Normandie Université

UNIVERSITÉ CAEN NORMANDIE

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité **INFORMATIQUE**

Préparée au sein de l'**Université de Caen Normandie**

Offline payment requirements for central bank digital currencies

Présentée et soutenue par

ATANGANA ATANGANA THOMAS OLIVIER

Thèse soutenue le 23/03/2026

devant le jury composé de :

M. KHOUKHI LYES	Professeur des universités - ENSICAEN	Directeur de thèse
M. MBAREK NADER	Professeur des universités - Université Bourgogne Europe	Président du jury
M. BARBIER MORGAN	Maître de conférences - UCN - Université de Caen Normandie	Membre du jury
M. GROSCLAUDE LIONEL	Chef d'entreprise - FIME EMEA	Membre du jury
MME TUCCI SARA	Directeur de recherche - CEA Paris-Saclay	Membre du jury
M. AIT SAADI NADJIB	Professeur des universités - UNIVERSITE VERSAILLES ST QUENT	Rapporteur du jury
M. BADONNEL REMI	Professeur des universités - TELECOM NANCY	Rapporteur du jury

Thèse dirigée par **KHOUKHI LYES** (Groupe de recherche en informatique, image et instrumentation de Caen)

MIIS

GREYC
Laboratoire de recherche en sciences du numérique

Offline Payments Requirements for Central Bank Digital Currencies

Olivier Atangana

March 2, 2026

To my mother, Martine Ng'abena, in loving memory.

Abstract

The strong interest in central bank digital currencies (CBDCs) arises in a context of increased digitization of payments and a growing search for more resilient and inclusive solutions. Among the desired features of CBDCs, offline payment constitutes a central challenge. It ensures the resilience of payment systems, promotes financial inclusion, and guarantees transaction continuity in the absence of network connectivity. However, unlike online payments, offline payments for CBDCs impose specific constraints and sometimes conflicting requirements in terms of security, privacy, fraud prevention, auditability, and integration with existing infrastructures. Consequently, this thesis focuses on the analysis and formalization of these offline payment requirements, as well as on the study of technical solutions capable of addressing them in a coherent manner. Accordingly, based on this analysis leading to a structured taxonomy, the thesis introduces several original frameworks illustrating different strategies for satisfying these requirements. The first framework, *PrivTEE-Pay*, relies on a single-ledger architecture and exploits trusted execution environments combined with cryptographic primitives such as blind signatures and zero-knowledge proofs (zk-SNARKs). The second framework extends a conventional payment architecture through the integration of a secure smart card, the *DigiVault card*, coupled with a smartphone. This combined approach also relies on privacy-enhancing technologies and on the fraud detection model *MarkoPayChain*, based on Markov chains. A third framework, *Block-PAD*, proposes a hybrid architecture combining a central ledger for monetary issuance and a blockchain for delayed synchronization of offline transactions. Finally, the thesis complements these contributions with an advanced offline fraud detection framework, based on a combination of expert rules, explainable machine learning models, and hidden Markov chains. Moreover, these different frameworks are experimentally evaluated using simulators and synthetic datasets dedicated to offline CBDC payments. The results show that the proposed solutions make it possible to address the requirements identified in the taxonomy, each through explicit trade-offs. This thesis thus provides concrete contributions to the design of resilient, secure, performant, auditable, and privacy-preserving offline CBDC payment systems that can integrate into existing payment infrastructures.

Keywords— CBDC; Offline Transactions; Security and Privacy; Auditability; Interoperability; Blind Signatures; Zero-Knowledge Proofs (zk-SNARKs); CKKS; Blockchain; Trusted Execution Environment; AI-Based Fraud Detection.

Résumé

L'intérêt manifeste pour les monnaies numériques de banque centrale (MNBC) s'inscrit dans un contexte de numérisation accrue des paiements et de recherche de solutions plus résilientes et inclusives. Parmi les fonctionnalités désirées pour les MNBC, le paiement hors ligne constitue un enjeu central. Notamment pour garantir la résilience des systèmes de paiement, l'inclusion financière et la continuité des transactions en l'absence de connectivité réseau. Cependant, contrairement aux paiements en ligne, les paiements hors ligne pour les MNBC imposent des contraintes spécifiques et des exigences parfois antagonistes en matière de sécurité, de confidentialité, de prévention de la fraude, d'auditabilité et d'intégration aux infrastructures existantes. En conséquence, cette thèse se concentre sur l'analyse et la formalisation de ces exigences de paiement hors ligne ainsi que sur l'étude de solutions techniques permettant d'y répondre de manière cohérente. De la sorte, sur la base de cette analyse dérivant vers une taxonomie structurée, la thèse introduit plusieurs frameworks originaux illustrant différentes stratégies de satisfaction de ces exigences. Le premier, *PrivTEE-Pay*, repose sur une architecture à registre unique et exploite des environnements d'exécution sécurisés combinés à des primitives cryptographiques telles que les signatures aveugles et les preuves à divulgation nulle de connaissance (zk-SNARK). Le second framework étend une architecture de paiement conventionnelle par l'intégration d'une carte à puce sécurisée, la *DigiVault card*, couplée à un smartphone. Cette approche combinée s'appuie elle aussi sur des privacy enhancing technologies et sur le modèle de détection de fraude *MarkoPayChain*, basé sur des chaînes de Markov. Un troisième framework, *Block-PAD*, propose une architecture hybride combinant un registre central pour l'émission monétaire et une blockchain pour la synchronisation différée des transactions hors ligne. Enfin, la thèse complète ces contributions par un framework avancé de détection de fraude hors ligne, fondé sur une combinaison de règles expertes, de modèles d'apprentissage automatique explicables et de chaînes de Markov cachées. Au reste, ces différents frameworks sont évalués expérimentalement à l'aide de simulateurs et de jeux de données synthétiques dédiés aux paiements MNBC hors ligne. Les résultats montrent que les solutions proposées permettent de répondre, chacune selon des compromis explicites, aux exigences identifiées dans la taxonomie. Cette thèse apporte ainsi des contributions concrètes à la conception de systèmes de paiement MNBC hors ligne résilients, sécurisés, performants, auditables, respectueux de la confidentialité et capables de s'intégrer dans des infrastructures de paiement existantes.

Mots-clés— MNBC ; Transactions hors ligne ; Sécurité et confidentialité ; Auditabilité ; Interopérabilité ; Signatures aveugles ; Preuves à divulgation nulle de connaissance (zk-SNARKs) ; CKKS ; Blockchain ; Environnements d'exécution sécurisés ; Détection de fraude basée sur l'IA.

List of Abbreviations

CBDC	Central Bank Digital Currency
DCEP	Digital Currency Electronic Payment
e-CNY	Electronic Chinese Yuan
e-krona	Electronic Krona
POS	Point of Sale
NFC	Near Field Communication
BLE	Bluetooth Low Energy
QR	Quick Response
OTP	One-Time Password
TPS	Transactions Per Second
TEE	Trusted Execution Environment
SE	Secure Element
HSM	Hardware Security Module
PUF	Physically Unclonable Function
OEM	Original Equipment Manufacturer
STA	Secure Trusted Application
AAD	Additional Authenticated Data
AES-GCM	Advanced Encryption Standard in Galois/Counter Mode
PET	Privacy-Enhancing Technology
PII	Personally Identifiable Information
AML	Anti-Money Laundering
CFT	Countering the Financing of Terrorism
KYC	Know Your Customer
SSI	Self-Sovereign Identity
DLT	Distributed Ledger Technology
ZKP	Zero-Knowledge Proof
zk-SNARK	Zero-Knowledge Succinct Non-interactive Argument of Knowledge
CKKS	Cheon–Kim–Kim–Song Homomorphic Encryption Scheme
HMM	Hidden Markov Model
ML	Machine Learning
AI	Artificial Intelligence
PBOC	People's Bank of China
EMV	Europay, Mastercard, and Visa
SEPA	Single Euro Payments Area

Contents

List of Abbreviations i

Acknowledgements xiii

1 Introduction 1
- 1.1 Context and Motivation . . . 2
  - 1.1.1 Security of Offline Transactions . . . 3
  - 1.1.2 Privacy of Transaction Data . . . 3
  - 1.1.3 Auditability . . . 4
  - 1.1.4 Interoperability and Integration . . . 4
- 1.2 Research Goals . . . 4
- 1.3 Methodology . . . 5
- 1.4 Contributions . . . 5
- 1.5 Outline . . . 6

2 Background 8
- 2.1 Introduction . . . 8
- 2.2 Digital Currencies Design in the Context of Offline payment . . . 8
  - 2.2.1 Cryptocurrencies . . . 8
  - 2.2.2 CBDC and E-money . . . 9
- 2.3 Offline Function Generalities . . . 9
  - 2.3.1 Offline Payment Concept. . . . 10
  - 2.3.2 Offline Payment Level . . . 10
  - 2.3.3 Offline Payment Modes . . . 11
- 2.4 CBDC Based Offline Feature . . . 11
  - 2.4.1 CBDC Architecture . . . 11
    - 2.4.1.1 CBDC Types . . . 11
    - 2.4.1.2 CBDC Access Models . . . 12
  - 2.4.2 CBDC System Management . . . 12
    - 2.4.2.1 Single Ledger Based System . . . 13
    - 2.4.2.2 Distributed Ledger Based System . . . 13
    - 2.4.2.3 Hybrid Model Based System . . . 14
  - 2.4.3 Offline Transfer Value . . . 14
    - 2.4.3.1 Offline Payment Scenario . . . 14
    - 2.4.3.2 Communication Technology . . . 14
    - 2.4.3.3 Transaction Authentication method . . . 14
- 2.5 Offline Capable Digital Currencies Protocols Taxonomy . . . 16
  - 2.5.1 Trust Anchor Axis . . . 16
  - 2.5.2 Centralised ledger-Based Protocols . . . 16
  - 2.5.3 Cryptographic-Primitive Axis . . . 17

2.5.4 Target-Property Axis . . . 17
2.6 Conclusion . . . 18
2.7 Chapter Summary . . . 18

3 Analysis of CBDC Offline Requirements and Existing Systems 20
3.1 Introduction . . . 20
3.2 Background and Related work . . . 20
3.3 Adversary Model for Dual Offline CBDC . . . 22
3.3.1 The Double Spending Problem . . . 22
3.3.2 Man-in-the-Middle Attacks . . . 22
3.3.3 Replay Attacks . . . 22
3.3.4 Forgery Attack . . . 22
3.3.5 Side Channel Attacks . . . 23
3.3.6 Physical Tampering and fault-induced attacks . . . 23
3.3.7 Data Breaches . . . 23
3.3.8 Profiling threat . . . 23
3.3.9 Data monetization . . . 23
3.3.10 Dataveillance . . . 23
3.3.11 Other Risks . . . 24
3.4 Frameworks Requirements . . . 24
3.4.1 Security Properties . . . 24
3.4.1.1 No Double Spending . . . 24
3.4.1.2 Unforgeability . . . 24
3.4.1.3 Undeniability . . . 24
3.4.1.4 Verifiability . . . 24
3.4.1.5 Integrity . . . 24
3.4.2 Privacy Feature . . . 25
3.4.2.1 Privacy Concept . . . 25
3.4.2.2 Types and Levels of Privacy . . . 25
3.4.2.3 Unlinkability . . . 26
3.4.3 Auditability . . . 26
3.4.3.1 AML/CFT Regulation Compliance . . . 26
3.4.3.2 Active and passive Audit . . . 26
3.4.3.3 Transparent and confidential Audit . . . 26
3.4.3.4 Traceability . . . 27
3.4.4 Performance . . . 27
3.4.4.1 Baseline metrics . . . 27
3.4.4.2 Scalability . . . 27
3.4.4.3 Other Metrics . . . 27
3.4.5 Extended Features . . . 27
3.4.5.1 Interoperability . . . 27
3.4.5.2 Programmability . . . 28
3.5 Existing Offline-Capable CBDC Solutions . . . 28
3.5.1 Digital Euro . . . 28
3.5.2 E-Cedi . . . 28
3.5.3 E-CNY . . . 29
3.5.4 E-Krona . . . 29
3.6 Conclusion . . . 29
3.7 Chapter Summary . . . 30

4 Offline Payment Based on a Single Ledger **31**
4.1 Introduction . . . 31
4.2 Background and Related Works . . . 31
4.3 PrivTEE-Pay . . . 32
4.3.1 The PrivTEE-Pay Model . . . 34
4.3.1.1 System Participants . . . 34
4.3.1.2 Trust Assumptions . . . 34
4.3.1.3 Threat Model . . . 35
4.3.1.4 Data Representation . . . 35
4.3.1.5 Requirements . . . 36
4.3.2 The PrivTEE-Pay Framework . . . 37
4.3.2.1 System Architecture . . . 37
4.3.2.2 Cryptographic Schemes . . . 37
4.3.2.3 System Operations . . . 38
4.3.2.4 System Analysis . . . 41
4.3.3 PrivTEE-Pay Implementation and Evaluation . . . 42
4.3.3.1 Experimental Setup . . . 42
4.3.3.2 Evaluation Metrics . . . 45
4.3.3.3 Experimental Results Evaluation . . . 46
4.3.4 Discussion . . . 50
4.3.4.1 Comparative analysis . . . 50
4.3.4.2 Addressing User Concerns and Profiling Risks . . . 51
4.3.4.3 Innovative CBDC Payment Approach . . . 51
4.3.4.4 Addressing Protocol Limitations and zk-STARK Advantages 51
4.4 Conclusion . . . 52
4.5 Chapter Summary . . . 52

5 Conventional-Ledger Offline Payments: DigiVaultCard Extension **54**
5.1 Introduction . . . 54
5.1.1 Background . . . 54
5.1.2 Related Work . . . 55
5.2 Materials and methods . . . 55
5.2.1 Key building blocks . . . 55
5.2.1.1 Offline functionality . . . 55
5.2.1.2 Blind Signature . . . 56
5.2.1.3 zk-SNARK Protocol . . . 56
5.2.1.4 Markov Chains. . . . 56
5.2.2 Assumptions and security design . . . 56
5.2.3 The proposed model . . . 56
5.2.3.1 User onboarding procedure . . . 56
5.2.3.2 Redemption protocol . . . 57
5.2.3.3 Offline payment protocol . . . 57
5.2.3.4 Contract checking . . . 60
5.2.3.5 Coin status transition . . . 61
5.2.3.6 Coin Deposit and Exchange . . . 62
5.2.4 MarkoPayChain system introduction . . . 62
5.2.4.1 Mathematical Modeling Approach . . . 62
5.2.4.2 Model Validation . . . 62
5.2.4.3 Interpretation of the Transition Matrix . . . 63

5.3 Performance Evaluation . . . 63
5.3.1 MarkoPayChain Simulation . . . 63
5.3.1.1 Test Scenario . . . 63
5.3.1.2 Results analysis . . . 64
5.3.1.3 MarkoPayChain Model and Hidden Markov Models for Fraud Detection . . . 64
5.3.2 Offline Payment Simulation . . . 65
5.3.2.1 Experimental Setup . . . 65
5.3.2.2 Experimental Procedure . . . 65
5.3.2.3 Evaluation summary and Comparative Study . . . 66
5.4 Discussion . . . 67
5.5 Conclusion . . . 68
5.6 Chapter Summary . . . 68

6 Offline Payments Enabled by Distributed Ledger: Block-PAD Framework 69
6.1 Introduction . . . 69
6.2 Background . . . 70
6.2.1 Securing Privacy of Offline Payments in a Decentralized World . . . 70
6.2.2 Blockchain and CBDC . . . 70
6.3 Related Work . . . 71
6.3.1 Decentralized e-Cash Approaches . . . 71
6.3.2 TEE-Based Offline Payment Solutions and Hardware wallet . . . 72
6.3.3 Compliant Privacy-Centric Model . . . 72
6.3.4 Key building blocks . . . 73
6.3.4.1 Blind Signature . . . 73
6.3.4.2 zk-SNARK . . . 74
6.3.4.3 CKKS . . . 75
6.3.4.4 Trusted Execution Environment . . . 75
6.3.4.5 Wallet-based Digital Identity . . . 75
6.3.4.6 PAD Model . . . 76
6.4 The Block-PAD Model . . . 77
6.4.1 Security Model . . . 77
6.4.2 System description . . . 77
6.5 System Operations . . . 78
6.5.1 Onboarding Phase . . . 79
6.5.2 Withdrawal Phase . . . 81
6.5.3 Contract Negotiation . . . 81
6.5.4 Contract Checking: $V(\Gamma()) \rightarrow \zeta$ . . . 83
6.5.5 Scenario of Bob Paying Carla . . . 85
6.5.6 Online deposit/transfer via PAD . . . 85
6.5.7 Funding/Defunding Operation . . . 91
6.6 Security Analysis . . . 92
6.6.1 Coalition Attacks Against Secret Sharing . . . 93
6.6.2 Coalition Attacks Against Carla's Identity . . . 93
6.6.3 Corrupted Trustees and Validators Risk . . . 93
6.6.4 Bob Fraud Risk: Double Incoming . . . 94
6.6.5 DDoS Attacks Vulnerability Risk . . . 94

6.6.6 Risks Related to Smart Contracts and Inherent to the Blockchain Network . . . 94
6.6.7 Security Properties Evaluation . . . 95
6.6.7.1 Traceability . . . 95
6.6.7.2 Unlinkability . . . 95
6.6.7.3 Undeniability . . . 96
6.6.7.4 Balance Integrity. . . . 96
6.7 Performance Evaluation . . . 96
6.7.1 Experimental Setup . . . 96
6.7.2 Results Analysis . . . 97
6.7.2.1 Digital Identity Management . . . 97
6.7.2.2 Withdrawal Transactions . . . 97
6.7.2.3 Transaction Settlement . . . 100
6.7.2.4 Funding Operation . . . 102
6.8 Discussion . . . 104
6.8.1 Comparative Analysis . . . 104
6.8.2 Full Consecutive Offline Payments Integrated Blockchain (Alice, Bob, Carla) . . . 106
6.8.3 Reconciling Privacy and Regulatory Constraints . . . 106
6.8.4 Hybrid Paradigm Model Based on Regulatory Compliance . . . 106
6.9 Conclusion . . . 106
6.10 Chapter Summary . . . 107

7 Advanced Fraud Mitigation 108
7.1 Introduction . . . 108
7.2 Background and Related Work . . . 109
7.3 Problem Formulation . . . 110
7.4 OffSim: Agent-Based Simulation for Offline CBDC Fraud Detection . . . 110
7.4.1 Model and Method . . . 111
7.4.2 Simulator Overview . . . 112
7.4.3 Parameters and Configuration . . . 113
7.4.4 Threat Model and System Assumptions . . . 113
7.4.5 Fraud Taxonomy in the Proposed Offline CBDC System . . . 115
7.4.5.1 Double Spending . . . 115
7.4.5.2 Fake Merchant or Phantom Terminal . . . 115
7.4.5.3 Timestamp Tampering . . . 115
7.4.5.4 Micro-Top-Up Storms . . . 116
7.5 Hybrid Explainable Rule and Machine Learning Framework for Offline CBDC Fraud Detection . . . 116
7.5.1 Methodology . . . 116
7.5.2 Experimental Setup . . . 116
7.5.3 Dataset Processing . . . 117
7.5.3.1 Data Cleaning . . . 118
7.5.3.2 Distribution Analysis . . . 118
7.5.3.3 Rules-Based Engine Implementation . . . 118
7.5.3.4 Exploratory Analysis . . . 121
7.5.3.5 Data Pipeline . . . 123
7.5.3.6 Feature Engineering . . . 123
7.5.4 Benchmarking Tree-Based Ensemble Models . . . 125

7.5.4.1 Candidate Models for On-Device Fraud Scoring 125
7.5.4.2 Implementation Plan 125
7.5.5 Results and Comparative Analysis 129
7.6 Hidden Markov Chain Model-enhanced Fraud Detection for Offline CBDC 135
7.6.1 Theoretical Foundations 135
7.6.1.1 Hidden Markov Model Formulation 135
7.6.1.2 GBDT-as-Emission: Integration with Machine Learning for offline fraud detection 136
7.6.2 Implementation and Integration 136
7.6.2.1 Construction of Ordered Sequences 136
7.6.2.2 Transition Matrix Estimation in detail 137
7.6.2.3 Forward Filtering on ML Outputs 137
7.6.3 Experiments and Results 137
7.6.3.1 LightGBM + HMM 137
7.6.3.2 Comparative Results 140
7.7 Discussion 141
7.8 Conclusion 143
7.9 Chapter Summary 143

8 Conclusion **145**
8.1 Summary of Technical Contributions 145
8.2 Perspectives 146
8.2.1 Extended Requirements 146
8.2.2 PrivTEE-Pay 147
8.2.3 Digivault-card MarkoPayChain 147
8.2.4 Block-PAD 147
8.2.5 Advanced Fraud Mitigation 147

A List of Publications **161**

List of Figures

1.1 History of offline payment. . . . 2

2.1 Different levels of offline payment. . . . 10
2.2 Different modes of offline (Adapted from: [1]) . . . 11
2.3 One-tier and two-tier retail CBDC architecture. . . . 12
2.4 Hybrid retail CBDC architecture . . . 12
2.5 Comprehensive layered architecture of offline digital currency protocols across centralised, distributed, and hybrid infrastructures. . . . 13
2.6 Design space of offline-capable digital currency protocols, structured around five key axes: levels of offline, infrastructure, trust hardware, cryptographic primitives, and target properties. . . . 19

3.1 Different types of privacy in offline payment. . . . 26

4.1 logical data flows between all system actors . . . 35
4.2 Architecture of transferable electronic payment . . . 37
4.3 CBDC withdrawal based eCash1 process . . . 40
4.4 Offline CBDC Transaction based PrivTEE-Pay . . . 40
4.5 A sequence diagram of the detailed transaction between the sender and the receiver using our secure offline payment protocol . . . 45
4.6 Online Withdrawal Results . . . 46
4.7 Performance evaluation of online withdrawal operations under increasing offline transactions. . . . 47
4.8 Establishment of a connection between Alice's and Bob's NFC devices . . . 47
4.9 Performance evolution of offline NFC transactions under increasing workloads. . . . 48
4.10 ZK proof computation characteristics during offline transaction processing. 48
4.11 Distribution of Encrypted Message Sizes during Offline Transactions . . . 49
4.12 Performance metrics of deposit operations under increasing offline workloads. 49
4.13 Throughput Scalability vs Number of Transactions . . . 49

5.1 Onboarding Registration Phase . . . 61
5.2 Offline Transaction block between Alice and Bob . . . 61
5.3 Heatmap of markov chain transition probabilities . . . 63
5.4 Fraud Detection Analysis for Offline CBDC Payments . . . 64
5.5 Results for the simulation with 50 transactions. . . . 65
5.6 Results for the simulation with 75 transactions. . . . 66
5.7 Results for the simulation with 75 transactions. Subfigures: . . . 66

6.1 Architecture of transferable CBDC based PAD. . . . 71
6.2 Simplified PAD schematic. See this link for more details. . . . 76

6.3 Overview of the proposed model. In addition to the client-facing flows, the commercial bank shard is connected to the central bank core ledger through a ledger shard link. This connection, consistent with the two-tier architecture in Fig.6.1, provides an implicit channel for transaction verification and audit. . . . 78
6.4 Simulation of users digital identity onboarding . . . 81
6.5 Offline transaction between Alice and Bob . . . 83
6.6 Overview of transaction verification result between Alice and Bob. . . . 85
6.7 Data structure for CBDC consecutive offline transaction . . . 85
6.8 Online deposit simulation . . . 87
6.9 Result simulation of decryption process . . . 89
6.10 Smart contract deployment for transaction in PAD Ledger . . . 91
6.11 Overview of recording transaction in PAD Ledger . . . 91
6.12 Funding transaction following the deployment of the smart contract. . . . 91
6.13 Funding transaction following the deployment of the smart contract. . . . 92
6.14 PAD system-centric Blockchain for CBDC payments. Annotations A1–A6 indicate the main process steps: A1 – Onboarding (Algs. 1–2), A2 – Withdrawal (Alg. 3), A3 – Offline Alice→Bob (Algs. 4–5), A4 – Offline Bob→Carla (Alg. 6), A5 – Online deposit/transfer via PAD (Algs. 7–10), A6 – Funding/Defunding (Alg. 11). See also Table 6.2 for a summary crosswalk. . . . 93
6.15 Digital identity onboarding and verification performance under varying user scales. . . . 98
6.16 Withdrawal results . . . 100
6.17 Offline transactions performance(Alice-Bob) results . . . 102
6.18 Offline transactions performance(Bob-Carla) results . . . 102
6.19 Funding transaction performance . . . 104

7.1 State-level architecture of OffSim-CBDC showing the main modules and interactions. . . . 113
7.2 Class diagram of the OffSim-CBDC agent system. . . . 114
7.3 Fraud oriented transaction generation pipeline in OffSim-CBDC. . . . 115
7.4 The proposed fraud detection framework . . . 116
7.5 Univariate empirical distributions of the most relevant numerical attributes in the raw transaction log. Each panel displays an un–transformed histogram with `bins`=80 and a kernel–density *rug* for visual reference. . . . 119
7.6 Empirical distributions of the principal *categorical / binary* variables. Each bar reports the relative frequency of the corresponding modality over the entire raw dataset. . . . 121
7.7 Class Imbalance visualization . . . 123
7.8 Distribution analysis after the rule engine process . . . 124
7.9 Pearson correlation matrix computed on the *post-rule engine* dataset (only offline transactions with `alert`=0). The diagonal elements are 1.0 by definition. Warm colors indicate positive correlations, while cool colors indicate negative correlations. . . . 125
7.10 Complete inference performance benchmarking of LightGBM and XGBoost (Tests conducted on 100,000 samples with calibrated probabilities). . . . 133
7.11 Global SHAP feature importance for LightGBM (a) and XGBoost (b). . . . 133
7.12 Comparison of LIME explanations for LightGBM and XGBoost in offline CBDC fraud detection. . . . 134

7.13 Comparison of permutation feature importance for LightGBM, XGBoost, and the Hybrid model in offline CBDC fraud detection. 135
7.14 LightGBM-HMM Hybrid Architecture 136
7.15 HMM with GBDT-as-Emission Hybrid 136
7.16 Transition matrix of the Hidden Markov Model (HMM) for legitimate and fraud states. The probabilities are as follows: from legitimate (0) to legitimate (0): 0.6669; from legitimate (0) to fraud (1): 0.3331; from fraud (1) to legitimate (0): 0.6663; and from fraud (1) to fraud (1): 0.3337. The matrix exhibits a marked symmetry in transitions, which indicates a weak temporal dependency between states. 140
7.17 Comparison between LightGBM and LightGBM + HMM: (a) Metrics and (b) Error Trade-off 141

List of Tables

2.1 Advantages of Hybrid CBDC System. . . . 15
2.2 Offline payment scenarios. . . . 15
2.3 Target properties of offline-capable protocols . . . 18

3.1 Comparison Table of Literature Survey . . . 21
3.2 Comparative Overview of Selected Digital Currency Systems . . . 29

4.1 Comparative Study of CBDC Offline Function Approaches . . . 33
4.2 Notations and Descriptions . . . 38
4.3 Comparison of Proposed Offline Payment Solutions Based on Conventional Ledger . . . 53

5.1 Key Variables and Their Descriptions . . . 58
5.2 Functions Utilized in the Payment Protocol . . . 60
5.3 Comparison of Metrics Across 50, 75, and 100 Transactions . . . 65
5.4 Proposed Solution Comparison . . . 67

6.1 Comparative Study of Various Solutions . . . 74
6.2 Crosswalk of process steps, algorithms, and sections . . . 79
6.3 Notation of RIO Onboarding . . . 81
6.4 Notations of Offline Sealed Transaction . . . 83
6.5 Notations for Algorithm 6 . . . 87
6.6 Notations for Algorithm 9. . . . 89
6.7 Notations for Algorithm 11 . . . 92
6.8 Onboarding and Identity Verification Metrics (averaged over runs). . . . 98
6.9 Results of the first 10 withdrawals for amounts of 50, 75, and 100. . . . 99
6.10 Results of offline transaction simulation (Alice to Bob) . . . 100
6.11 Results of payment simulations between Bob and Carla . . . 101
6.12 Results of Funding Simulation . . . 103
6.13 Comparative performance evaluation of offline payment solutions. . . . 105

7.1 OffSim-CBDC leveraging PaySim features and logic . . . 112
7.2 Dataset description . . . 117
7.3 Observed proportions for principal fraud-related binary attributes . . . 118
7.4 Summary statistics of numerical columns before transformation . . . 120
7.5 Category proportions for key binary/categorical fraud indicators . . . 120
7.6 Description of variables used in algorithm . . . 122
7.7 Impact of the `alert` column on downstream ML . . . 123
7.8 Summary of Engineered Features . . . 126
7.9 Input Features for Machine Learning Database . . . 126
7.10 Selected LightGBM Hyperparameters and Their Descriptions . . . 127

7.11 Selected Random Forest Hyperparameters . . . 128
7.12 Selected XGBoost Hyperparameters . . . 129
7.13 Performance of the Rule Engine . . . 130
7.14 Characteristics of the SMOTENC-Balanced Dataset . . . 130
7.15 Performance Comparison of Baseline Models . . . 131
7.16 Model artifacts and loading information for LightGBM and XGBoost. . . . 132
7.17 Inference benchmark by batch size for LightGBM and XGBoost (dataset: 100,000 samples; calibration: True). . . . 132
7.18 Comparative Analysis of LIME Explanations for Offline CBDC Fraud Detection . . . 134
7.19 Notations Used in Algorithm 1 (HMM with LightGBM-as-Emission) . . . 138
7.20 Build sequence metadata (seq_id, time) . . . 140
7.21 Comparative Performance: LightGBM vs. LightGBM+HMM . . . 141
7.22 Unified Comparison of Fraud Detection Approaches in Literature and Our Work . . . 142

Acknowledgements

Les travaux de thèse menés conjointement au sein du laboratoire GREYC de l'ENSICAEN, à l'Université de Caen Normandie, et au sein de Fime, m'ont permis de développer une expertise technique solide tout en m'enrichissant tant sur le plan professionnel qu'humain. À ce titre, je tiens à exprimer ma profonde reconnaissance à toutes celles et ceux qui, de près ou de loin, ont contribué à l'édification et à l'aboutissement de cette thèse.

