Genetic Algorithm solution for Cryptanalysis of Knapsack Cipher with Knapsack Sequence of Size 16

Dr. R. Geetha Ramani
Professor and the Head
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India

Lakshmi Balasubramanian
Department of Computer Science and Engineering
Pondicherry Engineering College
Puducherry, India

ABSTRACT
With growth of networked systems and applications such as e-commerce, the demand for internet security is increasing. Information can be secured using cryptography, anti-virus, malwares, spywares, firewall etc. In cryptology, cryptanalysis is a discipline where the ciphers are attacked and broke thus enabling the developers to strengthen the cipher. Nowadays cryptanalysis of ciphers is gaining popularity among the research world. One among the ciphers is the knapsack cipher. There are many methods to attack this cipher. One among them is the Genetic Algorithm (GA) approach. Using GA, researchers have attacked the knapsack cipher with a knapsack sequence of size 8. This supports the ASCII representation of the characters. The ASCII representation supports the regional languages like English, numerals and symbols. In this paper we propose the attack on knapsack cipher with knapsack sequence of size 16 using Genetic Algorithm. The increase in knapsack sequence size increases the strength of the knapsack cipher. The increase in knapsack sequence size also supports for Unicode representation. Unicode representation gives hold for many regional languages. Since any language information can be transmitted over the network, this approach supports attack on text of any language. Also there is no constraint on the length of text to be attacked. An analysis on the impact of various GA control parameters viz. initial population size, operators' type and probability, etc are also carried out in this research work.

General Terms
Cryptanalysis, Genetic Algorithm (GA), Knapsack Cipher, multilingual languages, Cryptography.

Keywords
Cryptanalysis, Genetic Algorithm (GA), Knapsack Cipher.

1. INTRODUCTION
With more and more development in the field of computer networks and internet, the need for network, computer and information security is also increasing. Computer and information security [19] deals with CIA: confidentiality, integrity, and authentication. There are different ways to secure information passed over the network. One such a technique is cryptology. Cryptology is the science and study of systems for secret communication. It consists of two complementary fields. They are cryptography and cryptanalysis. Cryptography is the science of building new powerful and efficient encryption and decryption methods. Cryptanalysis is the science and study of method to break cryptographic techniques i.e. ciphers. It is used to find loopholes in the design of cipher. It in turns helps in increasing the potency of cipher. A cipher is a pair of algorithms, which create the encryption and the reversing decryption. Some of the ciphers are Monoalphabetic Substitution cipher, Polyalphabetic Substitution cipher, Permutation cipher, Transposition cipher, Merkle-Hellman Knapsack cipher, Chor-Rivest Knapsack cipher and Vernam cipher. Cipher which is taken in our study, knapsack cipher is explained below.

Knapsack cipher is a public-key encryption system. It is based on the sum of subset problem. It belongs to NP hard problem where it is difficult to find the solution and if the solution is provided then it can be verified quickly. In the subset problem, S is a finite set. S belongs to N which is a set of natural numbers and target is t. The question is whether there exists S' which belongs to S whose elements sum to target t [3]. The Merkle-Hellman knapsack cipher was the first implementation of a public-key encryption scheme. The Merkle-Hellman knapsack cipher attempts to disguise an easily solved instance of the sum of subset problem, called a super increasing sum of subset problem, by modular multiplication and permutation [11]. A super increasing sequence is a sequence $(b_1, b_2, \ldots, b_n)$ of positive integers which satisfies the following condition b_i is greater than sum of all b_j where j=1 to i-1, for each i such that i is between 2 and n, The integer n is a common system parameter. M is the modulus, selected such that $M > b_1 + b_2 + \ldots + b_n$. W is a random integer, such that $1 \leq W \leq M - 1$ and $\gcd(W,M) = 1$. π is a random permutation of the integers $\{1, 2, \ldots, n\}$. The public key of A is $(a_1, a_2, \ldots, a_n)$, where $a_i = W\, b_{\pi(i)} \bmod M$, and the private key of A is $(\pi, M, W, (b_1, b_2, \ldots, b_n))$.

