Development of Blockchain-Based Framework for Securing Communication in Wireless Robotic Platforms

Alexander Alexandrov[1,*]

[1]Institute of Robotics – Bulgarian Academy of Sciences, Acad. Georgi Bonchev Str., Bl. 2, Sofia, 1113, Bulgaria

Abstract

Wireless robotic platforms are rapidly gaining traction across various industries due to their adaptability, mobility, and efficiency. However, their reliance on wireless communication makes them vulnerable to a variety of cybersecurity threats, including data breaches, denial-of-service attacks, and unauthorized access. To address these challenges, blockchain technology offers a decentralized, secure, and immutable framework for safeguarding communications.

This paper explores the development of a blockchain-based framework specifically designed to secure communication in wireless robotic platforms using the Byzantine Fault Tolerance (BFT) approach. It examines the potential of blockchain for decentralized authentication, data integrity, and security, and presents a proposed framework design along with future research directions.

Keywords

Blockchain, Wireless Robotic Platforms, Byzantine Fault Tolerance (BFT)

1. Introduction

Wireless robotic platforms have emerged as a transformative technology in domains such as manufacturing, healthcare, defense, and autonomous vehicles. These platforms rely on wireless communication technologies like Wi-Fi, Bluetooth, ZigBee, and 5G to perform critical tasks autonomously or in cooperation with other robots and centralized control systems. However, the open and dynamic nature of wireless networks exposes robotic platforms to significant cybersecurity risks. Data interception, unauthorized command injection, denial of service (DoS), and man-in-the-middle (MitM) attacks are just a few examples of the risks faced by these systems.

Blockchain technology [1, 2], initially developed for secure and decentralized financial transactions, offers an intriguing solution for these security challenges. Blockchain's decentralized nature, combined with its immutable ledger [3, 4], strong cryptographic protocols, and consensus mechanisms, provides an opportunity to enhance the security of wireless communication in robotic systems [5].

Wireless robotic platforms are composed of various components, including sensors, actuators, controllers, and communication modules [6]. The wireless nature of these platforms allows them to operate flexibly in distributed environments [7]. However, this flexibility comes with inherent security vulnerabilities: Lack of Centralized Control: The decentralized nature of many wireless robotic systems (such as robotic swarms) creates challenges in maintaining a centralized authority to enforce security policies [8, 9, 10].

Dynamic Topologies: Robotic platforms often operate in dynamic and changing environments, such as factories or battlefields, where network topology can change frequently. This makes traditional security methods, like static encryption keys or centralized access control, impractical.

High Exposure to Interference and Attacks: Wireless communication channels [11] are inherently vulnerable to interception, jamming, and injection attacks, threatening both the integrity and availability of communication between robots and control systems [12, 13]. These vulnerabilities make it essential to develop robust, adaptive, and scalable security solutions.

BISEC'2024: 15th International Conference on Business Information Security, November 28-29, 2024, Niš, Serbia
*Corresponding author.
akalexandrov@ir.bas.bg (A. Alexandrov)
0000-0002-8787-9235 (A. Alexandrov)

Blockchain is a decentralized and distributed ledger technology [14, 15], best known for underpinning cryptocurrencies such as Bitcoin. The core features of blockchain that make it relevant for securing wireless robotic platforms include:

Decentralization: Blockchain eliminates the need for a central authority by allowing each node in the network to maintain a copy of the ledger [16]. This fits well with decentralized robotic platforms, where multiple robots need to interact securely without relying on a central server [17].

Immutability: Once data is recorded on the blockchain, it cannot be altered or tampered with, providing a secure history of transactions [18] and communications. This immutability is valuable for ensuring the integrity of data exchanged between robotic systems.

Consensus Mechanisms: Blockchain uses consensus algorithms (such as Proof of Work or Proof of Stake) to validate and confirm transactions across the network [19]. This ensures that only authorized and verified communications are added to the ledger.

Cryptographic Security: Blockchain relies on cryptographic techniques (e.g., hashing and digital signatures) to secure data and ensure the authenticity of participants. These features of blockchain provide a robust foundation for developing a secure communication framework for wireless robotic platforms.