J'exprime tout d'abord ma sincère gratitude à mon directeur de thèse, Lyes Khoukhi. Ses orientations technico-méthodologiques, la pertinence de ses conseils et son exigence scientifique m'ont accompagné tout au long de ce travail doctoral et ont grandement contribué à mon évolution scientifique. Je garde également un souvenir très marquant de sa disponibilité constante et de son écoute attentive face à mes préoccupations. Cette même disponibilité et cette bienveillance caractérisent également Morgan Barbier, mon co-encadrant de thèse, dont les conseils académiques et professionnels ont été d'un apport considérable. Je remercie ensuite l'ensemble de mes collègues du laboratoire GREYC, en particulier les membres de l'équipe SAFE, ainsi que tout le personnel de l'ENSICAEN et de l'Université de Caen Normandie, qui œuvrent quotidiennement à l'amélioration de l'expérience de recherche des doctorants.

Je tiens par ailleurs à remercier spécialement Lionel Grosclaude, CEO de Fime et membre du jury de cette thèse. Puis, Noël Cathérine, SVP Services, Stéphanie El Rhomri, Rolf Baaij ainsi que Joël Di Manno pour leur soutien multidimensionnel tout au long de la réalisation de mes travaux de recherche. À travers eux, j'exprime également ma reconnaissance à l'ensemble de mes collègues de Fime, dont l'affection, la sympathie et le soutien moral ont contribué à créer un environnement de travail stimulant et bienveillant. Je souhaite adresser un remerciement tout particulier à Willy Royer, mon premier manager chez Fime. Son accompagnement et son soutien ont grandement facilité mon intégration professionnelle.

Je remercie également mes superviseurs lors de mes stages de recherche doctorale à l'étranger, notamment Zakaria Abou El Houda à l'Institut National de la Recherche Scientifique (INRS) du Canada, ainsi que Nima Afraz à l'University College Dublin (UCD). Tous ont accepté de m'accueillir au sein de leurs laboratoires respectifs et de m'accompagner dans l'approfondissement de certains aspects clés de mes travaux de recherche.

Ma gratitude s'adresse aussi au corps enseignant et administratif de Polytech Dijon, dont je suis diplômé en tant qu'ingénieur en informatique et électronique. Je tiens à remercier tout particulièrement Nader Mbarek, membre du jury de cette thèse qui m'a initié à la rigueur et à la grammaire de la recherche scientifique, ainsi que Wahabou Abdou, qui m'a recommandé pour cette thèse.

Je souhaite également exprimer ma reconnaissance à Armand Claude Abanda, titulaire de la Chaire UNESCO d'Afrique centrale ACCES-TIC, qui m'a encouragé à m'orienter vers les études en informatique et à en découvrir les nombreuses opportunités.

Enfin, je remercie du fond du cœur ma famille, et tout particulièrement mes parents,

Gervais Atangana et Martine Ng'Abena, qui ont toujours pourvu à mon éducation et à mes besoins avec un soutien indéfectible. Sans eux, je n'aurais sans doute pas surmonté les nombreuses épreuves rencontrées au cours de mon parcours. Mes sœurs et mon frère, malgré la distance qui nous sépare, ont toujours été présents lors des étapes majeures de mon cheminement universitaire, pour m'encourager, me soutenir et me témoigner leur affection. Je remercie également les familles Eydant à Dijon, Fouda à Caen, mes amis et connaissances, dont la présence, la complicité, la joie et le bonheur partagés au fil de nos échanges et de nos aventures ont grandement contribué à mon équilibre personnel.

Chapter 1

Introduction

Recent years have seen the emergence of digital technologies such as smartphones, cloud computing, blockchain, and artificial intelligence. Subsequently, the growing adoption of these digital technologies heuristically revolutionizes and transforms multiple domains of daily life, including payment systems with the advent of central bank digital currencies. Thus, CBDCs (Central Bank Digital Currency), which are a digitization of fiat currency, are developing within the orbit of the technological transition movement inherent in the global economy, marked by a preferential option for digital transactions. Following the 2008 financial crisis, which was succeeded by the all around flourishing of cryptocurrencies and the emergence of the coronavirus pandemic, many countries and organizations accelerated their digital transformation to align with electronic payment systems. Nearly 80% of payments were made via smartphone in 2021 in China [2].

In the proliferation of digital currencies, the digital issuance trend of CBDCs, given its advantages in terms of speed, cost, trust, and stability, is a real opportunity and a guarantee of safety for public and private financial actors [3]. Hence, CBDCs are a credible alternative to the volatility of cryptocurrencies and a complementary option for electronic payment in an economic context undergoing full digital transition. The multiple initiatives and research projects like the digital euro, E-CNY, E-Krona, E-Naira, etc., implemented by contributions from academic, governmental, and private institutions sufficiently demonstrate the growing interest in CBDCs. Driven by the wings of accessibility and resilience, the CBDC revolution implies enabling greater access to financial services, particularly for unbanked or underbanked populations. In this vein, offline payment is an interesting option as it provides a reliable digital payment method without dependence on continuous online infrastructure. Moreover, offline transactions offer resilience against network outages and service interruptions, ensuring continuity of financial operations even in the absence of connectivity. Additionally, deploying offline mode is favorably received in areas where internet connectivity is absent or intermittent, such as rural areas and regions with insufficient network infrastructure.

However, although CBDCs offer great promises, their deployment involves facing a plurality of complex challenges, notably security, privacy, interoperability, and offline transactions. Thus, offline payments imply that the payer and the beneficiary are not connected to the internet or a central network when the transaction is carried out [1]. In this context, ensuring security, efficiency, and confidentiality while complying with regulatory constraints on anti-money laundering (AML) and countering the financing of terrorism (CFT) is a paramount.

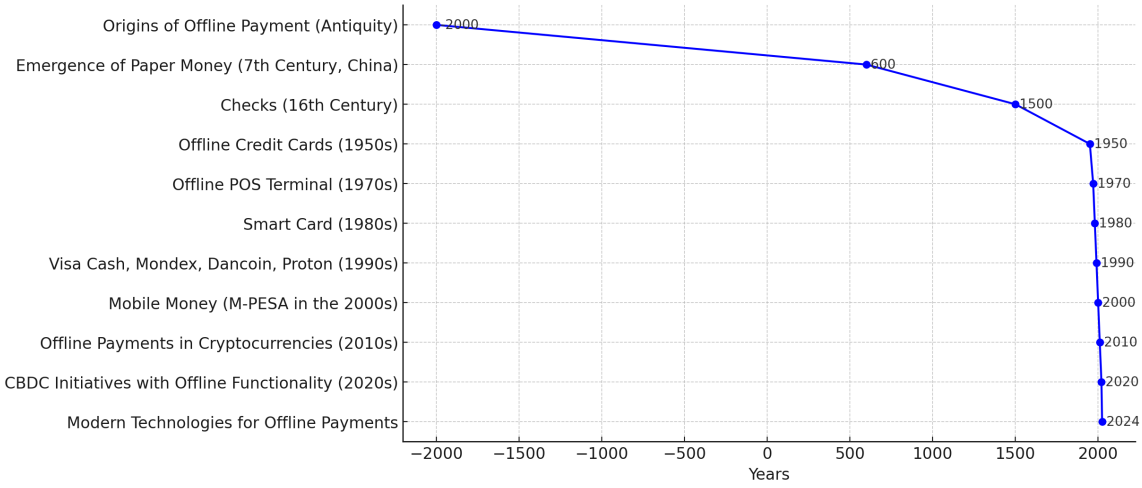


Figure 1.1: History of offline payment.

1.1 Context and Motivation

The origins of offline payment trace back to Antiquity with the use of precious metal coins in commercial exchanges. These early forms of physical currency, cumbersome to transport, gradually gave way to paper money in China by the 7th century, which was more flexible and easier to use. In the 16th century, traditional offline payment systems emerged with the introduction of checks. This check system enabled real-time transactions without requiring cash. The 1950s marked the debut of the first "offline" credit cards, like the Diners Club card, where transaction information was collected and sent for later processing [4]. The 1970s witnessed a significant shift with the transition to electronic payments, characterized by the development of offline Point-of-Sale (POS) terminals. These terminals facilitated card-based transactions without a real-time connection to the bank, which would later process the transaction data. A decade later, security was enhanced with the introduction of smart cards, securing offline transactions in banking and telecom sectors [5].

In the 1990s, Visa experimented with an offline payment system known as Visa Cash, which used a stored-value card. This card could be reloaded at ATMs and used for offline transactions with merchants. Similar systems, such as Mondex, Dancoin, and Proton, were also introduced during the same period, highlighting a broader movement toward digital cash and secure offline electronic payments [6]. In the 2000s, underbanked countries gained the ability to perform transactions via mobile phones using telecom networks from their mobile service providers. This era of "mobile money" was exemplified by M-PESA [7]. In 2009, a major shift occurred with the launch of the first cryptocurrency, Bitcoin, which operates on a new infrastructure allowing peer-to-peer online transactions. Although designed for online use, offline Bitcoin transactions began to appear around 2010, enabling cryptocurrency transactions without a network connection via bearer paper with private keys. Most solutions utilize communication methods like NFC (Near Field Communication), Bluetooth, or QR code exchange. Since 2020, several central banks have initiated projects to integrate offline functionality into their digital currencies, particularly to ensure financial inclusion in areas with limited or intermittent connectivity. Fig. 1.1 provides a brief overview of the offline payment history.

The main motivation of this thesis lies in addressing the challenges related to offline payments for CBDCs by establishing a wallet that offers the possibility of maintaining a

cash-like experience and improving financial inclusion, access, and resilience, particularly in regions where internet access is non-existent, limited, or intermittent. The detailed motivations underlying this research are as follows:

1.1.1 Security of Offline Transactions

The security requirements when considering implementing the offline payment function for digital currencies are defined in terms of protection against Double Spending [8]. It is crucial for offline payments to avoid the risk of double spending, where the same unit of digital currency could be spent multiple times before synchronization with the central network. In the proposals of [9, 10] it is impossible to prevent double spending without the intervention of a third party to finalise the transaction. Consequently, the recipient cannot spend immediately the monetary unit received. Given this limitation, I aim to address it by implementing a payment system based on transferable ownership. In this way, the receiver can spend right away the unit he has just obtained.

Another requirement is unforgeability. It turns out that digital currencies appear more susceptible to counterfeiting than hard currency [11]. Yet a digital currency should not only replicate the properties of cash but also surpass it, implying protection against forgery. Therefore, the proposed solution must be able to mitigate the risks of counterfeiting. Legislative reinforcement approaches aimed at substantially reducing the quantity of counterfeit money in circulation are interesting yet insufficient. The consequences caused by the circulation of counterfeit currencies can be disastrous. However, to overcome this flaw, I envisage a technical approach based on authenticating digital money through algorithmic encryption.

The security of offline payments also underscores the importance of verifiability, as the adopted payment protocol must ensure that transactions are recorded and not falsified. Cryptocurrencies profit extensively from distributed ledger technology to satisfy the property of verifiability [12]. In contrast, for CBDCs authors in [13] suggest integrating a KYC-based process to detect suspicious transactions. Yet such a process can, in many respects, erode privacy. Still, I aim to tackle this shortcoming by leveraging technologies that ensure verifiability without accessing shared information, or by employing eKYC based verification processes. Moreover, for optimal security, accountability or non-repudiation is also necessary. It guarantees, on the one hand, that the receiver has received authentic digital currency during payment; on the other hand, it ensures that the payer cannot deny having made the payment. Several studies emphasise a trusted party to satisfy the non-repudiation requirement [14, 15]. I intend to strengthen this property by integrating a proof-of-payment mechanism coupled with other authentication and authorisation mechanisms.

1.1.2 Privacy of Transaction Data

Digital payment security is a core component of the ecosystem. However, for several decades the design of payment systems has favoured security at the expense of privacy. With the emergence CBDCs we witness a shift that places both security and privacy at the forefront. In fact, preserving privacy is an essential aspect of CBDC deployment. A survey carried out by the European Central Bank revealed that users and payment system actors prize privacy highly [16]. Because users increasingly worry about the confidentiality of their financial transactions, payment professionals are actively developing privacy-centric solutions. Consequently, an offline payment function can reinforce the privacy attribute

in CBDC transactions. I therefore aim to ensure that transaction data remain private and shielded from profiling risks and intrusive surveillance such as citizen monitoring or Big Data-driven dataveillance [17, 18]. Moreover, by integrating privacy enhancing technologies within a cryptographic framework we can overcome the challenges inherent in protecting user privacy. Three guiding principles will shape the proposed frameworks to guarantee privacy: privacy by design, privacy by default, and the fortification of security through privacy measures.

1.1.3 Auditability

Digital currencies issued by central banks are, by definition, regulated money. Thus they must verify that every transaction complies with legal requirements, particularly those related to anti-money laundering (AML) and counter-terrorist financing (CTF). If offline payments were left unchecked, they could become a breeding ground for illicit activity; therefore, it is incumbent upon us to make them audit-ready. Yet many experts consider such a requirement incompatible with a privacy-focused system [10]. Against this tide, I aspire to build offline frameworks that reconcile regulatory obligations and privacy in a balanced manner exploring avenues such as contextual privacy.

1.1.4 Interoperability and Integration

Interoperability begins by guaranteeing compatibility with existing systems. CBDCs should not be viewed as adversaries but rather as complements to current payment methods. Deploying offline payment solutions demands smooth integration into central bank infrastructure and seamless interaction with both legacy and modern payment networks, including blockchain platforms. Equally important is data synchronisation: maintaining the integrity and precision of financial records is vital for verifiability and audit trails. Hence, ensuring reliable and secure synchronisation of transaction data once connectivity resumes is a key requirement.

1.2 Research Goals

The primary objective of this thesis is to propose a secure, confidential framework for offline payments using CBDCs. Specifically, I aim to:

1. Define the security and privacy requirements for an offline electronic wallet. It is the matrix of electronic currency, in all its contours and with its security constraints. This assumes initially ensuring the security services of a traditional computer system, namely integrity, non-repudiation, access control, confidentiality, availability, identity, and authentication. Additionally, it will be necessary to integrate a feature to protect against double spending and a hardware and/or software privacy protection module.

2. Design a lightweight communication protocol that guarantees transaction authenticity without requiring constant connectivity. The envisaged protocol must ensure transaction authenticity (obviously without overburdening the execution of the transaction itself) through encryption. Encryption tools such as asymmetric encryption, elliptic curve cryptography, and cryptographic signature are very interesting avenues, for the transmission of payment data (account-based or token-based), a

study of protocols such as the NFC protocol, BLE protocol, or the use of QR codes with single or mutual authentication should be considered.

3. Establish a synchronization mechanism between offline wallets and the central bank's network.The central network, for this purpose, is the central bank, the trusted third party, fund provider, and guarantor of the financial stability of the system. The main challenge will be to coexist offline and online electronic wallets, combine their exchanges so that fund transfer and recovery are achieved properly, simply, and transparently for the actors.

4. Integrate advanced proactive fraud detection mechanisms into the payment flow. Cyber attackers and cyber fraudsters are competing in skill to compromise digital currency systems. Consequently, as their techniques develop, innovation must provide a consequent response. In this case, to counter fraud techniques such as double spending, double incoming, replay, etc. Integrating an embedded system based on a stochastic method as well as machine learning within the proposed framework will ensure that illicit transactions are detected and, if necessary, canceled before their completion.

5. Ensure compliance with current privacy regulations while maintaining auditability. The proposed framework will comply with privacy and data protection standards, ensuring easier adoption and acceptance by users and regulators. By integrating privacy protection principles from the design phase and by default, the framework will address the growing concerns of users regarding the confidentiality of their financial transactions. For transparency needs, the system will be auditable by exploiting mechanisms compliant with contextual privacy.

1.3 Methodology

This thesis aims to ensure optimal security and confidentiality for the flow of CBDC payment transactions in offline mode. To achieve this, the research methodology relies on the dynamics of three action verbs: understand, analyze, and implement. Specifically, it involves immersing oneself in the latest research advances on offline CBDC projects through literature studies, participation in seminars, and technological monitoring on the subject. Then, in the analysis phase, it is necessary to identify the shortcomings and challenges of current CBDC solutions and examine the relevance of modern technologies used for offline payments involving digital currencies. Finally, it will be essential to implement a viable offline payment solution for the CBDC ecosystem. This final aspect will be structured into three moments: developing the solution by establishing a payment protocol, testing and validation, which will involve testing the developed solution in a simulated environment to verify its effectiveness and security. Finally, ensuring the integration of solutions with existing payment infrastructures by developing APIs and synchronization interfaces.

1.4 Contributions

This dissertation discusses and explain the offline payments requirements for CBDC to become a viable resilient solution when there is not internet connectivity or in the case of network outages. This thesis delivers four major novel frameworks that collectively

address the goals outlined above. The first framework, `PrivTEE-Pay`, is based on a single-ledger oriented architecture. This approach allows already established banking infrastructures to integrate CBDCs and offline payment without resorting to a new infrastructure as an extension or replacement. `PrivTEE-Pay` relies on the token-based retail CBDC (rCBDC) transaction format, which is designed to safeguard privacy.It offers a range of advantages, including end-to-end security and full privacy lifecycle protection. With this in mind, we aim to propose a new offline payment approach CBDC on a mobile device that combines a blind signature, the zk-SNARK protocol, and other strands from the literature such as cryptography, open-source technology, digital signatures, public key infrastructure, privacy-by-design, and digital certificates. This approach guarantees security against double-spending issues while preserving data privacy.

Furthermore, the second framework is equally based on a two-tier architecture with an offload wallet integration. It highlights an approach centered on a smart card, named `DigiVault card`, which leverages the computational capabilities of a mobile phone enhanced by recent advances in chip technology. This heterogeneity provides a reliable and secure environment for storage and multifactor authentication. Moreover, it enables proactive fraud protection by incorporating the Markovian model `MarkoPayChain`. From a privacy standpoint, it enhances privacy by reducing profiling risks through the inclusion of Privacy-Enhancing Technologies (PETs) such as blind signatures and zk-SNARKs. Finally, it is auditable by regulatory authorities via the management of zk-SNARK payment proofs recorded in the payment history.

`Block-PAD` combines a single ledger for minting with a distributed ledger to synchronize offline payments through blockchain. Alice transfers one unit of CBDC to Bob using a secure element; Bob can then forwards it to Carla, who receives her payment online. The framework adds a digital wallet, a supervisory authority, and incorporates homomorphic encryption, zk-SNARK, and blind signature to preserve confidentiality while meeting regulatory requirements. It guarantees unlinkability, immutability, balance integrity, prevention of double spending, and facilitates interoperability with traditional payment systems via an eID wallet.

Finally, we propose an advanced fraud detection framework. This framework complements the cryptographic mechanisms proposed in the previous frameworks. Thus, it articulates a rule engine with a machine learning model to detect fraud cases that are not covered by cryptography alone.

1.5 Outline

The remainder of this dissertation consists of seven chapters organised as follows: Chapter 2 introduces the background on digital currencies and offline payment modes. It presents, in particular, the definitions commonly accepted in academic and industrial literature. Moreover, it describes the various CBDC architectures that can host an offline payment function. Finally, it delivers a unified taxonomy of offline-capable digital currency protocols based on current technologies, systems, and implemented algorithms.

Chapter 3 details the requirements for offline payment by analysing them against threats, risks, and vulnerabilities that could compromise the system in this environment. This study focuses primarily on four axes: security, privacy, auditability, and performance. It concludes with a comparative assessment of several offline payment solutions under deployment.

Chapter 4 introduces PrivTEE-Pay, an offline payment framework based on a classic single-ledger architecture. Following a methodology structured around three axes, we

first present PrivTEE-Pay itself, which leads us to describe the actors, threat model, data representation, and requirements. The second axis focuses on a deep study of the framework, especially its architecture, cryptographic schemes, and operation execution. The third axis concerns system analysis and performance evaluation.

Chapter 5 describes an offline payment framework that integrates a smart card as an extension within a standard architecture. We specify the key building blocks of the system and then describe the proposed model, highlighting both the architecture and operational components. Additionally, we introduce MarkoPayChain, a fraud detection system based on Markov chains. Finally, we evaluate framework performance using metrics that significantly impact the system.

Chapter 6 presents the Block-PAD framework, an offline payment solution built on a hybrid architecture. We begin by introducing the technical background required for deploying the solution. Then we describe the model from a security perspective and its architectural representation. Next, we explain in detail all system operations. Finally, we analyse security and evaluate system performance.

Chapter 7 presents a hybrid, explainable fraud detection framework designed for CBDC offline payments. Relying on a rule engine, machine learning models such as LightGBM and XGBoost, and hidden Markov chains, this chapter complements the previously established architectural and cryptographic foundations by adding an additional security layer while preserving the transparency required by regulatory demands. It details the construction of a specific dataset via OffSim, the design of the detection pipeline, and the integration of explainability tools such as SHAP and LIME. Finally, it offers a benchmark of the various evaluated solutions.

The final chapter 8 concludes this manuscript by summarising the thesis contributions and opening perspectives for future research.

Chapter 2

Background

2.1 Introduction

In the previous chapter we highlighted the stakes, motivations, and objectives of this dissertation concerning offline payment requirements for central bank digital currencies. Consequently, designing systems capable of meeting offline payment demands necessitates a precise understanding of the environment and the constituent mechanisms of offline payment. Therefore, this chapter aims to present the theoretical, technological, and architectural foundations related to offline payment within the context of digital currencies. It revisits, once again, the architectural, organisational, and cryptographic choices recorded in the state of the art. To achieve this, the chapter is structured around several complementary pillars: the design principles of digital currencies in an offline payment context presented in Section 2.2; generalities and categories of offline operation, including concepts, levels, and modes presented Section 2.3 ; CBDC architectures and their technical characteristics described in Section 2.4 ; a taxonomy of offline payment protocols that identifies existing approaches and structuring trends presented in Section 2.5; and finally, Section 2.6 concludes the chapter and Section 2.7 presents the Chapter Summary. This review will serve as an analytical foundation for analysing the core requirements of a successful offline payment in CBDC.

2.2 Digital Currencies Design in the Context of Offline payment

2.2.1 Cryptocurrencies

Cryptocurrency refers to private virtual currencies created, exchanged, and stored electronically and autonomously through distributed ledger technology and cryptographic systems that ensure transaction security. Essentially, cryptocurrencies operate independently of a trusted third party, such as central banks, and are not linked to public money (e.g., bank deposits or cash), which has legal tender status. However, there are digital assets called "stablecoins" designed to maintain parity with a legal reference currency (e.g., CoinVertible). This stability is ensured through the controlled issuance or withdrawal of new units in response to market price fluctuations [19].

Cryptocurrencies are accepted by individuals who use them as a medium of exchange to send and receive funds. All transactions are confirmed through cryptographic processes, recorded on a distributed ledger, and stored in an electronic wallet. Today, multiple types

of cryptocurrencies exist, such as Bitcoin, Ethereum, and Litecoin, though Bitcoin, created in 2008 and launched in 2009, remains the most prominent [20]. All cryptocurrencies are based on distributed blockchain technology.

2.2.2 CBDC and E-money

2.2.2.0.1 E-money.

The advent of electronic money in the 1990s sparked a growing interest in offline functionality for digital payments. David Chaum, often regarded as the father of e-cash, identified its potential in terms of speed and cost [21]. Since then, offline functionality has been considered from the perspective of enhancing anonymity and security. Numerous protocols have been developed over the following decades to enable untraceable transactions that are robust against double-spending attempts and man-in-the-middle attacks [22–24].

Simultaneously, communication methods for offline exchanges, such as NFC, Bluetooth, and QR codes, have been widely utilized [25, 26].

2.2.2.0.2 CBDCs in the Digital Finance Landscape.

According to [27], a CBDC is a "digital form of central bank money that is distinct from balances in traditional reserve settlement accounts." Interest in CBDCs is increasing significantly with the wave of technological innovations in the global economy, characterized by a preference for digital transactions. Following the emergence of the COVID-19 pandemic, many countries and organizations accelerated their digital transformation to align with electronic payment methods. In 2021, nearly 80% of payments in China were made via smartphones [2]. Amid the rise of dematerialized currencies, the digital trend toward issuing CBDCs offers notable opportunities in terms of financial inclusion, speed, cost, trust, convenience, flexibility, and stability [3]. The widespread adoption of cryptocurrencies, since the advent of Bitcoin has accelerated research and development efforts for CBDCs. As a digital representation of fiat money, CBDCs provide a credible alternative and complementary option in the field of electronic payments. This is why numerous central banks worldwide have initiated or commissioned research on CBDCs [28–30]. However, while the use of CBDCs holds great promise, their implementation faces multiple interconnected challenges, including interoperability, instant and peer-to-peer payments, cross-border payments, anti-money laundering (AML), new policies, security, CBDC wallets, offline payments, and privacy, which could either catalyze or hinder CBDC adoption. [31–35].

2.3 Offline Function Generalities

Offline payment is essential to reproduce certain properties of cash and ensure operational resilience, allowing transactions to continue even in areas without internet connectivity or in cases of central infrastructure failure. These multiple advantages make offline payments a popular feature among the general population, especially among underbanked groups. However, implementing secure and efficient offline payments requires optimal solutions in terms of security, privacy protection, and performance.

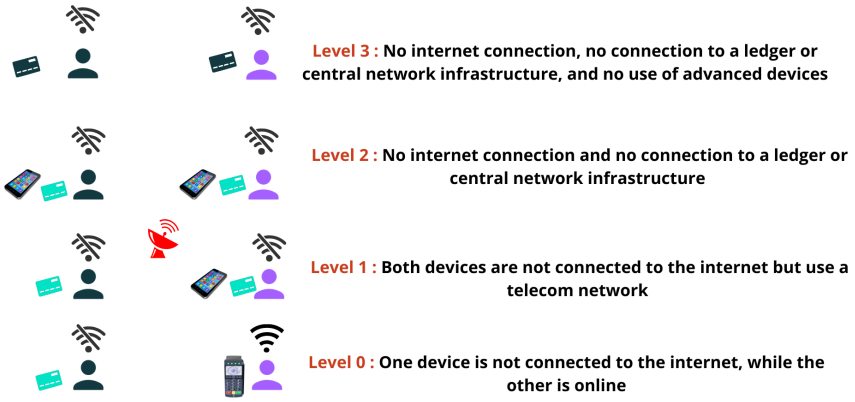


Figure 2.1: Different levels of offline payment.

