IMPROVING IOT MANAGEMENT WITH BLOCKCHAIN: SMART HOME ACCESS CONTROL

Andrej Gono[1]✉, Ivo Pisařovic[1], Martin Zejda[1], Jaromír Landa[1], David Procházka[1]

[1] *Mendel University in Brno, Czech Republic*

EUROPEAN JOURNAL OF BUSINESS SCIENCE AND TECHNOLOGY

Volume 10 Issue 2
ISSN 2694-7161
www.ejobsat.com

ABSTRACT

Smart IoT devices, such as lights, locks, washing machines, security cameras, etc., are becoming omnipresent in households and companies across all industries. However, most of these devices communicate over non-secure local protocols or via cloud services where security policies are not transparent. Vulnerabilities may lead to unauthorized access to such IoT devices. Blockchain is a technology that brings security by design and can be exploited also in the area of controlling access to IoT devices. The goal of the paper is to test the use of blockchain with IoT devices to increase the security of device usage while ensuring that the user experience remains efficient and user-friendly. Three approaches to use blockchain are proposed and tested: a) application without the blockchain using standard HTTPS protocol; b) an application using blockchain, where users sign the transactions themselves; c) an application using blockchain where the server signs the transactions. The paper successfully shows that blockchain can be used to enhance IoT device security, with an focus on user-friendliness testing to ensure the solutions are practical for everyday use.

KEY WORDS

blockchain, IoT, access control, smart home, Ethereum, Bloxberg, Solidity, smart contracts

JEL CODES

C88

1 INTRODUCTION

In recent years, the Internet of Things (IoT) has emerged as a transformative technology, integrating the physical world with digital systems to enhance efficiency, data collection, and automation. This integration has proliferated in various sectors, including healthcare, agriculture, and smart homes, yielding significant benefits. However, with the expansion of IoT applications, the need for robust security mechanisms has become paramount, especially

GONO, Andrej, PISAŘOVIC, Ivo, ZEJDA, Martin, LANDA, Jaromír, and PROCHÁZKA, David. 2024. Improving IoT Management with Blockchain: Smart Home Access Control. *European Journal of Business Science and Technology,* 10 (2): 225–241. ISSN 2694-7161, DOI 10.11118/ejobsat.2024.012.

considering the sensitivity and volume of data involved (HaddadPajouh et al., 2021).

The introduction of blockchain technology presents a promising solution to these security challenges (Dorri et al., 2017). Blockchain, a decentralized and distributed ledger technology, is renowned for its robust security features, transparency, and immutability. By integrating blockchain with IoT, a secure, transparent, and efficient framework for managing IoT devices and the data they generate can be created. This integration not only enhances security but also fosters trust among users and stakeholders involved in IoT ecosystems (Hosseini et al., 2023).

This article aims to explore the potential of blockchain technology in providing secure access to IoT devices and to provide a methodology on how to use blockchain for the purpose of managing access to IoT devices. Many theoretical principles have been devised for these purposes, but guidance for practical implementation is often lacking. Our effort involves a multi-faceted approach, primarily focusing on the integration of smart contracts within the Node-RED[1] environment, a development tool for connecting hardware devices, APIs, and online services in new ways. The basic requirements for the solution are enhanced security, while speed and user-friendliness are not compromised.

This paper also focuses on the use of blockchain in academic and research environments, the authentication of research data, and the potential of this technology in the context of a metaverse environment. The metaverse is defined as the connection of the virtual world to the real world, which occurs through the use of IoT devices. Consequently, the management of IoT devices is the next area of focus. The article is structured as follows. Section 2 reviews common security flaws associated with IoT devices and the usage of blockchain to limit those flaws. Section 3 contains a description of the proposed usage of blockchain and the description of technologies used. Section 4 describes the implementation results with emphasis of the smart contract deployment and the description of proposed approaches to using blockchain in IoT. In section 5, the drawbacks of blockchain usage are discussed.

2 REVIEW

In the realm of IoT, security is a primary concern. IoT devices often collect and disseminate sensitive information, making them vulnerable to cyber-attacks. The inherent security attributes of blockchain technology, such as cryptographic encryption and decentralized data storage, play a pivotal role in fortifying this data (Dorri et al., 2017).

Several scholars have contributed to the understanding of these vulnerabilities. Unwala et al. (2018) discuss the critical period when a new device joins an IoT network, highlighting the necessity of robust authentication processes to prevent unauthorized access. They delve into the security features across various IoT protocols, such as Z-Wave and Thread, to underscore the complexity and depth of security measures required.

Additionally, Dragomir et al. (2016) examine the security capabilities of established IoT communication protocols and emphasize the need for standardized, interoperable security solutions to protect against increasingly sophisticated threats. This paper stresses the importance of industry collaboration and standardization efforts by groups like the IEEE and IETF to fortify IoT security.

The evolving nature of IoT security threats is further highlighted by Parashar and Rishishwar (2017), who discuss how the interconnectedness of IoT devices facilitates rapid information exchange but also increases susceptibility to hacking and service disruptions. Their research calls for scalable security solutions that do not compromise the operational efficiency of IoT systems.

[1] Available at `https://nodered.org/`.

Recent research also identifies specific modern threats to IoT environments, such as Distributed Denial of Service (DDoS), Man In The Middle (MITM), and replay attacks, which exploit the inherent vulnerabilities of IoT devices attacks (Dorobantu and Halunga, 2020; Lalit et al., 2022; Rajendran et al., 2019).

These studies underscore the complexity of IoT security and the need for a multi-layered approach to protect against both current and future threats. To address these threats, various security measures have been proposed, including lightweight security models and techniques, and countermeasures against potential attacks (Dorobantu and Halunga, 2020; Lalit et al., 2022; Rajendran et al., 2019).

Traditionally, IoT networks have been dependent on centralized models for data processing and storage, potentially engendering single points of failure. Blockchain, as a decentralized mechanism, obviates these vulnerabilities, thereby augmenting the reliability and robustness of IoT networks. This decentralization concurrently diminishes the risk of service interruptions, ensuring continuous operation (Dorri et al., 2017; Hosseini et al., 2023; Polat, 2023).

In the context of IoT, blockchain serves as a tool to secure integrated devices and their communication. It is used to authenticate devices on the network, where each device can have its own unique identifier in the form of a blockchain address. This address is used to secure communication between devices and to verify their transactions (Dorri et al., 2017).

The decentralized nature of the blockchain means that data is not stored in a single location, which eliminates the risk of attacks on individual data stores and increases the resilience of the network against outages. This is critical for IoT applications (Hosseini et al, 2023). Each transaction inscribed onto a blockchain is immutably recorded in a tamper-resistant manner, which significantly reduces the likelihood of data misuse. Blockchain technology not only guarantees the integrity of the data collected and disseminated by IoT devices, but also preserves the quality and reliability of the data, which is essential for decision-making processes in various industries (Polat, 2023; Ramesh et al., 2020).