Researchers have worked on cryptanalysis using traditional methods like Tabu Search, Genetic Algorithm etc. This paper focuses on attack on knapsack cipher using Genetic Algorithm (GA). Genetic Algorithm [21] is an evolutionary computation technique, which is a rapidly growing area of Artificial Intelligence. Genetic Algorithm is inspired by Darwin's theory [6] about evolution. The survey of attack on knapsack ciphers using Genetic Algorithm (**GA**) is presented in the next section.

2. RELATED WORKS IN ATTACK ON KNAPSACK CIPHER USING GA
Spillman [14] is the one who first tried to attack Merkle-Hellman knapsack cipher by Genetic Algorithm. In his approach, binary encoding and random selection are used. The evaluation function determines the nearness of the target sum of the knapsack to the given sum of terms. The function ranges between zero and one with one indicating the exact match. The mating process used is simple crossover. The mutation process consists of three different ways. The characters are encoded as 8 bit ASCII characters. Initial

population of 20 random 15-bit binary strings is used. Overall, in Spillman's work a minimal level of experimentation is included. 5 ASCII characters are used and all characters are decrypted. The Spillman's algorithm searches on an average of less than 2% of the space. [14]. In the Spillman's report of work there is no information about the initial conditions for the GA. Clark, Dawson, and Bergen [2] worked in attacking the knapsack cipher. It is an extension of Spillman's work. It contains the modified version of the same fitness function. This approach also uses 8 bit ASCII characters. The number of generations required, and the percentage of the key space searched vary widely between runs. The fitness function does not accurately describe the suitability of a given solution. A reimplementation can be done and comparison is also possible. But they are unnecessary because the fitness function used is inappropriate and new.

Kolodziejczyk [9] extended the work of Spillman [14]. Certain restrictions were defined on the encoding algorithm. They are: only the ASCII code will be encrypted, then, the super increasing sequence will have 8 elements; this number of elements guarantees that each character has a unique encoding (There are 256 ASCII codes and 8-element length will allow encrypting 2^8 characters), and plaintext should not more than 100 character length [9]. Binary encoding is applied and the crossover is simple. The mutation process is to move between two random points. Overall the paper [9] uses a completely trivial knapsack. But the algorithm used in this work, searches about 50% of search space, when compared with Spillman's results which is about 2% [14]. Large size of population, high crossover probability and a small mutation probability are the most optimal arrangements for GA.

Poonam Garg and Aditya Shastri [12] enhanced the previously published attack and re-implemented it with variation of initial entry parameters and results are compared with Spillman [14] results. It included the encoding restrictions given by kolodziejczyk. The initial population size has range in between 10 to 100. The fitness function given by Spillman is applied. One point crossover is used. The mutation process is to move between two random points. The effect of crossover rate, initial population and mutation rate are analyzed. The 8-elements sequence of hard knapsack problem is used to encode 8 bits ASCII code. Correct results are obtained on an average after 115 generations and exploring 50% of the search space.

Raghavan Muthuregunathan, et.al. [13] proposed a hybrid technique that uses both Genetic Algorithm and Hill Climbing in attacking the knapsack cipher. A Parallel computation is done using MPI (Message Passing Interface) and the results are analyzed. A Synchronous Master Slave approach has been used. 8 bit binary encoding is applied. Spillman's fitness function is applied for evaluation [14]. A random selection of individuals is made. Single point crossover is used. Interval Flip and Neighbouring Effect are types of mutation carried out. The reason for mixing Hill Climbing and Genetic Algorithm (GA) is that it is better to mutate the weaker individuals than breeding them with stronger individuals. The computational time was approximately 4 times less compared to Garg's work [12] but there is no decrease in number of generations.

In the previous works all have attempted to attack knapsack cipher which uses 8 bits ASCII code to encode. In our work, cryptanalysis of knapsack cipher is done which encodes 16 bits UNICODE. In previous works researchers have given restriction that plaintext cannot be more than 100 character length. In our work the plaintext character can be of any length. There is no restriction in the size of plaintext.