2.3.1 Offline Payment Concept.

Offline payment generally refers to all transactions conducted without the use of the internet. This mode of payment includes transactions in cash, by check, via mobile, etc. However, when it comes to offline payments for central bank digital currencies (CBDCs), there is no consensus on a single definition. For some, offline payment refers to transactions carried out without internet access and without the possibility of connecting to a ledger that is, without internet and without LAN or WAN network infrastructure (Wi-Fi, Satellite) [1, 36]. Conversely, others consider that it includes transactions without internet, without telecom networks, and without the use of advanced devices like smartphones [37].

Overall, when considering offline payment from the resilience perspective (for instances where internet connectivity is lacking or network infrastructure fails), it is evident that offline refers to transactions conducted without internet connectivity and without relying on any intermediary network infrastructure (telecom or network) for transaction routing and validation. In this context, reference [1] highlights three types of resilience: short-term resilience, civil contingency resilience, and ongoing resilience. Furthermore, reference [38] argues that consensus is unnecessary and somewhat trivial because the choice of an offline solution depends on the specific choices and goals of each system (selection of actors in offline mode, type of outages...).

2.3.2 Offline Payment Level

In light of the various definitions of offline payments, I propose to theorize four levels of offline payment. So, I distinguish four connectivity levels from "partial offline" (one party online) to "full offline" (no internet, no telecom, no advanced device). Level 0 corresponds to partial offline mode, where one party (sender/receiver) has no internet connection, while the other party is online. Level 1 involves transactions without internet connection but with the use of a telecom network (e.g., GSM) [5]. Level 2 encompasses transactions without internet, without telecom networks ,and without connection to a ledger[39]. Finally, Level 3 refers to transactions without an internet connection, without telecom network, and without the use of advanced devices. This framework is based on the principle that offline payment should ensure broad accessibility for low-income populations, who may not always have the means to acquire advanced technological devices due to their high costs. Fig. 2.1 provides an overview of these four levels.

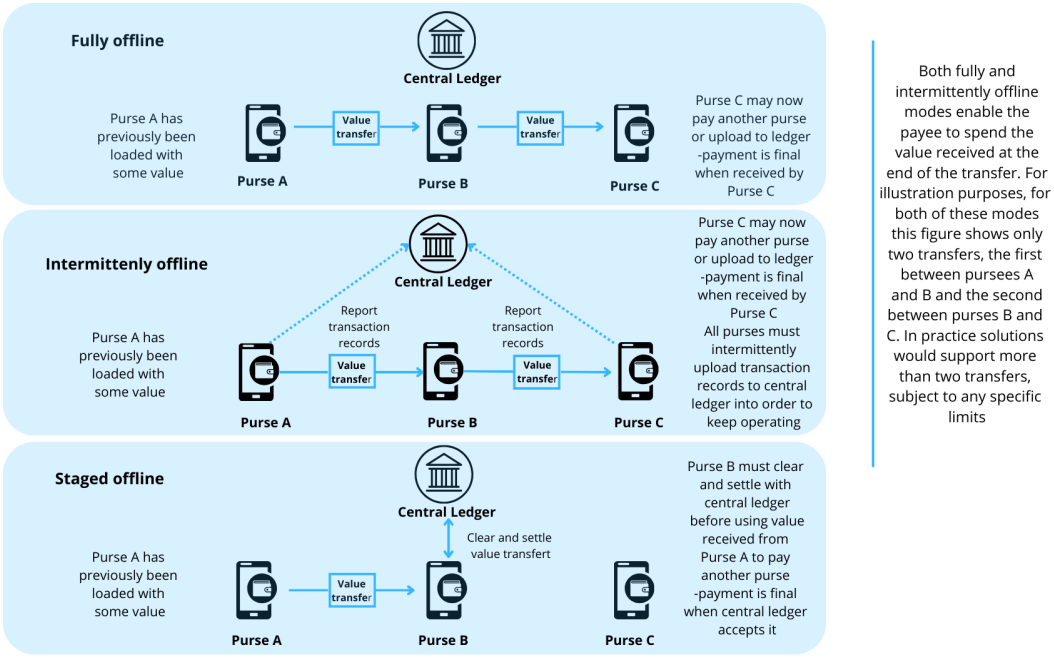


Figure 2.2: Different modes of offline (Adapted from: [1])

2.3.3 Offline Payment Modes

The offline payment function is also a desirable feature in CBDC systems to ensure resilience in the absence of an internet or network connection and to provide an experience similar to cash payments. As noted by [1], in the case of CBDC, various types of offline payments can be distinguished based on whether they connect to a ledger (offline or online):

1. Fully offline payments. These payments do not require synchronization with a ledger for recording and reconciliation operations. These transactions are immediate and are often associated with hardware wallets.

2. Intermittently offline payments. These transactions involve transferability properties and require delayed synchronization when an online connection becomes available. The solution proposed in this paper falls into this category.

3. Staged offline payments. These payments require a connection to the ledger to validate the offline transaction, and therefore, they lack a transferability property. Fig. 2.2 depicts the different modes of offline in the CBDC context.

2.4 CBDC Based Offline Feature

2.4.1 CBDC Architecture

2.4.1.1 CBDC Types

Generally, two types of models are distinguished based on their usage perspectives. The first type is retail CBDC (rCBDC). rCBDC is designed for public transactions. The other type of CBDC is wholesale CBDC (wCBDC), which involves tokenized transactions exclusively between financial institutions, such as central and commercial banks. However, there are modular systems based on a two-tiered CBDC architecture that can support both retail and wholesale transactions [40].

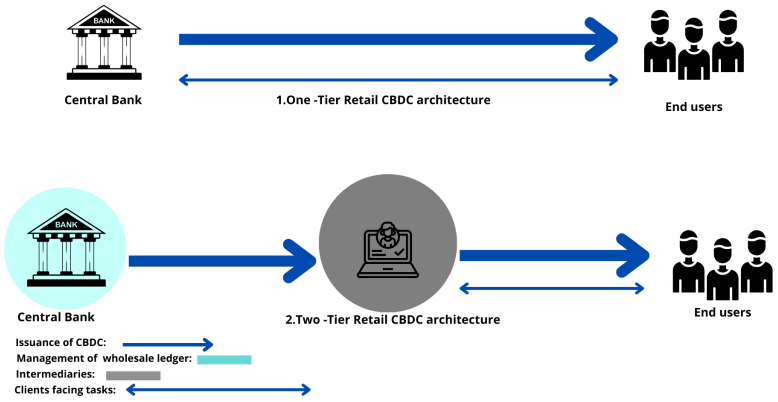


Figure 2.3: One-tier and two-tier retail CBDC architecture.

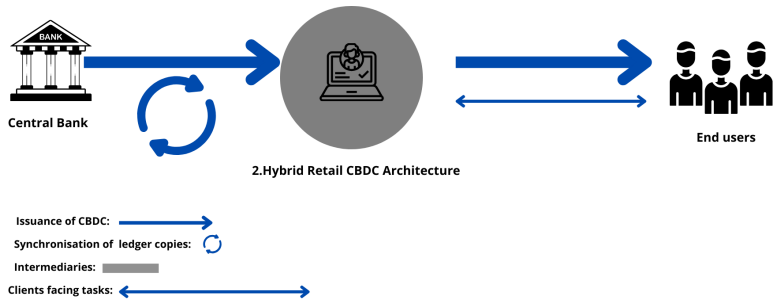


Figure 2.4: Hybrid retail CBDC architecture

2.4.1.2 CBDC Access Models

Retail CBDC can be implemented in three architectures, first described in [41]:

- Direct CBDC. In this model, CBDC funds are managed "directly" by the central bank (balance records, transaction updates) and made available to clients without intermediaries such as payment service providers or commercial banks, although intermediaries may be involved for KYC operations. An example of a direct CBDC is the e-naira project by the Central Bank of Nigeria.

- Intermediated CBDC. This model, widely explored in academic research and by payment industry professionals, limits the role of the central bank to issuing CBDCs, while commercial banks manage fund distribution and client identification and authentication. The E-krona and ECB projects are examples of CBDC projects following an intermediated model. Fig. 2.3 illustrates one-tier and two-tier retail CBDC architecture.

- Hybrid CBDC. In this model, intermediaries handle client onboarding and retail transactions, but the central bank periodically synchronizes retail balances as a backup. Fig. 2.4 provides an overview of this architecture.

Fig. 2.5 illustrates a comprehensive layered view of offline digital currency systems, mapping each taxonomy axis ledger architecture, noticeably service management, logical protocols, application gateway, and physical interface across centralised, distributed, and hybrid models. This visual synthesis highlights the functional responsibilities and cryptographic requirements at each layer of the stack.

2.4.2 CBDC System Management

CBDCs can also be classified based on their ledger management type, which can be either centralized, referred to as a single-ledger-based system, or distributed, known as



Figure 2.5: Comprehensive layered architecture of offline digital currency protocols across centralised, distributed, and hybrid infrastructures.

a distributed-ledger-based system. In both scenarios, the ledger (or digital ledger) will act as the central bank's technological tool to issue digital currency and preserve the transaction record.

2.4.2.1 Single Ledger Based System

2.4.2.1.1 Traditional Infrastructure

Traditional infrastructure refers to the use of conventional databases (data base with account balances) by much of the financial sector to facilitate user transactions. Clearly, a single entity administers the database, and only authorized users can perform read and write operations. Consequently, client trust is not based on the system itself but on the central authority managing it. Additionally, the system's integrity is ensured by internal mechanisms. Moreover, today there are already highly advanced traditional databases, so-called 'WORM databases' (Write Once, Read Many), which allow transaction data to be recorded but not deleted (e.g., Oracle-based database systems).

2.4.2.1.2 Traditional Infrastructure Applied to CBDC

In this configuration, a central manager oversees both the transactions and the associated transaction information. There are two possible types of operations, depending on the role of the central bank. When the central bank manages both the accounts and customer services, this is referred to as a direct operation, as seen with the eNaira in Nigeria. Conversely, when the central bank manages the accounts but does not handle customer services, this is known as an indirect operation, exemplified by the e-CNY in China.

2.4.2.2 Distributed Ledger Based System

2.4.2.2.1 Distributed Ledger and Blockchain concept

Unlike a single ledger, distributed ledgers distribute, replicate, and synchronize transaction records across multiple instances. Blockchain is a typical example of a distributed ledger. Thus, blockchain is a distributed system for writing, reading, and storing data,

characterized by desirable features of decentralization, autonomy, verification, immutability, traceability, and efficiency. It operates cryptographic protocols and consensus mechanisms, which can be governed and optimized by a smart contract [42].

As a distributed system, blockchain can be categorized into permissioned blockchains and permissionless blockchains. The key difference between them lies in the network's access rules. Permissioned blockchains restrict access to authorized participants only, while permissionless blockchains are open to everyone without restrictions.

Cryptocurrencies are a use case of blockchain, which also has other applications such as supply chain, insurance, healthcare, IoT, identity management, agricultural sector, and more. Despite the attractiveness of blockchain, it must overcome multiple challenges such as security and privacy. Hence, there are various frameworks in the literature aimed at making blockchain secure and private. Many of them utilize privacy-enhancing technologies.

2.4.2.2.2 Distributed ledger Applied to CBDC

In a distributed system applied to permissioned CBDCs, only approved authorities can verify transactions and manage the ledger. In this configuration, two operational models exist for central bank digital currencies, distinguished by the central bank's level of involvement. In the first model, known as a direct operation, the central bank oversees both account management and customer services. In the second model, termed an indirect operation, the central bank manages only the accounts, delegating customer services to intermediaries, as illustrated by Bakong in Cambodia.

2.4.2.3 Hybrid Model Based System

A distributed ledger system and single ledger can also coexist in a unified system without being conflated. Such an approach is often referred to as a hybrid model. Table 2.1 presents the conditions for defining a hybrid model.

2.4.3 Offline Transfer Value

2.4.3.1 Offline Payment Scenario

The scenarios considered for CBDC payments between payers and payees are varied and depend on the type of devices and interface used. Table 2.2 provides a non-exhaustive list of potential scenarios.

2.4.3.2 Communication Technology

A communication channel is required to transfer value between devices. In offline payment modes, multiple technologies can be used, including Near Field Communication (NFC), Bluetooth Low Energy (BLE), Wi-Fi, Quick Response (QR) codes, and others. Each of these solutions has its own advantages and disadvantages in terms of security, speed, data capacity, user experience, range, cost, privacy, and energy consumption.

2.4.3.3 Transaction Authentication method

There are two authentication methods required before executing transactions. The first, known as the account-based method, processes the transaction if the executor's identity as the account holder can be verified by a third party (such as a device or code) and if they

Table 2.1: Advantages of Hybrid CBDC System.

Key Features	Cash Money	Retail CBDC System		
		CBDC Based Token	CBDC Based Account	Hybrid CBDC
Usability Features				
24/7 Availability	✓	✓	●	✓
Peer-to-peer Transfers	✓	✓	✓	✓
Immediate Finality	✓	✓	✗	✓
Privacy and Control				
Pseudonymity	✓	●	●	✓
Contextual Privacy	✗	✗	✓	✓
Limits/Caps	✗	✗	✓	✓
Security/ Fraud Protection	✗	✓	✓	✓
Financial Features				
Programmable money	✗	●	●	✓
Interest-bearing	✗	✗	✓	✓
AML Compliance	✗	✓	✓	✓
Targeted disbursements	✗	✓	✓	✓

✓ Existing or Possible Feature
● Possible Feature
✗ Feature Not Possible or not Required

Table 2.2: Offline payment scenarios.

Offline Settlement Scenarii	Payer device	Payee device	Dedicated interface
Payer pays Payee by scanning a QR code on his smart device.	Smart phone 	Smart phone 	QR Code 
Payer pays Payee when both are connected through Bluetooth.	Smart phone 	Smart phone 	Bluetooth 
Payer pays Payee when both are connected through NFC.	Smart phone 	Smart phone 	NFC 
Payer pays Payee by scanning a QR code on his feature device	Feature Phone 	Feature Phone 	QR Code 
Payer pays Payee by typing his card to his smartphone	Card 	Smartphone 	NFC 
Payer pays Payee by typing his devices(watch,phone, cardetc) to the POS device	Phone//Card/Watch 	POS 	NFC 
Payer pays Payee when they both insert card in bridge device	Card 	Card 	Contact Chip 

have sufficient funds. If these conditions are met, the payer's balance is debited, and the recipient's balance is credited. In contrast, the second method is called the token-based method. This approach does not require a third party, as the transmitted digital asset itself carries the cryptographic proof allowing the payee to verify that the sender is the legitimate owner. Typically, the payer encrypts the asset with their private key, and the payee verifies it using the payer's public key.

It is worth noting that both methods can be integrated into a single system. For instance, the Layer 1 (online) may employ a token-based authentication method, while the Layer 2 (offline) could utilize either a native Layer 1 token method or a non-native Layer 2 token method. Similarly, the Layer 1 (online) might implement an account-based authentication method, with the Layer 2 (offline) supporting either a native Layer 1 account method or a non-native Layer 2 token method.

2.5 Offline Capable Digital Currencies Protocols Taxonomy

2.5.1 Trust Anchor Axis

Offline protocols can be distinguished by the trust anchor that protects private keys and critical state:

2.5.1.0.1 Software-only schemes

Purely cryptographic protocols rely exclusively on software to guarantee security, with no specialised hardware on the user side. Examples include anonymous coin-transfer proposals and group-signature variants [37, 43–47]. They simplify deployment and promote device agnosticism, but expose keys to malware and side-channel exfiltration on commodity phones.

2.5.1.0.2 Hardware-assisted schemes

Other designs embed secrets inside secure elements (SE), mobile Trusted Execution Environments (TEE), physical unclonable functions (PUF), or tamper-resistant modules. Representative implementations involve smart cards, eSIMs or TEE-based mobile wallets [39, 43, 48–52]. Hardware raises the bar against extraction and replay attacks, yet increases bill-of-materials, certification effort and supply-chain dependence.

2.5.1.0.3 Design trade-off

Selecting a trust anchor therefore balances portability and cost (software) against tamper resistance and regulatory confidence (hardware).

2.5.2 Centralised ledger-Based Protocols

2.5.2.0.1 Centralised and two-tier architectures

These offline protocols interface with traditional financial infrastructures, such as centralised banking systems. They include account-based models [39, 49–51, 53, 54] where the issuer updates a unique ledger, as well as coin-based models where transferable tokens represent value [43, 55–58]. Though mature and widely deployed, these systems

often trade flexibility and decentralised auditability for operational stability and regulatory familiarity. Coin-based schemes may also be integrated into distributed or hybrid infrastructures [59, 60].

2.5.2.0.2 Distributed and DLT-compatible infrastructures

This class of protocols uses distributed ledger technologies (DLTs) for operations such as funding, reconciliation, or post-offline settlement. Examples include systems based on UTXO models [61, 62], USO abstractions [63], or one-time programs (OTP) [52, 64] or eCoin [65]. These designs often prioritise privacy where UTXO-based protocols, in particular, enable anonymous peer-to-peer exchanges but introduce challenges for regulation, accountability, and enforceability [45, 48, 66–69].

2.5.3 Cryptographic-Primitive Axis

Offline-capable payment schemes differ sharply in the cryptographic building blocks they employ. We distinguish two strata :

2.5.3.0.1 Classical primitives

Early offline designs rely on conventional signatures (RSA/ECDSA), blind signatures or simple hash/MAC counters. They run fast on smart cards and rest on well-understood assumptions, yet offer only coarse privacy and remain vulnerable to quantum attacks.

2.5.3.0.2 Emerging primitives

Recent proposals integrate homomorphic encryption, ring signatures, lattice-based signatures, and zero-knowledge proofs (zk-SNARKs, zk-STARKs) to deliver unlinkability, selective disclosure and post-quantum resilience [53, 65, 70]. These techniques are still R&D-heavy; proof sizes, runtime overhead and library maturity are the main adoption barriers.

In practice, most pilot systems adopt a hybrid approach: classical signatures for core authentication combined with a lightweight ZKP or ring-signature overlay for privacy. Selecting a primitive therefore balances implementation simplicity against forward-looking security and regulatory demands.

2.5.4 Target-Property Axis

This axis groups the design goals that offline schemes explicitly try to achieve, beyond the infrastructure and the trust anchor. We focus on three clusters of properties that recur throughout the analysed protocols .

Most real-world prototypes optimise for one cluster at the expense of another: EMV-style cards favour resilience [71], DLT-token models maximise privacy [63], while hybrid ZKP schemes push scalability through batch validation [65]. Mapping each protocol onto these clusters highlights critical tradeoffs: no current design simultaneously achieves amount-hiding privacy, strong DoS tolerance and sub-linear reconciliation overheads. Table 2.3 showcases the target properties of offline-capable protocols while Fig 2.6 represents the design space of offline-capable digital currency protocol.

Table 2.3: Target properties of offline-capable protocols

Property cluster	Sub-dimensions	Offline relevance and rationale	Typical mechanisms
Privacy	Identity anonymity, amount confidentiality, metadata unlinkability	Mimicking cash requires hiding payer/payee links even when connectivity is lost; regulators may still require conditional auditability	Ring signatures [47], blinding [56], zk-SNARK/STARK [53], token shuffling.
Resilience	Energy autonomy, DoS resistance, post-outage recovery	Devices may operate without recharging or Internet for hours or days; protocols must remain responsive and secure	On-device counters [49], replay protections [51], deferred sync [65]
Scalability	Transaction throughput during burst, reconciliation state size, staged workflows	Large events (e.g., festivals, disasters) cause transaction surges that must not break the system	Hierarchical counters [50], batch proofs [48], staged offline design [45]

2.6 Conclusion

All in all, this chapter has clarified the technological, organisational and conceptual foundations of offline payment in digital-currency systems. In particular, by analysing the various types of digital money, the architectural models, the authentication mechanisms, the transaction-management approaches, and the protocol taxonomies proposed in the literature as well as the constraints that shape their design this chapter has highlighted the complexity of the offline ecosystem and the diversity of viable options. Consequently, several parameters must be taken into account when designing offline payment systems for future CBDCs: technical feasibility; distinctions between levels of offline capacity; value-storage modalities; access models; communication infrastructures; and cryptography technologies. Subsequently, the examination of the elements synthesized in this chapter forms the conceptual foundation upon which the theoretical and methodological developments presented in the following chapter rest. After analysing the foundations of offline payment, the next chapter will delve deeply into the analysis of CBDC offline requirements and existing systems.

2.7 Chapter Summary

This chapter has presented a systematic study of the essential elements that structure the offline function within digital currency systems. Initially, it examines the design principles of digital currencies in a disconnected context. Following this review, it describes cardinal notions and mechanisms of offline operation such as capacity levels and different operational modes. The analysis is enriched by an exploration of CBDC architectures, access models, communication infrastructures, and authentication mechanisms associated with offline transactions. Finally, a taxonomy of offline payment protocols identifies the main

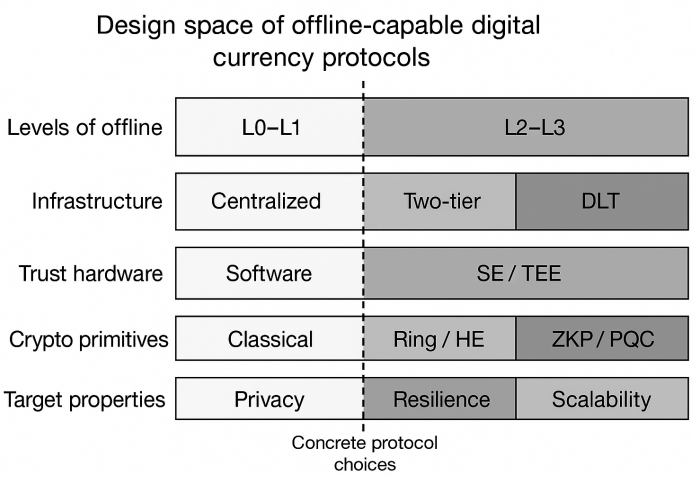


Figure 2.6: Design space of offline-capable digital currency protocols, structured around five key axes: levels of offline, infrastructure, trust hardware, cryptographic primitives, and target properties.

existing approaches and their respective contributions. This chapter serves as a preface that situates the fundamental aspects related to the CBDC offline payment ecosystem.

Chapter 3

Analysis of CBDC Offline Requirements and Existing Systems

3.1 Introduction

The previous chapter identified the general principles and technological foundations inherent to the ecosystem of digital currencies in a disconnected environment. However, beyond conceptual approaches and existing technological considerations, it is essential to examine precisely the requirements that emerge from the specific constraints of mobile environments, monetary authorities, financial infrastructures, and user needs. This bipolar analysis cannot be complete without also taking into account the systems and prototypes proposed by central banks and industry actors.

Thus, the objective of this chapter is to specify the functional, security, operational and regulatory requirements that an offline CBDC payment system must satisfy. These specifications derive from a formalisation of a comprehensive adversary model, covering contemporary attack vectors applicable to offline payments as well as informational threats. This approach illuminates monetary authorities' expectations in terms of security and clarifies the implications for confidentiality, inclusion and user trust.

The chapter is structured around the identification of threats affecting offline payment systems, the requirements that address these threats, and lessons learned from practical pilots. This methodology provides a specification framework upon which the solutions developed later in the manuscript will rest. The remainder of this chapter is organized as follows: Section 3.2 is about Background and Related work; Section 3.3 presents Adversary Model for Dual Offline CBDC; Section 3.4 introduces framework requirements; Section 3.5 presents Existing Offline-Capable CBDC Solutions while Section 3.6 and Section 3.7presents respectively Conclusion and Chapter Summary.

3.2 Background and Related work

The 2008 financial crisis and the rise of Bitcoin ignited a new era of digital currencies. A growing number of jurisdictions are exploring central bank digital currencies; as of June 2025, 39 are running pilots, most of which target financial inclusion [72]. A key design requirement emerging from these pilots is offline capability. This feature requires that transactions can be completed even when Internet connectivity is unavailable, thereby reproducing the resilience and cash-like privacy of physical banknotes. Providing such resilience without compromising security in particular double-spend protection, privacy,

Table 3.1: Comparison Table of Literature Survey

Work	Comparative Study		Privacy	Security	Scalability	Offline Analysis	Use Case	New Architectures
	Qualitative Analysis	Quantitative Analysis						
[73]	Yes	No	Yes	Yes	No	No	No	No
[1]	Yes	Yes	Partial	Yes	No	No	No	No
[77]	Yes	No	Yes	Yes	No	No	Yes	No
[78]	No	No	Yes	Yes	No	No	No	No
[74–76]	No	No	Yes	Yes	No	No	Yes	No
My thesis	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

or energy efficiency remains an open engineering and research challenge. In this chapter, we aim to analyze and highlight the requirements for deploying a viable offline CBDC payment solution.

Several studies have been conducted on the offline function within digital currencies. Some surveys focus on CBDCs. On August 31, 2021, [73] conducted an interesting survey on CBDCs, analyzing security and privacy aspects to address gaps and concerns inherent in the deployment of this digital currency. However, in addition to being limited solely to CBDCs, this study focuses exclusively on blockchain architecture and does not address the offline function, which can also integrate with and complement blockchain infrastructures.

In May 2023, [1] thoroughly examined the topic of offline payments in CBDCs across all aspects security, privacy, risks, and risk management. This study serves as a valuable tool for solution architects, financial institutions, and engineers looking to implement payment solutions. However, it is a generic study that does not delve into the specificities, challenges, and weaknesses of each type of offline system within its integration with a given architecture. Furthermore, new offline architectures, such as mesh and satellite networks, are not highlighted. The authors of the trilogy [74–76] not only address security and privacy aspects but also cover scalability aspects of offline payment. However, they do not mention practical case metrics that could illustrate and validate the documented design choices.

On April 28, 2022, [77] discussed the security requirements of the offline function according to the types of currencies, with particular attention to CBDCs, highlighting the advantages, limitations, and improvement plans of current proposals. However, this study does not specifically and deeply address privacy, which is only mentioned in terms of anonymity as a security property alongside unforgeability, non-repudiation, verifiability, etc. Additionally, the performance aspect of offline solutions is not addressed, and post-quantum security concerns are completely absent.

There are also some papers about offline payment in digital payment systems. On August 16, 2024, [78] presented a practical case study on offline and hybrid offline payment, addressing resilience and reliability with a focus on security and privacy characteristics. This study immerses readers in the infrastructures and devices supporting offline payments. However, it does not offer a comparative and critical analysis that could serve as an essential reference implementation tool for governments. Unlike prior work, this thesis introduces a multidimensional analysis, combining qualitative and quantitative comparisons, privacy and security evaluations, offline capability assessments, practical use cases, and innovative architectural designs. Table 3.1 summarizes these contributions, demonstrating the uniqueness and completeness of this research.

3.3 Adversary Model for Dual Offline CBDC

Security and privacy are serious challenges in CBDC systems in general, and particularly in offline payment solutions. It involves not only detecting cyber threats but also mitigating risks associated with vulnerabilities in the ecosystem. Notably by taking proactive and corrective measures. In this section, we analyze in general terms, independently of each system's specificity, the risks and threats that may arise during a transaction between a payer and a payee.

3.3.1 The Double Spending Problem

Double spending fraud involves using the same digital monetary unit more than once to fraudulently obtain financial gain. Online, users can protect themselves from double spending attacks by verifying spent units in the ledger database to confirm that a transaction has not already been executed. However, offline, such verification is not possible. Double spending thus represents a proven security risk for offline payments with immediate finality. Moroever, Double incoming fraud is a version of double spending in which Alice reuses the same digital monetary unit multiple times to credit her own account. Typically, this involves an online account that is credited through a deposit operation using a transaction that has already been used for an offline transaction.

3.3.2 Man-in-the-Middle Attacks

It involves an attacker secretly attempting to read, intercept, or tamper with a confidential transaction between two users without their knowledge. Typically, the attacker uses specific tools capable of sniffing transaction frames. This passive attack allows the adversary to cross-reference information and build a database of critical elements, which often serves as a prelude to active attacks. In such active attacks, the attacker no longer merely observes but directly interferes with the system.