Blockchain also enables the implementation of smart contracts, which are programs stored on the blockchain, triggered automatically when defined conditions are met. Within the IoT sphere, smart contracts can autonomously manage processes and interactions between devices, eliminating the need for human intervention. For instance, a smart thermostat could autonomously request and remunerate for heating oil as required, with the transaction securely recorded on a blockchain. Similarly, in the context of our project, it could authenticate access to smart locks on doors (Zheng et al., 2017).

The use of blockchain together with IoT has been explored in many papers. The authors in (Dorri et al., 2017) propose a novel blockchain-based architecture designed to provide lightweight, decentralized security and privacy for IoT without the overhead typically associated with blockchain technology. Their architecture is structured hierarchically and includes smart homes, an overlay network, and cloud storage, which coordinate data transactions using blockchain to ensure privacy and security. This architecture utilizes different types of blockchains depending on the network tier, which helps in reducing latency and computational demands that are typical in standard blockchain operations. The smart home tier uses a localized blockchain that requires no Proof of Work, thus, conserving resources. The overlay network facilitates data transactions between smart homes and external services with reduced overhead and increased privacy through clustering and selective transaction verification.

The paper from Alam (2019) discusses the integration of blockchain technology with the Internet of Things (IoT), focusing on how blockchain's secure, decentralized ledger enhances data security and communication among diverse, smart IoT devices, while also exploring the opportunities and challenges of this approach.

Belhadi et al. (2023) describes a blockchain-based system for improving security and efficiency in medical image segmentation within the Internet of Medical Things, using ensemble learning, genetic algorithms, and U-Net architectures, resulting in enhanced performance and robust security. In the case of security testing, blockchain has proven to be a reliable technology. It is mainly secured by asymmetric cryptography, where every operation to write to the blockchain or call a smart contract must be signed with a private key. This private key is typically encoded in a blockchain wallet, which also has a public key from which the wallet address is generated (Rawat et al., 2021).

Further proof that blockchain's significantly enhance IoT security, offering practical insights into blockchain's role in strengthening IoT access control mechanisms, is provided by Singh et al. (2023). In the paper, blockchain technology is applied to IoT access control, focusing on preventing cyber-attacks. It reviews various blockchain platforms for secure IoT access control and evaluates the efficacy of smart contracts in addressing security issues.

Sandner et al. (2020) depicts the synergistic use of blockchain, IoT, and AI, showcasing how blockchain ensures secure and transparent handling of IoT data, while AI analyzes this data for advanced decision-making.

Various applications across domains like smart agriculture, smart grid, smart home, smart transportation, banking, and finance are described in Tyagi et al. (2023). The paper also delves into the role of smart contracts and their functional architecture in these environments.

While these studies illustrate the broad applicability of blockchain in enhancing IoT security across various domains, the specific challenges and requirements of smart homes necessitate focused research. Smart homes are unique due to their integration of diverse, interconnected devices within a personal living space, requiring seamless and secure communication. This domain's specificity means that while general principles from other IoT applications can inform smart home security, dedicated solutions addressing the unique vulnerabilities and operational demands of smart homes are essential.

This paper focuses specifically on the role of blockchain in smart homes. For the implementation the smart contracts will be used. Finally, other technologies used in the practical part of the paper will be described.

2.1 Integrating Blockchain in Smart Homes for Enhanced Security and Automation

Smart homes, as the term suggests, are residences equipped with advanced technologies that enable automation and enhanced control over various home functions. These technologies encompass a range of applications such as energy management, healthcare, security, and automation of household tasks. Smart homes utilize IoT devices to monitor, control, and support residents, thereby improving their quality of life and promoting independent living. The key focus of smart home technology is to provide tailored services to users, optimizing comfort, efficiency, and safety within the home environment (Madakam and Ramaswamy, 2014).

Several security and privacy risks and challenges in the IoT, particularly in smart homes are identified in El-Azab (2021). These are the same security threats mentioned in the previous articles mentioned in the general IoT field.

Another article from Tyagi et al. (2023) proposes a new smart home gateway network architecture using blockchain technology. The proposed network is divided into three layers: device, gateway, and cloud, with blockchain employed at the gateway layer to ensure data integrity and security. The paper includes an experimental analysis demonstrating the effectiveness of this architecture over traditional centralized models.

Most of those and other articles that discuss the use of blockchain in smart homes make use of the Ethereum blockchain, as the most widely used and verified platform (Teutsch and Reitwießner, 2019).

2.2 Blockchain Platforms

In the context of our research, the comparison of multiple blockchain platforms to identify the most suitable one was performed. The analyzed platforms included Ethereum, Solana, Cardano, Polkadot, Bloxberg, Hyperledger Fabric and Binance Smart Chain (BSC). For each of these platforms, several factors such as algorithm

Tab. 1: Blockchain Platforms Comparison

Platform	Blockchain type	Algorithm	Theoretical speed (transactions per second)	Smart contract language
Ethereum	Public	Proof-of-stake	Dozens	Solidity
Cardano	Public	Proof-of-stake	Hundreds	Plutus
Solana	Public	Proof-of-history	Thousands	Rust
Bloxberg	Public	Proof-of-stake	Dozens	Solidity
Binance Smart Chain	Public	Proof-of-authority	Hundreds	Solidity
Hyperledger Fabric	Private	Pluggable consensus protocols	Thousands	Standard programming languages
Polkadot	Public	Nominated Proof-of-Stake (NPoS)	Thousands	

consensus, programming languages, transaction speed, cost of operation, and decentralization were considered.

Ethereum proved to be a very robust and reliable platform, mainly due to its Turing-complete programming language Solidity and strong developer community. It has switched to a proof-of-stake algorithm, reducing its energy consumption and increasing throughput. Ethereum is also compatible with a variety of tools and has great support for smart contracts and non-fungible tokens (NFTs). Solana, on the other hand, offers extremely high transaction speed thanks to its proof-of-history algorithm and sharding technologies. However, it faces security challenges and frequent outages, which reduce its reliability.

Cardano, like Ethereum, uses proof-of-stake and is known for its formal approach to development and scientific methodology. Its unique Ouroboros consensus system guarantees high scalability. Nevertheless, Cardano is not yet as stable and proven by various practical implementations. Binance Smart Chain combines proof-of-stake and proof-of-authority, which ensures high speed but also lower decentralization. Polkadot enables interoperability between different blockchains through its parachains, supporting a wide range of decentralized applications. However, neither Polkadot nor Binance Smart Chain are yet sufficiently stable and proven by various practical implementations, which limits their applicability for our project.

The Binance Smart Chain has proven to be a fast and reliable blockchain, but it is a network that is largely controlled by the commercial cryptocurrency exchange Binance. It is thus a blockchain that is largely centralized and thus could be unstable (Han et al., 2021). For the purposes of this research, it therefore appears unsuitable.

The same is true, although to a lesser extent, for the Cardano and Solana platforms. Ethereum's robust features, widespread adoption, and consistent updates make it sufficient for complex, long-lasting blockchain applications, also for the research projects. Bloxberg, originated as a fork of Ethereum, is more focused on experimentation in the academic sector.

