HiCoACR: A reconfiguration decision-making model for reconfigurable security protocol based on hierarchically collaborative ant colony

Zhuo Yi¹,², Ying Liao¹,³, Xuehui Du¹, Xin Lu¹ and Lifeng Cao¹

Abstract
Reconfigurable security protocols, with dynamic protocol configuration and flexible resource allocation, have become a state-of-the-art technology to guarantee the security of space-ground integrated network. However, reconfiguration decision-making for reconfigurable security protocols remains a major challenge in order to adapt to diverse secure service requirements and deploy higher security level but more complicated security strategies in nodes with limited resources and computing abilities. To handle this problem commendably, a hierarchically collaborative ant colony–based reconfiguration decision-making model called HiCoACR is proposed. This model, inspired by the ideas of hierarchical reinforcement learning and population collaboration, decomposes the reconfiguration decision-making problem into two sub-problems by introducing a two-level hierarchy ant colony consisting of the Explorer and the Worker. The Explorer controls directions of protocol reconfiguration and generates abstract scheduling sub-goals which are conveyed from the Worker. While the Worker schedules most suitable cryptogram resources for each sub-goal received and produces the optimal reconfiguration solution which is verified and re-optimized by a Lévy process–based stochastic gradient descent algorithm. Both the Explorer and the Worker adopt a modified version of ant colony algorithm to fulfill its targets, where a hierarchical pheromone is defined to reinforce positive behaviors of each ant colony. Experiment results suggest that HiCoACR outperforms baseline algorithms and possesses well model transferability.

Keywords
Reconfiguration decision-making, cryptogram resources scheduling and optimization, hierarchically collaborative ant colony, reconfigurable security protocol, space-ground integrated network

Introduction
Security protocol is a primary means for guaranteeing the security of network services, such as network communication, data transmission, and authentication. However, no one security protocol could sufficiently cater to all the diverse security requirements, especially in space-ground integrated network (SGIN), where severely limited resource further decreases the possibility of fulfilling diverse security requirements and deploying higher security level but more complicated security strategies.
Reconfigurable secure protocol (RSP), with adaptively dynamic protocol reconfiguration and flexible cryptogram resource configuration,\textsuperscript{1,5} offers a great possibility of alleviating resources constrains, improving resources utilization and enhancing network security in SGIN. Security protocol reconfiguration (SPR) generates protocols by determining proper protocol flow and its components, picking cryptogram resources that satisfy the functional requirements and performance indexes for each component, and then assembling these selected resources in terms of the protocol flow. Since the performance of all components determines the efficiency of the generated target protocol, one of the key issues in SPR is to determine the corresponding protocol reconfiguration flow and its components and schedule optimal cryptogram resources so as to form the target security protocol, which is defined as reconfiguration decision-making problem (RDMP) in this article.

Generally, the scales of cryptogram resources and the diversities in design standards, cryptosystem, and application situations of cryptogram resources enlarge the solution space of RDMP. What is worse, the uncertainty of protocol flow caused by reconfiguration granularity, further intensify this situation. In addition, RDMP is a combinatorial optimization problem, a nondeterministic (NP)-hard problem, where the optimal solution relies on both reconfiguration protocol flow and cryptogram resources. However, previous works mainly focus on resources scheduling and ignore the role of reconfiguration flow in RDMP. Thus, designing an accurate and efficient reconfiguration decision-making mechanism is of great significance to RSP.

To address this problem, a hierarchically collaborative ant colony-based reconfiguration decision-making model called HiCoACR is proposed. The HiCoACR model takes inspiration from hierarchical reinforcement learning\textsuperscript{5,14} and collaborative behaviors of biological populations,\textsuperscript{5} where the RDMP is decomposed into two sub-problems including sub-goal generation for resource scheduling and resource scheduling for sub-goals. And a hierarchical coordinated ant colony consists of the Explorer and the Worker is introduced to address these two sub-problems, respectively. In details, the Explorer controls directions of protocol reconfiguration and explores abstract resources scheduling sub-goals at lower temporal resolution in a latent state space, while the Worker schedules proper cryptogram resources for the sub-goals deriving from the Explorer and generates optimal solution for the target protocol. Both the Explorer and the Worker operate with an improved ant colony algorithm and a hierarchical pheromone is defined to reinforce the positive behaviors of the Explorer and the Worker. In addition, a Lévy theory-based stochastic gradient algorithm is adopted to verify and re-optimize the solution produced by the Explorer and the Worker so as to generate the best reconfiguration solution.

The key contributions of this article include (1) a reconfiguration decision-making model HiCoACR for reconfigurable security protocol that decouples RDMP into scheduling sub-goal generation and resources scheduling for sub-goals; (2) a hierarchical pheromone that indicates the reconfiguration policies of HiCoACR and its updating mechanism; (3) a Lévy flight-based stochastic gradient algorithm to verify and re-optimize the optimal solution.

Related works

Naturally speaking, the core of RDMP lies in optimization scheduling, which can be classified into static scheduling and dynamic scheduling. Conventional methods for static scheduling include genetic algorithm,\textsuperscript{7–9} particle swarm,\textsuperscript{10,11} heuristic search,\textsuperscript{12,13} and so on. Dynamic scheduling algorithms include load balancing,\textsuperscript{14–16} fuzzy stochastic optimization,\textsuperscript{10,17} deep reinforcement learning,\textsuperscript{18,19} and so on. A heuristic search algorithm is adopted to find the optimal reconfiguration scheme for the target security protocol in given solution space.\textsuperscript{20} A multi-objective evolutionary algorithm (MOEA)–based reconfigurable optimization algorithm is introduced to address energy-saving issues in reconfigurable sensor network, where a specific MOEA framework is applied to make a reasonable trade-off choice from the set of Pareto-optimal solutions according to their preferences and system requirements.\textsuperscript{21} A quality of service (QoS)-aware service composition algorithm is proposed to handle dynamic web service composition problem by taking the advantages of both global optimization and local optimization. In this article, mixed integer programming is adopted to find the optimal decomposition of global QoS constraints into local constraints and distributed local selection is introduced to find the best web services that satisfy these local constraints.\textsuperscript{22} A cuckoo’s breeding behavior–based heuristic method and a natural gene evolution–based heuristic method are proposed to find the best solution or near best solution of service composition, where the former applies a 1-OPT (algorithm that could reaches the solution by changing a single element in the current solution) heuristic to expand the search space in a controlled way and the later uses two memory structures to avoid the stagnation in a local optimum solution, and to ensure that exploitation and exploration are properly performed.\textsuperscript{23} A heuristic algorithm T-HEU (a trust-based heuristic algorithm) combining trust-based selection, convex hulls, and global optimization is proposed to cope with the service composition problem. In T-HEU, the trust-based selection method is used to filter untrustworthy component services, the convex hulls to reduce the search space in the process of service composition, and
the heuristic global optimization approach to obtain the near-optimal solution. A systematic approach based on a fuzzy linguistic preference model and an evolutionary algorithm is put forward to solve the service level agreement (SLA)-constrained service composition problems. Specifically, the weighted Tchebycheff distance is introduced first to model this problem, and the consequences of a step may unfold gradually over many subsequent steps and choices. In general, model-free reinforcement learning and model-based reinforcement learning are the main solutions. A FeUdal network, a novel architecture for hierarchical reinforcement learning, is proposed to solve sequential decision task such as playing computer games (such as Montezuma’s revenge), which provides great inspiration for method proposed in this article. All aforementioned works serve as inspiration for the computational methods developed in this article.

