IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 10, No. 2, June 2021, pp. 490~500
ISSN: 2252-8938, DOI: 10.11591/ijai.v10.i2.pp490-500

Cryptanalysis of Merkle-Hellman cipher using ant colony optimization

Hicham Grari[1], Siham Lamzabi[2], Ahmed Azouaoui[3], Khalid Zine-Dine[4]
[1,3]LAROSERI Laboratory, Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morocco
[2]Laboratory of Innovation in Management and Engineering for Entreprise (LIMIE), ISGA Rabat, Morocco
[4]Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco

Article Info

Article history:

Received Oct 21, 2020
Revised Feb 26, 2021
Accepted Mar 9, 2021

Keywords:

Ant colony optimization
Cryptanalysis
Merkle-hellman
Meta-heuristic
NP-hard

ABSTRACT

The Merkle-Hellman (MH) cryptosystem is one of the earliest public key cryptosystems, which is introduced by Ralph Merkle and Martin Hellman in 1978 based on an NP-hard problem, known as the subset-sum problem. Furthermore, ant colony optimization (ACO) is one of the most nature-inspired meta-heuristic optimization, which simulates the social behaviour of ant colonies. ACO has demonstrated excellent performance in solving a wide variety of complex problems. In this paper, we present a novel ant colony optimization (ACO) based attack for cryptanalysis of MH cipher algorithm, where two different search techniques are used. Moreover, experimental study is included, showing the effectiveness of the proposed attacking scheme. The results show that ACO based attack is more suitable than many other algorithms like genetic algorithm (GA) and particle swarm optimization (PSO).

This is an open access article under the CC BY-SA license.

Corresponding Author:

Hicham Grari
LAROSERI Laboratory
Faculty of Sciences, Chouaib Doukkali University
Route Ben Maachou, 24 000, El Jadida, Morocco
Email: grari.hicham@gmail.com

1. INTRODUCTION

Security and privacy protection of data is a great challenge in communication networks and computer systems. Cryptology is one of the most significant techniques for achieving information security covering two mutually unified subfields; cryptography and cryptanalysis. Cryptography is the study of building new powerful and efficient encryption and decryption algorithms using some mathematical problems as the theoretical basis. Cryptanalysis is the art of deciphering communications that are secured by cryptography; that is, finding, exploiting, and correcting weaknesses in cryptographic systems. The main challenge in cryptanalysis is to recover the plaintext or the key used for encryption.

It is become a common practice to use metaheuristics in cryptanalysis field. More recently, nature inspired metaheuristic algorithms have been used in cryptanalysis of many cipher. Especially, Ant colony optimization which is a promising approach that usually achieves considerably high performance in wide variety of problems.

In our previous work [1], we have proposed a new evolutionary way to attack Merkel-Hellman (MH) cryptosystem using ant colony optimization. We have modelled the cryptanalysis problem to a combinatorial problem in order to apply ACO metaheuristic, and the whole algorithm called MH-ACO was presented. In this paper, we intend to extend this algorithm by proposing two others approach which differ in the solution construction step, pheromone management rules and heuristic value. For the convenience of

description, we name them MH-BACO and MH-MACO. We will investigate their relative strengths and weaknesses by experimentation, concluding that MH-MACO approach is robust and efficient when compared to others attacks.

Many researchers have tried to attack Merkle-Hellman cipher using metaheuristics. Spillman [2] was the first to apply genetic algorithm for breaking the knapsack cipher. This attack is enhanced and re-implemented by Garg *et al.* [3], concluding that a high population size, high crossover probability and a low mutation probability increases the efficiency of GA attack. Abdul-Halim *et al.* [4] and Jain and Chaudhari [5] use a binary particle swarm optimization to attack Knapsack cryptosystem, they found that binary PSO is more efficient than the GA used by Spillman [2], and Garg *et al.* [3]. Furthermore, Sinha [6] proves that differential evolution is a much efficient technique than genetic algorithm for attacking the knapsack cipher. a cryptanalytic attack on the knapsack cryptosystem using binary variant of firefly algorithm is done by Palit [7] showing that the performance of firefly algorithm is much better than GA in this purpose.

Recently, Abdel-Basset [8] suggested a whole optimization-based attack using a sigmoid function for discretizing the search space, and the results show that the proposed algorithm is an efficient and robust for cryptanalysis of the Merkle-Hellman knapsack cryptosystem (MHKC) more than other algorithms. Kantour [9] proposed a parallel genetic algorithm for breaking the MHKC, their proposed scheme is enhanced with a deliberate cooperation among the search entities (GAs) via the migration operator. We mention also many other attacks in [10]-[20].

2. MERKLE-HELLMAN CRYPTOSYSTEM

In 1978 the famous Merkle and Hellman [21] public key cryptosystem was presented, which described an asymmetric cryptosystem based on a concrete case of the knapsack problem, which utilized a NP-complete subset sum problem (SSP) for its security. We can describe it is being as given a knapsack of volume S units and n items each of volume a_1, a_2 ... a_n units. We would like to find the number S by summing a subset of numbers from the set $A= \{a_1, a_2 \dots a_n\}$ so:

$$S = a_1x_1 + a_2x_2 + \dots + a_nx_n$$

An a_i item is fitted in the knapsack if the binary variable decision x_i is equal to *1*, the a_i item will not in the knapsack $x_i=0$.

