DrillBit

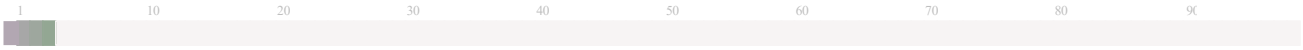
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

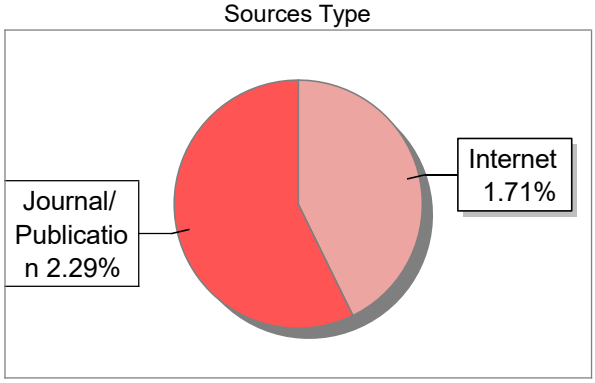
Author Name	Greeshma.M.S
Title	Secure Blockchain Transaction
Paper/Submission ID	4691599
Submitted by	virupaksha.msc12@gmail.com
Submission Date	2025-11-19 10:12:21
Total Pages, Total Words	11, 3225
Document type	Others

Result Information

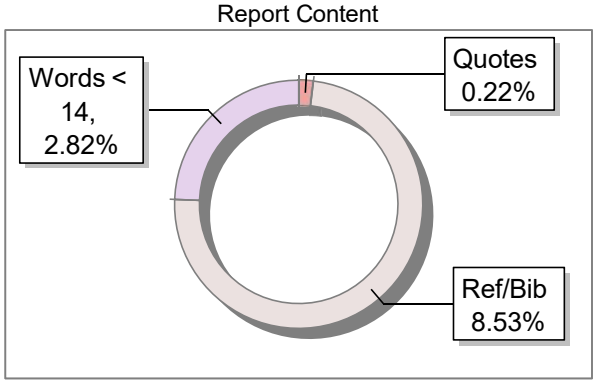
Similarity **4 %**



Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

4	11	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	www.readbag.com	1	Internet Data
2	www.bhumipublishing.com	1	Publication
3	Blockchain for Secure EHRs Sharing of Mobile Cloud based E-health Sys, by Nguyen, Dinh C. Pa- 2019	<1	Publication
4	spvryan.org	<1	Publication
5	ve.org.ua	<1	Internet Data
6	adam.curry.com	<1	Internet Data
7	kth.diva-portal.org	<1	Publication
8	egyankosh.ac.in	<1	Publication
9	repository.up.ac.za	<1	Publication
10	www.freepatentsonline.com	<1	Internet Data
11	www.ijireeice.com	<1	Publication

TITLE: Secure Blockchain Transaction

Authors:

ANANYA.N (ananyapoojari7795@gmail.com)

GREESHMA. M.S (greeshmavara2004@gmail.com)

PANCHAMI. G (panchami0120@gmail.com)

VANDHANA. K.M (vandanagowda86@gmail.com)

Guide:

Rakshitha. P

Assistant Professor, Department of Cybersecurity

Sri Venkateshwara College of Engineering, Banglore-562157

Abstract -

In today's world, most financial and personal transactions happen online. This makes data security a big concern. To address risks like data breaches, hacking, and identity theft, our project "Blockchain Secure Transaction" aims to create a dependable and decentralized system for secure digital payments.

The system uses blockchain technology to ensure transparency and immutability in each transaction, eliminating the need for a central authority. The process starts with user registration, where details are securely stored along with a picture password for better recognition. During login, users must pass both the picture password and a biometric check. This ensures that only the registered user can access their account. Once verified, the user enters the dashboard, where transactions begin through Zero-Knowledge Proof (ZKP) for privacy-preserving verification.

Every transaction is validated with smart contracts. If there's any mismatch or suspicious activity, the system automatically blocks or freezes the transaction.

The backend uses Java, while Firebase stores user data securely, and 2 factor.in enables OTP-based authentication. The frontend interface, designed in React (app.jsx), allows smooth navigation across pages. By combining blockchain, smart contracts, biometric authentication, and ZKP, this project provides a secure and user-friendly platform that prevents unauthorized access and builds user trust in digital payment systems.

