A Scalable Approach of Practical Byzantine Fault Tolerance Algorithms for IoMT Blockchains

Filippos Pelekoudas-Oikonomou
Instituto de Telecomunicações
Aveiro, Portugal
Faculty of Engineering and Science, University of Greenwich
Chatham Maritime, UK
f.pelekoudas@av.it.pt

Georgios Zachos
Instituto de Telecomunicações
Aveiro, Portugal
Faculty of Engineering and Science, University of Greenwich
Chatham Maritime, UK
g.zachos@av.it.pt

Georgios Mantas
Instituto de Telecomunicações
Aveiro, Portugal
Faculty of Engineering and Science, University of Greenwich
Chatham Maritime, UK
gimantas@av.it.pt

Jose Ribeiro
Instituto de Telecomunicações
Aveiro, Portugal
jcarlosvgr@av.it.pt

Joaquim Manuel C.S. Bastos
Instituto de Telecomunicações
Aveiro, Portugal
jbastos@av.it.pt

Jonathan Rodriguez
Instituto de Telecomunicações
Aveiro, Portugal
Faculty of Computing, Engineering and Science, University of South Wales
Pontypridd, UK
jonathan@av.it.pt

***Abstract*—Blockchain-based solutions for Internet of Things (IoT) networks constitutes a current trend in cybersecurity and brings significant benefits into current centralized IoT-based health monitoring systems by addressing security challenges. Complex and power intense blockchain solutions do not perform satisfactory in the resource-constrained IoT, and especially Internet of Medical Things (IoMT), devices of these systems due to the latter's limited processing power, storage capacity, and battery life. Therefore, in this paper, we propose a scalable Practical Byzantine Fault Tolerance (PBFT) consensus algorithm for IoMT blockchains to: i) enhance scalability in IoMT blockchains, ii) reduce communication overhead, iii) enhance security while reducing the computational cost for suitability to the resource constraint nature of IoMT devices, iv) facilitate decentralized accountability, and v) eliminate single point of failure.**

Keywords— IoMT, Security, Blockchain, PBFT

I. INTRODUCTION

The Internet of Things (IoT) has arose and managed to grow significantly in recent years, bringing substantial benefits to the healthcare industry by transforming it and introducing the Internet of Medical Things (IoMT), in which medical devices are linked in a way that anyone can have access from anywhere and at any time [1], [2]. By enabling IoMT-based healthcare monitoring systems that deliver tailored and user-centric healthcare services, while overcoming constraints such as time and location, the evolution and emergence of IoMT can play a significant role in improving citizens' quality of life [3], [4]. However, the wide variety of communication technologies (e.g., WLANs, Bluetooth, Zigbee) and types of IoMT devices (e.g., bio sensors, actuators) used in IoMT-based healthcare monitoring systems, as well as the fact that the transmission of personal and confidential healthcare information (e.g., patient's personal details and vital signs) between patients and healthcare providers is done via the Internet, raise many security concerns [5]–[8].

As a result, novel security mechanisms are urgently needed to manage the critical security concerns of IoMT edge networks in an effective and timely manner before they reach their full potential in the healthcare industry [9]. Inside this frame of reference, the industry and research community have identified blockchain technology as a disruptive technology that can be integrated into novel security solutions for IoMT edge networks, as it can support to: a) secure IoMT devices; and b) prevent unauthorized access during data transmission (i.e., tamper proof transmission of medical data) [3]. In spite of this, resource-constrained IoMT nodes (e.g., medical sensors, IoMT Gateways), which are the primary components of IoMT-based health monitoring systems [10], cannot afford the high resource requirements, in terms of computational cost and energy consumption of Proof-of-Work (PoW) consensus algorithms which are the most popular consensus algorithms for blockchain mechanisms in the industry [11]. On the other hand, lightweight algorithms have been proposed based on Practical Byzantine Fault Tolerance (PBFT) that implement a vote system to reach consensus inside a distributed network, rather than calculate a computationally intense puzzle [3]. Although PBFT is a viable solution for a blockchain network consisting of IoMT Gateways that play the role of the blockchain nodes, it comes with defects, such as poor scalability and high communication overhead due to the amount of message exchanges between the nodes of the network [3]. Therefore, there is an urgent need for more scalable approaches of PBFT algorithms in order to keep the benefits of a lightweight consensus mechanism for IoMT blockchain and address the pressing challenge of scalability in an effective and efficient manner.

