Integrating ROS and Gazebo Tools with a Network Security Module to Support Secure Autonomous Robot Coordination

Mattia Giovanni Spina, Stefano Gualtieri and Floriano De Rango[a]
Department of Computer, Modeling, Electronics and Systems Engineering (DIMES), University of Calabria, Via P. Bucci, Rende, Italy

Keywords: Robot Coordination, Security, ROS, Gazebo, Autonomous Robots.

Abstract: Multi-robots system coordination is an important aspect to consider when complex task needs to be performed. Even if robots are becoming always more autonomous, the collective behaviour and coordination strategy can improve the overall performance in terms of execution time increasing the robustness of the mission. However, few works addressed the issue of the network security related to the coordination strategy and the current modelling and simulations tools are not ready to model security aspects that can affect the task execution and in some case can compromise the mission The following paper proposes the integration of some additional module on well-known tools such as ROS and GAZEBO in order to extend the modelling aspects also on emerging trends to support technicians to evaluate the coordination strategies also form the security point of view.

1 INTRODUCTION

Robot coordination and multi-robot applications are gaining a lot of interest in these last years. Involving more robots in missions or complex tasks has been shown to produce many benefits in terms of success of the mission or in terms or reduction of the overall task execution time. A lot of attention in literature has been given in these years about the coordination strategy combining explicit coordination among robot or implicit coordination. However, in our opinion, a too few attentions have been given to security aspects related to the robot coordination. Even if a lot of work has been done on SLAM technique (Park and Lee, 2017) to improve the localization and perception of a surrounding environment in a robot and also if some protocol to distribute among robots partial built maps of the surrounding environment, no attention has been focused on some possible threats can be arise when a robot can behave maliciously or some attacks can be performed to degrade or compromise the task execution.

The main contributions of this paper are listed below:

1. ***Network Layer Design:*** this contribution is related to the introduction of a network layer in ROS where all communication paradigm supported is a publish/subscribe that is an application layer mechanism. In our case we simulated a network layer adopting an application layer paradigm. This has been led out introducing the channel model, the communication range and the routing layer to build the robot topology on the basis of the exchanged packets. The main faced issues have been the mapping of network functionalities at the application layer to simulate the network services.
2. ***Security Feature Design:*** ROS and Gazebo do not consider any network security features. This means that it is possible to model only some robot characterization such as movement and map building but it is not possible to consider possible security threats related to robotic applications. In future situations where multi-robot systems can be involved in complex tasks, the network security in the robot communications and data sharing can be a key issue to face because some critical operations could be compromised. This means that considering the current state of the art for the mentioned simulators, security is not supported (Rivera et al., 2019) (Mukhandi et

[a] https://orcid.org/0000-0002-3882-1678

Spina, M., Gualtieri, S. and De Rango, F.
Integrating ROS and Gazebo Tools with a Network Security Module to Support Secure Autonomous Robot Coordination.
DOI: 10.5220/0010566303690377
In *Proceedings of the 11th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2021)*, pages 369-377
ISBN: 978-989-758-528-9

al., 2019). At this purpose, some basic security features for supporting authentication, integrity check and encryption have been introduced and integrated in the simulation and modeling framework. More details on the network security aspects will be presented in the next sections.

All the proposal has been implemented integrating two simulators such as ROS (Robot Operating Systems) and GAZEBO that works at application layer. The paper is organized as follows: section 2 presents the work in literature related to robot coordination for unknown area discovery and recruiting tasks; section 3 presents the main tools adopted in our proposal to simulate map building of the unknown area and robot coordination; section 4 introduces all modules and robot models considered in our framework; the communication protocols and modules for the recruiting task and robot coordination are described in section 5; some security features and threats are presented in section 6 and finally conclusions are summarized in the last section.

2 RELATED WORK

Coordination of multi-robot systems received much attention in recent years due to its vast potential in real world applications. Simple robots work together to accomplish some tasks. However, the execution of complex task sometimes needs to involve multiple robots. In this last case, robot coordination become an essential point to guarantee. To perform this objective the communication among robots is a key element to consider and also possible threats to the communication should be accounted. Robots' coordination strategies can be broadly divided into two main categories: explicit coordination and implicit coordination.

