Advances in Computer Science Research, volume 89
2019 International Conference on Wireless Communication, Network and Multimedia Engineering (WCNME 2019)

A Hierarchically Collaborative Ant Colony Based Assembly Algorithm for Security Protocol

Zhuo Yi[1,2,*], Muming Sun[2], Lifeng Cao[1] and Xuehui Du[1]
[1]State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou Institute of Information Science and Technology, Zhengzhou, China
[2]Troop 78123 PLA, Chengdu, China
[*]Corresponding author

***Abstract*—Assembly of reconfigurable security protocol remains a major challenge for deploying higher security-level but more complicated security strategies in access points with limited resources and computing abilities. To handle this problem commendably, a hierarchically collaborative ant colony-based assembly algorithm was proposed. This algorithm decomposed the security protocol assembly problem into assembling directions controlling sub-task and cryptographic components selection sub-task. Directions control generated assembly sub-goals and cryptographic components selection schedules the best-fitted components for given sub-goals. Both sub-tasks adopted a modified version of ant colony algorithm to fulfil its targets. These two ant colony algorithms generate a candidate optimal solution collaboratively for the assembly problem. And a hierarchical pheromone was defined to reinforce positive behaviors of ant colony. Additionally, a Lévy theory based stochastic gradient algorithm was adopted to verify and re-optimize the optimal solution. Experiment results suggest that the proposed algorithm outperforms baseline algorithms in convergence and performance.**

Keywords—security protocol assembly; collaborative ant colony; reconfigurable security protocol; space-ground integrated network; decision-making

I. INTRODUCTION

Reconfigurable secure protocol (RSP), with adaptively dynamic protocol assembly and flexible resource configuration [1-2], could greatly improves resources utilization and enhances network security, especially in access points of space-ground integrated network (SGIN). Since performance of all components determine the efficiency of target protocol, the key issues in RSP is to determine the optimal secure access resources and corresponding protocol flow to generate the target protocol, which is defined as assembly decision-making problem (ADMP) in this paper. Generally, the scales of cryptogram resources and the diversities in design standards, cryptosystem, application situations of cryptogram resources enlarge the solution space of RDMP. What's worse, the uncertainty of protocol flow caused by reconfiguration granularity, further intensify this situation. However, previous works mainly focus on resources scheduling and ignores the role of assembly flow in ADMP. Thus, designing an accurate and efficient assembly algorithm is of great significance.

To address this problem, a hierarchically collaborative ant colony-based protocol assembly algorithm is proposed. The algorithm inspired by hierarchical reinforcement learning [3-4] and collaborative behavior of biological populations [5], decomposes the ADMP into two sub-problems including sub-goals generation of resource selection and resources selection for each sub-goal. The sub-goals generation sub-task explores abstract resources scheduling sub-goals at lower temporal resolution in a latent state-space, while cryptographic resources selection schedules proper cryptogram resources for the sub-goals to generates the most optimal solution for the target protocol. Both sub-tasks operate with an improved ant colony algorithm and a hierarchical pheromone is defined to reinforce the positive behaviors of populations. Additionally, a Lévy theory [6] based stochastic gradient algorithm is adopted to verify and re-optimize the solution.

II. FORMALIZATION OF THE PROBLEM

Definition 1 Assembly Model for RSP is defined as a quadruples $RSP = \langle AG, CR, CE, AS \rangle$, where AG denotes assembly goals, depicting functional and performance requirements of the target security protocol. CR refers to as all available cryptogram components including cryptocards, FPGAs, reconfigurable processors, etc. CR represents efficiency criteria for protocol performance evaluation. AS indicates assembly solutions of given protocol, which consists of protocol flow and corresponding cryptogram resources.

Definition 2 Assembly Goal (AG) denotes functional and performance requirements of target protocol.

As complex protocols are formally derived from basic security components [7], AG could be decomposed into a sub-goal set $AG = \{rg_1, rg_2, \cdots, rg_n\}$, where each sub-goal can be matched to a security component. $\forall rg_i \in AG, 1 \le i \le n$, there exits $req(rg_i) = (fc_i, pm_i, pf_i)$, where fc_i, pm_i, pf_i respectively indicate functionality, interface and performance required by sub-goal rg_t. Intuitively speaking, there are multiple decomposition schemes for a given AG due to differences in assembly granularity. Thus, there may exist other sub-goal sets $AG' = \{rg'_1, rg'_2, \cdots, rg'_m\}$ that satisfies $AG = \{rg_1, rg_2, \cdots, rg_n\} = \{rg'_1, rg'_2, \cdots, rg'_m\} = AG'$.

