Decentralized Management Protocol for Wireless Sensor Network in Industrial Internet of Things

Vasily Desnitsky
Laboratory of Computer Security Problems
St. Petersburg Federal Research Center of the Russian Academy of Sciences
St. Petersburg, Russia
desnitsky@comsec.spb.ru

2024 International Ural Conference on Electrical Power Engineering (UralCon) | 979-8-3503-8602-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/URALCON62137.2024.10718902

***Abstract*—The paper comprises construction and evaluation of a decentralized management protocol in a self-organizing decentralized wireless sensor network (WSN) within the framework of the Industrial Internet of Things. The protocol is focused on organizing and supporting the functioning and monitoring of the information security of the network in order to collect, process, store and analyze data from the WSN nodes in conditions of high dynamism and variability of the network structure, the set of the nodes, their physical location in space, the behavior of the nodes and the amount of their available resources. This paper focuses on three proposed basic algorithms that constitute this protocol, their essence and analysis. In particular, it reveals the features of two alternative protocols for forming a multilateral session of WSN nodes, which provides a secure decentralized network operation using a node role model and blockchain as a means to organize distributed secure data storage on WSN nodes. In addition, the proposed protocol is oriented to solve the problems of monitoring the information security of a self-organizing decentralized wireless sensor network and detecting attacks that exploit, particularly, the properties of self-organization and decentralization of the network.**

Keywords—wireless sensor network, decentralized management, protocol, monitoring, industrial Internet of Things

I. INTRODUCTION

At present, wireless sensor networks are becoming more and more widespread, functioning within various industrial cyber-physical infrastructures in transportation, industrial production, power generation, smart city infrastructures, etc. [1]. At the same time, there is a growing need for distributed interaction, information exchange and storage facilities directly integrated with the nodes of such networks.

A typical example of such WSNs are connected drones systems, in which networks spontaneously assembled from individual nodes for specific business tasks are able to improve the efficiency of target processes through coordinated and secure data collection, processing, storage, and analysis while operating in an untrusted environment [2, 3]. Therefore, the development and providing security of protocols for managing such WSNs seems to be particularly important at present [4, 5].

The decentralized management protocol developed in this paper is aimed at ensuring the connectivity of WSN nodes, reliability and continuity of communication functions in the network, taking into account the peculiarities of regular or spontaneous change of the functional roles of the network nodes. In particular, the proposed protocol utilizes an advanced distributed ledger mechanism, the blockchain, which guarantees the immutability of data collected and stored within the network perimeter [6, 7]. The distinctive features of the protocol include the possibility of flexible configuration of the used blockchain mechanism, which provides decentralized duplication of formed blocks across the currently available WSN nodes in automatic mode. In addition to uninterrupted collection, processing, storage and analysis of target WSN data, the proposed protocol also provides the ability to validate data stored on nodes in a distributed manner in order to guarantee data immutability against a potential attacker [8].

The contribution of this paper presents two alternative algorithms to organize a multiparty session for reliable and secure data exchange between WSN nodes. The two algorithms are also evaluated. In addition, an algorithm is proposed to ensure the integrity of application and service data generated at the network nodes by using a lightweight blockchain. The blockchain is deployed on the network nodes and provides distributed secure storage of the most important data with acceptable costs for storing this data. The results obtained in the paper are analyzed in the paper.

The remainder of the paper is organized as follows. Section 2 gives diagrams schematics and description of the three main developed algorithms used in decentralized management protocol. Section 3 provides information about the software implementation of the algorithms and analyzes them. Finally, Section 4 concludes the paper.

II. CONSTRUCTION OF DECENTRALIZED MANAGEMENT PROTOCOL

The developed protocol presents the means of the nodes functioning in a self-organizing decentralized WSN with a specific role model of network nodes. The following main roles are identified, data collector, data processor, data keeper, data and event analyzer [9]. In this section, three main algorithms are presented, which the developed decentralized management protocol in WSN is built on.

A. Handshake-based algorithm for multilateral session establishment

This algorithm is designed to search for WSN nodes that are in the wireless signal coverage area, use a common network PAN ID and “want” to establish or join an existing logical session. Such a network is formed within the application layer

The work is supported by a grant of the Russian Science Foundation, grant number 24-21-00486, https://rscf.ru/project/24-21-00486/.

of communications to solve the tasks of secure data collection, data exchange between nodes and data analysis in the interests of operational management and monitoring of information security of the network [10].

