Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

USING OF COLLEAGUE LEARNING MOBILE AGENTS FOR PROTECTING THE CONFIDENTIALITY OF MOBILE AGENTS IN A MULTI-AGENT ENVIRONMENT

S. M. M. Ebrahimi

Published online: 15 May 2016

ABSTRACT

With the growing use of the internet network, using tools that can help user to transact with the network is inevitable. One of the technologies that has been taken into consideration recently and has been used is the mobile agents technology mobile agent is software that can take an action independently and autonomously as an assistant to a person or to an organization. Mobile agents are used for searching information, data recovery, filtering, detecting obtrusive in the networks, games and entertainment, etc. one of the discussed issues related to the mobile agent is their security issue. For effective use and keeping mobile agents security various security issues must be taken into consideration. One of these issues is to keep the mobile agents confidentiality. In this paper, at first, by reviewing the existing methods, advantages and disadvantages will be investigated. In this paper, by taking advantages of the mobile agents features such as cooperation, learning, mobility and duplication, one effective method is presented to keep the mobile agents confidentiality.

Keywords: Mobile agents, security; confidentiality; learning; cooperation; mobility; duplication.

Author Correspondence, e-mail: smm_ebrahimi@yahoo.com

doi: http://dx.doi.org/10.4314/jfas.v8i3s.240

1. INTRODUCTION

Mobile agents are computer programs or software's which are able to move in a heterogeneous environment such as internet and help user to take actions. There are many issues concerning mobile agents technology. The most important of which is (the) security issue. Regarding the fact that the mobile agents are composed only of software, so their security issue is a difficult one. In this paper, a brief description of software agents will be represented, then, mobile agents, their characteristics and functions will be demonstrated. The main focus of this paper is the security of mobile agents in particular protecting their confidentiality. Then, the existing methods for the purpose of protecting the mobile agents will be represented. At the end, an effective method for the purpose of protecting the security of mobile agents by the use of mobile agent characteristics, such as, cooperation, learning, mobility and presentation duplication, will be simulated and evaluated. The results of the simulation show the protection of the mobile agents security in multi-agent environments by the use of this method.

2. AGENTS

Agents are software entities which are able to take an action in a heterogeneous environment such as internet on behalf of a person or an organization. These actions cover the whole range of things from E-commerce to E-learning. In association with agents there are various definitions and very definition refers to (reflect) the expectations that the exponent has of the word and concept of "agent". Term such as smart agents and multi-agent systems are widely used. These terms are used for definition of entities and systems with high operating range, from simple systems to complex systems. This diversity in application causes the consideration of various capabilities and viewpoints for agents. This comparison is valid in conjunction with software agents. A software agent can be a simple entity that is able to give a simple and a pre-determined answer only to environment changes, or it can be a complex entity which has various aims and uses various designs and plans to achieve its goals. Due to the widespread range of using the agents, various definitions are mentioned for each agent. In one definition, agent is "a software system which is located in environment and it is able to do flexible & independent actions in environment in order to achieve the intended purposes for it". Thus, an agent can be considered as an entity having a certain purpose and in order to achieve that purpose, makes a decision according to different situations and can blurt different behaviors (Milgrom, 2001). The characteristics of a software agent can be divided into three categories including: features relating to the identity and the nature of an agent, features which must be considered for each agent at runtime and features which are necessary for an agent to have to be balanced with other agents. A summary of agent features can be observed in table 1.

Table 1. Agent features (Barforoush, 2005)

Features	A brief summary of features
Autonomy	Addressing agent according to the defined purpose without calling
Situation-oriented	Dependent on the environment and operating conditions
Reactive	Understanding of the environment and responding to its changes
Action-oriented	Representation of defined targeted behaviors
Learning	Change of behaviors based on previous experiences
Veracity	Lack of false information transfer
Persistency	Having pre-determined purposes and internet processes
Social	Cooperating with other agents in multi-agent systems
Goal-oriented	Realization of the system target rather than achieving it without calling
Reasoning	Reasoning ability in selecting the action
Adaptability	The possibility of corresponding the action of an agent with overall system goals
Mobility	Ability to transfer from one environment to another
Benevolence	Compromise in profits related to cooperating agents
Delegacy	Taking actions in multi-agent systems
Competency	Assessment of the carried-out activity to achieve the goal and division of duties whenever necessary
Amenability	Assessment of achieving the system goal and continuing taking action to achieve the goal
Discourse	Reasoning ability in the selection of dependent-on-the-environment action
Rationality	Taking reasonable actions (not necessary correct) to achieve the goal

System based on agents and their features

There might be a situation in which an agent can take action lonely and without connection with other entities and provide acceptable results, but in order to increase efficiency and given the importance of interaction in the environment, the situation in which agent is considered separately is very limited. A group of agents usually interact with each other in an environment. These systems, in which a number of agents are considered as problem-solving entities, are called, multi-agent systems. An abstract view façade of an agent-based system is represented in figure 1. In such a system, an agent can act separately, or interact with other agents as an organizational unit to achieve a common goal. The interaction between agents can be separate in order to achieve individual goal or an organization of agents interact with another in order to achieve the goal. These agents affect the environment in which they are located and they are also affected by the environment either personally or organizationally. Affecting the environment by the agents maybe overlapped and various agents may affect the same environmental agent.

