Cryptanalytic Method of Searching for the Secret Key and Its Length on the Basis of Evolution Metaheuristics

Hussein AL-Ofeishat

Al-Balqa Applied University, Computer Engineering Department. Jordan

Abstract
This paper mainly studies, Crypto analytic Method of searching for the secret Key and its length**,** in this work a review of author's works devoted to solving the cryptanalysis problem of classical 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, experimental results are shown, which testify the possibility of using these methods for solving cryptanalysis problems.

Keywords:
Cryptanalytical Methods, Metaheuristics, Natural systems, Encryption algorithms

1. Introduction

Nowadays 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), algorithms of swarm intelligence, etc. The cryptanalytical methods of searching for a secret key and its length also include evolutionary methods [1].In models and algorithms of evolutionary computation, the main part is to create an initial model and rules, which allows the model to change (evolve) [2].The analysis of literature sources has shown that during the past several years different schemes of evolutionary calculations have been considered, including genetic algorithm, genetic programming, evolutionary strategies, 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 used an approximate mathematical models, which are obtained as a result of experiments with the model of the system [3]. In this article, Metaheuristics and ways of constructing metmodels will be examined, as will as approach to the integration of metamodels with evolutionary metheuristics to find the secret key and its length[4].

Manuscript received June 5, 2026
Manuscript revised June 20, 2026
https://doi.org/10.22937/IJCSNS.2026.26.6.15

2. 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 behavior 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 the 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 [5]. The theory of clonal selection suggests that the immune system is able to change itself (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 is capable of acquiring the necessary information to protect the human body from certain pathogenic environmental hazards. It is suggested that the immune system was expecting 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 crossing-over operator. Figure 1 shows the structure of the algorithm for clonal selection.

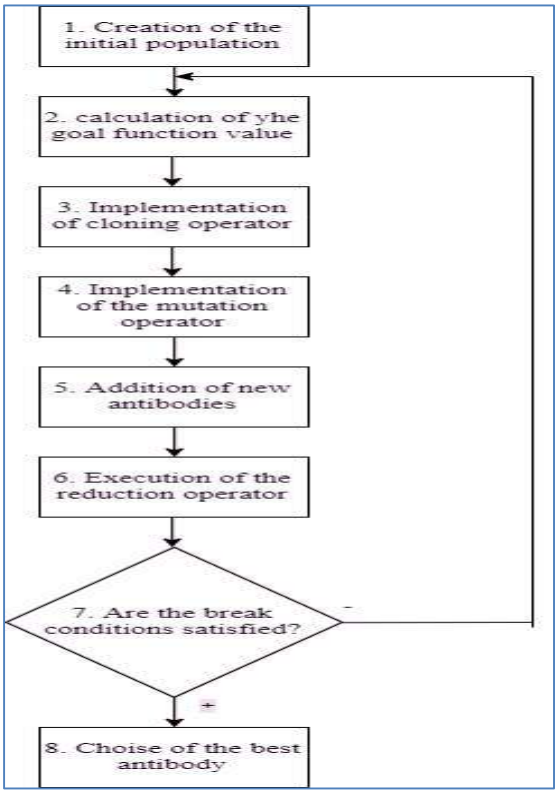

Figure 1. Structure of the algorithm for clonal selection

To solve the problem of finding a secret key, vertices are used as components, and solutions are used as an antibody.

There are three main ways to create a population:

Strategy of "blankets" (formation of the complete population). Practically cannot be implemented due to the big computational complexity;

- Strategy of "shotgun" (the formation of a sufficiently large subset of the total population). It is used the most;
- Strategy of "focusing" (formation of a population from varieties of one solution).

3. Methods and Material

Used 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 $f(x), x \in \lfloor x_{\min}, x_{\max} \rfloor$..... (1),

The goal function is represented in the form

$$F(x) = f(x) \to \min_{x} \quad .....(2)$$

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

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

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 makes 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 simulating annealing in the for

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

,

Where p_0 is the initial probability of reduction.

Probability selection based on a random selection scheme is determined by a simulated annealing as

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

For the simulation system, a combination of values of the input factors of the simulation model is determined, which allows to achieve a maximum / minimum of a certain response of the random variable. The response function is almost impossible to calculate analytically, however 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 have determined the active and efficient use of metaheuristic methods [5] as optimizers for problems of this type. Evolutionary metaheuristics (EM) are often used, namely: genetic algorithms and evolutionary strategies.

By genetic algorithm (GA) it is meant the heuristic search algorithm, which is used to solve optimization and modeling, problems by sequential selection, combination and variation of the required key parameters using mechanisms that resemble biological evolution.