3. GA SOLUTION FOR ATTACK ON KNAPSACK CIPHER

Cryptanalytic attack on knapsack cipher belongs to the class of NP-hard problem. Due to the constrained nature of the problem, the attack reaches to varying levels of performance optimization. Therefore, this paper is attempting for a optimal solution that further improves the robustness against cryptanalytic attack with high effectiveness. So far only 8 bit ASCII character encoding is used and constraints such as the plaintext can only be English and some symbols were made. In this paper knapsack sequence of size 16 is used. It leads to the next level of representation of characters from ASCII to Unicode representation. Unicode representation can be used for any regional languages like Greek, Latin, Hindi, Tamil etc. Thus the information transferred in the network to be attacked can be multilingual. Secret conversations in defense, Government sectors adapt the knapsack cipher with the knapsack sequence of size 16. Merkle-Hellman knapsack cipher is used to convert plaintext into ciphertext. Super increasing sequence is converted into non super increasing using the values W, M. Non super increasing sequence of knapsack cipher is more difficult to break and hence it is used as the public key. The public key is only available for attack. Ciphertext is calculated from the non super increasing sequence. If the non super increasing sequence is {21033, 63094, 16375, 11711, 23422, 58557, 16665, 54322, 64252, 39720, 32718, 63106, 63119, 18753, 21135, 42270} then the ciphertext for the plaintext "QUICK" can be calculated as shown in Table 1.

Table 1: Ciphertext calculation

PLAIN TEXT	UNICODE (in binary)	CIPHER TEXT
Q	0000000001010001	145096
U	0000000001010101	163849
I	0000000001001001	145109
C	0000000001000011	103125
K	0000000001001011	166244

Similarly plaintext is converted into cipher text which is attacked using Genetic Algorithm. Hence this paper attempts to attack the strengthened knapsack cipher using GA. This paper also presents the analysis on the effect of various genetic control parameters viz. initial population size, operator probabilities and selection process. In following subsections Genetic Algorithm process, Individual representation, Initial population, Fitness evaluation, Termination condition, Selection methods, Crossover and Mutation are explained.

3.1 Basic Genetic Algorithm Process

The whole Genetic Algorithm process will be continued until all the characters of the plaintext are found. The process is shown in Fig 1.

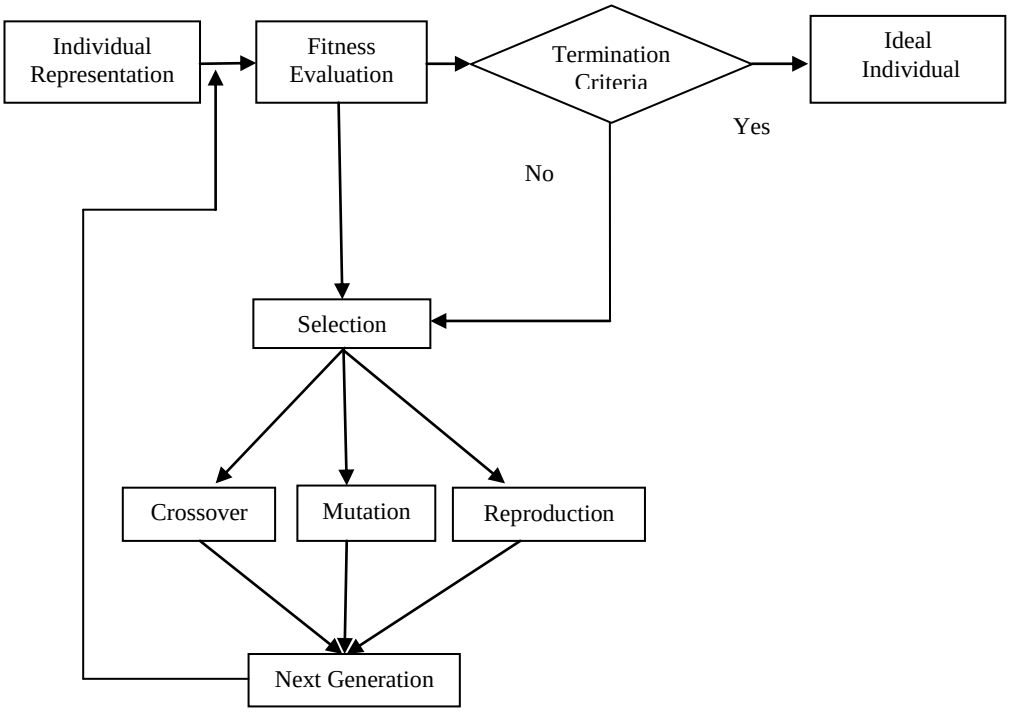

Fig 1: Genetic Algorithm Process

3.2 Individual Representation

Before the Genetic Algorithm can be put to work a method is needed to encode potential solutions to that problem in a form that a computer can process. Individuals are set of solutions which are initially randomly generated. For this problem, the approach that is used is to encode solutions as binary strings: sequences of 1's and 0's, where the 1's at each position represents the inclusion of the corresponding element in the non super increasing sequence solution. The binary representation is shown in Table 2.