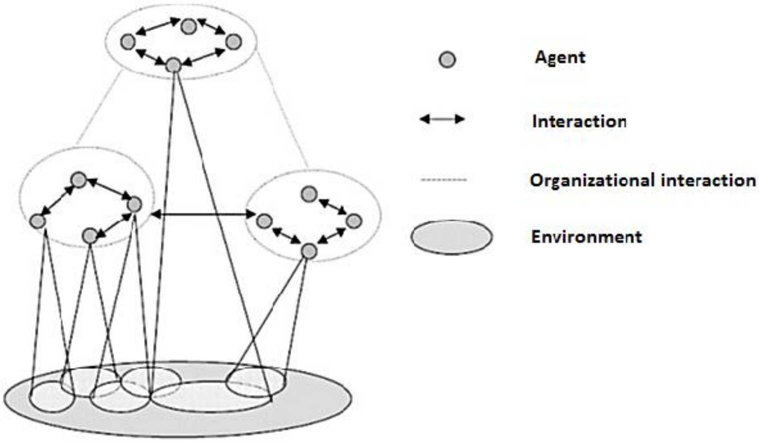

Fig.1. An abstract view façade of an agent-based system (Barforoush, 2005)

Using the multi-agent methods in saving problem enable as to use the interaction between problem-solving parts. In these systems, because of the fact that separate entities take part in problem solving, implementing one part of in "parallel" is possible. With the "redundancy" which exists in distribution systems, the possibility of error detection and its restoration increases, and the possibility of reusing parts of system and its existing knowledge is provided. In multi-agent systems, on infrastructure is provided for the system which is possible for the connection among agents by interaction protocols in the environment.

The existing agents are in an independent, autonomous multi-agent systems and they can interact with each other. These agents can follow their individual goals, or cooperate with each other to achieve common goals which are defined in the whole system. The interaction between is based on these features. So, the prominent features of a multi-agent system could be identified based on the system environment and the way the agents interact with each other.

Agents classification

In this part, we divide the existing agents into different classes. There are several dimensions for classification of the existing software agents (Kostakos, 2001).

At first, the agents might be classified according to their mobility or they could be classified according to their ability to move within some networks. This classification is the result of their mobility or being static.

Secondly, they might be classified according to their participation or interaction. The agents participation and cooperation come from this thought that agents have the ability to participate and cooperate. Agents have a written and symbolic model and use this symbolic model to obtain the cooperation with other agents.

Thirdly, agents might be classified according to the primary behaviors that they exhibit, including: dependence, learning and cooperation. Dependence refers to the agents that can take an action with no need to be guided by human being and this subject could be very important in some cases. Here, agents have individual internal modes and purposes and the might act according to user's intended method. Cooperating, with other agents is highly important. The cooperation of agents with each other, comes from this fact that several agents cooperate with each other to take an action. In order to cooperate with each other, agents are needed to be able to cooperate. For example, the ability to interact with other agents or probably human beings is some communicative languages. However it can be said that agents can coordinate their actions without cooperation. Finally, in order to these agent systems be smart, they should learn from their interaction with the environment. In addition, the learning efficiency must increase during the time. These three features are used to obtain four types of agents which are included in the classification including; collaborative agents, learning agents, interface agents and smart agents.

Fourthly, agents might be classified according to their functions (especially when the function is the main feature of the agent), such as, informative agents of the World Wide Web. These types of agents are usually used for each engines. Basically, they help the management of a huge amount of information in wide networks such as internet. These class of agents is called internet or informative agents, this informative agents can also be static, mobile or deliberative.

Fifth, they might be one kind of hybrid agents which have combined the features of two or more agents in one simple agent. Basically, agents are located in a multi-dimensional environment. For more clarification, this multi-dimensional space can be classified into a simple list. So, we can have the following list which determine seven type of agents.

- Collaborative Agents
- Interface Agents
- Mobile Agents
- Information/Internet Agents
- Reactive Agents
- Hybrid Agents
- Smart Agents

Figure 2 shows this classification.

