Dynamic Trust Management Framework for Robotic Multi-Agent Systems

Zikratov Igor[1], Oleg Maslennikov[1], Ilya Lebedev[1],
Aleksandr Ometov[2], and Sergey Andreev[2]

[1] Saint Petersburg National Research University of Information Technologies, Mechanics and Optics (ITMO University), St. Petersburg, Russia
[2] Tampere University of Technology, Korkeakoulunkatu 10, FI-33720, Finland
Email: aleksandr.ometov@tut.fi

Abstract. A lot of attention recently gone to multi-agent systems due to robotics automatic, pro-active, and dynamic problem solving behaviors. Over past decades, there has been a rapid development in agent technology which has enabled to provide or receive useful and convenient services in a variety of areas. In many of these services, it is required that security is guaranteed. Unless it is, these services would observe significant deployment issues. In this paper, a novel trust management framework for multi-agent systems focused on access control and node reputation is proposed. It is further analyzed utilizing a compromised device attack proving its suitability of the utilization.

Keywords: Multi-agent system, Security, Access control, Trust.

1 Introduction

Today, the swarm multi-agent robotics is one of the most significant and complicated fields of research recalling the fact that less than 5% of our planet, both land and oceanic, has been explored so far[1]. Modern robots employed to the surface research, protection and monitoring are extremely complicated entities equipped with a variety of sensing equipment [1]. Therefore, it is significant to keep them operational for as long as possible[2]. For example, wildfire fighting is one of the most physically challenging task faced by human-workers today. Autonomous machines can contribute a lot in the manner of this hard, dirty, exhausting and dangerous job. Devices can operate faster and more efficiently while keeping people away from unsafe locations[3]. Conventionally, those devices are supposed to cooperate with each other in order to reach common "targets"

[1] See: NOAA National Ocean Service: How much of the ocean have we explored? 2014. http://oceanservice.noaa.gov/facts/exploration.html
[2] See: Autonomous Fire Guard (AFG) concept. 2009. http://www.yankodesign.com/2009/08/21/firefighters-best-friend/
[3] See: The National Interagency Fire Center (NIFC): Incident Management Situation Report. 2016. http://www.nifc.gov/nicc/sitreprt.pdf

in distant areas [2]. This distributed approach has many advantages in achieving cooperative group performances, especially with low operational costs, less system requirements, high robustness, and flexible scalability. It has been widely recognized and well-studied over past years [3, 4].

In multi-agent systems, the network topology among all devices plays a crucial role in determining consensus. Commonly, the objective is to explicitly identify necessary and sufficient conditions for the network topology such that common agreement could be achieved under properly designed algorithms.

One of the most promising trends in robotics is development of the management tool allowing intellectual Multi-Agent System (MAS) group control. Particularly, the most attraction gone to the collaborative planning frameworks operating in the decentralized ad hoc way by forming a coalition [5]. This is due to higher scalability, operational coverage and network availability in cases of weak connectivity to control unit. Significant portion of the research focus in this field is related to the dynamic goals redistribution between the operating nodes in case of node's possible unpredictable breakdown [6].

Due to the ad hoc behavior of such networks and the operation in full mesh way, MAS becomes an attractive field for a wide range of attacks, such as: message capturing and retransmission, violation of integrity, unauthorized data access, denial of service, etc. [7]. Therefore, currently utilized trust managements schemes are significantly limited due the discretionary distinction and a mandates behaviors [8, 9]. We may classify the main groups of attacks on MAS as following [10]: (i) network layer related attacks; (ii) attacks on the identification and authentication of agents in the system; (iii) *compromised device* intrusion [11]. The main goals of this work are to develop a trust management framework suitable for resisting the compromised device attack.

The paper is organized as follows. In section 2 we overview the MAS decentralized systems and the corresponding attacks. Further, in section 3 we present a trust model resistant to discussed attacks. Next, in section 4 the compromised device intrusion attack detection in detailed. The last section describes the future work and provides conclusions.