Towards this direction, the focus of our research work is on the modification of the PBFT consensus algorithm leveraging a clustering algorithm in order to: i) enhance scalability in IoMT blockchains, ii) reduce communication overhead, iii) enhance security while reducing the

This research work was sponsored in part by the NATO Science for Peace and Security Programme under grant SPS G5797.

computational cost for suitability to the resource constraint nature of IoMT Gateways, iv) facilitate decentralized accountability, and iv) eliminate single point of failure.

Following the introduction, this paper is organized as follows. In section II, we revise variants of PBFT algorithm that exist in literature that focus on the scalability issue. In section III an overview of the Practical Byzantine Fault Tolerance algorithm is presented. In section IV we introduce our scalable PBFT-based approach for IoMT blockchains. In section V the simulation of the proposed architecture as well as the simulation results are presented. Finally, section VI concludes this paper.

II. Related Work

In this section, we provide an overview of variants of PBFT algorithm existing in literature that address the issue of scalability.

Hao *et al.* [12] propose a Dynamic PBFT variant which is a scalable protocol derived from PBFT and inherits many of its benefits. This protocol offers the same liveness and security as PBFT. This protocol is dynamic because it permits replicas and nodes to join or leave the consensus network without any downtime, while providing means for removing malicious and long-downtime nodes to improve the systems' resilience while including the concept of Participation Degree to determine whether a node is sufficiently active or not.

Gao *et al.* [13] propose T-PBFT, a novel multistage consensus algorithm that derives from the original PBFT, by leveraging EigenTrust model to create trustworthy consensus groups and decrease the number of consensus nodes in order to enhance efficiency. The proposed algorithm measures node trust based on the transactions between nodes in order to identify the highest-quality nodes, in terms of trust, in the network for constructing a consensus group. Authors propose the replacement of a primary node with a primary group with higher trust values to decrease the likelihood of view change process. Additionally, authors propose the strengthen of the primary group through group signature and mutual monitoring.

Fan *et al.* [14] are focused on the issue of digital signatures that are used in PBFT to validate the authenticity of messages across the various phases of the algorithm. To resolve this issue, authors present an efficient short-lived signature based PBFT variation that uses short-length cryptographic keys to sign and verify messages in PBFT for a limited time period and blockchain-assisted key distribution mechanisms to regularly update the keys. Additionally, authors provide effective strategies for speeding the software implementation of the BLS threshold signature scheme [15].

Veronese *et al.* [16] propose two asynchronous Byzantine fault-tolerant state machine replication (BFT) algorithms that outperform their predecessors in metrics such as: i) number of replicas, ii) trusted service simplicity and iii) number of communication steps, and enhance scalability. They modify the algorithm in order to utilise $2f+1$ replicas instead of the usual $3f+1$, where f is the maximum number of faulty or malicious nodes of the network, they create a more basic trusted service upon which this reduction of replicas is based, making a verified implementation easy and the execution of the algorithm requires the minimum number of communication steps, which is four and three, respectively for nonspeculative and speculative algorithms.

Feng *et al.* [17] introduce a scalable dynamic multi-agent PBFT (SDMA-PBFT) algorithm to enhance scalability, that is effectively implemented in a permissioned blockchain system and it can be extended to large-scale distributed blockchain systems. The proposed method is intended for dynamic hierarchical agent node selection. This variation of PBFT provides the advantages of a more flexible system due to the dynamic hierarchical design, a faster integration of a new node to participate in the system, reducing time delays, and by increasing the number of consensus nodes, SDMA-PBFT provides better performance, higher throughput and reduced message processing time while increasing scalability.

Xu *et al.* [18] introduces a concurrent PBFT algorithm with reputation assessment for integration of blockchain and supply chain named C-PBFT. The focus of this research work stays on the integration between blockchain and supply chain where authors provide the framework that ensures the effective management of data, the cluster classification of different peers of a supply chain, and the reputation assessment which is taking place leveraging multi criteria decision making (MCDM) and simple additive weighting (SAW).

Li Zhang *et al.* [19] propose a hierarchical approach to the basic PBFT algorithm named Group Hierarch PBFT (GH-PBFT). The proposed solution consists of two parts: the Group protocol and Hierarchy consensus. In their research work they describe the PBFT algorithm and by focusing on the consensus efficiency of PBFT, authors present their proposed solution in order to increase efficiency while preserving security.

