iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IX **Month of publication:** September 2026

DOI: https://doi.org/10.22214/ijraset.2026.84949

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IX Sep 2026- Available at www.ijraset.com

A Secure Blockchain-Based Offline Payment System with Fraud Window Analysis

Kuda Sravani Sandhya Sree
Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh

Abstract: ***The rapid growth of digital payment systems has increased the demand for secure and reliable payment mechanisms capable of operating in environments with intermittent or unavailable Internet connectivity. Conventional blockchain-based payment systems typically require continuous network access for transaction validation and settlement, which limits their applicability in rural areas, disaster-affected regions, and other low-connectivity environments. This research presents a Secure Blockchain-Based Offline Payment System with Explicit Fraud Window Analysis, which allows transactions to be created and stored locally during periods of network unavailability and synchronized with the Ethereum blockchain once connectivity is restored. The proposed system integrates an offline wallet, digitally signed transaction processing, local transaction verification, blockchain-based settlement, and a fraud-detection mechanism. Its central contribution is the explicit measurement of the fraud window, the time interval between offline transaction creation and blockchain synchronization, which is analysed alongside pending-transaction activity and trust-related parameters to estimate transaction risk. The system was implemented using Solidity, Ethereum, Hardhat, Ethers.js, JavaScript, Node.js, and MetaMask. The resulting prototype demonstrates the feasibility of secure offline transaction processing that preserves blockchain-based final settlement while providing measurable visibility into the security risk introduced by delayed synchronization.***

Keywords: Blockchain, Ethereum, Offline Payment, Fraud Detection, Fraud Window Analysis, Smart Contract

I. INTRODUCTION

A Secure Blockchain-Based Offline Payment System with Fraud Window Analysis is designed to enable digital payments when Internet connectivity is temporarily unavailable. In the proposed system, a transaction is created, digitally signed, and stored locally while the user is offline. Once connectivity is restored, the pending transaction is synchronized with the Ethereum blockchain and validated through a smart contract.The system introduces Fraud Window Analysis, which measures the time between transaction creation and blockchain synchronization:

$$FW = T_{sync} - T_{gen}$$

This window is analysed along with indicators such as duplicate transactions, nonce reuse, rapid transaction activity, and unusual payment behaviour to identify potentially suspicious transactions. Thus, the system combines offline payment capability, cryptographic security, blockchain-based settlement, and temporal fraud analysis to provide a secure and reliable payment mechanism for environments with intermittent network connectivity.

A. Background

Digital payment systems have become a critical part of modern financial infrastructure, but conventional digital payments generally require communication with a payment gateway, bank, or distributed ledger for authorization and settlement, so unreliable Internet connectivity can prevent or delay payments. The Reserve Bank of India (RBI) has recognized this need and introduced a framework for small-value digital payments in offline mode using cards, wallets, and mobile devices [5], [6]. Blockchain offers an alternative foundation for transaction integrity because transactions can eventually be recorded in an immutable ledger, and Ethereum smart contracts can enforce predefined rules through on-chain code [3], [4]. However, the blockchain cannot normally be consulted while a user is completely offline, so a practical offline blockchain payment system must separate the transaction process into offline authorization/local storage and onlinesynchronization/blockchain settlement. The present work focuses on this two-stage architecture and introduces an additional concept called Fraud Window Analysis.

B. Problem Statement

In a conventional blockchain payment the sequence User → Network → Blockchain → Confirmation requires continuous connectivity.

The proposed system instead follows User → Offline Authorization → Local Storage → Connectivity Restored → Blockchain Settlement, creating a period during which a transaction has been generated but not yet recorded on-chain — a security-relevant interval in which conflicting or suspicious transactions may occur before reconciliation. This paper therefore addresses: how can a payment be securely authorized during temporary network unavailability and later reconciled with a blockchain while explicitly analysing the window in which fraud may occur?