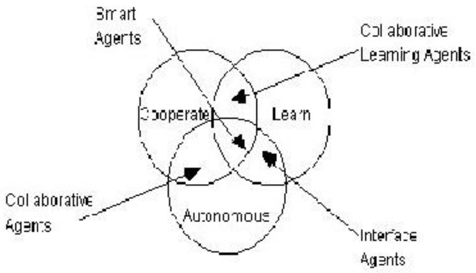

Fig.2. Classification of agents based on their features (Kostakos, 2001)

Mobile agent

Software agent technology results in two areas: 1) Artificial Intelligence, 2) Distributed Computing.

The purpose of the artificial intelligence is that it uses intelligent computing entities to do task that humans do. A software agent is a computer program whose purpose is to help user to do some task or a series of tasks. It is assumed that the agent is intelligent enough to do its own works. In other words, user acts as its representative, interacts with its user and it has enough reason to go to the place it wants, it can be linked with environment, learn from the environment and can demonstrate appropriate behavior.

A complex task can be done through the cooperation of several distributed agents on separate computers which are connected internally and every one of which share their own power and experience with others such a thing is more efficient than doing the task with a single computer. A mobile agent is a software entity which does a task independently and autonomously on behalf of a person or an organization. In environments in which mobility is considered to be a basic issue, mobile agents are autonomous software entities which are able to move. They move within the network whenever they are needed. They are the purposes and sent information to users automatically. Movement features allow agents to move among a variety of hosts platforms. In the host environment, agent is allowed to duplicate itself, produce new agents and destroy some other agents, communicate with other agents, find other agents or move toward other hosts. Moreover, the mobile agent can continue to run after leaving the source engine and going to another host.

The main feature which is specific to mobile agent is that the mobile agent is a package including code and some extra data which can move between receiver service and host machine. During the implementation of mobile code, the agents might produce some results, go to other host and then come back to the original location. The mobile agents are not limited to the system from which the performance is started. They can move freely among several network hosts. The mobile agent can be considered as a class including both code and state. State is a feature for figuring out what needs to be done. Agent is used after it stops its performance. Code is a class needed for the agent performance. That can move independently from one agent platform to another one to interact with other agents. It performs a specific task by user. Agent tasks its identity from the user, than it can be implemented without user's help.

Mobile agent security

When data must be transmitted, security issues such as encoding and decoding are required in the case of mobile agents, another security policy must be implemented according to their features and different security concepts must be considered. Mobile agent technology given that it is a program or a software, that can suspend its performance on a computer, transmit itself into another host and resume its performance on another computer, it gives calculation a new perspective. Given that day-to-day use of mobile agent grows, therefore security threats will increase.

To examine the issue of mobile agents security, two concepts should be investigated agent and host.

So, four types of security threats associated with mobile agents including (Bierman, Cloete, 2007) (Jansen, 2001), (Jansen, 2011):

- ➢ Threats which take place from host to agent.
- ➢ Threats which take place from agent to agent.
- ➢ Threats which take place from agent to host.
- ➢ Threats which take place from other entities to host.

Figure 3 shows types of attacks.

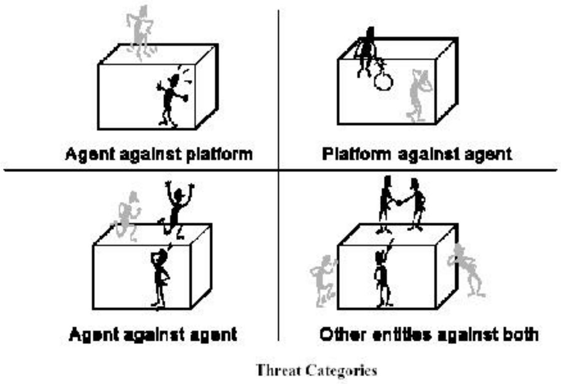

Fig.3. Security threats concerning mobile agents

Protecting mobile agents

The full use of mobile agent technology in unreliable network environments such as internet requires considering some security cases. There exists a large series of cases and sub-cases in association with mobile agents security which makes it difficult to distinguish among different types of issues and problems, so despite their existence, diagnosing proper solutions will be difficult. The intended issue, concerning agent security is that, whether a software agent can itself against one or several malicious hosts?

Compared with the hosts, mobile agents are weaker in this sense. They only consist of software. Therefore, they don't have hardware to establish their security. It must be stated explicitly that, protecting the mobile agents against malicious hosts is a difficult issue. A classification of security risks that can be confronted in mobile agent environments is as follows:

- Malicious hosts
- Malicious agents
- Malicious network entities

A malicious host can provide several types of security attacks on mobile agents and can avert its anticipated performance toward malicious purpose or changing data or other information to profit from agents activities. On the other side, agents can have malicious purposes against the hosts or other agents. For example, a virus or a Troy Hors can place itself as a mobile agent, and then attack the host sources. An agent can also be manipulated by other agents, so they cannot do their tasks.