Definition 3 Cryptography Resources (CR) denotes available cryptogram components in the access points

For each component, its attribute set is defined as a quintet $CC = (id, tp, fc, pf, pm)$, where id uniquely identifies a

component, tp points out its type, fc depicts its functionality, pf describes its performance, including execution time, energy consumption, safety level, etc., and pm denotes its input and output interfaces. All components together constitute the total cryptogram resources $CR = \{crc_i | i = 1,2,\cdots N\}$.

Definition 4 Component Efficiency (CE) refers to overall performance of resources or security protocol. Define the efficiency function as f_{CE}.

Definition 5 Assembly Solution (AS) depicts the result of decision-making for protocol assembly, which consist of a protocol flow and a set of cryptogram resources.

Based on Petri Net model [8], AS could be formulated as $AS = (State, Res, Flow)$. Where $Flow: State \leftrightarrow Res$ denotes one flow of target protocol, transition set $Res = \{c_1, c_2, \cdots, c_n\}$ are components of target protocol satisfying $Res \subset CR$ and all resources in Res together form the target protocol according to the $Flow$. Place $State$ refers to system states set. The solution with highest CE is the best AS AS_{best}.

Problem 1 ADMP problem: given $\{cc_i | i = 1,2,\cdots N\}$, target security protocol rsp and reconfiguration goal $AG(rsp)$, ADMP is to find optimal solution $AS_{best} = (\hat{S}, \hat{T}; \hat{F})$ that satisfies the following conditions,

(1) $\hat{T} = \{t_1, t_2, \cdots, t_m\} \subset CR$, and $\bigcup_{t \in \hat{T}} t \xrightarrow{\hat{F}} rsp$, where m is the size of $\hat{T}$and $\bigcup_{t \in \hat{T}} t \xrightarrow{\hat{F}} rsp$ indicates that all resources in $\hat{T}$together form protocol rsp according to protocol flow $\hat{F}$.

(2) There exists a sub-goal set $\{rg_1, rg_2, \cdots, rg_m\} = AG$ and a ono-to-one mapping $f: T \to RG$, where $\forall rg_j \in AG$, there is one and only one $t_i \in \hat{T}(i = 1,2,\cdots, n)$ satisfying $fc(t_i) = fc(rt_i), pm(t_i) = pm(rt_i), pf(t_i) \succ pf(rt_i)$, where $x \succ y$ means that ' x is better than y ', $fc(x), pm(x), pf(x)$ respectively denote functionality, interfaces and performance of x.

(3) $pf(RS_{best}) \succ pf(sp)$, which means that performance of AS_{best} is higher than the overall performance requirements.

(4) $\forall AS = (S, T; F)$, $f_{CE}(AS_{best}) \succ f_{CE}(AS)$.

III. A HIERARCHICALLY COLLABORATIVE ANT COLONY BASED ASSEMBLY ALGORITHM

This algorithm decomposes the ADMP into two sub-problems including sub-goals generation of resource selection and resources selection for each sub-goal. The former points out the directions of scheduling decision and the later implements the cryptogram resources selection sub-task, where both collaboratively generate an optimal solution of SDMP coordinately.

In details, at time *t*, the algorithm firstly apperceives a latent system state $s_{t+1}$and generates a cryptogram resources selection sub-goal $rg_{t+1}(0 \le t \le K-1)$, where K is the scale of sub-goals. The sub-goal rg_{t+1} points out the assembly directions and all $\{rg_{t+1}(0 \le t \le K-1)\}$ together cover the overall protocol assembly goals. Then, the algorithm follows the directions guided by sub-goal rg_{t+1} and schedules optimal resource $c_{t+1}^{'}$ for rg_{t+1} according to current system state s_{i+1}, which are combined to produce a candidate optimal solution. And a stochastic gradient algorithm based on Lévy Flight is conducted to verify and optimize the candidate solution.