Table 2: Binary repreaentation of the character

Character	Binary Representation
M (English)	0000000001001101
அ (Tamil)	0000101100010101

After representing the individuals, number of individuals to be generated initially is provided which is given in next subsection.

3.3 Initial Population

Process is started with a set of individuals which are ciphered plaintext called population. These individuals are randomly generated.

Randomly generated individuals with the initial population size of 3 are:

0100000100000000, 0110001000000000 and 0110001100000000

After the individuals are created they are evaluated using the function provided which is explained in next subsection.

3.4 Fitness Evaluation

In our work we used the following fitness function [14] to evaluate the generated individuals. Based on the fitness value obtained, it can be determined whether the optimal solution is reached or not.

$$\text{Fitness} = \begin{cases} 1 - \left\{\frac{Target - sum}{Target}\right\}^{\frac{1}{2}} & \text{if sum<=target} \\ 1 - \left\{\frac{Sum - Target}{Max\ Differenc\ e}\right\}^{\frac{1}{6}} & \text{if sum > target} \end{cases}$$

where,
MaxDifference = max(Target, FullSum − Target)
Target - the ciphertext.
Sum - the sum of the current chromosome.
FullSum - the sum of all components in the knapsack.

If the solution is not obtained the individuals are passed to next generation through selection methods and the individual chromosomes may be varied using genetic operators in next subsection.

3.5 Selection Methods and Genetic Operators

In the process of evolution, if the termination condition is not satisfied then the individuals are selected for next generation. Selection of individuals is done according to their fitness value obtained. Various selection techniques in GA are best selection, first selection, random selection, greedy over

selection, multi selection and tournament selection [20]. In our experiment tournament selection is used with different sizes.

The genetic operators used to vary the individual chromosomes are crossover and mutation. Crossover selects the parent individuals and creates the offspring. Different types of crossover explored in our experiment are single point crossover, two point crossover and multi crossover [20]. Samples of single point crossover are shown in Table 3 and Table 4.

Table 3: Sample1 of single point crossover

	Chromosome representation	**Plaintext character**
Before Crossover	0000000001000001	A
	0000000001100010	B
After Crossover	0000000001000010	*B*
	0000000001100010	A

Table 4: Sample 2 of single point crossover:

	Chromosome representation	**Plaintext character**
Before Crossover	0000101110000101	அ
	0000101110001010	ஊ
After Crossover	0000101110000110	ஆ
	0000101110001001	உ

After crossover is performed, mutation takes place. This is to prevent falling all solutions in population into a local optimum of solved problem. Types of mutation include bit inversion, order changing etc. Here we have used bit inversion kind of mutation [20]. Samples of mutation processes are shown in Table 5 and Table 6.

Table 5: Sample 1 of mutation process:

	Chromosome representation	Plaintext character
Before Mutation	0000000001000010	**B**
After Mutation	0000000001000011	**C**

Table 6: Sample 2 of mutation process:

	Chromosome representation	Plaintext character
Before Mutation	0000101110000101	அ
After Mutation	0000101110000110	ஆ

3.6 Termination Condition

If the fitness value is one then the chromosomes represent the UNICODE representation of the character which has been converted into cipher text using knapsack cipher, else the genetic evolution process take place until it obtains the optimal solution which represents the character.

4. EXPERIMENTAL RESULTS

Several sessions of experiment are conducted with sample test cases from different regional languages. In this paper the details of the experimental results are given for the following plaintext.

"The quick brown fox jumps on lazy dog."