The Merkle-Hellman cipher encrypts a message as a knapsack problem, the plain-text is divided into n-bit block. An example of Merkle-Hellman encryption is illustrated in Table 1, using 6 elements sequence: 1,4,7,11,19 and 27.

Table 1. Example of Merkle-Hellman encryption

Plaintext	Knapsack sequence	Ciphertext
1 0 1 0 1 1	1,4,7,11,19,27	1+7+19+27= 54
1 1 0 1 1 0	1,4,7,11,19,27	1+4+11+19 = 35
0 0 1 0 0 1	1,4,7,11,19,27	11+27 = 38

However, if the A set is a super-increasing sequence, meaning that: Each a_i element of the sequence fulfils the condition: $a_i > \sum_{j=1}^{i-1} a_j$ And $i \in \{2\dots n\}$.

In this concrete case, the knapsack is called an easy knapsack that can be solved is being as:

$$x_n = \begin{cases} 1 \; if \; S \geq a_n \\ 0 \; if \; S < a_n \end{cases} \text{ And For each j } \in [0,1,\dots,n-1] : x_j = \begin{cases} 1 \; if \; S - \sum_{k=j+1}^{n} x_k \, a_k \geq a_j \\ 0 \; if \; S - \sum_{k=j+1}^{n} x_k \, a_k < a_j \end{cases}$$

Using this feature, Merkle and Hellman [10] developed they public key cryptosystem, the private/public key are a sequence of number for a super-increasing/normal knapsack problem with the same solution. Merkle and Hellman suggested that such an easy knapsack be converted into a more complex trapdoor knapsack. This transformation involves the following steps:

1. Select a simple knapsack super-increasing sequence elements A'= (a'_1, a'_2,… a'_n)
2. Select an integer value m greater than sum of all elements of super-increasing sequence m >2 a'n.
3. Select another integer w that the gcd (m, w) =1, that is number m and w are reciprocally prime.
4. Find w^{-1} the inverse of the w mod m.
5. Construct the hard knapsack sequence.

A=w x A' (mod m) i.e. a_i=w x a'_i (mod m) for each i in {1… n.}. The trapdoor sequence A could be published as a public key (encryption key). The private (secret) key for this cipher consists of a simple knapsack sequence A' and the values m, w, w^{-1}.

3. ANT COLONY OPTIMIZATION

The research concept that has been underlined in this paper is applying of meta-heuristic approach ant colony optimization to break MH cryptosystem. Ant colony optimization [22] represents a class of population-based metaheuristics inspired by the behavior of real ant colonies. These ants are capable to find shortest paths between food sources and their nest, using an indirect communication mediated by the pheromone trail.

In a real ant colony, ants explore their environment for the search of food sources; each ant deposes a chemical substance called pheromone on his path. Thus, the others ants can smell this pheromone and they tend to choose, probabilistically, paths with strong pheromone concentrations. Old paths are less likely to be used because of the pheromone evaporation mechanism, allowing forgetting suboptimal path. This simple idea is implemented by the ACO methods to resolve and address hard combinatorial problems such as traveling salesman problems (TSP), quadratic assignment problems, vehicle routing problems, or constraint satisfaction problems.

The first ACO algorithm, called ant system (AS) introduced by Dorigo *et al.* [23] was applied to travelling salesmen problem (TSP). Other ACO variants where introduced which differ in the solution construction procedure and pheromone trails update, including ant colony system (ACS) presented by Dorigo and Gambardella [24], and min max ant system (MMAS) given by Stutzle and Hoos [25]. In ant colony system (ACS), the algorithm improves over ant system (AS) by an excessive exploitation of the search experience accumulated by the ants, using a more aggressive action choice rule explained is being as:

At each construction step, an ant chooses a random variable q uniformly distributed in [0, 1]. If q is less than a fixed parameter q_0 such as $0 \leq q_0 \leq 1$, the ant makes the best possible choice as indicated by the pheromone trails and the heuristic information, exploiting the past experiences. While with probability 1- q_0 it performs a biased exploration like an ordinary AS. Tuning the parameter q_0 allows modulation of these two choices (exploiting the current results or exploring new solutions).

4. PROPOSED APPROACH

As mentioned earlier, the problem can be stated is being as: Given an n-elements public key A={a_1, a_2…a_n} and S the target sum representing the ciphertext, find a particular subset of number from the set A such as : $S = \sum_{i=1}^{n} a_i x_i$ whith $x_i \in \{0,1\}$ and i = 1,…,n.

where x_i is a binary variable indicating whether or not item i was selected.