2 Background

In this section, we study a MAS operating in a decentralized way [12, 13]. We consider a group of N robots targeting the collaborative goal. During the initialization phase, each of the devices receives the utility function (goal) related data. The framework operation model is depicted in Fig. 1.

Each device's $R_i, (i = \overline{1, N})$ processing unit P_i consists of the corresponding computing unit CU_i, data transmitting unit DT_i, data receiving unit DR_i, current state determination unit CS_i, and a set of sensing devices SD_i. CU_i is communication with the other CU_j by transmitting system state information S_i^0 and the corresponding operational decisions $A_i^{k+1}, (k = 0, 1, 2, \ldots, N)$. Each CU_i also has a knowledge about the environmental data E_i^0 and it's own state

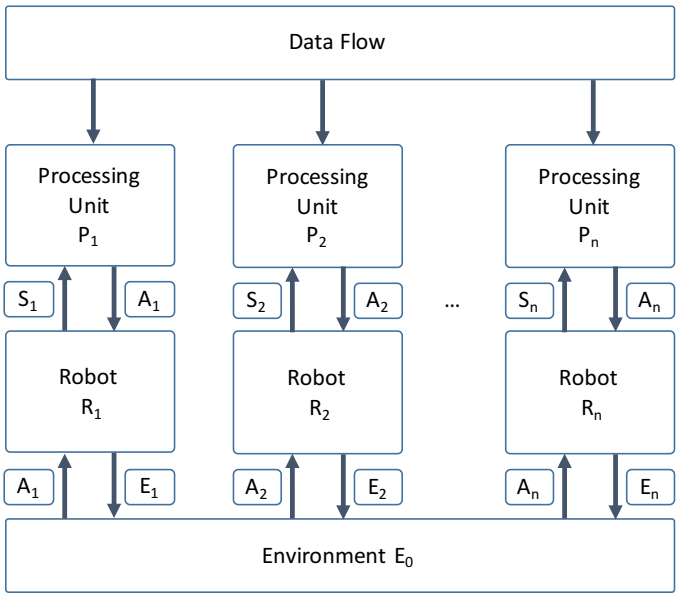

Fig. 1. Simplified decentralized MAS management framework.

S_i^0 for continuous updates of the utility function ΔY for any possible operational decision in current state. We select $max(\Delta Y)$ as our utility function.

As an attack, we define a malicious activity of the compromised device on k^{th} iteration of the system operation [14]. As a result, the next device decision A_i^{k+1} would not be selected according to the utility function. We also consider such attacks as: message capturing and retransmission, attack on the environment estimation, and attacks targeted to affect the group decision making protocols [15].

Conventionally, MAS utilizes the following techniques to enable secure ad hoc communications: state-appraisal function [16]; lightweight cryptography solutions [17], time-limiting solution [18], Buddy Security Model (BSM) [19, 20], and others. Interestingly, surrounding nodes in BSM are responsible for the security of each other by monitoring their environmental continuously. This is reached by means of BSM users exchanging defined *tokens* with confidential state information and potential security threads of the surrounding devices. By informing the neighboring nodes about nonstandard behavior or immersion on a new device, each agent brings its portion of stability to the system security and, as a result, his own one.

Today, BSM is getting more attraction today mainly due to its decentralized nature. On the other hand, utilization of this model in the robotic systems could be still affected by the compromised robot. The scenario of interest is the remote areas ad hoc network operation where providing reliable connection to

the centralized control unit is a challenging task. Therefore, physical capture of the device and compromised token are possible. In this work, we demonstrate an improved BSM by introducing the device's trust level slightly strengthening the complexity of aforementioned attack [21].