The results are obtained by varying population size, crossover type, crossover probability, mutation probability, reproduction rate, and various sizes in tournament selection. It is analyzed to find optimal solution to attack knapsack cipher using Genetic Algorithm. The implementation work is carried out in ECJ simulator [22]. The results present the time taken for attacking the plaintext considered (38 characters) and maximum number of generations (This corresponds to the greatest number of generations that the character (any character) taken to find the solution. For example, in one run 'q' may take maximum number of generations while in another 'b' may take the maximum number of generations. Here only the number of generations is considered irrespective of the character.),

4.1 Effect of Population Size

Different sizes of initial population taken for analysis are 10, 50, 100, 500, 1000, 5000, 10000 and 50000. The details are tabulated in Table 7 and depicted graphically in Fig.2 and Fig. 3. The output shows that if initial population increases the number of generations to obtain optimal solution for attacking the plaintext decreases which in turn decrease the time taken to arrive at the solution. Table 7 shows that initial population is inversely proportional to maximum number of generations. The maximum number of generations given in this test case is 20000. For this crossover type used is single point crossover and tournament selection is applied. The optimal initial population obtained in this experiment is 5000. It is used in further analysis.

Table 7: Experimental Results of Effect of Population Size

Initial Population	**Generation**	**Time in min (m:s:ms)**
10	14121	2:22:41
50	2717	0:30:19
100	2449	0:32:46
500	374	0:09:51
1000	292	0:14:23
5000	20	0:07:56
10000	23	0:14:22
50000	4	0:12:44

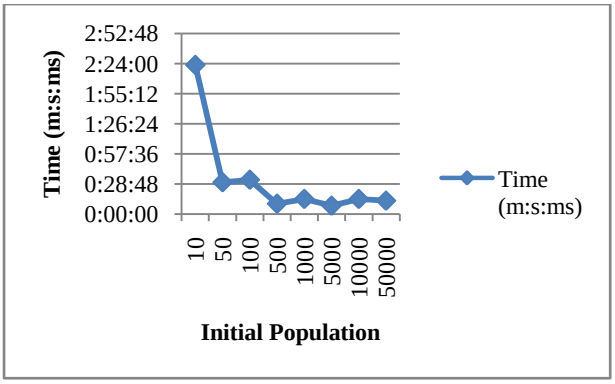

Fig. 2: Effect of Initial Population on time

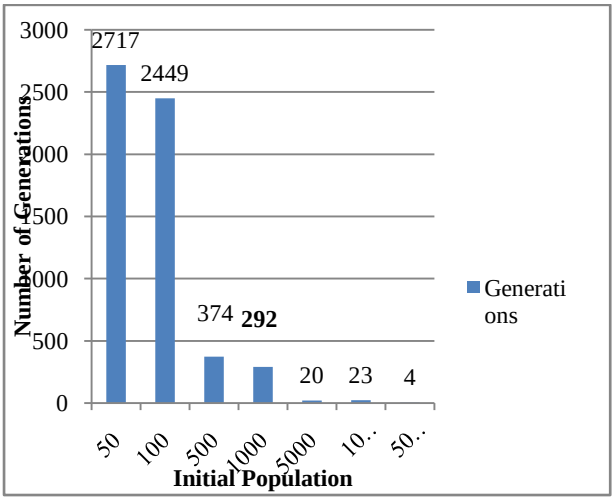

Fig. 3: Effect of Initial Population on generations

4.2 Effect of Genetic Operator Probabilities

The various genetic operators taken into consideration are crossover probability, mutation probability and reproduction probability. The different sets of values given for these parameters are from 0.1 to 0.9. As already mentioned initial population is set to 5000. Tournament selection and one point crossover is applied. The effect of genetic operator probabilities is tabulated in Table 8. The details of effect on time and generation are depicted graphically in Fig. 4 and Fig. 5 respectively. Crossover probability increases number of generation decreases and time also decreases. But mutation probability acts in reverse way. If it increases number of generation increases and time also increases. From this test it is found that crossover probability 0.9 and mutation probability 0.1 produces optimal result which is applied in next experiments

Table 8: Experimental Results of Effect of Genetic Operators

Initial population =5000, Cross over type=one point, Selection=tournament

	CP	MP	RP	Generations	Time in min (m:s:ms)
CASE 1	0.9	0.1	0.0	20	0:07:56
CASE 2	0.8	0.1	0.1	33	0:08:06
CASE 3	0.1	0.8	0.1	48	0:11:11
CASE 4	0.5	0.5	0.0	40	0:12:04
CASE 5	0.1	0.9	0.0	25	0:10:30

where,

CP – Crossover Probability MP – Mutation Probability RP – Reproduction Probability

