Reports of Enlarged Sessions of the
Seminar of I. Vekua Institute
of Applied Mathematics
Volume 30, 2016

THE USE OF GENETIC ALGORITHMS IN CRYPTANALYSIS

Zurab Kochladze Lali Beselia

Abstract. The article considers the possibility of using genetic algorithms in cryptanalysis, namely for cracking the Merkle-Hellman cryptosystem. The obtained analysis results lead us to conclusion that the use of genetic algorithms in cryptanalysis may be effective. For example, the genetic algorithm described in the article finds a cipher key faster than the well-known Shamir algorithm.

Keywords and phrases: A genetic algorithm, cryptanalysis of the Merkle-Hellman cryptosystem.

AMS subject classification (2010): 74K25, 74B20.

1 Introduction. In 1978 the famous Merkle-Hellman article [1] was published, which described an open key (asymmetric) cryptosystem based on a concrete case of the knapsack problem [2]. We can formulate it as follows: there is a knapsack of V volume and a set of $B = \{b_1, b_2, ..., b_n\}$ subjects, which have certain volumes. Our goal is to find such $B_i \subseteq B$ subset of B set, for the elements of which the following equation is worked out:

$$V = \sum_{i=1}^{n} b_i x_i, \tag{1}$$

where $x_i \in \{0, 1\}$, $(i = 1, 2, ..., n)$ in case $x_i = 1$. This means that we should place i subject into the knapsack, and in case $x_i = 0$-, then we should not place the subject into the knapsack. As it is known [2], the knapsack problem belongs to the NP grade task groups. However, in this concrete case, if B set is an extremely increasing sequence, i.e. each b_i member of the sequence fulfils the condition

$$b_i > \sum_{j=1}^{i-1} b_j,$$

then there is a linear algorithm [2] for the solution of the problem.

Using this feature, Merkle and Hellman developed an open key cryptosystem, in which the open cipher key is $A = \{a_1, a_2, ..., a_n\}$ non-extremely increasing sequence, where each member of sequence is obtained by the following rule:

$$a_i = b_i \cdot t(\mathrm{mod} m), \tag{2}$$

where $m, t \in Z$ and the following conditions are fulfilled: $m > \sum_{i=1}^{n} b_i, \quad (t, m) = 1.$

The key of the cipher is the (B, m, t) three. The open text, which represents a sequence of zeroes and ones, during encryption is divided into a block of n length and L quantity and acts as $x_i \in \{0, 1\}$ set. The encrypted text is represented with $S_1, S_2, ..., S_L$ sums, which are calculated by the formula:

$$S_j = \sum_{i=1}^{n} x_{ij} \cdot a_i. \tag{3}$$

For restoring the open text it is necessary to solve the above mentioned version of the knapsack problem by a linear algorithm when B extremely increasing sequence and m and t parameters are known. For this purpose each sum is multiplied by t^{-1} module m

$$S'_j = S_j \cdot t^{-1} (\mathrm{mod} m) \tag{4}$$

and the knapsack problem is solved by the above mentioned linear algorithm for each S'_j sum separately, when B extremely increasing sequence is known.

At a glance this system seemed to be protected from any cyber attacks and was the fastest open key system, the use of which was possible for encryption of vast texts. However, it turned out to have certain trapdoors [3], by use of which famous cryptologist A. Shamir created the polynomial algorithm and cracked the system [4].

2 Using the genetic key to crack the Merkle-Hellman algorithm. Genetic algorithms were first used for the solution of optimization problems. After some time they were also used in different fields of sciences. Use of genetic algorithms for cryptanalysis of cryptographic algorithms is a new trend, which has not yet developed in practical cryptology. There are several hundreds of works, the authors of which make their efforts to show that this approach may have advantages compared to other ones. In regard to this we tried to crack the already cracked Merkle-Hellman cryptosystem by means of genetic algorithms. Then we compared the results obtained by us to the results obtained by the Shamir algorithm. There are several works, where the Merkle-Hellman cryptosystem is cracked by genetic algorithms, though in all cases the attack is carried out on the basis of ciphertext [7]. Unlike these works and like the Shamir method, we search for the cipher key by attacking against the open key. We elaborated new heuristical methods, by which made the use of genetic algorithms more precise and faster. The research results and software given in this article may be used for cryptanalysis of other asymmetric cryptosystems as well.

3 Formulation of the problem. Our method of attack is quite different from the methods used in the above mentioned works. Besides, we created a genetic algorithm quite different from other genetic algorithms (different in the selection criterion and crossover process).