Third, the network entities that the host gives out, can attack other agents in transfer or attack other mobile agent systems and steel their secret information or damage their integrity. Here, their exist a range of security issues which is associated with each of these categories.

The purpose of this paper is to consider and focus on malicious host category and to analyze different security threats which are imposed on agents by malicious hosts. Here, a category of these attacks is provided and also solutions, which are implemented to identify the issues, are represented.

3. METHODS TO PROTECT MOBILE AGENTS

In this part of the focus is on criteria which are implemented or it is though that it reduces mobile agents vulnerability against malicious hosts. On the issue of mobile agent security both prevention mechanisms and detection mechanisms are used. Prevention mechanism is to protect mobile agents, while detection is usually performed to detect possible security violation. Following are useful methods in detection and those which are dedicated to prevention.

Four types of methods which are used for mobile agent security are as follows:

- Trust

  Providing a safe environment where in mobile agents are located freely and audaciously without fear of malicious hosts, can often reduce these threat classes which were explained. In implementing methods based on the notion of trust, a security policy must be created and must be used by the host.

- Recording and Tracking

  These methods used trajectory information of mobile agents which are made by keeping movement history. In order to implement this method, colleague agents or colleague hosts can be used.

- Cryptography

  Techniques under this method use encoding and decoding algorithm. Public and private keys digital signatures, digital time stamps, Hash functions are different types of this perspective.

- Time Technique

  This method is based on limitations or mobile agent life time.

Suggested methods

In this research a method is suggested to keep the mobile agents confidentiality, which is based on cooperation, learning and duplication of the agents. In order to maintain the mobile agent confidentiality, cryptography methods are used mainly. In this method, cryptography is used as a basis for keeping confidentiality. In addition, agents features are used, so far, less attention has been paid to this topic. The algorithm of the suggested method is as follows:

1. Each agent consults with other existing agents about safe or unsafe host (the appropriate number not all agents) before moving toward the intended host.
2. If the result of voting is positive (more than 30%), the agent moves toward the intended host, otherwise ignores moving toward the intended host.
3. If the agent decides to go;
    a) The agent makes several copies of itself on the host (to a number that makes it difficult to distinguish the main agent from the fake agents or reduces its possibility).
    b) The main agent initializes them with unrealistic data.
    c) All of them will be implemented on the host (because apart from the main agent which is hidden, other agents are implemented, so the cast of hearing increases from the main agent for the host and more time is spent to distinguish the main data, that this spending of time is not affordable for the host, so the possibility of hearing or leaking the data deceases and the possibility of keeping confidentiality increases).
    d) While performing, agents ask the host questions about other agents information which have been previously implemented on this host, and by alluring the host, they try to get it. (for example, if it provide information of the previous agent, entices it with more purchasing offer, if the host is not safe in terms of confidentiality, discloses this information and the agent figures out that this host is not safe in terms of confidentiality).
    e) After implementing, the main agent destroys other subordinate agents before leaving the host.
    f) During a meeting with other agents, the intended agent provides them with its own information about the intended host safe or unsafe.
    g) With this cooperation, the agents will learn which host is safe.

h) If the unsafe host does not reform its behavior, after a while it will be placed in the agents black list and actually, it will be removed from the environment executive cycle.

i) Considering the fact that the agent is implemented on the host, roughly safe in terms of safety (according to its duplication), so it provide the opportunity for the unsafe host to reform itself and to return again to the environment executive cycle.

Tip: If the d phase is done at early stage, the algorithm will be every hard and the possibility of the agent implementation, on many agents will be lost and even if the unsafe host reforms itself, it might not be able to return to the environment executive cycle.

Step of doing the task are shown in UML diagrams in figures 4, 5.