The problem so far is equivalent to a subset sum problem, which is well-known NP-complete. To overcome this problem, we propose an ant colony based algorithm, in order to tackle the problem as a combinatorial problem. The main procedure of our proposed method is described is being as: All ants constructs a solution in each particular round, and then pheromone trails are updated. The algorithm stops iterating when the maximum number of rounds is attained or a so good solution is achieved. Furthermore, a constructive approach is used to build solutions, each ant start building a feasible solution by iteratively adding appropriate components from all the allowed ones in a probabilistic manner until a complete solution is obtained. In the following, we will put forward three search space design based on problem feature, called MH-ACO proposed in [1], MH-BACO and MH-MACO, in each method we will describe the details about the solution construction procedure, then we will define an appropriate heuristic information and strategy to update pheromone trails. Next, we will present the used fitness function. Finally, the outline of the whole algorithm is presented.

4.1. MH-ACO procedure

4.1.1. Solution construction

At each round, each ant chooses an initial object randomly and then iteratively adds object from a set of candidates objects N_i that can be selected without violating resource constraints. Once N_i is empty, a solution is constructed. The search space is illustrated in Figure 1.

ACO algorithms are stochastic algorithms that make probabilistic decision in terms of the artificial pheromone trails and the local heuristic information. These two factors are combined to form the so-called probabilistic transition rule defined is being as:

$$P(i,j) = \begin{cases} \frac{\tau(i,j)^{\alpha}\ \rho(j)^{\beta}}{\sum_{k \epsilon N_i} \tau(i,k)^{\alpha} \rho(k)^{\beta}}\ if\ j \epsilon\ N_i \\ 0\ ,0\ therwise \end{cases} \quad (1)$$

In ACS, the transition rule is defined is being as

$$T(i,j) = \begin{cases} max_{l \epsilon N_i}(\tau(i,l)^{\alpha}\ \rho(l)^{\beta})\ , if\ q\ \leq\ q_0 \\ P(i,j)\ otherwise \end{cases}$$

where $P(i,j)$ is the probability to select the next object a_j within the set of available objects N_i in the the *i-th* construction step. And q is a random number in range [0,1], q_0 is a an exploitation parameter ($0 \leq q_0 \leq 1$).

The probability P(i,j) is based on two factors. First, the amount of pheromone trail τ(i,j) in the edge (i,j). Second the heuristic value ρ(j) representing the attractiveness of the object a_j. The parameters α and β are used respectively to control the pheromone effect and the heuristic value.

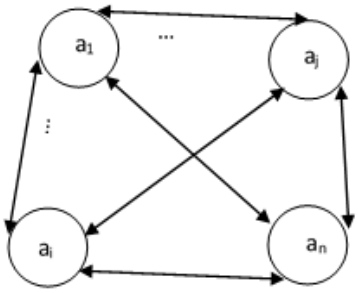

Figure 1. Search space for MH-ACO

4.1.2. Pheromone update

Once each ant has built a solution in a particular round. The best solution found S_{best} is retained, which will be used to update the pheromone trails. The pheromones over the edges constituting the tour of the best ant is updated using (2), so larger the fitness value, the greater is the amount pheromone deposited.

$$\tau_{i,j} = \sigma \times \tau_{i,j}\ + \Delta\tau_{i,j}(S_{best}) \quad (2)$$

$$and:\ \Delta\tau_{i,j}(S_{best}) = \begin{cases} Q \times F(S_{best})\ if\ arc(i,j)\ belongs\ to\ S_{best} \\ 0\ otherwise \end{cases}$$

where $\sigma\ \epsilon$ [0-1] is the evaporation rate. However, pheromone evaporation is a natural phenomenon that ensures that old pheromone should not have too strong influence on the future.

Q is some constant, and *F* is the fitness function used to evaluate each constructed solution (defined in the section 4.4). To update pheromone trail based on the best solution S_{best} , we have to lay pheromone on all pairs (a_i, a_j) of each different objects of S_{best} . So, the strategy is to increase the desirability of choosing together two objects of $S_{best.}$

4.1.3. Heuristic value

The possibility of using heuristic information ρ is important because it improve exploitation of the search space. In our ACO algorithm, we have used a dynamic heuristic information that depends on the partial solution constructed and therefore has to be computed at each time when the ant need to make a choice. The transition probability in (1) needs a heuristic value calculation method from the problem domain as an efficient search methodology. In our approach, the heuristic value is defined is being as:

Let S_i be the set of the selected objects at the *i-th* construction step, the heuristic value ρ(j) for a candidate object *j* is given is being as (3):

$$\rho(j) = \frac{a_j}{Sc} \text{ where } S_c = S - \sum_{k \in S_i} a_k \quad (3)$$

S_c is called the current knapsack capacity; S_c is calculated at each step by subtracting all the selected object in the partial solution S_i from the target sum S.

4.2. MH-BACO procedure

4.2.1. Solution construction

In this variant (named binary ant colony optimization), we have modelled the search space to a *n+1* nodes graph. Every node a_i is linked to the next node a_{i+1} by two different edges, the first edge is equal to *0* and the second edge is equal to *1* as explained in Figure 2.