When using encrypting tables, the key can be considered to be 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 the chromosome P is the element p_i.

In [6] deficiencies of this approach are noted, as genes are obtained dependent from each other, which leads to the possibility of obtaining incorrect solutions. The authors of [1, 5] propose 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 crypt analytical GA, in practice, an approach is used in which the key elements are considered as the genes of an individual. In order to avoid obtaining incorrect solutions for decimal chromosome coding, the rule is applied: when the same genes appear on the chromosome, the second repeating gene is replaced with the missing one. To determine the secret key as a function of fitness of individuals, the fact of coincidence of plaintext anciphertext is used. As an objective function, one can use the Jacobsen function [6,7,8] on the distribution of bigram frequencies in plain text.

An interesting development in the field of swarm intelligence is the bee algorithm, successfully 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 is considered in [9,10], where the implementation of the basic steps of the bee algorithm is proposed, and also a demonstration example of the cryptanalysis algorithm implementation is given.

An analysis of the results obtained in [11] allows us to state that with increasing the length of the key to the real one, applying only the genetic algorithm does not provide the expected result, regardless of the change in the error between the plaintext and the decrypted one.

The algorithm for calculating the secret key is proposed in this article. It consists of two stages:

First: the preparatory, in which the encoding and decoding of the text occurs.

Second: directly calculating the secret key from an open with the help of an attack based on the known plaintext and using a genetic algorithm.

The proposed algorithm consists of the following steps:

1. The initial population is formed randomly.
2. The crossing procedure gives m×n÷2 new keys. For parent rows, the point of division is randomly selected, the descendants are obtained by exchanging the cut-off parts.
3. To the obtained generation the mutation operator is applied. The bit of the individual of the population is inverted with a certain probability. Crossing and mutation is repeated several times.
4. 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.
5. 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) [12,13], in contrast to genetic algorithms, analyze the course of evolution at the phenotype level. In the 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 [14,15,16].

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 gradient method. The main drawback of 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":

$(x)= g(x)+ Z(x)$…….(6).

Where (x) is the global component of the model of the objective function, which is specified by the polynomial; 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 [17].But 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 [18]. Most often, three types of networks are used: a 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 a multilayer perceptron training is based on error correction (training with the teacher), with the most commonly used algorithm is reverse propagation (BP). This is an iterative gradient learning algorithm that provides minimization of the root-mean-square error.

Backward propagation algorithm

1. Number of iteration 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 $w_{ij}^{(k)}(n)$,

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

,

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

2. Set the training set

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

,

where $x_\mu - \mu$ -th training input vector, $d_\mu - \mu$ -th training output vector, $N^{(0)}$ – number of neurons in the input layer, $N^{(L)}$ – number of neurons in the output layer, P - the power of the learning set. The number of the current pair from the training set is

3. Calculation of the output signal for each layer (straight run)

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

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

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 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,

$f^{(k)}$ is the function of activation of neurons of the k-th layer. It is believed that

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

4. Calculation of the energy of the ANN error

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

5. Adjustment of the synaptic weights based on the generalized delta rule (reverse run)

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

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

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

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

6. Checking the termination condition

If n mod P 0, then 1, n n 1, transition to 3.

If n mod P 0 mo and $\frac{1}{P}\sum_{s=1}^{P} E(n-P+s) > \varepsilon$.....(16). = then n n 1, transition to 2.

If n mod P 0 and $\frac{1}{P}\sum_{s=1}^{P} E(n-P+s) < \varepsilon$..... ....(17),

then finished.

In the literature there is an opinion [6, 7] that when using metamodels in evolutionary algorithms, it is important part 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 a 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 thing is the work of the selection operator EM in selecting individuals in the next generation [19].

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, the prediction of 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 [20] is used as the criteria for assessing the quality of the meta-model, which depends on the difference in the ranks (numbers in the sorted by the FP list) of individuals that are calculated using the objective function.

If it would be 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 will be called basic. Each algorithm has its own model and as a result of their work we get one or several solutions.

Let the work of algorithms be performed asynchronously, and their interaction at the iteration h occurs by forming a metamodels taking into account individual models and the model generated by the previous iteration and generated by the decision algorithms.

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

When the specified exchange conditions are met, the current models of the basic algorithms (and, possibly, the corresponding solutions 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 itself can be represented as an optimization problem of finding the best element in 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 returns one or more of the best solutions found by the underlying algorithms. The results of computational experiments indicate that this methodology allows to increase the effectiveness of model-oriented algorithms, although it may require the development of more complex ways of aggregating models.

4. 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 way of co-operation) determines the balance between intensification and diversification of the search.

