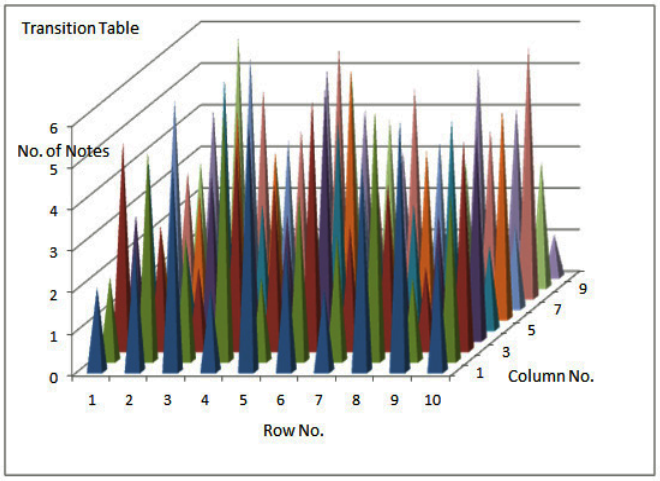


Figure 1: Transition Table for a NxN transition of characters with sp. no of candidate notes

The candidate notes for a particular transition can be found as:

Candidate notes T(ci,ci+1) = NMAT(i, map(ci+1)),

Where, ci is the last character and ci+1 is the character to be encoded.

When i=0 the mapping will be done as

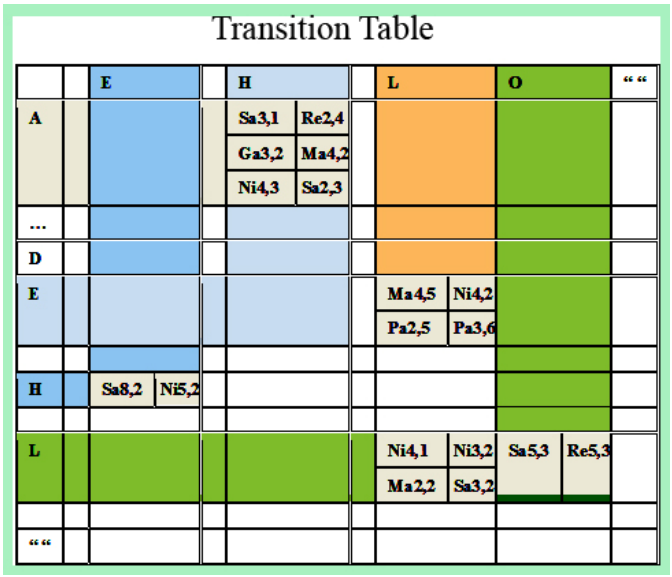
Candidate notes T(0,c) = NMAT(selected_start, map(c)),

Where, selected_start is the transition from none to the starting letter. The function map(c) will map corresponding candidate notes for the character c in a particular row.

In musical composition the transition of musical notes is not random that is stochastic. Some set of rules are there depending on which we can predict the next note provided we have knowledge of previous notes and the transition probabilities of all the other notes. Depending on the nature and mood specified by a musical composition/ genre a synthesis rule can be deducted. Various Indian classical ragas have their own set of notes and rules for the synthesis of notes into a musical composition. The transitional probability can be defined as the probability of occurrence of a certain note provided the last note was fixed. Transitional probability of a note can be found using ***Bayes' theorem***. The transitional probabilities of particular note will define the duration and tempo of the notes. A typical transitional probability matrix for the notes of a raga Bilawal using all the seven natural notes is given in the Table 2. In raga Bilawal five notes are never being used, so care should be taken while generating the candidate notes and domain of the notes for a particular transition table. In case of western composition all the twelve chromatic notes can be used, so western composition is best suited in the cases where a large note set is required.

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



The candidate note Sa8,2 signifies that Sa is in the eighth octave and the note will play in channel number 2. A particular cell is split to show the candidate notes for each transition.

For message "Hello" let the selected start be first row, then
Candidate notes for H= *NMAT(A,H)*={Sa3,Re2,Ga3,Ma4,Ni4,Sa2}
Candidate notes for E= *NMAT(H,E)*={Sa8,Ni5}

Figure 2: A Transition Table/Key Matrix to find Candidate notes for a particular character depending on its transition from the last character

The genesis rule will be defined on the basis of transition of notes. These genesis rules will define the velocity, tempo and duration of the notes. The concurrent play back of certain notes will also depend on genesis algorithm. Genetic algorithm with rule based system will be applied for the generation of musical sequence. The main aim of using

genesis rules is to make the encoded musical piece as natural as possible.

For every character we will get some candidate notes, the genetic algorithm will select and organise one of the candidate notes per character and form multi-chromosomes (see fig. 3). The multi-chromosomes are nothing but the combinatorial of candidate notes for each character. Then genetic algorithm will apply crossover and mutation on the multi-chromosomes for the optimal solution. A selected window size will be taken for the multi-chromosome which will take n number of characters to generate a musical sequence, as the window will progress the generated musical sequence will be concatenated to form the final musical sequence. Fig. 4 and Fig. 5 give the encryption and decryption algorithm.

