Consensus achievement method for a robotic swarm about the most frequently feature of an environment based on blockchain technology

V I Petrenko, F B Tebueva, S S Ryabtsev, I V Struchkov
North-Caucasus Federal University, Stavropol, Pushkin str., 1, 355017, Russia

E-mail nalfartorn@yandex.ru

Abstract. The technological development and popularity of swarm robotics actualizes the issues of increasing the efficiency and safety of consensus achievement between the swarm elements. Currently, most studies have little or no regard for information security issues when consensus achievement among swarm elements in the presence of robots with arbitrarily malfunctioning or malicious behavior. However, in many practical applications, when exploring the external environment, a swarm of one or more malicious robots may be sufficient to allow the current consensus mechanisms to fail. One of the promising ways to ensure information security in collective decision-making is the use of blockchain technology as a distributed system designed to work in conditions of lack of trust between the parties. The use of distributed ledger technologies, due to their complexity, leads to reducing efficiency in consensus achievement. The aim of the work is to improve the safety of the process of consensus achievement in swarms of robots through the use of blockchain technology in collective design making while maintaining the effectiveness of consensus achievement between robots. Increasing the safety of decision-making by a swarm will increase the stability and the possibilities of practical application of swarm robotic systems for solving problems in aggressive environments.

Keywords: Swarm robotics; Consensus achievement; Collective design making; Information security; Blockchain

1. Introduction
Swarm robotics is a promising direction in the development of intelligent systems and offers promising approaches to solving problems that require covering large areas and large volumes of space, including in unknown or hazardous environments. The spheres of application of swarms are: military objectives, search and monitoring, rescue operations, as well as many other applications.

The literature analysis on the use of swarm [1] showed that for the use of swarm in practice, it is necessary to solve a number of problems, one of which is to ensure information security of consensus achievement [2]. It should be noted that, with the declared high fault tolerance of swarm, most studies of swarm practically do not consider the security problems of collective design making (CDM) arising from the presence of robots with incorrect behavior (malicious robots), which can be a consequence of both a malfunction and a purposeful attacks. However, in many practical applications, one or more of these malicious robots (MRs) may be sufficient to disrupt the entire swarm by imposing bogus or
inappropriate alternatives during consensus achievement in a collective design making process. These features actualize the development of secure algorithms for swarm consensus achievement.

The formulation and classification of information security issues in swarms and comparison of swarms with similar technologies are given in the research [3]. In the article [4], the analysis of algorithms for the functioning of the swarm is supplemented by their stability check from the point of view of information security. In the paper [5], the main threats relevant to the swarm with various principles of construction and operation are analyzed.

One of the promising approaches to solving the problems of protecting the collective design making process from the impact of malicious robot is the use of technologies based on blockchain [6–8]. The advantages of using blockchain are due to the similarity of the basic principles of the organization of swarms with the principles of building distributed storage and data processing systems, such as decentralization, peer-to-peer, scalability. The objectives of using consensus achievement algorithms in blockchain systems are also similar to the objectives of the collective design making in the swarm. Firstly, the blockchain is a distributed design-making system designed to work in conditions of a lack of trust between the parties, which corresponds to the conditions for the functioning of the swarm. Secondly, blockchain systems have built-in mechanisms for maintaining the relevance of information, and, thus, swarm built using such technologies do not need additional nodes to confirm records. Third, the loss of an individual robot should not pose a threat to the consensus achievement process, as well as the loss of a separate node of a decentralized blockchain system.

The practical difficulties of using blockchain in collective design-making in swarm are discussed in papers [2,9], as a rule, they consist of a significant laboriousness of algorithms for consensus achievement and hardware costs in the blockchain and, as a consequence, a decrease in the speed of consensus achievement [10].

In addition, it is worth noting the limitations of the computing power of swarm and individual robots [11,12].

In this connection, the algorithms for consensus achievement in the swarm, despite a similar approach, require their modification for the tasks of the collective design making in the swarm. Thus, currently, there are a number of problems associated with the need to ensure an appropriate level of safety of swarm on the one hand, and with the impossibility of technical implementation of a number of promising solutions in this area, on the other hand. The aim of the paper is to increase the safety of the process of consensus achievement in the swarm through the use of blockchain technology in the collective design making while maintaining the effectiveness of consensus achievement between robots.