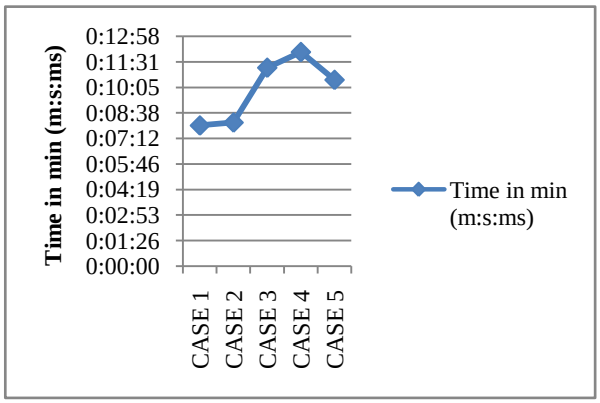

Fig. 4 : Effect of GA Operators' Parameters on time

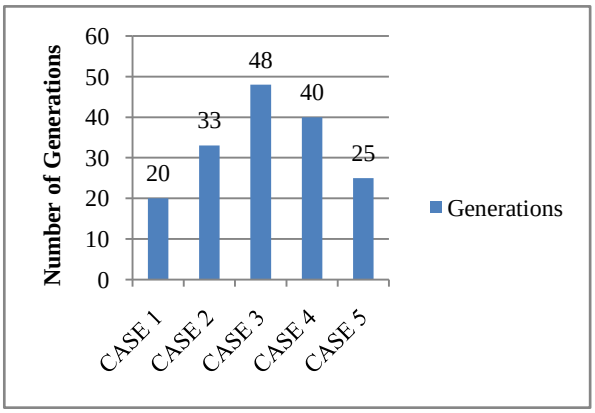

Fig. 5: Effect of GA Operators' Parameters on generations

4.3 Effect of Crossover Type

There are three kinds of crossover types explored. They are single point, two point and multi point crossover. With initial population 5000, crossover rate 0.9 and mutation rate 0.1 different crossover types are applied and outputs are obtained. The details are tabulated in Table 9 and are graphically depicted in Fig. 6 and Fig. 7. From the result it is found that one point crossover has high number of generation and maximum time whereas two point crossover has high number

of generation and less minimum time and multi point has less number of generations and time is less compared to one point crossover.

Table 9: Experimental Results of Effect of Crossover Type

Initial population=5000, Crossover probability=0.9, Mutation probability=0.1, Selection=tournament

Crossover Type	Generations	Time in min (m:s:ms)
One point	20	0:07:56
Two point	20	0:06:56
Multi point	16	0:07:18