2.1 Multi-agent trust model for robotic systems

This section describes the MAS operation in a steady-state mode, i.e. after the initialization phase. In current state S_i^0, each i^{th} robot $R_i, (i = \overline{1,N})$ collects the data from $CU_j, (i \neq j, j = \overline{1,N})$ of the other robots of his group. After this phase, it selects A_j^{k+1} according to the utility function ΔY and reports the corresponding decision to other devices with $w(A_i^{k+1})$ to $CU_j, (i \neq j, j = \overline{1,N})$. This message is based on the received information $S_1^0, S_2^0, \ldots, S_i^0 - 1, S_i^0 + 1, \ldots, S_N^0$ and current possible decisions $A_1^{k+1}, A_2^{k+1}, \ldots, A_i^{k+1} - 1, A_i^{k+1} + 1, \ldots, A_N^{k+1}$. After receiving this message, other agents are validating the received data regarding the decision made by i^{th} node.

In case the check by j^{th} device resulted in $\Delta Y_j, (i \neq j) \neq \Delta Y_i$, the trust level of i^{th} device is increased. By trust level we define an *aspiration* of the selected node to report valid information to others. Contrariwise, if the device has reported a non-optimal ΔY_i – its level of trust is decreased. As a result, we may define a new parameter being a set of "steps" l required to estimate a *deep-seated* level of trust per device A_i^{l+1} and, thus, for calculating ΔY_i^l on the next system iteration, the devices would rely more on highly trusted nodes.

Summarizing, the lower level of trust would not let the compromised robot to effect the system operation in the destructive way even by sending a valid-like token. Thereby, malicious node should behave like a faithful one for the interval of time at least equal to being pernicious, which goes contrary to the malicious needs logic.

3 Trust model development

In this section we define the notations for our security mechanism and discuss the implementation possibility. The developed trust model is presented in Fig. 3 where arrows represent multi-agent connections by additional channel, for example, a sensor module (visual, NFC, etc.), and the wireless radio links are depicted with dashed lines. For the ease of discussion, we further introduce the notations used in his section: $A = \{A_1, A_2, \ldots, A_N\}$ – possible operations that may be performed by the agent; $S = \{s_1, s_2, \ldots, s_N\}$ – a set of states during the communications phase; $V = \{F, T\}$ – a set of results given by report validation, where F corresponds to the invalid reply and T for a valid one; and r_l^m – a trust level determined by m^{th} agent for l^{th} one $(l \neq m)$. There are different ways of the model implementation according to Fig. 3.

Firstly, we assume the system being in state k, and we focus on the $2nd$ device in the network. It has reported a message $2A_2$ to the rest of users. As depicted in the figure, the $2nd$ robot has wireless connection to devices 1, 3 and 8, and

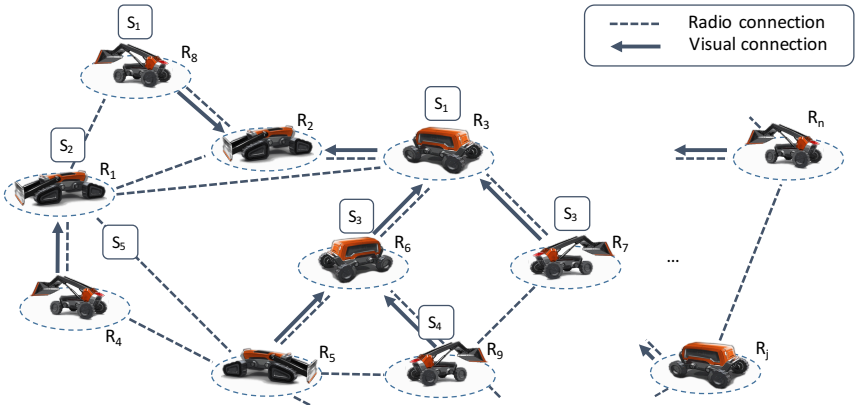

Fig. 2. Proposed trust model agents interactions.

all of these users have received the corresponding message. Moreover, devices 3 and 8 have a visual proof that 2*nd* device has performed the reported action. The set of possible system states S could be represented as following:

- s_1: the object is in a line of sight and reached via radio (for devices 3 and 8);
- s_2: the object is not in a line of sight and reached via radio (for device 1).

