Journal of Engineering and Applied Sciences 15 (1): 267-272, 2020
ISSN: 1816-949X

Cryptoanalytic Method of Searching for the Secret Key and ITS Length on the Basis of Evolution Metaheuristics

Hussein Al-Ofeishat
Department of Computer Engineering, Al-Balqa Applied University, Jordan

Abstract: This study mainly studies a cryptoanalytic method of searching for the secret key and its length, in this research, a review of works devoted to solving the cryptanalysis problem of classic and asymmetric encryption algorithms based on new technologies of artificial intelligence-bio-inspiration methods simulating the processes of evolution of wildlife was conducted. The main distinctive features of the application of these methods are described and experimental results are shown which demonstrate the possibility of using these methods for solving cryptanalysis problems.

Key words: Cryptanalytical methods, metaheuristics, natural systems, encryption algorithms, possibility, evolution

INTRODUCTION

Currently, when developing computer technologies that provide information security and information protection, cryptographic methods are widely used. To implement these methods, algorithms based on natural systems are used: Genetic Algorithms (GA) and algorithms of swarm intelligence. The cryptanalytical methods of searching for a secret key and its length also include evolutionary methods (Surakhi *et al*., 2017). In models and algorithms of evolutionary computation, the main idea is to create an initial model and rules which allows the model to change (evolve) (Law and Kelton, 2004). The analysis of literature sources has shown that during the past several years, different schemes of evolutionary calculations have been considered including genetic algorithms, genetic programming, evolutionary strategies and evolutionary programming.

Often, the convergence of the evolutionary algorithm requires a large number of calculations of the Objective Function (CF) which increases the execution time of the problem. To increase the speed of processing, instead of simulation models, metamodels use approximate mathematical models which are obtained as a result of experiments with a model of the system (Glover and Kochenberger, 2003).

In this study, metaheuristics and methods of constructing metamodels will be examined as well as an approach to integrate metamodels with evolutionary metaheuristics to find the secret key and its length (Al Ofeishat and Al-Rababah, 2009).

Statement of the main material: The algorithm for clonal selection which is based on the theory of clonal selection, proposed by Burnet to describe the behaviour and the ability of antibodies in the immune system will be considered. Based on the principles of the evolutionary theory of Darwin's natural selection, the theory of clonal selection suggests that lymphocytes (B-cells and T-cells) are used to destroy or neutralize an antigen (pathogenic microorganism). When a lymphocyte is selected and bound to an antigen, it multiplies and differentiates into plasma cells and memory cells. Plasma cells have a short lifespan and produce a large number of antibody molecules. The memory cells live for a long period, expecting the same antigen in the future. An important feature of the theory is that when a cell is selected and cloned, these clones undergo mutations which increases the effectiveness of antigen challenge (Dubrov *et al*., 2013).

The theory of clonal selection suggests that the immune system can change (the structure and specific gravity of the cells) in accordance with the environment. Through the blind selection process and the accumulation of changes, the immune system can acquire the necessary information to protect the human body from certain pathogenic environmental hazards. It has been suggested that the immune system expects a certain pathogenic microorganism.

The clonal selection algorithm, proposed by de Castro and von Zuben, minimizes the goal function. The choice of antibodies is based on affinity (proximity) which is based on the goal function. Selected antibodies are subjected to cloning and then mutations of proportional affinity (proximity). The mutated clone set competes with the existing antibody population for membership in the next generation. In addition, members of the population with low affinity (the farthest) are replaced by randomly generated antibodies.

Unlike the genetic algorithm, this algorithm does not use the crossover operator. Figure 1 shows the structure of the algorithm for clonal selection.