Formalization of RDMP

Definition 1. Reconfiguration model for SPR is defined as a quadruples \( RSP = (RG, RR, RE, RX) \), where \( RG \) denotes reconfiguration goals, depicting functional and performance requirements of the target security protocol. \( RR \) refers to as all available cryptogram resources including cryptocards, cryptographic devices developed on field-programmable gate arrays (FPGAs), or reconfigurable processors. \( RE \) represents reconfiguration efficiency criteria for protocol performance evaluation. \( RS \) indicates reconfiguration solutions of RDMP for the given protocol. One reconfiguration solution often consists of protocol flow and corresponding cryptogram resources.

Definition 2. \( RG \) denotes functional requirements and minimum performance indexes of the target protocol.

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

Definition 3. Cryptography reconfigurable cell (CRC) are independent cryptogram components which are reconfigurable in functional, structural, and temporal dimensions.

For each CRC, its attribute set is defined as a quintet \( CRC = (id, tp, fc, pf, pm) \), where \( id \) uniquely identifies a CRC, \( tp \) points out the type of CRC, such as atom CRC and compound CRC, \( fc \) depicts the functionality of CRC, \( pf \) describes the performance of CRC, including execution time, energy consumption, safety level, and so on, and \( pm \) denotes its input and output interfaces. All CRCs together constitute the total cryptogram resources \( RR = \{crc_i | i = 1, 2, \ldots, N\} \).

Definition 4. \( RE \) refers to overall performance of cryptogram resource or a security protocol. Efficiency of security protocol \( SP \) is recorded as \( RE(SP) \), and efficiency of a cryptogram resource \( crc_i \) is recorded as \( RE(crc_i) \). Efficiency of a protocol is usually evaluated using certain efficiency model, which is described in section “RE evaluation.”

Definition 5. Reconfiguration solution (RS) depicts the result of reconfiguration decision-making for SPR, which consist of a protocol flow and a set of cryptogram resources constituting the flow.

Based on Petri Net model, \( RS \) could be formulated as \( RS = (State, Res; Flow) \). Where \( Flow : State \leftrightarrow Res \) denotes the flow of a target protocol, transition set \( Res = \{c_1, c_2, \ldots, c_n\} \) are components of the target protocol satisfying \( Res \subset RR \), and all resources in \( Res \) together form the target protocol according to the \( Flow \). Place \( State \) refers to the system state set. As there exist more than one \( RS \) satisfying \( RG \), solution with highest \( RE \) is regarded as the best \( RS \) and recorded as \( RS_{best} \).

Problem 1. RDMP for SPR is defined as follows: given reconfiguration resource \( RR = \{crc_i | i = 1, 2, \ldots, N\} \), a target security protocol \( rsp \), and corresponding reconfiguration goal \( RG(rsp) \), RDMP is to find the optimal solution \( RS_{best} = (S, T; F) \) that satisfies the following conditions:

1. \( T = \{t_1, t_2, \ldots, t_m\} \subset RR \), and \( \bigcup_{t \in T} t \xrightarrow{F} rsp \), where \( m \) is the size of \( T \) and \( \bigcup_{t \in T} t \xrightarrow{F} rsp \) indicates that all cryptogram resources in \( T \) together form the target protocol \( rsp \) according to the protocol flow \( F \).
2. There exist a sub-goal set \( \{ rg_1, rg_2, \ldots, rg_m \} = RG \) and a one-to-one mapping \( f : T \rightarrow RG \), where \( \forall x \in RG, \) there is one and only one \( t_i \in T(i = 1, 2, \ldots, n) \) satisfying equations
\[
fc(t_i) = fc(rt_i), pm(t) = pm(rt_i), pf(t_i) > pf(rt_i),
\]
where \( x \succ y \) means that “\( x \) is better than \( y \),” \( fc(x), pm(x), pf(x) \) denotes functionality, interfaces, and performance of \( x \), respectively.

3. \( pf(RS_{best}) \succ pf(rsp) \), which means that the overall performance of \( RS_{best} \) is higher than the overall performance requirements of the target protocol \( rsp \).

4. \( \forall RS = (S, T; F), RE(RS_{best}) \succ RE(RS) \).

According to the above definitions, RDMP is to find the best solution with highest RE under the condition of meeting basic functional requirements and performance index. Figure 1 briefly illustrates the RDMP, where dotted lines with different colors represent different reconfiguration solutions and the green dotted line is the best solution.

**The HiCoACR model**

HiCoACR is a two-level hierarchy architecture consisting of two modules—the Explorer and the Worker, each of which implements certain sub-tasks of reconfiguration decision-making. The Explorer points out the reconfiguration directions of protocol flow and generates scheduling sub-goal, while the Worker implements the cryptogram resources scheduling sub-tasks, where both together generate an optimal solution of RDMP coordinately.

In details, at time \( t \), the Explorer apperceives a latent system state \( s_{t+1} \) and generates a cryptogram resources scheduling sub-goal \( rg_{t+1}(0 \leq t \leq K - 1) \), where \( K \) is the scale of sub-goals. The sub-goal \( rg_{t+1} \) points out the reconfiguration directions and all \( \{ rg_{t+1} (0 \leq t \leq K - 1) \} \) together cover the overall protocol \( RG \), which also indicate the components of the selected protocol flow. The Worker follows the directions guided by sub-goal \( rg_{t+1} \) and schedules optimal resource \( c_{t+1} \) for \( rg_{t+1} \) according to \( s_{t+1} \). Then all components \( \{ c_{t+1} \} \) are combined to produce the candidate optimal solution. And a stochastic gradient algorithm based on Lévy flight is conducted to verify and optimize the candidate solution so as to get the final optimal solution.