If the 2*nd* device is acting normally, his actual operation A_2 is equal to the reported one, i.e. devices 3 and 8 would increase the level of trust for the 2*nd* robot by $\Delta r_2^3 = \Delta r_2^8$ due to visual confirmation of the action. The device 1 has also received the report from the 2*nd* device but utilizing only wireless channel. In this case, his level of trust would be updated less than for other nodes $\Delta r_2^1 < \Delta r_2^{3,8}$. In this case, only the devices having direct connection to the robot are estimating its level of trust but it is kept unchanged for the rest.

Secondly, devices having direct connection to the evaluated one may forward the knowledge to their neighbors. They would transmit message of the type $i : A_i s_k^j v$, where i – is a reporting device identifier, A_i – a value reported by i^{th} device, s_k^j – is j^{th} device state, and v – is the validation result (either *true* or *false*). For this example, the corresponding messages would be: from robot 1 – $2 : A_i s_2^1 T$; from robot 3 – $2 : A_i s_1^3 T$; from robot 8 – $2 : A_i s_1^8 T$. Those messages are delivered to agents $4, 5, 6, 7, 9$. Note, for different ad hoc topologies and network dynamics, the results of the message distribution would vary. Therefore, the set of states S for this case could be represented as following:

- s_1: the object is in a line of sight and reached via radio (for devices 3 and 8);
- s_2: the object is not in a line of sight and reached via radio (for device 1).
- s_3: the subject s_1 is in a line of sight and reached via radio (for device 6 and 7).

– s_4: the subject s_1 is not in a line of sight and reached via radio (for device 9).
– s_5: subject s_2 is in a line of sight and reached via radio (for device 1).
– s_6: the subject s_2 is not in a line of sight and reached via radio (for device 1).

Subjects having different states $s_1, s_2, \ldots, s_6$ would obtain non-equal trust levels of the same subject based on the possibility to evaluate the device by themselves. An incremental trust scale for i^{th} object should be introduced as

$$\Delta r^i_{s_1} > \Delta r^i_{s_2} > \cdots > \Delta r^i_{s_6}. \tag{1}$$

The main goal for non-directly connected agents is to determine own state and to base the objective level of trust based on it. Correspondingly, if the value of $v = T$ the level of trust is increased by $\Delta r^n_{s_i}$. Similarly, if $v = F$ the level is decreased by the same amount. Important to note, receiving the contradictory data from different nodes may be caused by variety of factors starting with uncontrollable interference to basic fog, or, more probably, by the deliberate distortion of the message by one of the relaying nodes. In this case, the goal may be achieved by utilizing the preset information security policy of the MAS.

Summarizing, the second implementation of the proposed trust model would be more effective. It allows to receive actual trust information for higher number of agents, however, by increasing the number of signaling messages.

4 Detecting attacks on MAS

Proposed in the previous sections trust level management mechanism allows to withstand such threads as: messages A_i and S_i capture, modification and retransmission, and compromising the system operation by the attacker node that may attempt to influence the utility function ΔY. On the other hand, some MAS management protocols allow to select a *leading* device from the group which becomes responsible for handling decision-making system functionality [22, 23], i.e. updating the utility function goal for the entire network.

Like in any system with a single-node failure possibility, obtaining such a role by the attacker would cause the extermination of the system operation. This attack may be also executed by a set of devices in the group. In this case, the only way to detect this detrimental behavior is by monitoring ΔY by all the system agents and on each iteration of the network operation. In order to perform such a monitoring, all the agents, except for the reporting one, are obliged to recalculate the incrimination potential ΔY_{t+1} of i^{th} node on l^{th} step. In case $\Delta Y_{t+1} < \Delta Y_t$, receiving device decreases the level of trust for the reported one by $\Delta r^i_{Y_l}$ and vice versa. The resulting level of trust for i^{th} device could be calculated as

$$\Delta \mu^i > \alpha \Delta r^i_{s_i} + \beta \Delta r^i_{Y_l}, \tag{2}$$

where α and β – are the weighted verities of the reported agent and utility of its decision being selected according to the information security policy of MAS.