Our genetic algorithm is described in file “genetic2.h” created by us. In “genetic” class of the file four functions are described: the fitness function *(bool fitness (vector < populatcia > &v)), the crossover function (void crossover (vector < populatcia > &v)),*

the mutation function (void fitness$(vector < populatcia > \&v)$) and the selection function (void seleKtcia$(vector < populatcia > \&v)$).

a) The fitness function determines the extreme increase in each member (solution-candidate) of the population transmitted to it.

b) The selection function chooses the selection-candidates, which the most fulfill the fitness function, i.e. their fitness values are higher than those of others. In case the population size is L we choose only $L/5$ solution-candidates. Exactly these solution-candidates form new generations.

c) The crossover function receives the population of the solution-candidates. From this population we choose solution-candidates with $t1$ and $t2$ numbers in pairs by means of a random generator taking into account that $t1$ and $t2$ do not coincide with each other and the used pair is not repeated.

d) The mutation function changes one byte of each solution-candidate.

Our goal is, using the above described algorithm, to find such (u, m) pair, by which we will be able to find the extremely increasing sequence by the following formula:

$$b_i = a_i u (\mathrm{mod} m), \tag{5}$$

where $u = t^{-1}$.

The algorithm is realized on $C++$ language base. It consists of the preparation and main parts. In the preparation part the information-to-be-transmitted is ciphered by the Merkle-Hellman algorithm. We took $\{b_1, b_2, ..., b_n\}$ extremely increasing sequence, m module root and selected t multiplier, by means of which we calculated open key $a_i = b_i \cdot t(\mathrm{mod} m)$ and ciphered the information-to-be-transmitted by (3) formula.

The working chart of the main algorithms is as follows:

1. The initial population is represented by m root, which is initialized by random generator (it is represented in binary system). The size of each member (solution-candidate) of the population is $d * n$, where n is equal to the length of the open key and $d = 2$;

The solution-candidates are transformed into binary system.

2. Like the Shamir algorithm we take the first four members of the open key and calculate the inverse of t multiplier by m root $u = p * m/a_i$, where $u = t^{-1}$, $1 \leq p \leq a_i$, $1 \leq i < 4$. Thus, we receive the population all probable multipliers. We set limits for selecting u multiplier.

3. We determine the criterion for selection by the fitness function. The value of the fitness function is less than n, we pass to the following phase.

4. By means of the crossover function we carry out the crossover operation for the chosen solution-candidates.

5. For the received solution-candidates the second, third and fourth phases are repeated. In case the fitness function of any solution-candidate is equal to n, it means the desired result is obtained and the program stops functioning. Otherwise, we pass to the following phase.

6. The selection function chooses the $L/5$ (L is the size of the initial population) number solution-candidates, the fitness functions of which are higher.

7. We have indicated that the process will repeat 10 times. If this process is repeated, 10 times and we don't get the desired result, only in this case we use a mutation, or change the function of the gene, and then repeat the 2nd, 3rd and 4th steps. When We get the desired results, we stop working. But tests showed that non f the mutations feature is not needed, and the hybridization of a maximum of 5 times using we get the desired result.

By the use of genetic algorithms the Merkle-Hellman cryptosystem is cracked quite quickly. Consequently, we may conclude that use of genetic algorithms will be useful for cryptanalysis of other asymmetric cryptosystems as well.

REFERENCES

1. Merkle, R.C., Hellman, M.E. Hiding information and signatures in trap door knapsacks. Inform. *IEEE Trans. on Info. Theory,* September, **IT-24** (1978), 525-536.
2. Silvano, M., Paolo, T. Knapsack Problems: Algorithms and Computer Implementations. *Editor John Wiley & Sons*, 1990.
3. Salomaa, A. Public-Key Cryptography. *EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Berlin*, **23**, 1990.
4. Garg, P., Shastri, A. An improved cryptanalytic attack on knapsack cipher using genetic algorithm. *International Journal of Information Technology* **3**, 3 (2007), 145-152.
5. Muthuregunathan, R., Vekataraman, D., Rajasekaran, P. Cryptanalysis of knapsack cipher using parallel evolutionary computing. *International Journal of Recent Trends in Engineering*, **1**, 1 (2009), 260-263.
6. Shamir, A. A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory*, September **IT-30**, 5 (1984), 699-704.
7. Ramani, R. Geetha; Balasubramanian, Lakshmi. Genetic algorithm solution for cryptanalysis of knapsack cipher with knapsack sequence of size 16. *International Journal of Computer Applications (0975 8887),* **35**, 11 (2011), 17-23.

Received 17.25.2016; revised 28.10.2016; accepted 15.12.2016.

Author(s) address(es):

Zurab Kochladze
Iv. Javakhishvili Tbilisi State University
University str. 2, 0186 Tbilisi, Georgia
E-mail: zurab.kochladze@tsu.ge

Lali Beselia
Sokhumi State University
Anna Politkovskaia str. 9, 0186 Tbilisi, Georgia
E-mail: lalibeselia@mail.ru