III. Practical Byzantine Fault Tolerance

The Practical Byzantine Fault Tolerance (PBFT) algorithm is based on the Byzantines general's problem, in which multiple parties attempt to reach consensus without fully trusting each other and without knowing which party is malicious or faulty. It is primarily utilized in private blockchain networks. This algorithm's primary function relies upon three phases of message exchange in order to reach an agreement [20], i.e., pre-prepare, prepare, and commit.

The client node transmits a message to a primary node – replica 0 – which then broadcasts the message to all other nodes. In each consensus round, the replica 0 or primary node is replaced by a view change protocol, which means that each system node has the potential to be the replica 0 node. The replica 0 node assigns metadata and certificates to messages sent and verified by other replica nodes. This constitutes the pre-prepare phase. The prepare phase follows, during which all replica nodes multicast the message back to the other replica nodes by appending a new certificate. In the event that the pre-prepare and prepare certificates of the messages received by the replica are identical, then the replica will multicast a commit message. After receiving the commit message, the replicas execute the client's request and return a response [21]. PBFT uses symmetric cryptography (i.e., MAC) instead of public key signatures to authenticate messages. The PBFT algorithm is illustrated in Fig. 1.

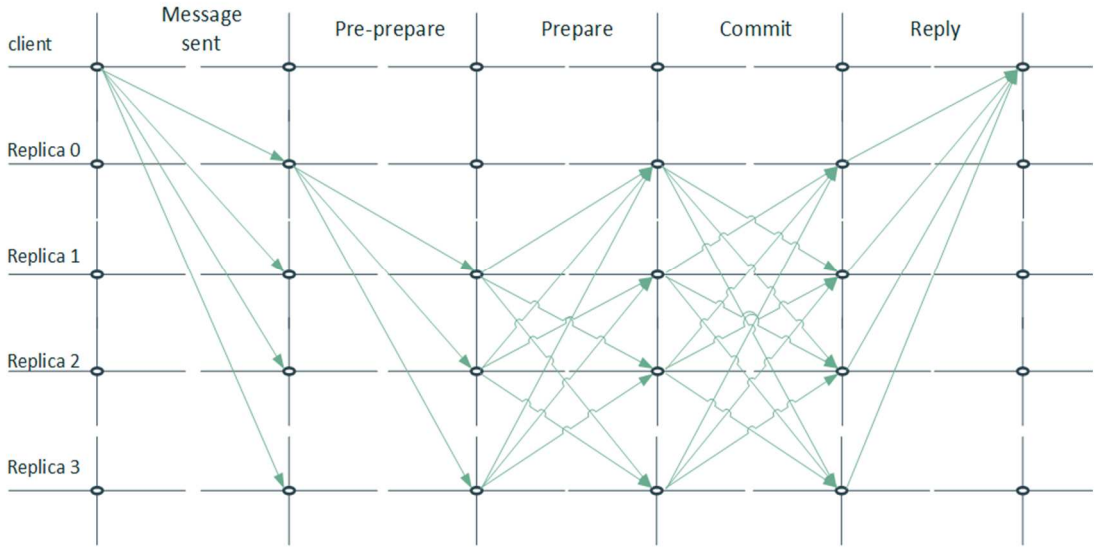

Fig. 1 PBFT timeline.

The security condition of the PBFT algorithm is that a vast majority of the nodes, at least two thirds of them, need to agree on the consensus before it can be considered secure. This indicates that a network with more than one third of nodes that are faulty or malicious cannot function securely and is at risk of being compromised. This leads to the conclusion that the security of a system improves in direct proportion to the increase in the system's total number of nodes. Therefore, the improved level of security achieved by the PBFT algorithm along with its high throughput, low latency, and low computational overhead make it a proper solution for permissioned blockchains in networks such as IoMT edge networks [22], [23]. On the other hand, it is worthwhile mentioning that two main drawbacks of the PBFT algorithm are the following: its susceptibility to Sybil attacks, and the excessive communication overhead that can be observed in the case of an increased number of nodes, leading to scalability issues. Both of these drawbacks are traced back to the algorithm's design. In this work, we focus on addressing the scalability issue by leveraging nodes clustering.

IV. Proposed PBFT-based Approach for IoMT Blockchains

In this section we present the scalable PBFT-based approach along with its communication overhead and communication complexity.