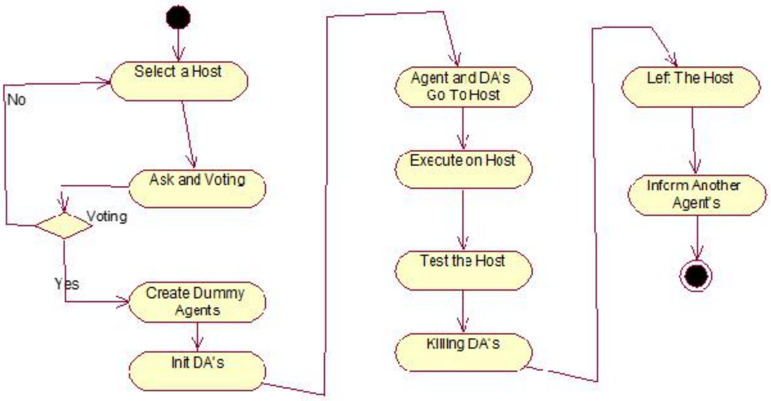

Fig.4. Activity diagram

In the state diagram which has been shown in figure 5, various modes of multi-agent systems are shown in different states. First, a host is chosen to go and the system is placed in the selection state. Next, the system starts the act of voting. Following that, the system create some fake agents and initializes them. Then, the system is placed in the run mode, and the main agent and other fake agent are implemented on the host. In the following, the system is placed on the state of the host test and it will be stated by the agents. The next state of the system is destroying the fake agents by the main agent because they are not needed. The next state is leaving the host by the main agents and finally noticing the agents will be implemented.

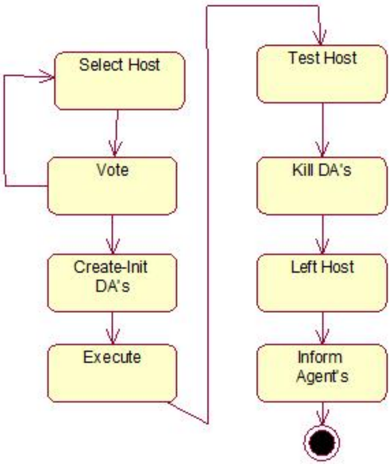

Fig.5. State diagram related to the suggested algorithm

In figure 6, which is our Usecase diagram and algorithm state, it has been shown that, at first the main agent consults with other agents and asks them questions about the host safe or unsafe, next, it receives the answer which is determined by Asking/Voting phrases. The produces fake agents and initializes them with Create/Init phrase, the fake agents are transformed to the host by acting Travel, after that they are implemented on the host by acting Execute and test the host by acting Test. Finally, by the Kill command of the main agent, deletes the fake agents before leaving the host and informs other agents with Inform phrase about the final answer associated with the host safe or unsafe.

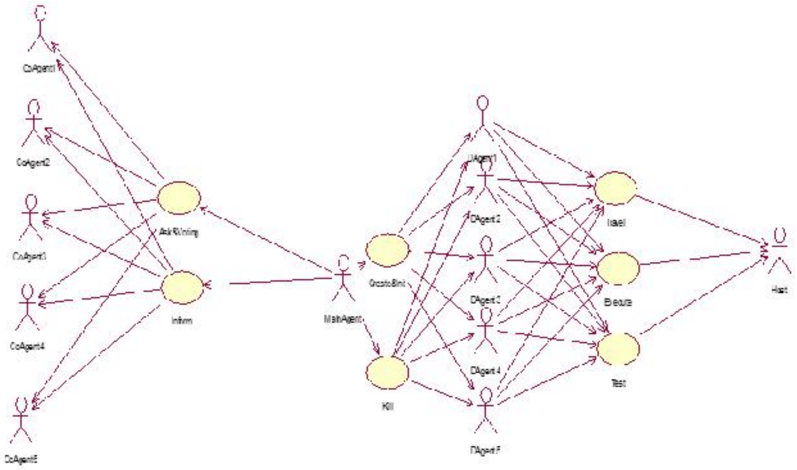

Fig.6. Usecase diagram

Figure 7, is the collaboration diagram which phases of mobile agents doing activities with other mobile agents, fake agents and also the host. In this diagram, as an example, five mobile agents are considered that shows question and answer phase from the mobile agents from number 1 to number 5. Phases 6 to 10 are phases of making fake agents and also initializing them. Phase 11 is the mobile agent decision related to going to the host. Phases 12 to 16 are phases of transmitting fake agents on the host. Phases 17 to 22 shows agent implementing on the host. In phases 23 to 28 doing experiments and interactions with the host take place. In phases 29 to 33, the fake agents are deleted by the main agents and finally. In phases 34 to 38, the main agent declares the host security or its malice.

