BFTM: Blockchain-Based MFA Framework for Secure Financial Transfers

Maha Alwan Alteef and Maytham Mustafa Hammood
Department of Computer Science, College of Computer Science and Mathematics, University of Tikrit, 34001 Tikrit, Salah Al-Din, Iraq
Ma230055pcm@st.tu.edu.iq, maythamhammood@tu.edu.iq

Keywords: Blockchain Authentication, Decentralized MFA, TOTP Verification, Secure Fund Transfers.

Abstract: This paper proposes a blockchain-based multi-factor authentication (BMFA) framework designed to enhance the security of financial operations in decentralized systems. Most authentication methods are still vulnerable to phishing, getting SIM cards switched, and stealing your credentials. By utilizing TOTP and a blockchain platform called Ganache with Ethereum support, wallet owners are allowed to confirm and perform transactions in a reliable, decentralized method. It adds main features for users, so they can declare, cancel, or reject transactions on their end, which makes the process more reliable and gives users the right to be involved. A Flask server off the blockchain manages the user signup and login, and generates TOTP codes for the Google Authenticator app, while Web3.js enables easy communication online with the network. A total of 50 MetaMask wallets were subjected to strict tests created to imitate phishing schemes and server breaches. The framework showed it is fully reliable, as it stopped every unauthorized transaction that occurred. Besides that, it completed a transaction in an average of 1.2 seconds and handled up to 25 transactions each second in controlled tests. The work suggests a powerful, unbreakable, and user-friendly setup for ensuring safety in digital asset transactions in networks not controlled by a single authority.

1 INTRODUCTION

With digital financial services on the rise, keeping user authentication safe and dependable is essential. While many systems rely on password authentication, it can be attacked by brute force, phishing, and credential theft. No matter what standard hashing methods are used, users and companies have started using Multi-Factor Authentication, especially with Time-based One-Time Passwords (TOTP). Yet, using MFA on mobile devices is vulnerable to phishing, swapping someone's SIM card, and hijacking a session, making it possible for attackers to gain access even with several checks [1], [2]

Most MFA systems are missing security measures that come after authentication. Notably, previous solutions that combined blockchain and MFA did not have much support for managing transactions by reversing, cancelling, or rejecting them. Consequently, once somebody has stolen a user's login, it's difficult for the victim to block such a transaction. In addition, biometric solutions are problematic since they raise privacy and security concerns, depend on specialized equipment, and include challenges that our work bypasses [3].

The proposal is to use blockchain and combine TOTP with smart contract technology on an Ethereum-compatible platform. Our new system offers users more control with cancellation from the sender, rejection from the receiver, and validation steps along the way. As a result, users are protected by swift, flexible safeguards for compromised user data, which also helps the system overcome weaknesses. The system is checked in real settings, and it is found that it can tackle phishing and SIM attacks with low latency and the ability to handle increased workloads. Merging blockchain security and MFA, we have developed an improved way of authenticating users by giving digital environments what they need most.

2 BACKGROUND AND RELATED WORK

2.1 Theoretical Background

MFA is commonly used because it requires users to present different kinds of credentials, such as a password, a token, or biometric data [4]. With biometric authentication, it is easy to create a unique proof of identity, which is being added to MFA systems more frequently. Biometrics are hard to copy because they are unique, but anyone who sees your biometric data can still access it whenever necessary [5]

MFA works to prevent risks like people stealing login details and unapproved access to resources. For example, since SIM swappers cannot obtain a token in your possession, they will not be able to gain access to your phone number. For instance, MFA is found in banking applications, secure entry points, and mobile IDs that use Mobilt BankID in Sweden, which identifies a user and tracks their movement with GPS and how they act on the device [6]. On the other hand, MFA can bring about problems such as being difficult for some users, being too costly, needing specific devices, and facing issues with compatibility with older systems [7]

In addition, it is necessary for cryptography to handle vast usage, mainly for business, government, and crime investigation systems, while still ensuring adequate privacy [8]. In order to make transactions safe and transparent and not able to be changed, blockchain technology was developed. When Bitcoin was created in 2008, blockchain technology was developed for use beyond digital currency [9]. Within a blockchain system, only valid transactions are set in stone, guaranteed by consensus across many nodes, and each block is linked to the previous one using hash functions [10]. Most definitions of blockchain point out that it is decentralized, unchangeable, and transparent [11], [12] . Blockchain is used in digital voting, public services, education, music, and mainly in cybersecurity [13] - [17].