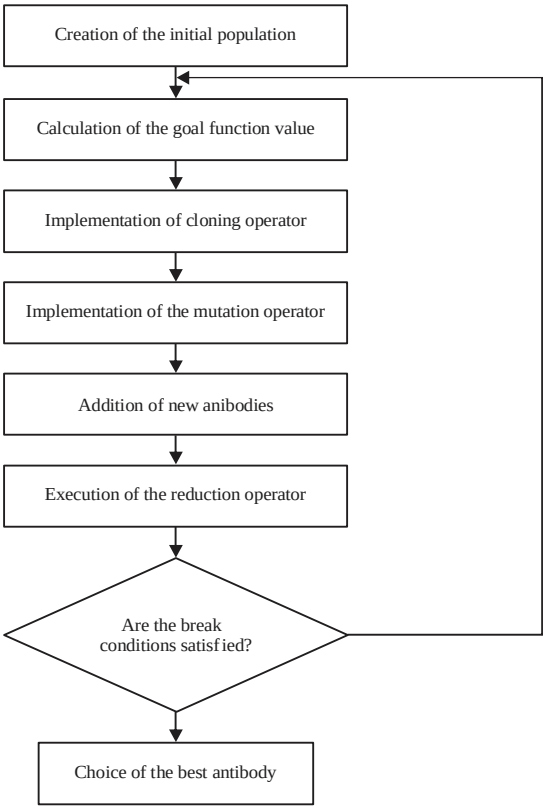

Fig. 1: Structure of the algorithm for clonal selection

To solve the problem of finding a secret key, vertices are used as components and solutions are used as antibodies. There are three main methods for creating a population:

- Strategy of "blankets" (formation of the complete population). Cannot be practically implemented due to the large computational complexity
- Strategy of "shotgun" (the formation of a sufficiently large subset of the total population) is used the most
- Strategy of "focusing" (formation of a population from variations of one solution)

MATERIALS AND METHODS

If there is an assumption regarding the solution, in this case, the algorithm will start working in the vicinity of the optimum. The goal function determines the fitness of the antibody in the population. At each iteration of the algorithm for clonal selection, the fitness of each antibody of the population is estimated using the goal function. In the case of finding the minimum of the function:

$$f(x),\ x \in \left[x_{min},\ x_{max} \right] \tag{1}$$

The goal function is represented in the form:

$$F(x) = f(x) \rightarrow \min_{x} \tag{2}$$

In the case of finding the optimal route, the value of the target function for the i-th antibody is calculated as the cost of the solution, i.e., the length of the secret key defined by the set of vertices xi:

$$F(x_k) = d_{x_{kM},\, x_{k1}} + \sum d_{x_{kj},\, x_{k,j+1}},\ k \in \overline{1,K} \tag{3}$$

Where:

M : The number of components of antibodies (vertices)
K : The number of antibodies (solutions)
$d_{x_{kj},\, x_{k,j+1}}$: The weight of the rib

In the early stages of the operation of the clonal selection algorithm, a random scheme (random selection of antibodies) is used to ensure the study of the entire search space. In the final stages, a selective scheme is used that creates the search (the current best antibodies are preserved). This combination does not require scaling and can be used to minimize the target function.

The probability of selecting a circuit on the basis of random selection is determined by simulated annealing as in the following:

$$p_r = p_0 \exp(-1/g(n)),\quad g(n) = \beta g(n-1),\ 0<\beta<1,\ g(0) = T_0,\ T_0 > 0 \tag{4}$$

where, p_0 is the initial probability of reduction. Probability selection based on a random selection scheme is determined by simulated annealing as:

$$p_r = p_0(1-\exp(-1/g(n))),\quad g(n) = \beta g(n-1),\ 0<\beta<1,\ g(0) = T_0,\ T_0 > 0 \tag{5}$$

For the simulation system, a combination of values of the input factors of the simulation model is determined which allows a maximum/minimum of a certain response of the random variable to be achieved. The response function is almost impossible to calculate analytically however, it can be calculated by running a system model.

The multi-extremity of the model response functions and the multidimensionality of the secret key search space and its length determine the active and efficient use of metaheuristic methods (Dubrov *et al*., 2013) as optimizers for problems of this type. Evolutionary Metaheuristics (EM) are often used, namely, Genetic algorithms and evolutionary strategies.

The Genetic Algorithm (GA) is a heuristic search algorithm which is used to solve optimization and modelling problems by sequential selection, combination and variation in the required key parameters using mechanisms that resemble biological evolution.