3 METHODOLOGY

Based on the research above, the Bloxberg was selected as the most suitable platform for our project. Bloxberg is specifically adapted for academic purposes and uses proof-of-authority, which increases efficiency. It was developed by the prestigious German research organization Max Planck Society and has strong support from other major academic institutions. Thanks to its compatibility with Ethereum, it allows the use of smart contracts written in the Solidity language and other features from the Ethereum blockchain. This platform is ideal for secure and transparent storage and management of research data, which is crucial for our needs. Bloxberg also guarantees a high level of security and data integrity, which is essential for our project.

For our use case, smart contracts, programs stored on the Ethereum or Bloxberg blockchain that execute exactly according to code, are very important, eliminating the risks of outages, fraud, censorship, or outside interference. Their uniqueness lies in their capability to autonomously enforce predefined rules and penalties, similar to traditional contracts, but with added automation. In the realm of IoT, smart contracts have big potential for automating complex processes and interactions between devices, enhancing data integrity, and enabling new models of service delivery and automated decision-making, thereby revolutionizing IoT ecosystems with their self-executing and self-enforcing nature (Taherdoost, 2023; Antonopoulos and Wood, 2019).

Smart contracts are used in many fields, from Decentralized finance (Schär, 2021) or Academic and education sphere (Palma et al., 2019; Palma et al., 2020) to IoT (Lone and Naaz, 2021) and Smart Homes (Lee et al., 2020).

For writing smart contracts, Solidity language (2024) is used. It exemplifies a high-level, statically-typed programming paradigm. The Solidity design facilitates the implementation of self-enforcing business logic within smart contracts, ensuring immutable transaction records. Since Solidity is a very recent language, it is important to be aware of the security issues and threats that Solidity often suffers from, as mentioned in (Staderini et al., 2022).

For writing code and then testing and deploying smart contracts, the Remix IDE is used. REMIX IDE is an open-source, web-based integrated development environment specifically designed for Ethereum smart contract development. It facilitates the writing, testing, debugging, and deployment of smart contracts in Solidity. Key features of REMIX IDE include an integrated debugger, static analysis tools, and a user-friendly interface for deploying contracts on the Ethereum or Bloxberg blockchain. Its browser-based accessibility and comprehensive toolset make it ideal for both novice and experienced developers in the field of blockchain technology (Ethereum, 2024).

For more advanced testing and detailed analysis, Ganache is used. Ganache is a part of the Truffle Suite, serving as a personal, local blockchain for Ethereum or Bloxberg development. It allows developers to deploy contracts, develop decentralized applications (dApps), and run tests in a private, risk-free environment. Ganache provides key features like simulated blockchain transactions, contract execution, and block mining, with the flexibility of both a desktop application and a command-line tool (Ganache CLI). This tool is essential for developers to test and refine smart contracts before deploying them to the public blockchain network (Truffle Suite, 2024). For creating an architecture that connects IoT hardware devices, smart contracts, APIs, and other services (cloud and online services) it is appropriate to use Node-RED.

Node-RED is a popular open-source tool used in IoT and home automation. It offers a visual programming interface that simplifies creating applications and automation flows, even for those without extensive coding knowledge. Users can easily connect and control data and devices using pre-built blocks called nodes, which are connected to define data flow. These nodes support various functionalities and communication protocols like MQTT and HTTP, allowing integration with a wide range of IoT devices and services. Node-RED also facilitates real-time monitoring and debugging, making it easier to optimize workflows. It's versatile in deployment, running on platforms from Raspberry Pi to cloud services.

To be able to integrate smart contracts into this tool, an in-depth exploration of the capabilities of Node-RED was performed. This process was twofold: first, it involved a thorough examination of the various libraries and modules available within the Node-RED ecosystem that facilitate communication with blockchain networks. Second, a practical implementation of smart contracts in the Node-RED environment was created.

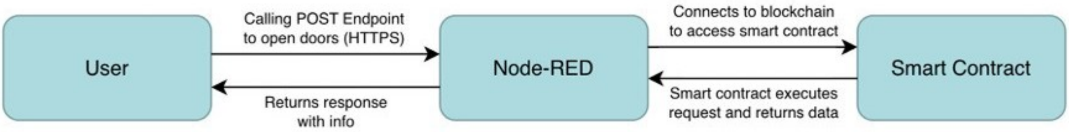

User
Calling POST Endpoint to open doors (HTTPS)
Returns response with info
Node-RED
Connects to blockchain to access smart contract
Smart contract executes request and returns data
Smart Contract

Fig. 1: Node-RED is the central point for controlling IoT devices. A user may interact with Node-RED that is connected to Smart Contract to verify the user's access

After the research and design phase, the development phase followed. The code for the smart contracts was created. This code, formulated in Solidity using the Remix IDE, was explicitly tailored to manage the locking and unlocking mechanisms of door locks. The smart contract on the Bloxberg blockchain was deployed using the Remix IDE tool.

Upon the successful development, next phase was the integration of this code within the Node-RED environment, see Fig. 1. This stage was critical in bridging the gap between the theoretical aspects of smart contracts and their practical application in real-world scenarios. The process involved deploying the smart contract to a test blockchain network, an essential step for verifying the functionality and security of the contract under simulated conditions. For this deployment, the Ganache tool, which provided a streamlined and controlled environment for testing, was utilized.

Parallel to the deployment, the appropriate Node-RED flow was created. This flow was intricately designed to interact seamlessly with the deployed smart contract, thereby facilitating the communication between the Node-RED interface and the blockchain network.

For our use case, a simple smart contract[2] showing the basic structure for securing access to a smart door was created. A new contract is deployed for each door, allowing for separate management. Thus, for each door, the contract may be slightly different and may contain different access rules. In a more complex system, there could be a single, large smart contract that also manages the individual IoT devices.

For creating an architecture that connects IoT hardware devices, smart contracts, APIs, and other services (cloud and online services) it is appropriate to use NODE-RED. Node-RED is a popular open-source tool used in IoT and home automation. It offers a visual programming interface that simplifies creating applications and automation flows, even for those without extensive coding knowledge. Users can easily connect and control data and devices using pre-built blocks called nodes, which are connected to define data flow. These nodes support various functionalities and communication protocols like MQTT and HTTP, allowing integration with a wide range of IoT devices and services. Node-RED also facilitates real-time monitoring and debugging, making it easier to optimize workflows.

4 RESULTS

4.1 Smart Contract Deployment

A smart contract that allows the owner to manage the list of users who have access to the home and allows these users to open and close door it the home based on their permissions was created. Events are used to inform others of events in the smart contract.

The contract was deployed through Remix IDE. For testing purposes, Remix IDE provides a way to deploy the contract to the virtual blockchain that runs on local device. In case of and deployment of a contract on a production blockchain, the blockchain wallets in a browser, for example MetaMask, through which we can sign and confirm the deployment of this

[2] The whole contract is available at `https://github.com/AndrejGono/IoTBlockchainPaper/blob/main/SmartHomeAccess.sol`.