Both the sub-goal generation process in the Explorer and the resource scheduling process for sub-goal in the Worker are trained by a modified version of ant colony algorithm. Meanwhile, a hierarchical pheromone \( phe \) is introduced to reinforce the positive behaviors of the Explorer and the Worker during training and to produce reconfiguration policy during reconfiguration decision-making. In essence, the greater the RE of a RS is, the higher the concentration of \( phe \) is and the policies that the related protocol flow and resources are to be selected with higher probabilities will form.

The overall design of HiCoACR model is illustrated in Figure 2 and equations (1)–(6) depict the forward dynamics of HiCoACR model

\[
s_{t+1}, rg_{t+1}, h^E_{t+1} = f^{End}(RG, s_t, h^E_t) \tag{1}
\]

\[
c_{t+1}, h^W_{t+1} = f^{W}(rg_t+1) \tag{2}
\]
\[
RS_i \subseteq C_i; \quad RS^{ca}_0 = f_{\text{Merge}} \left( \bigcup RS \right)
\]

\[
RS^{ca}_i, h^{Wg}_i = f_W^{\text{ant}} \left( RS^{ca}_0, h^{Wg}_0, h^{Wg}_0 \right)
\]

where \( h^{Wg}_i, h^{Wg}_{i+1}, h^{W}_i, h^{W}_{i+1}, h^{Wg}_0, h^{Wg}_{i+1} \) are intermediate variables, \( i \) in equation (3) indicates the \( i \)th epoch of reconfiguration solution exploring, and \( n \) in equation (4) is times of solution updating. Function \( f^{Eant} \) in equation (1) aims to generate the next sub-goal with current system state and RG, which is detailed in section “Sub-goal generation.” Function \( f^{Want} \) in equation (2) obtains the best-matched cryptogram resource for the sub-goal generated in equation (1), details of which are illustrated in the first part in section “Cryptogram resources scheduling.” Equation (3) combines all the resources in one epoch to get a feasible solution and function \( f_{\text{Merge}} \) computes the candidate solution by merging feasible solutions got in different epochs, which refers to the second part in section “Cryptogram resources scheduling.” Equation (4) verifies and optimizes the candidate solution with function \( f_W^{\text{ant}} \) and equation (5) chooses the final best solution between the optimized candidate solution and historical best solution. The details of function \( f_W^{\text{ant}} \) are given in the third part in section “Cryptogram resources scheduling.” Equation (6) details the rules of hierarchical pheromone updating and policy generating, which are described in section “Hierarchical pheromone updating.”

**Sub-goal generation**

The intention of the Explorer is to produce a sequence of sub-goals guiding the directions of reconfiguration. Each sub-goal represents a component of the protocol flow. As mentioned above, there may be multiple flows corresponding to the target protocol due to difference in reconfiguration granularity. Taking protocol flows in Figure 3, for instance, there are seven distinct flows between state \( s_{i-1} \) and state \( s_{i+3} \). At state \( s_i \), there are three alternative sub-goals to construct the protocol. The Explorer selects one and generates the sub-goal \( rg_i \). When the sub-goal \( rg_i \) is accomplished by the Worker, state transition from \( s_i \) to \( s_{i+1} \) is triggered.

Given certain security protocol, its RG could be expressed as a goal vector \( RG \in R^k \). And for each cryptogram resource \( c \in RR \), it will satisfy the requirements of one sub-goal. Thus, the Explorer first produces a goal vector for each cryptogram resource. All sub-goal vectors of the whole cryptogram resources could form a matrix.

To generate proper sub-goal, the Explorer adopts an ant colony algorithm with Q-learning, which combines the advantage of optimization theory, non-linear control, and reinforce learning. The Explorer takes current system state \( s_i \) and RG as input and outputs the sub-goal. In this algorithm, when producing sub-goals, an adaptive pseudorandom ratio selection rule is introduced at probability \( \theta_0 \), and the sub-goal generation policy indicated by flow pheromone is adopted at probability \( 1 - \theta_0 \). What is more, sub-goal generation policies are trained by updating flow pheromone, which is detailed in section “Hierarchical pheromone updating.”

At each system state, the Explorer generates a sub-goal according to policies given in equations (7) and (8).

\[
f^{Eant}(s_i) = \begin{cases} 
\arg \max \left( \left[ \xi_k(t) \right]^{\theta_1} \cdot \left[ \eta_k(t) \right]^{\theta_1} \right) ; \theta \leq \theta_0 \\
p_k(t) ; \theta > \theta_0
\end{cases}
\]
that could satisfy the functional requirements. And there may exist a cryptogram resources set matching sub-goal for each received sub-goal. Rules for cryptogram resource matching are depicted in equations (9) and (10):

\[
f_{\text{best}}(rg_i) = \left\{ \begin{array}{ll}
\arg \max \left[ (\tau_{\text{c}}(t))^\theta \cdot (\eta_{\text{c}}(t))^\beta \right] & : \theta \leq \delta \\
\arg \max \left[ (\tau_{\text{c}}(t))^\theta \right] & : \theta > \delta
\end{array} \right.
\]

\[
p_{\text{c}}(t) = \left\{ \begin{array}{l}
\sum_{c_i \in MT(rg_i)} (\tau_{\text{c}}(t))^\theta (\eta_{\text{c}}(t))^\beta, c_i \subset MT(rg_i) \\
0, c_i \notin MT(rg_i)
\end{array} \right.
\]

where \(\tau_{\text{c}}(t)\) is the amount of pheromone deposited for the sub-goal that triggers state transition \(ij\), and \(\alpha_1 \geq 0\) is the heuristic factor reflecting the relative importance of pheromones \(\tau_{\text{c}}(t)\) in guiding sub-goal generation process. And \(\eta_{\text{c}}(t) = 1/RE(\text{res}_{\text{c}})\) is heuristic function indicating the expectation that the sub-goal triggering state transition \(ij\) is generated, and \(\beta_1 \geq 1\) is the expectational heuristic factor reflecting the relative importance of heuristic factor \(\alpha_1\) in guiding sub-goal generation process. And \(\text{allowed}\), denotes possible states set that state \(i\) may transfer to. Sub-goal generation is implemented repeatedly until the target protocol ends.

**Cryptogram resources scheduling**

Once receiving a sub-goal conveyed from the Explorer, the Worker first computes best cryptogram resource matching the sub-goal. And then, cryptogram resources for all sub-goals together constitute the candidate optimal solution for RDMP. Finally, candidate optimal solution is verified and optimized to obtain the final optimal solution.