Explicit coordination refers to the direct exchange of information between robots, which can be made in the form of the unicast or broadcast of intentional messages. This often requires a dedicated on-board communication module.

Existing coordination methods are mainly based on the use of explicit communication that allows the accuracy of the exchange of information among the robots' swarm (De Rango et al.,2018), (Tropea et al.,2019). Instead, implicit coordination is usually associated with implicit communication, which requires the explorative robots to perceive, model, and reason others' behavior. In this case, an individual robot makes independent decisions on how to behave, based on the information it gathers through its own perception with others. When the robots use an implicit communication to coordinate, although the information obtained by the robots is not completely reliable, and the stability, reliability, and fault tolerance of the overall system can be improved (Palmieri et al., 2019), (Palmieri et al., 2018). However, in this last case, an increase in the execution task among robots can be observed. In our case we are interested in hybrid approach where robots applying SLAM can move and perceive the environment independently through its sensors, but it can also receive information about neighbor robots about part of the map already built by them in order to speed up the overall task of unknow space discovery.

Main contributions in comparison with the state of the art are related to the integration of multiple well-known tools for the robots modeling and simulations with some modules to account for the energy consumption, network layer modeling for supporting the topology discovery and for the security features to apply in the communication to reinforce the explicit coordination mechanism. These two aspects are essential to model and simulate real context where robots can move and where some threats can be present that can compromise the overall mission.

3 SIMULATION TOOLS

Different tool and technologies have been applied to implement our simulation scenario of robots under security threats and coordination strategies.

3.1 Robot Operating System (ROS)

Robot Operating Systems (ROS) (Gatesichapakorn et al., 2019) is a framework for the design and programming of robot. It can create a robot network where many processes can be connected. Moreover, it offers all functionalities to design a distributed system providing also services typical of an operative system (OS) such as: hardware abstraction, device controller through drivers, process communication, application management (package) and other features. Processes inside ROS can be represented through graph structure where nodes can send, receive and route messages coming from other nodes. Nodes can also be sensors and/or actuators. Some basic elements of ROS are recalled in the following:

1. **Roscore:** it is the master node that provides the names registration and the discovery service of

the other nodes. It can also set the connections among nodes. If this node is not instantiated, no communication is allowed among nodes.

2. **Nodes:** These are the entities that can store data or computing tasks. Every process that needs to interact with other nodes inside the ROS network needs to be instantiated as a node and it should be connected to the master node (Roscore). In our case, robots are represented by nodes.
3. **Messages:** they are the structure that represent the messages exchanged among nodes. ROS presents different default messages. However, it is possible customize new messages with additional info.
4. **Topic:** When a node sends data, it needs to publish data on a particular data structure called topic. It is like a publish/subscribe paradigm where nodes can exchange data publishing their data and other nodes can receiver these data if they subscribe on the same topic. This paradigm allows a separation between data generation and data consumption.

3.2 Gazebo

Gazebo is an open-source 3D robotic simulator (Raje and Sumit, 2020). It integrates the dynamic physic engine called ODE (Open Dynamic Engine) that is written in C/C++. It is equipped with a rendering tool in OpenGL and it provides code to support robots, sensors and actuators simulator. It supports a high-resolution realistic rendering of the environment where robots can move including lights and shadows in the image detected by cameras. It can model sensors that can perceive the surrounding environment such as laser sensors, cameras (with the large angle view) and sensors such as Microsoft Kinect. Gazebo is very useful for robotic modeling applications allowing also complex and detailed simulations. A well-designed simulator allows to quickly test algorithms, to design robots, to execute regression test and to train artificial intelligence systems using realistic scenarios. Gazebo offers the possibility to simulate with high precision and efficiency multi-robot systems for complex indoor and outdoor environments. It can be integrated with ROS through the package Gazebo-ROS.