3.3.3 Replay Attacks

In computer security in general, a replay attack aims to fraudulently replay a message from a different context into the intended (original and expected) context, with the goal of deceiving honest participants into believing they have successfully completed the message transmission[79]. In offline transaction, a replay attack involves reusing a transaction that has already been accepted by one recipient and fraudulently replays it with another recipient. This type of attack can follow a Man-in-the-Middle attack.

3.3.4 Forgery Attack

This type of attack involves fraudulently forging digital currency outside the official mechanisms provided by the system and using it to execute a validated transaction with a recipient. In many cases, the attacker gains unauthorized access to sensitive cryptographic keys or even a master key. Additionally, counterfeiting a transaction can be achieved through cryptanalysis. This involves analyzing the cryptographic protocol securing the transactions, potentially enabling the attacker to reverse-engineer the cryptographic keys protecting the transaction.

3.3.5 Side Channel Attacks

This attack involves the adversary exploiting information unintentionally leaked by the device during its operation to infer sensitive data. This could include secret keys, internal manipulated data, and more. Outside the device, the attacker can recover this information by analyzing power consumption, operation execution time, electromagnetic emissions, thermal heat, or acoustic noise.

3.3.6 Physical Tampering and fault-induced attacks

These attacks encompass all methods involving the direct manipulation of hardware or its execution environment. The goal is to compromise the internal operation of the wallet. For instance, an attacker may attempt to modify data (T.Phys-manipulation). Additionally, they can inject faults by subjecting the device to electrical, optical, or electromagnetic stress, this is known as fault injection. The attacker may also exploit functional vulnerabilities (T.Abuse-Func), attempt to clone the device, or induce deficiencies in random number generation (T.RND), which is critical for tasks such as key generation.

3.3.7 Data Breaches

This risk involves the leakage of users' personal and transactional data. Such leaks can be caused by a third-party device compromise(security vulnerability), human negligence, intentional human action (rooting, jailbreaking smartphone etc), or a cyberattack. Another typical case is unauthorized access to a user's personal information or offline transaction history in the event of device loss.

3.3.8 Profiling threat

The risk of profiling is related to the fact that a third-party entity analyzes how consumers spend their money. To this end, transaction data and offline payment metadata are analyzed and compiled to create a consumption profile (including spending habits and preferences). This type of process is often used on online platforms for targeted advertising.

3.3.9 Data monetization

User data represents a valuable resource for various digital platforms and financial institutions due to the insights they can extract to achieve their objectives. This wealth of information can give rise to a new type of risk: data monetization abuse. The latter involves exploiting or transforming users' personal and transactional data without their consent for economic profit. This practice often follows user profiling, particularly when the goal is not simply to sell raw data.

3.3.10 Dataveillance

This risk involves mass surveillance of users (without their consent) based on their transactional profiles. This violation of consumer consent inherently requires data management centers, such as cloud platforms, to monitor and exploit all of this data.

3.3.11 Other Risks

There are other risks that can weaken the offline payment environment. These include device obsolescence, lost value due to torn transactions (where funds have exited the payer's wallet but have not been credited to the payee's wallet), and supply chain compromise.

3.4 Frameworks Requirements

3.4.1 Security Properties

3.4.1.1 No Double Spending

The aforementioned attacks highlight several essential properties needed to secure offline transactions. The first is protection against double-spending, which requires a legitimate authority to authenticate both sender and receiver, track previous transactions for the receiver, and link each transaction to its originator.

3.4.1.2 Unforgeability

The second property is unforgeability, based on the requirement of proof-of-ownership of the asset. This property implies verifying the asset's authenticity and ensuring proof-of-ownership. It also involves preventing any adversarial party from forging the wallet. Consequently, protection against double-spending and the requirement for unforgeability can be grouped under the property of traceability.

3.4.1.3 Undeniability

Transaction undeniability is another critical security property, ensuring that, on one hand, Alice is the transaction's author, and on the other hand, that the receiver cannot deny having received the transaction or claim it is fake [14].

3.4.1.4 Verifiability

It consists, on the one hand, in ensuring that the payer is who they claim to be the legitimate owner of the funds and, as such, has the right to transfer funds they have access to. This aspect of verifiability is called authentication. On the other hand, it involves verifying the validity of transaction details (such as the recipient's identity, amount, etc.) and their compliance with the system's rules. This aspect of verifiability is called authorization.

3.4.1.5 Integrity

The integrity of a payment system integrating offline payments involves protecting both online and offline transaction flows, as well as the process of money creation, from any form of tampering. In other words, all operations related to the creation, transfer, and deletion of money must be carried out in accordance with the rules governing the system's operation.

3.4.2 Privacy Feature

3.4.2.1 Privacy Concept

Privacy is an essential element in the digital economy, which requires the use of personal and individualized data of individuals. Thus, privacy acts as a barrier against the misuse of this data, offering protection against super control and manipulation of power, while safeguarding human dignity, safety, and self-determination for citizens [18].

When discussing privacy, several terms are often mistakenly used interchangeably, leading to confusion regarding the intended goals. Therefore, we aim to clearly define each term, specifying the appropriate meaning of each terminology. The concept of anonymity means that there is no information on the identity of the payer and/or payee, whereas pseudonymity denotes that the wallet account is known, but the owner is not revealed. In this case, data may or may not exist. The term privacy, in contrast, means that data on the individual exists but is kept secret and accessible only to designated individuals. The notion of managed privacy means that the central bank offers different privacy options (e.g., eCNY), while contextual privacy (adaptive privacy) refers to the fact that, depending on the settings in the cryptographic system's terms and conditions, the level of privacy may vary. Finally, confidentiality is transaction data exists but is only accessible on a need-to-know basis.

Although it is a key feature, privacy has long been ignored or superficially treated in research, as noted by the author of [80]. However, recent trends in research on the subject, while still a minority demonstrate growing awareness[18, 75, 81]. The unavoidable questions related to the CBDC privacy system are as follows: what information needs to be protected? Personal information? Transaction data? Who should it be protected from financial institutions? Government? Network provider? What data protection mode should apply Anonymous? Pseudonymous? Confidential? Contextual privacy? What level of protection? High? Medium? Low? Or should data minimization be considered? Providing a technical response to these concerns is essential to avoid misuses. In any case, the envisaged choices should align with privacy principles and be compatible with AML controls to safeguard individuals' financial freedom. Reconciling privacy and offline payment thus becomes a real challenge.

3.4.2.2 Types and Levels of Privacy

There are multiple types and levels of privacy, which vary by jurisdiction. We can distinguish three primary levels: strong/high/hard privacy, baseline/medium privacy, and low/soft privacy. However, if the terms privacy and confidentiality are interchangeable, we can have limited confidentiality, controllable confidentiality, and selective confidentiality. In truth, privacy levels are considered in relation to their degree of anonymity. This is why some prefer qualifications like full anonymity/full private, semi-private, and transparency. There is also a classification based on the type of cryptographic method involved: complete privacy, limited privacy, and controllable privacy. Alternatively, privacy can be classified by the type of data to protect: identity privacy, confidential transaction, data privacy, or based on the nature of synchronization: balance reconciliation and transactional reconciliation. All things considered, these concepts enrich the terminology surrounding privacy, bringing important nuances. However, it is crucial not to confuse transparency (pseudo-anonymity), anonymity, and confidentiality, which, when considered separately, are distinct concepts and therefore not interchangeable. Finally, all these levels of privacy can be applied according to the nature of the stakeholders in the CBDC

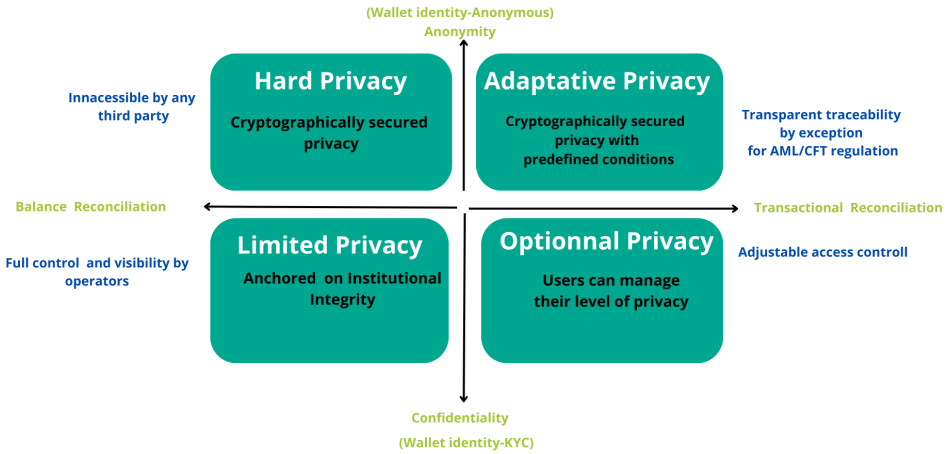


Figure 3.1: Different types of privacy in offline payment.

ecosystem (central bank, commercial bank, users, other PSPs, etc.). Fig. 3.1 provides a comprehensive view of the different privacy cases in the context of offline payments.

3.4.2.3 Unlinkability

For a given offline payment transaction, the identity of the payer or payee must not be linked to each other. Full unlinkability includes both user unlinkability and transaction unlinkability. In user unlinkability, the real-world identities of the payer and recipient must remain dissociated from their other transactions. Additionally, for transaction unlinkability, it must be impossible to link a transaction to previous transactions that funded the used funds.[82].

3.4.3 Auditability

This property ensures that an entity, such as auditors, can oversee user transactions and verify their compliance with applicable rules.

3.4.3.1 AML/CFT Regulation Compliance

Regulatory compliance is subject to the jurisdiction of each state. As a result, it can vary from one jurisdiction to another. However, a fundamental and common aspect across all jurisdictions is the fight against money laundering (AML) and the financing of terrorism (CFT). A major challenge is therefore ensuring that an operation that is technically legal is also legally compliant, and conversely, that an operation that is technically illegal is also legally illegal.[83]

3.4.3.2 Active and passive Audit

An audit is considered active when the auditor is able to inspect an offline transaction before its validation. Conversely, an audit is considered passive when this inspection occurs after validation or without direct intervention.

3.4.3.3 Transparent and confidential Audit

This audit mode involves auditors verifying the compliance of a transaction with predefined audit rules by analyzing transaction information transmitted in plaintext by the

transaction initiator. Unlike active audits, passive audits assume that transaction information is encrypted, and consequently, only authorized auditors are capable of decrypting and verifying it.

3.4.3.4 Traceability

Traceability refers to how offline transaction information is recorded in the ledger. There is three modes of traceability: full transaction records, partial transaction records and no transaction records. In full transaction records mode the entire transaction history is recorded in the central bank's ledger, including participant identities, amounts, balances, and metadata). Under partial transaction records configuration, the central bank has restricted access to transactional information. In no transaction records configuration, no offline transaction data is synchronized to the ledger.

3.4.4 Performance

3.4.4.1 Baseline metrics

The `Speed` metric measures the time required for a transaction to be validated, executed, and confirmed to the transaction sender. It is measured in seconds. Moreover, the `throughput` metric measures the number of transactions an apparatus (or system) can process per unit of time. In an offline environment, `latency` refers to the local processing time between the local submission of a transaction and its final validation (e.g., verification of a cryptographic proof).

3.4.4.2 Scalability

Commonly, the motivations behind offline payments often highlight resilience and cash-like properties, especially the desired privacy. However, scalability is also a valid reason to promote offline payments. The offline feature can significantly alleviate overloaded servers and databases processing online transactions [74]. It is viewed as a tool providing an alternative processing route capable of handling transactions outside the system's ledger. In the case of CBDCs, scalability depends on the offline entity (Payer, intermediary, or back-end infrastructure) and may also depend on the offline mode choice (fully offline, intermittently offline, staged offline).

3.4.4.3 Other Metrics

Additional metrics may apply depending on the implementation context, including: message size, energy consumption, computational load, etc.

3.4.5 Extended Features

3.4.5.1 Interoperability

Interoperability among payment systems is a crucial characteristic, as it enables both cross-border payments and the coexistence of multiple payment methods for transactions involving different digital currencies. Technically, the lack of standards significantly impedes the interfacing of various payment methods, both offline and online/offline. A digital wallet can be used for various purposes beyond payments, thus sparing users from managing multiple wallets (e.g., Digital ID cards, cryptocurrency, credit/debit cards,

travel/loyalty cards) [84]. Moreover, an open-source solution can be developed quickly, at a low cost, and can foster interoperability and safety. Transparency and openness, inherent characteristics of open source, help create a trusted ecosystem [61, 85, 86].

3.4.5.2 Programmability

Offline programmability is a first step towards offline smart contracts, as it will allow funds (e.g., financial aid in emergency situations) to become immediately available or restrictions to be placed for certain user categories. In this way, smart contracts can manage funds as they move offline. This property makes offline payments not only cash-equivalent but even superior to cash. This is why certain institutions and central banks are considering its deployment for specific use cases, such as business travel. For instance, employees could receive pre-programmed corporate funds for a limited period or in specific geographic areas [87].

3.5 Existing Offline-Capable CBDC Solutions

3.5.1 Digital Euro

The Digital Euro is the Eurosystem's digital currency project. Security and privacy are central to this project [16, 88]. Offline payment plays a significant role in this project [89].

Offline Wallet. A software-based solution has been recommended to operate in an offline wallet. The main functions associated with this component are wallet lifecycle management, limits on holdings and transactions, and offline payment processing.

Offline Distribution. The offline distribution component is the intermediary entity that orchestrates funding, defunding, and synchronization operations for users' devices. The primary functionality of this component is online reconciliation.

Offline Issuance. The European Central Bank seeks to maintain total control over the amount of euros used offline. The primary function highlighted here is liquidity management through the monitoring of transaction operations (issuance, funding, redemption, and defunding).

3.5.2 E-Cedi

The Bank of Ghana (BoG) leads the e-Cedi project, a digital currency offered to consumers in this state to promote financial inclusion and an efficient payment ecosystem [90]. The proposed architecture for the pilot phase follows a two-tier model with a token-based payment rail. For this purpose, several technologies, such as compliant phone apps, smart devices, and smart cards, enable transactions. Regarding system security and privacy, particularly for offline payments, the BoG has opted for technology provided by Giesecke+Devrient (G+D Filia) [91]. In terms of privacy, the architectural design ensures a level of privacy associated with an architectural layer. Thus, the central bank only has an aggregate view of transactions, and only wallet issuers can monitor the activities of individual wallet holders. Additionally, the use of PINs (Personal Identification Numbers) combined with various channels such as biometric authentication enhances user security.

Table 3.2: Comparative Overview of Selected Digital Currency Systems

Digital currency system	Kind	Asset model	Privacy approach	Security protocol	Offline mode	Trusted environment
Digital Euro (Tiered model) [89]	Experiment	Token / Account	SSI + Data minimization	Not specified	Intermittently	Hardware-based / Software-based
eCedi [90]	Pilot	Token	Confidentiality	Native layer 1 Token	Intermittently	Hardware-based
E-Krona [95]	Pilot	Account; Token / Synthetic	Confidentiality	Non-native layer 2 Account	Staged	Hardware-based / Software-based
E-CNY [93]	Prototype	Account-based	Anonymity Management	Native layer 1 Account	Intermittently	Hardware-based / Software-based

3.5.3 E-CNY

E-CNY, or digital yuan, is China's digital currency developed and launched by the People's Bank of China (PBOC). Consequently, the digital yuan is being tested within an open infrastructure based on a conventional approach (single ledger) and an indirect operations management model (financial intermediary institutions), called the Digital Currency Electronic Payment system (DCEP) [92]. In terms of security features, DCEP has opted for multi-level security, such as a digital certificate system, digital signature, and encrypted storage [93]. However, in terms of privacy features, DCEP has adopted anonymity management based on transaction amounts. In fact, the digital yuan application in China allows offline payments through the "tap to pay" feature, even when the phone is off or out of battery, using NFC technology. In case of loss or theft, users can deactivate the feature via another device [94].

3.5.4 E-Krona

As of the writing of this thesis, Sweden's e-Krona pilot project is in phase 4, which is also its final phase [95]. This phase particularly focuses on integrating secure offline payments into the e-Krona pilot ecosystem. The e-Krona pilot has been deployed in a DLT-based system solution designed on the Corda platform. The Riksbank has prioritized a solution combining payment cards and mobile apps. Moreover, this phase introduces the "shadow wallet," a new component in addition to the two existing ones. The shadow wallet manages users' offline transactions, particularly for the load/unload of e-Krona designated for online transactions and for synchronizing all transactions carried out in offline mode. Regarding privacy, the e-Krona relies on traditional KYC, with the privacy feature diluted by traceability requirements [96]. Table 3.2 presents comparative study of digital currencies based payment rail.

3.6 Conclusion

All in all, the analysis presented in this chapter has clarified the fundamental requirements that CBDC payment systems operating offline must satisfy. A careful examination of security, privacy protection, auditability and performance requirements revealed that preventing double spending, authenticating transactions, ensuring integrity, resisting attacks, achieving resilience, verifying correctness, and preserving confidentiality remain indispensable conditions for guaranteeing the viability and widespread adoption of offline payments. Moreover, lessons learned from practical pilots highlight the diversity of approaches. These experiential insights complement the theoretical analysis by revealing unforeseen constraints and recurring limitations in current prototypes. Consequently, there is a need to design integrated solutions capable of simultaneously addressing the technical, security and societal imperatives of offline payment. This observation paves the way for the developments proposed in the following four chapters. Those chapters will present frameworks that aim to coherently and innovatively address the identified gaps

particularly with regard to security, privacy and proactive fraud detection.

3.7 Chapter Summary

This chapter offers a comprehensive analysis of the requirements necessary for CBDC payments in offline mode, focusing on security, privacy, regulatory compliance and performance. By first examining the threats and risks inherent to CBDC payments, it identifies the essential features required for deploying CBDCs in a disconnected environment. The analysis is further enriched by lessons from several practical pilots conducted by central banks and industry actors. These experiments provide concrete insight into the challenges encountered under real world conditions, highlighting both the strengths and limitations of current solutions and the structural tensions that persist around offline payment. Thus, this chapter supplies a solid analytical foundation to motivate the contributions developed in the remainder of the manuscript.

Chapter 4

Offline Payment Based on a Single Ledger

4.1 Introduction

The preceding chapter underscored the cardinal requirements for designing resilient CBDC payment systems. It inventoried the threats affecting the system and transactions in offline mode, and specified the security, privacy, auditability, and compliance features required when connectivity is intermittent or unavailable. However, integrating these properties simultaneously into an autonomous payment, devoid of immediate connectivity, demands advanced technical approaches. The present chapter sets out to lay the foundations for a robust conceptual framework that executes secure transactions based on a single ledger.

This chapter presents PrivTEE-Pay, a comprehensive privacy-preserving framework for secure offline CBDC payments. In practice, it combines David Chaum's blind signature scheme (based on ecash [56]), zk-SNARKs, digital certificates, and Trusted Execution Environments (TEEs). The goal is to demonstrate that operational anonymity can coexist with regulatory compliance (AML/CFT), double-spending prevention, and quantum-attack resistance within a traditional architecture. Thus, the chapter adds the indispensable triptych of privacy: privacy by design, privacy by default, and user-centric privacy for a modern CBDC system.

The chapter proceeds as follows. First, it offers an exhaustive review of existing work on privacy in offline payments. Next, it presents the underlying cryptographic primitives. Then, it describes the proposed architecture and fully operationalizes the protocols certification, withdrawal, payment, verification, coin destruction, and deposit. Particular attention is given to mechanisms that guarantee anonymity. Finally, the chapter concludes with a system analysis, performance evaluation, and discussion. The remainder of this chapter is structured as follows: Section 4.2 presents the background and related work; Section 4.3 introduces PrivTEE-Pay; Section 4.4 concludes the chapter; and Section 4.5 summarizes it.

4.2 Background and Related Works

In the wake of the 2008 financial crisis and the ongoing COVID-19 pandemic, the world experienced profound financial shifts. These events contributed to the emergence of cryptocurrencies, including Bitcoin, and accelerated the shift towards digital economies. In this context, the central bank digital currency (CBDC) has been identified as a reliable and

credible form of digital money. However, while the use of CBDCs offers great promises, its implementation is facing a plurality of challenges that are intertwined with each other. Among others, offline payments and privacy, which has the potential to catalyze or hinder the adoption of CBDCs. Privacy is one of the desired attribute of a digital euro among both citizens and professionals and offline payments are a popular characteristic among the general population, particularly among underbanked layers. This is mainly because it offers the possibility of improving financial inclusion, access, resilience, and maintaining a cash-like experience. Thus, reconciling privacy and offline payments is a real challenge.

Research on offline payment functions has attracted significant attention since the emergence of e-cash. This payment method is especially prominent in CBDC systems, offering advantages, such as resilience, cash-like characteristics, inclusion, universal access, and reduced transaction costs. Despite this interest, few protocols or frameworks exist for offline CBDC payments in the current landscape.An offline CBDC payment involves transferring value between devices without a ledger system connection, often in the absence of the Internet or telecom connectivity.

The Offline Payment System (OPS) protocol proposed in [39] utilizes a two-tier certificate infrastructure established by a delegated financial authority. It employs cryptographic algorithms, enabling secure payments from Alice to Bob without Bob needing a security device. However, it may not fully meet the non-forgery and DDoS attack prevention requirements. An extension of the OPS using an ISO/IEC 7816-compatible smart card and NFC interface addresses these issues by setting transaction time limits and using cryptographic keys on the smart card [51]. Another proposal employs One-Time Programm (OTP) for each monetary unit, to ensure confidentiality, anonymity, and programmability [52]. However, vulnerabilities exist during OTP transmission and framework forgery risks. Reference [97] proposes a flexible, programmable two-tier architecture solution that requires a secure TRE (Tamper Resistant Element) to counter hardware attacks, offers various transaction scenarios, and adaptable confidentiality levels. However, integrating the blockchain introduces de-anonymization risks. Another proposition suggests a universal, programmable solution resistant to quantum attacks, utilizing a new architecture, a trusted physical wallet and introducing "cyber coin" and "Mint" concepts [37]. It enables coin freezing in case of suspected criminal activity. An eID-based approach suggests that an electronic identity provider issue an e-wallet in a secure TEE (Trusted Execution Environment) chip [59]. Countersignatures divide transaction control between the payer and recipient, preventing the compromise of one device from affecting the entire transaction. Table 4.1, shows a comparison table of these solutions with our approach.

4.3 PrivTEE-Pay

In this section, I introduce PrivTEE-Pay, an offline transaction framework applied to a conventional retail CBDC architecture. PrivTEE-Pay provides a prototype implementation of the theoretical solution proposed in [43]. In line with that, PrivTEE-Pay is designed to meet the requirements discussed in the previous chapter. Concretely, the approach proposed in this framework seeks to satisfy three objectives: first, it guarantees end-to-end security for transactions; second, it preserves privacy throughout the transaction lifecycle; and finally, at the performance level, it ensures a smooth user experience during offline payments. Specifically, I propose a payment system that highlights several properties:

- Digitally replicate the attributes of cash while enhancing protection against coun-

Table 4.1: Comparative Study of CBDC Offline Function Approaches

CBDC Offline Function	**Wallet Type**	**No double Spending**	**Unforgeability**	**Additional Hardware**	**Privacy Type**	**Risks and Issues**
Cash	no	Yes	Yes	no	Data identity	Counterfeiting
[39]	Software	Yes	no	TEE	Data identity	• DDoS attack, Quantum attack, Physical attack • Limited transaction • Low privacy breach
[51]	Hardware	Yes	Yes	Trusted hardware module	Data identity	• Physical attack • Supply chain attack • Limited transaction • Low privacy breach
[52]	Hardware	Yes	Yes	TEE	Data privacy	• Physical attack • Framework could be hacked
[97]	Hardware & Software	Yes	Yes	TRE	Fully configurable level privacy	• Quantum attack • Re-identification in blockchain • Privacy breach depends on implementation
[37]	Hardware	Yes	Yes	Trusted physical module	Data privacy	• Disintermediation of the banking system • Need to build a new architecture
[59]	Software	Yes	no	TEE	Data transaction	• Forged payment
Our approach	Free software	Yes	Yes	TEE	Data privacy	• TTP (bank) usurpation when the currency is signed

terfeiting and double spending. The advent of CBDCs is not merely an ambition to digitise fiat currency; it aims to deliver a payment system that is substantially superior to cash in terms of security.

- Ensure transaction consistency. Indeed, the sum of coins spent during an offline session must not exceed the amount withdrawn online. Accordingly, offline transaction amounts are capped and an online synchronisation delay with the commercial bank becomes operational in all users' wallets.

- Strengthen transaction privacy by employing privacy-enhancing technologies such

as zk-SNARKs and blind signatures. In this way, offline transactions occur without revealing the identities of the parties (payer and payee) or the transaction amounts to the central bank. However, the commercial bank or any intermediary financial institution proceeds with user enrolment in the conventional manner so that they hold accounts; it is not intended to access users' transaction details.

- Guarantee transaction verifiability through contract management and zero-knowledge proofs of transactions. In case of suspected fraud or anomaly, competent authorities may audit the contracts to confirm regulatory compliance on the payee side. Evidently, guaranteeing privacy must not come at the expense of anti-fraud requirements.

- Provide satisfactory offline throughput. This assumes that the cost of privacy in terms of performance does not negatively impact throughput. The payment protocol and underlying cryptographic operations will be evaluated against the metrics documented in the state of the art.

4.3.1 The PrivTEE-Pay Model

This subsection describes the actors in PrivTEE-Pay, their roles, the adversarial model associated with the system, and the requirements that I want PrivTEE-Pay to guarantee. It is also worth noting that these various requirements implicitly entail certain trusted assumptions.

4.3.1.1 System Participants

The system consists of three types of actors: the central bank, intermediaries (commercial banks, financial institutions, etc.), and users. At a high level, these actors operate as follows. The central bank defines and steers monetary policy. Moreover, it is responsible for coin issuance and management of the single ledger. Commercial banks manage accounts and customer service; they also serve as relays for withdrawal and deposit operations issued by users. Users, on their part, can execute fund transfers between themselves or with a merchant in offline mode. To this end, users maintain a software wallet running on a mobile phone. The wallet is made available by the user's commercial bank; it is downloadable from an application store such as the Apple Store or Google Pay. Through this application, users can manage cryptographic keys, digital certificates, track holdings, and enable both offline and online transactions. Fig. 4.1 highlights the logical data flows between all system actors.

4.3.1.2 Trust Assumptions

Our goal is to design a system in which each user's funds and the integrity of the monetary system are safeguarded from interference by an external actor. The central bank is trusted to mint CBDC in compliance with its monetary policy. The distribution of minted CBDC occurs only if the withdrawal request is valid; this trust extends to financial institutions authorised to relay distribution operations.

We trust the registration authority either the commercial bank or another intermediary responsible for onboarding users to perform KYC checks. Evidently, this implies a trusted server bank account infrastructure. Consequently, we rely on user applications based on FLOSS (Free/Libre and Open Source Software), which allows source code modification,

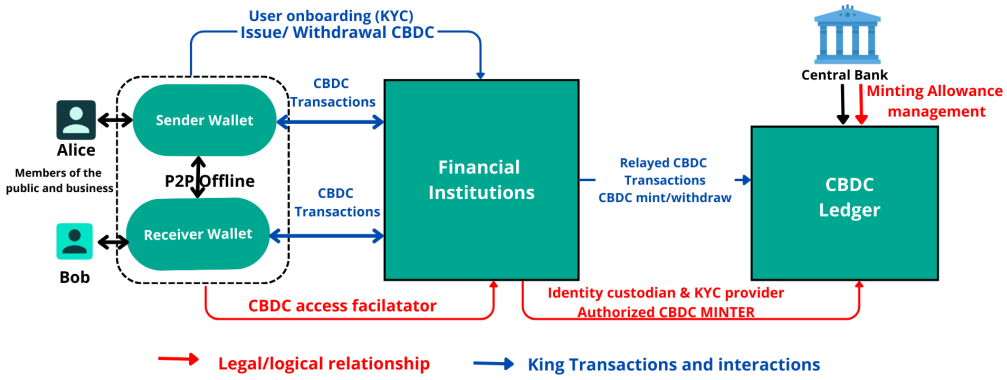


Figure 4.1: logical data flows between all system actors

vulnerability detection, and vendor independence. We also trust the application store through which users download the wallet made available online by their commercial bank.