A. Sub-goal Generation

As mentioned above, there may be multiple flows corresponding to the target protocol owing to difference in assembly granularity. Taking protocol flows in Figure Ⅰ for instance, there are seven distrinct flows between state $s_{i-1}$and state s_{i+3}. At state s_i, there are 3 alternative sub-goals to construct the protocol. The sub-task selects one and generates the sub-goal rg_i. When the sub-goal rg_i is accomplished, state transition from s_i to s_{i+1} is triggered.

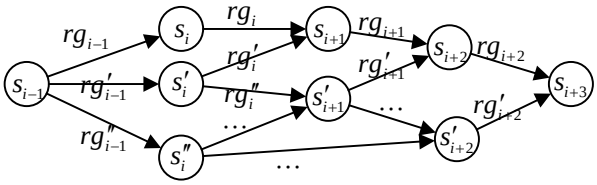

FIGURE I. PROTOCOL FLOWS AND STATE TRANSITION

To generate proper sub-goal, an ant colony algorithm with Q-learning is adopted for training, which combines the advantage of optimization theory, non-linear control and reinforce learning. This sub-task takes current system state s_i and RG as input and outputs the sub-goal. When producing sub-goals, an adaptive pseudorandom ratio selection rule is introduced at probability θ_0, and the sub-goal generation policy indicated by flow pheromone is adopted at probability $1 - \theta_0$. What's more, sub-goal generation policies are trained by updating flow pheromone, which is detailed in Sect. 4.3. At each system state, a sub-goal is generated according to policies shown in eq. 1-2.

$$f^{Eant}(s_t) = \begin{cases} argmax\left([\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}(t)]^{\beta}\right); \theta \le \theta_0 \\ p_{ij}^{k}(t); \theta > \theta_0 \end{cases} \quad (1)$$

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ik}(t)]^{\beta}}{\sum_{s \subset allowed_i} [\tau_{is}(t)]^{\alpha} \cdot [\eta_{is}(t)]^{\beta}}, k \subset allowed_i \\ 0, k \not\subset allowed_i \end{cases} \quad (2)$$

$\tau_{ij}(t)$ is the amount of pheromone deposited for the sub-goal that triggers state transition ij, and $\alpha \ge 0$ is heuristic factor controlling the influence of $\tau_{ij}(t)$. $\eta_{ik}(t) = \frac{1}{f_{CE}(res_{ik})}$ heuristic function indicating the expectation that the sub-goal triggering state transition ij is generated, $\beta \ge 1$ is the expectational heuristic factor. And $allowed_i$ denotes possible states set that state i may transfer to. Sub-goal generation is implemented repeatedly until the target protocol ends.

B. Cryptogram Resources Selection

Once a sub-goal is generated, the algorithm computes best cryptogram resource matching the sub-goal. And then, cryptogram resources for all sub-goals together constitute the candidate optimal solution for ADMP. Finally, candidate optimal solution is verified and optimized to obtain the final optimal solution.

1) Secure access resource matching for sub-goal

Scenario of resource matching for a sub-goal is depicted in Figure Ⅱ, where given $\forall rg_i(i = 1,2,\cdots t)$, there may exist a cryptogram resources set $RR_i \in RR$ that could satisfy the functional requirements and performance index of the sub-goal rg_i. Cryptogram resource matching aims at finding best-suitable cryptogram resource for each received sub-goal.

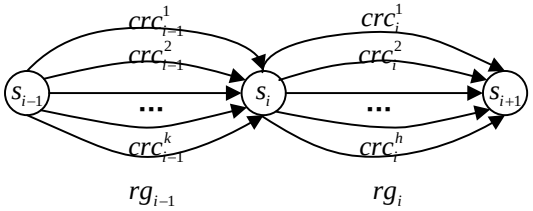

FIGURE II. SCENARIO OF CRYPTOGRAPHIC RESOURCE MATCHING

Similarly, to address this issue, an ant colony algorithm that includes just once cryptogram resource matching is introduced. Each matching aim at finding the preponderant cryptogram resource for current sub-goal. Rules for cryptogram resource matching is depicted in eq. 3-4.