Wireless robotic platforms are increasingly being deployed in numerous industries, ranging from manufacturing and logistics to healthcare, and defense. These platforms rely heavily on efficient and secure communication among robots to ensure coordinated operations.

The decentralized nature of these platforms, coupled with the need for real-time communication, makes them vulnerable to various cyber-attacks, including data breaches, denial of service (DoS) attacks, and the possibility of compromised nodes [20].

Blockchain technology has emerged as a powerful solution to these security challenges due to its decentralized, tamper-resistant, and transparent nature. In particular, consensus algorithms like Byzantine Fault Tolerance (BFT) have proven to be effective in securing communication in distributed systems, even in the presence of malicious or faulty nodes . Blockchain technology is a decentralized distributed ledger that allows multiple participants to agree on the state of a system without relying on a central authority. It achieves security through cryptographic techniques, consensus algorithms, and an immutable ledger of transactions [8]. Blockchain's primary advantage in wireless robotic platforms is its ability to establish trust between autonomous nodes (robots) in an untrusted environment.

Wireless robotic platforms consist of multiple robots that communicate wirelessly to perform tasks such as exploration, mapping, or surveillance. These platforms must be resilient to communication failures and security threats, which makes blockchain an appealing solution. However, the consensus mechanism employed in the blockchain is key to ensuring the system's fault tolerance and security.

2. Related Works

The implementation of reliable secure communication between wireless robotic platforms as autonomous Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) is crucial for the network security of systems handling sensor data, as it can detect attempts by hackers and bots to hack the network, steal sensitive data, or initiate DOS or DDOS attacks. The present study focuses on the development of a new Blockchain-Based Framework for securing the communication between wireless Robotic platforms connected in network. The authors in [16] propose method named Extended-BATMAN (E-BATMAN) incorporates the concept of blockchain into the BATMAN protocol using MANET. As a secure, distributed, and reliable platform, Blockchain solves most BFT security issues, with each node performing repeated security operations individually.

The authors in [21] propose Byzantine fault-tolerant (BFT) as consensus mechanism aimed at addressing possible hardware errors, network congestion or interruptions, and malicious attacks in distributed systems. It ensures that nodes can reach consistent decisions in untrusted environments by solving Byzantine fault problems.

The authors in [22] propose blockchain-based technology called IoT-enabled Efficient Practical Byzantine Consensus-based Reputation (EPBCR). This approach effectively monitors the post-production business processes of electronic devices by using hybrid consensus and reputation optimization algorithms. The authors in [23] demonstrate the use of a novel blockchain technology aided peer-to-peer connection (P2P)-based access control protocol is proposed for the distributed ad hoc networks.

The authors in [24] propose splitting the underlying blockchain into sidechains, thereby reducing mining complexity and reducing the number of packets needed for communication while maintaining true decentralization. The model is compared with standard blockchain & sidechain implementations in terms of access time, reading delay, and writing delay.

3. Proposed design of Blockchain-Based Framework using Byzantine Fault Tolerance approach

Byzantine Fault Tolerance (BFT) and its Role in Blockchain

Byzantine Fault Tolerance (BFT) refers to a system's ability to tolerate arbitrary failures, including failures caused by malicious or misbehaving nodes, also known as Byzantine nodes. In a BFT system, the honest nodes must reach consensus even if some nodes are acting arbitrarily or maliciously. BFT algorithms ensure that the system continues to function correctly as long as the number of Byzantine nodes does not exceed a certain threshold.

In blockchain-based communication systems, BFT consensus algorithms provide a decentralized method of ensuring that all participating robots agree on the validity of transactions (messages) even in the presence of compromised robots. This makes BFT an ideal choice for securing communication in wireless robotic platforms, where nodes may fail or be attacked.

In a blockchain-based framework for wireless robotic platforms, robots must communicate securely and reach consensus on the state of the network. This is achieved through the use of a blockchain ledger, where each block contains transactions (messages exchanged between robots) and is secured using cryptographic hash functions. The BFT algorithm ensures that all honest robots agree on the order and validity of transactions, even if some robots are compromised.