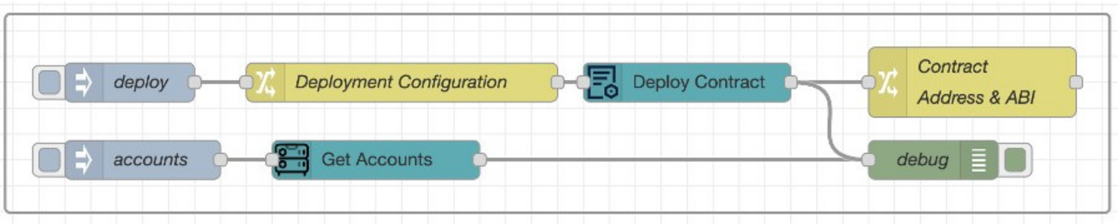


Fig. 2: A flow in NODE-RED for deploying a contract

contract needs to be integrated. However, the goal was to integrate this contract into Node-RED, so it was deployed automatically through the Remix IDE service.

To integrate the contract into Node-RED it was necessary to compile the contract in Remix IDE and find out what is its ABI (Application Binary Interface) and ByteCode, with which it will be possible to deploy the contract directly from Node-RED. The ABI acts as an interface between the smart contract and an external caller, such as applications or other contracts. ByteCode is a string of bytes that an Ethereum virtual machine (EVM) can execute. After writing and testing the code in Solidity, this code is compiled into ByteCode.

Ganache was used for contract deployment, which provides a local test blockchain on which the contracts to test transactions can be deployed.

In Ganache, it is necessary to create a new Ethereum or Bloxberg environment and Ganache will create a separate test blockchain, within which it will also create a list of Bloxberg addresses loaded with Bloxberg tokens. This blockchain is running on address *localhost* and port 7545.

Upon the creation of a contract and a blockchain for deployment, a new flow in Node-RED that deploys the contract automatically is also created (see Fig. 2).

This flow contains multiple nodes:

- Deploy: This node initiates the deployment process, serving as the control input that starts the operation when activated. This Note could be replaced by an HTTP Request node, which allows the contract to be deployed by calling a REST API endpoint.
- Deployment Configuration: This node contains predefined parameters essential for the deployment process, like bytecode, ABI and arguments for contract constructor.
- Deploy Contract: This node performs the deployment of the contract, executing the process using the provided configuration details to launch the contract into the desired environment.
- Contract Address & ABI: This node outputs the contract address and ABI, which are essential for interfacing with the deployed contract, providing a means for applications and users to interact with it.

Once a contract has been deployed, a flow is created that allows the methods of that contract to be called. Two HTTP endpoints are provided for users to call, thereby opening and closing doors. It is assumed that the user who calls this method has the necessary permissions to control the door. This flow can be seen in Fig. 3.

- [post] */open:* This HTTP input node listens for POST requests on the /open endpoint. When a request is received, it triggers the

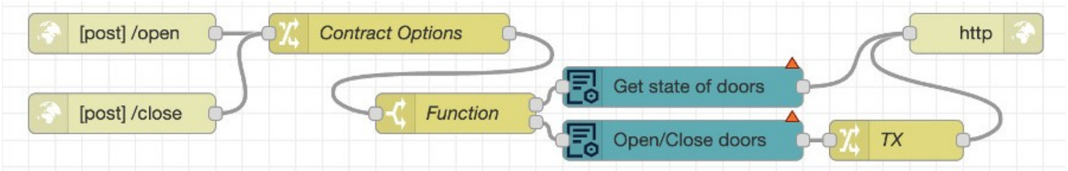


Fig. 3: A flow in Node-RED for calling functions to open and close the doors

flow to call a function on the smart contract that checks whether the provided address has access rights. If access is granted, the function will proceed to open the doors.

- Contract Options: Configures the necessary parameters to interact with the smart contract, such as the smart contract's address, ABI, gas limit for Ethereum or Bloxberg transactions and also translates http body attributes to flow attributes, so other nodes know what to do.
- Function: This node constructs the transaction object to interact with the smart contract. It contains the logic to call the 'open door' function within the smart contract, which includes specifying the function name and arguments based on the incoming request data.
- Get state of doors & Open/Close doors: These nodes execute the interaction with the smart contract. Since the 'open' action involves changing the state (by granting access and opening the door), it would utilize the Open/Close doors node to initiate a transaction on the blockchain. If there's a 'call' to simply check the status of blockchain without changing the state, it would go through the Get state of doors node.
- TX: This node is set to process the transaction. It would handle the transaction response from the blockchain network, or event trigger which includes a transaction hash or additional information.
- HTTP: This node sends the HTTP response back to the requester. It would communicate the outcome of the request, indicating whether the door was successfully opened or if the request was denied.

The corresponding process for closing the door would be similar using the close endpoint.

4.2 Proposed Secured IoT Approaches using Blockchain

It is logical that adding security measures will reduce the speed and user-friendliness of the solution. Our implementation contains a simple process where the user calls an HTTP endpoint on some server via a web or mobile app, which then calls a function on the IoT device. In our case, the smart doors that are opened or locked by the call. We have included an intermediary in this process, which is the blockchain, or smart contract, which is deployed on this blockchain and serves as middleware. The smart contract performs an additional verification that the user with the blockchain address through which this contract calls has indeed been granted permissions. In this case, the blockchain acts as a database in which the blockchain addresses of users are stored, and each one is assigned information about whether it has access to a given door. Communication through this intermediary logically slows down the whole process, because the blockchain acts as a robust and secure, but slow, database. Much depends on the use of the particular network. The Bloxberg, which is one of the fastest networks because new blocks are created in it and added to the network every 7 seconds was used. Every transaction, and therefore call or operation performed by a smart contract, is stored in a block within a maximum of 10 seconds, and thus is forever written in the history of the blockchain.

In order to assess the efficacy of proposed approaches, a solution that makes use of blockchain was put in a contrast with a solution that does not. Concurrently, a balanced approach that would permit the utilisation of blockchain and its advantages while maintaining user-friendliness and simplicity was proposed. This balanced approach is an application that employs blockchain, wherein transactions are not signed by end-users, as is customary, but rather, the private key for signing transactions is stored on the server that users interact with.

Three variants for architecture have been proposed: 1) an application without the use of blockchain; 2) an application using blockchain, where users sign the transactions themselves; 3) an application using blockchain where the server signs the transactions.

The architecture of each solution can be seen in Fig. 4.