The input to this algorithm is the parameter n, which defines the minimum number of participants (WSN nodes) that are required to initiate a multilateral session and is set by the network organizers. Also the input contains the value of the parameter Δ, which forms a reserve to reduce the probability that due to spontaneously disconnected nodes from the network, the number of functioning nodes will fall below the set value n. Fig. 1 shows the flowchart of this algorithm for the case of a linear WSN fragment with nodes A ↔ B ↔ C in the form of a sequence diagram.

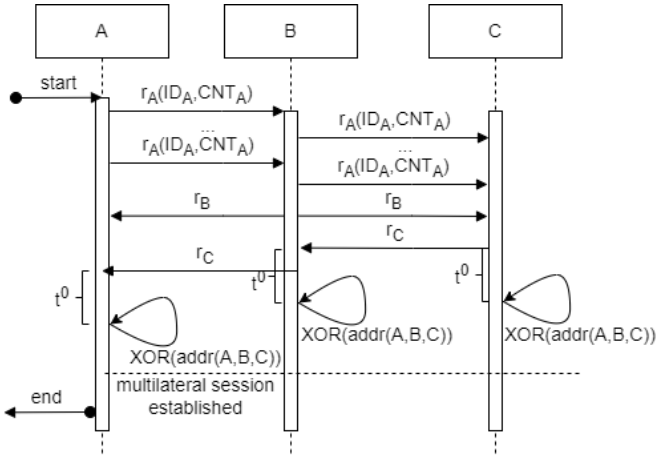

Fig. 1. Diagram of the handshake-based algorithm on an example of a WSN fragment with 3 nodes.

The operation of the main steps of this algorithm is explained below.

- Initially, any node functioning within the current communication and computing environment of nodes and wishing to go into the formed logical session broadcasts its readiness to the network with the help of broadcast commands r_A sent with a specified frequency (e.g., once per second). Such commands are sent as part of the used ZigBee network protocol [11]. Each command contains, first, a unique identifier ID_A of the node and, second, a unique value of the command counter to demonstrate the node's readiness to join the generated multilateral session CNT_A. The former identifier represents the unique physical (MAC) address of the ZigBee module, while the latter one is initialized with a zero value, after which it is incremented by 1 after each such command.
- Next, each node in the network in the ready state rebroadcasts to all of its neighboring nodes the readiness commands it has received. It retransmits it to all neighboring nodes except the node which the command was received by the node from. If some instance of the command to this node has already been received and forwarded to the network, this command is ignored. At the same time, each node collects and constantly updates data on the status of nodes that are currently in the readiness state.
- When a node reaches the information that $(n+\Delta)$ nodes are in the readiness state, including this node, the node sends a broadcast command to create a logical session, and also starts repeating it with a specified frequency (e.g., once per second). In addition, the node waits for the same commands from other nodes that have expressed readiness (i.e., from nodes in its list). This node collects a table of such acknowledgements from all $(n+\Delta)$ nodes. When a node receives acknowledgements from all $(n+\Delta)$ nodes, it stops broadcasting the commands.
- Next, the node waits for a constant time t^O required to deliver all remaining broadcast commands over the network. If during this time interval none of the participants sent their command, it means that all participants of the information exchange, in fact, agreed on the set of the participants, and the session can be started. Otherwise, the waiting time is prolonged by the value of the interval t^O.
- Each node, having information on the physical addresses of each participant in the information exchange of the session being established, calculates a single shared session identifier as a bitwise XOR of the binary representations of all addresses. Note that due to the commutativity of this operation, the order of its application is unimportant, and the resulting value is the same on the sides of all nodes. An optional final step can be a multilateral exchange of this identifier as an additional correctness confirmation for receipt by all participants of the session. The size of the session identifier is equal to the size of the physical address of the node. Note also that the identifier differs depending on the composition of the nodes that decided to organize such a session. The session identifier will uniquely identify the current network session. The session identifier is not subject to secrecy requirements with respect to any remaining WSN nodes.