3.3 Slam-Gmapping: Navigation

It is a module able to simulate the robot's movement applying the Gmapping technique (Abdelrasoul et al., 2016). This last one is a highly efficient Particle Filter technique such as Rao-Blackwellized designed to re-build a map on the basis of data received by specific sensors such as Laser whose robot is equipped. These filters have been recently introduced to face issues such as SLAM (Simultaneous Localization and Mapping) (Ibáñez et al., 2017). In this approach, each particle maintains an individual environment map. This specific technique has the objective to reduce the uncertainty related to robot location and it is optimized for long-range laser sensors.

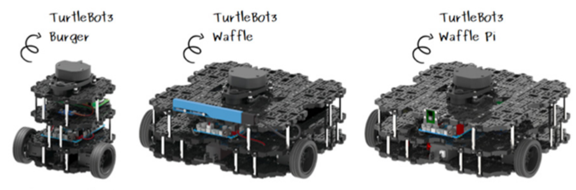
SLAM represents a process that allows a robot to move in an unknown environment building at run-time an environment map localizing itself inside the MAP. It applies well-known techniques such as Kalman filter, Covariance Intersection and GraphSLA. SLAM algorithms can be applied and adapted on the basis of the available resources in order to reach a targeted objective. SLAM can be applied in many robotic applications involving UAV, underwater rovers or home robots. The module supporting SLAM can be very useful for our purpose because we can model the robot movement avoiding collision during the movement and focusing more on other objectives such as security or coordination strategies. SLAM will support the robot navigation system supporting a robot in detecting its position in the reference frame related to the map and to plan a path toward a target position. A robot to move needs of an environment representation building a map and interpreting correctly all info included in this map representation. Even if in many applications the robot can move with pre-loaded maps, in our case, we applied SLAM to build MAP at run-time on the basis of data collected by sensors.

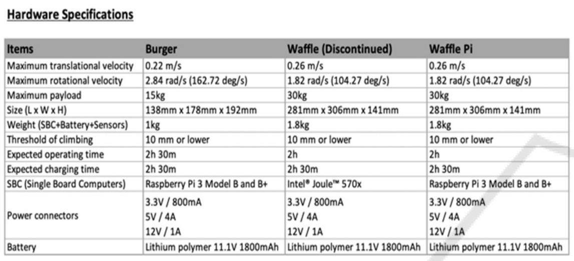
4 MODULES AND COMPONENTS FOR THE ROBOT ASSESMENT

In the following sub-sections all modules and tools adopted to simulate robots, sensors and environment where robots move will be briefly presented.

4.1 Turtlebot 3

It is considered in our evaluation the Turtlebot3 robot such as presented in (ROS.org "About Turtlebot 3"). It represents a low-cost robot with open-source control software and based on ROS environment. It is often used in the academic environment, in the research field and for prototyping embedded solutions. TurtleBot3 can be customized and it is possible to extend its basic functionality introducing additional modules focusing on specific target and actions. The basic features include three different versions: Burger, Waffle and Waffle Pi. Each of these versions present different physical and technical characteristics. In our case we considered the first one because on the basis of its characteristics it can present a lower energy consumption prolonging more the battery lifetime such as shown in Figure 1.





Hardware Specifications

Items	Burger	Waffle (Discontinued)	Waffle Pi
Maximum translational velocity	0.22 m/s	0.26 m/s	0.26 m/s
Maximum rotational velocity	2.84 rad/s (162.72 deg/s)	1.82 rad/s (104.27 deg/s)	1.82 rad/s (104.27 deg/s)
Maximum payload	15kg	30kg	30kg
Size (L x W x H)	138mm x 178mm x 192mm	281mm x 306mm x 141mm	281mm x 306mm x 141mm
Weight (SBC+Battery+Sensors)	1kg	1.8kg	1.8kg
Threshold of climbing	10 mm or lower	10 mm or lower	10 mm or lower
Expected operating time	2h 30m	2h	2h
Expected charging time	2h 30m	2h 30m	2h 30m
SBC (Single Board Computers)	Raspberry Pi 3 Model B and B+	Intel® Joule™ 570x	Raspberry Pi 3 Model B and B+
Power connectors	3.3V / 800mA 5V / 4A 12V / 1A	3.3V / 800mA 5V / 4A 12V / 1A	3.3V / 800mA 5V / 4A 12V / 1A
Battery	Lithium polymer 11.1V 1800mAh	Lithium polymer 11.1V 1800mAh	Lithium polymer 11.1V 1800mAh

Figure 1: Three types of Turtlebot.