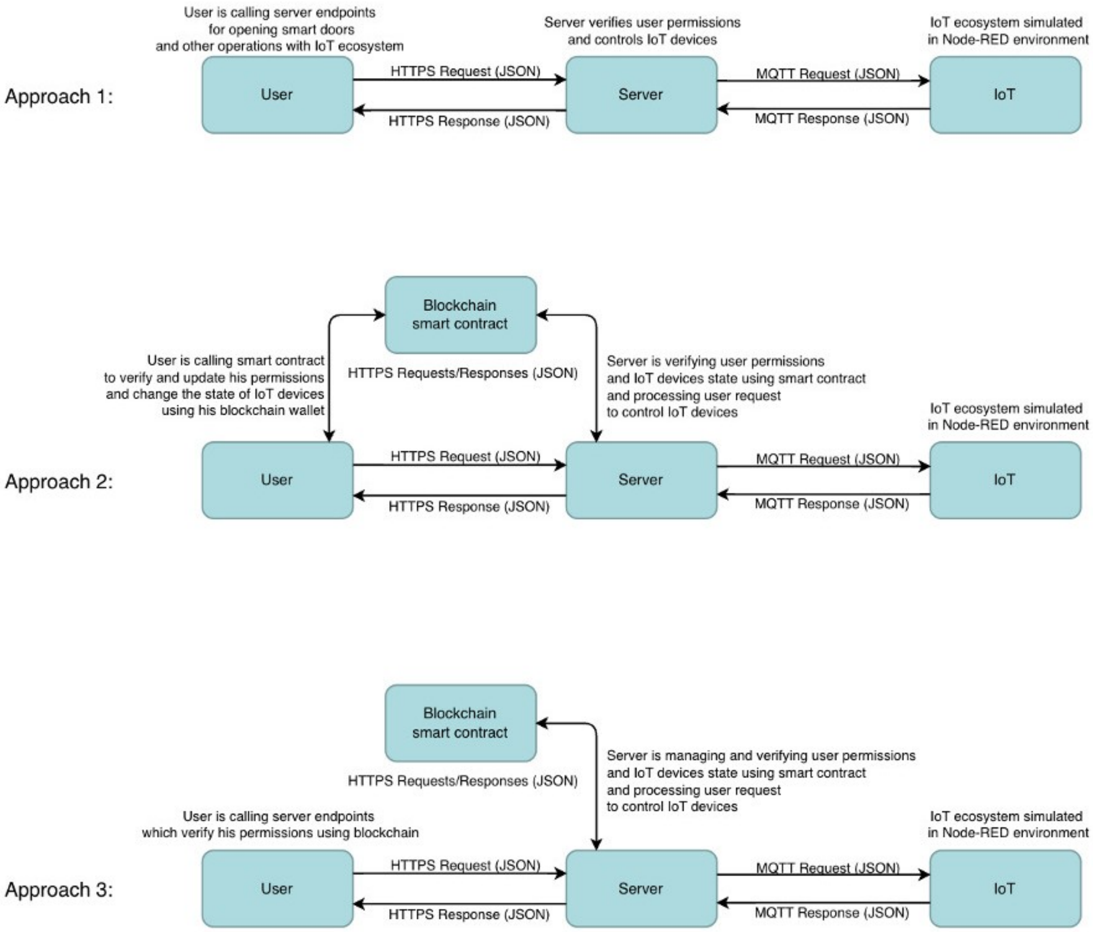

User is calling server endpoints
for opening smart doors
and other operations with IoT ecosystem
Server verifies user permissions
and controls IoT devices
IoT ecosystem simulated
in Node-RED environment
Approach 1:
User
HTTPS Request (JSON)
HTTPS Response (JSON)
Server
MQTT Request (JSON)
MQTT Response (JSON)
IoT
Blockchain
smart contract
User is calling smart contract
to verify and update his permissions
and change the state of IoT devices
using his blockchain wallet
HTTPS Requests/Responses (JSON)
Server is verifying user permissions
and IoT devices state using smart contract
and processing user request
to control IoT devices
IoT ecosystem simulated
in Node-RED environment
Approach 2:
User
HTTPS Request (JSON)
HTTPS Response (JSON)
Server
MQTT Request (JSON)
MQTT Response (JSON)
IoT
Blockchain
smart contract
HTTPS Requests/Responses (JSON)
Server is managing and verifying user permissions
and IoT devices state using smart contract
and processing user request
to control IoT devices
User is calling server endpoints
which verify his permissions using blockchain
IoT ecosystem simulated
in Node-RED environment
Approach 3:
User
HTTPS Request (JSON)
HTTPS Response (JSON)
Server
MQTT Request (JSON)
MQTT Response (JSON)
IoT

Fig. 4: The architecture of proposed solutions

The approach 1 is a standard communication model, without the use of blockchain. This approach is fast, but not as secured as when using blockchain.

The approach 2 is using blockchain, where each user has their own blockchain wallet, which they manage and protect themselves. This approach is very secure but slow and thus not suitable for dynamic things like smart doors. On the other hand, this approach is optimal for more tech-savvy users and for a sensor that does not store data very often (e.g. only once per hour). However, this complicates the whole process because it introduces an extra step. In order to interact with the blockchain, the user must download a wallet. When calling a smart contract, the user must provide his blockchain address and sign this operation with his private key. This work is handled for the user by their blockchain wallet, such as Metamask, which is open-source and generally considered secure and reliable. However, the user has a large number of other wallets to choose from. For some users, however, this may be a no-go, as they will not have the technical knowledge to operate the blockchain wallet and sign transactions through it.

The approach 3 is focused on an alternative to usage of blockchain. The users use a classic API and authentication via the blockchain is performed by the server.

To test the security of the approach, the call to a transaction with the user's address was signed with a different, custom private key.

In this case, the node does not accept the transaction at all and rejects it with an error message. This is because authentication works on the principles of asymmetric cryptography, where the blockchain address serves as the public key (it is an encrypted public key) and each blockchain wallet has its private key encrypted belonging to it. Thus, in order to impersonate another user, their private key would have to obtained. The key is stored securely on the user's device and is managed by the wallet or blockchain application they use.

4.3 User Testing of the Designed Application

As stated in the introduction, the primary objective of the proposed solution is to enhance the security of the system through the utilisation of blockchain technology, while maintaining the user-friendliness of the system. As previously outlined in the research, the utilisation of blockchain introduces an additional intermediate authentication step to the overall system, thereby enhancing the overall security. However, this intermediate step results in a more complex and time-consuming process within the application, for example, when unlocking a smart IoT door.

Consequently, user testing was conducted to ascertain whether typical users would be able to utilise and operate a blockchain application without the experience being negatively affected. Additionally, the objective was to ascertain whether individuals lacking familiarity with blockchain or the technology in question would be able to utilise the blockchain application. Furthermore, the testing was designed to ascertain user perceptions regarding the use of blockchain in terms of security and the time required to complete the process. As the blockchain authentication process comprises a number of distinct stages, the testing also sought to ascertain which of these stages proved most challenging for users and which constituted a barrier to their engagement.

A total of 43 respondents participated in the testing phase. All participants were required to solve a hypothetical scenario in which they are in possession of a door equipped with an Internet of Things (IoT) lock, which they are able to control via a mobile application. A prototype of the mobile app was created in Figma, in which the entire process was simulated. Users were required to operate the app iteratively in order to complete the process of unlocking and opening the door.