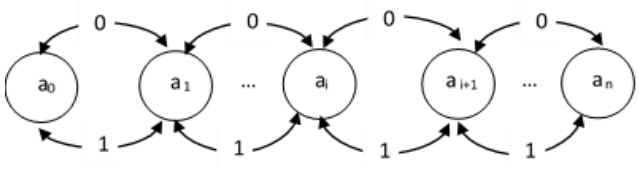

Figure 2. Search space for MH-BACO

In a particular tour, each ant constructs a feasible solution considered as a candidate key, it will consist of a path from the first node a_0 to the last node a_n, through the edge *0* or *1* between each 2 nodes a_i and a_{i+1}. The probabilistic transition rule in this case is defined is being as (4):

$$P(i,j) = \frac{\tau(i,j)^{\alpha}\,\rho(j)^{\beta}}{\tau(i,0)^{\alpha}\rho(0)^{\beta}+\tau(i,1)^{\alpha}\rho(1)^{\beta}} \; whith\; j\epsilon\{0,1\} \quad (4)$$

In this approach, ants have a limited view towards objects. In fact, in each iteration ants can only move to the next object in the search space as shown in the Figure 2. However, they can decide to select or not this object.

4.2.2. Pheromone update

The pheromone strategy used in this method is the same used in MH-ACO. More formally, the pheromone values are updated is being as (5):

$$\tau_{i,j} = \sigma \times \tau_{i,j} + \Delta\tau_{i,j}(S_{best}) \quad (5)$$

$$and:\ \Delta\tau_{i,j}(S_{best}) = \begin{cases} Q \times F(S_{best}) \; if\; edge(i,j)\; belongs\; to\; S_{best} \\ 0\; otherwise \end{cases}$$

where $\Delta\tau_{i,j}$ is the amount of pheromone deposited on the edges between each two objects contained in the best solution S_{best}, the value of $\Delta\tau_{i,j}$is proportional to the best solution quality.

4.2.3. Heuristic value

The heuristic value used in MH-BACO is defined is being as: Let S_i be the set of the selected objects at the i-th construction step, the heuristic value $\rho(0)$ (which means that the object is not selected) and $\rho(1)$ (which means that the object is not selected) is given is being as (6):

$$\rho(0) = \frac{a_j}{S_c},\ \rho(1) = \frac{S_c}{a_j} \quad (6)$$

With $S_c = S - \sum_{k\in S_i} a_k$ called current knapsack capacity.

4.3. MH-MACO procedure

4.3.1. Solution construction

Named modified MH-MACO, it combine the advantages of the two previous strategies. First, by giving to ants a wide view of objects. Thus, each ant can make a choice to move to any object (according to 7). Second, ants decide whether to select (it will be fitted in the partial solution under construction) or not the visited object. The search space is defined as a fully connected graph like MH-ACO, where nodes

represent objects of the Knapsack. Similar to MH-BACO algorithm, there are two sub-edges between each 2 objects in the graph, one for selecting and the other for deselecting the corresponding object, as illustrated in Figure 3.

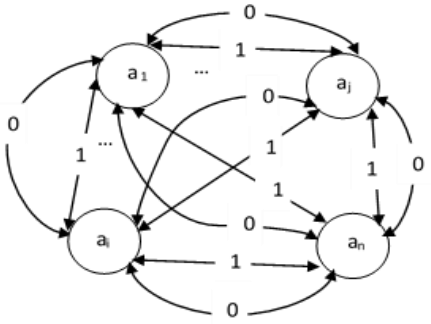

Figure 3. Search space for MH-MACO

In each round, each ant can decide whether to select an object or not. If an ant chooses sub-edge 1 (or 0) of object O_i, it means the object is selected (or deselected) by that ant. Note that, ants are only allowed to select one of the sub-edge (0 or 1) of each node. Based on the (7), an ant choose its next path and the process continues until its visits all objects. At the end of each cycle. Each ant built a solution in the form of n-bit vector, where 1 means selecting and 0 means deselecting the corresponding object.

The probabilistic function of transition, denoting the probability of an ant at node a_i to choose the path p (0 or 1) to reach the node a_j , is defined is being as (7):

$$P(i,j,p) = \begin{cases} \frac{\tau(i,j,p)^{\alpha}\,\rho(j,p)^{\beta}}{\sum_{k\epsilon N_i}\sum_{l\epsilon[0.1]}\tau(i,k,j)^{\alpha}\,\rho(k,l)^{\beta}} & \text{if } j\epsilon\, N_i \\ 0 \text{ Otherwise} \end{cases} \quad (7)$$

where N_i is the set of objects not yet visited. Two parameters are used to calculate the probability of moving from a state i to another state j; first, the amount of pheromone trail $\tau(i,j,p)$ which reflects the potential tend for ants to select (p=1) or deselect (p=0) the object a_j. And second, the heuristic value $\rho(j,p)$ representing the attractiveness of the (de)selection the object a_j.