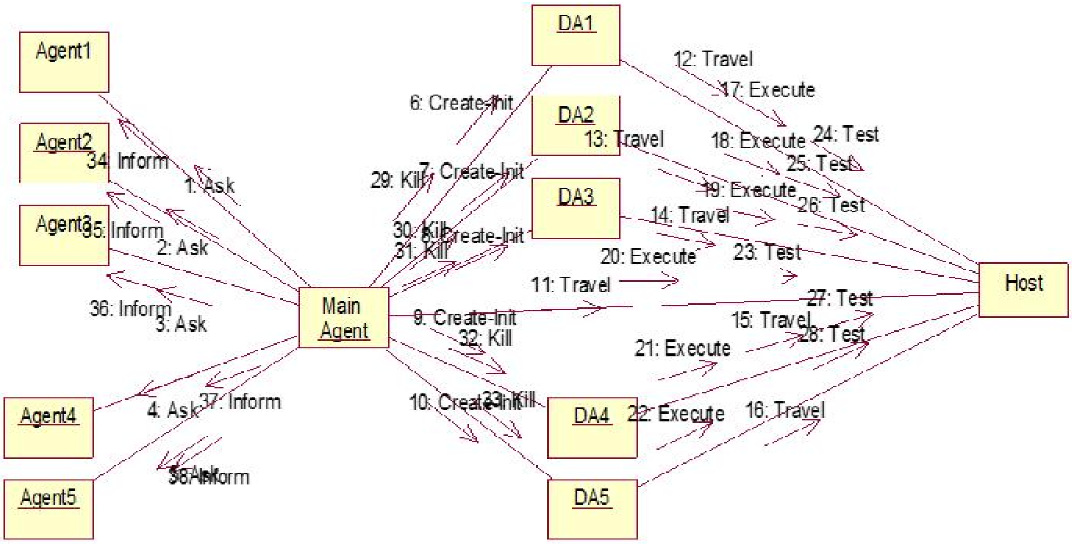

Fig.7. Collaboration diagram

4. SIMULATION AND EVALUATION

In this method, the mobile agent is located in a multi-agent and multi-host environment. The intended agent at first consults with several agents before moving toward the intended host (either accidently or those which have previously meet this host) and starts voting. If the voting result is based on going toward the host, the agent will move toward that host. Before implementing on the host (it is supposed that the host can reach only to the agent data that implements on it), the agent provides several copies of itself.

If we consider 'n' as the copy number, according to (Cormen, 2009), the number of agents that must be investigated on average to elicit the real agent (assuming that the host have information about the agent and the agent is identifiable for the host), is equal to $T(n) = \frac{n}{2}$, so the possibility of finding the intended agent if distribution is normal and the possibility of finding the intended agent among the agents which follows this distribution is equal to $P(FA) = \frac{1}{\frac{n}{2}} = \frac{2}{n}$.

Therefore in this state, given that $\lim_{n\to\infty}\left(\frac{2}{n}\right)$ tends toward zero, so the more is the number of copied agent, the less if the possibility of finding the intended agent, but the number should not be to the extend that makes it possible to use this method. Because cryptography is used in this method, so if the data and the agent code are encrypted by one method (here we consider RSA as a basic method), we consider the decoding time for each agent equal to 't', so the decoding time, according to the average number of agents which must be investigated on average, will be $\frac{n}{2}$ equal to $\frac{nt}{2}$. Now, if we consider this cryptography method as RSA method as an example, (Malekian, 2013), in this case, according to table 2. The time of finding the intended agent TF(A) can be considered like table 3. The following results are achieved by using a computer with processor specification of 2.4GHz and 2GB DDR3 memory.

Table 2. The time required to decoding the RSA algorithm

The number of digits for encryption	The computation time for decoding
50	4 Hour
75	104 days
100	74 years
200	4 million years
300	5 * 1015 years
500	4 * 1025 years

Table 3. Time Find agent

N Digit	2	3	4	5	6	7	8	9	10
50	4h	6h	8h	10h	12h	14h	16h	18h	20h
75	104d	156d	208d	260d	312d	354d	406d	458d	510d
100	74y	111y	128y	165y	202y	239y	276y	313y	350y

So in this method, the decoding time will increase. According to the table, if the agent number is 10, with the number of digits to encode the decoding time, it equals to 20 which is an acceptable time.

Following this method, the agents after implementing on the host, by interacting with the host and asking some questions, in other words, they pump it for information and entire it to disclose the information of other agents which have been implemented on it. In this case:

A) If the host is a Safety Host, which is called have White Host, refuses such an action, so the agent figures out that the host is a safe one and so it will increase its safety grade.

B) If the host is a Malicious Host, which is called here Red Host it is tempted and discloses its other information. Here the intended agent will figure out that this host is a dangerous and an unsafe one, so it reduces its safety grade.

C) If the host is a Malicious Host, but it is a self-controlled host which is called Red-White Host, it is not tempted and close not disclose other agents information, but the agent can't find out that the host is an unsafe one, so we must do something which keeps its confidentiality against the host, and it is possible with the above-mentioned method.

After applying this method and scaling of system (network) several times, according to a number of meetings that might happen on a host, the agents will find out by informing each other (will learn, here we have learning) that which host is safe and which one is unsafe. The unsafe hosts are deleted little by little from the executive cycle and there remains only the safe hosts and those which are not safe in terms of privacy, but represent themselves like safe ones. Agents keep their confidentiality against the second type of host according to their duplication method.

In simulation of the intended algorithm, the hosts are divided into 3 types of White, Red and Red-White Hosts.