Considering two clients, Alice and Bob: Alice, requiring a secure device with TEE for fund transfers, benefits from data confidentiality, integrity, and atomicity; thus she has no access to her private key, preventing transaction fabrication outside the offline wallet. Alice and Bob adopt ARM TrustZone TEE technology, which we trust [98]. Bob, the fund receiver, does not need a secure device. Secure and authenticated channels mediate all communications. Passwords and biometrics enforce access control, requiring approval of spending; secure storage thwarts rollback.

We assume that network-security measures operate flawlessly, thereby limiting denial-of-service attacks dramatically. Our model relies on proven cryptographic infrastructure. Consequently, we assume the reliability of cryptographic primitives digital signatures, commitment schemes, hash functions, etc. Moreover, we trust the PKI infrastructure responsible for securely managing and distributing users' public-key certificates.

4.3.1.3 Threat Model

Several threats can interfere with and undermine the integrity of the system, particularly on the security and privacy fronts. In this context users are not trusted, mainly because they may attempt to orchestrate illicit transactions. For instance, Bob might try to perform an offline transaction destined for his own offline wallet (Bob to Bob) in order to increase his holdings unlawfully. Another scenario occurs when he wants to use the coin spent in an offline transaction to fund his online account via a deposit procedure (Bob to Bob). These two cases are often referred to as double-incoming attacks.

A further fraud scenario involves Alice using the same coin for a transaction to Bob and another one to herself. This case typifies double spending, and Alice can re-send the same transaction to Bob. Such a fraudulent attempt is what we call a replay attack.

There is also a threat of a man-in-the-middle attack where an adversary may intercept or sniff a transaction frame. Subsequently, he could replay the transaction. The privacy of the coin is a concern, especially after the deposit phase: the central bank can trace the coin and identify the author of the original withdrawal. Moreover, all participants are susceptible to undermining the privacy of honest users.

4.3.1.4 Data Representation

Funds are represented as digital tokens. Analogously, they resemble electronic banknotes issued by the bank with a fixed value (5€, 10€, 20€) etc. Each "electronic banknote"

carries a serial number and is independent and self-sufficient; thus an e-coin does not contain any transaction history. Consequently, this format makes coin traceability almost impossible. This representation model was prototyped by David Chaum's original eCash design.

4.3.1.5 Requirements

4.3.1.5.1 Security

We consider the system secure if it satisfies the following requirements:

- **Operations authentication:** All operations must be orchestrated by system actors whose identities have been clearly verified.
- **Unforgeability:** An e-coin may only be issued by the central bank, and all users of the coin must be able to verify its authenticity.
- **Access control operations:** Authenticated parties must possess the necessary rights to execute a transaction.
- **Transaction integrity:** The system must protect transactions against attempts at alteration, falsification, or duplication by parties other than those authorized during the transaction.
- **Supply consistency:** The system must guarantee consistency of the monetary supply in circulation. In other words, the total amount of e-coins involved in users' operation flows must exactly match the number of coins issued and not yet destroyed. Evidently, such a property tolerates no unauthorized creation or unjustified loss of e-coin, and it presumes correct synchronisation after offline operations.
- **Transaction irrefutability:** This denotes non-repudiation. Thus, the author of a transaction cannot deny having created it, and the recipient cannot later deny having received it. Non-repudiation helps establish accountability for all parties involved in transactions.

4.3.1.5.2 Privacy

The protection of privacy is guaranteed by the following requirements:

- **Anonymity:** This applies first to the e-coin itself, which is completely untraceable. In this sense, the bank does not know the link between a coin and its user. The anonymity extends to all system actors whose identities should remain unknown to others unless they are parties to the transaction. Conversely, the identity of the central bank must be publicly known, especially during the issuance phase.
- **Confidentiality:** Outside the authorised parties in a transaction, no one may access the content of the transaction (e.g., the value of the coin).
- **Unlikability:** The identity of the payer must not be linked to the payee, the size, or any metadata of the transaction (time, location, etc.). Moreover, payments by the same sender should not be linkable to one another.

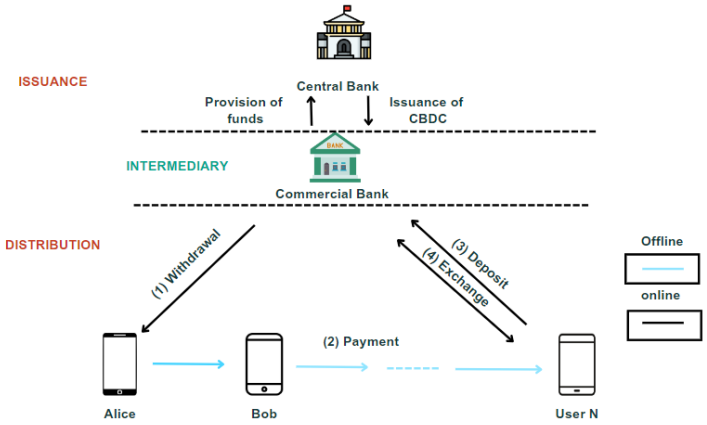


Figure 4.2: Architecture of transferable electronic payment

4.3.1.5.3 Auditability / Regulatory Compliance

A commercial bank may be authorised to audit suspicious transactions by inspecting the central database only if it holds a legal mandate from the competent judicial authority. The central bank maintains an up-to-date database of spent or exchanged coins. It can therefore prove that a coin has been spent, but it cannot identify who spent it.

4.3.1.5.4 Performance

The system's offline performance should be comparable to the average for CBDC payment systems operating offline. Throughput, latency, and scalability metrics must be evaluated to test the framework's viability.

4.3.2 The PrivTEE-Pay Framework

4.3.2.1 System Architecture

The proposed framework employs a two-tier architecture with a single ledger account. The system architecture requires a TEE and a certain number of functionalities. So, when Alice performs the onboarding step, the first registration step checks the TEE hardware and trusted OS for authenticity using a certificate from the equipment manufacturer (OEM). The initial functions of the system involve first setting up a device key pair, resetting the balance and nonce, and confirming the key's origin. Then, it proceeds to check and save the server's certificate. After the main steps can commence. In Fig. 4.2, we outline the system with the various steps. (1) is the withdraw. It consists of withdrawing of coins signed by the central bank and reduces Alice's online balance. (2) is payment. This step performs offline encrypt transaction. (3) and (4) are respectively deposit and Exchange steps.

4.3.2.2 Cryptographic Schemes

4.3.2.2.1 Blind Signature

According to Chaum, "a blind signature is used to create a cryptographic signature for a message without the signer learning the contents of the message being signed" [56]. This was used in our protocol for offline transactions. Leveraging user smartphone, each digital coin is 'blinded.' Basically, "a user's smartphone or other device can simply "blind"

a desired number f(x) by multiplying it by a randomnumber b that it chooses and raises to a denomination power" [57].

4.3.2.2.2 zk-SNARK

To mitigate double-spending, the zk-SNARK protocol, a non-interactive variant of Zero-Knowledge Proof, validates the transmitted coins by guaranteeing the uniqueness of each transaction. In fact, "using a succinct argument, the client would be able to verify the correctness of the result, using resources that are significantly smaller than those necessary to perform the task from scratch" [99]. Subsequently, combined with the blind signature, the zk-SNARK protocol maintains confidentiality and ensures transaction legitimacy, adhering to the zk-SNARK principle used in Zcash [100].

4.3.2.2.3 Diffie-Hellman protocol

We employ the Diffie-Hellman key exchange protocol to establish a link between the coin and money returned after a transaction or refund. Finally, two other technologies are used in our framework: digital signatures and digital certificates [101].

4.3.2.3 System Operations

Deploying the core functions of the system involves prior certification of the user as well as the application of the device he will be using. This is the wallet provisioning step. The remaining steps are related to the payment transaction. The main variables of the system are listed in 4.2.

Table 4.2: Notations and Descriptions

Notations	Descriptions
(Vk_A, Sk_A)	Key pair for Alice
Vk_B	Bob's public key
(J_k, Z_k)	zk-SNARK key pair
$cert_A$	Certificate generated by the server for vk_A
$cert_B$	Certificate generated by the server for Bob's public key
PK	Bank's public key for coin's verification
$((e, n), d)$	RSA keys corresponding to specific coin values established by the central bank
b	Blinding factor
f	Hash of the coin's public key (vk_A)
ff	Blinded hash of public key
s'	Blinded signature of ff
s	Unblinded signature
π	Proof string
$\vec{x}$	Public input (as blinded signed coin)
$\vec{w}$	Witness
db	Decision bit

The process begins with Alice generating a pair of signature keys (Vk_A, Sk_A) and sharing the authentication key with the server. The server then checks for a record associated with this key, and upon confirmation, the online balance is reset. Next, the server generates a certificate for Alice with server's private key, which incorporates the Alice's public key (Vk_A). This certificate is then sent to Alice. So, when she makes an offline

payment using this method on their device, Bob can verify Alice's authenticity through Alice's certificate. This ensures a secure transaction can take place with confidence.

4.3.2.3.1 Secure Trusted Application (STA) Registration Protocol

Alice initiates the registration process by connecting her secure device to the server. She generates a key pair and verifies its authenticity with the server. If confirmed, the server creates a certificate for Alice's key. This certificate is both stored on the server and sent to Alice along with is then stored by the server and sent with a pair of a proving key J_k and verification Z_k to Alice. Upon obtaining the certificate, Alice proceeds to use her device to validate it. The device cross-verifies the certificate using the authentication key from the server, which was previously stored in the device. If the validation process is successful, the certificate is then securely stored on the device for future reference.

4.3.2.3.2 Withdraw Protocol

Algorithm 1 outlines the procedure by which Alice securely transfers coins using blind signatures from her online balance to her offline wallet. The process commences with the cryptographic hashing of the coin's public key, followed by the server's blind signing this hash. Subsequently, Alice unblinds this signature to derive the authentic signature of the coin, which she then archives as a novel electronic currency unit. The procedure adopted for blind signatures is the one developed by David Chaum in [57] and presented in Fig. 4.3: The blind signature procedure, involves multiple steps to guarantee secure transactions. To commence, a client verifies himself with his commercial bank, which grants access to their device. This device acquires a public denomination key from the central bank for a specific coin value. Then, the device generates a coin's key pair alongside a blinding factor for enhanced security. After hashing and blinding the coin's public key, the device transmits this, along with an authorization for the coin withdrawal, to the commercial bank via a secure channel. The commercial bank deducts the amount from the client's account and authorizes the transaction with its own signature. This data is transmitted to the central bank, which applies a blind signature using its private key and sends it back. The client's device then unblind the signature using the blinding factor and securely stores the completed digital coin for future use.

4.3.2.3.3 Offline Sealed Transaction Protocol

Contract Negotiation. Through Algorithm 2, Alice negotiates a payment contract with Bob. Bob sends her a payment request detailing the amount and including a certificate. Alice uses a digital coin to sign the payment contract. Signing a contract with a valid coin is Alice's directive to pay Bob, identified by his certified public key in the contract. If one coin isn't enough for the total amount, clients can sign a contract using multiple coins. Alice then generates a payment proof using the 'proof' function. This function produces a proof string π (comprising eight elliptic curve points) using a public input ($\vec{x}$), a private witness $\vec{w}$, and the Bank's public key for coin's authentication, PK. In this case, the witness is Alice's private key. Alice then sends her public key (Vk_A), her certified public key ($cert_A$), the signature (s), the zk-SNARK proof, Bob's public key ($cert_B$), and the nonce. The transaction is encrypted using the private key (which Alice can't access as it's in the TEE) through the hashgraph method and sent via NFC.

Contract Checking. In Algorithm 3, Bob performs a series of checks on the validity of the transaction and the legitimacy of the sender and receiver. Bob's system records proofs and nonces to validate transactions. It rejects transactions if the proof is not as-

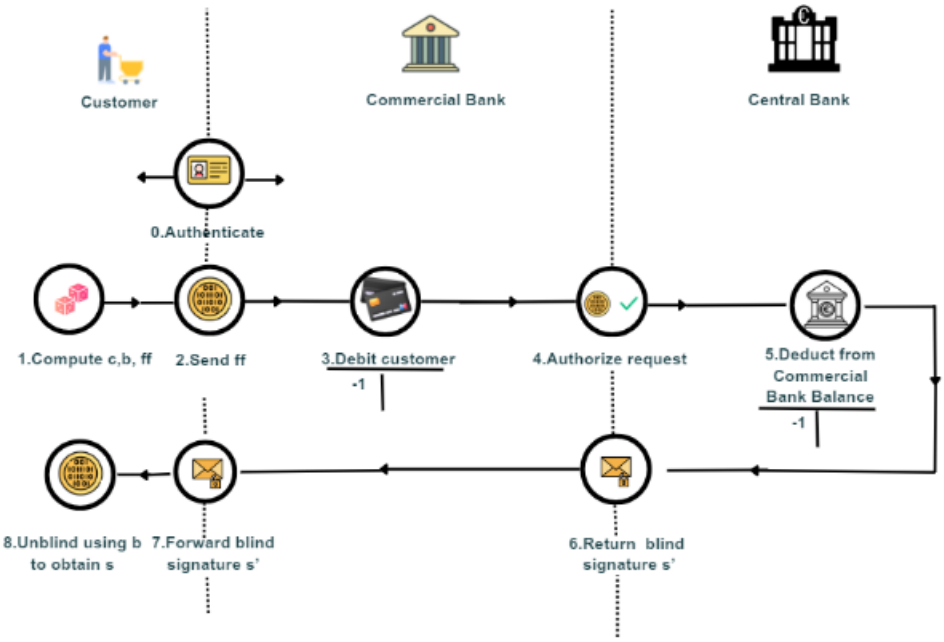


Figure 4.3: CBDC withdrawal based eCash1 process

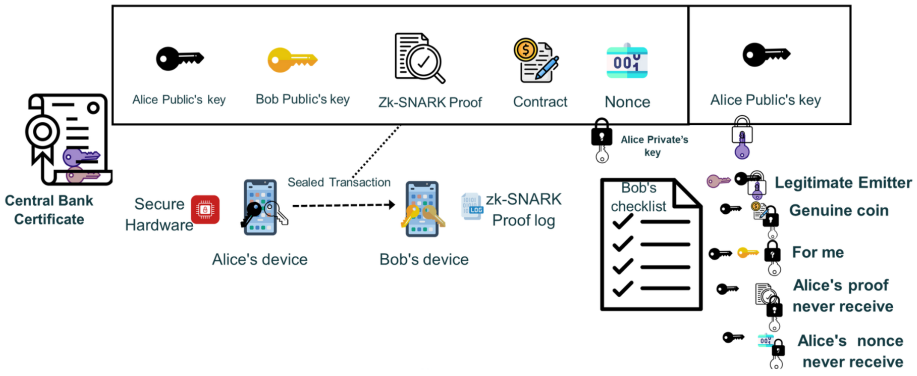


Figure 4.4: Offline CBDC Transaction based PrivTEE-Pay

sociated with Bob or if duplicate proofs are received, even with different nonces, as this could indicate fraudulent attempts. If the transaction passes these checks, Bob acknowledges receipt (to Alice) and stores the coin. Then Alice confirms to Bob that she has received the acknowledgment. Fig. 4.4 illustrates the sealed transaction between Alice and Bob.

4.3.2.3.4 Coin's Destruction

Concerning Algorithm 4, once the actual payment has been acknowledged, the original coin is hashed and then destroyed. At the end of the procedure Bob will have the original coin's version. It's worth highlighting that in situations where Alice's coin destruction process isn't entirely thorough, the coin becomes unusable once Bob receives the confirmation of the transaction's completion. This aspect resolves the double-spending problem.

Algorithm 1 Coin's Withdrawing

Purpose: Alice intends to transfer x coins with blind signatures from her online balance (maintained on the server S) to her offline wallet.

Require: Private inputs: b, d, Sk_A

Require: Public inputs: f, ff, s', e, n, Vk_A

Output: s

1: Compute $f \leftarrow Hash(Vk_A)$
2: Calculate $ff \leftarrow BlindHashedKey(f, b, (e, n))$
3: Send ff to the server
4: Obtain $s' \leftarrow BlindlySignCoin(ff, d)$
5: Calculate $s \leftarrow UnblindSignature(s', b)$
6: Store the new minted coin with $Store_New_Coin(Sk_A, s)$
7: Check if $ff \equiv f^b \pmod{n}$
8: Verify the signature of the coin using the server's public key
9: **if** the signature is valid **then**
10: Proceed with the transaction
11: Update Alice's online balance and offline wallet accordingly
12: **end if**

This is exactly what happens with cash, because when the issuer spends his money, he loses it to the recipient.

4.3.2.3.5 Coin Deposit and Exchange

Algorithm 5 describes the process the bank use to ensure through monitoring of its database that the coins are not already defunded by performing the same verification as Bob when he receives a contract. Moreover, clients can exchange their old coins for new ones after verification. Fig. 4.5 illustrates the detailed transaction between the sender and the receiver using our secure offline payment protocol.

4.3.2.4 System Analysis

4.3.2.4.1 Security

The system counters the threat of offline double-spending in two ways. First, the payee's device maintains a payment log. It must record the nonce and the zero-knowledge proof received in each offline transaction to guarantee their uniqueness. The bank likewise ensures that every eCoin and its synchronization transactions are unique by keeping a ledger of spent coins. It tracks the nonce of any deposit operation performed through online synchronization, and it records the corresponding zero-knowledge proof to confirm that the pair is unique; naturally it also logs the serial number of each eCoin. However, because this infrastructure is centralized, it presents a single point of failure: if the database becomes corrupted, fraud can occur. Given this latent risk, it may be prudent to perform synchronization over a distributed network such as a blockchain.

Second, the system employs the public key certificates of recipients. If an issuer replays a transaction for another payee possessing a different key certificate, no certificate match will exist. Withdrawal and deposit operations must also rely on a method that limits the number of transactions in order to satisfy security requirements and mitigate

Algorithm 2 Offline Contract Payment Transaction

Purpose: Alice wants to transfer x amount to Bob offline by negotiating a contract payment containing a payment instruction.

Require: Private inputs: $\vec{w}, \vec{x}, J_k, Vk_B, Contract, nonce, PK$

Require: Public inputs: $cert_B$

Ensure: Encrypted transaction data C

1: $receiver \leftarrow cert_B$ ▷ Bob selects receiver
2: Send payment request to Alice
3: Alice retrieves an electronic coin corresponding to the requested amount
4: SignContractUsingECoin(Contract, s) ▷ Alice signs the contract with the e-coin
5: $\pi \leftarrow$ GenerateProof$(\vec{w}, \vec{x}, J_k)$
6: $E \leftarrow$ Encrypt_Transaction$(\pi, Vk_B, Contract, nonce, PK)$ ▷ Transaction data encryption
7: Send the encrypted transaction data E to Bob

DDoS attacks. Furthermore, because the system follows the eCash-1 model, the bank can potentially trace an eCoin back to its original owner by combining various statistical techniques. This is why the eCash-2 model relies on a mix-network architecture to address that risk. Finally, part of the security depends on integrating secure elements such as TEE or TRE (Tamper-Resistant Element) modules, which are increasingly sophisticated. Nonetheless, zero risk against side-channel attacks does not exist.

4.3.2.4.2 Auditability

Upon reflection, although the system permits inspection for regulatory compliance, it remains only partially auditable. The audit is performed solely by the bank. A modern payment system, however, must be audited by an independent third party, either a single auditor or multiple auditors, to preserve neutrality and impartiality. Verifying signatures, including the blind signature used here, also ensures that an eCoin is authentic (issued by the bank) and not forged. Without this verifiability property, reliable auditability cannot be guaranteed. Moreover, auditability is limited to eCoins rather than all participants.

4.3.2.4.3 Privacy

The implemented system protects consumers from profiling and subsequent data monetisation. Withdrawal, payment, and deposit operations resemble cash transactions. The advent of CBDCs with offline payment aims to replicate the properties of physical cash. By executing transactions offline, the system offers a high level of privacy through coin anonymity. Nevertheless, it still requires users to hold a bank account. Future work will aim to eliminate the need for a bank in managing cash, mirroring how cash can be used without an account. This does not preclude intermediaries from monitoring fund movements to prevent financial instability.

4.3.3 PrivTEE-Pay Implementation and Evaluation

4.3.3.1 Experimental Setup

The framework proposed in this chapter was implemented using the Offsim-CBDC simulator. Offsim-CBDC is fully written in Python and executed with Python 3.11. For data collection, export into structured CSV datasets, as well as for subsequent quantitative

Algorithm 3 Transaction Verification by Bob

Purpose: Bob checks that the transaction is valid by verifying Alice's legitimacy as the sender, the authenticity of the coin, and the correctness of the zk-SNARK proof.
Require: Private inputs: $cert_A$, Vk_B, $Contract$, s, f, Vk_A, $\vec{x}$, π, Z_k
Require: Public inputs: PK
Ensure: Decision bit db
Step 1: Verify Alice's public key
if $cert_A$ = FALSE( )**then**
    Transaction fails and exit
end if
Step 2: Verify coin's authenticity
if VALIDATE_COINSIGNATURE(Vk_B, Pk, $Contract$, s, f, Vk_A, π) = False **then**
    Transaction fails and exit
end if
Step 3: Check receiver validity
if RECVERIFY(Vk_B) = False **then**
    Transaction fails and exit
end if
Step 4: Verify the zk-SNARK proof
$db \leftarrow$ VERIFY_PROOF(π, Z_k, $\vec{x}$)
if $db = 0$ **then**
    Transaction fails
end if
Step 5: Final decision
if all checks pass **then**
    Transaction is valid and associated with Vk_B
end if

evaluation and visualisation, the `pandas` library was employed. The hardware architecture relies on a 12th-Gen Intel®Core™i7-1265U CPU @ 1.8 GHz machine, equipped with 32 GB DDR4 RAM and 512 GB SSD storage. The development platform is Jupyter Lab installed on Windows 10 Professional (64-bit). All system components, including cryptographic primitives, networking modules and device-communication interfaces, are implemented via Python libraries.

Thus, offline peer-to-peer (P2P) transfers are physically emulated using two ACS ACR122U PICC Interface NFC readers connected through USB. These devices operate independently to represent respectively the payer and payee endpoints. The set-up establishes a bidirectional NFC link that mimics device-to-device exchange. The `pyscard` library (Python Smart Card API) is used to establish and manage this communication. In this context, the PC/SC protocol transmits and authenticates low-level Application Protocol Data Units (APDUs). Conversely, transaction data are serialised before encryption and transmission using JSON payloads. The JSON messages encapsulate sender/receiver public keys, the coin structure, zero-knowledge proof nonces, and timestamps. Payload authentication is achieved through AES-GCM associated data (AAD), linking it to the recipient's fingerprint and protocol identifier. Data transmission occurs in two modalities depending on context. First, using the Ipv8 framework: with Ipv8, messages are logically encapsulated between peers or towards central banks, and it also simulates P2P routing. Second, via NFC exchanges that reproduce the characteristics of a physical transport

Algorithm 4 Coin Destruction in Alice's Device

Require: Private inputs: s, $Hashed_Coin$
Require: Public inputs: None
Ensure: Acknowledgment status (Ack)
Step 1: Upon receiving acknowledgment
$Hashed_Coin \leftarrow$ HashCoin(s)
Step 2: Destroy the hash
clearKey($Hashed_Coin$)
Step 3: Check destruction result
if $Hashed_Coin$ is successfully cleared **then**
    $Ack \leftarrow$ "Acknowledgment Successful and Coin Destroyed"
else
    $Ack \leftarrow$ "Acknowledgment Failed or Coin Not Destroyed"
end if

Algorithm 5 Bank Coin Verification and Exchange

Require: Private inputs: Vk_B, $bank_Publickey$, s, f, $proof$, C, $coins(e, n)$
Require: Public inputs: None
Ensure: Verification status ($Vstatus$)
Step 1: Monitor the bank's database for coin defunding
if DepositCoin(Vk_B, $bank$, s, f, $proof$, C) = False **then**
    $Vstatus \leftarrow$ "Coin Already Defunded or Verification Failed"
    Exit
end if
Step 2: Handle coin exchange requests
$NewCoins \leftarrow$ ExchangeOldCoinsForNew($coins(e, n)$)
if $NewCoins$ are successfully generated **then**
    $Vstatus \leftarrow$ "Coins Exchanged Successfully"
else
    $Vstatus \leftarrow$ "Coin Exchange Failed"
end if

layer.

Moreover, to orchestrate cryptographic operations, the `PyCryptodome 3.20.0` library was imported into the Jupyter notebook. Concretely, RSA-2048 is used for asymmetric encryption, hybrid key exchange, and Chaum's blind signature generation—particularly during coin issuance by the central bank. For asymmetric encryption of offline transaction payloads, AES-256 in GCM mode is employed to ensure confidentiality and authenticity. The SHA-256 hash algorithm is also utilised; consequently, coin identifiers, proof references, and transaction indexing mechanisms rely on cryptographic digests from SHA-256. Cryptographically secure randomness is applied to keys, nonces, and blinding factors, generated via the OS entropy source through PyCryptodome's `get_random_bytes()` API.

To simulate a secure enclave, a Trusted Execution Environment (TEE) was emulated by a software component named `TEEEmulator`. This module encapsulates all cryptographic operations particularly proof generation and coin issuance. The software TEE ensures data isolation and integrity within the Python runtime. Additionally, the TEE exposes a `TEEProof` interface that cryptographically signs JSON payloads representing zero-knowledge attestations. These proofs are verifiable by any peer using standard

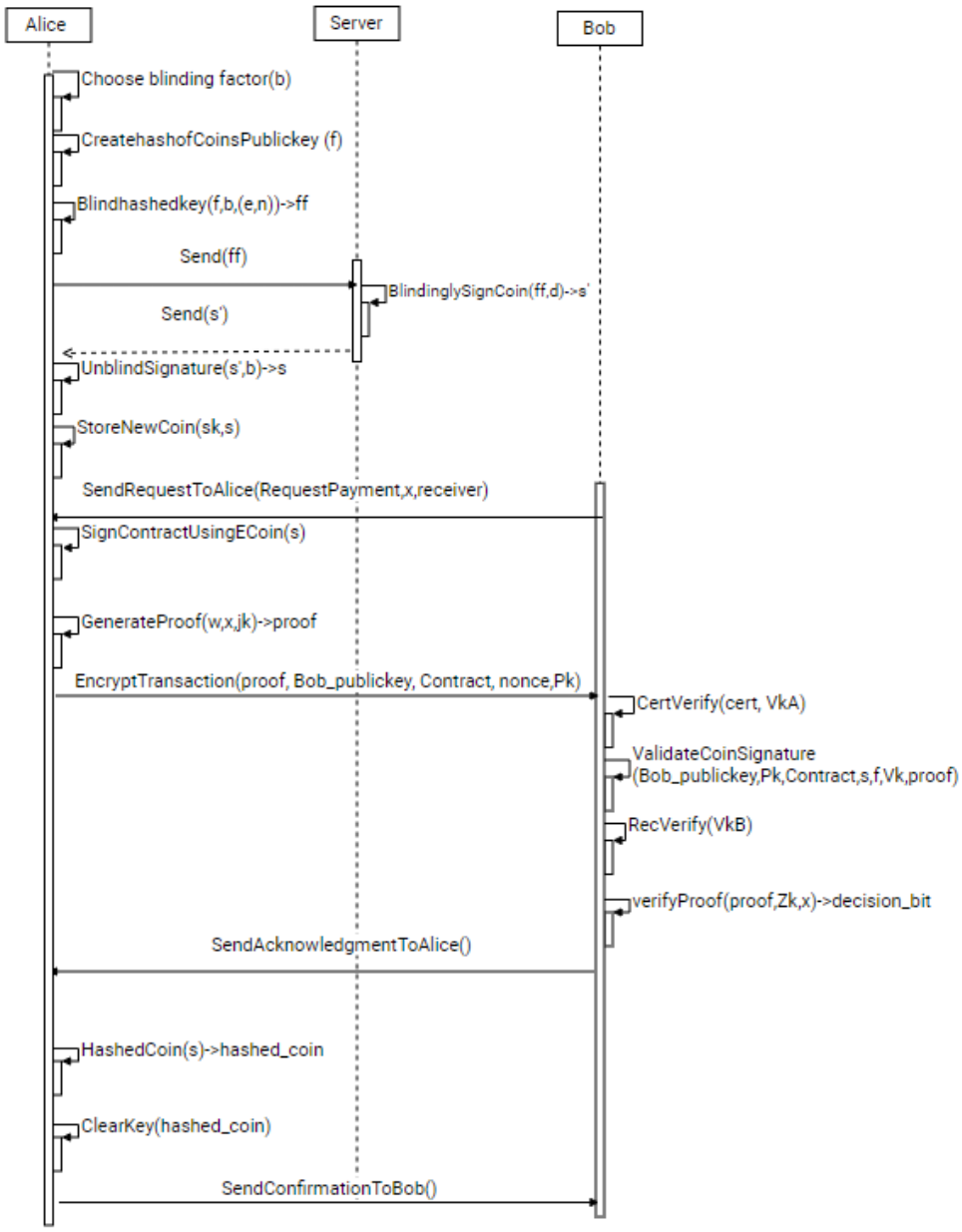


Figure 4.5: A sequence diagram of the detailed transaction between the sender and the receiver using our secure offline payment protocol

RSA-2048 digital signatures, thereby guaranteeing authenticity and non-repudiation.