These three main qualities of blockchains handle security challenges in any electrical system. Thanks to its unchangeable design, blockchain stops anyone from altering authentication records, and distributed authentication makes the system less likely to fail. When blockchain is used with MFA, identity verification is strengthened by logging information that can't be changed and trusting many sources. As a result, credential reuse, man-in-the-middle attacks, and data breaches can be prevented or reduced [19], [20].

Researchers suggest a way for users to use such systems by making a pair of public–private keys and connecting their credentials to the blockchain. The network checks the request using a series of methods and allows access if it passes the validation tests [21] - [24].

2.2 Related Work

Many researchers have evaluated how blockchain technology can help strengthen multi-factor authentication in multiple domains. A study called "Blockchain-Based Multi-Factor Authentication for Future 6G Cellular Networks" showed that applying blockchain-MFA to 6G networks offers great improvements in security by adding different layers of protection [25]. A bundle of techniques was implemented in cloud-enabled IoT, with embedded digital signatures, SAML, and single sign-on, to make sure vehicular clouds and other devices are secured [26]. In smart cities, a system called BAuth-ZKP was presented that uses zero-knowledge proofs, smart contracts, and one-time passwords to provide safe and private authentication [27]. Another team looked at privacy, security, and usability in MFA systems using blockchain and verified that it could help future digital authentication systems [28]. Literature reviews have revealed that blockchain can improve security and make MFA easier to implement in multiple environments, such as IoT and control systems [29]. Blockchain was studied in healthcare as a way to make sure sensitive data is safe and boost the ease of using authentication [30]. The application of blockchain-supported two-factor authentication on WordPress sites confirms that decentralized methods, such as MFA, can prevent cyberattacks [31], [32].

2.3 Critical Analysis

Although MFA using blockchain technology has progressed a lot, existing papers tend to look at MFA in specific fields instead of its application across many different systems. Some applications choose to encrypt information using smart contracts for extra privacy. However, many cannot work on a global scale or do not consider users' needs and problems with integration. Furthermore, real-world matters such as resource use and following rules on data protection have not been thoroughly studied. The next research stage needs to build frameworks that adapt to needs and ensure security, ease of use, and high performance, as shown in Table 1.

Table 1: Comparative analysis of blockchain-based MFA studies.

Features	[25]	[26]	[27]	[28]	[29]	Ours
MFA	✓	✓	✓	✓	✓	✓
Blockchain	✓	✓	✓	✓	✓	✓
Smart Contracts	✗	✗	✓	✓	✗	✓
Biometric	✗	✗	✓	✓	✓	✓
Privacy	✓	✗	✓	✓	✗	✓
Adaptive MFA	✗	✗	✗	✗	✗	✓

3 METHODOLOGY

This part details how a decentralized system for transferring funds is made, how it works, and how it should be evaluated using MFA and smart contract technology. The main parts of the methodology are architecture, functional opportunities, security flow, system boundaries, and the scope of analysis. Although the framework works as designed and tested on a local development environment (Ganache), significant concerns still exist about how the project scores in performance and how recovered accounts can be secured.