$$f_{lant}^{W}(rg_t) = \begin{cases} argmax\left(\left[\tau_{c_i}(t)\right]^{\alpha} \cdot \left[\eta_{c_i}(t)\right]^{\beta}\right); \theta \le \delta \\ p_{c_i}(t); \theta > \delta \end{cases} \quad (3)$$

$$p_{c_i}(t) = \begin{cases} \frac{\left[\tau_{c_i}(t)\right]^{\alpha} \cdot \left[\eta_{c_i}(t)\right]^{\beta}}{\sum_{c_j \subset mt(rg_i)} \left[\tau_{c_j}(t)\right]^{\alpha} \cdot \left[\eta_{c_j}(t)\right]^{\beta}}, c_i \subset MT(rg_i) \\ 0, c_i \subset MT(rg_i) \end{cases} \quad (4)$$

Where $\delta \in [0,1]$and a random walk rule depicted in eq. 3 is adopted when $\theta \le \delta$, otherwise a determinate matching policy shown in eq. 4 is adopted. $\tau_{c_i}(t)$is the amount of pheromone for cryptogram resource c_i and $\eta_{c_i}(t)$ is heuristic function indicating the expectation that c_i matches rg_i. $MT(rg_i)$ is resources set that matching sub-goal rg_i. c_i is the i-th component of the solution after each matching process.

2) Candidate solution producing

Candidate solution producing adopts the idea of concentrating group wisdom. Once all epochs are accomplished, to incorporate preponderant components of each solution, all generated solution are embedded into preponderant component extracting vectors using a linear projection ϕ. Then, all vectors are pooled by summation to produce the candidate solution assuming that $AS_0^{ca}$is the candidate optimal solution and AS_i denotes the i-th solution, then there exists

$$\mathrm{AS}_0^{\mathrm{ca}} = \sum_{\mathrm{i}=1}^{\mathrm{K}} \phi(\mathrm{AS}_\mathrm{i}) \quad (5)$$

Where K denotes the scale of all solutions.

3) Solution verification and optimization

To guarantee that the real optimal solution is obtained, a Lévy Flight [6] based stochastic gradient algorithm is employed, which introduces the advantages of frequent short-distance steps and accidental long-distance steps of Lévy flight into stochastic gradient policies.

Assume AS_0^{ca} the initial candidate solution andX_i^{ca} to be the i-th updating of AS_0^{ca}, the solution updating rule is formulized as follows

$$AS_{i+1}^{ca} = \left(f_{CE}(AS_i^{ca}) - f_{CE}(AS_{lbest})\right) \oplus \sigma \oplus Levy(\xi) + AS_i^{ca} \quad (6)$$

Where $\sigma$is step factor controlling the range of optimization and $\sigma = 1$can be used usually in most cases, AS_{lbest}denotes the historically best solution, $Le\acute{}vy(\xi)$ provides the random step length from a Lévy distribution.

$$Levy(\xi) \sim \mu = t^{-1-\xi}, 0 < \xi \le 2 \quad (7)$$

For ease of calculation, eq. 8 is adopted to calculate the Lévy random number.

$$\mathrm{Levy}(\xi) \sim \frac{\phi \times \mu}{|\nu|^{1/\xi}} \quad (8)$$

Where μ, ν follow the normal distribution and $\xi = 1.5$.

$$\phi = \left(\frac{\Gamma(1+\xi) \times sin(\pi \times \xi/2)}{\Gamma\left(\frac{1+\xi}{2}\right) \times \xi \times 2^{(\xi-1)/2}}\right)^{1/\xi} \quad (9)$$

To speed the convergence process of optimal solution optimization, some solutions are discarded and substituted by new solutions with certain probability. These new solutions are generated based on random walk policy as follows.

$$\mathrm{RS}_{\mathrm{i}+1}^{\mathrm{ca}} = \mathrm{RS}_\mathrm{i}^{\mathrm{ca}} + \gamma(\mathrm{RS}_\mathrm{i}^{\mathrm{ca}} - \mathrm{RS}_\mathrm{k}^{\mathrm{ca}}) \quad (10)$$

Where RS_k^{ca} is a discarded solution and γ is step factor.