4.3.3.2 Evaluation Metrics

The system's performance is evaluated at three levels: the withdrawal phase, the offline transfer phase, and the deposit phase. Naturally, the offline payment phase will be scrutinised with the greatest attention; other phases are analysed only in relation to the metrics of offline transactions, particularly to determine whether they impact offline operations. Accordingly, the selected performance metrics for evaluating transaction efficiency include latency, CPU load, cryptographic proof generation and verification times, message size, and throughput. The simulator automatically logs detailed performance data into structured CSV datasets for each transaction phase.

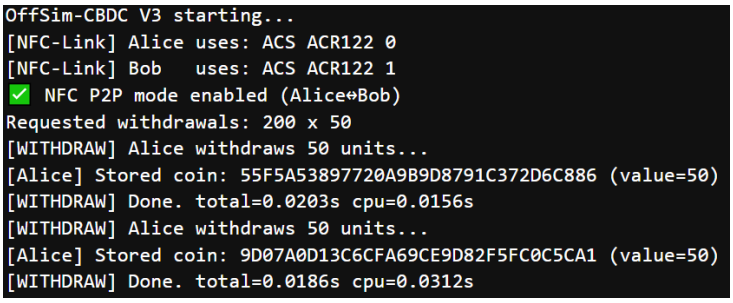


Figure 4.6: Online Withdrawal Results

4.3.3.3 Experimental Results Evaluation

4.3.3.3.1 Online Withdrawal performance

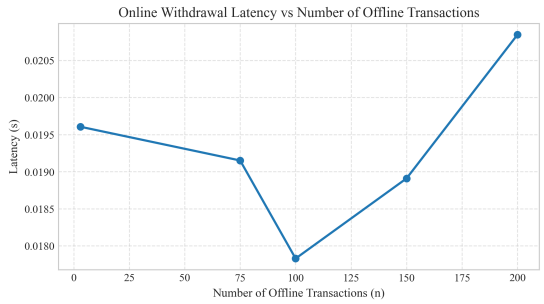
The online withdrawal phase represents the first transaction stage. It consists in provisioning coins into the user's wallet. Consequently, it is indispensable for subsequent offline transactions. The withdrawal phase was retrospectively analysed with respect to the offline transactions performed thereafter. Thus, within this simulation, the experimental configuration adopted was as follows: each withdrawal operation was executed five times with a constant value of 200 units. For every withdrawal operation, the number of subsequently simulated offline transactions varied gradually between **50, 75, 100, 150 and 200**. Fig. 4.6 highlights the establishment of the communication channel between the two NFC devices. It also shows a sample of results from the withdrawal phase. In Fig. 4.7a, one can observe that the latency of online withdrawal is almost identical, with a slight variation between **0.018 s** and **0.021 s**. This trend demonstrates an optimisation of the blind-signature operation between the bank and the user. It also indicates that the latency metric does not affect the fluidity of subsequent offline transactions. This evidence opens the door to a potential scalability of the mechanism for massive offline transactions.

In the same vein, Fig. 4.7b shows that the corresponding **throughput** metric is also stable. It oscillates between **50 tx/s** and **56 tx/s**. This stability follows from the analysis of the latency metric. Indeed, observing throughput suggests that the signing process on the bank's side does not constitute a bottleneck. It confirms the system's ability to supply the offline network with fresh currency. Moreover, CPU load in Fig. 4.7c follows a similar trend. It varies between **0.018 s** and **0.020 s**. Overall, this graph, together with other metrics from this phase, reinforces the system's capacity to efficiently prepare multiple offline sessions without compromising responsiveness.

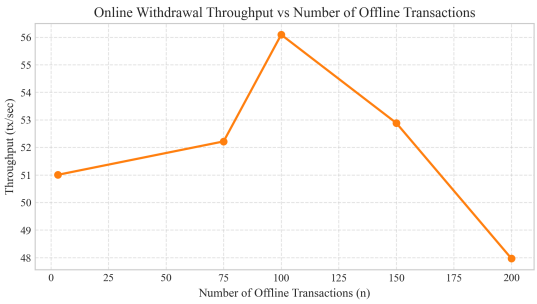
4.3.3.3.2 Offline transaction performance

The offline transaction phase constitutes the core of the *OffSim-CBDC* experiment, simulating peer-to-peer exchanges via NFC between two agents without network connectivity. Fig 4.8 highlights the establishment of a connection between Alice's and Bob's NFC devices as well as the successful offline peer-to-peer transfer. Fig. 4.9a and Fig. 4.10c present the evolution of key metrics. In this case, latency, throughput, CPU load, and cryptographic times are shown as a function of the number of simulated transactions.

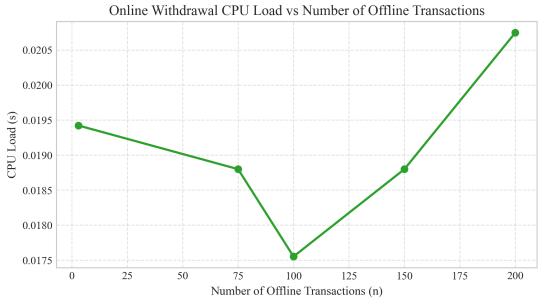
Thus, the average latency (Fig. 4.9a) ranges between **0.03 s** and **0.04 s** per transaction. This trend reflects a non-linear but overall stable variation. Such stability suggests that real-time exchanges are feasible. The delays remain well below the perceptible threshold for users.



(a) Online Withdrawal Latency vs Number of Offline Transactions



(b) Online Withdrawal Throughput vs Number of Offline Transactions



(c) Online Withdrawal CPU Load vs Number of Offline Transactions

Figure 4.7: Performance evaluation of online withdrawal operations under increasing offline transactions.

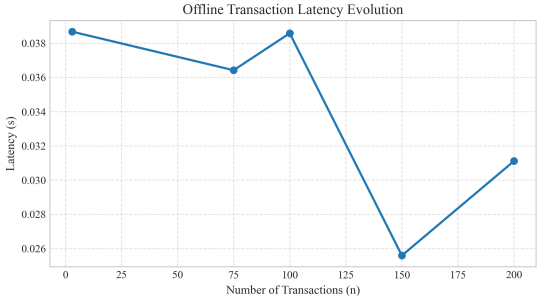
```
Present a tag on Alice reader then on Bob reader (presence required
both sides)...
[NFC-Link] Alice presence OK, UID=89 7B 2B 03
[NFC-Link] → (4664 B) Alice → Bob (logical transfer)
[NFC-Link] Bob presence OK, UID=BA FC 29 03
[NFC-Link] ← Bob received packet
[Bob] Stored coin: 80E15DD6C2E0A0B79D03CDCA9D7F5B01 (value=50)
[OFFLINE] OK. latency=0.0231s zk(gen=0.0030s, ver=0.0010s) msg=4664B
Present a tag on Alice reader then on Bob reader (presence required
```

Figure 4.8: Establishment of a connection between Alice's and Bob's NFC devices

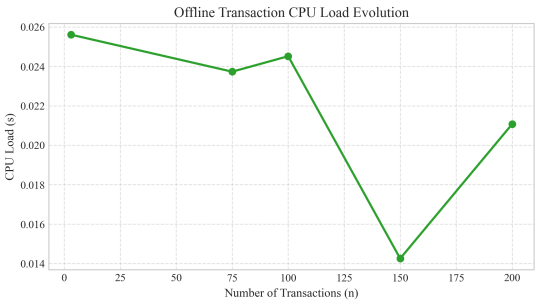
Furthermore, CPU load in Fig. 4.9b stays around **0.02–0.025 s**. This observation indicates an efficient execution of AES-GCM and RSA-OAEP primitives used to seal and transmit packets securely. Throughput (Fig. 4.9c) varies between **25 tx/s** and **38 tx/s**. Nevertheless, a brief drop is observed for **150 transactions** before recovery at **200 transactions**. This momentary dip results from JSON serialization overhead.

Regarding cryptographic times, Fig. 4.10a and Fig. 4.10b show a pronounced asymmetry between the generation and verification of zero-knowledge proofs (ZK). Generation time reaches about **0.018 s**, compared to **0.0006 s** for verification. This asymmetry is illustrated comparatively in Fig. 4.10c. Evidently, the computational cost concentrates mainly on the proof issuer. Nevertheless, both operations complete within a second. Such operational duration validates the feasibility of integrating zero-knowledge proofs on constrained NFC devices.

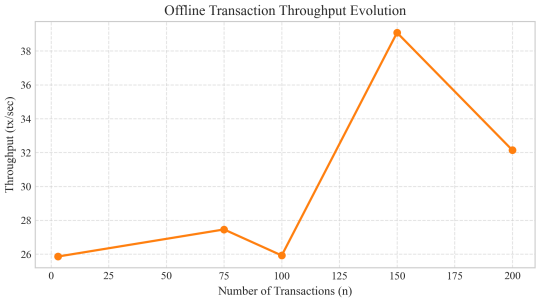
Finally, Fig. 4.11 shows the distribution of encrypted message sizes exchanged during the offline phase. Packets measure an average of **4.67 kB**. This stable value, independent of transaction count, reflects a uniformity in cryptographic metadata and JSON structure.



(a) Offline Transaction Latency Evolution

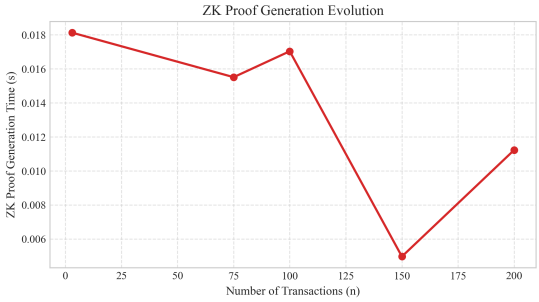


(b) Offline Transaction CPU Load Evolution

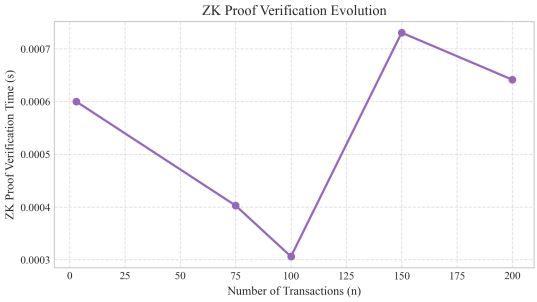


(c) Offline Transaction Throughput Evolution

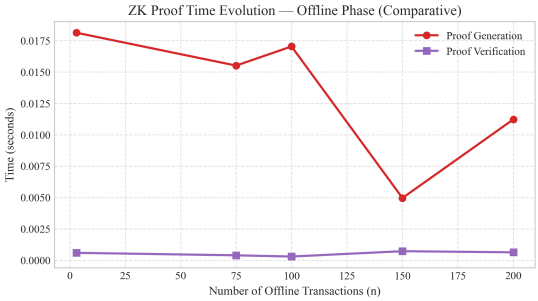
Figure 4.9: Performance evolution of offline NFC transactions under increasing workloads.



(a) ZK Proof Generation Evolution



(b) ZK Proof Verification Evolution



(c) Comparative Evolution of ZK Proof Times During Offline Phase

Figure 4.10: ZK proof computation characteristics during offline transaction processing.

4.3.3.3.3 Deposit phase and Global Throughput Scalability

The deposit phase corresponds to the reintegration of offline tokens into the central ledger. Fig.s 4.12a and 4.12b exhibit a low latency (approximately **0.003–0.004 s**) and a high

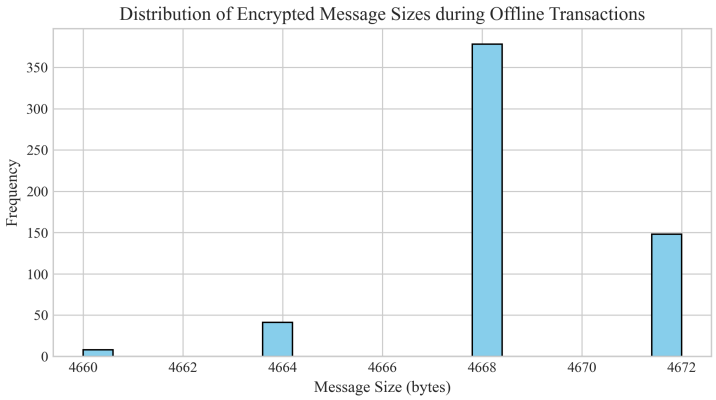
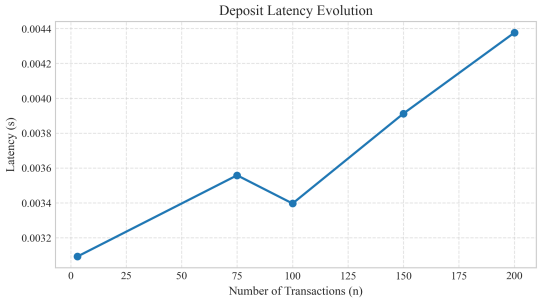
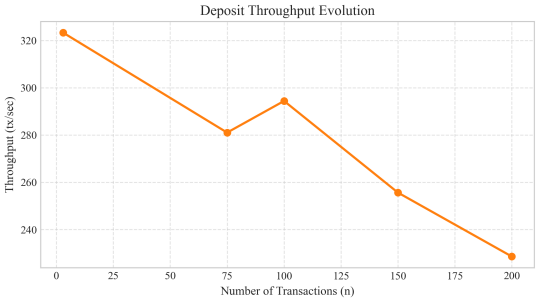


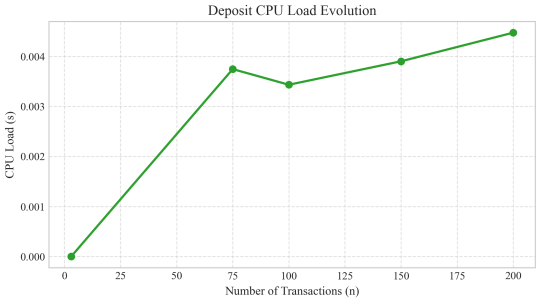
Figure 4.11: Distribution of Encrypted Message Sizes during Offline Transactions



(a) Deposit Latency Evolution



(b) Deposit Throughput Evolution



(c) Deposit CPU Load Evolution

Figure 4.12: Performance metrics of deposit operations under increasing offline workloads.

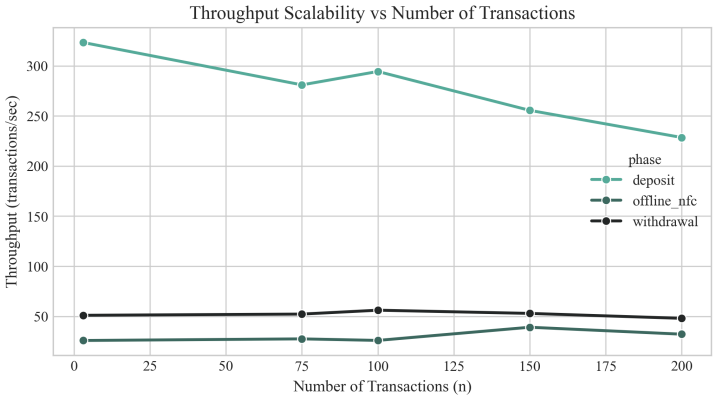


Figure 4.13: Throughput Scalability vs Number of Transactions

throughput, exceeding **300 tx/s**. These figures can be explained by the inexpensive nature of bank-side operations, which are essentially limited to signature verification and transaction recording. A global scalability analysis (Fig. 4.13) reveals that withdrawal maintains an almost constant throughput around **50 tx/s**. It also shows that the offline phase sustains an average between **25 tx/s** and **40 tx/s**. Finally, deposit surpasses the other two phases in terms of throughput (**300 tx/s**). Reflecting on this disparity, one sees that operational scalability is affected by computational cost: the higher the computational burden, the less performant the scalability. In the conducted experiment, it goes without saying that the major computational load concentrates during peer-to-peer exchanges, particularly because the combination of encryption, ZK proofs, and NFC communication is most demanding. Given these overall results, the proposed Framework retains stable and predictable performance even under increasing loads.

4.3.4 Discussion

The preliminary chapters highlighted CBDC payment requirements such as security, privacy protection, and performance. From a performance perspective, the joint analysis of experimental metrics from the PrivTEE-Pay framework revealed smooth preparation of coins/tokens during withdrawal. It also demonstrated that real-time offline transactions orchestrate with delays below **50 ms**. Moreover, the PrivTEE-Pay framework illuminated the cryptographic feasibility of ZK proofs on embedded NFC environments. Finally, it achieved a highly performant, linear, and low-cost post-transaction reintegration.

4.3.4.1 Comparative analysis

Table 4.3 displays a range of offline digital payment approaches built on conventional architectures. Some solutions report complete experimental performance data, whereas others provide no such metrics at all [39, 51, 54]. Nevertheless, even though these approaches are limited in operational evaluation,they still highlight recurring issues related to integration with the secure element, forgeability concerns, and the lack of strong privacy guarantees.

In contrast, the work by [49, 50] evaluates their protocol under real-world conditions. Consequently, reported latencies vary significantly. In [49], transmissions range from 112–568 ms, whereas scenarios involving 20 blocks and 60 nodes exceed 1.5 s. This variability makes the solution ill-suited for fast transactional environments such as public transport. The 2024 proposal by the same authors slightly improves issuer-side performance (275 ms), yet remains penalised on the receiver side (≈1600 ms) , likely due to bilinear pairing operations required for double-spend control.

Moreover, [53] offers stronger privacy guarantees, thanks to advanced cryptographic proofs generated via a PET mechanism. However, this integration incurs a noticeable computational overhead. Thus, creating a payment is orchestrated in roughly 210 ms, with an additional 90 ms during acceptance within a window of 50 unsigned transactions. Despite its robustness, the solution therefore operates in a relatively high-latency band.

Furthermore, [102] evaluates the energy consumption associated with offline payments stored in a secure element. From this assessment it follows that performance remains reasonable, with less than 23 ms of extra energy expenditure over 100 transactions. Nevertheless, the absence of measurements regarding double-spend detection and cryptographic execution itself limits direct comparability with other works.

Compared to these approaches, Block-PAD proposes a framework oriented toward full offline resilience with lower latencies (40–47 ms) and a stable throughput (≈ 23.57 TPS).

Unlike solutions based on a single-ledger architecture, Block-PAD relies on a flexible design capable of integrating adaptive privacy primitives while avoiding the heavy cryptographic overhead observed in traditional PET architectures. This solution therefore appears more balanced, offering a robust compromise between performance, privacy, and offline feasibility.

Finally, PrivTEE-Pay is an operational framework that reproduces NFC transactions in offline mode. It ranks among the most performant systems, with latencies of 30–45 ms, a stable throughput between 35 and 40 TPS, and cryptographic proofs verified swiftly (generation: 0.018 s; verification: 0.0006 s). Due to its lightweight computation and ability to operate in real-world conditions, PrivTEE-Pay stands out as one of the most efficient CBDC solutions from an operational and security standpoint. It achieves near real-time performance while remaining compatible with a decentralized offline architecture.

4.3.4.2 Addressing User Concerns and Profiling Risks

Our solution is centered around incorporating privacy principles into CBDC infrastructures. Through the utilization of privacy enhancing technologies (PET) that merge blind signatures with zk-SNARK, our approach safeguards client identities, transaction amounts, and detailed activities from central banks. This prevents the tracing of a coin's original user based on payment history. The recipient only possesses payment proof without sender identity or transaction metadata. However, commercial banks manage Know Your Customer (KYC) and monitor fund movements to prevent financial instability due to bank disintermediation. Users express concerns about banks knowing their expenses in detail, not their identification. Profiling risks pertain to understanding consumer spending habits, not their monetary identity. We developed a privacy-focused system emphasizing recipient client privacy for Anti-Money Laundering (AML) and Combating the financing of terrorism (CFT). Client revenue transactions are managed by associated contracts, facilitating inspection by commercial banks and regulatory authorities for revenue anomalies. Offline transaction amounts might be capped, and withdrawn amounts shouldn't surpass spent amounts without synchronization with the online commercial bank.

4.3.4.3 Innovative CBDC Payment Approach

Our system introduces an innovative CBDC payment approach, different from the typical bank deposit-based models. It replicates the security of banknotes in digital form, enhancing security against counterfeiting. Coins signed via signature keys can be exchanged for new ones before key expiration, mitigating security risks even in case of key compromise. if value signature keys are compromised, banks can revoke them via the app. Our open-source, software-based solution doesn't necessitate new infrastructure, prevents fraud and provides quantum-resistant privacy due to the blinding factor. However for interoperability purpose we can extend the use of this protocol to a smart card and consider the case of payment from an offline holder to a user who does not have one but can connect to the blockchain via their identifier and carry out defunding.[104, 105]

4.3.4.4 Addressing Protocol Limitations and zk-STARK Advantages

Albeit our protocol offers a proven guarantee of security and privacy protection, it can be improved by replacing zk-SNARK with zk-STARK, whose theoretical and experimental potential makes it an alternative and enhanced solution[106]. Unlike zk-SNARKs, zk-STARKs do not require an initial setup phase. This characteristic eliminates the risks

associated with generating and securely storing the initial parameters, which, if compromised, could compromise the security of the entire system. The zk-STARKs are resistant to quantum attacks and provide future protection against threats from quantum computers. Moreover, zk-STARKs proofs offer faster proof generation, which is crucial in high-volume systems processing thousands of transactions per second. Nevertheless, zk-STARKs being a relatively recent technology, it is important to seriously evaluate their reliability and test them in plethora of use cases. Additionally, they are more resource-intensive considering the size of the zk-STARK proof.

4.4 Conclusion

This chapter demonstrated, through the `PrivTEE-Pay` framework, that a conventional infrastructure can be adapted to meet the requirements of offline CBDC payments. By jointly integrating blind signatures, the zk-SNARK protocol, certified identities and secure execution environments, it is possible to design an offline payment framework for CBDC that places privacy protection at its core. In addressing the objectives set out in the introduction, the chapter proved that decoupling identity data from transactional data can be achieved without compromising verifiability, non-repudiation or double-spend prevention. It also highlighted several limitations and avenues for improvement, such as enhancing proof efficiency by adopting zk-STARKs.

All things considered, these findings have major implications. They show that privacy must be regarded as a foundational pillar in the design of CBDC protocols rather than an optional add-on. They further reveal that technological choices concerning cryptographic proofs will strongly influence both adoption and regulatory compliance. The remainder of the thesis builds on these foundations, weaving together the successive contributions of Block-PAD and PrivTEE-Pay to present a coherent vision of offline CBDC payments and to examine their prospects for integration into broader application environments.

4.5 Chapter Summary

This chapter proposes a comprehensive framework called `PrivTEE-Pay`. It guarantees the confidentiality of offline CBDC payments through a combined use of blind signatures, zk-SNARKs, digital certificates and a secure execution environment (TEE). The key challenge is to demonstrate that an offline payment solution can be implemented without new infrastructure by relying on proven cryptographic standards. Concretely, the approach allows complete separation of transactional data from user identity while preserving verifiability and resistance to double spending. Moreover, the protocol ensures that a recipient receives only proof of payment validity, with no access to the payer's identity or history. The chapter details all operational protocols: client and device certification, withdrawal, payment, verification, secure coin destruction, and bank deposit. It validates experimentally that `PrivTEE-Pay` meets AML/CFT requirements. Finally, it identifies improvement prospects, notably the adoption of zk-STARKs.

Table 4.3: Comparison of Proposed Offline Payment Solutions Based on Conventional Ledger

Solutions	**PET**	**Privacy Type**	**Secure Element**	**Risks/Drawbacks**	**Performance Evaluation**
[39]	No	Confidentiality	Yes	Forgeability / DDoS	Not provided
[49]	No	Confidentiality	Yes	Weak privacy	112–568 ms transmission; up to 1590 ms in distributed mode
[50]	No	Controllable anonymity	Yes	Payee anonymity not guaranteed	Withdraw $<$ 400 ms; Spend: 275 ms (payer), 1600 ms (receiver)
[54]	No	Semi-private	No	Partial offline capability	Not provided
[59]	No	Confidentiality	Yes	Not provided	Not provided
[43]	No	Data privacy	Yes	TTP usurpation risk	Not provided
[53]	Yes	Data privacy	Yes	Large cryptographic proofs	$\sim$ 210 ms payment creation; $\sim$ 90 ms acceptance per 50 unsigned elements
[51]	No	Confidentiality	Yes	Supply chain attack	Not provided
[102]	No	Confidentiality	Yes	Secure element dependency	$<$ 23 ms extra cost for 100 tx; no security latency evaluation
[103]	Yes	Selective disclosure	Yes	Coordination required between smartcard and mobile	$\sim$ 88% success at 50–100 tps (simulated)
[65]	Yes	Privacy-preserving	No	Moderate TPS compared to high-throughput protocols	40–47 ms latency; 23.57 TPS; $\sim$ 100% success
PrivTEE-Pay	Yes	Privacy-preserving	No	NFC dependency; serialization overhead	30–45 ms latency; 35–40 TPS; ZK: 0.018 s generation, 0.0006 s verification; $\sim$ 100% success

Chapter 5

Conventional-Ledger Offline Payments: DigiVaultCard Extension

5.1 Introduction

After establishing in the previous chapter a solution that enables secure offline payments based on a single-ledger infrastructure, this chapter turns to the security of wallets operating offline. Even though smartphone devices equipped with a TEE already provide transaction security, they remain vulnerable to physical attacks (side-channel, power analysis, etc.) and software compromises such as rooting or jailbreaking. Therefore, this chapter augments the architecture by introducing a smart card that functions as a Hardware Security Module (HSM) and an additional fraud-prevention layer for offline mode.

Accordingly, this chapter focuses on presenting DigiVault Card and MarkoPayChain. DigiVault Card is a secure offline payment card that offloads cryptographic operations to a smartphone, while MarkoPayChain is a probabilistic model designed to proactively detect fraud patterns. The objective is to show how this architecture meets critical requirements: key security, double-spend prevention, cryptographic verifiability, privacy, and the ability to handle resilient transactions without network connectivity. It also illustrates how the hybrid approach surpasses purely mobile solutions by offering strong authentication and secure storage of values. The chapter opens with an overview of the key building blocks and the proposed model in Section 5.2. The proposed model describes the various protocol phases: onboarding, coin withdrawal, offline payment, contractual verification, deposit, and synchronization. A dedicated Sub-section analyzes the functioning of the MarkoPayChain model and its role in early detection of suspicious transactions. Then, Section 5.3 , is an experimental evaluation of key metrics and Section 5.4 is about the discussion. Finally, Section 5.5 concludes the chapter while Section 5.6 presents its summary.

5.1.1 Background

The last decade has been marked by spectacular innovations in the payment industry. At the heart of these innovations are instant payments, open banking, mobile payments, electronic wallets (such as Apple Pay, Google Wallet, and Alipay), and the tokenization of payment methods. These innovations aim to enhance the speed, security, and convenience of payments while incorporating emerging technologies such as blockchain and artificial intelligence. To address the surge of new markets and meet the evolving expectations of consumers, private entities like Fime are specializing for supporting and evaluating trusted digital payment experiences across fintech, banking and smart mobility, including

biometric and CBDC technologies. Indeed, CBDCs, which are digital currencies issued by central banks, represent a significant shift in the payment industry. Historically, transactions have focused on security techniques within the transaction distribution chain. With the advent of CBDCs, we are now witnessing a human-centric paradigm shift where the concept of security is closely associated with privacy and other consumer-desired features such as offline payment capabilities.