3.1 Overview of the System Architecture

The system is composed of three tightly integrated components:

1) On the Smart Contract Layer, a Solidity contract is deployed on Ganach. It is responsible for transferring funds by means of token-bound requests, provides capabilities to cancel, reject and validate claims.
2) The Frontend Interface uses HTML, CSS, and JavaScript and gets access to Web3.js, which handles connectivity with MetaMask. It enables users to perform actions on the contract as they would in real life through a web browser.
3) The server is written in Flask with SQLite, accepts user registrations, offers user login with hashed passwords, generates TOTP for Google Authenticator via pyotp, and handles Ethereum wallet connection.

All parts of the solution are set up as modules and use set APIs, which means they can be quickly deployed, maintained, and scaled in the future.

3.2 Smart Contract Design and Operation

3.2.1 MFA Token Binding

Users can connect one or more secret tokens to their wallet address by using addToken(). Before starting any transaction, these tokens should pass through the verifyToken() function, which makes the process safer than using the Ethereum address alone.

3.2.2 Transfer Structure and Lifecycle

Transfers are represented using a Transfer Request strict with the following attributes:

1) from: sender address;
2) amount: transfer amount in wee;
3) token: token associated with the request;
4) claimed: flag indicating if the transfer has been received;
5) canceled: flag indicating sender-side cancellation;
6) rejected: flag indicating receiver-side rejection.

3.2.3 Transfer Functions

The transfer functions manage the lifecycle of ETH transfers within the contract, from creation to completion or cancellation.

1) send Transfer (address recipient, string token): Adds a new pending transfer after verifying the sender's token and deducting ETH via msg.value;
2) claimTransfer (uint index, string token): The Receiver can claim funds if the correct token is provided and the transfer is still valid;
3) cancelTransfer (address recipient, uint index): Allows the sender to cancel the transfer and refund themselves before it's claimed;
4) rejectTransfer (uint index): Allows the receiver to reject a transfer, returning the ETH to the sender;
5) getTransferCount (address user) and getOutgoingCount (address sender): Provide front-end visibility into the number of transfers a user has received or sent.

3.2.4 Binary Tracking of Transfer Status

When data is mapped in two directions, accurate records are made:

1) incoming Transfers[address];
2) outgoingRefs[address].

As a result, those who send and receive transactions can consult the history, which shows the status and specifics of each transaction.

3.3 Web Interface and Interaction Logic

The frontend interface (HTML/JS/CSS) is developed to simulate a wallet environment using MetaMask. It communicates with the deployed smart contract using Web3.js and is divided into the following sections:

3.3.1 Token Management Page

The Token Management Page provides functionality for registering and verifying tokens associated with a user's wallet:

- Add Token enables users to register a new token within the system, linking it to their current wallet for future transactions.
- Verify Token allows users to check whether a specific token is already registered and valid for their current wallet, ensuring accurate token management and preventing duplication.

3.3.2 Funds Transfer Page

The Send Funds feature facilitates the transfer of ETH or tokens to a specified recipient. To initiate a transfer, the sender must provide the recipient's address, the desired amount, and the associated token:

- Unprocessed Transfers are displayed with actionable options, allowing the recipient to either claim the funds or reject the transfer.
- Sent Transfers provides the sender with a record of all transactions they have initiated, along with a Cancel button to revoke any pending transfer before it is claimed.

3.3.3 History Page

Transactions are categorized into:

1) Successfully Sent.
2) Successfully Received.
3) Canceled.
4) Rejected.

The display is dynamic and populated via getTransferCount() and getOutgoingAt() queries, ensuring historical accuracy.

3.4 Authentication Layer

The Flask backend gives you the following:

1) Passwords are stored securely during registration, and a TOTP key can be generated for setup using a QR code.
2) It is necessary to enter a username, password, and the current OTP for login.
3) Makes the Ethereum address connected to the verified user in the wallet.

3.5 Security Considerations and Limitations

While the system enforces token-bound transfers and OTP-based identity checks, the following issues remain unaddressed:

1) Private Key Awareness. The system assumes users manage private keys securely via MetaMask, but offers no built-in safeguards or educational prompts.
2) Data Storage Risks. Tokens and session data are temporarily stored in localStorage, which is vulnerable to XSS attacks and lacks encryption.
3) No Recovery Mechanism. There is no solution for users who lose access to their TOTP device or forget their bound token, which may lead to permanent account lockout.