After aforementioned solution optimization, assume RS_n^{ca} to be the optimized solution, and then the final optimal solution RS_{best} can be obtained with (17).

$$AS_{best} = argmax\left(f_{CE}(AS_{best}), f_{CE}(AS_n^{ca})\right) \quad (11)$$

C. Hierarchical Pheromone Updating

Additionally, a hierarchical pheromone is defined to reinforce the positive feedback of each sub-task, which is defined as follows

$$\mathrm{phe} = \{\mathrm{phe}^{\mathrm{f}}, \mathrm{phe}^{\mathrm{r}}\} \quad (12)$$

Where the upper pheromone $phe^f = \{phe_{ij}^f | i, j \in N\}$ denotes pheromone on sub-goals and phe_{ij}^f is the amount of pheromone on the sub-goal triggering state transition ij. While the lower pheromone $phe^r = \{phe_i^r | i \in N, 0 < i < Num\}$ refers to as pheromone for cryptogram resources and phe_i^r is the amount of pheromone on the i-th cryptogram resource.

A pheromone updating process will be triggered when sub-goal generating or cryptogram resource selection is accomplished. Due to the differences in adopted ant colony algorithms, different pheromone updating rules are adopted.

Pheromone updating for sub-goals

Pheromone updating for sub-goals occurs mainly in the sub-goal generation phase. As sub-goal generation adopts the Ant colony algorithm with Q-learning, once the sub-goal is generated, the pheromone of this sub-goal is updated according to the following rule.

$$phe_{ij}^f(t+n) = (1-\rho_f)phe_{ij}^f(t) + \rho_f\left(\Delta phe_{ij}^f + \gamma \cdot \max_{k \in allowed_j} phe_{ik}^f(t)\right) \quad (13)$$

$$\Delta phe_{ij}^f = \frac{AE(X_{best}^t) - AE(X_{best}^{t+n})}{AE(X_{best}^{t+n})} \quad (14)$$

Where ρ_f denotes the volatilization factor of protocol flow pheromone. Δphe_{ij}^f depicts increment of sub-goal pheromone triggering state transition ij. And, γ is the discount factor for computing the accumulated reward. Finally, after the algorithm is trained, the sub-goals pheromone could be transferred into sub-goal generation policies.

1) Updating of pheromone for cryptogram resources

Pheromone updating for cryptogram resources is triggered in two occasions, namely the cryptogram resource matching phase and the solution verification and optimization phase. During the cryptogram resource matching phase, when the best-suitable cryptogram resource for each received sub-goal is found, pheromone for this best-suitable cryptogram resource will be updated. Meanwhile, when the optimal solution is generated in solution verification and optimization phase, pheromones for all cryptogram resources constituting the optimal solution will be updated. The updating rule can be formulized as follows.

$$phe_c^r(t+n) = (1-\rho_r)phe_c^r(t) + \Delta phe_c^r \quad (15)$$

Where ρ_r denotes the volatilization factor of resources pheromone, Δphe_c^r depicts the increment of pheromone on cryptogram resources c. In cryptogram resource matching phase, $\Delta phe_c^r = \frac{W}{f_{CE}(c)}$. While in solution verification and optimization phase, $\Delta phe_c^r = \frac{W}{f_{CE}(X_{lbest})}$, where X_{lbest} is local optimal solution in an epoch and W is a constant variable. When the hierarchical pheromone is trained, it could be transferred into scheduling policies.

IV. EXPERIMENTS

A. Settings

Experiments are carried out to analyze the convergence and performance of the proposed algorithm. A simplified cryptogram component set listed in table I.

TABLE I. SIMPLIFIED CRYPTOGRAM COMPONENTS

Security Service	Cryptographic algorithm and components
Confidentiality	S-DES; DES; 3-DES; IDEA; Blowfish; CAST-128; CAST-256; A5/1; A5/2; RC4; RC5; RC6; Elgamal;
Integrity	MD4; MD5; SHA-1; SHA-2; RIPEMO-160;
Authentication	Hash, RSA, Schnorr; ElGamal, DSS, ECC
Anti-replay	Linear Congruence, Normal distribution, Monte
Non-repudiation	Hash, RSA, Schnorr; ElGamal, DSS, ECC

Security protocols including Hsieh's scheme [9], Jiang's scheme [10], Priauth [11], Chen's scheme [12] are adopted as target security protocols.

B. Convergence Analysis

Convergence analysis focuses on the influences of cryptogram resources requirements of security protocol, total scale of cryptogram resources, the step length of global optimization, the pheromone volatility factor and the population collaboration on the algorithm convergence. And the stopping condition for training is set to be that the deviation of two adjacent solutions is less than 0.01.