It is evident from the Section III that as the number of connected IoMT blockchain nodes (i.e., Gateways) in the network increases the total communication overhead of the PBFT algorithm does not scale sufficiently. Therefore, our proposed solution is to elaborate clustering techniques in order to improve the scalability and the total communication overhead. It is worthwhile mentioning that the following approach can be applicable with a number of clustering algorithms and each algorithm could contribute differently to the final result. However, we do not choose any specific clustering algorithm to present in this work because our aim is to demonstrate how the concept of clustering improves the scalability of the PBFT algorithm, and not specific clustering algorithms. Thus, we assume that the nodes are grouped based on clustering, and a Cluster Head (CH) has been elected prior to the initiation of the process. For this reason and given the fact that clustering as well as the CH election are one-time processes, while the PBFT consensus algorithm runs continuously in the distributed network with many iterations, we do not take into consideration the amount of communication overhead the clustering process and the CH selection add in the total.

To begin with, we have proceeded to the following assumptions:

1. We assume a network of N connected IoMT blockchain nodes (e.g., IoMT Gateways) that participate in a private permissioned blockchain, as shown in Fig. 2. Each green dot corresponds to a blockchain node, while each blue dot corresponds to an IoMT sensor or actuator.
2. The IoMT blockchain nodes possess enough resources to execute the PBFT algorithm and reach a consensus regarding a transaction proposal and the latest update of the distributed ledger.
3. In the assumed network, as shown in Fig. 3, the blockchain nodes have already been grouped into K clusters and K Cluster Head (CH) nodes have been elected, where K is given by the following equation:

$$K = \lceil log_2 N \rceil. \quad (1)$$

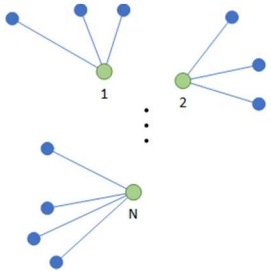

Fig. 2 IoMT nodes.

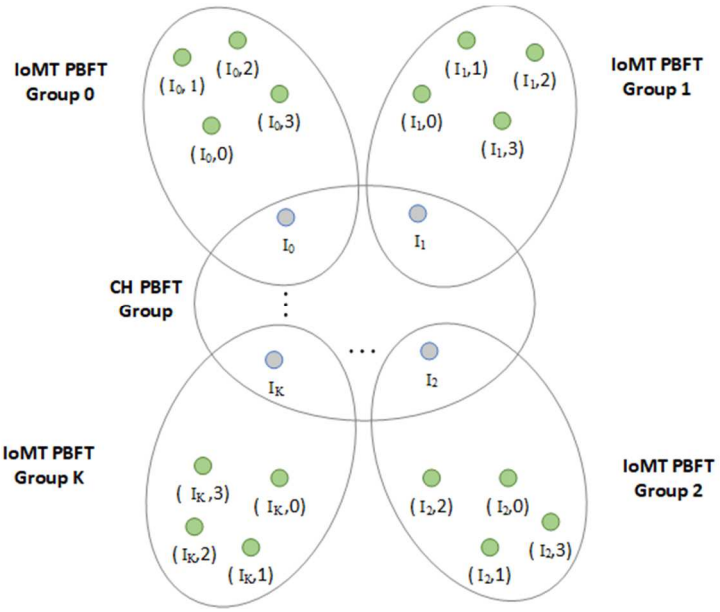

Fig. 3 Node clustering.

4. Consequently, the total communication complexity is $O(N^2)$ [21] where N is the number of connected IoMT blockchain nodes.
5. We define as "*IoMT PBFT Group*" a group consisting of a number of IoMT blockchain nodes (i.e., Gateways) and one CH node that is one of the IoMT blockchain nodes of the group.
6. We define as "*CH PBFT Group*" the group consisting of the CHs nodes of each "*IoMT PBFT Group*".

In the *CH PBFT Group* each CH node is identified with a unique identifier I_i related to its device ID, where i is the group. Each CH node can communicate directly with: i) every IoMT node included in its *IoMT PBFT Group* as well as, ii) any other CH node that lead another *IoMT PBFT Group.* In each *IoMT PBFT Group*, an IoMT node is represented by a pair of numbers (I_i, J) of which the first one (i.e., I_i) is the corresponding CH identifier of the leading CH node and the second one (i.e., J) is the node identifier of the IoMT node itself. In an *IoMT PBFT Group*, the nodes can communicate directly with each other and with the leading CH node, but they cannot communicate directly with IoMT nodes of other *IoMT PBFT Groups* or CH nodes.