4.3.2. Pheromone update

The pheromone update is intended to make solution components belonging to good solutions more desirable for the following iterations. It consists to increases the level of the pheromone of solution components that are associated with the best solution obtained in each cycle. More explicitly, the pheromone values are updated is being as (8):

$$\tau_{i,j,p} = \sigma \times \tau_{i,j,p} + \Delta\tau_{i,j,p}\,(S_{best}) \quad (8)$$

where:

$$\Delta\tau_{i,j,p}(S_{best}) = \begin{cases} Q \times F(S_{best})\ if\ arc(i,j,p)\ belongs\ to\ S_{best} \\ 0\ otherwise \end{cases}$$

$\Delta\tau_{i,j}$ is the amount of pheromone deposited on the edges between each two objects contained in the best solution S_{best}, the value of $\Delta\tau_{i,j,p}$ is proportional to the best solution quality.

4.3.3. Heuristic value

Like *MH-BACO* procedure, the heuristic value used in this model is defined as:

$$\rho(j,0) = \frac{a_j}{S_c}\ , \rho(j,1) = \frac{S_c}{a_j} \quad (9)$$

with $S_c = S - \sum_{k \in S_i} a_k$

4.4. Fitness function

The fitness function a key component in the success a research algorithm, allowing a relevant qualification of the candidats solutions. An efficient fitness function helps the search algorithm in exploring the search space more efficiently towards promising solutions.

Let $M=\{m_1,m_2,\ldots,m_n\}$ and $m_i \epsilon \{0,1\}$ be an arbitrary solution, the fitness function used in this paper is that proposed by Spillman [2] is being as (10):

$$F = \begin{cases} 1 - \left(\frac{|\text{Target}-\text{Sum}|}{\text{Target}}\right)^{\frac{1}{2}} & \text{if Sum} \leq \text{Target} \\ 1 - \left(\frac{|\text{Target}-\text{Sum}|}{\text{MaxDiff}}\right)^{\frac{1}{6}} & \text{if Sum} > \text{Target} \end{cases} \quad (10)$$

With Sum $=\sum_{j=1}^{n} a_j m_j$, Full Sum $=\sum_{j=1}^{n} a_j$ and Max Diff = max {Target, Full Sum-Target}. The Target is the Ciphertext value. A fitness value of 1 means that the correct solution has been found.

4.5. Proposed algorithm

Figure 4 shows the Merkle-Hellman cryptosystem and its cryptanalysis by our based-ACO algorithm. The input to the algorithm is the ciphertext and the public key. The task of MH-ACO algorithm is to recover the plaintext.

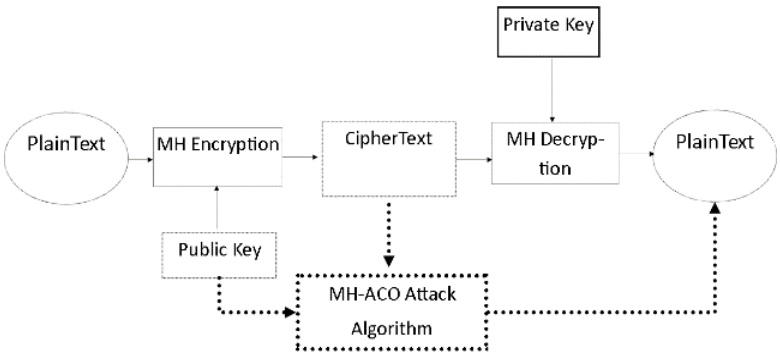

Figure 4. Layout of MH cipher and our attack algorithm

We will explain the main steps of our proposed algorithm in the following subsections:
Outline of ACO Algorithm:

Input: Public Key and Cipher text.
Perform initialization of all parameters of ACO
repeat
for each ants N do
Construct candidate key using the specific probabilistic transition rule
Evaluate the generated candidate key using the FF according to (3)
End for
Update best key information and pheromone values following the specific rules.
until: The algorithm reach the termination condition
Output: The best Key.

5. EXPERIMENTAL RESULTS AND DISCUSSION

In this paper, a series of experiments have been conducted to evaluate the effectiveness of the proposed methods. We have implemented our algorithm with *C++* language. All experiments were performed on an Intel Core i7-4712MQ CPU processor (2.30 GHz and 4 GB RAM). The purpose of this section is threefold: (1) to investigate the performance of different schemes designed for MH cryptanalysis,

(2) to study the settings of related parameters, and (3) to evaluate the performance of MH-MACO by comparing it with some existing algorithms.

5.1. Performances study

In order to evaluate the performance of the three proposed strategies (MH-ACO, MH-BACO and MH-MACO), we apply the same parameters value is being as: We have set α to 1.2, β to 1 and σ to 0.95 (where α, β and σ are weight of pheromone, heuristic value and evaporation rate respectively). The number of ants N where set to 15 and q_0 to 0.8. To make an overall comparison, the 'MACRO' word is used. Table 2 shows the result of encoding the 'MACRO' word using Merkle-Hellman algorithm with 15 elements sequence is being as:

- Private Key: (1, 3, 7, 13, 26, 65, 119, 267, 504, 1007, 2013, 4027, 8053, 16107, 32213).
- Public Key: (21031, 63093, 16371, 11711, 23422, 58555, 16615,54322, 1098, 46588, 6722, 34475, 47919, 51446, 16438).
- m=65423, w=21031, w^{-1}=5363.