Consider the Message "Hello"
The Candidate notes for the Characters be:

H={Sa3.7, Re2.4, Ni3.2, pa3.1}
E= {Ni4.2, Re4.1, Ma4.1}
L= {Da4.1, Ma4.2, Re4.3}
L= {Ma7.4, Ni7.3}
O= {Ma2.2, Ni5.1}

Where the Notes are followed with the octave and channel number, then the multi-chromosomes can be

Multi-Chromosomes

"Sa3,Ni4, Da4, Ma7, Ma2"
"Re2, Re4, Ma4, Ni7, Ni5"
"Ni3, Ni4, Da4, Ma7, Ni5"
.....
.....
"Pa3, Ma4, Re4, Ni7, Ni5"

and all other combinations of the candidate notes for five characters

Genetic Algorithm applies straight crossover operation to generate the best possible musical sequence.

Figure 3: Representation of Multi-Chromosomes

VI. Encryption and Decryption Algorithm

1) *Encryption Algorithm:*

ALGORITHM: ***INPUT*** *(plain text message, transition table/ key matrix, maximum no. of iteration),* ***OUTPUT*** *(Musical Sequence)*

- *Generate the transition matrix.*
- *Determine the candidate notes for each character of plain text and initialise the population for GA.*
- *For each window do following:*
  - i) Generate a total of n number of musical sequence (multi-chromosomes) as initial population for the window frame, where n is the total population size.
  - ii) For each window frame do following until the termination criteria reached:
    1) Select multi-chromosomes from the population.
    2) Apply multi-crossover operator on selected population.
    3) Perform multi-bit mutation operator on the features velocity, duration and tempo.
    4) Calculate fitness value of successor population (child chromosomes).
    5) If *fitness*(Parental Chromosome) ≤ *fitness*(Child Chromosome), then add child chromosome else add parental chromosome into the population.
    6) If (maximum iteration reached), then select the fittest multi-chromosome and add it at the end of final musical sequence else, iteration=iteration+1.
- *Obtain the final musical sequence as encrypted message.*

Consider the message "Hello"
The candidate notes for the characters be

Last Character	Current Character	Candidate Notes
	H	{ Sa3, Re2, Ni3, pa3}
H	E	{Ni4, Re4, Ma4}
E	L	{ Da4, Ma4, Re4}
L	L	{ Ma7, Ni 7}
L	O	{ Ma2, Ni5 }

The musical notes are followed with the octave number, the multi chromosomes for the given message with a window size of five notes can be given as "Sa3,Ni4,Da4,Ma7,Ma2", "Re2,Re4,Ma4,Ni7,Ni5" and all other combinations of the candidate notes together (see fig. 3). The Genetic algorithm will find the best combination of the candidate notes.

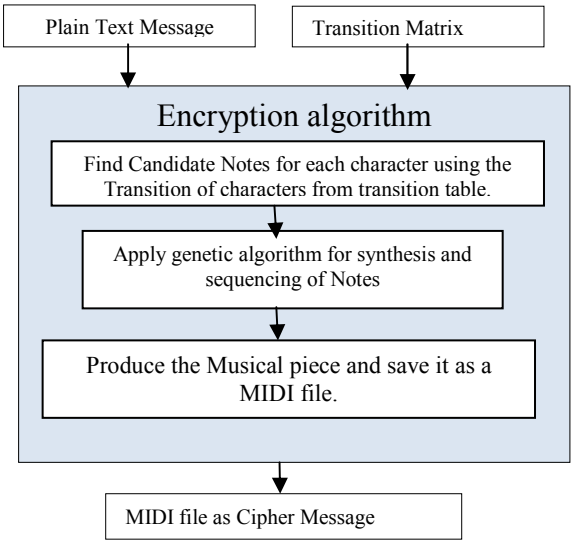


Figure 4: Encryption Algorithm

2) Decryption Algorithm:

The decryption will be done simply by searching the tones in the transition table starting with the first note in the selected_start row. The second and so on notes will be found as the transition of last character to the present note by reverse engineering the determination of candidate notes in the encryption process.

$Charcter_i$ = map(row(selected_start), current_note), if i=1
$Charcter_i$ = map(row(last_note),current_note), otherwise.