3.1. Cryptographic Hash Functions

Cryptographic hash functions are essential for ensuring the integrity and authenticity of messages in a blockchain-based communication system. A hash function is a mathematical function that maps an input of arbitrary size to a fixed-size output (the hash value). In a blockchain, hash functions are used to secure the contents of each block and to link blocks together in a chain.

Mathematically, a cryptographic hash function H can be defined as:

$$H : \{0,1\}^* \rightarrow \{0,1\}^n \tag{1}$$

where $\{0,1\}^*$ represents the set of all binary strings of arbitrary length, and $\{0,1\}^n$ is the set of binary strings of fixed length n.

Key properties of cryptographic hash functions include:

- Preimage resistance: Given a hash value h, it is computationally infeasible to find an input x such that $H(x) = h$.
- Second preimage resistance: Given an input x and its corresponding hash $H(x) = h$, it is infeasible to find a different input x' such that

$$H\left(x'\right) = H\left(x\right) \tag{2}$$

- Collision resistance: It is computationally infeasible to find two different inputs x and x' such that

$$H\left(x\right) = H\left(x'\right) \tag{3}$$

- Deterministic: The same input will always produce the same hash value.

In a wireless robotic platform, each robot generates a cryptographic hash of the messages it sends. Other robots can verify the integrity of the received messages by recalculating the hash and comparing it with the transmitted hash. This ensures that messages have not been tampered with during transmission.

3.2. Digital Signatures

Digital signatures provide a mechanism for verifying the authenticity and integrity of a message. In a blockchain-based framework for wireless robotic platforms, each robot is assigned a public-private key pair, where the private key is used to sign messages, and the public key is used to verify the signatures of other robots. Mathematically, a digital signature scheme consists of three algorithms:

1. Key Generation: Generates a pair of keys ($\mathtt{pk}$, $\mathtt{sk}$), where $\mathtt{pk}$ is the public key and $\mathtt{sk}$ is the private key.
2. Signing: The signing algorithm takes a message $\mathtt{m}$ and a private key $\mathtt{sk}$, and produces a signature σ:
$$\sigma = Sign\left(\mathtt{sk}, \mathtt{m}\right) \tag{4}$$
3. Verification: The verification algorithm takes a message $\mathtt{m}$, a signature σ, and a public key $\mathtt{pk}$. It returns "True" if the signature is valid and "False" otherwise:
$$Verify\left(\mathtt{pk}, \mathtt{m}, \sigma\right) \in \{True, False\} \tag{5}$$

Digital signatures in blockchain ensure that messages exchanged between robots are authentic and have not been forged. When a robot sends a message, it signs the message with its private key. The recipient robots verify the signature using the sender's public key, ensuring that the message was indeed sent by the correct robot and has not been altered.

3.3. Public-Key Cryptography

Public-key cryptography, also known as asymmetric cryptography, enables secure communication between robots in a wireless robotic platform. Each robot generates a public-private key pair, where the public key is shared with other robots, and the private key is kept secret.

In public-key cryptography, encryption and decryption are performed as follows:

Encryption: The sender encrypts the message $\mathtt{m}$ using the recipient's public key $\mathtt{pk}$:

$$\mathtt{c} = Encrypt\left(\mathtt{pk}, \mathtt{m}\right) \tag{6}$$

Decryption: The recipient decrypts the ciphertext $\mathtt{c}$ using their private key $\mathtt{sk}$:

$$\mathtt{m} = Decrypt\left(\mathtt{sk}, \mathtt{c}\right) \tag{7}$$

This ensures that only the intended recipient can decrypt and read the message. Public-key cryptography is essential for ensuring the confidentiality of messages exchanged between robots in a blockchain-based communication system.

3.4. Blockchain Data Structure

In a blockchain-based communication system, data (messages exchanged between robots) is recorded in blocks, and each block is linked to the previous block using cryptographic hashes, forming a chain of blocks. Each block contains the following components:

- Transactions: The set of messages exchanged between robots.
- Timestamp: The time at which the block was created. Previous Block Hash: The cryptographic hash of the previous block in the chain.

- Nonce: A random value used in the consensus process (e.g., Proof of Work).
- Block Hash: The cryptographic hash of the current block, calculated based on the block's contents.