Table 2. MH encryption of the word 'MACRO'

Character	ASCII Code	Target sum (ciphertext)
M	10110010	65728
A	10000010	37646
C	11000010	100739
R	01001010	103130
O	11110010	128821

Table 3 summarizes the results of 100 runs for the cryptanalysis of the message 'MACRO' using MH-ACO, MH-BACO and MH-MACO algorithms, the number of key searched before locating the reel one is recorded in the Table. It can be seen from the results that MH-MACO performs better than MH-ACO, MH-BACO, it can break the message after reaching only 437 keys, while MH-ACO, MH-BACO requires 563 and 645 key respectively. In Table 4, we illustrates the average search space (ASS) and the success rate (SR) for each model, a maximal success rate (SR) is obtained for the 3 algorithms.

Table 3. Number of keys searched to break MH cryptosystem

	Number of Key Searched		
Character	MH-ACO	MH-BACO	MH-MACO
M	555	640	435
A	580	613	385
C	620	750	566
R	510	575	322
O	550	692	478
Average	563	654	437

Table 4. Comparison of cryptanalytic results of different approaches

Character	MH-ACO		MH-BACO		MH-MACO	
	ASS	SR (%)	ASS	SR (%)	ASS	SR (%)
M	1.5%	100	1.9	100	1.3	100
A	1.8	100	1.8	100	1.1	100
C	3.1	100	2.2	100	1.7	100
R	1.3	100	1.7	100	0.9	100
O	1.6	100	2.1	100	1.4	100
Average	1.8	100	2.0	100	1.3	100

The second part of the performances analysis is to compare our models with different value of *n* (number of element in the knapsack). Comparison results are illustrated in Table 5. As we can observe, MH-MACO algorithm achieves a higher success rate when the capacity of the knapsack is increasing, reaching 85% and 57% for *n* equal to 25 and 40 respectively. And exploring a lower average search space. The most characteristics of *MH-MACO* search technique are explained in:

The first one is that *MH-MACO* allows ants to explore all objects, while in *MH-ACO* algorithm, exploration is no longer posible when a stopping criterion is met. Therefore, ants do not have the ability to observe all objects to make a choice. In *MH-MACO* algorithm ants have the possibility to select or not visited

elements. Unlike *MH-ACO* algorithm, where every visited object by the ant is immediately selected. Finally, the main advantage of MH_MACO over MH_BACO is that each ant has a broad view to objects to be visited, at the same time with the opportunity of selection or not. However, in MH_BACO ants moves with a limited view, ants could only choose whether or not to select the next object.

Table 5. Comparison results whith diffent values of *n*.

	MH-ACO		MH-BACO		MH-MACO	
n	ASS	SR (%)	ASS	SR (%)	ASS	SR (%)
10	35	100	42	100	19	100
15	1.8	100	2.0	100	1.3	100
25	1.2	78	2.7	52	1.8	85
40	0.006	36	0.005	41	0.003	57

5.2. Parametric sensitivity analysis

The following experiments consist of studying the impact of q_0 parameter (which determines the relative importance of exploitation versus exploration) in MH-MACO model. We ran our algorithm for the word 'MACRO' with different values of q_0. For the others parameters , we have set α to 1.2, β to 1 and σ to 0.95. The number of ants N where set 15.

In Figure 5 we report the Fitness value evolution through the number of cycle, as we can observe, when q_0 is equal to 1, an early stagnation of the search, the system has ceased to explore new possibilities and no better solution is likely to be found anymore. This undesirable behavior is due to an excessive exploitation. Whereas, with q_0 equal to *0*, the transition rule is reduced to a pure ant system without any exploitation, in this case the convergence into the best solution is more delayed (reached after 50 cycle). The best performance results are obtained when alternating both exploration and exploitation (q_0=0.8), the correct key is located after only 29 cycles.

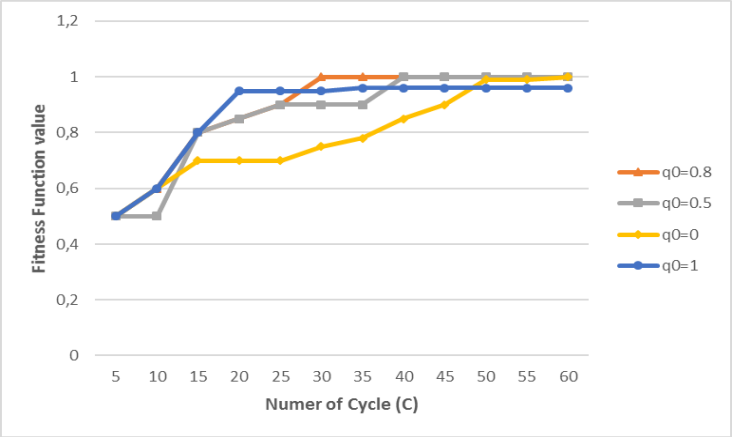

Figure 5. Fitness value evolution for different values of q0

5.3. Comparison with others attacks

The performance of the MH-MACO Algorithm is measured and compared with the GA that is used by Spillman [2] and PSO used by Jain and Chaudhari [5]. The number of ants used for this purpose is N=15 allowing the best search space rate. For the word 'MACRO' the comparison results are illustrated in Table 6.