II. RELATED WORK

Research on offline payments spans traditional offline electronic cash, blockchain-based offline payments, and hardware-assisted approaches. Early offline e-cash systems focused on double-spending prevention through anonymity, unforgeability and conditional traceability, but relied on complex cryptographic protocols and dedicated e-cash structures [9]–[11]. More recent work uses blockchain and smart contracts for eventual settlement: Jie et al. [1] proposed a Secure and Flexible Blockchain-Based Offline Payment Protocol using trusted execution environments (TEEs) with loosely synchronized clocks, and other mobile-wallet approaches combine Ethereum smart contracts, Bluetooth, and digital signatures through offline tokens [8]. Reviews of this space report offline tokens, trusted third parties, decentralized identity, TEEs and cryptographic validation as the common mechanisms for preventing double spending and token replication [13]–[15]. TEE-based systems can further protect wallet state and private keys in hardware-isolated environments [13], [14]; the present prototype does not implement a production TEE and identifies hardware-backed protection as future work.

These works demonstrate that offline blockchain payment is feasible, but the proposed project's distinguishing contribution is not a new cryptographic primitive: it is the explicit temporal analysis of the interval between offline transaction generation and blockchain synchronization, combined into a single pipeline with offline authorization, structured signatures, local storage, blockchain reconciliation and smart-contract validation, summarized against related approaches in Table I.

Approach	Offline Op.	Blockchain	Double-Spend Handling	Hardware Dep.	Main Focus
Traditional offline e-cash	Yes	No	Deposit-time detection	No	Cryptographic e-cash
Offline blockchain tokens	Yes	Yes	Token validation/reconciliation	May vary	Offline token transfer
TEE-based blockchain payment	Yes	Yes	Hardware-protected state	Yes	Secure execution
Jie et al. protocol [1]	Yes	Yes	Protocol + TEE mechanisms	Yes	Security & flexibility
Proposed prototype	Yes (auth.)	Yes	Smart contract + rule-based analysis	No	Fraud Window Analysis

TABLE I: COMPARISON OF OFFLINE PAYMENT APPROACHES

III. METHODOLOGY

A. System Architecture

The system has five functional layers: an offline wallet layer (MetaMask/browser), a transaction engine that creates, signs and validates transactions, a local storage layer (browser localStorage), a Node.js/Express synchronization and fraud-analysis backend, and an Ethereum smart-contract settlement layer, as shown in Fig. 1.

B. Offline Transaction Model and Processing

An offline transaction is represented as TX = {sender, receiver, amount, nonce, timestamp, expiry, txID, signature} and is signed before local storage, using *EIP-712* typed structured-data signing, which standardizes hashing and signing of typed data and supports domain separation via chain ID and verifying-contract address [2]. The wallet signs the transaction, records the generation timestamp, stores it locally with status PENDING, and — once connectivity is restored — forwards it to the backend, which performs fraud analysis before submitting valid transactions to the smart contract and updating the local status.