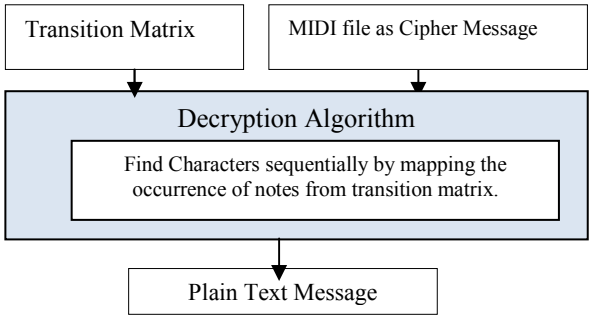


Figure 5: Decryption Algorithm

VII. Implementation, Result and Discussion

The proposed algorithm was implemented in MATLAB. At first the transition table was generated. A genesis rule was provided depending on the transitional probabilities of notes. Candidate musical notes were used to generate the initial population for the genetic algorithm. Depending on the transition of characters the candidate notes were provided to the genetic algorithm for proper synthesis and sequencing. The genetic algorithm has to do the proper selection, mutation, crossover and sequencing of the musical notes. The result of the genetic algorithm was found quite satisfactory with respect to simple substitution cipher. The runtime of the algorithm basically depended on the complexity of the selection, mutation and crossover of the chromosomes. The no of iterations and the fitness function had an important role to play. It was found that greater the no. of iterations and threshold value for the fitness function the generated musical piece was more realistic, but at the same time the time complexity increased linearly. The resultant musical piece is hard to break as the key is usually like a one-time pad and guessing or producing the key is nearly impractical. The proposed algorithm does not use any cover file to hide the message so there is no such payload problem as happens with steganography. It is totally different from standard midi file steganography techniques which generally use least significant bits or the control messages for the replacement; rather the proposed algorithm generates musical notes for each character on its own.

VIII. Conclusion and Future Work

Music is like a language. Expressing moods and temperaments is proprietary to the musician itself. So hiding messages in music makes it a private cipher language which can be understood by the musicians provided they have a prior knowledge that the musical piece in not a musical piece indeed it is a ciphered creed. Even though the musical piece doesn't sound natural and is guessed to have message encrypted in it, it is hard to get the message without the key used to decrypt the message. So using musical language it is possible to encrypt the message. The permutation and combinations required to decrypt the message makes it nearly impossible provided the intruder doesn't have the key.

References

[1] Kahn, David. "The Codebreakers: The Comprehensive History Of Secret Communication From Ancient Times To The Internet", 1996
[2] AbuTaha, M., Farajallah, M., Tahboub, R., & Odeh, M. (2011). Survey Paper: Cryptography Is The Science Of Information Security. International Journal of Computer Science and Security (IJCSS), Vol. 5, No. 3, 2011, pp.298.
[3] Davies, D. A brief history of cryptography. Information Security Technical Report, Vol. 2, No. 2, 1997, pp.14-17.
[4] Rhoads, Geoffrey B. "Audio steganography." U.S. Patent No. 6,330,335. 11 Dec. 2001.
[5] Sams, Eric. "Musical cryptography." CRYPTOLOGIA, Vol. 3, No. 4, 1979, pp.193-201.
[6] William Chambers, Robert Chambers, "A chapter on bells" , Chambers's Journal, Volume 24, page-78
[7] Klüber, Johann Ludwig. Kryptographik. 1809.
[8] Kircher, Athanasius. Musurgia universalis.: 1650. 1988
[9] Coudert, Allison P., Richard Henry Popkin, and Gordon M. Weiner, eds. Leibniz, mysticism and religion., International Archives of the History of Ideas, Vol. 158. Springer, 1998.
[10] Bourne, Joyce. The concise Oxford dictionary of music. OUP Oxford, 2004.
[11] Tatlow, Ruth. "Bach and the Riddle of the Number Alphabet". Cambridge University Press, 1991.
[12] Sams, Eric. "Elgar's Cipher Letter to Dorabella.", The Musical Times Vol 111, No. 1524, 1970, pp. 151-154.
[13] Sadie, Stanley E. "The new Grove dictionary of music and musicians." (1980).
[14] Dutta S, Chakraborty S, Mahanti N.C., "A novel Method of Hiding Message Using Musical Notes", International Journal of Computer Application , vol.1, No.16, 2010.
[15] 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, CNSA 2011 (Springer).
[16] Dutta, Sandip, Chandan Kumar, and Soubhik Chakraborty. "A Symmetric Key Algorithm for Cryptography using Music." International Journal of Engineering and Technology, Vol 5, No 3, 2013, pp. 3109-3115.
[17] M. Yamuna ,Sankar A., Ravichandran S., V. Harish Encryption of a Binary String Using Music Notes and Graph theory, International Journal of Engineering and Technology , Vol 5, No 3 Jun-Jul, 2013, pp. 2920-2925.
[18] Yamamoto, Kotaro, and Munetoshi Iwakiri. "A standard MIDI file steganography based on fluctuation of duration." Availability, Reliability and Security, 2009. ARES'09. International Conference on. IEEE, 2009.
[19] Adli, Alexander, and Zensho Nakao. "Three Steganography algorithms for MIDI files." Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on. Vol. 4. IEEE, 2005.
[20] Whitley, Darrell. "A genetic algorithm tutorial." Statistics and computing, Vol. 4, No. 2, 1994, pp.65-85.
[21] Jacob, Bruce. "Composing with genetic algorithms." (1995).
[22] Gartland-Jones, Andrew, and Peter Copley. "The suitability of genetic algorithms for musical composition." Contemporary Music Review, Vol. 22, No. 3, 2003, pp.43-55.
[23] Fortier, Nathan, and Michele Van Dyne. "A Genetic Algorithm Approach to Improve Automated Music Composition.", INTERNATIONAL JOURNAL OF COMPUTERS, Vol. 5, No. 4, 2011, pp.525-532.