After clustering and the establishment of a number of *IoMT PBFT Groups* and the *CH PBFT Group*, the PBFT consensus algorithm is operating in this newly established clustered network as follows: each *IoMT PBFT Group* runs a PBFT consensus algorithm within the grouped IoMT nodes, while the *CH PBFT Group,* consisting of K CH nodes, runs a PBFT consensus algorithm with the CH nodes as participants. It is worthwhile noting that each CH node participates in two PBFT consensus processes. Subsequently, the final issue that needs to be addressed is how each Group will communicate and how the ledgers of the various groups will be synchronized.

Whenever a new transaction needs to be validated and the ledger to be updated, the client IoMT node (I_i, J) – as it is described in Section III - will initiate the PBFT consensus process. The overall process can be described in the following steps:

1. IoMT node (I_i, J) initiates a transaction proposal between the nodes of the *IoMT PBFT Group i*;
2. Then the CH I_i propagates the transaction proposal to the other CHs of the *CH PBFT Group*;
3. Each CH device (i.e., CH node 1, CH node 2, … CH node M), other than the CH I_i , makes the same transaction proposal as a client inside its own corresponding *IoMT PBFT Group*;
4. Each *IoMT PBFT Group* initiates the PBFT consensus algorithm. In the *IoMT PBFT Group I,* the node (I_i, J) initiates the PBFT process as a client while in the other *IoMT PBFT Groups,* the client that initiates the PBFT process is the corresponding CH;
5. Each *IoMT PBFT Group* will run a PBFT algorithm and provide an output regarding the consensus with a regular PBFT voting process;
6. The CH I_i initiates a final PBFT algorithm where each CH's vote is the output of the PBFT of its corresponding *IoMT PBFT Group* that took place in Step 5;
7. CHs produce the final output, and each CH will propagate the result to its corresponding *IoMT PBFT Group* and each IoMT node will update the state of the ledger.

We define the Total Communication Overhead (TCO) of the PBFT process as the sum of the Communication Overhead (CO) of all locally executed PBFT consensus algorithms (i.e., inside each *IoMT PBFT Group* and the *CH PBFT Group*). As we have defined the total number of IoMT nodes as N in the

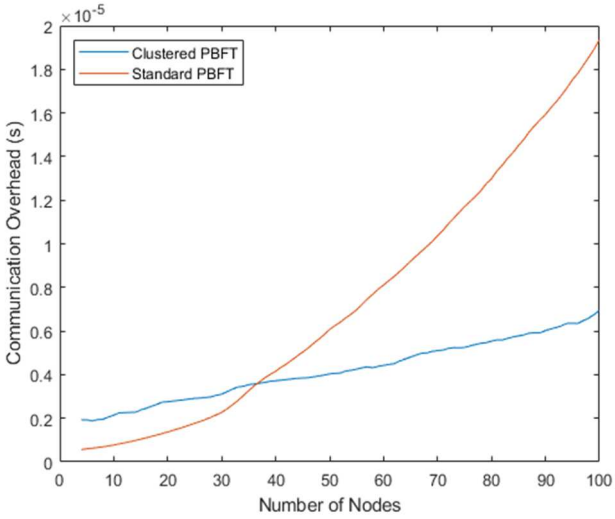

Fig. 4 PBFT and Clustered PBFT comparison.

IoMT PBFT Group, and the number of CH nodes as K, the TCO is given by the following equation 2:

$$TCO = CO_{CH\ PBFT\ Group} + \sum_{i=0}^{K-1} CO_{IoMT\ PBFT\ Group\ i} \quad (2)$$

Furthermore, the Communication Complexity (CC) of the PBFT process in the *CH PBFT Group* is given by the equation 3, and the communication complexity of the PBFT process in each *IoMT PBFT Group* is given by the equation 4:

$$CC_{CH\ PBFT\ Group} \sim O(K^2) \sim O(\lceil log_2 N \rceil^2) \sim O(N); \quad (3)$$

$$CC_{IoMT\ PBFT\ Group\ i} \sim O\left(\left(\frac{N}{K}\right)^2\right) \sim O\left(\left(\frac{N}{\lceil log_2 N \rceil}\right)^2\right) \sim O\left(\frac{N^2}{N}\right) \sim O(N). \quad (4)$$

As a result, the Total Communication Complexity (TCC) is given by the equation 5 that leads to equation 6:

$$TCC \sim O(N) + \sum_{i=0}^{K-1} O(N) \sim O(N) + K * O(N) \sim O(N) + \lceil log_2 N \rceil * O(N); \quad (5)$$

$$TCC \sim O(N * log_2 N). \quad (6)$$

According to (6), it is worthwhile mentioning that the total communication complexity of the proposed approach (i.e., $O(N * log_2 N)$) is lower than the initial one (i.e., $O(K^2)$).