This work is devoted to the modification of the previously developed by the authors of the method of reaching a consensus on the attribute of the external environment [13] for swarm through the use of blockchain technology while maintaining the efficiency of the collective design making. The results obtained will make it possible to use the developed method in the future as a basis for the development of a wide range of solutions for ensuring the safety of swarm. For example, to extract a decision chain in order to investigate and analyze the collective design making process to identify harmful effects.

2. Informal statement of the problem

We will consider the issues of collective design making in swarm robotics on the example of the problem of reaching consensus using the classical formulation, as such an approach is used the study [14].

In this case, the purpose of the swarm will be to make a collective design and reach a consensus on the most common color of the mosaic (in all our experiments - white) of the black and white grid.

1. Robots are limited to a square room size $X \times X \text{m}^2$, which is bounded by four walls. The grid consists of $|B|$ black and $|W|$ white tiles. Each tile is $X \times X \text{m}^2$.

The difficulty of the problem can be varied by changing the ratio between the percentage of black tiles and $B$ white tiles $W$ and the size of one tile.

In a relatively simple task, the difference between the percentage of white and black tiles should be large (for example, $B = 34\%, W = 66\% \rightarrow \approx 0.52$), while in a relatively complex problem the difference is small (for example, $B = 48\%, W = 52\% \rightarrow \approx 0.92$).
2. At the end of a successful launch, all individual swarm robots will have the same opinion corresponding to the most common color.

Robots can communicate with each other only if there is no other robot interfering with communication and if the distance between robots is less than $d_n = 50$ sm. This is a constructive limitation in order to simulate the capabilities of the data transmission channel in the swarm, which will facilitate testing with real robots in future studies.

To simulate a swarm, we used the ARGoS software with the ARGoS-Epuck add-on installed [15]. An experimental scene was constructed containing 20 e-puck robots moving on a surface marked with black and white cells and able to perceive the color of the surface under them, Figure 1 (a) [16]. The E-puck robot has a diameter of 7 sm, a wheeled platform with a maximum speed of 16 sm/s, RGB LED lighting, 8 proximity sensors, a sensor for detecting surface color, and a module for local information exchange, consisting of 12 IR transceivers.

Before starting the experiment, robots are randomly placed inside the arena. An example of the initial configuration of an experiment for an arena measuring 2 by 2 meters ($S = 2$), the number of robots equal to 20 ($N = 20$) and the percentage of black cells $P_b = 40$ is shown in Figure 1 (b).

Each robot’s trajectory is a broken line as the robot alternates between straight-ahead movement and rotation in place. Each of the two phases lasts for a random period of time, the value of which obeys an exponential distribution with a given average value. The direction of rotation is also randomly selected. Each robot is equipped with a rangefinder to determine the distance to other robots and obstacles. When an obstacle appears in the field of view, the robot turns around and continues to move in the opposite direction from the obstacle.

The task of robots is to achieve a consensus regarding the prevalence of the number of black or white cells. The key role in the implementation of the method is made by a smart contract, implemented in the “solidity” language, figure 2. The main purpose of the contract is decentralized aggregation and processing of data received from the elements of the swarm.

The contract stores data using a DataUnit structure, which records the number of black and white measurements, the robot’s current opinion of the dominant feature, as well as the number of times the robot has updated information in the decision chain, that is, in the blockchain. Objects of this structure are stored in an associative array, the key of which is the robot’s address in the blockchain.
Figure 2. UML diagram of a smart contract for consensus achievement by swarm elements

To interact with the elements of the swarm, the contract provides 4 functions. The setInfo function adds or updates the data sent by the robot about the measurements of the external environment to the blockchain. In order to ensure the safety of the design-making process, the smart contract ignores "expired" data published more than 25 blocks ago, data from a robot that published more than 50 transactions in one block, as well as data from a robot with a different version of the blockchain.

To perform this function, the robot needs to send 900 finney, which corresponds to 0.9 ether. This restriction will not allow uncontrolled changes to the data stored in the blockchain and thereby protect them from unscrupulous or out-of-order elements of the swarm. The amount of 900 finney was assigned taking into account the fact that the reward for the created block during mining is 1 ether, which will be enough to perform this function, taking into account its price, as well as the commission provided for in the Ethereum network. It is worth noting that we are talking exclusively about the private Ethereum network and the amounts described above are not monetary equivalent.

The getColor function allows the elements of the swarm to find out the sign prevailing in the external environment, taking into account the data placed in the blockchain by all the elements of the swarm. The dominant feature in the external environment is formed as a weighted sum of all measurements stored in the blockchain:

$$Q(a) = \frac{\sum_{i=0}^{N} a_i q_i(a)}{\sum_{i=0}^{N} c_i},$$

where $c_i$ – total number of measurements made by the $i$-robot; $q_i(a)$ – the number of measurements obtained for the feature $a$ by the $i$-robot.