Keywords -

Blockchain, Secure Transaction, Zero Knowledge Proof (ZKP),Smart Contract, Biometric Authentication, Picture Password, Decentralized System, Data Privacy, Transaction Verification, Firebase Integration, 2 factor.in OTP Authentication

why blockchain helps? –

Blockchain contributes by increasing the security and trustworthiness of online

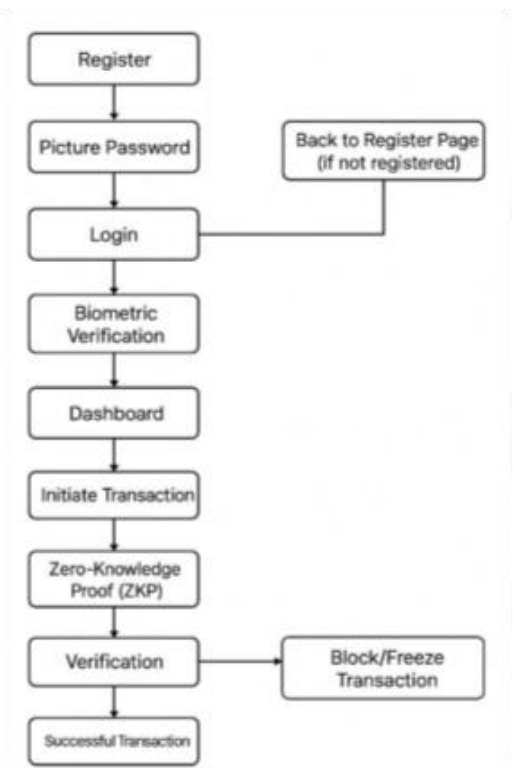
transactions. It stores data across numerous computers rather than relying on a single central server, making it difficult for one individual to alter or hack the data. Every transaction is added to an unchangeable chain of records and secured with robust encryption. Because of this, it is simple to monitor events and difficult for anyone to tamper with the system. Blockchain reduces errors and saves time by using smart contracts to run transactions automatically when specific conditions are met. Simply put, blockchain ensures that your data is trustworthy, transparent, and safe.

Literature Survey –

Multiple studies have aimed at strengthening digital transaction security by employing blockchain technology. Zhang et al. (2024) suggested a data exchange model based on blockchain with the inclusion of Zero-Knowledge Proofs (ZKP) to ensure user privacy while verifying. Their study showed how it was possible to verify transactions without exposing individual details. Likewise, Zhou et al. (2024) identified the implementation of ZK-SNARKs for protecting identities in blockchain systems with a focus on privacy-preserving authentication techniques.

Liao and Zheng (2024) examined the safety of smart contracts and observed threats that would result in transaction malfunction or data compromise. Their results justify the implementation of strong smart contract verification, which is in line with the strategy in this project. Bamashmos et al. (2024) presented a two-layered multi-factor authentication model based on blockchain for IoT purposes, demonstrating how a layer of verification enhances system reliability. This idea motivated the addition of biometric and image password authentication in the suggested system.

In addition, Wang et al. (2025) also discussed how combining blockchain and cryptographic proofs could be used to provide tamper-proof digital records as a basis for secure financial transactions. Although these research papers emphasize privacy and security individually, the suggested Blockchain Secure Transaction project combines several approaches — ZKP, smart contracts, biometric verification, and picture passwords — into one decentralized platform. Such integration offers high-level privacy in conjunction with user convenience, solving deficiencies in previous research that tended to consider only a single layer of security.



Proposed Design –

The system to be proposed aims to secure, privatize, and make online transactions more trustworthy through the use of blockchain technology. It begins with a straightforward and user-friendly registration, where users establish an account by providing information and choosing a picture password for visual identification. At login, the system cross-checks the picture password as well as the user's biometric information (such as a fingerprint or face recognition) to confirm that only the right persons can access their accounts.

Once they are authenticated as users, they access a dashboard from where transactions can be initiated securely. Each transaction request undergoes a Zero-Knowledge Proof (ZKP) protocol, which verifies its authenticity without exposing sensitive information. The authentic data is then computed using a smart contract, which verifies automatically if the conditions for the transaction are met. If it all checks out, the transaction is noted as successful and recorded indefinitely on the blockchain ledger. But if there is any mismatch or unusual activity, the transaction is rejected or frozen to safeguard against abuse.