The distinctive features of this algorithm include the use of the multilateral handshake principle as a means to clarify the composition of the nodes of a multilateral session and to develop a single unique session identifier on which basis the further communication between nodes will take place. At the same time, this algorithm is sufficiently scalable to organize multilateral sessions on a large set of WSN nodes interacting at the network level. In addition, we note that this algorithm does not require significant consumption of communication and computing resources of WSN nodes and therefore can be used, including in ZigBee networks in conditions of low-power microcontrollers and single-board computers, as a hardware-software platform of network nodes. At the same time, the large use of broadcasting messages during the session establishment process, however, creates additional traffic, and which effect may impact negatively the fulfillment of WSN application scenarios.

B. *Algorithm for multilateral session establishment based on node clustering*

This algorithm is an alternative to the previous algorithm for organizing a multilateral session by WSN nodes. The essence of the algorithm is reduced to the sequential pairwise clustering of nodes, when two nodes desiring to enter the organized multilateral session connect with each other, forming a base cluster [12]. The formed cluster then recursively continues to search for other nodes also willing to contact its nodes. Thus, the formed clusters are successively enlarged, each time forming a unique identifier shared by the nodes within the cluster. At the final stage of the session organization process, some cluster is merged with another similar cluster or a single node and they form a single merged cluster with the generation of a single unique session identifier. Fig. 2 exposes a generalized scheme of this algorithm in the form of a sequence diagram on an example of a WSN fragment consisting of four nodes.

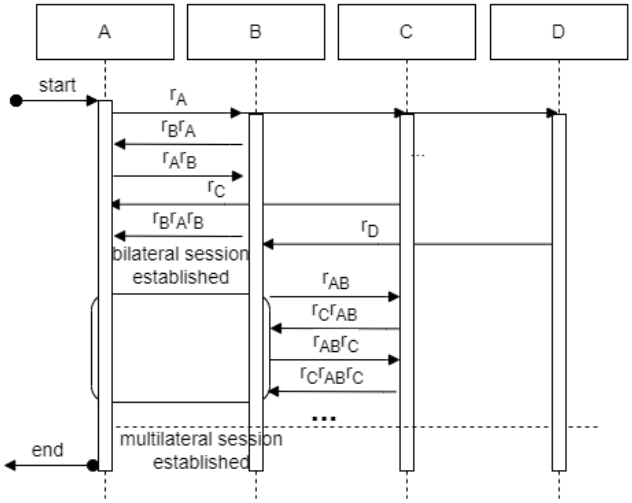

Fig. 2. Schematic diagram of the algorithm for multilateral session establishment based on node clustering.

The operation of the main steps of this algorithm is explained below.

- Each of the nodes wishing to participate in organizing a multilateral session within the WSN sends a broadcast request. For example, node *A* sends a request r_A to all nodes within the wireless signal range that have a given PAN ID value. The identifier r_A includes the physical address of node *A*. In response to this request, node *B* has generated and sent the command $r_B r_A$, thereby confirming its agreement to form a pairwise link with node *A*. To identify which request this response is made to, the response message includes, particularly, a repeat of the identifier r_A as well as an identifier r_B including the physical address of node *B*.
- After receiving the response to the initial request, node *A* selects one of the nodes as the node to form a pairwise link and sends an acknowledgement to node *B* in the form of an inverse message $r_A r_B$ as its acknowledgement. According to this protocol, the selection of the particular node that node *A* accepts as the current partner node, is delegated directly to node *A*. Among the possible candidates, the node whose readiness response comes before the responses from other nodes can be selected. Note also that in the example in Fig. 2, the request r_C from node *C* is ignored, because by the time it arrives, the pairwise link between nodes *A* and *B* is already in the establishment process.
- The final acknowledgement in the form of the $r_B r_A r_B$ command is sent from node *B* to node *A*. After that, the paired session between these two nodes is considered established. After that, within the framework of this protocol, nodes *A* and *B* become a single unit of interaction for establishing paired sessions with other nodes or groups of nodes in the given WSN, thus iteratively forming a multiparty session between all nodes in the given WSN that are in range of the wireless signal, have the same PAN ID and have expressed their willingness to participate in this multiparty session. This algorithm terminates when such a multiparty session is formed.