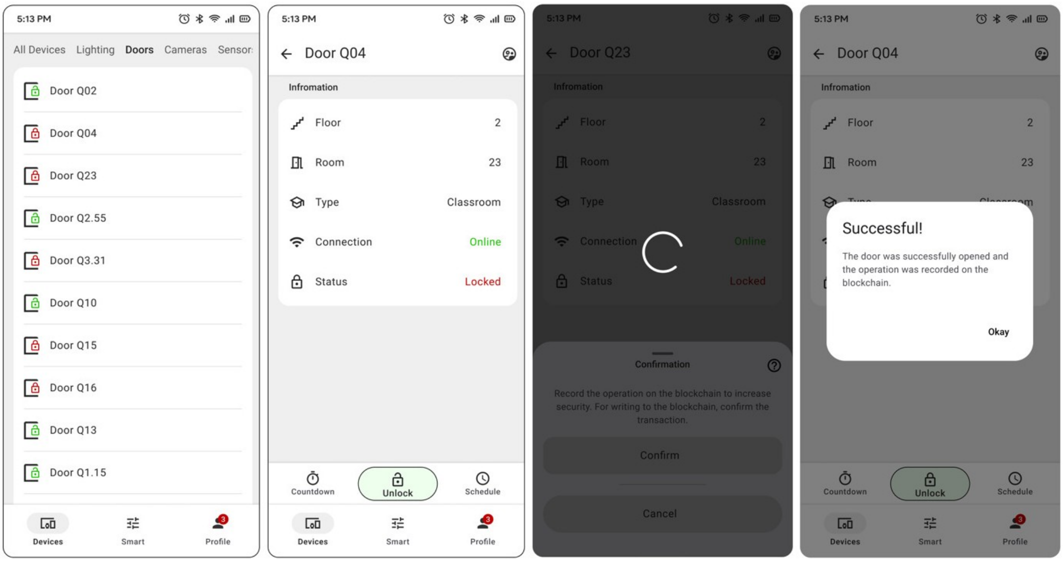
The respondents represented the productive segment of the population, aged 20 to 60. They were categorised according to their educational background as follows: college technical (STEM), college non-technical (humanities), and no college degree.

A control group of ten respondents was tasked with unlocking an IoT door in an application without utilising a blockchain. In the context of the proposed solutions, this constituted Approach 1, which is commonly employed in classical blockchain-free solutions, whereby the user controls the IoT device by connecting to a server. The application is illustrated in Fig. 5.

The remaining 33 respondents were provided with a prototype mobile application, which was an implementation of the proposed Approach 2. This approach was selected due to its robust security and comprehensive user experience, which increases the potential for user error during the door unlocking process. The rationale behind the robust security measures is that the user retains control of the entire process, eliminating the need to delegate any functions to the server. This approach is therefore recommended.

The IoT door unlocking process with blockchain commences in a similar manner to the process without blockchain, but involves a few additional steps. These supplementary steps can be observed in Fig. 6.

In the initial phase of blockchain-based authentication, the user was required to select the appropriate blockchain wallet with which to sign the transaction. The signing of the transaction was conducted in a separate application, which is the reason for the discrepancy in visual style between the second and fourth screens in Fig. 6. This is a distinct application, designated as Metamask, to which the user was redirected.



Infromation

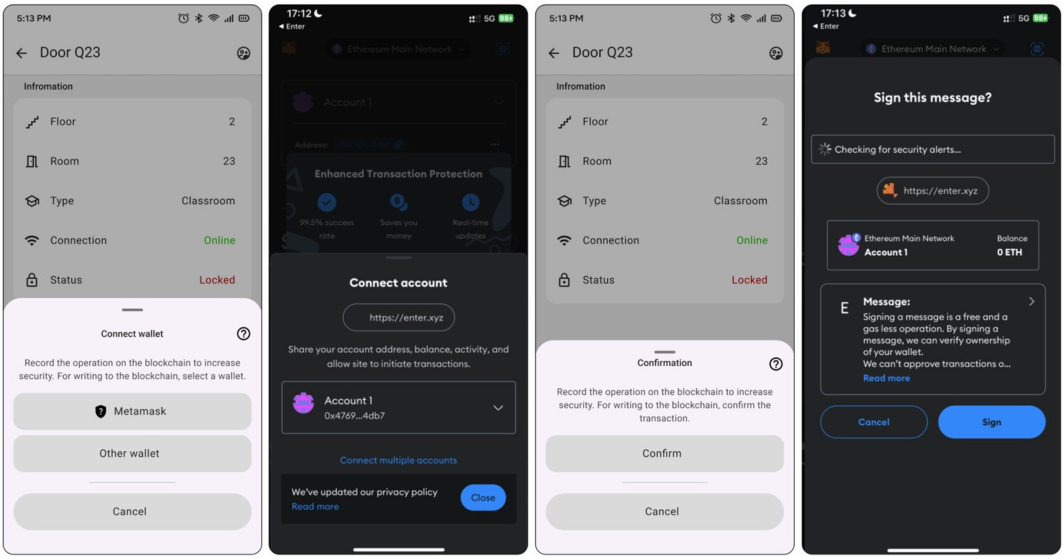
Floor 2
Room 23
Type Classroom
Connection Online
Status Locked

Record the operation on the blockchain to increase security. For writing to the blockchain, confirm the transaction.

Successful!

The door was successfully opened and the operation was recorded on the blockchain.

Fig. 5: Application for user testing with Approach 1 (without blockchain)



Infromation

Floor 2
Room 23
Type Classroom
Connection Online
Status Locked

Record the operation on the blockchain to increase security. For writing to the blockchain, select a wallet.

https://enter.xyz

Share your account address, balance, activity, and allow site to initiate transactions.

Account 1
0x4769...4db7

We've updated our privacy policy

Record the operation on the blockchain to increase security. For writing to the blockchain, confirm the transaction.

Checking for security alerts...

https://enter.xyz

Ethereum Main Network
Account 1
Balance 0 ETH

Message:
Signing a message is a free and a gas less operation. By signing a message, we can verify ownership of your wallet.
We can't approve transactions o...

Fig. 6: Application for user testing with Approach 2 (with blockchain)

The wallet is capable of accommodating multiple accounts. The specific account is selected by the user on the second screen of Fig. 6. Following the selection of the account, the user is prompted for confirmation, as illustrated on the third screen. Upon confirmation, the user proceeds to sign a transaction, which is a smart contract on the blockchain, to verify their permissions (see Fig. 4).

4.4 Evaluation of Testing

Of the 33 respondents included in the test group, 32 were able to complete the entire verification process. Of the 33 respondents, 14 were previously unacquainted with blockchain technology, having no prior awareness of it. Another 12 respondents had heard of blockchain technology but had not utilized it.

A total of 13 respondents from the test group indicated that they had not completed a college education. A further four respondents held a non-technical college degree.

Given this data, the use of blockchain appears to be suitable for people without a university or technical education, and also for people without any experience with blockchain.

The average completion time for the process in the group that did not utilise blockchain technology was 19.7 seconds. In contrast, the group that employed blockchain technology and was required to undergo multiple verification steps took, on average, 59 seconds to complete the process. The entire process was completed in a maximum of 120 seconds. This is, therefore, almost three times longer.

For this reason, respondents in the blockchain group were asked how they perceived the process in terms of time. Only one respondent, out of 33, found the process to be very efficient, stating that it was fast and minimally time consuming. 12 respondents considered the process to be efficient, stating that it was relatively quick and that the time invested was reasonable. 12 respondents considered the time to be less efficient, indicating that although the process was longer than anticipated, it was still manageable. Conversely, 7 respondents considered the time to be inefficient, indicating that the process was too long and that the time invested was unreasonable.