Table 6. Comparison of cryptanalytic results obtained BPSO, GA and MH-MACO

Character	PSO		IGA		MH-MACO	
	ASS	SR (%)	ASS	SR (%)	ASS	SR (%)
M	20.94	100	2	100	1.3	100
A	19.62	100	0.2	100	1.1	100
C	50.45	100	6	100	1.7	100
R	37.13	100	1.0	100	0.9	100
O	57.39	100	0.1	100	1.4	100
Average	53.10	100	1.9	100	1.3	100

As we can see the average search space (ASS) obtained with MH-MACO algorithm is mush better than PSO for all letters 'MACRO'. Comparing with GA results, as we can see the performances of our algorithm is so close to GA results with an ASS equal to 1.3 for MH-MACO and 1.9 for GA. Also, the percentage of success rate (SR) obtained is maximal.

6. CONCLUSION

In this paper, we introduced a novel search procedure based on the ant colony optimization metaheuristic, for cryptanalysis of Merkle-Hellmen public key cryptosystem. We have defined the essential components of our algorithm such as solution construction, heuristic value, fitness function and the strategy to update pheromone trails. In order to evaluate the effectiveness of our algorithm, a set of experiments were performed that confirmed the robustness of our model. The best results are those obtained with the MH-MACO procedure, which is characterised by a broad view to objects to be visited, with the opportunity of selection or deselection of objects. The fitness function used in this paper is that proposed by Spillman, which still needs improvement in future work to perform cryptanalysis more efficiently, especially the possibility of using specifics elements extracted from the problem being treated. We hope also to hybridize our algorithm with a local search technique, in order to improve the convergence speed and success probability of our algorithm; especially for high dimension cryptanalysis problem.

REFERENCES

[1] H. Grari, A. Azouaqui, K. Zine-Dine, M. Bakhouya, and J. Gaber, "Cryptanalysis of Knapsack Cipher Using Ant Colony Optimization," *Smart Application and Data Analysis for Smart Cities (SADASC'18)*, 2018, doi: 10.2139/ssrn.3185322.

[2] R. Spillman, "Cryptanalysis of Knapsack Ciphers Using Genetic Algorithms," *Cryptologies*, vol. 17, no. 4, pp. 367-377, 1993, doi: 10.1080/0161-119391867999.

[3] P. Garg, A. Shastri, and D. C. Agarwal, "An enhanced cryptanalytic attack on Knapsack Cipher using Genetic Algorithm," *Transaction on Engineering. Computing and Technology*, vol. 1, no. 12, pp. 4071-4074, 2007, doi: 10.5281/zenodo.1330067.

[4] M. F. AbdulHalim, B. A. Attea and S. M. Hameed, "A binary Particle Swarm Optimization for attacking knapsacks Cipher Algorithm," *2008 International Conference on Computer and Communication Engineering*, Kuala Lumpur, Malaysia, 2008, pp. 77-81, doi: 10.1109/ICCCE.2008.4580572.

[5] A. Jain, N. S. Chaudhari, "Cryptanalytic Results on Knapsack Cryptosystem Using Binary Particle Swarm Optimization," In *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14. Advances in Intelligent Systems and Computing*, Springer, Cham, vol. 299, 2014, doi: 10.1007/978-3-319-07995-0_37.

[6] S. N. Sinha, S. Palit, M. A. Molla, A. Khanra, and M. Kule, "A cryptanalytic attack on Knapsack cipher using Differential Evolution algorithm," *2011 IEEE Recent Advances in Intelligent Computational Systems*, Trivandrum, India, 2011, pp. 317-320, doi: 10.1109/RAICS.2011.6069326.

[7] S. Palit, S. N. Sinha, M. A. Molla, A. Khanra, and M. Kule, "A cryptanalytic attack on the knapsack cryptosystem using binary Firefly algorithm," *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)*, Allahabad, India, 2011, pp. 428-432, doi: 10.1109/ICCCT.2011.6075143.

[8] M. Abdel-Basset, D. El-Shahat, I. El-henawy, A. K. Sangaiah, and S. H. Ahmed, "A Novel Whale Optimization Algorithm for Cryptanalysis in Merkle-Hellman Cryptosystem," *Mobile Networks and Applications*, vol. 23, pp. 723-733, 2018, doi: 10.1007/s11036-018-1005-3.

[9] N. Kantour and S. Bouroubi, "Cryptanalysis of Merkle-Hellman Cipher Using Parallel Genetic Algorithm," *Mobile Networks and Applications*, vol. 25, pp. 211-222, 2020, doi: 10.1007/s11036-019-01216-82019.