4.2 Modules

The robot has been designed considering a modular approach. This means that each robot is composed by a set of modules executing specific tasks. Each module has been implemented in Python and a conceptual scheme of all modules implemented are presented in Figure 2 below.

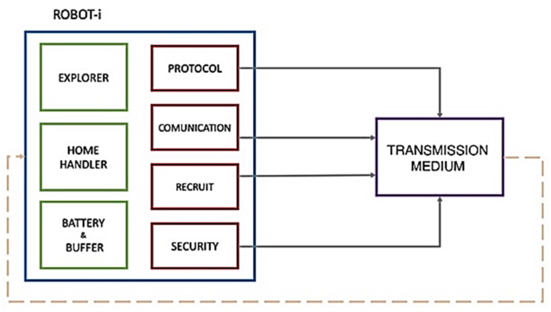


Figure 2: Block diagram of modules implemented in ROS.

The channel module has been designed to simulate the physical channel that allow the communication among nodes. In our case we considered this channel as a broadcast wireless channel able to simulate collisions, transmission and propagation delay in the data forwarding. Through this channel module it is possible to monitor all packets that travel on the network and this will be useful in the security analysis that will be presented in the next sections.

The specific component included in the robot model are now briefly introduced:

1. **Explorer Component:** It is the module that manages the exploration task. It is an important module that allows the exploration of unknown or known spaces in an autonomous way. It allows the implementation of exploration strategy that can use local knowledge of cells to explore or novel points where to move. Moreover, this module is connected with all the other components coordinating the other modules.
2. **Battery-Buffer Component:** it is an internal module for the management of the battery and the message buffer. It is considered in our case a simple battery discharge model that considers the time as variable to reduce the energy. On the other hand, the buffer is considered to store video frame produced by cameras on robot about the surrounding environment. In our case, it is considered a simple model to save a number of frames proportional to the travelled distance.
3. **Security Manager Component:** it is the module designed to manage all security features. It manages the authentication, the cryptography to support the confidentiality in the data forwarding and the key negotiation and exchange. In our case we considered ephemeral keys and this means that the adopted keys are applied for a limited amount of time and then they need to be re-generated and exchanged again. This approach has been used to mitigate the key leak issue.
4. **Home Handler Component:** this module manages the charging stations for robots. It is essential because it allows robots to come back home when their resources are exhausting. After coming back to some of deployed base stations, they can re-charged and it is possible also to download all frames produced and stored during the exploration.
5. **Protocol Component:** this module allows the neighbor discovery and the topology table update. The neighbor discovery is useful to understand in a given time which robots are directly connected to a specific robot. The topology table building and propagation is important instead to build the overall topology. This last one is essential when we need to involve/recruit some robots in more complex tasks. Through these two submodules is possible to maintain in the time the robots' topology

under dynamic condition such as robot movements.

6. **Recruit Component:** it is a key component because it focused on the logic to recruit other robots on the basis of the perceived environment and on the basis of the task to be executed. It is also related to the communication protocol that can be implanted and it can affect the overall performance of the coordination and cooperation strategy.
7. **Communication Component:** it is related to the communication protocol and to the periodical data forwarding to let other robots know about neighbors and already explored maps. The communication strategy is based on the communication protocol selected and it is related to the module presented above. This module used in a joint way with the protocol module are useful to reduce the exploring task because they try to reduce the overlapping in the exploration among robots.

5 COMMUNICATION NETWORK LAYER PROTOCOLS

In order to model the topology table building and propagation, it has been designed a network layer protocol based on the publish-subscribe paradigm. This choice has been determined by the basic programming module implemented in ROS tool. Avoiding to violate this basic feature offered by the tool, we implemented a network layer adopting this novel paradigm but considering all physical conditions in the data propagation such as channel modules (such as explained in the previous section), radio propagation range etc. For our purposed we considered a modified and enhanced version of the ***Link State*** routing protocol where some metrics such as *residual energy*, *buffer space* and *robot distance* has been considered to build the robot connectivity graph. The link-state based topology has been essential to offer the possibility to the robot to recruit other robots in order to cooperatively explore a specific area reducing the exploration task time. In addition to the link-state protocol, it has been considered also another simple approach to recruit robots that is based on a progressive recruiting request forwarding. This recruiting protocol has been called ***expanding ring - recruiting request*** protocol because it propagates the recruit request considering an incremental hop in the propagation whereas the target number of robots to be recruited is not reached.

5.1 Link State Routing Module

The routing protocol implemented in the routing module is the link state (LS). It is a protocol that supports a local periodical update to build the neighbor table and an event driven topology update forwarded in broadcast to all robots to build an overall consistent topology. The Link State Update (LSU) packet considered for our purpose brings some information useful for the specific coordination task such as robot coordinates (X,Y), residual energy, timestamp, sequence number etc.

Table 1: Some protocol packets field in the LS protocol.

Protocol Packet		
Type	**Fields**	**Explanation**
string	source	ID of the robot that's currently handling the protocol pkt
string	destination	ID of the robot to which the source forwards the protocol pkt
float64	timestamp	Timestamp of protocol pkt
String[]	routing table	Current routing table of the source robot

Some fields included in the recruiting protocol are presented in table II.

The protocol uses Dijkstra to build the minimum spanning tree among robots. The metric adopted is the minimum hop count and the residual energy is important to evaluate if the recruited robot has enough energy to reach the target position where to perform the task. The routing module is flexible and it is possible to select the local broadcast for updating the neighbor table and it is possible to establish also the metric to build the minimum spanning tree. This module is connected with the security module when some protection mechanism is applied in the LSU or HELLO packet (local broadcast) update.
In the following some fields included in the link-state routing protocol are presented.

Table 2: Protocols packet fields in the recruiting protocol.

Recruit Packet		
Type	**Fields**	**Explanation**
int32	TTL	Time to Live for the Expansion Ring
string	recruit_originator	ID of the robot that requested the recruitment
string	src_hop	ID of the robot that's currently handling the recruit pkt
string	dst_hop	ID of the robot to which the src_hop forwards the recruit pkt
string	dest_recruit_originator	ID of the robot that the recruit_originator requested to be recruited
float64	timestamp	Timestamp of recruit pkt
geometry_msgs/Point	src_position	Current coordinates of the src_hop robot in the map
geometry_msgs/Point	originator_position	Coordinates of the originator robot in the map

5.2 Expanding Ring: Recruiting Request

This strategy consists in forwarding the recruit request at the beginning setting the TTL=1. This allows a propagation of the *recruit request* just on the first ring and on the direct neighbors. If no robots are available to be recruited, after the *recruit request* timeout, a novel request is sent with TTL=2 and so on. This will allow to reach farer robots using intermediate robots to forward the recruit requests. Two possible recruiting requests are possible:

- **Specific Recruit Request:** a robot can recruit another robot that, after accepting the request, will move towards the position indicated by the recruiter.
- **General Recruit Request:** a robot can recruit another robot that is at the base station to recharge. This request will activate a robot that was inactive at the base station.

On the basis of the recruiting requests presented above, two conditions have been considered to be managed:

- The robot exhausts its resources coming back to the base station. In this case the robot requests a specific recruit requests by other active robots.
- Incremental recruit: every pre-fixed amount of time a robot is resumed and it can come back to scan the area in order to reduce the overall tack of map building;

In the recruiting phase, a robot can be recruited if it has enough resources to be recruited. In our case it is considered the battery level to know in advance if, on the basis of the position where the robot should go, it has enough energy to go there. It is preferred the recruit if a robot that is inactive on the base station in order to involve it in the space exploration task. Secondly, the robot will be involved also considering the distance. This means that the robot nearest in terms of base station will be selected. When a robot is terminating its resources, it will go to recharge on the base stations that is closer to the robot.

6 SECURITY EVALUATION IN ROBOT COORDINATION

In this section different security threats scenarios will be considered. In the first scenario a personification attack is considered and then an authentication procedure is proposed as countermeasure.

Then, a second scenario where a specific integrity attack has been considered with the correspondent mitigation countermeasure; the last scenario considered an attack to the confidentiality. Performance metrics considered for the comparison between secure and not secure recruiting strategies are the following: number of exchanged packets, dropped packets and energy consumption.

Another parameter accounted in the performance evaluation has been the cryptography algorithm applied. In our case we considered elliptic curve cryptography applying three different elliptic curves: **secp192r1**, **secp256k1**, **secp384r1** (Shaikh et al., 2017), (De Rango et al.,2020).

6.1 Authentication Attack

In the first scenario, 2, 3 and 4 robots have been considered in the simulation and not security features have been considered. In the network a malicious robot has been accounted and it informs all robots that it already explored the MAP in order to disincentive other robots to explore the unknown area. This attack will determine that each robot will present an incomplete MAP and the overall task will fail. Under this attack robots think that the task is complete and they terminate to explore the area. In Figure 3 and Figure 4 it is possible to see the number of exchanged packets under attack and under a legacy behavior. It is possible to see as the exchanged packets are less among robots under attack because they assume that the overall task has been performed. However, this reduction in the protocol and control overhead leads to an incomplete MAP formation such as it is possible to see in Figure 5.

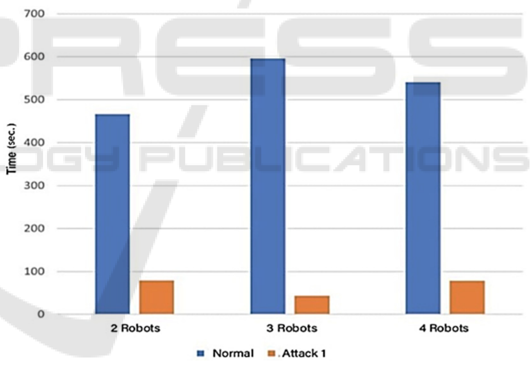


Figure 3: Task execution time for increasing number of robots in legacy conditions or under attack.

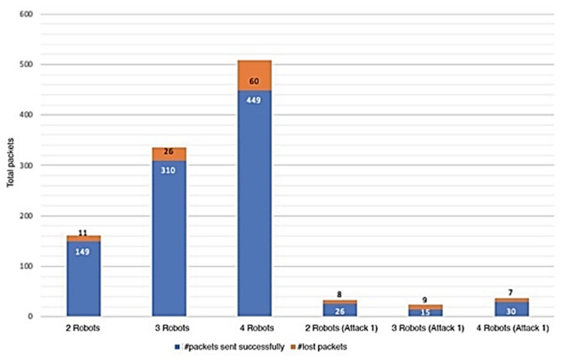


Figure 4: Number of transmitted and lost packets for increasing robot number and under attack 1.

In the following it is shown the number of sent packets. It is shown in Figure 5 the map building under attack condition or in absence of malicious behavior. It is possible to see as on the left side, a correct map building is observed where red, green ad grey colors represent the scanned map by legacy robots. On the contrary, on the right side it is possible to see the map building considering the same simulation time when an attack is performed. In the right figure, the area with oblique lines represents the unexplored areas. This testifies as an attack in this situation can compromise the mission and the task.

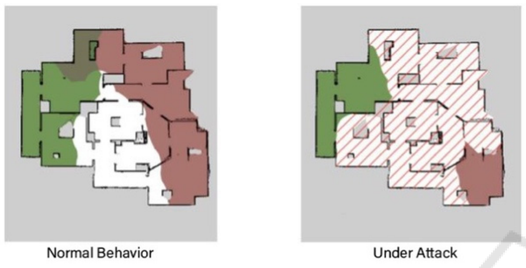


Figure 5: Map building in a scenario with and without attack.

6.2 Mitigation at Scenario 1 Attack

In this case, a considered countermeasure is the application of authentication procedure. In particular, the robot starts an authentication procedure before accepting the recruiting request. In this case the following steps will be performed:

- The RSA has been applied to negotiate the symmetric keys considered to encrypt the recruit requests and messages where info about map is included.
- The message integrity is guaranteed signing the message through ECC cryptography.
- It the symmetric keys are successfully exchanged between two robots, this assures that the key and robots are authenticated.

6.3 Integrity Attack

In the second scenario it is attacked the communication module because a *man in the middle* attack (MiM) is considered. In this case, the MiM robot will try to modify the packet info erasing the info included in the packets. This will determine the sensible reduction of the robot cooperation determining an independent behavior of each robot that will not know the already discovered area by other robots. This attack will produce a useless packet exchange among robots that want to share their info without obtaining the main task objective and an increase in the task completion time will be observed. In Figure 6 it is shown the map to be discovered by each robot and in Figure 7 it is shown the task execution time under attack or under a legacy robots' behavior. The same type of attack can be applied on other component or procedure such as the recruiting phase and requests.

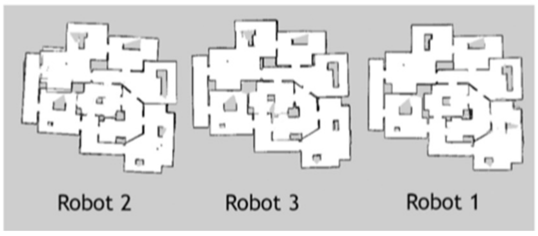


Figure 6: Map representation within each robot.

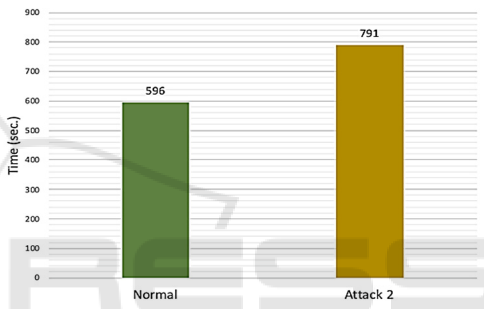


Figure 7: Task execution time for a scenario without attack and another one (Attack 2) with security attack.

6.4 Mitigation at the Scenario 2 Attack

The countermeasure adopted in the second scenario is the adoption of hash function in order to guarantee the integrity. In particular, the SHA256 has been applied to manage the message integrity.

6.5 Eavesdropping Attack

Such as explained in section 1, ROS does not include in its features any security features for messages created to offer integrity and confidentiality. In particular, all topics in the publish/subscribe architecture are public and all robots can see all messages that fall in the communication range. This kind of issue is well-known, in literature, as "Eavesdropping Attack" and it is usually used as a starting point for more dangerous attacks.

6.6 Mitigation at the Scenario 3 Attack

In order to avoid possible attack to the confidentiality all robot communications have been encrypted using a block encryption such as AES-128 (Kousalya and Kumar, 2019) using ephemeral symmetric keys such as explained in section 4. Moreover, in order to mitigate the reflection attack, we managed a couple of keys to manage encryption and decryption for ingoing and outgoing traffic. It is possible to see in as the ROS system, without our extension allows robots to see all info exchanged in the packets. It is possible to see as in the considered system the cryptographic approach is effective hiding all info and providing confidentiality property and complicating the attacker work. In the following it is shown as the ECC curve selected in the proposed approach to digitally sign the packet can affect the execution time.

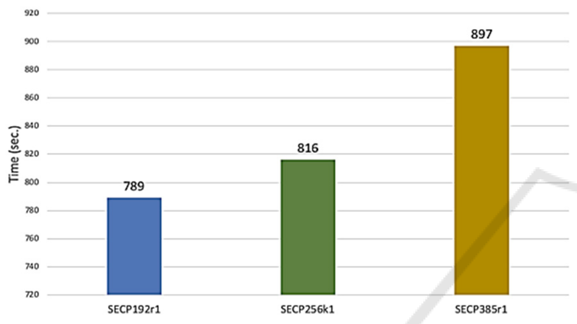


Figure 8: Task execution time under different ECC curves.