**Cryptogram resource matching for sub-goal**. Scenario of cryptogram resource matching for a sub-goal is depicted in Figure 4, where given \(\forall \text{rg}_i(i = 1, 2, \ldots, t)\), there may exist a cryptogram resources set \(RR_i \subset RR\) that can satisfy the functional requirements and performance indexes of the sub-goal \(rg_i\). Cryptogram resource matching aims at finding the best-suited cryptogram resource for each received sub-goal.

Similarly, to address this issue, an ant colony algorithm that includes just once cryptogram resource matching is introduced. Each matching process aims at finding the preponderant cryptogram resource for current sub-goal. Rules for cryptogram resource matching are depicted in equations (9) and (10).

![Figure 4. Scenario of cryptogram resource matching for sub-goals.](image)

\[
\phi_{\text{c}}(RS_i) = \sum_{i = 1}^{K} \phi(RS_i)
\]

where \(\delta \in [0, 1]\) and a random walk rule depicted in equation (9) is adopted when \(\theta \leq \delta\); otherwise, a determine matching policy shown in equation (10) is adopted. \(\tau_{\text{c}}(t)\) is the amount of pheromone for cryptogram resource \(c_i\) and \(\eta_{\text{c}}(t)\) is heuristic function indicating the expectation that \(c_i\) matches \(rg_i\). \(MT(rg_i)\) resources set that matching sub-goal \(rg_i\), \(c_i\) is set to be the \(i\)th component of the solution after each matching process. And \(\alpha_2 \geq 0\) is the heuristic factor reflecting the relative importance of accumulated pheromones \(\tau_{\text{c}}(t)\) in guiding resources matching process, \(\beta_2 \geq 1\) is the expectational heuristic factor reflecting the relative importance of heuristic factor \(\alpha_2\). All components in an epoch constitute a solution for RDMP.

**Candidate solution producing**. Candidate solution producing adopts the idea of concentrating group wisdom. Once all epochs are accomplished, to incorporate preponderent components of each solution, all generated solution are embedded into preponderent component extracting vectors using a linear projection \(\phi\). Then, all vectors are pooled by summation to produce the candidate solution for the RDMP. Assuming that \(RS_0^{\text{ca}}\) is the candidate optimal solution and \(RS_i\) denotes the \(i\)th solution, then there exists

\[
RS_0^{\text{ca}} = \sum_{i = 1}^{K} \phi(RS_i)
\]

where \(K\) denotes the scale of all epochs and there exists \(RE(RS_0^{\text{ca}}) > RE(RS_i)\).

**Solution verification and optimization**. In case that the candidate solution is a relative maximum one rather than the absolute one, an optimal solution verification and optimization process is indispensable to guarantee that the real optimal solution is obtained. To address this issue, a Lévy flight-based stochastic gradient algorithm is employed. The main idea is to introduce the advantages of frequent short-distance steps and accidental long-distance steps in Lévy flight into stochastic gradient policies.

Assume \(RS_0^{\text{ca}}\) to be the initial candidate solution and \(X_i^{\text{ca}}\) to be the \(i\)th updating of \(RS_0^{\text{ca}}\), the solution updating rule is formulated as follows.
These new solutions are generated based on random substituted by new solutions with certain probability. The hierarchical pheromone is defined as pheromone points out the reconfiguration policies of the Explorer and the Worker and for sharing wisdom. In HiCoACR, besides the proposed hierarchical coordination algorithm, a hierarchical pheromone is defined for reinforcing the positive feedback of the Explorer and the Worker for sub-goals occurs mainly in the sub-goal generation phase. As it is mentioned in section “Sub-goal generation,” 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.

**Pheromone updating for sub-goals.** Pheromone updating for sub-goals occurs mainly in the sub-goal generation phase. As it is mentioned in section “Sub-goal generation,” 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.

\[
phe_i(t + n) = \left(1 - \rho_{t}\right) \text{phee}_i(t) + \rho_{t} \left(\Delta \text{phee}_i + \gamma_1 \cdot \max_{k \in \text{allowed}} \text{phee}_k(t)\right)
\]

where \(\rho_t \in (0, 1)\) denotes the volatilization factor of protocol flow pheromone, whose value directly affects the direction controlling ability of the Explorer and convergence speed of the ant colony. \(\Delta \text{phee}_i\) depicts increment of sub-goal pheromone triggering state transition \(ij\). And, \(\gamma_1\) is the discount factor for computing the accumulated reward. Finally, after the model is trained, the sub-goals pheromone may form a hypergraph and could be transferred into sub-goal generation policies using equation (8).

**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, pheromone for all cryptogram resources constituting the optimal solution will be updated. The updating rule can be formulated as follows.

\[
phe^c_i(t + n) = (1 - \rho_{c}) \text{phec}_i(t) + \Delta \text{phec}_i
\]

where \(\rho_c \in (0, 1)\) refers to as the volatilization factor of resources pheromone, which directly affects the ratio of...
choosing resources as best-suitable resources that have been selected before; $\Delta \text{phe}_c$ depicts the increment of pheromone on cryptogram resources $c$. In cryptogram resource matching phase, $\Delta \text{phe}_c = W/\text{RE}(c)$. While in solution verification and optimization phase, $\Delta \text{phe}_c = W/\text{RE}(X_{\text{best}})$, where $X_{\text{best}}$ is local optimal solution in an epoch and $W$ is a constant variable. Once the hierarchical pheromone is trained, the hierarchical pheromone could be transferred into resources scheduling policies.

**Model details**

**RE evaluation**

RE evaluation is the fundamental of model training in reconfiguration decision-making, which evaluate the efficiency of both cryptogram resources and security protocol. For ease of RE evaluation, evaluation indexes used in this article include E-time, security level, resources consumption, energy consumption, and so on. E-time refers to as the time consumption that a protocol or cryptogram resource fulfills certain functionality. Security level indicates the security strength of a protocol or cryptogram component. Meanwhile, resources consumption represents the resources used by a protocol, such as memory and central processing units (CPUs). Thus, with these main indexes, a RE vector $\vec{r}e$ can be constructed according to Definition 4

$$\vec{r}e = \langle \text{time}, \text{sec}, \text{cpu}, \text{memory}, \text{energy} \rangle$$  \hspace{1cm} (22)

where $\text{time}$, $\text{sec}$, $\text{cpu}$, $\text{memory}$, and $\text{energy}$ refer to as E-time, security level, CPU consumption, memory consumption, and energy consumption, respectively. Analysis shows that these indexes can be classified into two categories, namely efficiency gain indexes and efficiency loss indexes. For instance, security level belongs to the efficiency gain indexes, where the higher the security level, the higher the RE is. While E-time and resources consumption belong to the efficiency loss indexes, where the lower the E-time and resources consumption, the higher the RE is. The goal of reconfiguration decision-making is to decrease efficiency loss and increase efficiency gain so as to find the solution with highest RE.