When using encrypting tables, the key can be considered a permutation (p_1, p_2, …, p_n). Therefore, the chromosome in the GA must also specify a permutation. It should also be understood how to implement the representation of individual genes of an individual. In the elementary case, encryption can be performed by assigning the corresponding genes to the individual elements of the key, i.e., the i-th gene of chromosome P is the element p_i.

Haykin (1999) deficiencies in this approach were noted as obtained genes are dependent on each other which leads to the possibility of obtaining incorrect solutions. Surakhi *et al*. (2017) and Dubrov *et al*. (2013) proposed an alternative approach to solving similar problems. This approach involves the use of an intermediate representation of a set of genes through some rule or object from which the key is formed. In this case, an important task is to define an intermediate solution which is represented as a bit string for the use of standard genetic operators.

When implementing cryptanalytical GA, in practice, an approach is used in which the key elements are considered the genes of an individual. To avoid obtaining incorrect solutions for decimal chromosome coding, the rule is applied as follows: When the same genes appear on the chromosome, the second repeating gene is replaced with the missing gene. To determine the secret key as a function of the fitness of individuals, the coincidence of plaintext and ciphertext is used. As an objective function, one can use the Jacobsen function (Haykin, 1999; Al-Ofeishat, 2012; Gräning *et al*., 2007) on the distribution of bigram frequencies in plaintext.

An interesting development in the field of swarm intelligence is the bee algorithm which has successfully been used to find extremes of complex multidimensional functions. The algorithm of the cryptanalytical method of searching for a secret key and its length on the basis of a bee algorithm was considered in Branke and Schmidt (2004) and Jaskowski and Kotlowski (2008) where the implementation of the basic steps of the bee algorithm was proposed and a demonstration example of the cryptanalysis algorithm implementation was given.

An analysis of the results obtained in Afonin (2011) demonstrates that with increasing length of the key to the real key, applying only the genetic algorithm does not provide the expected result, regardless of the change in the error between the plaintext and the decrypted keys. An algorithm for calculating the secret key is proposed in this study. It consists of two stages.

- First, the preparation, in which the encoding and decoding of the text occur
- Second, directly calculating the secret key from an open text with the help of an attack based on the known plaintext and using a genetic algorithm

The proposed algorithm consists of the following step:

- The initial population is formed randomly
- The crossing procedure gives m×n÷2 new keys. For parent rows, the point of division is randomly selected and the descendants are obtained by exchanging the cut-off parts
- The mutation operator is applied to the obtained generation. The bit of the individual of the population is inverted with a certain probability. Crossover and mutation are repeated several times
- Of the new members of the population, i.e., from public keys, private keys and corresponding coefficients were found. With their help, the text is deciphered
- The fitness function is the error between the open and decrypted text

The algorithm ends when the error has a value close to zero. Evolutionary Strategies (ES) (Jin *et al*., 2003; Afonin, 2009), in contrast to genetic algorithms, analyse the course of evolution at the phenotype level. In ES, each individual is characterized by a fitness function and a chromosome line. The fitness Function (FP) depends on the Objective Function (OF) of the problem.

In the ES algorithms, the values of the mutation step and the rotation angle are adapted, the main operator is the mutation operator, implemented by means of the normal distribution law.

There are three main approaches to the integration of metamodels with evolutionary metaheuristics for finding the secret key and its length: polynomials, kriging and neural networks (Takialddin *et al*., 2017; Chernyshev *et al*., 2014; Al-Ofeishat *et al*., 2018).

To find the secret key, it is optimal to use polynomials of the second degree. Calculation of the unknown coefficients of the polynomial is carried out by the method of least squares or the gradient method. The main drawback to this approach is the considerable time required to calculate the coefficients in the case of a long key length. Kriging is a combination of the global model and local “deviations”:

$$(x) = (x)+Z(x) \quad (6)$$

where (x) is the global component of the model of the objective function which is specified by the polynomial and Z(x) is the Gaussian function with zero expectation and covariance which simulates local deviations from the global model.

Calculation of model parameters is realized using the maximum likelihood method. The main advantage of kriging is that with its help, the calculation of the confidence interval is carried out without additional calculations (Jain and Chaudhari, 2018). However, the need to perform matrix transformations to calculate the model yield significantly increases the computation time with increasing dimensionality of the problem.