Unlike the previous algorithm, the distinctive features of this one include the iterative nature of this algorithm, where the clustering process has hierarchical features, i.e. the process of formation and enlargement of node clusters can be extended in time. For example, if at some point in time it was not possible to involve the required $(n+\Delta)$ nodes in the organized session, the session formation process can be put on hold until the nodes with the required PAN ID appear and are ready to enter the session. Therefore, the advantage of using this session organization algorithm is its focus on situations with unstable network coverage, wireless signal interference, complex terrain with obstacles or low power transceiver interfaces. At the same time, the need for pairwise linking of nodes during the establishment of a multilateral session causes a larger number of command forwardings, which can significantly lengthen the actual session establishment time.

C. *Network operation algorithm*

This algorithm determines the functioning of a self-organizing decentralized WSN in terms of ensuring the integrity of data generated at the nodes of the network [13]. Within the framework of the proposed protocol, this algorithm provides opportunities to organize coordinated distributed data storage on the WSN nodes with the guarantee of invariability of the network operation data, including historical data stored on different nodes of the network.

In particular, this algorithm provides proof of the integrity of a data sample by the owner of that data. And such proof can be provided to other nodes in the network that need to verify that this data has not been maliciously modified. Note also that a node may need to prove the integrity of some piece of data with respect to data that is currently stored on that node but was originally created by some other node. The process of proving the immutability of data following its transmission to the destination node is a validation at the destination node by using blockchain technology. Fig. 3 shows a generalized

scheme of this algorithm through using a sequence diagram with an example of three nodes.

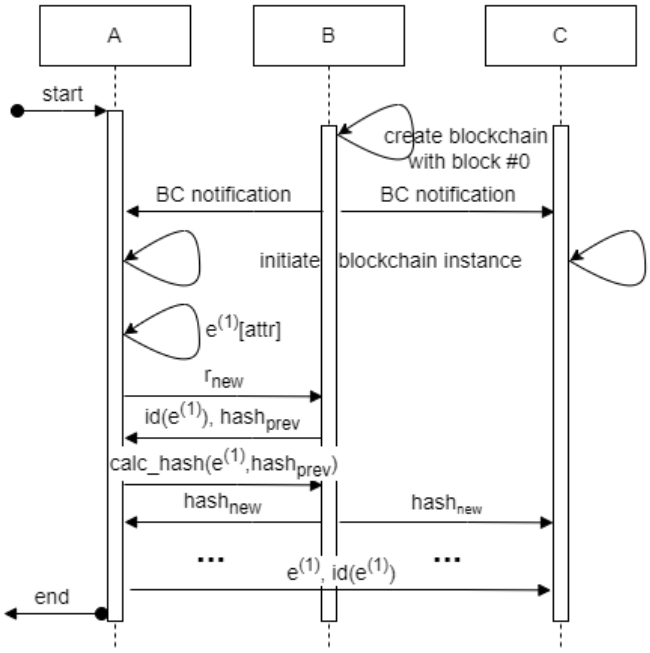

Fig. 3. A diagram for the WSN functioning algorithm

The operation of the main steps of this algorithm is explained below.