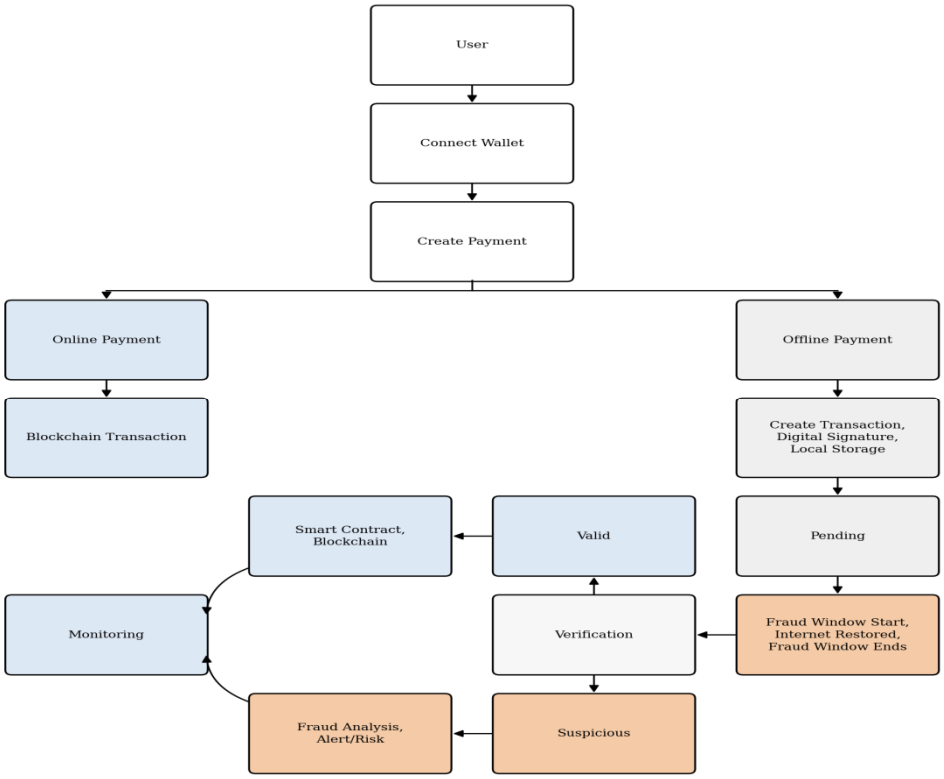


Fig. 1 Proposed system architecture

C. Fraud Window and Risk Analysis

The central concept of this work is the Fraud Window, defined as FW = Tsync − Tgen, where Tgen is the time the offline transaction is generated and Tsync is the time it is synchronized with the blockchain. A large FW does not by itself indicate fraud; it provides temporal context that is combined with other indicators (Table II) within a 60-second analytical window used by the prototype. The resulting risk score is modelled as Risk = f(FW, D, N, B, A, R, S), where D, N, B, A, R and S denote duplicate-transaction, nonce-inconsistency, burst, abnormal-amount, receiver-diversity and spending-concentration indicators respectively; the implementation uses rule-based scoring rather than a trained model, producing an APPROVE, FLAG, or BLOCK decision.

TABLE II: FRAUD INDICATORS USED BY THE PROTOTYPE

Indicator	Description	Concern
Transaction delay	Time between generation and sync	Extended exposure
Duplicate transaction ID	Same transaction resubmitted	Replay
Duplicate nonce	Sender nonce reused	Replay/conflict
Rapid transaction burst	Multiple tx in short interval	Suspicious activity
Abnormal amount	Deviates from normal activity	Anomalous payment
Receiver diversity	Unusual number of receivers	Account abuse
Spending concentration	High spend within window	Possible doublespend

D. Blockchain and Smart Contract Layer

The Ethereum blockchain provides the settlement layer, with the smart contract enforcing sender verification, transaction uniqueness, nonce validation, expiry checking, balance/escrow validation, processed-transaction tracking, and settlement events [3], [4]. Fig. 2 shows the settlement workflow from offline transaction creation through fraud-window calculation and signature verification to on-chain recording.

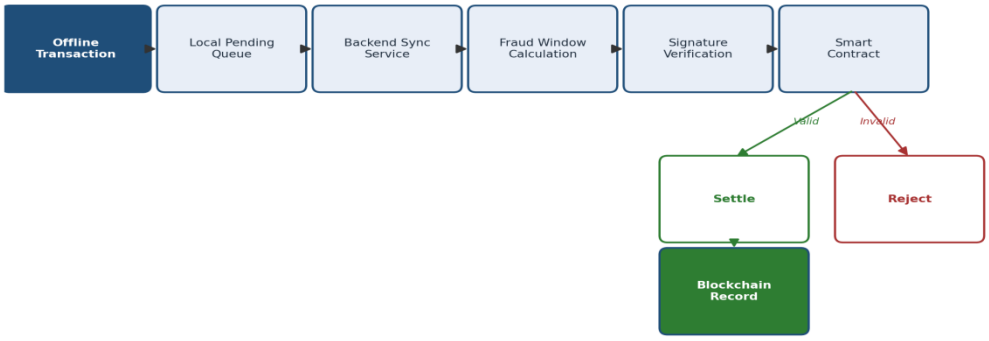


Fig. 2 Offline-to-blockchain settlement workflow

E. Implementation

A local prototype was implemented with Hardhat providing an Ethereum-compatible development chain, MetaMask for wallet interaction, Solidity for the smart contract, Ethers.js for blockchain interaction, and a Node.js/Express backend with a JSON ledger and HTML/JavaScript dashboard for monitoring pending and settled transactions.

IV. RESULTS AND DISCUSSION

A. Functional Results

The prototype was exercised across the full pipeline Offline Creation → Local Storage → Fraud Analysis → Synchronization → Smart Contract → Settlement. Table III summarizes functional test outcomes; these are functional prototype results and not large-scale performance benchmarks.