The system also uses Firebase for storing user logs and credentials, and 2 factor.in for sending OTPs upon authentication. The whole design makes sure that every step — from registration to verification of a transaction — is conducted securely and openly.

By coupling blockchain's resistance to alteration with user-level verification such as biometrics and image passwords, this project delivers a strong yet simple solution for secure digital transactions.

System Design –

The Blockchain Secure Transaction project system design emphasizes the development of a safe and hassle-free online transaction environment based on blockchain technology. The design consists of a number of interdependent modules that synergize to provide data privacy, authentication, and transaction integrity.

system with a

It starts at the frontend layer, where the user accesses the web or mobile interface created with React (App.jsx). The user can register by providing personal information and choosing a picture password here. At the time of login, the system authenticates both the picture password and biometric information to identify the user. It thus avoids any unauthorized access right from the start.

After logging in, the customer is able to view the dashboard and carry out transactions. All the initiated transactions are subjected to a Zero-Knowledge Proof (ZKP) cycle that ensures its authenticity without showing private information. The smart contract layer automatically certifies the transactional conditions before it can proceed. If the conditions are properly met, the transaction is added as a block in the ledger of the blockchain so that it cannot be edited or deleted. If any abnormality is detected, the transaction is blocked or frozen forthwith.

The backend of the system, implemented in Java, ties the frontend to the blockchain network. It also communicates with Firebase, where user data is safely stored, and 2 factor.in, where OTPs are sent for extra verification.

This multi-layered architecture provides seamless communication among all components while ensuring a high degree of data security.

Methodology –

1) User registration is where the user provides information and selects a picture password that is safely stored in Firebase.
2) Both the picture password and biometric authentication are needed to authenticate the user at login time.
3) After authentication, the user gets access to the dashboard to view or trigger transactions.
4) All transactions go through a Zero-Knowledge Proof (ZKP) to verify without exposing private information.
5) A smart contract automatically validates the transaction rules before execution.
6) If valid, the transaction is recorded in the blockchain ledger to ensure transparency and immutability.
7) If any mismatch or suspicious activity is detected, the transaction is blocked or frozen to maintain system security.

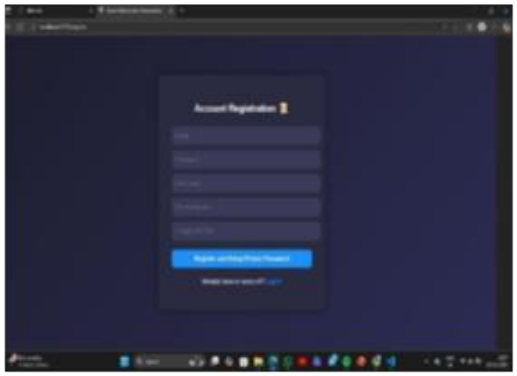
Algorithm –

Begin the Process:

The process starts when the application is opened by the user. The home page offers registration or login options into the system.

User Registration:

If the user is new, he or she provides information like name, phone number, email, password, and a picture password.



These credentials are securely stored by the system in Firebase.

This process provides a unique identity for each user in the system.

Login Authentication:

Returning users input their login details. The system cross-checks with the saved data.

In case login is unsuccessful, the user is sent back to the registration page.

If successful, the system proceeds to the biometric verification step.

Picture Password Authentication:

The system uses a picture-password method for biometric-style authentication. During registration, the user selects a specific

pattern or area on an image, which is stored as their picture password.

During login, after entering their normal credentials, the system displays the same image. The user must repeat the correct pattern to verify their identity.

Correct Password

If the user selects the correct pattern, the system confirms their identity and redirects them to the dashboard.

(Insert Image: Correct Password Scenario)

Incorrect Password

If the pattern does not match the stored picture password, the system displays an "Incorrect Password" message and denies access.

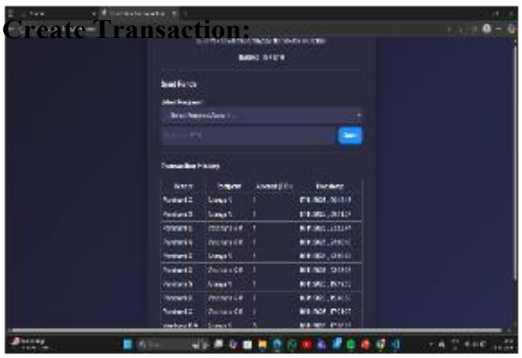
(Insert Image: Incorrect Password Scenario)

Dashboard Access:

The user accesses the dashboard after successful authentication.