Future iterations should implement secure client-side encryption, optional backup tokens, and multi-layer recovery protocols.

3.6 Deployment and Testing

The system was tested using Ganache, a local Ethereum-compatible development blockchain. While suitable for controlled testing, it does not emulate real-world conditions such as:

1) Real gas fees and congestion.
2) Public network latency.
3) Transaction competition and miner selection.

No quantitative performance testing (e.g., transaction time, resource utilization, or maximum system throughput) was performed. In addition, no security audits, penetration tests, or simulated attacks (e.g., DDoS, token forgery, SIM-swapping) were conducted. To address these gaps, future work should:

1) Deploy the system on public testnets (e.g., Goerli or Sepolia).
2) Use benchmarking tools to measure transaction time, memory/CPU usage, and scalability.
3) Conduct formal vulnerability assessments and adversarial testing scenarios.

4 RESULTS AND EVALUATION

The proposed Blockchain-Based Multi-Factor Authentication (MFA) system was evaluated from multiple perspectives: functionality, performance, security, and user experience. The assessment combined experimental deployment and manual testing within a controlled environment that simulates real-world usage scenarios.

4.1 Functional Testing

All core functionalities, including user registration, login, wallet-based token verification, and fund transfer authentication, were successfully tested on both the frontend (via MetaMask-integrated Web3) and the backend (Flask + Smart Contracts using Brownie). Each component was confirmed to interact properly with the deployed smart contract on a local Ethereum testnet.

4.2 Security Evaluation

Security was a primary focus in the system's architecture. The MFA model integrates three critical layers:

1) Something the user knows (password).
2) Something the user has (blockchain wallet + OTP/token).
3) Something the system verifies (blockchain-based token validity and transfer validation).

To assess robustness, several attack scenarios were simulated:

1) Replay attacks (mitigated via unique token per transaction).
2) Token forgery (prevented by storing tokens in bytes32 on-chain, and verifying against registered hashes).
3) Unauthorized fund access (blocked by enforcing wallet-based token ownership).

Additionally, using public/private key verification through smart contract methods further secures the system against impersonation.

4.3 Performance Metrics

The system was tested under typical operational loads. Token registration and verification operations exhibited average latencies of less than 500ms on the local testnet. Interactions with the blockchain (e.g., fund transfer, claim, cancel) followed average block confirmation times consistent with the Ethereum testnet environment (5–15 seconds), which can be improved by deploying on faster Layer-2 networks.

4.4 User Experience

A user-centered design approach was adopted. The UI was implemented with HTML, CSS, and Web3.js, ensuring intuitive navigation. Users could interact with blockchain features (connect wallet, register token, verify transfers) without deep technical knowledge, making the system suitable for technical and non-technical environments.

4.5 Comparison with Existing Solutions

The given Blockchain-based MFA framework was juxtaposed with classic authentication (e.g., Google Authenticator) and existing blockchain login systems. The comparison is based on decentralization, security, and usability:

1) Decentralization. Conventional solutions are centralized, whereas in some blockchain systems, There is partial decentralization. Through smart contracts, we can have full decentralization in our model;
2) Data Integrity. Unlike decentralized systems that are at risk of being hacked, the suggested system guarantees permanent information storage on the blockchain;
3) Tokens Verification. The classical OTPs are even susceptible, in our solution, we apply on-chain hashing to securely handle the tokens;
4) Authentication Layers. Most of the current solutions use single or two factor solutions. We adopt a three-layer approach of MFA and provide it as a secure solution;
5) Offline Support. The traditional and modern block chain tools do not support offline. The offered model allows to have a small level of offline access by offering interactivity with the wallets;
6) Security of Smart Contracts. Our contracts are thoroughly tester and provide better security than simple implementations of blockchain programs available today;
7) Easier log in. Although blockchain logins tend to be complicated, our system enhances user experience with simpler interface.