TABLE III: FUNCTIONAL TEST SCENARIOS

Test Case	Expected Behaviour	Prototype Behaviour
Wallet connection	Wallet should connect	Supported
Offline storage	Pending tx stored locally	Supported
Digital signing	Tx cryptographically signed	Supported
Synchronization	Tx forwarded for settlement	Supported
Duplicate transaction	Duplicate detected	Supported (tx tracking)
Duplicate nonce	Reuse detected	Supported (nonce check)
Invalid signature / Expired tx	Transaction rejected	Supported
Fraud analysis	Suspicious tx flagged	Supported
Blockchain settlement	Valid tx recorded	Supported

B. Fraud Window Behaviour

Table IV lists five illustrative observations used only to demonstrate the FW = Tsync − Tgen calculation and are not measured experimental results. Since the prototype uses a 60-second analytical window, TX05 (75 s, Fig. 3) would receive additional scrutiny, but the elevated delay alone does not prove fraud — it is combined with the indicators of Table II, and representative combined outcomes are shown in Table V.

TABLE IV: ILLUSTRATIVE FRAUD WINDOW OBSERVATIONS

Transaction	Generation	Sync	Fraud Window
TX01	10:00:00	10:00:12	12 s
TX02	10:00:20	10:00:41	21 s
TX03	10:01:00	10:01:28	28 s
TX04	10:01:30	10:02:20	50 s
TX05	10:02:00	10:03:15	75 s

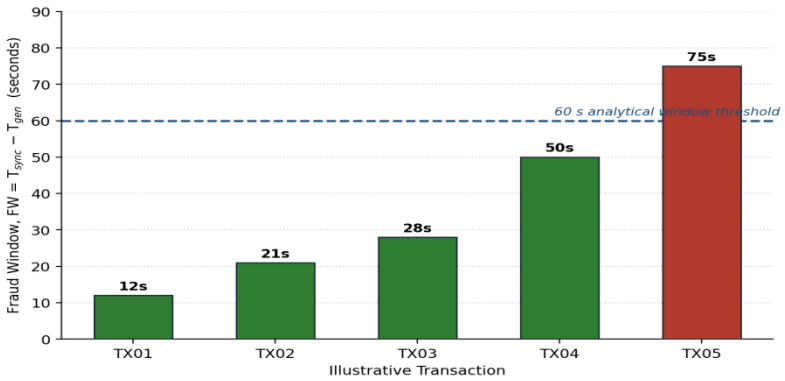


Fig. 3 Illustrative fraud window trend

TABLE V: EXAMPLE FRAUD ANALYSIS SCENARIOS

Scenario	Fraud Window	Indicator	Decision
Normal transaction	Short	No anomaly	APPROVE
Long sync delay	>60 s	No other anomaly	Additional scrutiny
Duplicate transaction ID	Any	Duplicate detected	BLOCK
Duplicate nonce	Any	Nonce reused	BLOCK/FLAG
Rapid transaction burst	Short	Multiple rapid payments	FLAG
Abnormal amount	Moderate	Amount anomaly	FLAG
Invalid signature / Expired	Any	Auth failure / expiry	BLOCK

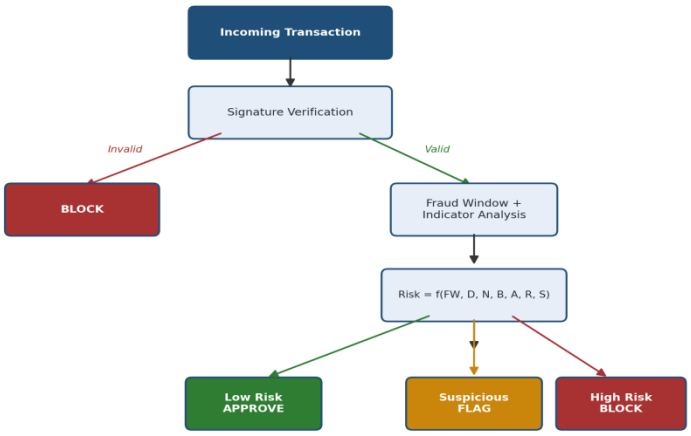


Fig. 4 Fraud detection decision flow

C. Discussion

The prototype shows that offline authorization and blockchain settlement can be combined within one architecture: a user can create and authorize a payment without immediate connectivity, while settlement occurs once connectivity is restored. EIP-712 gives structured, standardized signing, but does not itself provide replay protection, so nonces and transaction identifiers remain necessary [2]. The Fraud Window makes explicit a temporal exposure already recognized in offline-cash research, that double spending may only become observable at reconciliation [9], by measuring Tsync − Tgen directly as an input to the fraud-analysis layer, which combines it with duplicate, nonce, burst, amount, receiver and spending-concentration indicators (Table VI) rather than treating delay alone as proof of fraud.