Without loss of generality, a linear model is adopted as efficiency evaluation model to compute the RE of a security protocol or a cryptogram resource

$$\text{RE}(x) = \vec{r}e(x) \odot (\vec{\omega})^T$$  \hspace{1cm} (23)

where $\text{RE}(x)$ denotes RE of $x$, $\vec{\omega}$ indicates the coefficient vector of efficiency indexes, and $\vec{r}e(x)$ refers to as efficiency vector of $x$. The efficiency evaluation model could be extended or modified to adapt to different practical application scenarios. Incentive function $\tanh(\text{RE}(x))$ could also be introduced to highlight efficiency difference of two solutions.

However, it is worth noting that considering the features of efficiency indexes, components of $\vec{r}e$ cannot be calculated by simply doing weighted summation while computing the RE of a security protocol. Instead, each component of $\vec{r}e$ is calculated with certain rules. In detail, when calculating value of time in $\vec{r}e$, it is equivalent to the shortest time problem in an activity on edge (AOE) network. When calculating value of $\text{sec}$, the Short board effect should be the calculation basis.

**Reconfiguration decision-making algorithm based on HiCoACR**

Once HiCoACR is trained, the reconfiguration decision-making can be done to address the RDMP with HiCoACR. The reconfiguration decision-making algorithm takes the cryptogram resources and trained hierarchical pheromone as input, and output the optimal resources set and corresponding protocol flow for the target protocol.

In details, the algorithm begins at the initial system state $s_0$. Then the Explorer generates the sub-goal according to sub-goal generation policies transferred from the sub-goal pheromone and the Worker schedules the best cryptogram resource for this sub-goal with scheduling policies derived from cryptogram resources pheromone. The Explorer and the Worker keep doing these steps until the system reaches end state $s_{\text{end}}$, where all components are scheduled and could constitute the whole protocol. The algorithm procedure is depicted in Algorithm 1.

Where $\text{PolicyCalculation}$ calculates policies with the hierarchical pheromone through equations (8) and (10). More details of function $\text{SubgoalGeneration}$ and function $\text{ResourceMatching}$ are depicted in sections “Sub-goal generation” and “Cryptogram resource matching for sub-goal,” respectively. With this algorithm, an optimal solution for RDMP could be obtained finally.

**Algorithm 1.** Reconfiguration decision-making algorithm.

Require: Cryptogram resources $RR$, hierarchical pheromone $\text{phe}$

Output: Optimal resource $RS$

1: $s_{\text{current}} \leftarrow s_0; RS \leftarrow \emptyset$.
2: $\pi_{\text{flow}} \leftarrow \text{PolicyCalculation}(\text{phe}'); \pi_{\text{res}} \leftarrow \text{PolicyCalculation}(\text{phe}')$
3: while $s_{\text{current}} \neq s_{\text{end}}$
4: $r_{\text{current}} \leftarrow \text{SubgoalGeneration}(s_{\text{current}}, \pi_{\text{flow}})$
5: $c \leftarrow \text{ResourceMatching}(r_{\text{current}}, RR, \pi_{\text{res}})$
6: $RS \leftarrow RS + c$
7: $s_{\text{current}} \leftarrow s_{\text{current}} + r_{\text{current}}$
8: end while
9: output ($RS$)
Table 1. Simplified reconfigurable cryptogram resources.

| Security service | Cryptogram algorithm and components |
|------------------|--------------------------------------|
| Confidentiality  | S-DES, DES, 3-DES, IDEA, Blowfish, CAST-128, CAST-256, A5/1, A5/2, RC4, RCS, RC6, ElGamal, RSA, AES, ECC |
| Integrity        | MD4, MDS, SHA-1, SHA-2, RIPEMD-160 |
| Authentication   | Hash, RSA, Schnorr, ElGamal, DSS, ECC |
| Anti-replay      | Linear Congruence, Normal distribution, Monte |
| Non-repudiation  | Hash, RSA, Schnorr, ElGamal, DSS, ECC |

Table 2. Security protocols.

| Categories     | Protocols                                    |
|----------------|----------------------------------------------|
| Authentication | Hsieh’s scheme,30 Jiang’s scheme,31 Priauth,32 Roy-Chowdhury’s scheme,33 and Chen’s scheme34 |
| Communication  | Mahmoud’s scheme,35 Khalil’s scheme,36 and RiSeG37 |
| Key agreement  | Tseng’s scheme38 and Amin’s scheme39 |

Experiments and discussion

Experiments setup

To explore the properties of HiCoACR, experiments are carried out to analyze its convergence, performance, and transferability. The subjects in our experiments include a reconfigurable cryptogram resources set and a security protocols set. Table 1 lists a simplified reconfigurable cryptogram resources set, which includes a mass of atomic and compound cryptogram resources in diverse forms such as cryptocards, cryptographic devices developed on FPGAs or reconfigurable processors, and so on. The developing platforms, performances, and other attributes are left out in Table 1. And for any one resource list in Table 1, there may be multiple physical devices with different performances corresponding to it.

Security protocols adopted to test HiCoACR are listed in Table 2, which include authentication protocols, communication protocols, and key agreement protocols. Convergence and performance experiments are mainly tested on authentication protocols, but communication protocols and key agreement protocols are utilized to demonstrate the transferability of HiCoACR. Moreover, details of these security protocols can be derived from related references.

The simulation mainly includes three procedures. The first phase is data preparing, where data such as functionality, interface, and performance of all cryptogram resources and the composition of target protocols have to be provided. In fact, we extract these data of cryptogram resources and target protocols and organize them in JSON format instead of operating physical devices which simplifies the training process. Usually, resource files with different amounts of cryptogram resources are prepared to simulate different scales of cryptogram resources.

The second phase is model adjusting, where the inputs and parameters of our algorithm are adjusted to simulate different scenarios. For instance, the scale of cryptogram resources is set to different values by inputting different resource files while testing their influences on convergence ratio. While examining the influence of pheromone volatility factors on convergence ratio, parameter $\rho_f$ in equation (19) and $\rho_r$ in equation (21) are set to different values between 0 and 1. Parameters in simulations are initialized as follows. Set $\alpha_1 = 2, \beta_1 = 2, \theta_0 = 0.2$ for equations (7) and (8). Set $\alpha_2 = 1, \beta_2 = 4, \delta = 0.3$ for equations (9) and (10). Set $\sigma = 1, \xi = 1.5, \gamma_0 = 1.5$ for equations (12)–(15). Set $\rho_f = 0.4, \rho_r = 0.35, \gamma_1 = 0.5, W = 10$ for equations (19)–(21). Let the numbers of ants in the Explorer and the Worker be 50.