- According to the scheme, it is assumed that node *B* has the data keeper role, while nodes *A* and *C* can have any role of WSN nodes, including the role of a network coordinator. Note that node *B* is responsible for organizing the data storage of the WSN nodes. Data is stored on the nodes in a distributed manner, and in general, this node *B* does not store all the available data of the network. First, to the data fragments that need to be stored using the blockchain Node *B* assigns a global block identifier which these data will be addressed to. Second, it controls and synchronizes the order of created blocks. And, third, it sends notifications to all nodes about each new block created by broadcast messages.
- The algorithm starts with the data keeper node creating a new blockchain, adding an initial block with $id = 0$, and sending a notification to the rest of the multilateral session. After that, the other nodes initialize their local copies of the blockchain.
- When an event occurs (e.g., event $e^{(1)}[attr]$ on node *A*, where *attr* is a list of event attributes) or a piece of data that the node possessing it wishes to preserve with the possibility of further provability of its immutability, the event or data is written to the blockchain. In the case of the event $e^{(1)}$, for this purpose node *A* requests keeper *B* to create a new block by using the r_{new} command. In response, the data keeper returns a new sequence number $id(e^{(1)})$ and the hash value of the previous block $hash_{prev}$. With a response message, node *A* sends to node *B* a new hash calculated by hashing the event $e^{(1)}$ and the previous hash $calc\_hash(e^{(1)},hash_{prev})$. Finally, node *B* sends the sent hash to all WSN nodes that have their own copy of the blockchain to update the final state of the blockchain.
- Proving the immutability of the event data $e^{(1)}$ by node *A* to node *C* is done by sending the event $e^{(1)}$ and its sequence number in the blockchain $id(e^{(1)})$. After receiving this data, node *B* finds the block with $id(e^{(1)})+1$ in its copy of the blockchain and takes its hash value, calculates the hash value of event $e^{(1)}$ and compares the two values. If the two values do not match, then the integrity of the received data is broken, and therefore it is not reliable. Otherwise, we can conclude that the attack of illegitimate modification of the data $e^{(1)}$ was not performed. .
- Validation of any piece of data stored on the blockchain, that is verification of the immutability of that data, is performed in a trusted manner within each of the nodes that required such proof. For this purpose, based on alerts from the data keeper node, its instance of the blockchain is created on each node of the WSN. Moreover, the features of such validation include checking the immutability of the data without the need for a node to possess full instances of the data blocks. In other words, a complete block with the data fragment of interest can be stored on one or several WSN nodes, while the remaining nodes of the network store only the value of the global identifier of this data, as well as the hash value of this block.

This algorithm is based on the following hypothesis. In order to verify the correctness of some data fragment during the validation procedure, it is required to compare the correctness of hash values of this data with the corresponding hash values from other blocks. However, in fact, due to the spontaneous nature of the network nodes, the composition of nodes and network topology may change over time. At the same time, a node that stores one or more intermediate blocks with data for the validation period may not be available to the node conducting such validation. Nevertheless, the inaccessibility of the data block is not critical, because during the validation process it is enough to have a reduced copy of the blockchain on this node, which as most of the blocks contains pairs of the form (*id*, *hash*), where *id* is the identifier of the block, i.e. its serial number within the blockchain, *hash* is the hash value of the block. Therefore, despite the dynamic distributed nature of WSN functioning, the proposed algorithm of network functioning in terms of ensuring the invariability of important application and service data on the network nodes.

This algorithm helps to improve the level of information security of the WSN in terms of ensuring the protection of application and service data of nodes in the process of their operation from attacks of illegitimate data modification. Due to the application of blockchain technology, data can be protected by checking hash values of later blocks in the chain of the distributed ledger. Therefore, such protection is appeared to be particularly effective in case of protecting either historical data important for the subsequent operation of WSN nodes and

application services, or current operational data, starting from a time point when the corresponding block specifying this data sample becomes to be not the last one in the blockchain. In other words, the integrity protection of some new piece of data becomes effective from the time point when some other trusted node also saves some of its new data into the blockchain.

The distinctive features of the proposed algorithm of WSN functioning with the use of blockchain technology include the use of lightweight blockchain to organize data protection in conditions when hardware and computing resources of the WSN are insufficient to deploy a full-fledged blockchain with distributed storage of all the data that are recorded in it during the operation of the network. Thus, the proposed algorithm allows us to painlessly delete some of the data recorded in the blockchain, if the node, the owner of this data, concludes that they are unnecessary for the further operation of the network. At the same time, such deletion of data does not cancel the possibility of further validation of the remaining data in the blockchain, located both above and below the blocks without data. This is possible due to the preservation of hash values in blocks whose stored data has been deleted. The limitations of the proposed algorithm include the fact that it can only detect data integrity attacks that were performed outside the verifying node and only after the data has been placed on the blockchain. Limitations of the blockchain used also include the lack of blockchain branching capability. The algorithm can be particularly useful in investigating information security incidents, when there is a need to verify the integrity of historical data, rather than current data stored on WSN devices.