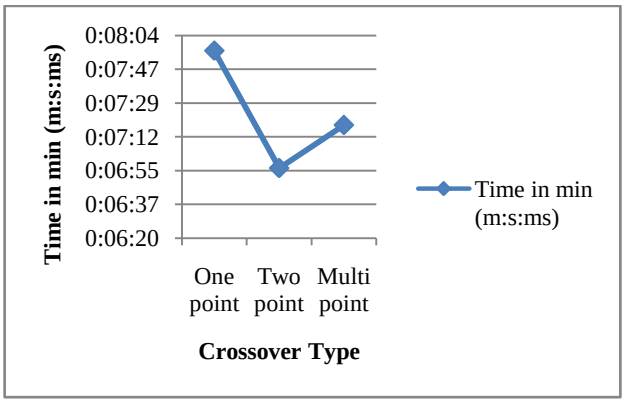

Fig. 6: Effect of Crossover Type on time

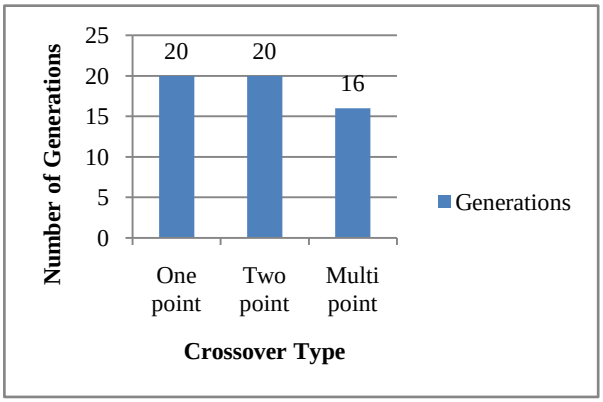

Fig. 7: Effect of Crossover Type on generations

4.4 Effect of tournament size

The selection method used is tournament selection. Different sizes of tournament can be selected. Since the above experiment shows that one point crossover is not suitable to obtain optimal solution it is applied for two point crossover type. Tournament sizes used to analyze the impact are 2, 3, 4, 5 and 6. The details are tabulated in Table 10 and graphically represented in Fig. 8 and Fig. 9. The output shows that tournament size 5 is an optimal way to obtain output.

Table 10: Experimental Results of Effect of Tournament Size

Tournament Size	Generations	Time in min (m:s:ms)
2	20	0:06:56
3	25	0:09:00
4	12	0:06:48
5	11	0:05:42
6	68	0:19:09

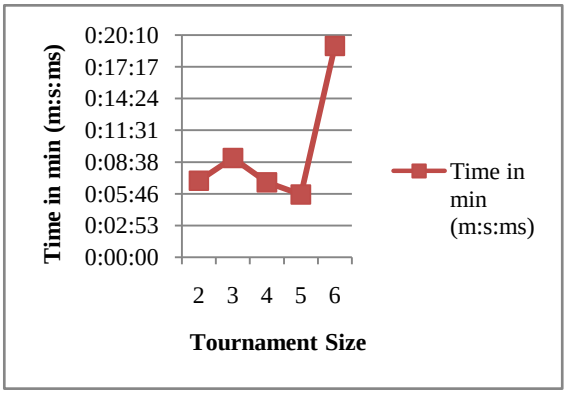

Fig. 8: Effect of Tournament Size on time

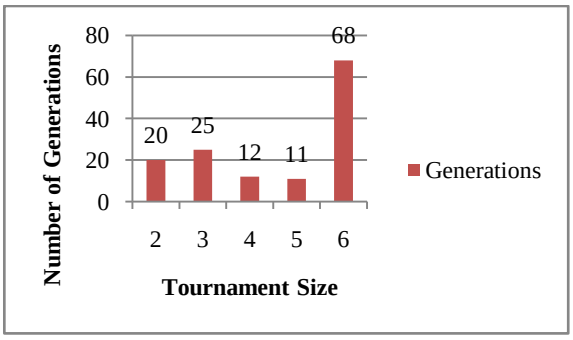

Fig. 9: Effect of Tournament Size on generations

Experiments are conducted with the test case and the effects of population size, effects of genetic operator probabilities, effects of crossover types, effects of tournament selection size are analyzed.

After a complete analysis on impacts of all the above parameters another test case with more than 100 characters was chosen and the optimal values for the parameters (found from the analysis) were set accordingly and runs were taken. The results are presented below.

The test case taken was:

India is my Country; all Indians are my brothers and sisters. I love my country and I am proud of its rich and varied heritage, I shall always strive to be worthy of it. I shall give my parents, teachers and all elders respect, and treat everyone with courtesy. To my country and its people, I pledge my devotion. In their well being and prosperity alone lies my happiness.

This test case contains 373 characters. Initial population of 5000, tournament selection with tournament size 5, two point crossover with a crossover probability of 0.9, mutation

probability of 0.1 is chosen as the parameters. The time taken for finding the plaintext for this test case was 45:88 seconds.

5. CONCLUSION

This paper presents the attack of knapsack cipher of knapsack sequence of size 16 using Genetic Algorithm. This leads to the cryptanalysis of plaintext encrypted using knapsack cipher of knapsack sequence size 16. So the plaintext can be multilingual and its length needed not be limited to 100. This helps a lot in cryptographic field to increase the strength of the knapsack cipher. It in turn provides high security of information in any regional language. An improvised solution has been proposed by tuning the various genetic parameters like initial population size, selection methods, crossover type, crossover rate, mutation rate, reproduction rate. These parameters are tuned and their effect is analyzed to get an optimal solution. From the experiments conducted it is found that optimal solution can be obtained if initial population size is 5000, crossover probability is 0.9, mutation probability is 0.1 crossover type that can be used is two point and tournament size is 5. Our test shows that initial population size is inversely proportional to number of generations and time. Crossover probability is also inversely proportional to number of generations and time. Thus Genetic Algorithm offers a powerful tool for the cryptanalysis of knapsack cipher. This work can be extended for other ciphers.