A) White Hosts are those that their safety possibility is equal to 1, $P(SH) = 1$.

B) Red Hosts are those that their safety possibility is equal to 0, $P(SH) = 0$.

C) Red-White Hosts are those that their safety possibility is equal to 0, $P(SH) = 0$, but they are considered by the agents to be a safe host $P(SH) = 1$.

D) The unknown hosts which are willing to unsafety, but have reformed themselves at the right time and they return again to the executive cycle, these are called Gray-to-White Hosts and their safety possibility is from 0 to 1 ($0 \leq P(SH) \leq 1$).

E) The unknown hosts which are willing to unsafety, but they don't return themselves at the right time and they are deleted from the executive cycle, they are Gray-to-Red Hosts and their safety possibility is from 0 to 1 $0 \leq P(SH) \leq 1$.

In this paper, we used Brahms simulators NASA product, to do the suggested algorithm. In this simulation, we used 10 hosts whose type are represented in the left hand in table 4. In the different types of hosts, White means being safe, Red means being malicious and dangerous, White-to-Gray refers to the unknown state of the host which can reform its behavior and return again to the executive cycle and eventually the final type is Gray-to-Red which refers to the uncertainly of the host which cannot reform its behavior and it will be deleted little by little from the executive cycle.

Table 4. Results of simulation

Type		Round1	Round2	Round3	Round4	Round5	Round6	Round7	Round8	Round9	Round10	Round11	Round12
White	Host 1	16	15	17	16	15	16	17	18	20	21	22	23
Red	Host 2	6	7	6	8	9	8	7	5	3	1	0	0
Gray To White	Host 3	13	12	11	12	13	12	11	13	14	15	16	17
White	Host 4	11	11	10	11	12	12	13	14	15	17	18	17
Gray To Red	Host 5	9	8	10	8	9	10	9	7	4	2	0	0
White	Host 6	8	8	9	10	8	8	9	10	11	13	14	14
Red	Host 7	10	11	12	11	10	9	7	7	6	3	1	0
Gray To White	Host 8	7	8	7	7	6	7	9	10	11	12	14	14
White	Host 9	8	9	8	10	10	9	11	12	13	15	15	15
Red	Host 10	12	11	10	7	8	9	7	4	3	1	0	0
Sum		100	100	100	100	100	100	100	100	100	100	100	100

In diagram 1, the number of hosts meeting is represented by mobile agents. This diagram, based on table 4 shows the mobile agents in every phase of meeting, determine their meeting number of the host. According to diagram, diagram convergence is determined in seventh stage, thus the degree of hostile hosts meeting diminishes little by little to the extend that they will be deleted from the executive cycle, but the degree of safe hosts meeting will be added in every phase. Therefore by observing this diagram, the malicious hosts and safe hosts will be determined.

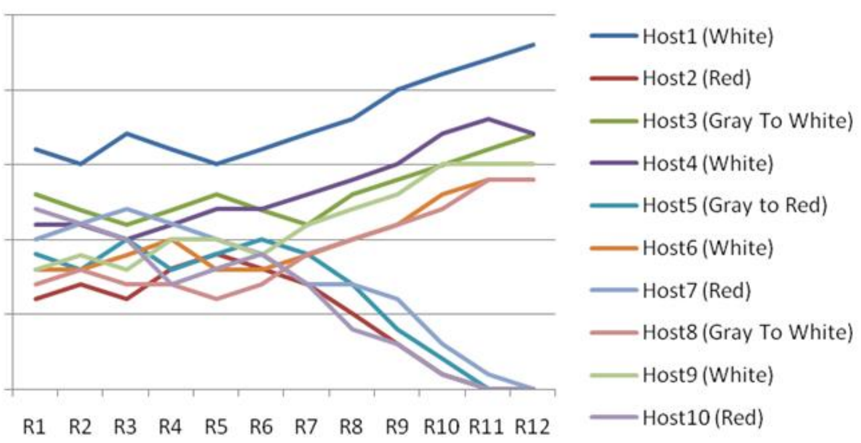

Diagram 1. The number of hosts meeting using the proposed method (10 host and met 100 times in each round of 10)

If agents features such as learning and cooperation are not used, as they are not used in previous methods, and only cryptography and decoding methods are used to keep confidentiality, dangerous hosts are not identifiable and the results are represented in table 5.