Offline payments, valued for their resilience and financial inclusion potential, face major obstacles including the security of cryptographic key storage, fraud protection, limited computational capacity of certain devices, user privacy and transaction privacy, as well as compliance requirements concerning anti-money laundering (AML) and counter-terrorism financing (CFT). In response to these challenges, a few solutions such as [43, 52] have been proposed using mobile devices, but very few have leveraged smart cards. However, this research proposes a new approach. It identifies the potential and gaps in the use of smart cards for offline digital payments and offers a hybrid solution that maximizes privacy and security, particularly in protecting against double-spending and fraud [107].

5.1.2 Related Work

Reference [51] proposes an offline CBDC payment solution inspired by [39]. It is implemented on the AKIS Smart Card, which complies with ISO/IEC 7816 and ISO/IEC 14443A. This card protects stored data from unauthorized access and tampering. Its operating system safeguards confidential information against external threats. Conversely, [102] offers hardware wallets based on Physically Unclonable Function (PUF) technology. These include standalone smart card wallets and a USIM/eSIM type for mobile usage. PUF technology ensures a unique inborn ID and a hardware root of trust (RoT). Ongoing work under ISO/IEC 20897-1 supports these developments.

Moreover, [37] advocates for a universal trusted physical wallet. It guarantees decentralized CBDC storage and circulation through a simple "Touch and Pay" process. Its non-custodial design combines strong privacy and security that withstands post-quantum threats. Furthermore, [108] introduces an offline payment solution compliant with EMV standards. Called EOPMA, it relies on mutual authentication and reverse hash chains to prevent exceeding credit quotas during multiple offline payments. This solution, though implemented on NFC phones, extends the existing EMV protocol. In contrast, our approach hybridizes smart cards and mobile devices within a single offline payment protocol.

5.2 Materials and methods

5.2.1 Key building blocks

This section briefly presents the cryptographic technologies and methods that are central to our protocol:

5.2.1.1 Offline functionality

This is broadly defined as the absence of internet connectivity and connection to a ledger. In this scenario, the usual communication channels are NFC (Near Field Communication), Bluetooth, and QR codes.

5.2.1.2 Blind Signature

As described by David Chaum, this involves an author, A, masking a message m with a blinding factor b and transmitting it to a recipient, B, who signs it with s without knowing the content. This research utilizes the cryptographic method applied in eCash2, where the message is replaced by the coin sent to the bank for signature [57].

5.2.1.3 zk-SNARK Protocol

A non-interactive variant of the Zero-Knowledge Proof protocol, which provides proof of possessing information (proof of knowledge) without revealing the information itself. This protocol involves three key algorithms: the key generation algorithm, the proof provider algorithm, and the verification algorithm [99].

5.2.1.4 Markov Chains.

A Markov chain is a stochastic process that describes a sequence of random variables over time, whether discrete or continuous. In a Markov chain, the prediction of a future state depends solely on the current state, without influence from past states.

5.2.2 Assumptions and security design

The offline payment protocol is orchestrated within a 2-tier global architecture for CBDC payments, as highlighted in [43]. Additionally, the storage of keys is managed within the card, as it is assumed that neither the issuer nor the receiver can fully trust each other's mobile devices. Consequently, the mobile phones used in the context of the offline payment protocol are presumed to be insecure and susceptible to hacking.

5.2.3 The proposed model

5.2.3.1 User onboarding procedure

This phase focuses on the client registration protocol and the smart card registration protocol. The smart card is referred to as a DVcard. This protocol involves registering the user Alice in the system to enable her to perform offline CBDC payment operations using a certified public key. Alice opens her DigiVault app (DV app) by authenticating biometrically and generates an asymmetric key pair (Prk, Puk). She transmits the public key to the server, which checks if this public key is not yet registered. If not, the server proceeds to register it and initializes Alice's online account to 0; otherwise, the registration fails. Using its private key, the server then generates the digital certificate for the public key (Puk_{cert}). Finally, it returns the certificate to Alice, confirming the opening of her online account.

Next, Alice proceeds to register her smart card (specifically Alice's secure processing environment) on the server. For this, on her DV app, she enters the original PIN associated with the DVcard and brings the DVcard to the NFC Reader of the mobile phone. She initializes the card by generating an asymmetric key pair for the DVcard ($\text{DVpk}_a, \text{DVsk}_a$) in the secure processing environment (SPE). This step resets the offline balance and the nonce value. Then the master secret key embedded in the SPE signs the public key so that when the output obtained is sent to the server, the latter can ensure that the public key of the card indeed comes from the DVCard. Thus, the DVcard sends a registration request for DVpk_a to the server containing the user Alice's public key (Puk_a), the public

key of the DVcard, and the proof of origin. The server will first check if puk (i.e., Alice) is registered and if Puk_a is not already associated with a certified public key of a smart card. If not, it will generate a certificate for DVpk_a, otherwise it will cancel the request. It then returns the certificate certDVpk_a to user Alice along with a pair of keys, a proving key JK and verification ZK, for verification. Finally, the server updates its database by associating Puk_a with certDVpk_a. Then Alice, through the DigiVault app, sends a command to the DVcard to trigger the certificate verification function. This uses the server's authentication key pre-installed in the card.

5.2.3.2 Redemption protocol

Alice wishes to load x coins with a blind signature onto her DVcard from her online account for an offline CBDC payment from her offline wallet.

1. Alice opens her DigiVault app on her mobile phone and generates blind coins, which are cryptographic representations of currency "blinded" by introducing a blinding factor to mask essential information and preserve confidentiality.

2. She enters her PIN code for the DigiVault app and brings the DVcard close to the NFC reader.

3. Her mobile phone creates a digitally signed request containing the blinded coins along with encrypted hashes, which is then sent to Alice's bank for signature through a secure channel.

4. Alice's bank locks the corresponding amounts on the deposit account to ensure the funds are available. The bank forwards the blinded coins to the central bank for processing, routing them through a mix network.

5. The central bank debits the reserve account of the bank and digitally signs the blinded coins.

6. The coins are returned to Alice's bank, which in turn forwards the newly minted, signed coins to Alice.

7. Alice unblinds the coins using her DigiVault card. She brings her DVcard close to the mobile phone's NFC reader, and the DigiVault app transmits the updated balance coin to the DVcard.

8. The DVcard updates its balance with the newly received coin value. It is noteworthy that the coin possesses USO (Unforgeable, Stateful, Oblivious) features.

5.2.3.3 Offline payment protocol

The following steps occur when Alice wishes to transfer an amount of x coins to Bob in offline mode:

1. **Payment Contract Negotiation.** Bob initiates the payment by selecting the amount to pay on his DV app and brings his card close to his mobile phone.

2. **Reading and Initiating Payment.** The DVcard reads the amount and initiates a payment request via its DV app, specifying the amount to be paid.

3. **Creation of Payment Request.** The DVcard reads the selected amount for Bob and creates a digitally signed payment request, which is sent to Bob's mobile phone.

4. **Exchange Between Phones.** Alice and Bob place their phones side by side, and the signed payment request is transmitted to Alice's phone.

5. **Authorization by Alice.** Alice views the payment request (the contract) and authorizes it by entering her PIN and swiping her DVcard on the mobile phone. The mobile phone transmits the request signed by Bob.

6. **Verification and Signing by the Card.** The DVcard verifies Alice's PIN, subtracts the e-coin corresponding to the requested amount, and signs the payment contract using the electronic coin.

7. **Fraud Analysis.** The transaction is analyzed using a Markov chain to assess whether the payment amount corresponds to a fraudulent scenario.

8. **Proof of Payment Generation.** The mobile phone generates a zk-SNARK proof of payment through a function that takes as inputs the DVcard's private key, the electronic coin, and the proving key. The full proof generation process is orchestrated by Alice's mobile phone.

9. **Encryption of the Transaction.** The DVcard encrypts the transaction with its private key DVska and returns it to the mobile phone. The encrypted payload includes: the zk-SNARK proof of payment, the signed contract, the card's public key, the card's certified key, the signed coin and nonce, and the key verification elements.

10. **Final Exchange Between Phones.** Alice and Bob place their phones side by side once again, and the fully signed contract by both parties is transmitted to Bob.

Table 5.1 and Table 5.2 present, respectively, the description of the variables and functions used for the offline payment process in Algorithm 6.

Table 5.1: Key Variables and Their Descriptions

Variable	**Description**
Dvsk_a	Private key for Alice's Dvcard
Dvpk_a	Public key for Alice's Dvcard
Dvsk_b	Private key for Bob's Dvcard
Dvpk_b	Public key for Bob's Dvcard
$\text{certDv}_{\text{pk}_a}$	Certificate of Dvpk_a
$\text{certDv}_{\text{pk}_b}$	Certificate of Dvpk_b
s	Unblinded signed coin
π	zk-proof
n	Nonce
Jk, Zk	zk-SNARK keys pair
ρ	Signed contract
τ	Encrypted transaction

Algorithm 6 Offline Payment Process

Initiate Payment from Bob's side:
  $s \leftarrow \Theta(\text{amount})$ ▷ Select appropriate token(s)
  $\text{payment_request} \leftarrow \Gamma(\text{amount}, s, \text{certDv}_{\text{pk}_b})$ ▷ Create Payment Request
  $\Delta(\text{payment_request}, s, \text{Alice}_{\text{mobile}})$ ▷ Send Payment Request to Alice's mobile
if $\Sigma(\text{Alice}_{\text{mobile}}, \text{pin}, s)$ **then** ▷ Alice authorizes payment
  $\Omega(\text{Dvcard}_a, \text{payment_request}, s)$ ▷ Process Payment on Alice's Dvcard
end if

Create Payment Request:
  $\text{sign_request} \leftarrow \text{SignRequest}(\text{amount}, s, \text{certDv}_{\text{pk}_b})$
  return sign_request

Send Payment Request:
  $\text{NFC_Transfer}(\text{request}, s, \text{Alice}_{\text{mobile}})$

Alice Authorizes Payment:
  $\text{AuthenticateUser}(\text{Alice}_{\text{mobile}}, \text{pin}, s)$
  return true ▷ Assume user is authenticated successfully

Process Payment on Alice's Device:
if $\text{MarkovChainAnalysis}(\text{Dvcard}_a.\text{transactionhistory}, \text{request.amount}, s) == \text{false}$ **then**
  AbortTransaction()
end if
$\pi \leftarrow \Xi(\text{Dvcard}_a, \text{request.amount}, s, Jk)$ ▷ Generate Zk Proof
$\tau \leftarrow \Lambda(\pi, \text{Dvsk}_a, \rho, \text{certDv}_{\text{pk}_a})$ ▷ Encrypt Transaction
$\Upsilon(\tau, s, \text{Dvpk}_\text{b}, \text{Bob}_{\text{mobile}})$ ▷ Transmit Contract back to Bob

Table 5.2: Functions Utilized in the Payment Protocol

Function	Description
Φ	Initiate Payment
Θ	Select Token
Γ	Create Payment Request
Δ	Send Payment Request
Σ	Alice Authorizes Payment
Ω	Process Payment
Ξ	Generate zk-SNARK Proof
Λ	Encrypt Transaction
Υ	Transmit Contract

5.2.3.4 Contract checking

The following steps occur when Bob wishes to verify the transaction in offline mode:

1. Bob accepts the payment and enters his PIN for his DVcard, then swipes the card on the mobile phone.

2. The mobile phone forwards the PIN and the payment request to the DVcard.

3. The DVcard checks the PIN and various signatures (the DVcard's public key, the authenticity of the coin, the transaction recipient). Bob's DVcard records the transaction history, which contains the inputs, outputs, nonce, the hash of the zk-SNARK proof, and the issuer's identifier.

4. The proof verification is performed on the mobile phone using the verification key. The resulting hash is transmitted to Bob's DVcard. If any check fails, the transaction fails. In other words: the DVcard's public key is not certified, or the recipient is not legitimate, or the proof is not associated with the recipient, or the nonce or zk-SNARK proof hash appears twice in the transaction history.

5. If all verifications pass, the transaction is validated and associated with the issuer's public key.

6. The DVcard increases its balance by the amount of the payment request.

7. The DVcard reports that the transaction was successful to Bob's mobile phone.

8. Bob's mobile phone transmits the receipt to the mobile network, which then forwards it to Alice's DVcard.

Fig. 5.1 shows the details of the registration process and Fig. 5.2 presents the offline transaction block in the DigiVault system. Fig. 5.2 also illustrates two blocks, both encrypted with Bob's public key. The first block contains transaction data, which Alice encrypts with her private key. The second block contains her public key certificate, issued by a trusted authority. The certificate proves Alice's identity and her participation in the system.

Alice's private key allows Bob to decrypt the transaction data block with his corresponding public key. Bob's public key encrypts the entire transaction. Only Bob can decrypt it, ensuring he is the authorized recipient. In the encrypted transaction data

block, there are inputs for the tokens Alice spends and outputs for the tokens Bob receives. These ensure a precise accounting balance. The Signed Coin field identifies the specific monetary unit transferred to Bob. It is blindly signed by the bank, making it tamper-proof.

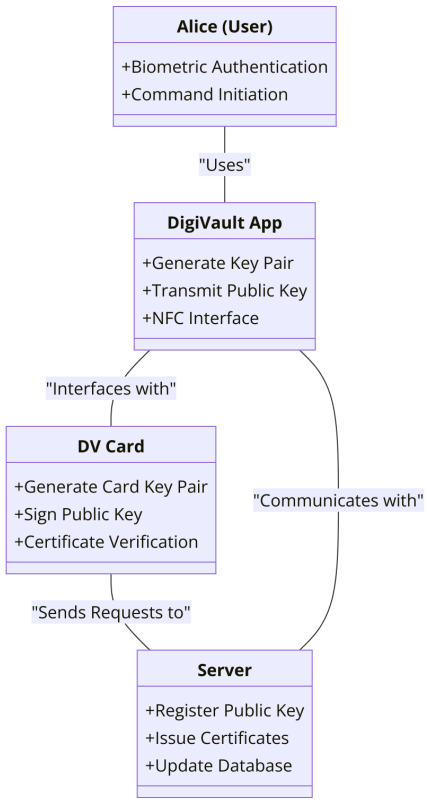


Figure 5.1: Onboarding Registration Phase

.

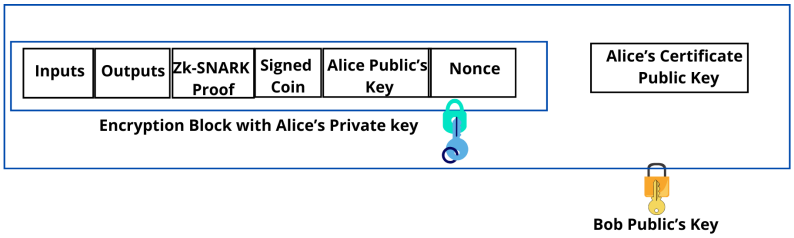


Figure 5.2: Offline Transaction block between Alice and Bob

.

5.2.3.5 Coin status transition

Once Alice's DVcard receives the receipt, the token used in inputs acquires the status of "spent token." The tokens issued for the recipient transition to the status of "unspent token." During settlement transactions, the value of spent tokens, labeled as inputs matches the value of unspent tokens labeled as outputs. User balances are calculated by aggregating all unspent tokens owned at any given time.

5.2.3.6 Coin Deposit and Exchange

The deposit or exchange operation, which triggers the synchronization procedure, is performed as follows:

1. The end user Carole chooses to manage their card via the DigiVault App, enters the PIN for their card, and brings it close to the mobile phone.
2. If the card has saved offline transactions, they are sent to the mobile phone.
3. The mobile forwards the transaction to the bank database through a negotiated contract. The bank performs the same verifications as Bob did when receiving a CBDC payment.
4. Carole taps her DigiVault card against her smartphone's NFC reader. This triggers the phone's DigiVault App to wirelessly transmit the card's updated balance information.
5. The card securely receives the updated balance data from the app. It then performs two actions: firstly, the card's internal balance is adjusted to reflect the new value; secondly, any previously stored offline transaction data is cleared, and the offline transaction counter is reset to zero.

5.2.4 MarkoPayChain system introduction

A markov chain model, named MarkoPayChain, is proposed for detecting fraud in offline CBDC transactions within a hybrid environment.

5.2.4.1 Mathematical Modeling Approach

The proposed approach initially involves defining the different states of the model based on transaction data, such as detection of small amounts, large amounts, transaction frequencies, etc.

Then, proceed to calculate the probabilities of transactions between these states based on the parameters of the transaction history. Finally, it will be a matter of using these probabilities to predict the probability that a future transaction will be fraudulent based on observed transitions. Thus, the proposed mathematical model is as follows: Let a Markov chain M_k define for an offline payment model, there exists a state x in M_k representing a specific configuration of the system at a given time defined by $\Lambda = (\alpha, \beta)$ where the state space, $\alpha \in \{\text{low, medium, high}\}$ describes the category of transaction amount, $\beta \in \{0, 1\}$ denotes the fraud status (Non-fraudulent (0) or fraudulent (1)). M_k represents a transition matrix $P = (p_{ij})_{(i,j)\in\Lambda^2}$ describing the transition probabilities from one state i to another j over a period of time. Considering S as the set of all possible states, $\forall i \in \Lambda, \sum _j \in \Lambda p_ij = 1$.

5.2.4.2 Model Validation

We validate our model using a synthetic dataset. The MarkoPayChain model simulates real-world transaction patterns, including potential fraud cases. We assume the synthetic data was generated securely. First, transaction amounts are uniformly sampled between \$5 and \$500, reflecting typical retail ranges. Second, each user is assigned 50 transactions over a random period of up to 365 days. A timestamp is generated for each transaction

to mimic varying frequencies. Third, 5% of transactions are labeled as fraudulent. Fraud is triggered by high transaction frequencies in short intervals or by amounts greatly exceeding a user's historical average. The synthetic dataset aligns with real-world patterns, referencing publicly available statistics. Fig. 5.3 presents the transition matrix, showing likely transitions and highlighting suspicious ones suggestive of fraud.

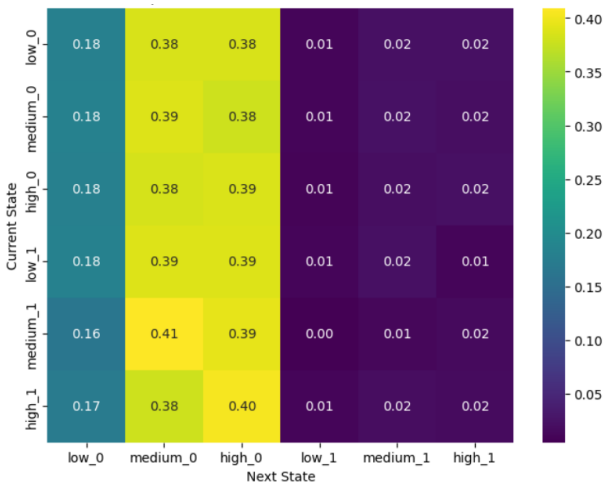


Figure 5.3: Heatmap of markov chain transition probabilities

5.2.4.3 Interpretation of the Transition Matrix

The heatmap illustrates the transition matrix of the proposed Markov model, where each row represents a starting state and each column an ending state. The values indicate the probability of transitioning from one state to another, with colors ranging from blue (low probabilities) to yellow (high probabilities). States are defined by two factors: transaction amount category ("low", "medium", "high") and fraud status ("0" for non-fraudulent, "1" for fraudulent).

5.3 Performance Evaluation

The quantitative evaluation of our solution will be carried out in two stages. First, we will evaluate our Markov chain-based fraud detection system in isolation. Secondly, we will evaluate the offline payment phase.

5.3.1 MarkoPayChain Simulation

5.3.1.1 Test Scenario

The proposed test scenario consisted of establishing a baseline behavior that corresponds to normal transactions. For this, it involved conducting a series of small and medium non-fraudulent transactions. Subsequently, there was a trigger for suspicious activity materialized by a sudden shift to high-value transactions that could potentially be fraudulent. Thus, for the baseline behavior we have Sequence 1: `['low_0', 'low_0', 'medium_0', 'medium_0', 'low_0']`. Then we introduce the suspicious behavior with Sequence 2: `['high_1', 'high_1', 'medium_1', 'high_1']`.

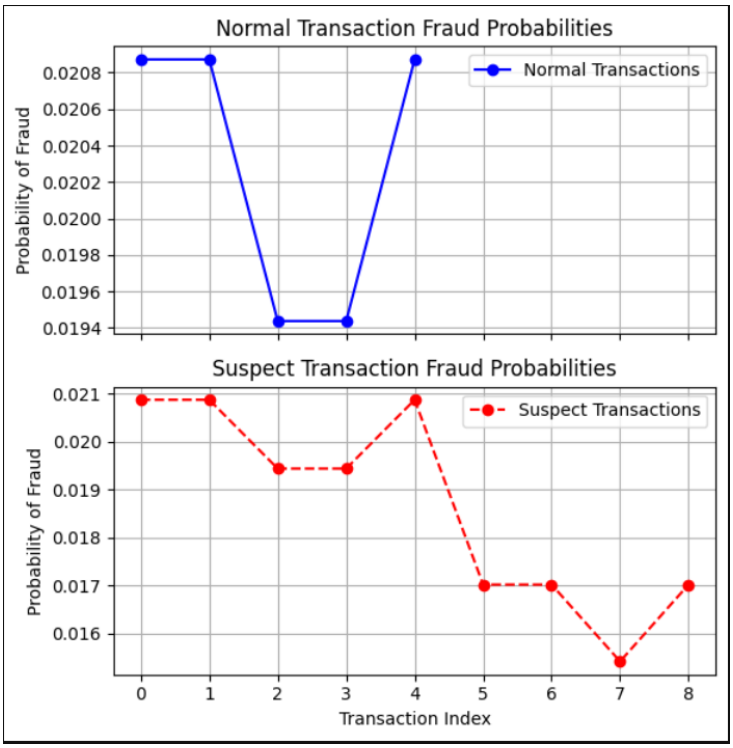


Figure 5.4: Fraud Detection Analysis for Offline CBDC Payments

5.3.1.2 Results analysis

The low fraud probabilities in the graph of normal transactions indicate a consistent low risk for typical legitimate activities. In Fig. 5.4, the slightly higher minimum fraud probability for suspicious transactions (0.016) compared to normal transactions (0.0194), despite different transaction values, is explained by the Markov model's probabilistic nature. Suspicious transactions are classified based on abrupt deviations, such as high amounts or frequencies, which inherently raise their base fraud probability even without confirmed fraud. In contrast, normal transactions, following predictable low risk patterns, result in lower fraud probabilities, reducing false positives, and enhancing model reliability.

5.3.1.3 MarkoPayChain Model and Hidden Markov Models for Fraud Detection

Hidden Markov Models (HMMs) have long been used for fraud detection, particularly in internet-connected environments [109]. While both MarkoPayChain and HMMs detect fraudulent behavior, they differ in approach and data handling. MarkoPayChain relies solely on observable transaction characteristics, such as amount and fraud status, making it computationally lightweight and suitable for real-time offline fraud detection. In contrast, HMMs incorporate hidden states representing latent factors, requiring more intensive computations due to simultaneous estimation of hidden and observable states. Thus, MarkoPayChain is better suited for resource-constrained offline environments.

5.3.2 Offline Payment Simulation

5.3.2.1 Experimental Setup

To evaluate our solution, we opted to use the CBDC-SIM tool[110]. CBDC-Sim is an open-source simulation tool designed to model Central Bank Digital Currency (CBDC) ecosystems. It provides functionalities for simulating transactions between agents (households, firms, central banks) and analyzing metrics such as throughput, latency, and settlement behavior.

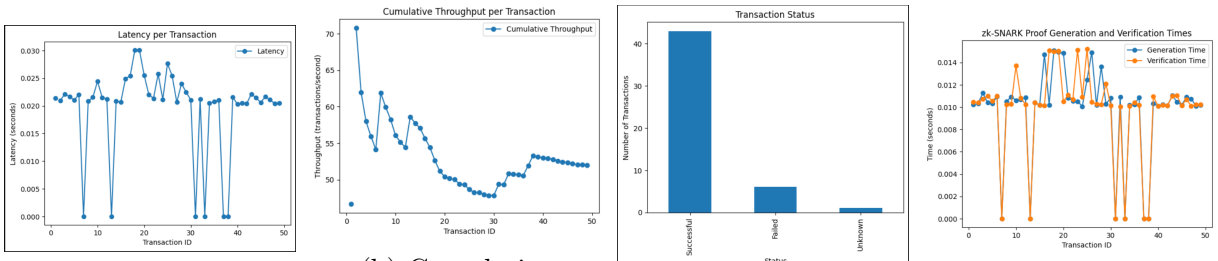
5.3.2.2 Experimental Procedure

The aim is to assess the performance of the modified CBDC-Sim framework in an offline payment setting using the following metrics: Throughput (transactions per second), Latency (time taken per transaction), zk-SNARK proof generation and verification times, Fraud detection time (using Markov Chains), Success and failure rates of transactions. The results obtained with our solution are shown in Table 5.3.

Table 5.3: Comparison of Metrics Across 50, 75, and 100 Transactions

Metric	**50 Transactions**	**75 Transactions**	**100 Transactions**
Latency	0.010–0.025 seconds	~0.020 seconds (stable)	0.010–0.030 seconds
Throughput	Stabilizes at ~50 tps	Stabilizes at ~50 tps	Stabilizes at ~45 tps
Success Rate	~90%	~90%	~88%
zk-SNARK Generation	Stable at ~0.011 seconds	Stable at ~0.011 seconds	Stable at ~0.011 seconds
zk-SNARK Verification	Fluctuates (up to 0.015)	Reduced variability	More frequent fluctuations

Latency per Transaction. For, 50 Transactions, the latency varies between 0.010 and 0.025 seconds. Occasional drops reach near zero. See Fig. 5.5a. The latency remains stable around 0.020 seconds for 75 Transactions. Fewer extreme drops occur. See Fig. 5.6a. Finally, for 100 transactions, the latency stays stable overall. More frequent spikes and drops appear. Peak latency reaches 0.030 seconds. Lowest latency approaches zero. See Fig. 5.7a



(a) Latency per Transaction

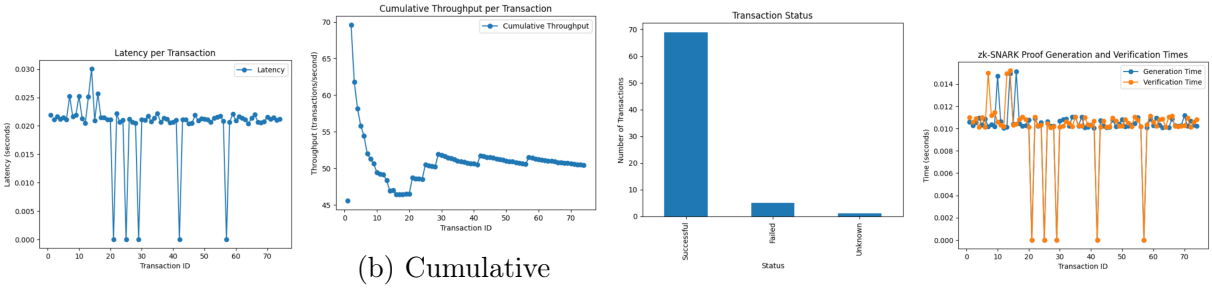
(b) Cumulative Throughput per Transaction

(c) Transaction Status Distribution

(d) zk-SNARK Proof Times

Figure 5.5: Results for the simulation with 50 transactions.

Cumulative Throughput per Transaction. For 50 Transactions, the throughput starts at 70 transactions/second. It declines rapidly. It stabilizes at 50 transactions/second after transaction 30. See Fig. 5.5b. The Throughput follows a similar pattern for 75 transactions. It stabilizes earlier at 50 transactions/second after transaction 20. See Fig. 5.6b. Finally, for 100 Transactions, the throughput declines gradually. It stabilizes at 45 transactions/second after transaction 50. See Fig. 5.7b.