The third phase is model training, where the convergence, performance, and model transferability are analyzed. During this phase, the proposed model is executed and a best reconfiguration solution for the target protocol is generated. Meanwhile, the policies for reconfiguration decision-making are stored, which can be transferred to reconfiguration decision-making for other target protocols.

Convergence analysis under different parameters

Convergence analysis mainly 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 model convergence. And the stopping condition for training is that the deviation of two adjacent solutions is less than 0.01.

Resources requirements of protocol and total scale of cryptogram resources. The total size of cryptogram resources carried on nodes and the cryptogram resources requirements of security protocol directly affect the performance of the reconfiguration decision model. To verify the influences, authentication protocols listed in Table 2 are chosen as target security protocol to represent different scales of cryptogram resources requirements, and we range the total scale of cryptogram resource from 0 to 500.

The influences of cryptogram resources requirements or scale of cryptogram resources on model convergence are illustrated in Figure 5. The $x$-axis represents the...
total scale of cryptogram resource and the \( y \)-axis denotes the convergence time. From Figure 5, it can be drawn that when cryptogram resource requirement is fixed, the larger the size of the total cryptogram resources, the slower the convergence rate. The reason is that the larger the size of the total cryptogram resources is, the bigger the searching space of RS is, which results in slower convergence speed. Thus, larger scale of cryptogram resource leads to lower model convergence rate.

Meanwhile, when the value of \( x \)-axis is set to be fixed, the convergence time of Hsieh’s scheme, Priauth, Jiang’s scheme, Roy-Chowdhury’s scheme, and Chen’s scheme increase in turn, where the resources requirement increases in turn as well. That is to say that when the size of the total cryptogram resources is fixed, the greater size the resources requirement of protocol is, the slower the convergence rate is. Similarly, the reason is that the times of loops enlarge with the increase of resources requirement, which leads to slower convergence as well. Even when concurrency strategies are adopted to optimize the algorithm performance, the tendency will remain the same for the reason that the next loop depends on the pheromone updated in last loop.

**Step type in solution verification and optimization phase.** In solution verification and optimization of HiCoACR, a Lévy flight–based stochastic gradient decline algorithm is adopted. The convergence rate of this stochastic gradient decline algorithm greatly influences the convergence rate of HiCoACR, where the Lévy step plays a vital role. To analyze the influence of different step types on model convergence, the fixed step and the random step are taken as benchmarks. And \( \text{Lévy}(\xi) = 1 \) for the fixed step and \( \text{Lévy}(\xi) = 2 \times \text{rand}(1) \) for the random step, where \( \text{rand} \) is a random number generator.

From Figure 6, we can find that the Lévy flight step outperforms the random step and fixed step in convergence. After deep analysis, we find that the Lévy flight–based stochastic gradient decline algorithm adopts the Lévy flight step which combines long step and short step, and steps will be adjusted according to the distance to the optimal solution, resulting in faster convergence speed. In addition, compared to fixed step, random step possesses faster convergence because randomness of step length increases the probability to find the best reconfiguration solution.

**Volatility factor of pheromone.** In HiCoACR, a hierarchical pheromone is defined to reinforce positive behaviors, which is adopted to imitate memory ability of ant colonies and is updated in each loop. Meanwhile, the pheromone volatility factor is used to simulate memory deterioration. To examine the effects of volatility factor, the total size of cryptogram resources is set to be 100 and choose Hsieh’s scheme as target security protocol. As both the flow pheromone and cryptogram resource pheromone have its own volatility factor, the influence of these two volatility factors is tested separately. Value of volatility factor for resources pheromone is set to be 0.08, 0.1, 0.12, 0.15, 0.2, 0.25, 0.3, and 0.35, while value of volatility factor for flow pheromone is set to be 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, and 0.4.

From Figure 7, we can find intuitively that larger volatility factor possesses less convergence time, which means the larger volatility factor, the faster the convergence speed. The reason is that the volatility factor denotes forgetting probability of history interactive
information among populations. When a pheromone gets to 0, the sub-goal generation action or matching action is forgotten, which influences the ability of solution searching. When the volatility factor gets larger, the pheromones of unvisited cryptogram resources or protocol flow which are relatively less than that of frequently visited ones will drop to 0 much faster and the probability that the visited ones are selected again will get larger, leading to the speed-up of convergence process. However, larger volatility factor leads to less diversity of reconfiguration policies due to the rapid decline of pheromone, which decreases the randomness of sub-goal generation and cryptogram resources matching.

In addition, when the volatility factor of protocol flow and cryptogram resources are set equal, the convergence time of the former is shorter than that of the latter. Because volatility factor of protocol flow not only affects the directions of reconfiguration, but also influences the resources pheromone. When the volatility factor of protocol pheromone gets larger, the probabilities that unvisited protocol flow will be selected is lower, which will lead to the pheromones of corresponding cryptogram resources in unvisited protocol flows lower than that of cryptogram resources in frequently visited protocol flows. This further increases the convergence speed. Thus, the influence of volatility factor of protocol flow pheromone is greater than that of resources pheromone. And either of these two volatility factors is set unsuitable, the collaboration relationship of the Explorer and the Worker will be broken.

Population collaboration. Population collaboration is one of the most vital mechanism to decrease the searching space of reconfiguration solutions in HiCoACR. To analyze its influence on model convergence, change the ant number in the Explorer and the Worker to adjust the degree of collaboration of the Explorer and the Worker. First, let the numbers of ants in the Worker to be 50, set numbers of ants in the Explorer to be 0, 10, 20, ..., 100. Then let the numbers of ants in the Explorer to be 50, set numbers of ants in the Worker to be 0, 10, 20, ..., 100. Meanwhile, set the total size of cryptogram resources to be 100 and choose Hsieh’s scheme as target security protocol.

From Figure 8, when the number of ants in the Explorer or in the Worker drops to 0, the convergence time gets longer. Meanwhile, when the number of ants reaches a certain value, the convergence time is tending toward stability. The reason is that when the ant number in the Explorer or the Worker equals 0, there exists only one population and interpopulation collaboration disappears, which results in that HiCoACR degenerates to Cuckoo search algorithm when number of ants in the Explorer is 0 or degenerates to ant-Q algorithm when number of ants in the Worker equals 0. And at that moment, the convergence time approximately reaches the ceiling. In addition, according to the experiments results, number of ants in the Explorer and the Worker is more suitable to be set to about 50 in terms of the given protocol samples.