The final result of the function is the number of the attribute of the external environment with the highest score $Q(a)$.

The vote function updates the robot's opinion of the dominant feature in the blockchain. Unlike setInfo, when executing this function, the robot does not need to send additional funds, which allows the swarm elements to quickly maintain its current status.

When the isConsensusAchieved function of the smart contract is called, the robot receives current information about whether the entire swarm has reached a consensus. This decision is based on the entire set of information posted on the blockchain. As part of the experiment, the decision was made based on the opinion published by the rest of the robots about the most common feature of the environment.

Figure 3 shows a block diagram of the behavior algorithm of the swarm element. Before starting the experiment, using the geth program, the local version of the Ethereum blockchain is initialized and launched, in which the smart contract described above is published. During the experiment, the elements of the swarm are connected to this process to publish the data they have collected, as well as receive and process the data collected by other elements of the swarm. In addition, throughout the experiment, each robot is mining the cryptocurrency it needs to send data to the blockchain. This action is not shown in the flowchart as it is performed continuously in the background. The difficulty of opening a new block depends on the computing power of the swarm and is set so that an average of 2 new blocks appear per minute.

After initialization of the blockchain, robots perform a series of actions, shown in Figure 3, until a consensus is achieved between the elements of the swarm about the most common feature of the external
environment. So, by the first action, the robot determines the color of the surface underneath, and stores the result in its memory, and also increases the counter of perfect measurements:

$$q(id, a) = q(id, a) + 1,$$

where $id$ – unique identifier of the robot; $a$ – measurement result (white or black).

The next step is to try to establish a connection and synchronize the blockchain with other elements of the swarm. As part of the experiment, the maximum range of wireless communication between robots was limited to 50 centimeters.

After completing this step, the robot calls the $getColor$ function of the smart contract to check if its own opinion about the most common feature of the environment differs from the prevailing opinion of the swarm, according to its version of the blockchain. In case it differs, the robot initiates the transaction and sends its updated opinion using the vote function of the smart contract.

The next step for the robot is to check if it has enough funds to publish its measurements of the external environment on the blockchain. If so, it initiates a new transaction using the $setInfo$ function of the smart contract. Otherwise, the robot skips this step and moves around the arena. As part of the experiment, the maximum speed of the robot in a straight line was limited to 10 cm / s. The movement algorithm consists of two phases, a phase of rectilinear movement and a phase of rotation in place. During the straight motion phase, the robot moves straight for an average duration of 40 seconds. After this period of time, or when it encounters an obstacle, the robot enters the phase of rotation in place, the average duration of which is 4.5 seconds, the direction of rotation, each time being chosen randomly. After completing the turn, the robot resumes straight motion.

At the last step, the robot calls the $isConsensusAchieved$ function of the smart contract to find out if a consensus has been reached in the swarm. In the event that the function returns a negative result, the robot repeats the cycle again, if the function returns a positive result, the work of the algorithm is considered complete.

### 3. Results

To assess the effectiveness of the used as an analog and blockchain approaches, the following metrics were used:
– probability of completion \( (E_N) \), i.e. the probability of making the best decision, calculated as the proportion of runs
– time of results completion \( (T_N^{correct}) \): this is the number of seconds it takes for the robots in the swarm to have a "white" opinion. The metric is calculated over all experimental runs converging to "white"; chains converging to "black" are not counted.

The experimental results are shown in the figure 4a and 4b.

A group of experiments demonstrates the results of the algorithm when the difficulty of the task changes, i.e., when the ratio of white and black cells on the field changes. To carry out these experiments, the following parameters were used: a field measuring 2 by 2 meters, 20 e-puck randomly placed in the arena before the start of the experiment. The percentage of black cells ranged from 34% to 48% in 2% increments. A series of 50 experiments was performed for each configuration. In Figure 4 (a), the abscissa shows the percentage of black cells placed on the field, and the ordinate shows the probability of making a correct decision about the dominant feature in the environment by all robots at the end of the experiment. It can be seen from the graph that in most cases robots make the correct decision about the dominant feature, but the probability decreases markedly with increasing complexity of the task. So, with 48% black cells, the probability of consensus achievement with the correct result \( E_N = 0.6 \), which is slightly above the threshold level \( E_N = 0.5 \) – the probability of a random correct choice of one of the two signs of the environment.