The time required to complete the process has an adverse effect on the user-friendliness of the application, which may ultimately discourage users from utilising it. In light of the fact that there is no means of accelerating the verification of a transaction on the blockchain, it is evident that none of the aforementioned steps can be circumvented. Consequently, there is limited scope for improvement. Further testing would be required to ascertain whether users are willing to accept this time frame. In the event that they are not, we recommend utilising Approach 3 (see Fig. 4), whereby a portion of the process is conducted on the server, thereby reducing the overall time required for the user.

In order to ascertain whether users would be amenable to an extension of the allotted time, respondents were invited to provide their assessment of the application in terms of its user-friendliness. A total of 29 out of 33 respondents rated the application as excellent, good, or fair. Four respondents provided negative ratings, citing a lack of understanding and the difficulty of the process, not just the blockchain component, as the primary reasons for their assessment. These respondents had no university degree and no experience with IoT and blockchain technologies.

Additionally, while testing, users were afforded the opportunity to access the app's help functionality should they require clarification regarding the appropriate course of action in a given scenario. The tooltip provided an explanation of relevant blockchain terminology, facilitating a more comprehensive understanding of its advantages. However, only 10 respondents availed themselves of this assistance. The majority of respondents demonstrated sufficient understanding to complete the process independently, even if they lacked prior experience with blockchain.

5 DISCUSSION

Adding blockchain as an intermediary in the process of controlling smart doors has increased security at the expense of speed and user-friendliness, which is in line with expectations. Performance testing has shown that each transaction through the blockchain takes a minimum of 7 seconds, which may be limiting for some applications.

Performance testing is usually done using load tests, where a method or process is run many times in a row, and latency and response time are monitored under this increased load. However, such testing is more challenging in the case of blockchain testing because the same transaction (or smart contract call) from a single blockchain address to the blockchain can only be send once per block. A second transaction can only be send after the previous transaction has been included in the block and that block has been verified by the blockchain node the application is communicating with. This means the user can call the smart contract method only once in each block, that is, only once every 7 seconds. This is a common security measure of most cryptocurrencies, where the possibility of sending multiple transactions in a single block could lead to multiple spending of identical funds. Thus, in practice, there will never be an "overload" of blockchain requests from a single user.

However, it may happen that thousands of users will interact with the blockchain or a certain smart contract at one moment. This could significantly slow down the machine that processes the transaction, and therefore increase the waiting time of 7 seconds for a transaction to be executed significantly. Again, this is virtually impossible to test because the blockchain itself runs on a large number of nodes, and it is never know which node a user will be communicating with. Ideally, the user communicates with the closest node, but the blockchain also tries to distribute the network communication among the other nodes. How a given node handles the load depends on its hardware parameters, and of course each node is different. However, blockchain networks, whether Bitcoin, Ethereum or Bloxberg, work on the principle that they contain what is called a mempool, which is a small database containing transactions to be included in a new block and verified. If a blockchain node receives more transactions than it can include in a block, it simply stores the remaining transactions temporarily in the mempool and verifies them in the next block.

Again, it is not possible to determine the exact number of transactions that will fit in a single block. Bloxberg, like Ethereum, does not have a fixed limit block size in megabytes, like Bitcoin, for example. Instead, it uses the concept of "gases", where each transaction costs a certain amount of gases depending on what computational resources are needed to execute it, and each block has a gaseous limit that determines how many transactions can fit in it. The gas limit for a block changes dynamically depending on the decisions made by the miners (in the case of Proof-of-Authority on Bloxberg, the universities make the decisions) and the current network. The gas limit for a block is in the range of millions of gas units. The typical gas limit per block is around 15 million gas. A simple transaction as well as our transactions for locking/unlocking doors have around 21,000 gas. This means that theoretically, if a block contained only simple transfer transactions, a block would fit approximately 714 transactions (15,000,000/21,000 = 714). In practice, however, blocks contain a mix of different transaction types, and the actual number of transactions in a block will depend on their specific complexity and the amount of gas used. In situations where data retrieval from the blockchain is the sole objective, waiting for a new block is unnecessary, and immediate access to the data is available. However, in the context of the paper and in the majority of cases, there is an additional requirement to write data to the blockchain. For instance, after unlocking or locking a door, it becomes necessary to record its current state on the blockchain. Unfortunately, in such scenarios, users must patiently await the inclusion of this operation in a block. What can be said for sure is that each user transaction takes a minimum of 7 seconds. If the network is congested and the nodes cannot keep up with adding all new transactions to a block (the blocks are full), the user may wait several times longer for a transaction. If the user should wait longer than a few seconds to unlock the door, such an application does not make sense and will certainly not be accepted by the normal user. A solution could be the use of private blockchains, in which new blocks would be included in the block more frequently.

6 CONCLUSION

This research demonstrated the effective integration of blockchain technology with Internet of Things (IoT) devices, focusing on securing access control in smart homes, specifically through smart door locks. The practical application involved using Bloxberg smart contracts within the Node-RED environment to enable secure and autonomous control of smart door locks. This integration showed a substantial improvement in IoT security, leveraging tools like Remix IDE and Ganache for smart contract development and testing.

The principal contributions to the field are the results of the user testing. The user testing, which involved 43 participants, revealed that 97% of users could complete the blockchain-based verification process within an average time of 59 seconds, indicating that it is a feasible technology for real-world applications. Regardless of their technical background, users were able to operate the blockchain-enabled app with proficiency.

Three architectural approaches for integrating blockchain with the Internet of Things (IoT) were put forth for consideration. A solution that does not utilise blockchain technology, a fully decentralised user-managed blockchain solution and a blockchain solution that is assisted by a server.

The fully decentralised approach offers robust security, but it can be complex and time-consuming for users. The server-assisted approach strikes a balance between security and usability by offloading certain processes to a server, thereby rendering it more practical for everyday use. The findings indicate that blockchain-based applications have the potential to improve the security of IoT systems without negatively impacting the user experience. The server-assisted approach, in particular, represents a promising solution for industries seeking to develop secure and user-friendly IoT applications.

7 ACKNOWLEDGEMENTS

This paper was supported by grant No. IGA-PEF-TP-22-006 (Opportunities to use metaverse technology to support business processes) of the Internal Grant Agency FBE MENDELU and by project CZ.02.1.01/0.0/0.0/16_017/0002334 Research Infrastructure for Young Scientists co-financed by Operational Programme Research, Development, and Education.

8 REFERENCES

Alam, T. 2019. Blockchain and its Role in the Internet of Things (IoT). *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 5 (1), 151–157. DOI: 10.32628/CSEIT195137.

Antonopoulos, A. M. and Wood, G. 2019. *Mastering Ethereum: Building Smart Contracts and DApps*. 1st ed. O'Reilly.

Belhadi, A., Holland, J.-O., Yazidi, A., Srivastava, G., Lin, J. C.-W. and Djenouri, Y. 2023. BIoMT-ISeg: Blockchain Internet of Medical Things for Intelligent Segmentation. *Frontiers in Physiology*, 13, 1097204. DOI: 10.3389/fphys.2022.1097204.

Dorobantu, O. G. and Halunga, S. 2020. Security Threats in IoT. In *2020 International Symposium on Electronics and Telecommunications (ISETC)*, pp. 1–4. DOI: 10.1109/ISETC50328.2020.9301127.