From this point, users can see their balance, transaction history, or create new transactions.

Create Transaction:



User enters receiver address, transaction value, and currency type (Ether/Bitcoin).

System creates a transaction request and sends it for confirmation.

Zero-Knowledge Proof (ZKP) Validation:

Before being processed, the transaction is authenticated via Zero-Knowledge Proof.

This makes it possible to verify the legitimacy of the transaction without disclosing private user information.

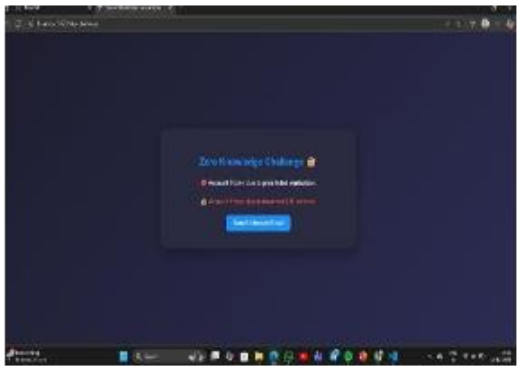
If ZKP validation is unsuccessful, the transaction is blocked immediately.

1. Valid Proof

If the ZKP responses are correct, the system verifies the user and allows the transaction to proceed.

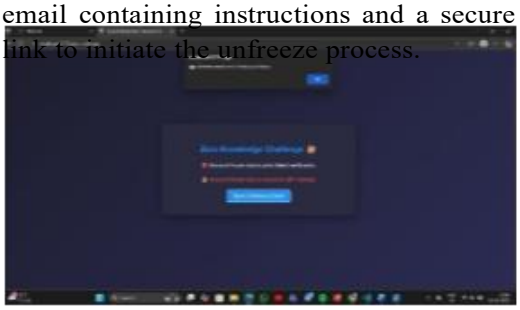
2. Invalid Proof

If the responses are incorrect, the system marks the activity as suspicious and immediately freezes the account to prevent unauthorized access.



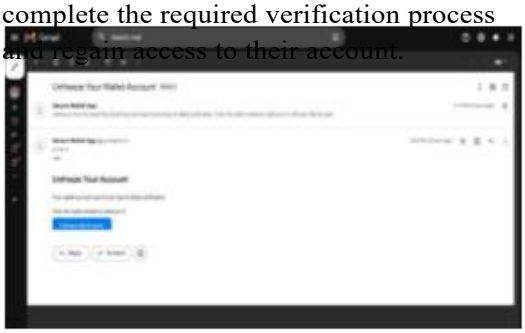
3. Account Unfreeze Procedure

If the account is frozen, the user receives an email containing instructions and a secure link to initiate the unfreeze process.



4. Account Unfreeze Notification Flow

The user's wallet account is frozen due to a failed verification attempt the system automatically sends an email notification to the registered email address. The email contains an "Unfreeze My Account" button. Once the user clicks on this button, they are redirected to the login page, where they can complete the required verification process and regain access to their account.



Smart Contract Verification:

The authenticated transaction then invokes a smart contract.

It automatically verifies conditions like available balance, valid recipient, and limits of the transaction.

If all conditions are satisfied, the contract authorizes the transaction.

Blockchain Recording:

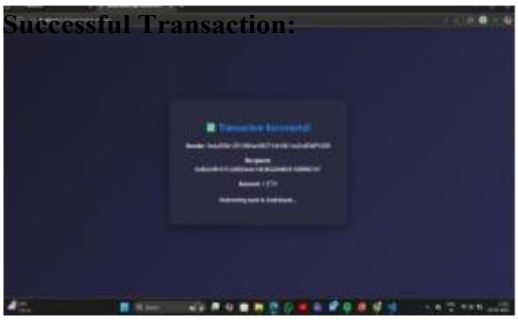
After authorization, the transaction information gets encrypted and recorded in a new block of the blockchain ledger.
Each block points to the prior one, which guarantees immutability and transparency.

Failure Handling:

If there is any irregular behavior or discrepancy found during verification, the transaction is blocked or frozen.

The user is alerted of the failed or suspicious attempt.