References

[1] Surakhi, O. M., Qatawneh, M., & OFEISHAT, H. (2017).A Parallel Genetic Algorithm for Maximum Flow Problem. *International Journal of Advanced Computer Science and Aplications*, *8*(6), 159-164.

[2] Low AM, Kelton V.D. Simulation modeling. 3rd ed. Trans. with English. St. Petersburg: Peter, Publishing. group BHV. 2004. 848 p.

[3] Glover F.W., Kochenberger G.A. Handbook of Metaheuristics. Kluwer, Boston. 2003. 570 p.

[4] Al Ofeishat, H. A., & Al-Rababah, A. A. (2009).Real-time programming platforms in the mainstream environments. *IJCSNS*, *9*(1), 197.

[5] Dubrov EO, Ryazanov AN, Sergeev AS, Chernyshev Yu.O. Development of methods for cryptanalysis of the ciphers of permutations and replacements in information security systems based on evolutionary optimization methods // Radioelectronic devices and systems for infocommunication technologies: a scientific conference dedicated to the day of radio. - Moscow, 2013. - p. 220-224.

[6] Haykin S. Neural Networks: A Comprehensive Foundation. Second Edition. Prentice-Hall. 1999. 874 p.

[7] Al-Ofeishat, H. A. (2012). Scheduling In Heterogeneous Grid-Systems. *Australian Journal of Basic and Applied Sciences*, *6*(10), 1-10.

[8] Gräning L., Jin Y., Sendhoff B. Individual-based Management of Meta-models for Evolu-tionary Optimization with Applications to Three-Dimensional Blade Optimization.// Evolu-tionary Computation in Dynamic and Uncertain Environments. Springer. 2007. P. 225-250.

[9] Branke J., Schmidt C. Sequential sampling in noisy environments. / In Xin Yao et. al., editor. // Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII). (Birmingham, UK, September 18-22, 2004). Berlin/Heidelberg, Germaany: Springer. 2004. P. 202–211. DOI: 10.1007/978-3-540-30217-9_21

[10] Jaskowski W., Kotlowski W. On Selecting the Best Individual in Noisy Environments. // The proceedings of GECCO'08.(July 12–16, 2008, Atlanta, Georgia, USA). P. 961-968.

[11] Afonin P.V. Construction of hybrid systems of simulation based on evolutionary metaevristics and neural networks // Survey of applied and industrial mathematics. Collection of works Macro Symposium "Fuzzy systems, soft computing and intelligent technology." 2011. T.4. C.38-39.

[12] Jin Y., Huesken M., Sendhoff B. Quality measures for approximate models in evolutionary computation./ In Alwyn M. Barry, editor.// GECCO 2003. Genetic and Evolutionary Computation Conference: Workshop Program. Chicago. 2003. P. 170–173.

[13] Afonin P.V. Evolution control method for evolutionary strategies based on neural network metamodels.Proceedings of the Vth International Scientific and Practical Conference "Integrated Models and Soft Computing in Artificial Intelligence".(May 28-30, 2009, Kolomna).M .: Fizmatlit. 2009. T.2. C.563-574.

[14] Al Smadi Takialddin, K. A. S., &Orayb, O. AL-Smadi, High-Speed for Data Transmission in GSM Networks Based on Cognitive Radio. *American Journal of Engineering and Applied Sciences*, *10*(1), 69-77.

[15] Chernyshev Yu.O.,Sergeev A.S., Dubrov E.O. Application of bioinspired optimization algorithms for cryptanalysis of classical and asymmetric cryptosystems // Informatics: problems, methodology, technologies: materials of the XIV International Scientific and Methodical Conference / VSU. - Voronezh: Publishing house VSU, 2014, with. 206-210.

[16] Hussein, A. L., Trad, E., & Al Smadi, T. (2018).Proactive algorithm dynamic mobile structure of Routing protocols of ad hoc networks. *IJCSNS*, *18*(10), 86.

[17] Jain, A., &Chaudhari, N. S. (2018). A novel cuckoo search strategy for automated cryptanalysis: a case study on the reduced complex knapsack cryptosystem. *International Journal of System Assurance Engineering and Management*, *9*(4), 942-961.

[18] Chaudhari, N. S. (2019). An Improved Genetic Algorithm and A New Discrete Cuckoo Algorithm for Solving the Classical Substitution Cipher.

[19] Lasry, G. (2018). *A methodology for the cryptanalysis of classical ciphers with search metaheuristics*.kassel university press GmbH.

[20] KamelAriffin, M., Abubakar, S., Yunos, F., &Asbullah, M. (2019). New Cryptanalytic Attack on RSA Modulus N= pq Using Small Prime Difference Method. *Cryptography*, *3*(1), 2.