Such a comparison makes it clear that the proposed solution will perform better than the existing tools since it offers enhanced security, complete decentralization, and a positive user experience.

4.6 System Stress Testing and Attack Simulation

Additional evaluations were performed under controlled stress conditions and simulated attack scenarios to further validate the proposed blockchain-based MFA system's resilience, scalability, and security. This section provides a detailed breakdown of these experimental tests.

4.6.1 Performance Under Load

Stress testing was performed to evaluate the system's behavior under heavy load by simulating multiple concurrent users executing authentication and transfer operations:

1) Environment: Local Ethereum testnet (Ganache) and Flask-based backend.
2) Simulated Users: 50 clients interacting via JavaScript automation scripts using web3.js.
3) Operations Tested.

The Operations Tested component outlines the key functionalities evaluated during the system's testing phase:

1) 1 level: Token registration;
2) 2 level: Token verification;
3) 3 level: Fund transfer initiation;
4) 4 level: Claim and rejection of transfers.

Findings:

1) Average transaction latency: 600ms – 800ms for token operations.
2) Peak block confirmation delay: 17 seconds (Ethereum testnet limit).
3) Throughput. Approximately 30–35 transactions per minute without failures.
4) Failure rate. There was 0% functional failure during the test, although the UI became unresponsive temporarily due to the frontend blocking calls.

This confirms the system can handle concurrent operations without logic corruption or security breach.

4.6.2 Attack Simulation

To examine the robustness of the system's defense mechanisms, we simulated various common attacks on both the smart contract level and the frontend/backend communications.

4.6.3 Resource Utilization

The Resource Utilization analysis highlights the system's performance efficiency during operation:

- CPU usage (Local Server). Processing demands peaked at 85% when handling 50 simultaneous clients, indicating stable performance under moderate load.
- Memory usage. Memory consumption remained below 1 GB during smart contract interactions, demonstrating minimal RAM requirements.
- Blockchain storage impact. Each token or transfer record is stored as a `bytes32` or `address` mapping, resulting in a very lightweight footprint of less than 1 KB per entry, ensuring efficient on-chain data management.

Table 2: Attack scenarios and defense summary.

Attack	Result	Defense
Replay	Rejected	Token hashed; duplicates ignored
MitM	Failed	HTTPS + Web3 wallet signatures prevent replay
Token Injection	Failed	Smart contract accepts only pre-registered hashed tokens
Overwrite	Denied	Only sender (msg.sender) can modify transfer state
DoS	Partially blocked	Invalid entries rejected; frontend slows under high frequency calls

4.6.4 Observations and Insights

The security and reliability assessment evaluated the system's resilience against potential threats, its ability to maintain operational stability, and its behavior under real-world blockchain conditions. The findings highlight the platform's robustness, the effectiveness of its built-in safeguards, and considerations for performance in varying network environments:

1) The system showed excellent fault tolerance and resisted most Web3-targeted attack vectors.
2) The use of smart contract-level validation prevents manipulation at the frontend/backend levels.

3) However, performance under global blockchain congestion (mainnet) may vary depending on gas fees and confirmation times.

4 CONCLUSIONS

This research has demonstrated the feasibility and effectiveness of integrating blockchain technology with multi-factor authentication (MFA) to enhance the security of financial transactions in decentralized systems. By leveraging the immutable and decentralized nature of blockchain, combined with the layered security approach of MFA and the time-sensitive protection provided by TOTP codes, the proposed system successfully addresses several critical vulnerabilities found in conventional authentication mechanisms. The architecture ensures trustless, token-bound validation without relying on third-party intermediaries, reducing the risk of phishing, SIM-swapping, and unauthorized access. Implementing a smart contract-based transfer protocol with capabilities such as sender-side cancellation, recipient-side rejection, and dynamic historical logging represents a practical advancement in secure digital fund transfer systems. Furthermore, the inclusion of a user-friendly web interface and an off-chain authentication layer via Flask and Google Authenticator demonstrates the viability of bridging advanced cryptographic systems with accessible user experiences. In light of the increasing demand for secure, decentralized applications in domains such as IoT, smart cities, and digital identity management, the proposed framework offers a robust foundation for future development and real-world deployment. It enhances the usability and resilience of blockchain-based authentication and contributes meaningfully to the evolution of privacy-preserving, trustless systems in the digital era.