Neural networks are a "powerful" device for approximating complex dependencies (Jain and Chaudhari, 2019). Most often, three types of networks are used: multilayer perceptron, a network based on radial basis functions and a support vector machine. To improve the efficiency of solving the problem of searching for a secret key using a multilayer perceptron, modifications of the BP algorithm and methods for optimizing the network structure for a particular task are used.

For multilayer perceptron, training is based on error correction (training with the teacher) with the most commonly used algorithm being reverse Propagation (BP) which is an iterative gradient learning algorithm that provides minimization of the root-mean-square error.

Backward propagation algorithm

- Number of iterations of training n = 1, initialization by uniform distribution on the interval (0,1) or [-0.5, 0.5] of displacements (thresholds) $b_j^{(k)}(n)$ and weights $w_{ij}^{(k)}(n)$:

$$i \in \overline{1, N^{(k-1)}}, \quad j \in \overline{1, N^{(k)}}, \quad k \in \overline{1, L} \tag{7}$$

where, $N^{(k)}$ is the number of neurons in the k-th layer and L is the number of layers

- Set the training set:

$$\left\{(x_\mu, d_\mu) \mid x_\mu \in R^{N^{(0)}}, d_\mu \in R^{N^{(L)}}\right\}, \mu \in \overline{1, P} \tag{8}$$

where, x_μ is the μ-th training input vector, d_μ is the μ-th training output vector, $N^{(0)}$ is the number of neurons in the input layer, $N^{(L)}$ is the number of neurons in the output layer and P is the power of the learning set. The number of the current pair from the training set is μ = 1.

- Calculation of the output signal for each layer (straight run):

$$y_i^{(0)}(n) = x_{\mu i} \tag{9}$$

$$y_i^{(k)}(n) = f^{(k)}(s_j^{(k)}(n)), s_j^{(k)}(n) = \sum_{i=0}^{N^{(k-1)}} w_{ij}^{(k)}(n) y_i^{(k-1)}(n), \quad j \in \overline{1, N^{(k)}}, k \in \overline{1, L} \tag{10}$$

where, $N^{(k)}$ is the number of neurons in the k-th layer, k is the layer number, L is the number of layers, $w_{ij}^{(k)}(n)$ is the weight of the connection from the i-th neuron to the j-th neuron on the k-th layer at time n, $y_j^{(k)}(n)$ is the output of the j-th neuron on the k-th layer and $f^{(k)}$ is the activation function of neurons of the k-th layer. It is believed that:

$$w_{0j}^{(k)}(n) = b_j^{(k)}(n), \quad y_0^{(k-1)}(n) = 1 \tag{11}$$

- Calculation of the energy of the ANN error:

$$E(n) = \frac{1}{2}\sum_{j=1}^{N^{(L)}} e_j^2(n), e_j(n) = y_j^{(L)}(n) - d_{\mu j} \tag{12}$$

- Adjustment of the synaptic weights based on the generalized delta rule (reverse run):

$$w_{ij}^{(k)}(n+1) = w_{ij}^{(k)}(n) - \eta \frac{\partial E(n)}{\partial w_{ij}^{(k)}(n)} \tag{13}$$

where, η is the parameter that determines the speed of training (for large η training is faster but the risk of obtaining the wrong decision increases), 0 <η< 1:

$$\frac{\partial E(n)}{\partial w_{ij}^{(k)}(n)} = y_i^{(k-1)}(n) g_j^{(k)}(n), \; i \in \overline{0, N^{(k-1)}}, k \in \overline{1, L-1} \tag{14}$$

$$g_j^{(k)}(n) = \begin{cases} f'^{(L)}(s_j^{(L)}(n))(y_j^{(L)}(n) - d_{\mu j}), & k = L \\ f'^{(k)}(s_j^{(k)}(n)) \sum_{l=1}^{N^{(k+1)}} w_{jl}^{(k+1)}(n) g_l^{(k+1)}(n), & k < L \end{cases} \tag{15}$$