V. Performance Evaluation

Followingly, we present the simulation results of the comparison between the standard PBFT and the proposed scalable PBFT-based approach in terms of communication overhead. MATLAB R2018b was the platform where we performed simulations for both the standard PBFT and the proposed scalable PBFT-based approach for 4 up to 100 nodes. We have clustered the nodes according to equation (1) and the simulation results are presented in Fig. 4. We have assumed that the transmission of a message between two nodes has a stable time duration, during both the standard PBFT and the proposed scalable PBFT-based approach, and we have not taken into consideration external factors that add or reduce communication overhead (e.g., communication channel properties).

In Fig. 4 the y-axis represents the total communication overhead which is the time duration for the messages to be exchanged between nodes in one iteration of both cases, while the x-axis represents the number of nodes that take part in both cases. the orange curve (standard PBFT) demonstrates that the communication overhead is increasing exponentially as the number of participating nodes is increasing linearly. On the other hand, the blue curve (scalable PBFT-based approach) demonstrates that that the increase of the communication overhead is almost linear in proportion to the increase of number of participating nodes. The results lead to the conclusion that up to a point of certain nodes (e.g., 37), it is preferable and more convenient to use the standard PBFT because the communication overhead of the standard approach is lower. After that threshold (i.e., 37 for the given simulation setup), we notice that a clustered approach of PBFT provides better results in terms of communication overhead in comparison to the standard PBFT.

VI. Conclusion and Future Work

In this paper, we proposed a scalable PBFT-based approach for IoMT blockchains. In our proposal we leverage the concept of clustering order to: i) enhance scalability in IoMT blockchains, ii) reduce communication overhead, iii) enhance security while reducing the computational cost for suitability to the resource constraint nature of IoMT devices, iv) facilitate decentralized accountability, and iv) eliminate single point of failure. It is noteworthy to highlight that our approach can be adapted according to different clustering algorithms and the final result may vary for every algorithm. However, it is worthwhile mentioning that clustering improves the scalability of the PBFT algorithm. As future work, we intend to generate a full-scale algorithm based on our approach, implement the proposed algorithm in a virtual environment and evaluate it in terms of performance metrics such as communication overhead, time consumption and energy efficiency.

REFERENCES

[1] F. P. Oikonomou, G. Mantas, P. Cox, F. Bashashi, F. Gil-Castineira, and J. Gonzalez, "A Blockchain-based Architecture for Secure IoT-based Health Monitoring Systems," pp. 1–6, Dec. 2021, doi: 10.1109/CAMAD52502.2021.9617803.

[2] F. P. Oikonomou, Pelekoudas, J. Ribeiro, G. Mantas, J. M. C. S. Bastos, and J. Rodriguez, "A Hyperledger Fabric-based Blockchain Architecture to Secure IoT-based Health Monitoring Systems," 2021, Accessed: Jan. 20, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9647521/.

[3] F. Pelekoudas-oikonomou et al., "Blockchain-Based Security Mechanisms for IoMT Edge Networks in IoMT-Based Healthcare Monitoring Systems," 2022.

[4] E. Karavatselou, M. A. Fengou, G. Mantas, and D. Lymberopoulos, "Profile management system in ubiquitous healthcare cloud computing environment," Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST, vol. 263, pp. 105–114, 2019, doi: 10.1007/978-3-030-05195-2_11.

[5] M. Papaioannou et al., "A Survey on Security Threats and Countermeasures in Internet of Medical Things (IoMT)," Trans. Emerg. Telecommun. Technol., 2020, doi: 10.1002/ett.4049.

[6] G. Mantas, D. Lymberopoulos, and N. Komninos, "A new framework for ubiquitous context-aware healthcare applications," in Proceedings of the IEEE/EMBS Region 8 International Conference on Information Technology Applications in Biomedicine, ITAB, 2010, doi: 10.1109/ITAB.2010.5687758.