Mathematically, the hash of a block B can be represented as:

$$H(B) = H(Trans||Tstamp||PBH||N) \tag{8}$$

Where: $H(B)$ represents the Hash of the current block, $||$ denotes concatenation, $Trans$ represents the set of transactions, $Tstamp$ represents the Time Stamp, PBH represents the Previous Block Hash, N is Nonce (random value).

The hash of each block depends on the hash of the previous block, which ensures that if any block is modified, the hashes of all subsequent blocks will change, making it easy to detect tampering.

3.5. Byzantine Fault Tolerance in Wireless Robotic Platforms

In wireless robotic platforms, robots must coordinate and communicate securely, even if some robots are faulty or malicious. The Byzantine Fault Tolerance (BFT) approach ensures that the system can reach consensus on the state of the blockchain, even if up to f robots out of n total robots are faulty or malicious. This section provides a mathematical overview of BFT consensus algorithms and their application to blockchain-based communication in wireless robotic platforms.

3.6. Byzantine Generals Problem

The Byzantine Generals Problem is a classic problem in distributed systems that illustrates the challenge of reaching consensus in the presence of faulty or malicious nodes. The problem can be described as follows:

There are n generals (robots) who must agree on a common plan of action. Some generals may be traitors (faulty or malicious) and may send conflicting or false information to the other generals. The goal is for all loyal generals to agree on the same plan, even if some generals are traitors.

Mathematically, the system is said to be Byzantine Fault Tolerant if it can reach consensus as long as the number of faulty or malicious robots f satisfies:

$$f < \frac{n}{3} \tag{9}$$

This means that the system can tolerate up to n Byzantine robots and still reach consensus.

3.7. Practical Byzantine Fault Tolerance (PBFT)

Practical Byzantine Fault Tolerance (PBFT) is one of the most widely used BFT consensus algorithms. PBFT is designed for systems where the number of participants (robots) is relatively small, and it ensures that the system can reach consensus even in the presence of faulty or malicious robots. The PBFT algorithm operates in rounds, where each round consists of three phases:

1. Pre-Prepare: The leader robot proposes a block of transactions to the other robots.
2. Prepare: Each robot receives the proposed block and broadcasts a prepare message to all other robots, indicating that it has received the block.
3. Commit: Each robot receives prepare messages from other robots and broadcasts a commit message if it has received enough prepare messages. Once a robot receives enough commit messages, it considers the block to be committed and adds it to its local copy of the blockchain.

Mathematically, let n be the total number of robots, and let f be the number of faulty or malicious robots. In PBFT, a robot considers a block to be committed if it receives commit messages from at least $n - f$ robots. This ensures that the block is committed by a majority of honest robots.

The time required to reach consensus in PBFT depends on the network latency and the number of message exchanges between robots. Let t_p represent the time for the pre-prepare phase, t_c represent

the time for the prepare phase, and t_f represent the time for the commit phase. The total consensus time $Tcons$ is given by:

$$Tcons = t_p + t_c + t_f \tag{10}$$

Each phase involves the exchange of messages between robots, and the number of messages exchanged grows quadratically with the number of robots. Specifically, in each phase, each robot sends messages to all other robots, resulting in $\mathcal{O}\left(n^2\right)$ messages per phase. Therefore, the total number of messages exchanged in PBFT is $\mathcal{O}\left(n^2\right)$.

3.8. Security Analysis of BFT-Based Blockchain Communication in Wireless Robotic Platforms

A BFT-based blockchain framework for wireless robotic platforms offers several security guarantees, including:

- Resilience to Byzantine Failures: The system can tolerate up to $f < n/3$ faulty or malicious robots and still reach consensus.
- Integrity and Authenticity: Cryptographic hash functions and digital signatures ensure the integrity and authenticity of messages exchanged between robots.
- Non-Repudiation: Once a robot signs and broadcasts a message, it cannot deny having sent the message, as the digital signature provides undeniable proof of authorship.
- Confidentiality: Public-key cryptography ensures that only the intended recipient can decrypt and access the message content.

However, BFT-based systems also face some challenges:

- Scalability: The number of messages exchanged in PBFT grows quadratically with the number of robots, which can limit the system's scalability.
- Latency: PBFT requires multiple rounds of message exchanges, which can introduce latency, especially in large networks or networks with high communication delays.