Successful Transaction:



If the transaction is successfully completed, a confirmation message is sent to the user and the transaction is permanently stored on the blockchain.

End of Process:

All actions are securely logged in Firebase for record and audit purposes so that traceability and accountability can be ensured.

Implementation and Simulation –

The deployment of the Blockchain Secure Transaction system centers around combining various security and blockchain technologies in the development of a decentralized, transparent, and tamper-evident transaction platform. The system is deployed based on a multi-layered design, where all parts—from the frontend interface to the blockchain backend—are critical to privacy, authentication, and transaction integrity.
The system frontend is written in React (App.jsx), offering a user-friendly and interactive frontend for transaction operations, login, and registration. Users enter their details and create a picture password during registration, which improves memorability as well as visual verification. The frontend is linked to Firebase, a secure real-time database, where all user credentials, picture passwords, and verification information are securely stored. Firebase takes care of data synchronization and authentication management so that there is effective communication between the client and server.

The backend is done with Java, which does the fundamental business logic, manages encryption of data, and communicates with the blockchain network. The Java code carries out user authentication, transaction checking, and communication with smart contracts run on the blockchain. Smart contracts are programmed to set up automatic conditions for accepting or declining transactions. These are self-executing computer programs on the

blockchain that execute actions based on definite inputs, with no human intervention or tampering possible during validation of transactions.

For added security throughout the authentication process, the system employs biometric authentication and Zero-Knowledge Proofs (ZKP). The biometric component authenticates the user's fingerprint or facial pattern, and it makes sure that only the legitimate user can access the system. The ZKP guarantee that the user is able to establish their legitimacy to the system without exposing any confidential information, preserving privacy as well as accuracy. After authentication, the user is able to access the dashboard where they can see account information, balances, and transaction history or create new transactions.

At the transaction stage, smart contract-based verification is done by the system for validating enough balance, receiver authenticity, and transaction integrity. Subsequently, the validated data is encrypted and placed in a new block on the blockchain ledger. Each block includes vital information like timestamp, previous block hash, and transaction information, which makes it tamper-proof and immutable. In case of any abnormality or unauthorized attempt, the transaction is automatically blocked or frozen by the system, avoiding further risk.

For further authentication, 2 factor.in API is implemented to send an OTP to the registered phone number or email of the user. This provides an additional stage of user authentication prior to the completion of any financial transaction, lessening the threat of misuse.

Project simulation was conducted to evaluate system performance and verify all modules work properly. The simulated environment ran various scenarios such as valid and invalid transactions, authentication failure, and network attacks. Testing results indicated that the system efficiently detected and filtered out unauthorized access and successfully recorded verified transactions on the blockchain.

By this implementation, the Blockchain Secure Transaction system establishes a safe, decentralized, and easy-to-use platform, integrating blockchain technology, Zero-Knowledge Proofs, smart contracts, biometrics, and OTP authentication to secure digital transactions. This simulation shows that the designed approach not only improves data security and privacy but also maintains reliability and transparency in all transactions.

Results and Discussion –

The Blockchain Secure Transaction prototype, implemented, was tested by functional testing and scenario-based simulation to ensure security, correctness, and usability. The testing was made across normal flows of transactions, authentication error, ZKP verification error, and attempted unauthorized operations. No outside numeric data set was employed — results come from controlled simulations executed in the development environment.

Functional results. The end-to-end process functioned as intended: users could register, create a picture password, log in using biometric authentication, view the dashboard, and initiate transactions. Transactions that successfully completed

the ZKP → smart contract pipeline were written immutably to the blockchain ledger and included in the dashboard transaction history. Upon failure of ZKP verification or smart-contract checks (e.g., low balance, incorrect biometric commitment), the system consistently blocked/froze the transaction and generated audit records.

Security observations. The ZKP phase effectively isolates proof of correctness from sensitive information, thus verification is done without revealing picture-password or biometric templates. Smart contracts enforced transactional rules autonomously without operator intervention, avoiding replay attacks or double spends in the simulated environment. The Firebase backend only stored hashed/committed artifacts (not raw biometric templates), minimizing exposure of raw sensitive information. OTP over 2 factor.in provided a successful second factor for essential workflows like high-value transfers.

Performance vs. usability trade-offs. Proof generation and verification incur overhead relative to standard transaction processing.