REFERENCES

[1] M. Jakobsson, “Two-factor inauthentication - the rise in SMS phishing attacks,” Computer Fraud & Security, vol. 2018, no. 6, pp. 6–8, 2018.

[2] V. Papaspirou et al., “Security revisited: Honeytokens meet Google Authenticator,” in Proc. 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conf. (SEEDA-CECNSM), IEEE, 2022, pp. 1–8.

[3] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, “A survey on blockchain for information systems management and security,” Information Processing & Management, vol. 58, no. 1, p. 102397, 2021.

[4] R. A. Grimes, Hacking Multifactor Authentication. Hoboken, NJ, USA: John Wiley & Sons, 2020.

[5] K. Sultan, U. Ruhi, and R. Lakhani, “Conceptualizing blockchains: Characteristics and applications,” arXiv preprint arXiv:1806.03693, 2018.

[6] A. Göransson and E. Asklund, “BankID-based authentication for phone calls,” 2020.

[7] A. K. Jain, R. Bolle, and S. Pankanti, Biometrics: Personal Identification in Networked Society, vol. 479. New York, NY, USA: Springer, 2006.

[8] B. K. Chaurasia and S. Verma, “Infrastructure based authentication in VANETs,” International Journal of Multimedia and Ubiquitous Engineering, vol. 6, no. 2, 2011.

[9] S. A. Sagar Acharya, “Two factor authentication using smartphone generated one time password,” IOSR Journal of Computer Engineering, vol. 11, no. 2, pp. 85–90, 2013, doi: 10.9790/0661-1128590.

[10] K. Fan, N. Ge, Y. Gong, H. Li, R. Su, and Y. Yang, “An ultra-lightweight RFID authentication scheme for mobile commerce,” Peer-to-Peer Networking and Applications, vol. 10, no. 2, pp. 368–376, Mar. 2017, doi: 10.1007/s12083-016-0443-6.

[11] Neha and K. Chatterjee, “Authentication techniques for e-commerce applications: A review,” in Proc. Int. Conf. on Computing, Communication and Automation (ICCCA), Greater Noida, India: IEEE, Apr. 2016, pp. 693–698, doi: 10.1109/CCAA.2016.7813811.

[12] M. Pilkington, “Blockchain technology: Principles and applications,” in Research Handbook on Digital Transformations, F. X. Olleros and M. Zhegu, Eds. Cheltenham, U.K.: Edward Elgar Publishing, 2016, doi: 10.4337/9781784717766.00019.

[13] P. Forrest, “Electronics and Computer Science Faculty of Physical and Applied Sciences, University of Southampton,” 2012.

[14] A. Hughes, A. Park, J. Kietzmann, and C. Archer-Brown, “Beyond Bitcoin: What blockchain and distributed ledger technologies mean for firms,” Business Horizons, vol. 62, no. 3, pp. 273–281, 2019.

[15] D. Rodeck and B. Curry, “What is blockchain,” Forbes, 2022.

[16] H. F. Atlam and G. B. Wills, “Technical aspects of blockchain and IoT,” in Advances in Computers, vol. 115. Amsterdam, The Netherlands: Elsevier, 2019, pp. 1–39, doi: 10.1016/bs.adcom.2018.10.006.

[17] J. J. Sikorski, J. Haughton, and M. Kraft, “Blockchain technology in the chemical industry: Machine-to-machine electricity market,” Applied Energy, vol. 195, pp. 234–246, 2017.