Performance analysis
Performance analysis of HiCoACR model aims to compare model performance with certain benchmark algorithms. Set size of whole cryptogram resources to be 100, take ant colony algorithm, Cuckoo search algorithm, and reinforcement learning as benchmark algorithms and set authentication protocol in Table 2 as target security protocol.
From Table 3, it could be seen that HiCoACR model possesses less time cost and much higher accuracy, which indicates that HiCoACR outperforms given benchmark algorithms in accuracy and efficiency. According to above designation, HiCoACR model divides the reconfiguration solution searching process into two sub-process including controlling directions of protocol flow and cryptogram resources matching. Sub-goal generation phase implemented by the Explorer screens the reconfiguration solutions with higher RE and controls the directions of sub-process according to protocol flows using ant colony algorithm with Q-learning, which greatly narrows the solution space. Meanwhile, cryptogram resources scheduling phase executed by the Worker is consist of sub-goal matching, candidate solution producing, solution verification and optimization and responsible for reconfiguration solution producing and optimization. Compared to above benchmark algorithms, HiCoACR adopts the advantages of population collaboration and reinforcement learning. When the number of ants in the Explorer is 0, HiCoACR degrades into Cuckoo search algorithm. When the number of ants in the Worker is 0, HiCoACR degenerates into ant colony algorithm with Q-learning. Thus, HiCoACR outperforms the given benchmarks in performance.

### Model transferability analysis

Model transferability analysis aims at verifying the transferability of trained model on different categories of security protocols. Model transferability mainly reflects in the acceleration of convergence under the condition of ensuring accuracy in reconfiguration decision-making. Thus, average speed-up ratio is adopted to assess the model transferability. And speed-up ratio is defined as the ratio of convergence time of baseline and convergence time of given model, namely

\[ sr = \frac{t_b}{t_m} \quad (24) \]

where \( sr \) is speed-up ratio of a certain protocol under mode \( m \), \( t_b \) denotes convergence time of baseline, and \( t_m \) refers to as convergence time of given mode \( m \). Meanwhile, average speed-up ratio is weighted mean of all speed-up ratio for each protocol in the test sample.

Three modes are taken into consideration in this experiment including without pre-training, pre-trained with same category, and pre-trained with different categories. For mode of without pre-training, no reconfiguration policy is trained ahead of time and the pheromone of both protocol flow and cryptogram resources is initialized with 0. For mode of pre-trained with same category, reconfiguration policy has been generated according to security protocols the same category with the test samples. For mode of pre-trained with different categories, reconfiguration policy has already been generated with security protocols whose category is different with the test samples.

Set size of total cryptogram resources to be 100, take the first three authentication protocols in Table 2 as training samples and adopt rest two authentication protocols, key agreement protocols and communication protocols as test samples. Meanwhile, ant colony algorithm, Cuckoo search algorithm, and reinforcement learning are selected as benchmark algorithms as well. Set the time cost of the mode without pre-training as baseline, we can see from Table 4 that average speed-up ratio of pre-trained with protocols in same category is larger than that of pre-trained with protocols in different categories. The reason is that the mode pre-trained with same category provides more useful information for reconfiguration than the mode pre-trained with different categories. Meanwhile, we can find that average speed-up ratio of HiCoACR is higher than that of given benchmarks regardless of application modes. The reason is that both protocol pheromone and resources pheromone can provide reconfiguration policy during decision-making, which results in less time cost in HiCoACR than other benchmark algorithms. Thus, HiCoACR outperforms the given benchmark in model transferability.

In conclusion, the hierarchically collaborative ant colony-based reconfiguration proposed in this article...
enjoys higher convergence speed, better performance, and model transferability. However, as the ant colony algorithm has the shortcomings of long convergence time and is easy to get into the local optimal solution, our model may exist the same problem in some degree, although an adaptive pseudorandom ratio selection rule and a parallel computing mechanism are adopted in the simulation to alleviate the two shortcomings.

**Conclusion**

Reconfiguration decision-making is one of the key processes to reconfigurable security protocol design, which determines the performance of security protocol. This article combines the idea of hierarchical reinforcement learning and collaborative behavior of biological populations and proposes a reconfiguration decision-making model named HiCoACR based on hierarchical coordinated ant colony algorithm. It takes the advantages of reinforcement learning and population collaboration and handles the RDMP commendably. The method of this article may also be applied to decision-making problem such as military task programming, route planning. In addition, in order to eliminate the problems of long convergence time and falling into local optimal solution, we will keep on improving our model in the future study.

**Acknowledgements**

The authors would like to thank the Associate Editor and all the anonymous reviewers for their insightful comments and constructive suggestions.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (grant nos 61773399 and 61403400) and China Postdoctoral Science Foundation special funded project (grant no. 2017T100792).

**ORCID iD**

Zhuo Yi https://orcid.org/0000-0002-6065-3244

**References**

1. Bossuet L, Fischer V, Gaspar L, et al. Disposable configuration of remotely reconfigurable systems. *Microprocess Microsyst* 2015; 39: 382–392.
2. Yoon S, Park H and Yoo HS. Security issues on smarthome in IoT environment. *Lect Notes Electr En* 2015; 330: 691–696.
3. Vezhnevets AS, Osindero S, Schaul T, et al. FeUdal networks for hierarchical reinforcement learning. *PMLR* 2017; 70: 3540–3549.
4. Kulkarni TD, Narasimhan KR, Saeedi A, et al. Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation. In: *Proceedings of the annual conference on neural information processing systems (NIPS)*, Barcelona, Spain, 5–10 December 2016, pp.3675–3683. Neural Information Processing Systems Foundation.
5. Niu Q, Zhou T and Wang L. A hybrid particle swarm optimization for parallel machine total tardiness scheduling. *Int J Adv Manuf Tech* 2010; 49(5–8): 723–739.
6. Reynolds AM. Cooperative random Lévy flight searches and the flight patterns of honeybees. *Phys Lett A* 2006; 354(5): 384–388.
7. Wu AS, Yu H, Jin S, et al. An incremental genetic algorithm approach to multiprocessor scheduling. *IEEE T Parall Distr* 2004; 15(9): 824–834.
8. Xu Y, Li K, Hu J, et al. A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Inform Sci* 2014; 270(6): 255–287.
9. Bryan AN and James CB. A genetic algorithm methodology for complex scheduling problems. *Nav Res Log* 2015; 46(2): 199–211.
10. Abraham A, Liu H, Zhang W, et al. Scheduling jobs on computational grids using fuzzy particle swarm algorithm. *Future Gener Comp Sy* 2010; 26(8): 1336–1343.
11. Kong X, Sun J and Xu W. Particle swarm algorithm for tasks scheduling in distributed heterogeneous system. In: *Proceedings of the sixth international conference on intelligent systems design and applications*, Jinan, China, 16–18 October 2006, pp.690–695. New York: IEEE.
12. Fan PO and Winley GK. A Heuristic search algorithm for flow-shop scheduling. *Informatica* 2008; 32(4): 453–464.
13. Wen Y, Xu H and Yang J. A heuristic-based hybrid genetic-variable neighborhood search algorithm for task scheduling in heterogeneous multiprocessor system. *Inform Sci* 2011; 181(3): 567–581.
14. Mencia C, Sierra MR and Varela R. Depth-first heuristic search for the job shop scheduling problem. *Ann Oper Res* 2013; 206(1): 265–296.
15. Hu J, Gu J, Sun G, et al. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In: Proceedings of the international symposium on parallel architectures, algorithms and programming. Dalian, China, 18–20 December 2011, pp.89–96. New York: IEEE.