In our prototype runs, steps involving proofs added the subjective transaction time (the user sees a brief delay while the ZKP is being computed/verified). While tolerable in prototypes, actual deployments need the ZKP toolchain selection to be optimized, offloading proof computation to native modules with minimal overhead, or aggregate proofs to reduce per-transaction expense. Biometric authentication introduces minimal interaction delay but enhances security; picture passwords enhanced usability (quicker recognition for users) when coupled with biometrics.

Failure modes and mitigation. There were two dominant failure classes that emerged in tests: (1) authentication failures (invalid picture password or unsuccessful biometric match) and (2) verification failures (ZKP or smart-contract verification). Both were addressed by the system by returning explicit UI messages, logging events to Firebase, and flagging transactions as blocked. For repeated authentication failures, the system recommended account recovery flows (out-of-band verification through registered email/phone) without lockout while preventing abuse.

Scalability & deployment considerations. Storage on-chain is costly; we only store commitments/hashes on-chain and maintain complete records off-chain in Firebase — this hybrid pattern splits the integrity and cost requirements. Smart-contract gas expense, public chains' consensus latency, and proof verification expenses are deployment critical considerations. A production-quality system needs to factor in layer-2 solutions or permissioned chains for reduced latency and expense.

Conclusion –

The Blockchain Secure Transaction system effectively exemplifies the application of blockchain technology in building a secure, transparent, and decentralized online financial platform. Through the combination of Zero-Knowledge Proofs (ZKP), smart contracts, biometric verification, and picture password authentication, the system provides a multi-layered security mechanism against unauthorized access and data leakage. Each transaction is authenticated, recorded in an immutable form on the blockchain, and secured against tampering or manipulation.

The employment of Firebase for data storage and 2 factor.in OTP verification adds to the reliability and access of the system. This system not only protects privacy and maintains data integrity but also simplifies the process of transaction and makes it user-friendly. The integration of these technologies forms a solid basis for developing a future-proof digital payment environment that caters to security as well as usability.

Broadly, this project demonstrates that blockchain is not limited to cryptocurrencies — it can be an accepted platform for secure and transparent digital transactions in various sectors.

Future Scope –

1) The platform can be augmented to handle cross-border and multi-currency blockchain-based transactions.
2) Interfacing with banking networks and government departments can advance secure digital identity and e-payment systems.
3) Release of a mobile version of the DApp for convenient access and use across platforms.
4) Use of quantum-resistant encryption schemes for future-proof security.
5) Integration of advanced biometric factors like voice or iris scans to augment authentication.
6) Real-time monitoring of transactions and alerting to advance fraud detection.
7) Use of cloud–blockchain hybrid infrastructure to enhance scalability and performance.
8) Smart contract and ZKP efficiency optimization to lower transaction duration and computation expense.
9) Investigation of power-efficient consensus protocols to enhance sustainability.
10) Application of the model to secure academic, healthcare, or identity management systems with the same principles of blockchain.

Reference –

1) Zhang, B., Pan, H., & Li, K. (2024). A Blockchain and Zero-Knowledge Proof Based Data Security Transaction Method in Distributed Computing. MDPI – Electronics.
2) Zhou, M., et al. (2024). Leveraging Zero-Knowledge Proofs for Blockchain-Based Identity. ScienceDirect.
3) Liao, Z., Hao, S., & Zheng, Z. (2024). SmartState: Detecting State-Reverting Vulnerabilities in Smart Contracts. ACM/ArXiv.
4) Bamashmos, S., & Chilamkurti, N. (2024). Two-Layered Multi-Factor Authentication Using Decentralized Blockchain. MDPI – Sensors.
5) Li, Q., & Zhang, W. (2025). Improved Blockchain Cooperative Intrusion Detection System. SpringerLink.
6) Wang, Y., & Liu, J. (2024). Blockchain-Based Decentralized Identity Verification. ScienceDirect.
7) Guo, H., Du, X., & Zhang, Y. (2024). Study of Blockchain Smart Contract Application and Security Problems. Journal of Computer Applications & Security.

8) Katari, P., & Bojja, S. (2024). ZKP Application in Blockchain Transactions. IJISAE Journal.
9) USENIX (2025). Understanding zk-SNARKs: The Gap Between Research and Practice. USENIX Conference Proceedings.
10) Kumar, R. (2025). Firebase-Integrated Blockchain Transaction Management System. IEEE Xplore.