6. REFERENCES

[1] B. Delman, Genetic Algorithms in Cryptography, Master of Science Thesis, Rochester Institute of Technology, 2004.

[2] Clark.A, Dawson.Ed, and Bergen.H, Combinatorial Optimisation and the Knapsack Cipher, *Cryptologia*, Taylor & Francis, 1996;*20*(1), p. 85-93.

[3] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C., Introduction to Algorithms, Second Edition. Cambridge, Boston: MIT Press, McGraw-Hill, 2001

[4] Darwin, Charles Robert, The Origin of Species Vol. XI, The Harvard Classics. NewYork: P.F. Collier &Son, 1909-14

[5] David Kahn, The Codebreakers— The Story of Secret Writing, 1967, ISBN 0-684-83130-9.

[6] Goldberg, D. E, Genetic Algorithms in Search, Optimization, and Machine Learning, Boston: Addison-Wesley, 1989.

[7] Ingo Rechenberg, Evolutionsstrategie, Stuttgart, Frommann-Holzboog, 1994.

[8] John Holland, Adaption in Natural and Artificial Systems, Ann Arbor, MI: The University of Michigan Press, 1975.

[9] Kolodziejczyk.J, Miller.J and Phillips. P, The application of genetic algorithm in cryptoanalysis of knapsack cipher, In Krasnoproshin.V, Soldek.J, Ablameyko.S and Shmerko.V (Eds.), *Proceedings of Fourth International Conference PRIP '97 Pattern Recognition and Information Processing,* 1997, p. 394-401.

[10] Liddell and Scott's Greek-English Lexicon, Oxford University Press, 1984.

[11] Menezes, A., van Oorschot, P., & Vanstone. S., Handbook of Applied Crytography, Boca Raton: CRC Press, 1997.

[12] Poonam Garg and Aditya Shastri, An Improved Cryptanalytic Attack on Knapsack Cipher using Genetic Algorithm", *International Journal of Information Technology,* 2006;3.

[13] Raghavan Muthuregunathan, Divya Venkataraman, and Parthiban Rajasekaran, Cryptanalysis of Knapsack Cipher using Parallel Evolutionary Computing, *International Journal of Recent Trends in Engineering*, Academy Publishers, 2009;1(1), p. 260-263.

[14] Spillman Richard, Cryptanalysis of knapsack ciphers using genetic Algorithms, *Cryptologia*, Taylor & Francis, 1993;17(4), p. 367-377.

[15] Tomassini, M, Parallel and Distributed Evolutionary Algorithms: A Review, In K. Miettinen, M. M ¨ akel¨ a, P. Neittaanm¨aki and J. Periaux (Eds.), *Evolutionary Algorithms in Engineering and Computer Science*, Chichester: J. Wiley and Sons, 1999, p. 113 – 133,.

[16] Tom Davis, Cryptography, 2000. Available: http://www.geometer.org/mathcircles.

[17] Yaseen, I.F.T., & Sahasrabuddhe, H.V. (1999), A genetic algorithm for the cryptanalysis of Chor-Rivest knapsack public key cryptosystem (PKC), In *Proceedings of Third International Conference on Computational Intelligence and Multimedia Applications,* 1999, p. 81-85,.

[18] Tech-FAQ Available: http://www.tech-faq.com.

[19] Genetic Algorithm Available: http://en.wikipedia.org/wiki/Genetic_algorithm

[20] Genetic Algorithms Available: http://www.obitko.com/tutorials/

[21] Sinkov, A. "Elementary Cryptanalysis: A Mathematical Approach", New York: Random House, 1968.

[22] Sean Luke, Liviu Pnait, Gabriel Balan, Sean Paus, Zbigniew Skolicki, elena Popovici, Keith Sullivan, Joseph Harrison, Jeff Baskett, Rober Hubley, Alexander Chircop, "Evolutionary Computation System in Java". [Available at] http://cs.gmu.edu/~eclab/projects/ecj/