5 Selected numerical results

In order to validate the usability of our model, we have conducted a set of simulation results utilizing V-REP robotics framework[3]. To prove the effectiveness, we compare the changes of ΔY_i^l over the framework operation according to section 3.

Initial system setup is described as following. Each agent has a complete knowledge about MAS goals, corresponding distances between agents and the number required to solve the goal per each target. After all the agents have exchanged this data, each agent is selecting the closest target comparing his R and corresponding other agents' R_{min} distances to it. If $\Delta Y_i^l = A_i(min) - A_i$ is positive, the agent reports about his decision to proceed with current target, otherwise, it waits.

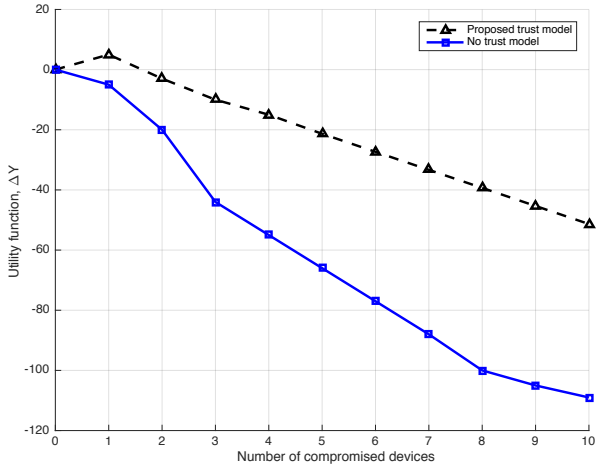

Fig. 3. Utility function on number of compromised devices dependency (uniform distribution of agents).

For our experiment, we used 50 agents uniformly distributed over the circular area with a radius of 50 meters. Number of targets is three with the corresponding 5, 3 and 2 agents required. Each agent has radio coverage of 30 meters and line of sight of 7 meters. We vary the number of compromised nodes in the system to validate the framework operation.

Regardless of the system class, the trust model utilization reduces the impact of attacks on the system efficiency (See Fig. 3 and 4). For the second class,

[3] See: V-REP `http://www.k-team.com/mobile-robotics-products/v-rep`

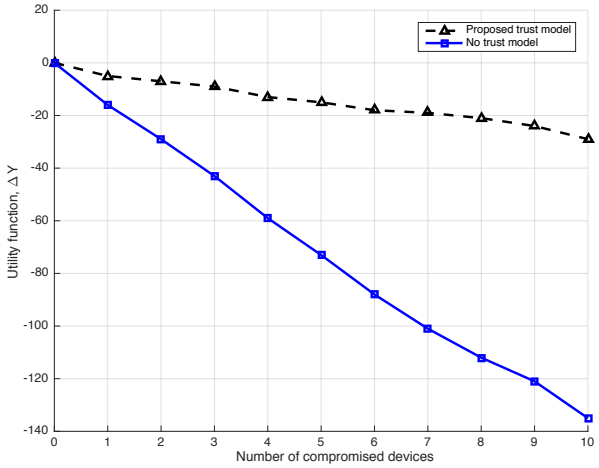

Fig. 4. Utility function on number of compromised devices dependency (each agent has at least one visual neighbor).

where each agent has at least one neighboring node with the corresponding visual contact, the benefits are more significant (Fig. 4). The proposed model also may bring negative impact on the system efficiency, particularly, when the compromised node has been the best possible selection for a target, however, it was ignored due to the confidence level reduction.

6 Conclusions

In this work, we have developed a trust model for decentralized robotic MAS networks. Our framework provides group access based on the devices' level of trust selected and dynamically updated over time. It was successfully evaluated and could be utilized for modern MAS systems.