[10] T. Mandal and M. Kule, "An improved cryptanalysis technique based on Tabu search for Knapsack cryptosystem," *Int J Control Theory Appl*, vol. 16, no. 9, pp. 8295–8302, 2016.

[11] H. Singh, "Contravening esotery: cryptanalysis of knapsack cipher using genetic algorithms," *arXiv preprint arXiv:1606.06047*, vol. 140, no. 6, 2016.

[12] Z. Kochladze and L. Beselia, "Cracking of the Merkle–Hellman cryptosystem using genetic algorithm," *Int J Sci Technol*, vol. 3, no. 1–2, pp. 291–296, 2016.

[13] J. C. Bansal and K. Deep, "A modified binary particle swarm optimization for knapsack problems," *Appl Math Comput*, vol. 218, no. 22, pp. 11042–11061, 2012, doi: 10.1016/j.amc.2012.05.001.

[14] L. Beselia, "Using genetic algorithm for cryptanalysis cryptoalgorithm Merkle-Hellman," *Comput Sci Telecommun*, vol. 48, no. 2, pp. 49–53, 2016.

[15] G. Lokeshwari, S. Susarla and S. U. Kumar, "A modified technique for reliable image encryption method using Merkle-Hellman cryptosystem and RSA algorithm," *J Discret Math Sci Cryptogr*, vol. 18, no. 3, pp. 293–300, 2015, doi: 10.1080/09720529.2014.968367.

[16] G. Lokeshwari, S. U. Kumar and S V. Susarla, "An efficient image encryption using Merkle-Hellman, elgamal and genetic algorithm for color images" *Appl Mech Mater*, vol. 719-720, pp. 1140-1147, 2015, doi: 10.4028/www.scientific.net/AMM.719-720.1140.

[17] A. Shamir, "A polynomial time algorithm for breaking the basic Merkle Hellman cryptosystem," *IEEE Trans Inf Theory IT*, vol. 30, no. 5, pp. 699–704, 1984, doi: 10.1109/TIT.1984.1056964.

[18] L. M. Adleman "On beaking generalized knapsack public key cryptosystems," *In: ACM Proceedings of 15th STOC*, 1983.
[19] S. Jain and N. S. Chaudhari, "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*, vol. 9, pp. 942–961, 2017, doi: 10.1007/s13198-017-0690-9.
[20] H. Grari, A. Azouaqui, K. Zine-Dine. "A cryptanalytic attack of simplified-AES using ant colony optimization," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4287-4295, 2019, doi: 10.11591/ijece.v9i5.pp4287-4295.
[21] R. Merkle and M. Hellman, "Hiding information and signatures in trapdoor knapsacks," in *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 525-530, Sep. 1978, doi: 10.1109/TIT.1978.1055927.
[22] M. Dorigo, "Optimization, Learning and Natural Algorithms," PhD thesis, Politecnico di Milano, 1992.
[23] M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimization by a colony of cooperating agents," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29-41, Feb. 1996, doi: 10.1109/3477.484436.
[24] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," in *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, Apr. 1997, doi: 10.1109/4235.585892.
[25] T. Stutzle and H. Hoos, "Improvements on the ant system: Introducing the MAX-MIN ant system," in A*rtificial Neural Nets and Genetic Algorithms*, Springer, Vienna, Wien. Germany, 1998, pp. 245-249, doi: 10.1007/978-3-7091-6492-1_54.

BIOGRAPHIES OF AUTHORS

Hicham Grari is a PhD student within the LAROSERI Laboratory at Faculty of Sciences-Chouaib Doukkali University, El Jadida/Morocco. He holds an Engineer Degree in Computer Science in 2005. His doctoral research investigates the use of metaheuristics in cryptology field.

Siham Lamzabi received her PhD in Computer Science on 2016 from Faculty of Sciences, Mohammed V University in Rabat Morocco. Since 2017, she is a professor of Computer Science in the Institute of Engineering and Business (ISGA), Rabat Morocco. Her research interests focus on Routing protocols, Internet network, scale free network and security.

Ahmed Azouaoui received his license in Computer Science and Engineering in June-2001 and Master in Computer Science and Telecommunication from University of Mohammed V - Agdal, Rabat, Morocco in 2003. He received his PHD in Computer Science in 2013 and Engineering at Department of Computer Science, ENSIAS (National School of Computer Science and Systems Analysis), Rabat, Morocco. Currently, he is an Associate Professor at Department of Computer Science, Faculty of sciences, University Chouaib Doukkaly, El Jadida, Morocco. His areas of interest are Information systems, Coding Theory, Security, Pattern Recognition and Artificial Intelligence.

Khalid Zine-Dine received his PhD degree from the Mohammed V University in Rabat, Morocco, in 2000. He spent four years in bank Information System as a network & system security project manager. Currently, he is a Full Professor at Faculty of Sciences - Mohammed V University in Rabat - Morocco. His research interests are in the area of wireless ad hoc and sensor networks, Mobility, cloud computing, Security, Smart Grid and system & network architectures and protocols. Dr. Zine-Dine was a co-organizer and co-chair of conferences and is supervising PhD Thesis and involved in funded research projects.