TABLE VI: COMPARISON WITH EXISTING RESEARCH

Feature	Trad. E-Cash	Jie et al. [1]	Offline BC Wallets	Proposed
Offline capability	Yes	Yes	Yes	Yes
Blockchain settlement	No/limited	Yes	Yes	Yes
Smart contracts	No	Yes	Yes	Yes
TEE	No	Yes	May vary	Not implemented
Fraud Window measurement	Not central	Time assumptions	Not central	Central feature
Rule-based fraud analysis	Varies	Protocol-based	Security checks	Yes

The system should be viewed as a prototype and analytical framework rather than a production-ready or hardware-backed payment protocol: browser localStorage and a JSON ledger are adequate for demonstration but not for production persistence, the local Hardhat chain does not reflect public-network conditions, and the fraud engine is rule-based rather than learned, limitations that motivate the future directions noted in Section V.

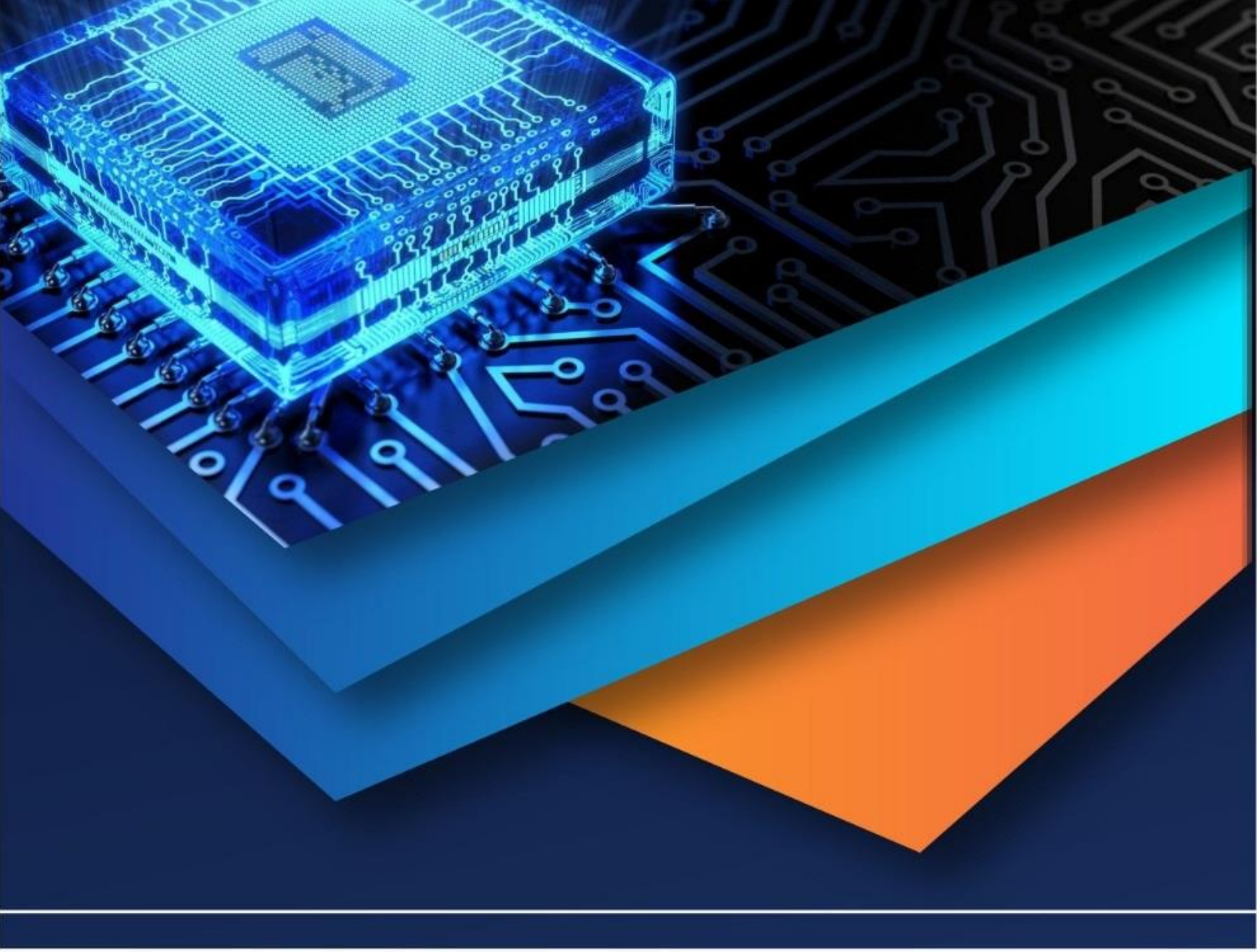
V. CONCLUSION

This paper presented a Secure Blockchain-Based Offline Payment System with Fraud Window Analysis for payment scenarios in which network connectivity is temporarily unavailable. The architecture separates payment authorization from blockchain settlement: transactions are created, signed and stored locally offline, then synchronized with a backend and submitted to an Ethereum smart contract for validation and settlement once connectivity returns. Its main analytical contribution is the Fraud Window, FW = Tsync − Tgen, which is combined with transaction-level indicators, duplicate IDs, nonce reuse, bursts, abnormal amounts, receiver diversity and spending concentration, rather than used alone as proof of fraud.

The prototype demonstrates the feasibility of integrating offline authorization, cryptographic signing, local storage, backend synchronization, smart contracts and temporal fraud analysis, though it remains a proof of concept; hardware-backed key storage (TEE), adaptive/learned fraud detection, offline device-to-device communication (QR/NFC/Bluetooth), production-grade databases, and controlled testnet performance evaluation are identified as the primary directions for future work.