In Figure 4 (b), the ordinate is the time to consensus achievement in seconds. The graph shows the tendency for the time to reach consensus to increase, as well as its spread with the increasing complexity of the problem.

It is worth noting that the probability of consensus achievement has decreased by 10-15% compared to a similar method, without using the blockchain, described in [16]. This fact may be associated with a decrease in the frequency of information updates in a distributed swarm information environment. In this experiment, this frequency is regulated by two parameters: the reward for the open block and the cost of sending data to the blockchain network.

4. Conclusion
This work makes it possible to develop the approach proposed earlier by the authors [13] of consensus achievement for the further provision of information security, collective design making, swarm based on blockchain in the following directions:

1. Analysis of malicious robot’s behavior in order to identify the most effective strategies for the attacker.
2. Reducing the computational costs of the blockchain for more profitable use in the swarm and for efficient management of the system logging system in the blockchain, which will allow processing information security incidents in the swarm based on the analysis of the collective design making transaction chain.

3. Analysis of vulnerabilities of swarm from blockchain to attacks on swarm and development of approaches to protection against them.

The presented approach can be applied not only in swarm, but also in other similar technologies, the tasks in which may require a decentralized approach to management and consensus achievement.

5. Acknowledgment

The reported study was funded by Russian Ministry of Science (information security), project №10/2020.

References

[1] Bayindir L. A review of swarm robotics tasks // Neurocomputing. 2016. Vol. 172. P. 292–321.

[2] Beskopylny A. Blockchain in Robotic Distributed Multi-Level Systems // Adv. Robot. Mech. Eng. 2018. Vol. 1, № 4. P. 59–61.

[3] Higgins F., Tomlinson A., Martin K.M. Threats to the swarm: Security considerations for swarm robotics // Int. J. Adv. Secur. F., Tomlinson, A., Martin, K. M. (2009). Threat. to swarm Secur. considerations swarm Robot. Int. J. Adv. Secur. 2(2), 288–297. https//pdfs.semanticscholar.or. 2009. Vol. 2, № 2. P. 288–297.

[4] Komarov I.I. et al. Postanovka zadachi obespecheniya informacionnoj bezopasnosti roevy`x robototexnichskix sistem // Sci. Bus. Dev. ways. 2015. Vol. 3, № 45. P. 66–72.

[5] Basan A.S., Basan E.S., Makarevich O.B. Analiz problem obespecheniya informacionnoj bezopasnosti dlya sistem podvizhny`x ob`ektov s ispol`zovaniem besprovodny`x tehnologij // Izvestiya Kabardino-Balkarskogo nauchnogo centra RAN. 2018. Vol. 6, № 86. P. 30–43.

[6] Ferrer E.C. The Blockchain: A New Framework for Robotic Swarm Systems. 2018. P. 1037–1058.

[7] Nguyen T.T., Hatua A., Sung A.H. Blockchain Approach to Solve Collective Decision Making Problems for Swarm Robotics // International Congress on Blockchain and Applications. 2020. P. 118–125.

[8] Strobel, V., Ferrer, E.C., Dorigo M. Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario: Robotics track // Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS. 2018. Vol. 1. P. 541–549.

[9] Odelu V. Blockchain Approach to Solve Collective Decision Making Problems for Swarm Robotics // International Congress on Blockchain and Applications. Springer International Publishing, 2020. Vol. 1010, № January. 1–10 p.

[10] Hamann H. Introduction to Swarm Robotics // Swarm Robotics: A Formal Approach. Cham: Springer International Publishing, 2018. P. 1–32.

[11] Swarm robotic platforms [Electronic resource]. 2019. P. 9–13.

[12] Navarro I., Matía F. An Introduction to Swarm Robotics // ISRN Robot. Hindawi, 2013. Vol. 2013. P. 1–10.

[13] Petrenko VI, Tebueva FB, Ryabtsev SS, Gurchinsky MM, Struchkov IV. Consensus achievement method for a robotic swarm about the most frequently feature of an environment // IOP Conference Series: Materials Science and Engineering. 2020. Vol. 919, № 4.
[14] Valentini G. et al. Collective perception of environmental features in a robot swarm // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, 2016. Vol. 9882 LNCS. P. 65–76.

[15] Pinciroli C. et al. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems // Swarm Intell. 2012. Vol. 6, № 4. P. 271–295.

[16] E-puck robots [Electronic resource]. Access mode: http://www.e-puck.org/ (date of access: 11.11.20)