III. Implementation and Discussion

In this paper, a software implementation of the three algorithms presented above is performed by using simulation modeling and Java programming language. The simulated wireless sensor network is essentially represented using a multi-agent system deployed within a JVM virtual machine. Each node of the wireless sensor network is represented using some specially designed operating system process managed using this virtual machine. In this case, the isolation of the processes ensures that the modeled WSN nodes are autonomous and that the nodes' representation of the current state of other nodes in the network is significantly limited. Also, the decentralized nature of the modeled WSN is ensured by the possibility of creating, deleting and managing an unlimited number of operating system processes. The initializing configuration file with initial modeling parameters, including, in particular, the initial number and composition of nodes of the modeled network, characteristics of software and hardware nodes, etc., is used for the modeling. A hierarchical structure of Java classes with inheritance for different roles of wireless sensor network nodes is applied. Technological flexibility of such structure allows us to organize a sufficiently safe and reliable transfer of roles between the nodes of WSN with minimal computational costs by serialization of the current state of the corresponding class instance, as well as seamless transfer to a new node. After that, by deserializing this instance, the destination node is found to be capable of further use of this role in the network. In the modeling process, this allows for a single interface to access class objects of the node roles and manipulate these objects in a unified manner.

Below are the results of analytical calculation and comparison of communication complexity of the proposed algorithms for establishing a multilateral session based on handshake (A_H) and based on node clustering (A_{CL}) [14, 15]. This index determines the network load on the WSN communication channels as a whole during multilateral session establishment. Let c denotes the number of WSN nodes participating in the formation of a multilateral session. In the calculations, c is always greater than 1. The *Compl* complexity index is computed by the number of message forwardings directly related to the process of establishing the multilateral session. When setting *Compl* values, message instances forwarded between two specific addressees, a sender node and a receiver node, are considered. In general, due to the non-predeterministic nature of the WSN network topology and the possible presence of indirect forwarding between two addressees (i.e. multi-hop forwarding) each forwarding is counted as one forwarding, regardless of the number of intermediate nodes involved in the forwarding process. Therefore, to learn the real number of hops of such messages in the network in the resulting estimates of the algorithm's communication complexity, it is necessary to introduce additional coefficients based on the average number of hops per forwarding of one message between the sender node and the receiver one.

As an assumption, we take a fact that in the modeling both analyzed algorithms are operated in conditions of starting activity of all nodes and their readiness to organize a multiparty session. Also, to make the evaluation more correct, broadcast messages are considered as similar corresponding sets of unidirectional messages sent to all nodes of the network. In addition, the evaluation is performed in conditions of a single information exchange without any failures, delays, errors and omissions of reactions to the received messages. In our opinion, these assumptions simplify the evaluation procedure of the algorithms, but do not have a significant impact on the resulting evaluations. According to the scheme of the algorithm A_H shown in Fig. 1, its complexity is calculated as

$$Compl(A_H) = 2 \cdot c \cdot (c-1) \tag{1}$$

and therefore the complexity of the algorithm is quadratic and in the asymptotic expression it is assumed to be $O(c^2)$. According to the scheme in Fig. 2, the complexity of the A_{CL} algorithm is calculated as

$$Compl(A_{CL}) = 2 \cdot c^2 + 2 \cdot c - 4 \tag{2}$$

as a result, in asymptotic expression it also turns out to be equal to $O(c^2)$. Thus, despite the similar limiting nature of the communication costs, nevertheless, when there are a constrained number of nodes involved in establishing a multilateral session in conditions of communication resource lack or insufficient stability of wireless channels, the use of the A_H algorithm turns out to be more preferable.

In addition to the theoretical justification of correctness and performance verification of the algorithms proposed, conducted using modeling of the WSN in order to fully utilize the algorithms in practice, it is needed to perform a comprehensive analysis, testing and validation of these algorithms. Future work should plan to empirically test the robustness of the proposed algorithms with respect to non-malicious errors and failures that may occur in the process of establishing a multilateral session. As examples of such tests, we highlight the following: sudden loss of communication with one of the nodes during the establishment of a multilateral session and loss of one or more acknowledgments during the establishment process. In general, such non-malicious errors and failures may arise mainly as a consequence of communication problems at the network layer. However, such situations can also result from deliberate legitimate node actions, including spontaneous ones. For example, a node may suddenly need to disconnect from the network to perform maintenance or reconfigure its equipment in a service mode. Also, a node may leave the range of the wireless signal due to the peculiarities of the business tasks imposed on it. Alternatively, a node may also need to change its PAN ID and, if necessary, join another WSN operating in parallel.

In addition, the developed algorithms should be additionally tested for the actual absence of deadlock situations related to the synchronization of actions of the participants in the information exchange. At the level of theoretical analysis, the proposed algorithms are analytically tested for the absence of such situations, but empirical evaluation will also allow us to confirm the correctness of program implementations of these algorithms. In particular, it is advisable to apply a method of model checking, which allows us to make a directed automated search of various input data with playing various variants of chains of actions of WSN nodes [16]. It should be noted also that the expediency of using the method of testing on the model is conditioned in particular by a fact that the communication process, including the involvement of nodes located in a previously unknown network topology and not synchronized with each other in time form a significant non-determinism of the network management process. Therefore, without playing a significant number of test scenarios with the use of automation tools would be difficult and not very efficient in terms of the result of such testing and the resources involved.

IV. Conclusion

The paper proposes two algorithms for organizing multi-party sessions in self-organizing WSNs with role-based control, as well as an algorithm for secure operation of WSNs in terms of ensuring data integrity of network nodes. Thus the three algorithms form a protocol for decentralized control of WSNs. As further steps in this direction it is supposed to investigate possible vulnerabilities and attacks of this protocol, as well as the construction of data sets of operation logs on the example of a specific wireless sensor network. Such data sets will be intended for a development of software components to detect crucial types of attacks on such networks.

References

[1] S. Kaur and S. Sharma, “Role of the internet of things in smart cities: A review,” 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2023, pp. 686–689.

[2] H. Singh and D. Verma, “Approaches for data analysis in WSN,” 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2022, pp. 521–527.

[3] A. M. Eltahlawy and M. A. Azer, “Using blockchain technology for the internet of vehicles,” 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2021, pp. 54–61.

[4] M. Sharma, F. Gebali, H. Elmiligi, and M. Rahman, “network security evaluation scheme for WSN in cyber-physical systems,” 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2018, pp. 1145–1151.

[5] S. Ji, Q. Pei, Y. Zeng, C. Yang, and S. -p. Bu, “An automated black-box testing approach for WSN security protocols,” 2011 Seventh International Conference on Computational Intelligence and Security, Sanya, China, 2011, pp. 693–697.

[6] T. Singh, R. Vaid, and A. Sharma, “Security Issues in Blockchain Integrated WSN: Challenges and Concerns,” 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2022, pp. 1–5.

[7] N. Badri, L. Nasraoui, and L. A. Saidane, “Blockchain for WSN and IoT applications,” 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Hammamet, Tunisia, 2022, pp. 543–548.

[8] V. Aleksieva, H. Valchanov, A. Haka, and D. Dinev, “Model of controlled environment based on blockchain and IoT,” 2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES), Plovdiv, Bulgaria, 2023, pp. 1–4.

[9] V. Desnitsky and A. Meleshko, “Modeling and analysis of secure blockchain-driven self-organized decentralized wireless sensor networks for attack detection,” 2024 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2024 (paper accepted for publication).

[10] K. M. Harsha and D. James, “A novel approach to aggregate and secure data in wireless sensor networks,” 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 1665–1670.

[11] W. Wang, G. He, and J. Wan, “Research on Zigbee wireless communication technology,” 2011 International Conference on Electrical and Control Engineering, Yichang, China, 2011, pp. 1245–1249.

[12] Z. Wang, Q. Wen, Y. Sun, and H. Zhang, “A fault detection scheme based on self-clustering nodes sets for wireless sensor networks,” 2012 IEEE 12th International Conference on Computer and Information Technology, Chengdu, China, 2012, pp. 921–925.

[13] D. Singh and Gaganpreet, “Using Trust-Based Data Integrity Verification and Energy-Aware Hierarchical Data Aggregation for WSN,” 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2022, pp. 528–533.

[14] T. Roughgarden, “Communication Complexity (for Algorithm Designers),” Foundations and Trends in Theoretical Computer Science Journal, 2016, vol. 11, Number 3–4, pp. 217–404.

[15] E. Kushilevitz, “Communication complexity,” Advances in Computers, Elsevier, vol. 44, 1997, pp. 331–360.

[16] W. Chen and W. Xiao, “Model checking and analyzing the security protocol for wireless sensor networks,” Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, Harbin, China, 2011, pp. 4093–4096.