4. Mathematical Modeling of BFT-Based Communication in Wireless Robotic Platforms

To optimize the performance of a BFT-based blockchain framework for wireless robotic platforms, we need to model the system's performance mathematically. In this section, we provide a mathematical model for key performance metrics, including message propagation time, consensus time, and fault tolerance.

4.1. Network Model

Consider a wireless robotic platform consisting of n robots $R_1, R_2, ..., R_n$, where each robot communicates wirelessly with others. The communication network can be represented as a graph $G = (V, E)$, where V represents the set of robots and E represents the set of communication links between them. Each edge $(R_i, R_j) \in E$ represents a communication link between wireless robot node R_i and wireless robot node R_j, and may be associated with a communication delay $d\left(R_i, R_j\right)$ and bandwidth $b\left(R_i, R_j\right)$.

4.2. Message Propagation Time

When a robot node sends a message, the message must be propagated to all other robot nodes in the network. Let t_m represent the time it takes to propagate a message `m` to all nodes. The propagation time

depends on the network topology, communication delays, and bandwidth constraints. Mathematically, we can model the message propagation time as:

$$t_m = \max_{R_i \in V} \left(\sum\nolimits_{(R_i, R_j) \in P} d\left(R_i, R_j\right) \right) \tag{11}$$

where P represents the set of communication paths from the sender node to the recipient node.

4.3. Fault Tolerance

The fault tolerance of the BFT-based blockchain framework is determined by the number of faulty or malicious robots f that the system can tolerate. As mentioned earlier, the system can tolerate up to $f < n/3$ faulty robots. This ensures that the majority of wireless nodes are reliable and can reach consensus.

5. Optimizing BFT-Based Blockchain Communication in Wireless Robotic Platforms

To optimize the performance of a BFT-based blockchain framework for wireless robotic platforms, several strategies can be employed:

- Sharding: Sharding divides the network into smaller groups (shards), each responsible for processing a subset of transactions. This reduces the communication overhead and improves scalability.
- Lightweight Consensus Algorithms: In resource-constrained environments, lightweight consensus algorithms such as Delegated BFT (DBFT) can be used to reduce the number of message exchanges and improve consensus speed.
- Latency Reduction Techniques: Techniques such as message aggregation, where multiple messages are combined into a single message, can reduce the number of message exchanges and lower latency.

5.1. Applications of BFT-Based Blockchain Communication in Wireless Robotic Platforms

BFT-based blockchain communication systems can be applied to a wide range of applications in wireless robotic platforms, including:

1. Swarm Robotics: Swarm robotics involves large groups of robots that coordinate to perform tasks. BFT-based blockchain ensures secure communication and coordination among robots, even if some robots are faulty or compromised.
2. Autonomous Vehicles: Autonomous vehicles rely on secure communication to exchange information about road conditions, traffic, and obstacles. BFT-based blockchain can prevent malicious attacks that could compromise vehicle safety.
3. Healthcare Robotics: In healthcare, robots are used for tasks such as surgery, patient monitoring, and drug delivery. BFT-based blockchain ensures that sensitive medical data is exchanged securely between robots and healthcare providers.

5.2. Key Features of the Framework

The wireless robotic platform represents a network of multiple robots communicating with each other over wireless communication channels. Each robot acts as a node in the blockchain network, participating in the communication and consensus process. Robots are the nodes which serve as independent entities in the wireless network, equipped with sensors, processing units, and RF- based communication hardware. The new developed blockchain-based framework design for securing communication in wireless robotic platforms, is shown on the block diagram on Fig. 1.