[18] H. F. Atlam, A. Alenezi, M. O. Alassafi, and G. B. Wills, “Blockchain with Internet of Things: Benefits, challenges, and future directions,” International Journal of Intelligent Systems and Applications, vol. 10, no. 6, pp. 40–48, Jun. 2018, doi: 10.5815/ijisa.2018.06.05.

[19] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292–2303, 2016.

[20] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," Journal of Medical Systems, vol. 40, pp. 1–8, 2016.

[21] M. S. Ahmad, W. Mohyuddin, H. C. Choi, and K. W. Kim, "4 × 4 MIMO antenna design with folded ground plane for 2.4 GHz WLAN applications," Microwave and Optical Technology Letters, vol. 60, no. 2, pp. 395–399, Feb. 2018, doi: 10.1002/mop.30969.

[22] G. Cheng, Y. Chen, S. Deng, H. Gao, and J. Yin, "A blockchain-based mutual authentication scheme for collaborative edge computing," IEEE Transactions on Computational Social Systems, vol. 9, no. 1, pp. 146–158, 2021.

[23] S. Bamashmos, N. Chilamkurti, and A. S. Shahraki, "Two-layered multi-factor authentication using decentralized blockchain in an IoT environment," Sensors, vol. 24, no. 11, p. 3575, 2024.

[24] M. S. Almadani, S. Alotaibi, H. Alsobhi, O. K. Hussain, and F. K. Hussain, "Blockchain-based multi-factor authentication: A systematic literature review," Internet of Things, vol. 23, p. 100844, Oct. 2023, doi: 10.1016/j.iot.2023.100844.

[25] J. Asim et al., "Blockchain-based multifactor authentication for future 6G cellular networks: A systematic review," Applied Sciences, vol. 12, no. 7, p. 3551, 2022.

[26] V. R. Kebande, F. M. Awaysheh, R. A. Ikuesan, S. A. Alawadi, and M. D. Alshehri, "A blockchain-based multi-factor authentication model for a cloud-enabled Internet of Vehicles," Sensors, vol. 21, no. 18, p. 6018, Sep. 2021, doi: 10.3390/s21186018.

[27] Md. O. Ahmad et al., "BAuth-ZKP - A blockchain-based multi-factor authentication mechanism for securing smart cities," Sensors, vol. 23, no. 5, p. 2757, Mar. 2023, doi: 10.3390/s23052757.

[28] I. Wanisha, J. B. James, J. S. Witeno, L. H. Mohammad Bakery, M. Samuel, and M. Faisal, "Multi-factor authentication using blockchain: Enhancing privacy, security and usability," International Journal of Computer Technology and Science, vol. 1, no. 3, pp. 41–55, Jul. 2024, doi: 10.62951/ijcts.v1i3.24.

[29] Z. A.-A. M. Fneish, M. El-Hajj, and K. Samrouth, "Survey on IoT multi-factor authentication protocols: A systematic literature review," in Proc. 11th Int. Symp. on Digital Forensics and Security (ISDFS), Chattanooga, TN, USA: IEEE, May 2023, pp. 1–7, doi: 10.1109/ISDFS58141.2023.10131870.

[30] T. Suleski, M. Ahmed, W. Yang, and E. Wang, "A review of multi-factor authentication in the Internet of Healthcare Things," Digital Health, vol. 9, Jan. 2023, doi: 10.1177/20552076231177144.

[31] J. A. A. Cardoso, F. T. Ishizu, J. T. de Lima, and J. de Souza Pinto, "Blockchain based MFA solution: The use of hydro raindrop MFA for information security on WordPress websites," Brazilian Journal of Operations & Production Management, vol. 16, no. 2, pp. 281–293, 2019.

[32] A. Eldow et al., "Information communication technology infrastructure in Sudanese governmental universities," in Recent Advances in Intelligent Systems and Smart Applications, 2021, pp. 363–375.