1) Levy step of the global optimization

In solution verification and optimization, a levy flight based stochastic gradient decline algorithm is adopted. To analyze the influence of levy flight step on convergence, the fixed step and the random step are taken as benchmarks.

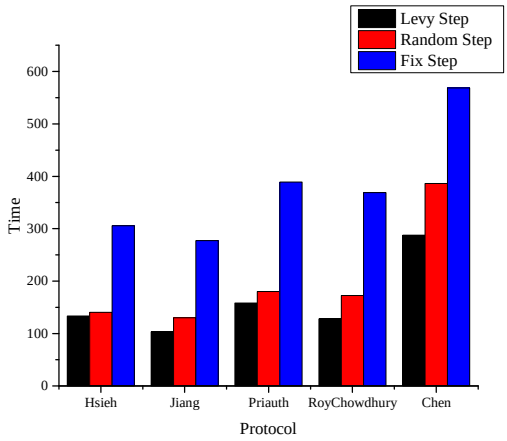

FIGURE III. INFLUENCE OF LEVY STEP

From Figure III, we can find that the levy flight step outperforms the random step and fixed step in convergence. After analysis, we can see that the levy flight based stochastic gradient decline algorithm combines long step and short step, where steps can be adjusted according to the distance to the optimal solution, resulting in faster convergence. Additionally, compared to fixed step, random step possesses faster convergence.

2) Volatility factor of pheromone

Set total size of cryptogram resources be 100. Choose Hsieh's scheme as target protocol. As both flow pheromone and resource pheromone have its own volatility factor, the influence of these two volatility factors are tested separately.

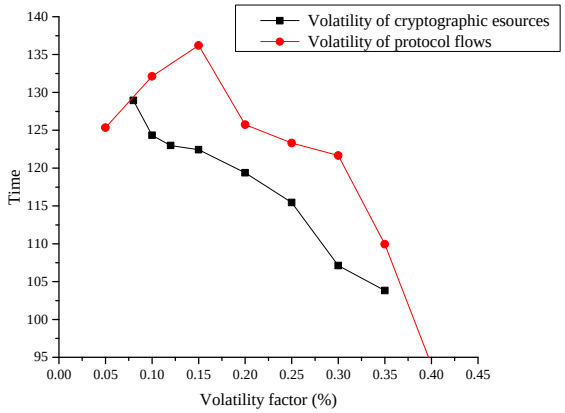

FIGURE IV. INFLUENCE OF VOLATILITY FACTOR

From figure IV, we can find that the pheromone volatility factor directly influences the convergence rate. Intuitively, larger volatility factor leads to faster convergence rate. As the volatility factor get larger, the pheromones of unvisited cryptographic resources or protocol flow drop to 0 much faster and the probability that the visited ones are selected again get larger, leading to the speed-up of convergence, which also loses the randomness. Additionally, the influences of volatility factor for protocol flow pheromone and resources pheromone are equal, which further points out the signification of coordination.

3) Population collaboration

Set the total size of cryptogram resources to be 100, and choose Hsieh's scheme as target protocol. Change ant number in population of each sub-task to adjust the rate of collaboration.

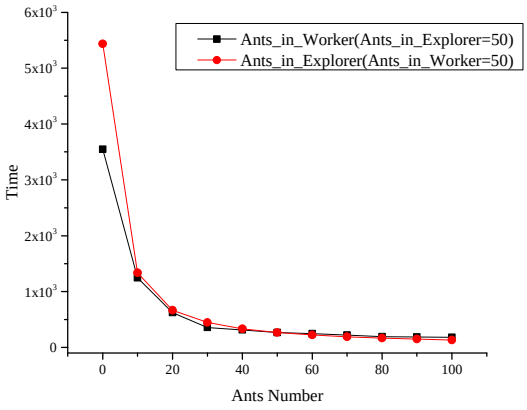

FIGURE V. INFLUENCE OF POPULATION COLLABORATION

From Figure V, when ant number in the population of either sub-task drops to 0, the convergence gets slower. When the ant number in population in both sub-task equals 0, population collaboration disappears and degenerates to Cuckoo Search algorithm and Ant-Q algorithm respectively, and the convergence time reaches the ceiling.