REFERENCES

[1] W. Jie, W. Qiu, A. S. V. Koe, J. Li, Y. Wang, Y. Wu, J. Li, and Z. Zheng, "A Secure and Flexible Blockchain-Based Offline Payment Protocol," IEEE Trans. Computers, vol. 73, no. 2, pp. 408–421, Feb. 2024.
[2] R. Bloemen, L. Logvinov, and J. Evans, "EIP-712: Typed structured data hashing and signing," Ethereum Improvement Proposals, no. 712, 2017.
[3] Ethereum Foundation, "Introduction to Smart Contracts," Ethereum Developer Documentation.
[4] Ethereum Foundation, "Anatomy of Smart Contracts," Ethereum Developer Documentation.
[5] Reserve Bank of India, "Framework for Facilitating Small Value Digital Payments in Offline Mode," RBI/2021-22/146, Jan. 2022, updated Dec. 2024.
[6] Reserve Bank of India, "Offline Retail Payments using Cards/Wallets/Mobile Devices — Pilot," Aug. 2020.
[7] R. Videira, "The offline cash puzzle solved by a local blockchain," IET Blockchain, 2024.
[8] "A simplified scheme for secure offline electronic payment systems," High-Confidence Computing, vol. 1, no. 2, Art. 100031, 2021.
[9] C. S. Fan, "Date Attachable Offline Electronic Cash Scheme," The Scientific World Journal, 2014.
[10] "A fair offline electronic cash scheme with multiple-bank in standard model," J. Chinese Institute of Engineers, vol. 42, no. 1, 2019.
[11] "Secure Electronic Cash Scheme with Anonymity Revocation," Mobile Information Systems, 2016.
[12] A. Dmitrienko et al., "Secure Wallet-Assisted Offline Bitcoin Payments with Double-Spender Revocation," Proc. ACM Asia CCS, 2017.
[13] J. Yoon and Y. Kim, "Offline Payment of CBDC Based on a Trusted Platform Module," J. Cybersecurity and Privacy, vol. 5, no. 2, 2025.
[14] N. Evangelou, R. Chotkan, B. Nasrulin, and J. Decouchant, "Taming Double-Spending in Offline Payments with Reputation-Weighted Loan Networks," arXiv, 2025.
[15] S. Chenna et al., "A Review of Offline Blockchain Transaction Systems," IET Blockchain, 2026.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY