Supporting connectivity of VANET/MANET network nodes and elastic software-configurable security services using blockchain with floating genesis block

Alexey Busygin\textsuperscript{1*}, Maxim Kalinin\textsuperscript{1}, and Artem Konoplev\textsuperscript{1}

\textsuperscript{1}Peter the Great St. Petersburg Polytechnic University, Institute of Computer Science and Technologies, Department Information Security of Computer Systems, 195251 Polytehnicheskaya st. 29, Russian Federation

Abstract. This paper considers the tasks of supporting the connectivity of nodes in communication networks of unmanned transport (VANET/MANET-networks). High dynamics, decentralization and absence of hierarchy in the networks of this type actualize the task of supporting the connectivity of nodes with software-configurable security services, providing the network protection. It is offered to use a Blockchain technology based system for VANET/MANET network topology and authentication data distribution and storage. The issue of unlimited blockchain growth preventing this method from being implemented in VANET/MANET networks is considered. The existing solutions of this issue are analyzed and drawbacks are identified. A notion of blockchain with floating genesis block is introduced and its advantages over similar ideas are demonstrated thus allowing it to be used to resolve the issue of continuously growing blockchain within the systems with staling transactions as a whole and in VANET/MANET networks in particular.

1 Introduction

The development of M2M (machine-to-machine) telecommunications, where wireless device systems transfer information from one device to another as well as informatization and cybernetization of vehicles has allowed network technologies to penetrate into the transport vehicle sphere and developed a new type of communication vehicular systems – VANET (vehicular ad hoc networks, in other words, peer-to-peer vehicular networks).

Along with that the mobility of VANET/MANET networks determines high dynamics in changes of network topology, its uncertain structure, unclear network perimeter, because

\textsuperscript{a}With financial support from the Ministry of education and science of the Russian Federation as part of the Federal target program "Research and development of priority areas for Russia's research and process complex for 2014-2020" (Agreement No. 14.578.21.0224 dated 03.10.2016, unique identifier of this agreement RFMEFI57816X0224)
\textsuperscript{*}Corresponding author: a.busygin@ibks.spbstu.ru

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).
a high mobility of all interaction process participants is added to rerouting of links due to their breaks and enabling/disabling of nodes. In this connection, it is extremely important to maintain the connectivity of nodes with software-configurable security services that ensure their protection.

For the networks of this type there are typical information security threats intended to disrupt the node connectivity such as routing attacks (forging, modification, blocking transmission of routing information, Hello flood, Sinkhole), selective transmission of packages (Black hole) and virtualization of network segments (Wormhole, Sybil) [1]. At the present moment there is a number of general protection methods offered against the abovementioned threats, for example in papers [1, 2]. However, for successful application of these methods it is necessary to know the network topology and have the node identification and authentication system in place. Meeting these requirements for self-organizing networks is a rather difficult task. In paper [3] a secure method for network topology information and authentication data distribution and storage using the Blockchain technology is offered. The issue of using blockchain for VANET/MANET protection consists in a continuous increase in this data structure size resulting in quick depletion of disc space on the vehicular system networks nodes. Moreover, the time required to add new nodes into an unmanned vehicle network increases significantly. This paper offers a solution to eliminate these drawbacks allowing connectivity of VANET/MANET nodes with program-configurable security services in conditions of directed cyberattacks.

2 Review of literature

S. Nakamoto has offered a method for reducing the blockchain data being stored on fully functional nodes by pruning the old transactions, which are not needed for adding and verification of new blocks [4]. The disadvantage of this method is the need to load and verify the entire blockchain when adding new fully functional nodes. Without access to the full list of transactions being stored in blockchain these fully functional nodes cannot verify the current state of blockchain system.

Another approach is to reduce the volume that is taken up by blockchain data by removing excessive data from the blocks, for example, by excluding some metadata and etc [5]. In a similar way in paper [6] it is offered to use the public keys in compressed form to cut down the size of transactions. This approach only slows down the rate of blockchain growth, but does not resolve the issue, which becomes critical in case of high transaction rate within the system.

Blockchain can also be used as a mechanism of synchronization between the nodes changing the system state[7, 8]. At a first step a transaction signaling that the time window starts is recorded to blockchain, during this period the system state is changed. During this time the node exchange messages changing the system state, but these are not saved in blockchain (these messages are called off-chain transactions). When this time window is completed, a transaction with total result of all off-chain transactions execution is recorded to blockchain. This approach only slows down the blockchain growth, but does not resolve the issue as a whole.

Another approach [9-11] is known to be used in database control systems – horizontal scaling of data storage (sharding). In this approach it is proposed to divide data stored in blockchain between several nodes. In this situation the blockchain size is not reduced and the issue of its growth is not solved. Another solution with similar features consists in storing in blockchain data hashes only. The data itself is shared between the distributed hash table nodes (Fig. 1) [12].
a high mobility of all interaction process participants is added to rerouting of links due to their breaks and enabling/disabling of nodes. In this connection, it is extremely important to maintain the connectivity of nodes with software-configurable security services that ensure their protection.

For the networks of this type there are typical information security threats intended to disrupt the node connectivity such as routing attacks (forging, modification, blocking transmission of routing information, Hello flood, Sinkhole), selective transmission of packages (Black hole) and virtualization of network segments (Wormhole, Sybil) [1]. At the present moment there is a number of general protection methods offered against the abovementioned threats, for example in papers [1, 2]. However, for successful application of these methods it is necessary to know the network topology and have the node identification and authentication system in place. Meeting these requirements for self-organizing networks is a rather difficult task. In paper [3] a secure method for network topology information and authentication data distribution and storage using the Blockchain technology is offered. The issue of using blockchain for VANET/MANET protection consists in a continuous increase in this data structure size resulting in quick depletion of disc space on the vehicular system networks nodes. Moreover, the time required to add new nodes into a unmanned vehiclenetwork increases significantly. This paper offers a solution to eliminate these drawbacks allowing connectivity of VANET/MANET nodes with program-configurable security services in conditions of directed cyberattacks.

2 Review of literature

S. Nakamoto has offered a method for reducing the blockchain data being stored on fully functional nodes by pruning the old transactions, which are not needed for adding and verification of new blocks [4]. The disadvantage of this method is the need to load and verify the entire blockchain when adding new fully functional nodes. Without access to the full list of transactions being stored in blockchain these fully functional nodes cannot verify the current state of blockchain system.

Another approach is to reduce the volume that is taken up by blockchain data by removing excessive data from the blocks, for example, by excluding some metadata and etc [5]. In a similar way in paper [6] it is offered to use the public keys in compressed form to cut down the size of transactions. This approach only slows down the rate of blockchain growth, but does not resolve the issue, which becomes critical in case of high transaction rate within the system.

Blockchain can also be used as a mechanism of synchronization between the nodes changing the system state [7, 8]. At a first step a transaction signaling that the time window starts is recorded to blockchain, during this period the system state is changed. During this time the node exchange messages changing the system state, but these are not saved in blockchain (these messages are called off-chain transactions). When this time window is completed, a transaction with total result of all off-chain transactions execution is recorded to blockchain. This approach only slows down the blockchain growth, but does not resolve the issue as a whole.

Another approach [9 -11] is known to be used in database control systems – horizontal scaling of data storage (sharding). In this approach it is proposed to divide data stored in blockchain between several nodes. In this situation the blockchain size is not reduced and the issue of its growth is not solved. Another solution with similar features consists in storing in blockchain data hashes only. The data itself is shared between the distributed hash table nodes (Fig. 1) [12].

![Fig. 1. Data storage diagram in distributed hash-table outside of the blockchain.](image)

3 Using blockchain with a floating genesisblock

Let us consider the basic model of the blockchain. The blockchain is a ordered list of transactions changing the value of a set of variables from some data domain. For clarity let us consider the changed values of one variable \( s \) (Fig. 2).
The first transaction (transaction $tx_2$ of the block $blk_0$) sets the initial value of the variable $s$ to be equal to $s_0$. Each next transaction (for example transaction $tx_i$ of the block $blk_i$) changes the variable value relative to its last value. It is possible to obtain the current value of the variable by implementing all transactions from the list. In other words, by summing up all changes made by every transaction:

$$s = s_0 + \sum_{i=1}^{n} \Delta s_i$$  \hspace{1cm} (1)$$

The issue of this approach consists in uploading and processing the entire transaction list, whose size continuously increases. In order to resolve this issue, the authors of this article propose to fix (in Fig. 3 every $i = 2$ of blocks) the current value of the variable $s$ on a regular basis in a special block. Let us call it a fixing block.

This fixing block does not store the changed values of the variable, but its current values $s^{(k)}_0$ at the moment of recording block generation, which is similar to the block storing initial value of the variable $s_0$ in the base model.
The first transaction (transaction 2\text{tx} of the block 0\text{blk}) sets the initial value of the variable \( s_0 \) to be equal to 0. Each next transaction (for example transaction 2\text{tx} of the block 2\text{blk}) changes the variable value relative to its last value. It is possible to obtain the current value of variable by implementing all transactions from the list. In other words, by summing up all changes made by every transaction:

\[
 s_0^{(k)} = s_0^{(k-1)} + \sum_{i=1}^{n_{k-1}} \Delta s_i^{(k-1)}
\]

\( n_{k-1} \) is the quantity of changes in the variable value between recording blocks \( k-1 \) and \( k \).

Each fixingblock stores all required information about initial status of the system, therefore it can be used as a new genesisblock, which makes it possible to prune all the blocks that precede it. The fixingblock to be confirmed by a sufficient number of further blocks is selected as a new genesisblock. The genesisblock is not fixed and "floats further" with addition of new blocks.

\[
 s_{0}^{(k)} = s_{0}^{(k-1)} + \sum_{i=1}^{n_{k-1}} \Delta s_{i}^{(k-1)}
\]

\( m \) is the number of recording blocks after genesis block storing value \( s_0^{(k)} \).

\( n_i \) is the number of changed variable values between fixing blocks \( i \) and \( i+1 \).

This proposed modification of the blockchain is vulnerable to an attack on new fully functional nodes and or those nodes that lost connection to the network for a long time.

\[
 s = s_0^{(k)} + \sum_{j=k}^{m} \sum_{j=1}^{n_j} \Delta s_j^{(i)}
\]
(more than $t$ blocks). While attacking the malicious node generates a fake blockchain beforehand and tries to send it on new nodes as a legitimate one.

An approach involving the blockchain download from trusted fully functional nodes is proposed for protection against this attack. In this case, the floating genesis block will be protected by a digital signature of this trusted node. This approach allows avoiding storage of the block headers preceding the floating genesis block by introducing a trust model and decreasing the system decentralization.

Table 1 gives the results of comparative analysis of existing methods for solving the blockchain growth issue with modification proposed in this paper. The number of variables, whose values are recorded in the blockchain as well as the transaction rate are assumed to be constant during this analysis.

**Table 1.** Comparative analysis of methods for continuously increasing of the blockchain.

| Method                                      | Necessity to download and verify of all blockchain transactions during startup of new fully functional node | Blockchain size by time | Possible local access to blockchain |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------|-------------------------|-------------------------------------|
| S. Nakomoto’s method                        | Yes                                                                                                      | $O(t)$                  | Yes                                 |
| Reduced size of serialized data             | Yes                                                                                                      | $O(t)$                  | Yes                                 |
| Off-chain transactions                      | Yes                                                                                                      | $O(t)$                  | Yes                                 |
| Sharding                                    | Yes                                                                                                      | $O(t)$                  | No                                  |
| Distributed hashtable                       | Yes                                                                                                      | $O(t)$                  | No                                  |
| Blockchain with floating genesis-block      | No                                                                                                       | $O(1)$                  | Yes                                 |

This comparative analysis shows that the blockchain with floating genesis block in distinction from earlier proposed fully functional nodes is worth considering. Considering that blocks are downloaded from trusted nodes this modification allows resolving the blockchain growth issue completely. This leads to reduced time required to add new nodes to the blockchain. However, the blockchain with floating genesis block is applicable only to the cases when stale transactions can be deleted. The routing data of VANET/MANET networks features this property. At the end of a significant time interval the information about VANET/MANET networks and links between them lose its relevancy, which makes it possible to use the blockchain with floating genesis block to protect the connectivity of networks of this type.

It should be noted that the solution proposed can be used jointly with other methods considered in this article: reducing the blockchain serialized data, off-chain transactions and sharding.

**4 Conclusion**

The authors of this article have proposed a blockchain modification based on a floating genesis block making it possible to resolve issues caused by an unlimited growth of blockchain and apply secure routing protocols based on the Blockchain technology. Secure routing in its turn ensures that the VANET/MANET network nodes connection with
software defined security services as well as control the system as a whole.

References

1. C. Karlf, D. Wagner, Proc. 1st IEEE International Workshop on Sensor Network Protocols and Applications (2003)
2. W. Wang, B. Bhargava, M. Linderman, DNCMS (2009)
3. A.Busygin, A. Konoplev, Proc. of the 26th Research and technical conference on Methods and technical information security applications (2017)
4. S. Nakamoto, Bitcoin: A peer-to-Peer Electronic Cash System [online], Available at:https://bitcoin.org/bitcoin.pdf (2009)
5. ScaleChainblockchain compression[online], Available at:https://github.com/ScaleChain/calechain/blob/master/data/docs/blockchain-compression.md(2016)
6. D. Bozhko, A. Troshichev, GOST N Blockchain: Compressed Signature and Public Key Recovery with GOST R 34.10-2012, Zero Nights [online], Available at: https://2017.zeronights.org/wp-content/uploads/materials/ZN17_Troshichev_GOST_COMPRESSED_SIGNATURE.pdf (2017)
7. J. Poon, T. Dryja, The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments [online], Available at: https://lightning.network/lightning-network-paper.pdf (2016)
8. Raiden specification [online], Available at: https://raiden-network.readthedocs.io/en/stable/spec.html (2017)
9. L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, Proc. of the 2016 ACM SIGSAC Conference on Computer and Communications Security (2016)
10. A.E. Gencer, R. Renesse, E.G. Sirer, International Conference on Financial Cryptography and Data Security (2017)
11. Etheriumsharding specification [online], Available at: https://github.com/ethereum/sharding/blob/develop/docs/doc.md(2017)
12. M. Ali, J. Nelson, R. Shea, M.J. Freedman, USENIX ATC’16 (2016)