C. Performance Analysis

Set size of whole cryptogram resources to be 100, and take ant colony algorithm [13], cuckoo search algorithm [14] and reinforcement learning [15] as benchmark algorithm.

TABLE II. PERFORMANCE ANALYSIS

Algorithms	**Items**	**Protocols**			
		Hsieh	Jiang	Priauth	Chen
Ant Colony	Time	2267.5	3016.7	2659.4	5453.3
	Accuracy	88.9%	86.7%	87.6%	82.5%
Cuckoo Search	Time	1653.8	2543.1	1956.6	4431.6
	Accuracy	92.5%	91.7%	92.1%	90.2%
Reinforcement Learning	Time	455.3	642.5	574.2	996.3
	Accuracy	98.8%	98.0%	98.3%	95.6%
HiCoACS	Time	234.8	323.2	277.9	534.3
	Accuracy	99.5%	99.4%	99.5%	99.1%

From table II, it could be seen that our algorithm possesses less time cost and higher accuracy, and outperforms given benchmark algorithms in accuracy and efficiency. This result in some degree demonstrates the advantages of combining hierarchical reinforcement learning and collaborative behavior of biological populations.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and all the anonymous reviewers for their insightful comments and constructive suggestions. The authors acknowledge the National Natural Science Foundation of China (Grant Nos. 61502531, 61403400, 61773399 and 61802436).

REFERENCES

[1] L. Bossuet, V. Fischer, L. Gaspar, et al., "Disposable configuration of remotely reconfigurable systems," MICROPROCESS MICROSY, vol. 39, no. 6, pp. 382-392, 2015.

[2] S. Yoon, H. Park, H. S. Yoo, "Security Issues on Smarthome in IoT Environment," Lecture Notes in Electrical Engineering, no. 330, pp. 691-696, 2015.

[3] A. S. Vezhnevets, S. Osindero, T. Schaul, et al., "FeUdal Networks for Hierarchical Reinforcement Learning," Proc ICML2017, 2017, pp. 3540-3549.

[4] T. D. Kulkarni, K. R. Narasimhan, A. Saeedi, et al.. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation," Proc NIPS2016, 2016, pp. 3675-3683.

[5] Q. Niu, T. Zhou, L. Wang, "A hybrid particle swarm optimization for parallel machine total tardiness scheduling," INT J ADV MANUF TECH, vol. 49, no. 5-8, pp. 723-739, 2010.

[6] A. M. Reynolds, "Cooperative random lévy flight searches and the flight patterns of honeybees," PHYS LETT A, vol. 354, no. 5, pp. 384-388, 2006.

[7] A. Datta, A. Derek, J. C. Mitchell, et al., "Secure protocol composition," FMSE, 2003, pp. 11-23.

[8] M. Nielsen, G. Plotkin, G. Winskel, "Configuration structures, event structures and Petri nets," THEOR COMPUT SCI, vol. 410, no. 41, pp. 4111-4159, 2009.

[9] Hsieh, B. Wen, Leu, S. Jenq, "Anonymous authentication protocol based on elliptic curve Diffie–Hellman for wireless access networks," WIREL COMMUN MOB COM, vol. 14, no. 10, pp. 995-1006, 2014.

[10] Q. Jiang, J. Ma, G. Li, et al., "An Efficient Ticket Based Authentication Protocol with Unlinkability for Wireless Access Networks," WIRELESS PERS COMMUN, vol. 77, no. 2, pp. 1489-1506, 2014.

[11] He, J Bu, S Chan, et al., "Privacy-Preserving Universal Authentication Protocol for Wireless Communications," IEEE T WIREL COMMUN, vol. 10, no. 2, pp. 431-436, 2011.

[12] C. Chen, D. He, S. Chan, et al., "Lightweight and provably secure user authentication with anonymity for the global mobility network," INT J COMMUN SYST, vol. 24, no. 3, pp. 347-362, 2011.

[13] A. Colorni, M. Dorigo and V. Maniezzo, “Distributed Optimization by Ant Colonies,” ECAL91, 1991, pp. 134-142.

[14] X. S. Yang, S. Deb, “Cuckoo Search via Lévy flights,” NaBIC 2009, 2010, pp. 210-214.

[15] A. Thomas, S. I. Marcus, “Reinforcement learning for MDPs using temporal difference schemes,” IEEE CDC’97, 1997, pp. 577-583.