[7] M. Karageorgou, G. Mantas, I. Essop, and J. Rodriguez, "Cybersecurity attacks on medical IoT devices for smart city healthcare services," 2020, Accessed: May 29, 2022. [Online]. Available: http://gala.gre.ac.uk/id/eprint/26517/.

[8] G. Zachos, I. Essop, G. Mantas, K. Porfyrakis, J. C. Ribeiro, and J. Rodriguez, "Generating IoT Edge Network Datasets based on the TON_IoT Telemetry Dataset," IEEE Int. Work. Comput. Aided Model. Des. Commun. Links Networks, CAMAD, vol. 2021-Octob, pp. 1–6, 2021, doi: 10.1109/CAMAD52502.2021.9617799.

[9] I. Essop, J. C. Ribeiro, M. Papaioannou, J. Rodriguez, G. Zachos, and G. Mantas, "Generating datasets for anomaly-based intrusion detection systems in iot and industrial iot networks," Sensors, vol. 21, no. 4, pp. 1–31, 2021, doi: 10.3390/s21041528.

[10] M. Papaioannou, G. Zachos, I. Essop, G. Mantas, and J. Rodriguez, "Towards a Secure and Usable User Authentication for Mobile Passenger ID Devices for Land/Sea Border Control," IEEE Access, 2022, doi: 10.1109/ACCESS.2022.3164245.

[11] P. P. Ray, D. Dash, K. Salah, and N. Kumar, "Blockchain for IoT-Based Healthcare: Background, Consensus, Platforms, and Use Cases," IEEE Syst. J., pp. 1–10, 2020, doi: 10.1109/jsyst.2020.2963840.

[12] X. Hao, L. Yu, L. Zhiqiang, L. Zhen, and G. Dawu, "Dynamic practical byzantine fault tolerance," 2018 IEEE Conf. Commun. Netw. Secur. CNS 2018, 2018, doi: 10.1109/CNS.2018.8433150.

[13] S. Gao, T. Yu, J. Zhu, and W. Cai, "T-PBFT: An EigenTrust-based practical Byzantine fault tolerance consensus algorithm," China Commun., vol. 16, no. 12, pp. 111–123, 2019, doi: 10.23919/JCC.2019.12.008.

[14] X. Fan, "Scalable Practical Byzantine Fault Tolerance with Short-lived Signature Schemes," Proc. 28th Annu. Int. Conf. Comput. Sci. Softw. Eng., pp. 245–256, 2018, [Online]. Available: http://dl.acm.org/citation.cfm?id=3291291.3291316.

[15] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," J. Cryptol., vol. 17, no. 4, pp. 297–319, 2004, doi: 10.1007/s00145-004-0314-9.

[16] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Verissimo, "Efficient byzantine fault-tolerance," IEEE Trans. Comput., vol. 62, no. 1, pp. 16–30, 2013, doi: 10.1109/TC.2011.221.

[17] L. Feng, H. Zhang, Y. Chen, and L. Lou, "Scalable dynamic multi-agent practical byzantine fault-tolerant consensus in permissioned blockchain," Appl. Sci., vol. 8, no. 10, 2018, doi: 10.3390/app8101919.

[18] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent Practical Byzantine Fault Tolerance for Integration of Blockchain and Supply Chain," ACM Trans. Internet Technol., vol. 21, no. 1, pp. 1–17, 2021, doi: 10.1145/3395331.

[19] L. Zhang and Q. Li, "Research on Consensus Efficiency Based on Practical Byzantine Fault Tolerance," 10th Int. Conf. Model. Identif. Control. ICMIC 2018, no. Icmic, pp. 2–4, 2018, doi: 10.1109/ICMIC.2018.8529940.

[20] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," 2018, Accessed: Dec. 20, 2021. [Online]. Available: https://iris.uniroma1.it/bitstream/11573/1337256/1/DeAngelis_PBFT_2018.pdf.

[21] M. Castro, "Practical Byzantine Fault Tolerance," 2001.

[22] M. Salimitari and M. Chatterjee, "A Survey on Consensus Protocols in Blockchain for IoT Networks."

[23] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," ACM Trans. Comput. Syst., vol. 20, no. 4, pp. 398–461, Nov. 2002, doi: 10.1145/571637.571640.