- Checking the termination condition
If n mod p>0, then μ = μ+1, n = n+1 and transition to 3.
If n mod P = 0 mo and

$$\frac{1}{P}\sum_{s=1}^{P} E(n - P + s) > \varepsilon. \tag{16}$$

= then n = n+1 and transition to 2.
If n mod P = 0 and:

$$\frac{1}{P}\sum_{s=1}^{P} E(n - P + s) < \varepsilon \tag{17}$$

then end

RESULTS AND DISCUSSION

In the literature, there is an opinion (Haykin, 1999; Al-Ofeishat, 2012) that when using metamodels in evolutionary algorithms, it is important that there is not an error in the metamodels approximation but the correct selection.

The random nature of the objective function can have a negative effect on the operation of the EM, leading to wandering of the search process, loss of the rate of convergence of the algorithm and falling into local optima. The main approach is to run a series of runs of the model for one individual solution and calculate the average value of the OP. It is believed that the main focus is the work of the selection operator EM in selecting individuals in the next generation (Lasry, 2018). The accuracy of the metamodel varies from generation to generation due to a change in the location of the current

population in the search space and changes in the data used to build the metamodel. Therefore, predicting the number of control individuals in the next generation based on the quality of the metamodel which is calculated for the current generation, may be erroneous. The rank correlation prank (Ariffin *et al*., 2019) is used as the criteria for assessing the quality of the metamodel which depends on the difference in the ranks (numbers in the sorted by the FP list) of individuals who are calculated using the objective function. If it were possible to first assess the quality of the metamodel in the current generation and then use this estimate to determine the controlled individuals in the same generation, then this method could be effectively used to correctly select the individuals in the next generation.

Consider the task of developing hybrid metaheuristics that use not one but several model-oriented algorithms. The number of algorithms is called basic. Each algorithm has its own model and as a result of their work, we obtain one or several solutions.

Let the algorithms be performed asynchronously and their interaction at iteration h occurs by forming metamodels considering individual models and the model generated by the previous iteration and generated by the decision algorithms.

The key stages of metaheuristics are considered. After the initialization phase which is controlled by metaheuristics, all the algorithms of the model are launched. The solutions are independently generated in steps and each algorithm updates its own model.

When the specified exchange conditions are met, the current models of the basic algorithms (Possibly, the corresponding solution variants) are used by control metaheuristics to form a new aggregated model. In this case, the previous aggregated model can also be used and the process of formation can be represented as an optimization problem of finding the best element in the model space.

Next, the generated aggregated model is sent to the basic algorithms where it can both be used in combination with their own model and replace it completely. When the completion condition is met, the metaheuristics return one or more of the best solutions found by the underlying algorithms.

The results of computational experiments indicate that this methodology increases the effectiveness of model-oriented algorithms, although it may require the development of more complex methods for aggregating models.

CONCLUSION

The methodology considered in the article for constructing the meta-search for a secret key allows us to diversify the work of the basic algorithms and reduce the probability of completing the search in areas that do not contain a global solution. The exchange of information between the basic algorithms creates the prerequisites for improving the efficiency of the search process which is of a global nature. Thus, the choice of a specific information exchange scheme (the method of co-operation) determines the balance between intensification and diversification of the search.

REFERENCES

Afonin, P.V., 2011. Construction of hybrid systems of simulation based on evolutionary metaevristics and neural networks. Survey of applied and industrial mathematics. Construction of hybrid systems of simulation based on evolutionary metaevristics and neural networks. Survey of applied and industrial mathematics. (FSSCIT'11), Moscow, pp: 38-39.

Afonin, P.V., 2009. Evolution control method for evolutionary strategies based on neural network metamodels. Proceedings of the 5th International Scientific and Practical Conference on Integrated Models and Soft Computing in Artificial Intelligence, May 28-30, 2009, Kolomna, pp: 563-574.

Al Ofeishat, H.A. and A.A. Al-Rababah, 2009. Real-time programming platforms in the mainstream environments. Int. J. Comput. Sci. Network Secur., 9: 197-204.

Al-Ofeishat, H., E. Trad and T. Al Smadi, 2018. Proactive algorithm dynamic mobile structure of Routing protocols of ad hoc networks. Int. J. Comput. Sci. Network Secur., 18: 86-90.