The main advantage of the proposed approach is to allow continuous and secure communications for robotic ad hoc networks facing lack of reliable connection to the centralized control unit. The secondary benefit is time driven dynamic trust level updates determining the trust level actuality, i.e. the joining device would be required to operate for a significantly long time in order to achieve valuable decision making right.

References

1. L. Hernandez, C. Baladron, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, J. Lloret, D. Chinarro, J. J. Gomez-Sanz, and D. Cook, "A multi-agent system ar-

chitecture for smart grid management and forecasting of energy demand in virtual power plants," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 106–113, 2013.
2. Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 427–438, 2013.
3. W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. of 2005 American Control Conference*, pp. 1859–1864, IEEE, 2005.
4. V. R. Lesser, "Reflections on the nature of multi-agent coordination and its implications for an agent architecture," *Autonomous agents and multi-agent systems*, vol. 1, no. 1, pp. 89–111, 1998.
5. O. M. Shehory, K. Sycara, and S. Jha, "Multi-agent coordination through coalition formation," in *Intelligent Agents IV Agent Theories, Architectures, and Languages*, pp. 143–154, Springer, 1997.
6. M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.
7. Y. Jung, M. Kim, A. Masoumzadeh, and J. B. Joshi, "A survey of security issue in multi-agent systems," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 239–260, 2012.
8. D. Bell and L. LaPadula, *Secure Computer Systems: Unified Exposition and Multics Interpretation*, vol. MTR-2997 R. Bedford, Mass.: MITRE Corp., 1976.
9. M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in operating systems," *Communications of the ACM*, vol. 19, no. 8, pp. 461–471, 1976.
10. F. Higgins, A. Tomlinson, and K. M. Martin, "Threats to the swarm: Security considerations for swarm robotics," *International Journal on Advances in Security*, vol. 2, no. 2&3, 2009.
11. S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Security in pervasive computing*, pp. 201–212, Springer, 2004.
12. O. Kachirski and R. Guha, "Effective intrusion detection using multiple sensors in wireless ad hoc networks," in *Proc. of the 36th Annual Hawaii International Conference on System Sciences*, pp. 8–pp, IEEE, 2003.
13. A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 48–60, 2004.
14. K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 2, pp. 245–257, 2011.
15. S. Basagni, "Distributed clustering for ad hoc networks," in *Proc. of Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)*, pp. 310–315, IEEE, 1999.
16. N. M. Karnik and A. R. Tripathi, "Security in the ajanta mobile agent system," *Software: Practice and Experience*, vol. 31, no. 4, pp. 301–329, 2001.
17. T. Sander and C. F. Tschudin, "Protecting mobile agents against malicious hosts," in *Mobile agents and security*, pp. 44–60, Springer, 1998.
18. F. Hohl, "Time limited blackbox security: Protecting mobile agents from malicious hosts," in *Mobile agents and security*, pp. 92–113, Springer, 1998.
19. J. Page, A. Zaslavsky, and M. Indrawan, "A buddy model of security for mobile agent communities operating in pervasive scenarios," in *Proc. of the second workshop on Australasian information security, Data Mining and Web Intelligence, and*

Software Internationalisation, vol. 32, pp. 17–25, Australian Computer Society, Inc., 2004.
20. J. Page, A. Zaslavsky, and M. Indrawan, "Countering security vulnerabilities using a shared security buddy model schema in mobile agent communities," in *Proc. of the First International Workshop on Safety and Security in Multi-Agent Systems (SASEMAS 2004)*, pp. 85–101, 2004.
21. I. A. Zikratov, I. S. Lebedev, and A. V. Gurtov, "Trust and reputation mechanisms for multi-agent robotic systems," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp. 106–120, Springer, 2014.
22. Y. Hong, J. Hu, and L. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol. 42, no. 7, pp. 1177–1182, 2006.
23. W. Ni and D. Cheng, "Leader-following consensus of multi-agent systems under fixed and switching topologies," *Systems & Control Letters*, vol. 59, no. 3, pp. 209–217, 2010.