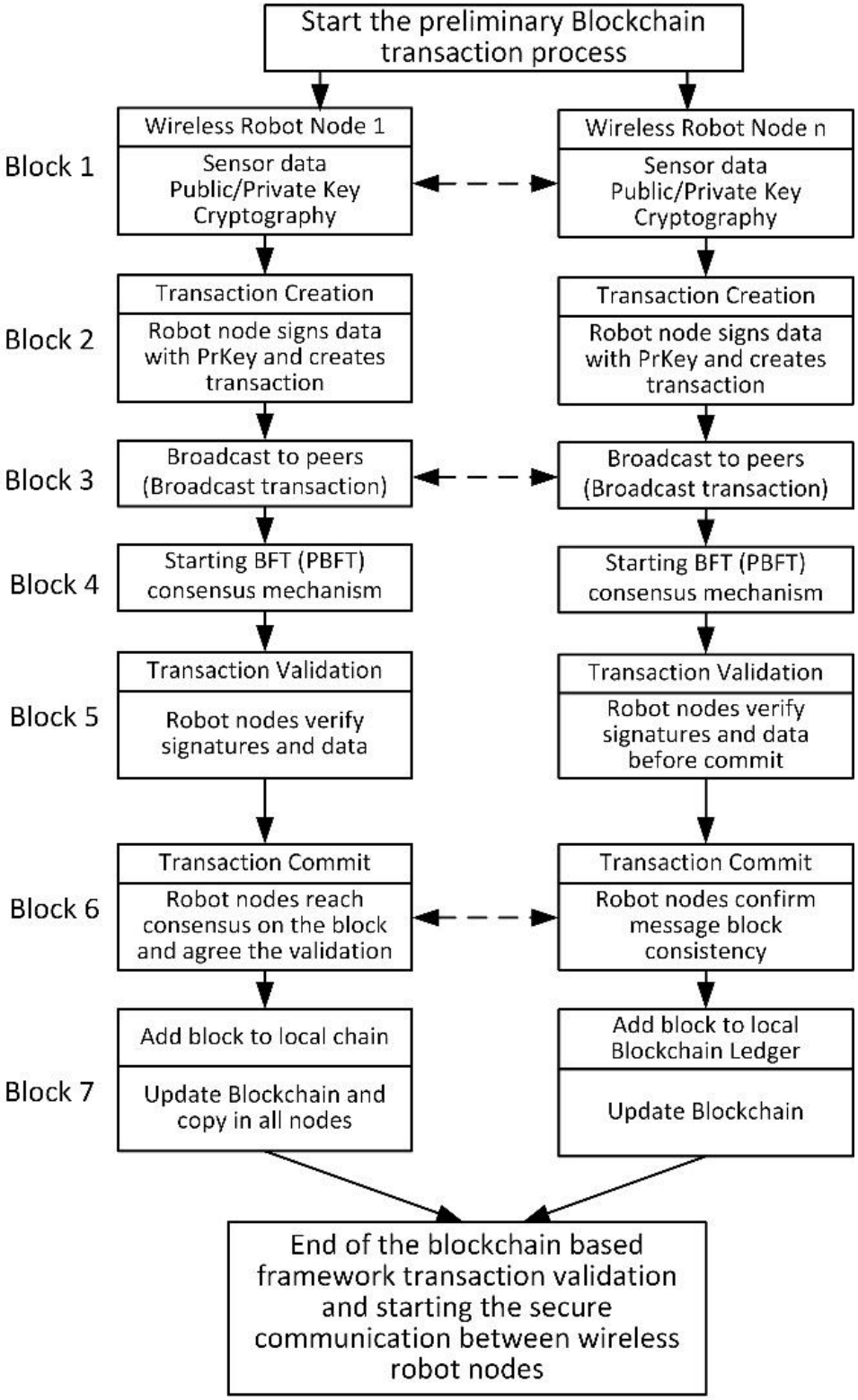

Figure 1: Blockchain-Based Framework for Securing Communication in Wireless Robotic Platforms Using Byzantine Fault Tolerance (BFT).

Key Components of the Block Diagram

- **Block 1.** Public-Private Key Cryptography: Each robot has a public-private key pair used for signing and verifying transactions (messages).
- **Block 2.** Transaction Creation & Signing When a robot generates new data (e.g., sensor readings,

control signals), it signs the message using its private key, creating a transaction. The transaction contains:

1. Message content: The data to be shared with other robots.
2. Digital signature: The signature generated by the robot's private key, ensuring message authenticity.
3. Timestamp: Time information for synchronization.

This step ensures that only authorized robots can send messages, and the recipient robots can verify that the message has not been tampered with.