Al-Ofeishat, H.A., 2012. Scheduling in heterogeneous grid-systems. Aust. J. Basic Applied Sci., 6: 1-10.

Ariffin, M.R.K., S.I. Abubakar, F. Yunos and M.A. Asbullah, 2019. New cryptanalytic attack on RSA modulus N = pq using small prime difference method. Cryptography, Vol. 3, No. 1. 10.3390/cryptography3010002

Branke, J. and C. Schmidt, 2004. Sequential Sampling in Noisy Environments. In: Parallel Problem Solving from Nature-PPSN VIII. PPSN 2004, Yao, X., E.K. Burke, J.A. Lozano, J. Smith and J.J. Merelo-Guervós et al., (Eds.)., Lecture Notes in Computer Science, Vol. 3242. Springer, Berlin, Heidelberg, ISBN: 978-3-540-30217-9, pp: 202-211.

Chernyshev, Y.O., A.S. Sergeev and E.O. Dubrov, 2014. Application of bioinspired optimization algorithms for cryptanalysis of classical and asymmetric cryptosystems. Proceedings of the 14th International Scientific and Methodical Conference Informatics: Problems, Methodology, Technologies, (IPMT'14), VSU.-Voronezh, pp: 206-210.

Dubrov, E.O., A.N. Ryazanov, A.S. Sergeev and Y.O. Chernyshev, 2013. Development of methods for cryptanalysis of the ciphers of permutations and replacements in information security systems based on evolutionary optimization methods. Proceedings of a Scientific Conference Dedicated to the Day of Radio on Radio Electronic Devices and Systems for Infocommunication Technologies, (DDR'13), Moscow, pp: 220-224.

Glover, F. and G.A. Kochenberger, 2003. Handbook of Metaheuristics. 1st Edn., Kluer Academic Publishers, USA., ISBN-13: 978-1402072635, Pages: 570.

Gräning, L., Y. Jin and B. Sendhoff, 2007. Individual-based Management of Meta-models for Evolutionary Optimization with Application to Three-Dimensional Blade Optimization. In: Evolu-tionary Computation in Dynamic and Uncertain Environments, Yang, S.X., Y.S. Ong and Y.C. Jin (Eds.). Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-49772-1, pp: 225-250.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. 2nd Edn., Prentice-Hall International Inc., New Jersey, USA., ISBN-13: 9780139083853, Pages: 842.

Jain, A. and N.S. Chaudhari, 2018. A novel cuckoo search strategy for automated cryptanalysis: A case study on the reduced complex knapsack cryptosystem. Int. J. Syst. Assurance Eng. Manage., 9: 942-961.

Jain, A. and N.S. Chaudhari, 2019. An improved genetic algorithm and a new discrete cuckoo algorithm for solving the classical substitution cipher. Int. J. Applied Metaheuristic Comput., 10: 109-110.

Jaskowski, W. and W. Kotlowski, 2008. On selecting the best individual in noisy environments. Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008, July 12-16, 2008, Atlanta, Georgia, USA., pp: 961-968.

Jin, Y., M. Huesken and B. Sendhoff, 2003. Quality measures for approximate models in evolutionary computation. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003), July 12-16, 2003, Chicago, pp: 170-173.

Lasry, G., 2018. A Methodology for the Cryptanalysis of Classical Ciphers with Search Metaheuristics. Kassel University Press GmbH, Kassel, Germany, ISBN: 978-3-7376-0458-1, Pages: 247.

Law, A.M. and W.D. Kelton, 2004. Simulation Modeling and Analysis. 3rd Edn., Publishing Group BHV, St. Petersburg, Russia, Pages: 847.

Surakhi, O.M., M. Qatawneh and H. Ofeishat, 2017. A parallel genetic algorithm for maximum flow problem. Int. J. Adv. Comput. Sci. Aplic., 8: 159-164.

Takialddin, A.S., K.A. Smadi and O.O. AL-Smadi, 2017. High-speed for data transmission in gsm networks based on cognitive radio. Am. J. Eng. Applied Sci., 10: 69-77.