Dorri, A., Kanhere, S. S. and Jurdak, R. 2017. Blockchain in Internet of Things: Challenges and Solutions. *arXiv*. DOI: 10.48550/arXiv.1608.05187.

Dragomir, D., Gheorghe, L., Costea, S. and Radovici, A. 2016. A Survey on Secure Communication Protocols for IoT Systems. In *2016 International Workshop on Secure Internet of Things (SIoT)*, pp. 47–62. DOI: 10.1109/SIoT.2016.012.

El-Azab, R. 2021. Smart Homes: Potentials and Challenges. *Clean Energy*, 5 (2), 302–315. DOI: 10.1093/ce/zkab010.

Ethereum. 2024. *Remix – Ethereum IDE* [online]. Available at: https://remix.ethereum.org. [Accessed 2024, January 3].

HaddadPajouh, H., Dehghantanha, A., Parizi, R. M., Aledhari, M. and Karimipour, H. 2021. A Survey on internet of Things Security: Requirements, Challenges, and Solutions. *Internet of Things*, 14, 100129. DOI: 10.1016/j.iot.2019.100129.

Han, J., Huang, S. and Zhong, Z. 2021. Trust in DeFi: An Empirical Study of the Decentralized Exchange. *SSRN Electronic Journal*. DOI: 10.2139/ssrn.3896461.

Hosseini, S. M., Ferreira, J. and Bartolomeu, P. C. 2023. Blockchain-Based Decentralized Identification in IoT: An Overview of Existing Frameworks and Their Limitations. *Electronics*, 12 (6), 1283. DOI: 10.3390/electronics12061283.

Lalit, M., Chawla, S. K., Rana, A. K., Nisar, K., Soomro, T. R. and Khan, M. A. 2022. IoT Networks: Security Vulnerabilities of Application Layer Protocols. In *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, pp. 1–5. DOI: 10.1109/MACS56771.2022.10022971.

Lee, Y., Rathore, S., Park, J. H. and Park, J. H. 2020. A Blockchain-Based Smart Home Gateway Architecture for Preventing Data Forgery. *Human-centric Computing and Information Sciences*, 10, 9. DOI: 10.1186/s13673-020-0214-5.

Lone, A. H. and Naaz, R. 2021. Applicability of Blockchain Smart Contracts in Securing Internet and IoT: A Systematic Literature Review. *Computer Science Review*, 39, 100360. DOI: 10.1016/j.cosrev.2020.100360.

Madakam, S. and Ramaswamy, R. 2014. Smart Homes (Conceptual Views). In *2014 2nd International Symposium on Computational and Business Intelligence*, pp. 63–66. DOI: 10.1109/ISCBI.2014.21.

Palma, L. M., Vigil, M. A. G., Pereira, F. L. and Martina, J. E. 2019. Blockchain and Smart Contracts for Higher Education Registry in Brazil. *International Journal of Network Management*, 29 (3), e2061. DOI: 10.1002/nem.2061.

Palma, L. M., Vigil, M. A. G. and Martina, J. E. 2020. Blockchain-Based Academic Record System. In *2020: Anais Estendidos do XX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pp. 49–56. DOI: 10.5753/sbseg_estendido.2020.19269.

Parashar, A. and Rishishwar, S. 2017. Security Challanges in IoT. In *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 446–449. DOI: 10.1109/AEEICB.2017.7972351.

Polat, Y. 2023. *Build a Blockchain Network for Trusted IoT: Use Hyperledger Fabric and Node-RED to Create a Blockchain Network to Secure IoT Data* [online]. IBM Developer. Available at: https://developer.ibm.com/patterns/build-a-blockchain-network-for-trusted-iot

Rajendran, G., Nivash, R. S. R., Parthy, P. P. and Balamurugan, S. 2019. Modern Security Threats in the Internet of Things (IoT): Attacks and Countermeasures. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–6. DOI: https://doi.org/10.1109/CCST.2019.8888399.

Ramesh, V. K. C., Kim, Y. and Jo, J.-Y. 2020. Secure IoT Data Management in a Private Ethereum Blockchain. In *IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 369–375. DOI: 10.1109/COMPSAC48688.2020.0-219.

Rawat, D. B., Chaudhary, V. and Doku, R. 2021. Blockchain Technology: Emerging Applications and Use Cases for Secure and Trustworthy Smart Systems. *Journal of Cybersecurity and Privacy*, 1(1), 4–18. DOI: 10.3390/jcp1010002.

Sandner, P., Gross, J. and Richter, R. 2020. Convergence of Blockchain, IoT, and AI. *Frontiers in Blockchain*, 3, 522600. DOI: 10.3389/fbloc.2020.522600.

Schär, F. 2021. Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets. *Federal Reserve Bank of St. Louis Review*, 103 (2), 153–174. DOI: 10.20955/r.103.153-74.

Singh, R., Kukreja, D. and Sharma, D. K. 2023. Blockchain-Enabled Access Control to Prevent Cyber Attacks in IoT: Systematic Literature Review. *Frontiers in Big Data*, 5, 1081770. DOI: 10.3389/fdata.2022.1081770.

Solidity. 2024. *Solidity Programming Language* [online]. Available at: `https://soliditylang.org/`. [Accessed 2024, January 3].

Staderini, M., Pataricza, A. and Bondavalli, A. 2022. Security Evaluation and Improvement of Solidity Smart Contracts. *SSRN Electronic Journal*. DOI: 10.2139/ssrn.4038087.

Taherdoost, H. 2023. Smart Contracts in Blockchain Technology: A Critical Review. *Information*, 14 (2), 117. DOI: 10.3390/info14020117.

Teutsch, J. and Reitwiessner, C. 2019. A Scalable Verification Solution for Blockchains. *ArXiv*. DOI: 10.48550/arXiv.1908.04756.

Truffle Suite. 2024. *Ganache* [online]. Available at: `https://trufflesuite.com/ganache`. [Accessed 2024, January 3].

Tyagi, A. K., Dananjayan, S., Agarwal, D. and Thariq Ahmed, H. F. 2023. Blockchain–Internet of Things Applications: Opportunities and Challenges for Industry 4.0 and Society 5.0. *Sensors*, 23 (2), 947. DOI: 10.3390/S23020947.

Unwala, I., Taqvi, Z. and Lu, J. 2018. IoT Security: ZWave and Thread. In *2018 IEEE Green Technologies Conference (GreenTech)*, pp. 176–182. DOI: 10.1109/GreenTech.2018.00040.

Zheng, Z., Xie, S., Dai, H., Chen, X. and Wang, H. 2017. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pp. 557–564. DOI: 10.1109/BigDataCongress.2017.85.

AUTHOR'S ADDRESS

Andrej Gono, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: andrej.gono@mendelu.cz (corresponding author)

Ivo Pisařovic, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: ivo.pisarovic@mendelu.cz

Martin Zejda, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: martin.zejda@mendelu.cz

Jaromír Landa, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: jaromir.landa@mendelu.cz

David Procházka, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: david.prochazka@mendelu.cz