- **Block 3.** Broadcast to Peer Robots: Once the transaction is created, it is broadcasted to all peer robots in the network. This broadcast is an important step in decentralized communication, where robots do not rely on a central server but instead communicate with all nodes in the network.
- **Block 4**. Byzantine Fault Tolerance (BFT) Consensus Mechanism: The core component of the system is the BFT consensus algorithm, specifically Practical Byzantine Fault Tolerance (PBFT) in this case. PBFT works in several phases:

  1. Pre-Prepare Phase: The leader robot proposes the block (containing transactions).
  2. Prepare Phase: Each robot verifies the block's contents, ensuring the transactions are valid (digital signatures and hash validation).
  3. Commit Phase: Once robots have received enough prepare messages, they broadcast commit messages to confirm the block's validity.

  The consensus mechanism ensures that even if up to f robots are faulty or malicious (Byzantine nodes), the system can still achieve agreement on the state of the blockchain, provided that $n \geqslant 3f + 1$ (where n is the total number of robots).
- **Block 5**. Transaction Validation & Preparation: Once robots receive transactions, they perform validation to ensure that the messages are legitimate. This involves verifying the digital signatures of the transactions to confirm that the sender is authentic, and checking the data integrity using hash functions to ensure the message has not been altered. If the transaction passes validation, it proceeds to the preparation stage for consensus.
- **Block 6**. Transaction Commit (BFT-PBFT): After the preparation and commit phases, robots reach consensus on the validity of the transactions. In the Commit Phase, wireless robots exchange messages confirming that the proposed block (set of transactions) is valid. The consensus is only finalized when enough robots agree on the validity of the transactions. This prevents Byzantine nodes from altering or corrupting the blockchain.
- **Block 7**. Add Block to Local Blockchain Ledger: Once consensus is reached, each robot adds the new block to its local copy of the blockchain. This block contains valid transactions (messages between robots) and cryptographic hashes linking it to the previous block, ensuring immutability.

Every robot maintains a synchronized copy of the blockchain, ensuring that all communication history is consistent across the network

6. Conclusion

The proposed blockchain-based framework for securing communication in wireless robotic platforms using a Byzantine Fault Tolerance (BFT) approach provides a robust solution to the security challenges faced by these platforms. BFT consensus algorithms ensure that robots can reach consensus on the state of the system, even in the presence of faulty or malicious robots. Cryptographic techniques such as hash functions, digital signatures, and public-key cryptography ensure the integrity, authenticity, and confidentiality of messages exchanged between robots.

While BFT-based systems offer strong security guarantees, they also face challenges related to scalability and latency. Future research should focus on optimizing BFT-based communication systems for

wireless robotic platforms by exploring techniques such as sharding, lightweight consensus algorithms, and latency reduction strategies. As wireless robotic platforms continue to evolve, BFT-based blockchain frameworks will play a critical role in ensuring secure and reliable communication in a wide range of applications.

Declaration on Generative AI

The author has not employed any Generative AI tools.

References

[1] N. Zdravković, M. Bogdanović, M. Trajanović, V. Ponnusamy, Implementing blockchain technology for health-related early response service in emergency situations, in: Proceedings of International Conference on Deep Learning, Computing and Intelligence: ICDCI 2021, Springer, 2022, pp. 237–243.

[2] P. Vijayakumar, S. Pavithraa, S. Harithaa, N. Zdravković, A web deployment of secured ecg signal medical record transactions using blockchain, Annals of the Romanian Society for Cell Biology 25 (2021) 19937–19951.

[3] N. Zdravković, J. Jović, M. damnjanović, Secure credentialing in e-learning using blockchain, in: Proc. of the 11th International Conference on e-Learning, 2020, pp. 39–43.

[4] D. Saveetha, G. Maragatham, V. Ponnusamy, N. Zdravković, An integrated federated machine learning and blockchain framework with optimal miner selection for reliable ddos attack detection, IEEE Access (2024).

[5] N. Mahalingam, P. Sharma, An intelligent blockchain technology for securing an iot-based agriculture monitoring system, Multimedia tools and applications 83 (2024) 10297–10320.

[6] S. Alsubai, A. Alqahtani, H. Garg, M. Sha, A. Gumaei, A blockchain-based hybrid encryption technique with anti-quantum signature for securing electronic health records, Complex & Intelligent Systems (2024) 1–25.

[7] A. Raj, S. Prakash, An efficient blockchain-based access control framework for iot-healthcare system, Wireless Personal Communications 136 (2024) 1017–1045.

[8] A. Kumar, S. Kumar, An advance encryption and attack detection framework for securing smart cities data in blockchain using deep learning approach, Wireless Personal Communications 135 (2024) 1329–1362.

[9] D. Xu, J. Gao, L. Zhu, F. Gao, Y. Han, J. Zhao, B-tor: Anonymous communication system based on consortium blockchain, Peer-to-Peer Networking and Applications 16 (2023) 2218–2241.

[10] W. Liang, J. Zhao, Y. Liu, Y. Liang, J. Li, Fairness resource allocation based on blockchain for secure communication in integrated iot, EURASIP Journal on Advances in Signal Processing 2023 (2023) 115.

[11] L. He, F. Li, H. Xu, W. Xia, X. Zhang, X. Tao, Blockchain-based vehicular edge computing networks: the communication perspective, Science China Information Sciences 66 (2023) 172301.

[12] R. Ameri, M. R. Meybodi, An improved cellular goore game-based consensus protocol for blockchain, Cluster Computing (2024) 1–26.

[13] O. G. Bautista, M. H. Manshaei, R. Hernandez, K. Akkaya, S. Homsi, S. Uluagac, Mpc-abc: Blockchain-based network communication for efficiently secure multiparty computation, Journal of Network and Systems Management 31 (2023) 68.

[14] Y. Liu, F. R. Yu, X. Li, H. Ji, V. C. Leung, Blockchain and machine learning for communications and networking systems, IEEE communications surveys & tutorials 22 (2020) 1392–1431.

[15] J. C. Priya, R. Praveen, K. Nivitha, T. Sudhakar, Improved blockchain-based user authentication protocol with ring signature for internet of medical things, Peer-to-Peer Networking and Applications (2024) 1–20.

[16] M. Xu, Y. Zou, X. Cheng, Byzantine fault-tolerant wireless consensus, in: Wireless Consensus: Theory and Applications, Springer, 2024, pp. 95–140.
[17] N. Zafar, A. Khanna, S. Jain, Z. Ali, J. Ahamed, Safeguarding iot: Harnessing practical byzantine fault tolerance for robust security, in: International Conference on Data Analytics & Management, Springer, 2023, pp. 287–301.
[18] H. Wang, W. Tan, J. Wu, P. Liu, Opbft: Optimized practical byzantine fault tolerant consensus mechanism model, in: AI and Analytics for Public Health: Proceedings of the 2020 INFORMS International Conference on Service Science, Springer, 2022, pp. 123–135.
[19] C. Jatoth, R. Doriya, Iov block secure: blockchain based secure data collection and validation framework for internet of vehicles network, Peer-to-Peer Networking and Applications 17 (2024) 3964–3990.
[20] U. Singh, S. K. Sharma, M. Shukla, P. Jha, Blockchain-based batman protocol using mobile ad hoc network (manet) with an ensemble algorithm, International Journal of Information Security (2024) 1–11.
[21] N. Yang, X. Yang, R. Chai, C. Liang, Scalable blockchain-based access control algorithm for large-scale iot networks with byzantine nodes, in: International Conference on Communications and Networking in China, Springer, 2023, pp. 126–141.
[22] L. Sharma, R. K. Gupta, C. S. Lamba, A. Kumar, P. Lathar, Efficient practical byzantine consensus-based reputation method for iot based electronic waste tracking and tracing system using blockchain, Multimedia Tools and Applications (2024) 1–34.
[23] M. Wu, Y. Gao, Y. Xiao, Blockchain-aided access control for secure communications in ad hoc networks, in: Ad Hoc Networks: 11th EAI International Conference, ADHOCNETS 2019, Queenstown, New Zealand, November 18–21, 2019, Proceedings 11, Springer, 2019, pp. 87–98.
[24] N. Balani, P. Chavan, M. Ghonghe, Design of high-speed blockchain-based sidechaining peer to peer communication protocol over 5g networks, Multimedia Tools and Applications 81 (2022) 36699–36713.