7 CONCLUSIONS

The following paper proposed to consider the security aspects in the modeling and performance evaluation of multi-robots systems. The coordination strategy can be degraded or compromised by some possible threats and it is important to include security features in the communication protocols and on-board to robots in order to protect them by possible cyber-attacks. Current simulation tools such as ROS and GAZEBO do not include in the basic features modules able to consider security aspects and how security aspects affect some constrained resources such as battery and communication channel. Some modules have been integrated in ROS and GAZEBO to extend the modeling aspects to security. Some mechanisms to support authentication, integrity and encryption have been implemented. Moreover, some security attacks have been applied to show how mission or task can be compromised. Security features have been introduced to mitigate these attacks and performance evaluation has been evaluated in a legacy or under attack scenario.

REFERENCES

S. Park and G. Lee, "Mapping and localization of cooperative robots by ROS and SLAM in unknown working area," 2017 56th Annual Conf. of the Society of Instrument and Control Engineers of Japan (SICE), Kanazawa, 2017, pp. 858-861.

S. Rivera, S. Lagraa, C. Nita-Rotaru, S. Becker and R. State, "ROS-Defender: SDN-Based Security Policy Enforcement for Robotic Applications," 2019 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2019, pp. 114-119.

M. Mukhandi, et al., "A novel solution for securing robot communications based on the MQTT protocol and ROS," 2019 IEEE/SICE International Symposium on System Integration (SII), Paris, France, 2019, pp. 608-613.

F De Rango, N Palmieri, XS Yang, S Marano, "Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks," in Soft Computing, Vol. 22 (13), 2018, pp.4251-4266.

M. Tropea, N. Palmieri, and F. De Rango, F. "Modeling the Coordination of a Multiple Robots Using Nature Inspired Approaches". In Italian Workshop on Artificial Life and Evolutionary Computation (pp. 124-133). 2019, Springer, Cham.

N Palmieri, XS Yang, F De Rango, S Marano, "Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption," in Neural Computing and Applications, Vol. 31 (1), 2019, pp.263-286.

N Palmieri, XS Yang, F De Rango, AF Santamaria, "Self-adaptive decision-making mechanisms to balance the execution of multiple tasks for a multi-robots team," in Neurocomputing, Vol. 306, 2018, pp.17-36.

S. Gatesichapakorn, J. Takamatsu and M. Ruchanurucks, "ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera," 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 2019, pp. 151-154.

Raje, Sumit. (2020). Evaluation of ROS and Gazebo Simulation Environment using TurtleBot3 robot.

Y. Abdelrasoul, A. B. S. H. Saman and P. Sebastian, "A quantitative study of tuning ROS gmapping parameters and their effect on performing indoor 2D SLAM," 2016 2nd IEEE Int. Symposium on Robotics and Manufacturing Automation, Malaysia, 2016, pp. 1-6.

A. L. Ibáñez, R. Qiu and D. Li, "An implementation of SLAM using ROS and Arduino," 2017 IEEE Intern. Conference on Manipulation, Manufacturing and Measurement on the Nanoscale (3M-NANO), Shanghai, China, 2017, pp. 1-6.

ROS, "About Turtlebot 3", ROS.org. [Online]. Available: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/

J. R. Shaikh, M. Nenova, G. Iliev and Z. Valkova-Jarvis, "Analysis of standard elliptic curves for the implementation of elliptic curve cryptography in resource-constrained E-commerce applications," 2017

IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS), Tel-Aviv, Israel, 2017, pp. 1-4.

F. De Rango, G. Potrino, M. Tropea, and P. Fazio, P. “Energy-aware dynamic Internet of Things security system based on Elliptic Curve Cryptography and Message Queue Telemetry Transport protocol for mitigating Replay attacks”. Pervasive and Mobile Computing, 61, 2020, 101105.

R. Kousalya and G. A. Sathish Kumar, "A Survey of Light-Weight Cryptographic Algorithm for Information Security and Hardware Efficiency In Resource Constrained Devices," in ViTECoN, Vellore, India, 2019, pp. 1-5.