Table 5. Results of simulation

Type		Round1	Round2	Round3	Round4	Round5	Round6	Round7	Round8	Round9	Round10	Round11	Round12
White	Host1	16	15	17	16	15	16	15	15	16	15	15	16
Red	Host2	6	7	6	7	7	7	7	6	7	6	7	6
Gray To Wh	Host3	13	12	11	11	13	12	11	12	11	12	12	13

ite													
White	Host4	11	11	10	11	12	12	13	12	12	11	12	11
Gray To Red	Host5	9	8	9	8	9	9	8	7	8	8	7	9
White	Host6	8	8	9	10	8	8	9	10	10	9	10	8
Red	Host7	10	11	12	11	10	9	8	9	8	10	9	10
Gray To White	Host8	7	8	7	7	6	7	8	8	9	8	7	7
White	Host9	8	9	8	9	10	9	10	11	10	9	10	8
Red	Host10	12	11	11	10	10	11	11	10	9	12	11	12
Sum		100	100	100	100	100	100	100	100	100	100	100	100

In diagram 2, the number of hosts meeting by the mobile agents are represented. This diagram, based on table 5 shows the mobile agents in every phase of meeting, determine their meeting number of the host. In this diagram, since mobile agents features are not used in cryptography the degree of hostile hosts meeting and the degree of safe hosts meeting do not make any difference in each phase, and according to our proposed method, the hostile hosts are not identifiable whether they reform their behavior or not.

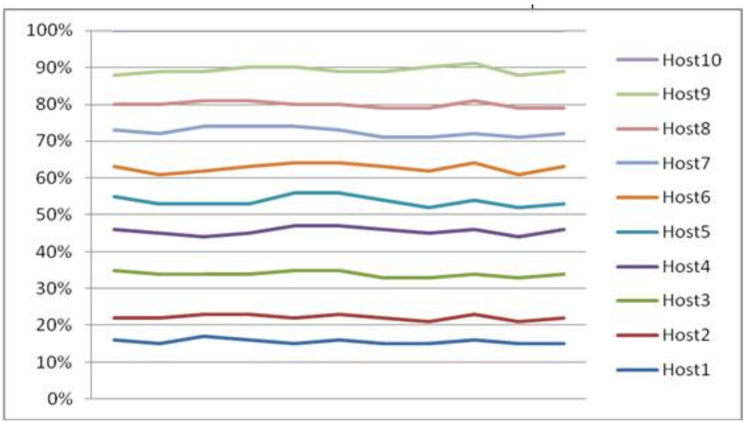

Diagram 2. The number of hosts meeting without using the features of mobile agents

5. CONCLUSION

Most of the methods which have been used so far to protect mobile agents confidentiality, use cryptography methods. Agents have features which can be used to protect mobile agents confidentiality. As they have been previously referred to, these features including cooperation, learning, duplication, mobility, independence, etc. in this paper, we have achieved on effective method from these features along with the basic method of keeping confidentiality which is cryptography to keep mobile agents confidentiality in multi-agent environments. This method was presented and evaluated algorithmically in the previous section. Moreover, the assimilation results showed that in this method, the mobile agents confidentiality have increased. Other features of this method is that if there are dangerous hosts in a multi-agent environment which violate the mobile agents confidentiality if they do not reform themselves, by using this method, they will be deleted little by little from the executive cycle and if they reform themselves. They do return to the executive cycle little by little, that results from this simulation indicate this fact. Maybe it could be said that the main problem here is the overload which has been imposed on the system by duplicating the agents and implementing them on the host that of course according to the simulation results, the number is not that much that makes the resulting overload noticeable.

6. REFERENCES

[1] Barforoush, A. A., Masoumi, B., Ayatollah Zadeh Shirazi, M., "An Introduction to Distributed Artificial Intelligence (introduced agents and multi-agent systems)", Tehran, Jelveh, 2005,

[2] Bierman E., Cloete E., “Classification of Malicious Host Threats in Mobile Agent Computing,” ACM International Conference, Vol. 30, pp.: 141-148, 2007.

[3] BRAHMS Language Specification TM99-0008 Version 2.11–Final 24 June, 2012.

[4] Cormen. Thomas, “Introduction to algorithms”, 3rded, 2009.

[5] Jansen. W. A., “Privilege Management Scheme for mobile agent Systems”. Workshop on security of mobile multi agent Systems, Autonomous Agents Conference, 2001, First International.

[6] Jansen W., “Countermeasures for Mobile Agent Security”, Component-based Software Engineering: Putting the Pieces Together, 2011.

[7] Kostakos, V., Taraschi, Agents (pp 4-6), 2001.

[8] Malekian, E., "Network attacker and prevention", Tehran, Nas, 2003.

[9] Milgrom, E., Final guidelines for the Identification of relevant Problem Areas where Agent Technology is Appropriate, Project Report CIDETI, 2001.

How to cite this article:
Ebrahimi S. M. M. Using of colleague learning mobile agents for protecting the confidentiality of mobile agents in a multi-agent environment. J. Fundam. Appl. Sci., 2016, *8(3S), 579-598.*