16. Das S, Viswanathan H and Rittenhouse G. Dynamic load balancing through coordinated scheduling in packet data systems. In: Proceedings of the twenty-second annual joint conference of the IEEE computer and communications societies, San Francisco, CA, 30 March–3 April 2003, pp.786–796. New York: IEEE.

17. Yan H and Luh PB. A fuzzy optimization-based method for integrated power system scheduling and inter-utility power transaction with uncertainties. IEEE T Power Syst 2002; 12(2): 756–763.

18. Fagan D, Fenton M, Lynch D, et al. Deep learning through evolution: a hybrid approach to scheduling in a dynamic environment. In: Proceedings of the international joint conference on neural networks, Anchorage, AK, 14–19 May 2017, pp.775–782. New York: IEEE.

19. Chen W, Xu Y and Wu X. Deep reinforcement learning for multi-resource multi-machine job scheduling. In: The 25th IEEE international conference on network protocols, Toronto, Canada, 10–13 October 2017, pp.1–2. New York: IEEE.

20. Chen H, Clark JA and Jacob JL. A search-based approach to the automated design of security protocols. Report, York Computer Science Technical Report Series (“Yellow Reports”), Department of Computer Science, University of York, UK, April 2004.

21. Yang E, Erdogan AT, Arslan T, et al. Multi-objective evolutionary optimizations of a space-based reconfigurable sensor network under hard constraints. In: Proceedings of the ECSIS symposium on bio-inspired, learning, and intelligent systems for security, Edinburgh, 9–10 August 2007, pp.72–75. New York: IEEE.

22. Alrifai M and Risse T. Combining global optimization with local selection for efficient QoS-aware service composition. In: Proceedings of the international conference on World Wide Web, Madrid, 20–24 April 2009, pp.881–890. New York: ACM.

23. Chifu VR, Pop CB, Salomie I, et al. Optimising the semantic web service composition process using bio-inspired methods. Int J Bio-Inspir Com 2013; 5(4): 226–238.

24. Li J, Zheng XL, Chen ST, et al. An efficient and reliable approach for quality-of-service-aware service composition. Inform Sci 2014; 269(4): 238–254.

25. Zhao X, Shen L, Peng X, et al. Toward SLA-constrained service composition: an approach based on a fuzzy linguistic preference model and an evolutionary algorithm. Informa Sci 2015; 316: 370–396.

26. Ida M, Russek EM, Jin CH, et al. The successor representation in human reinforcement learning. Nat Hum Behav 2017; 1: 680–692.

27. Roijers DM, Vamplew P, Whiteson S, et al. A survey of multi-objective sequential decision-making. J Artif Intell Res 2013; 48(1): 67–113.

28. Datta A, Derek A, Mitchell JC, et al. Secure protocol composition. In: Proceedings of the ACM workshop on formal methods in security engineering, Washington, DC, 30 October 2003, pp.11–23. New York: ACM.

29. Niesen M, Plotkin G and Winkelg. Configuration structures, event structures and Petri nets. Theor Comput Sci 2009; 410(41): 411–4159.

30. Hsieh WB and Leu JS. Anonymous authentication protocol based on elliptic curve Diffie–Hellman for wireless access networks. Wirel Commun Mob Com 2014; 14(10): 995–1006.

31. Jiang Q, Ma J, Li G, et al. An efficient ticket based authentication protocol with unlinkability for wireless access networks. Wireless Pers Commun 2014; 77(2): 1489–1506.

32. He D, Bu J, Chan S, et al. Privacy-preserving universal authentication protocol for wireless communications. IEEE T Wirel Commun 2011; 10(2): 431–436.

33. Roy CA and Baras JS. A lightweight certificate-based source authentication protocol for group communications in hybrid wireless/satellite networks. In: Proceedings of the IEEE global telecommunications conference, New Orleans, LO, 30 November–4 December 2008, pp.1–6. New York: IEEE.

34. Chen C, He D, Chan S, et al. Lightweight and provably secure user authentication with anonymity for the global mobility network. Int J Commun Syst 2011; 24(3): 347–362.

35. Mahmoud EA, Taha S, Misic J, et al. Lightweight privacy-preserving and secure communication protocol for hybrid ad hoc wireless networks. IEEE T Parall Distr 2014; 25(8): 2077–2090.

36. Khalil I, Bagchi S and Shroff N. Analysis and evaluation of SECOS, a protocol for energy efficient and secure communication in sensor networks. Ad Hoc Netw 2007; 5(3): 360–391.

37. Cheikhrouhou O, Koubaa A, Dini G, et al. RiSeG: a ring based secure group communication protocol for resource-constrained wireless sensor networks. Pers Ubiquit Comput 2011; 15(8): 783–797.

38. Tseng HR. A secure and privacy-preserving communication protocol for V2G networks. In: Proceedings of the IEEE wireless communications and networking conference, 1–4 April 2012, pp.2706–2711. New York: IEEE.

39. Amin R and Biswas GP. A novel user authentication and key agreement protocol for accessing multi-medical server usable in TMIS. J Med Syst 2015; 39(3): 33.

40. Colorni A, Dorigo M and Maniezzo V. Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life (ECAL), Paris, 1 January 1991, pp.134–142. Amsterdam: Elsevier.

41. Yang XS and Deb S. Cuckoo search via Lévy flights. In: Proceedings of the world congress on nature and biologically inspired computing (NaBIC), Coimbatore, India, 9–11 December 2009, pp.210–214. New York: IEEE.

42. Thomas A and Marcus SI. Reinforcement learning for MDPs using temporal difference schemes. In: Proceedings of the IEEE conference on decision and control (IEEE CDC), San Diego, CA, 12 December 1997, pp.577–583. New York: IEEE.