(a) Latency per transaction

(b) Cumulative throughput per transaction

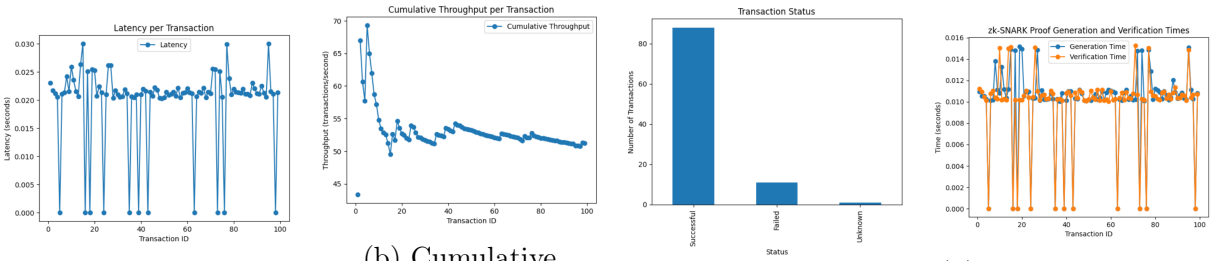
(c) Transaction status distribution

(d) zk-SNARK proof times

Figure 5.6: Results for the simulation with 75 transactions.

Transaction Status Distribution. For 50 Transactions, about $\sim 90\%$ of transactions were successful. However, about $\sim 8\%$ failed. Finally, about $\sim 2\%$ were unknown. Refer to Fig. 5.5c. The distribution resembled the 50-transaction scenario with 75 Transactions. Most transactions succeeded. Few were failed or unknown. Refer to Fig. 5.6c. For, 100 Transactions, about $\sim 88\%$ of transactions were successful while about $\sim 10\%$ failed and about $\sim 2\%$ were unknown. The slight increase in failures likely resulted from higher complexity. Refer to Fig. 5.7c.

zk-SNARK Proof Generation and Verification Times. The proof generation times remained stable at 0.011 seconds for 50 Transactions and verification times spiked occasionally to 0.015 seconds. Refer to Fig. 5.5d. For 75 Transactions, the proof generation times stayed consistent. Verification times showed fewer fluctuations compared to 50 transactions. Refer to Fig. 5.6d. Finally, for 100 transactions, the proof generation times stayed at 0.011 seconds. Verification times fluctuated between 0 and 0.015 seconds. These fluctuations suggest higher computational demand. Refer to Fig. 5.7d



(a) Latency per Transaction

(b) Cumulative Throughput per Transaction

(c) Transaction Status Distribution

(d) zk-SNARK Proof Times

Figure 5.7: Results for the simulation with 75 transactions. Subfigures:

5.3.2.3 Evaluation summary and Comparative Study

Fig. 5.5 shows the system's performance for 50 offline transactions. Latency fluctuates between 0.010 and 0.025 seconds. Occasional drops near zero appear due to lower cryptographic complexity. Throughput stabilizes at 50 transactions per second after 30 transactions. The $\sim 90\%$ success rate confirms reliability. Proof generation remains steady at 0.011 seconds. Verification occasionally peaks at 0.015 seconds.

Fig. 5.6 shows the system's performance for 75 transactions. Latency stabilizes around 0.020 seconds. Fewer sharp drops indicate better handling of intermediate loads. Throughput settles at 50 transactions per second. The $\sim 90\%$ success rate remains unchanged.

Failed transactions stay minimal. Proof generation remains steady. Verification times show reduced variability, reflecting improved efficiency.

Fig. 5.7 shows the system's performance for 100 transactions. Latency remains stable but peaks more frequently, reaching 0.030 seconds. Throughput declines gradually and stabilizes at 45 transactions per second after 50 transactions. The success rate drops slightly to $\sim 88\%$. Failed transactions rise to $\sim 10\%$. Proof generation remains steady. Verification times fluctuate more often, reflecting increased cryptographic demand.

Table 5.4 summarizes a comparative study of offline payment solutions in terms of security, privacy, and scalability. Our solution offers the best compromise among these criteria.

Table 5.4: Proposed Solution Comparison

Offline Solutions	**Double Spending**	**Additional Hardware**	**P.E.T**	**Scalability**
[107]	No	No	No	50 tr in 15 ms; 75 tr in 19 ms; 100 tr in 22.5 ms
[52]	No	Yes	No	Sending 100 tokens takes 39 s (2.5 TPS)
[66]	No	Yes	No	After 50 parts: median time = 548 ms (91.24 TPS)
[44]	No	Yes	No	Token of 64 chars encoded with UTF-32, delivered in 0.0047 s
Our Solution	No	No	Yes	50 tr: max 70 TPS, min 50 TPS; 75 tr: min 50 TPS; 100 tr: min 45 TPS

5.4 Discussion

The implementation of an offline Central Bank Digital Currency (CBDC) payment solution often requires making operational technological choices, particularly to address challenges related to security, privacy, performance, and user experience. In terms of security, preventing theft and double spending is critical. While the security modules integrated into modern mobile phones are effective, they cannot be fully trusted in adversarial environments, especially under offline constraints. This issue is compounded by the need for broad accessibility and financial inclusion, as most integrated security modules are proprietary solutions [111]. Digitally serves as a case in point. The DigiTally solution relies on a specialized SIM card (overlay SIM) to securely execute offline payment operations. However, this dependency on a modified SIM poses a significant barrier to widespread adoption and deployment, as it requires physical and software modifications that are not natively supported by most phones or operators [112].

This security dilemma encourages consideration of two types of approaches. The first approach involves completely abstracting away from trusted hardware in the design. The OVERDRAFT protocol is illustrative in this regard, particularly in its approach to preventing double spending. However, relying on node reputation means that fraud detection remains probabilistic rather than deterministic [113]. A key contribution of this work lies in its proactive treatment of double-spending and fraud risks. Unlike many offline payment solutions that rely on reputation scores, post-settlement reconciliation, or punitive revocation mechanisms [66], the proposed offline solution, integrated with the MarkoPayChain model, enables real-time, local fraud assessment prior to transaction completion. Nevertheless, the reliance on synthetic data for validating MarkoPayChain constitutes a significant limitation. Although the dataset was designed to reflect realistic transaction patterns, real world fraud behavior is often more complex, adaptive, and context-dependent.

The second approach involves the strict use of payment cards for offline payments. These generally offer advanced security for securing funds [51]. However, they suffer from computational limitations that prevent the deployment of advanced privacy-enhancing technologies (PETs) such as zk-SNARKs, particularly when generating proofs. From a privacy perspective, the combined use of blind signatures and zk-SNARKs in our solution significantly strengthens selective traceability (through certified keys and verifiable proofs) and transaction confidentiality compared to traditional offline payment schemes. Beyond its ability to provide a pragmatic middle ground that is compatible with Anti-Money Laundering (AML) and Counter-Terrorist Financing (CFT) requirements, the proposed DigiVault/MarkoPayChain architecture offers operational feasibility. By combining smart cards and mobile devices in a hybrid architecture, the solution explicitly acknowledges the limitations of purely mobile-based offline payments while avoiding the rigidity of fully hardware-centric approaches.

5.5 Conclusion

In conclusion, the work presented in this chapter has shown that hybridising the DigiVault Card with a mobile terminal constitutes a credible hardware architecture for secure key storage and orchestration of offline payments. Functionally, it integrates blind signatures, zk-SNARKs, digital signatures and the MarkoPayChain model. By analysing these various protocol mechanisms, the chapter established that this model can achieve the privacy, security and proactive fraud-detection objectives outlined in the introduction. Beyond that, these results have several implications. First, they illustrate the decisive role of the smart card in securing the offline wallet. Second, they demonstrate the cryptographic computational capabilities of the mobile terminal to compensate for the constraints of the smart card. Finally, they underscore the importance of incorporating fraud-detection mechanisms suitable for disconnected environments. Even with a robust protocol, fraud remains a critical dimension that requires thorough analysis and advanced complementary tools. The following chapter focuses on offline payements enabled by distributed ledger.

5.6 Chapter Summary

This chapter presents a complete solution for CBDC offline payments applied to a single-ledger infrastructure. It relies on a secure card (DigiVault Card) and a smartphone. The card manages keys and secures balances, while the phone handles intensive operational computation such as generating cryptographic proofs. Operationally, the protocol is built upon three main technologies: blind signatures for privacy during value withdrawal; zk-SNARKs to validate transactions with the recipient without revealing sensitive information; and an embedded Markov chain model (MarkoPayChain) for proactive fraud pattern detection. Moreover, the chapter fully outlines all phases of the protocol from onboarding, withdrawal, offline payment, recipient verification, to synchronization with the banking system. Experimentally, the feasibility of a secure offline payment is validated: the solution achieves a success rate above 88 %, a latency in the range of 10–30 ms and proof generation times around 11 ms. These results confirm the operational viability of secure offline payments.

Chapter 6

Offline Payments Enabled by Distributed Ledger: Block-PAD Framework

6.1 Introduction

Following the introduction of the two frameworks that orchestrate CBDC offline transactions using traditional infrastructure, it becomes essential to examine concrete mechanisms for achieving secure offline payments oriented around a distributed ledger-enabled infrastructure. The previously presented framework offers a theoretical, cryptographic and operational structure but is not capable of simultaneously managing the constraints of hybrid offline payment, contextual privacy, interoperability and security. Therefore, this chapter aspires to guarantee these properties at a significantly higher level by integrating an additional infrastructure.

The goal of this chapter is to introduce Block-PAD, a blockchain-enabled framework designed to ensure resilience, flexibility and security for CBDC transactions in autonomous environments. This chapter highlights the theoretical and practical motivations underlying the design of Block-PAD. In particular, it demonstrates how combining distributed ledgers, digital identities and deferred synchronization can guarantee balance integrity while preserving user experience. It thus represents the most developed system in this thesis to address the primary challenges of offline payment.

To achieve this goal, the chapter begins with a background section that situates prior work and identifies gaps in existing solutions. It then presents the overall architecture of Block-PAD along with the cryptographic and distributed principles it relies upon. The discussion continues with identity management, local circulation of value units, and mechanisms for synchronizing with the global chain. Subsequently, it addresses the security properties ensured by the proposed framework. Finally, the chapter concludes with a performance evaluation.

The remainder of this chapter is organized as follows: Section 6.2 presents the background; 6.3 discusses related work; Section 6.4 delves into the proposed model; Section 6.5 illustrates system operations; Section 6.6 covers security analysis; Section 6.7 presents performance evaluation; Section 6.8 presents the discussion; Section 6.9 concludes the chapter and Section 6.10 summarizes it.

6.2 Background

The interest in Central Bank Digital Currencies (CBDC) is gauged by the growing number of projects and research initiatives initiated by governments, academic institutions, and private sector actors to make them operational. This increased attractiveness for electronic money issued and controlled by central banks follows the evolutionary curve reflecting the trend of preferential options for digital payment observed among users over the last decade. Yet, despite this fervor and widespread research, very few projects have entered the production phase [72]. Indeed, for CBDCs to be a credible payment alternative, it is crucial that they offer advantageous features and correct the pitfalls of traditional payment systems, all without undermining the monetary policy of central banks. In this vein, public and private service actors must address a range of technical, legal, economic, and social challenges that are evidently interconnected. This research focuses on technical aspects in harmony with its legislative ramifications, notably on privacy protection, security, and resilience orchestrated by offline payment without infringing legal constraints.

6.2.1 Securing Privacy of Offline Payments in a Decentralized World

The emergence of cryptocurrencies following the economic crisis of 2008 highlighted the potentials and characteristics of blockchain technology. The application fields of blockchain are so varied that it can justly be qualified as a technological revolution in the domain of computer science, on par with artificial intelligence. Yet, like AI, it does not escape concerns related to privacy protection and security, especially when it comes to crypto payments, which additionally bear the risk of illicit use oriented towards money laundering or terrorism financing. Moreover, offline payments are intended to increase resilience in case of internet connection failure but do not escape the aforementioned problems. Subsequently, it is incumbent to establish a new paradigm that leverages the advantages of offline payment and blockchain while simultaneously guaranteeing privacy and regulatory compliance.

6.2.2 Blockchain and CBDC

Blockchain technology, due to its multiple desirable properties, is revolutionizing various sectors of everyday life, similar to the payment industry. With the advent of Bitcoin and other cryptocurrencies, it has highlighted the advantages and benefits of this type of distributed system for the payment industry. CBDC and stablecoins incorporate in their projects the use of a blockchain, whether public or private. However, the major problem facing CBDCs is inevitably privacy. Indeed, blockchains do not natively guarantee privacy. At least, not in the sense of being the opposite of transparency. At the same time, it is essential not to fall into the other extreme, which would consist of neglecting the need for regulatory compliance compatibility. Reference [43] proposed an offline CBDC payment scheme that ensures privacy. An intuitive approach, for instance, might involve conducting a consecutive offline payment with blockchain integration for defunding operations, using the recipient's public key for the transaction. The recipient could then connect to their public account and proceed with the defunding. This straightforward approach breaks the guarantee of privacy in offline mode and exposes users to the backchain problem. To overcome these pitfalls, it is therefore necessary to customize blockchain systems so that they meet the standards of regulatory.

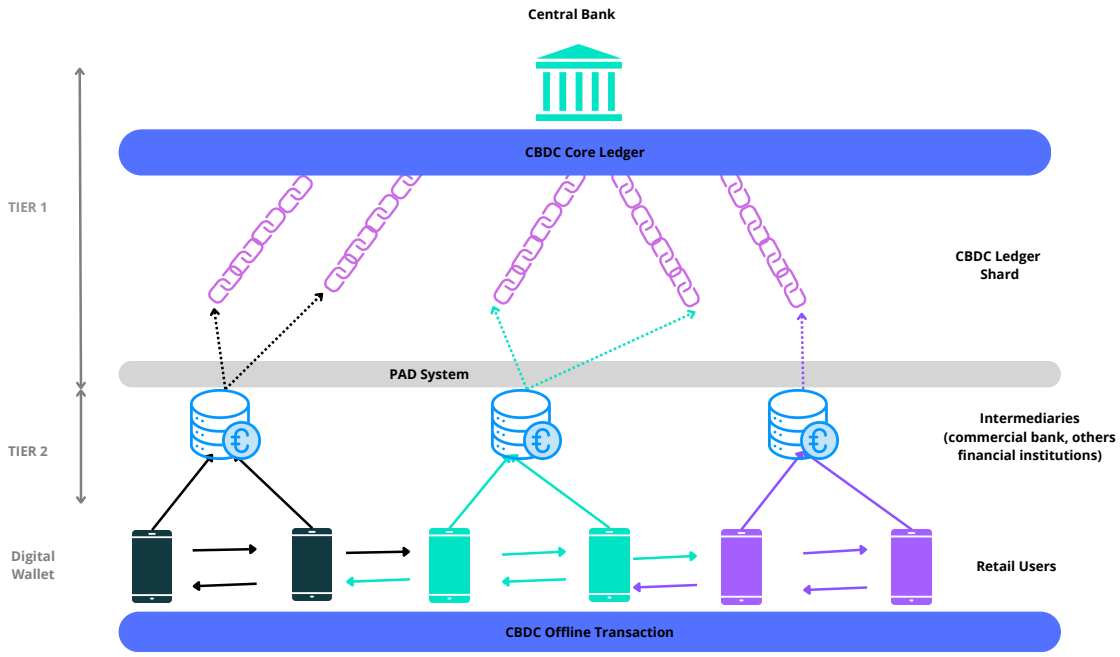


Figure 6.1: Architecture of transferable CBDC based PAD.

This work extends the research in [43] by relying on a two-tier architecture offering end-users an offline payment solution presented in Fig. 6.1. The core novelty of this work is the operational integration of the Privacy Accountability Description (PAD) model into a CBDC transaction flow that supports consecutive offline transfers and on-chain reconciliation. Rather than presenting a mere collection of privacy-enhancing technologies, we demonstrate how PAD enables contextual privacy, i.e., deciding what, when, and to whom information should be disclosed under trustees/validators thresholds. This integration provides strong privacy guarantees while ensuring regulatory traceability and transferability of digital cash.

6.3 Related Work

6.3.1 Decentralized e-Cash Approaches

The protocol proposed in [70] stands out for its decentralized offline electronic cash system, offering compact and divisible schemes that enhance payment security and deposit verification efficiencies. It addresses scalability issues inherent in blockchain and privacy concerns in centralized payment infrastructure management, although its implementation complexity remains a challenge, especially in real world environments. Furthermore, the detection of double spending is not immediate but occurs posteriorly during the deposit phase, which introduces the potential for unidentified vulnerabilities due to the novel nature of the system.

On the other hand, [46] proposes an e-cash payment system free from specific storage systems, offering a secure transferability functionality. This system is more efficient and significantly more confidential than older approaches. However, it does not address regulatory compliance issues and only detects double spending at the time of deposit.

6.3.2 TEE-Based Offline Payment Solutions and Hardware wallet

Reference [39] introduces a secure offline payment system based on digital signatures and a Trusted Execution Environment (TEE). Although it integrates well with retail CBDC infrastructures, robust privacy is not assured, since the protocol primarily targets traditional banking designs. In a subsequent effort, [77] improves upon [39] by mitigating the risks of forgeability and DDoS attacks. Then [51] provides a smart card implementation of [39], incorporating [77]'s enhancements and opening the door to blockchain connectivity. However, none of these three works ([39], [77], [51]) address privacy and regulatory considerations comprehensively.

Moving to a more innovative perspective, [52] describes a system leveraging One-Time Programs (OTP) and reflective architectures for offline, asynchronous P2P CBDC transactions. It utilizes the TEE of smartphones, ensuring a high degree of privacy and programmability. Still, a malicious wallet app could compromise OTP executions by modifying or halting embedded libraries. Moreover, the protocol does not cover regulatory compliance aspects.

Reference [59] relies on eIDAS Advanced Electronic Signatures (AdES) and X.509 Attribute Certificates to handle offline trust verification for complex CBDC transactions. It supports both account-based and token-based models, aligns with ISO 20022 for Anti-Money Laundering (AML) compliance, and can manage intricate data structures. However, it does not integrate Privacy-Enhancing Technologies (PETs), offering limited confidentiality, and the protocol's reliance on evolving "middle phone" technologies may demand future updates.

Reference [37] proposes a universal hard wallet equipped with quantum-resistant bits for storing and transferring centrally minted CBDC units in a decentralized manner. Offering strong privacy by design, high security, and financial inclusion, it requires no synchronization and places no transaction limit. Despite its convenience, the approach does not explain how to ensure regulatory compliance in an offline context. As a result, scenarios such as large-scale acquisitions of these hardware wallets for illegitimate purposes remain a concern.

Reference [71] provides Integrated Circuit Card (ICC) specifications for payment systems, including asymmetric cryptography, digital certificates, and configurable payment limits. While this structure offers robust security and effective fraud risk management, privacy is only "soft," as users' identities are not strongly protected.

6.3.3 Compliant Privacy-Centric Model

Reference [82] presents a comprehensive approach to privacy and regulation within CBDC, combining cryptographic techniques with distributed system design. Although transactions are anonymous in principle, each transaction is tagged, allowing it to be traced back to the original holder. Furthermore, while the protocol accounts for malicious actors, it provides only limited consideration of coalition attacks by maintainers who could revoke a user's privacy by consensus.

In contrast, many smart contract-based privacy solutions, other than [114], revolve around a "privacy pool." These solutions use zk-SNARK proofs to demonstrate that transaction funds originate from honest users. While these privacy pools add a layer of confidentiality and help meet compliance requirements, they rely on social consensus (to define associations among users) and might struggle with regulatory evolution, thereby questioning the feasibility of true privacy by design in CBDC systems.

Additionally, [115] guarantees payment confidentiality on the blockchain using ZKPs and threshold homomorphic encryption for anonymous balance updates, adding identity authorities for user certification. Despite its decentralized regulatory control, this protocol does not fully protect user identity privacy and overlooks privacy breaches via coalition attacks.

[64] proposes a CBDC payment system oriented toward account-based transactions by leveraging commitments, nullifiers, and ZKPs. At its core is an unspent account state (UAS) that enables three different privacy levels in line with regulatory requirements: fully private, semi-private, and entirely transparent. While this design integrates digital identity and offers privacy by design, private wallets remain vulnerable to illicit acquisitions on the black market. [49] proposes a secure and efficient double transaction scheme for offline CBDC payments in centralized banking systems. Its combination of digital signature cryptography with Secure Elements (SE) and TEEs makes it reliable in terms of security, but less so for privacy sensitive data is recorded for regulatory compliance.

Similarly, [97] introduces a secure protocol leveraging digital signatures for consecutive CBDC payments, Tamper Resistant Element (TRE) and aiming to provide "contextual privacy" offline. However, synchronization remains a challenge, as public key certificates imply third party knowledge of user identities.

Reference [50] specifically targets mobile platforms with an anonymous offline payment solution (OAPM) based on SE+TEE hardware. This approach enables "controllable anonymity," allowing authorities to unmask the payer in case of illicit activities. Nevertheless, the payee's anonymity is not guaranteed, compliance management is tied to the currency issuer, and there is no property enabling transferability.

Reference [54] outlines a retail CBDC (rCBDC) payment protocol that functions offline for the point of sale (POS) but stays online for the buyer. Relying on a shared symmetric key and time-based one-time protocol, it ensures payer anonymity but not that of the payee, thus offering only partial offline capability.

Reference [43] serves as a robust and secure payment solution oriented toward privacy while also being compatible with regulatory compliance. It uses multiple PETs (including blind signatures and zk-SNARK) and ensures that synchronization does not reveal the origin of funds, though it predominantly focuses on transaction privacy. Building on [43], the present research aims to integrate this solution into a decentralized system to guarantee end-to-end contextual privacy for both issuer and receiver, fulfilling regulatory requirements. Table 6.1 compares these approaches in detail.

6.3.4 Key building blocks

6.3.4.1 Blind Signature

Blind signature is a cryptographic process whereby an entity signs a message without knowing its content. In the context of electronic payment, Chaum initially proposed a first version (ecash1), which he later improved. This protocol aims to guarantee the payer's privacy by breaking the link between the identity of the requester who asked the bank to sign the coin and the transactions that they will carry out afterward. In this paper, we adopt the scheme proposed in ecash2 [57]: Let $f(x)$ be the hash function representing the monetary value issued by the central bank for a coin. Alice, a user of the system, generates a random number b via her device. She then applies a signature using fractional powers according to the calculation:

$$(f(x)b)^n$$

Table 6.1: Comparative Study of Various Solutions

Proposed solution	Wallet	Offline function	Privacy Type	Additional hardware	Digital Identity	DLT Integration	P.E.T.*
[70]	Software	Yes	Strong Privacy	No	No	Yes	No
[82];[114]	Software	No	Strong Privacy	No	No	Yes	Yes
[115]	Software	No	Confidential Transaction	No	No	Yes	Yes
[64]	Software	No	Controllable Privacy	No	Yes	No	Yes
[39]	Software	Yes	Soft Privacy	SE+TEE	No	No	No
[51]	Smart Card	Yes	Soft Privacy	No	No	Possible	No
[52]	Software	Yes	Data Privacy	SE+TEE	No	No	No
[59]	Software	Yes	Medium Privacy	No	Yes	No	No
[37]	Hardware	Yes	Strong Privacy	No	No	No	No
[46]	Software	Yes	Data Privacy	No	No	No	No
[49]	Software	Yes	Soft Privacy	SE+TEE	No	No	No
[97]	Software	Yes	Contextual Privacy	TRE	No	Possible	No
[50]	Software	No	Controllable Privacy	SE+TEE	No	No	No
[54]	Software	Yes	Semi-Private	No	No	No	No
[71]	Software/Hardware	Yes	Soft Privacy	Must be specified	No	No	No
[43]	Software	Yes	Strong Privacy	SE+TEE	No	Yes	Yes
Our solution	Software	Yes	Contextual Privacy	SE+TEE	Yes	Yes	Yes

*P.E.T. = Privacy Enhancing Technologies.

where b is the blinding factor and n is the exponent associated with the coin's denomination. The coin's authenticity is guaranteed by its ability to perform fractional power calculations on a number, a task difficult for other entities to replicate. Thus, the bank receiving the blinded coin applies a signature of the form :

$$((f(x)b)^n)^{1/n} \mod C.$$

Then the coin is added to a mix network, a system designed to anonymize electronic communications by mixing cryptographic messages from different users, making it extremely difficult to trace the source or destination of a specific message, to an unspent coin list by the bank, which then returns the signed coin to Alice. She can then apply the unblinding via an operation that 'reverses' the effect of the blinding factor, possibly a division by b or an equivalent operation to cancel the effect of b^n and make the coin usable.

6.3.4.2 zk-SNARK

Zk-SNARK is a variant of zero-knowledge proof based on three algorithms: key security parameters, a proof generator, and a proof verifier [99]. The idea behind Zk-SNARK is to allow the Verifier to generate a public argument associated with a corresponding private argument known only to them, in order to verify a proof presented in the form of a polynomial for a given witness. For instance, if we need to prove the knowledge of solutions x and y for the equation $x^3 + y^4 = z$ without revealing x and y, we use our knowledge of x and y to construct a polynomial. The polynomial is the compilation of the initial calculation (often represented as an arithmetic circuit). We then create a cryptographic commitment based on this polynomial. In other words, the prover 'commits' to certain values without revealing them, often using an elliptic curve-based cryptosystem. We then provide proof that this commitment corresponds to a solution of the equation, without revealing the values of x and y. Finally, three conditions should be satisfied: completeness, succinctness, and soundness.

6.3.4.3 CKKS

Homomorphic encryption methods are cryptographic techniques for performing arithmetic operations on encrypted data without the need to decrypt them. In this vein, [116]proposes a homomorphic encryption based on an efficient approximate calculation process. This process uses not integer numbers as is done in many cryptographic processes for the calculation of asymmetrical keys, but rather integer degree polynomials. If $\mathbb{R}_q$ is defined as a sufficiently large polynomial space (the encrypted text space) with polynomials R_0, R_1, and R_2, it is possible to consider performing operations from R_1 to R_2 and obtaining R_3, another polynomial so that the operations performed from R_2 to R_1 correspond homomorphically to the underlying clear text operations. Specifically, for a palette of polynomials a, s, e, and b in $\mathbb{R}_q$ where q is the coefficient modulus, n the degree of the polynomial space, a and b being uniformly random and e, a random "small" one, one can complexify the calculation of secret key s (which has the coefficients ± 1 or 0) to part of the public key $(a, -a \cdot s + e)$. Consequently, even if a, s, and e are given, it is easy to calculate $a \cdot s + e$, but it is practically impossible to find s from a and $a \cdot s + e$. This public key is therefore used to encrypt a message m (to which a smaller prime number p is attached) so that the encryption message is in the form $C_m = (C_1, C_2)$ with $C_1 = (-a \cdot s + e) + R_0 + mp$ and $C_2 = aR_0 + R_2$.

6.3.4.4 Trusted Execution Environment

Plenty of devices today incorporate additional dedicated hardware elements to enhance the security of software applications, especially in terms of data integrity and confidentiality. This is the case with Trusted Execution Environments found in modern smartphones, which enable secure code execution and protect the data stored against various forms of attacks. These include counterfeiting, side-channel attacks, jailbreaking, roll-back attacks, man-in-the-middle, and replay attacks. While a variety of architectures are offered in the market, this research adopts the Global Platform standard that has standardized the TEE model [98]. In this paper, we will utilize this technology for the protection against the aforementioned attacks, for biometric access, key generation, and digital identity.

6.3.4.5 Wallet-based Digital Identity

Several initiatives have emerged to standardize the digital identity of individuals or organizations [117]. The frenzy for digital identity, theorized since the deployment of the internet and accelerated since the COVID-19 pandemic, is hardly surprising given the stakes in security, privacy, and accessibility to digital inclusion. Digital identity, which can be defined as the set of information available on the internet (including personal data, online behaviors, credentials, and digital access rights) associated with a person or an organization, can be used to authenticate and authorize individuals in online transactions, access services, or personalize user experiences. Among the fields of application for digital identity, wallets are particularly notable, especially in terms of security and fraud, as well as privacy and data control. While the concept of identity is generally agreed upon, its implementation is subject to caution, hence the emergence of various paradigms and implementation models. The paradigm proposed by EIDAS is widely explored in projects in Europe today, particularly concerning the EUDI wallet [84].

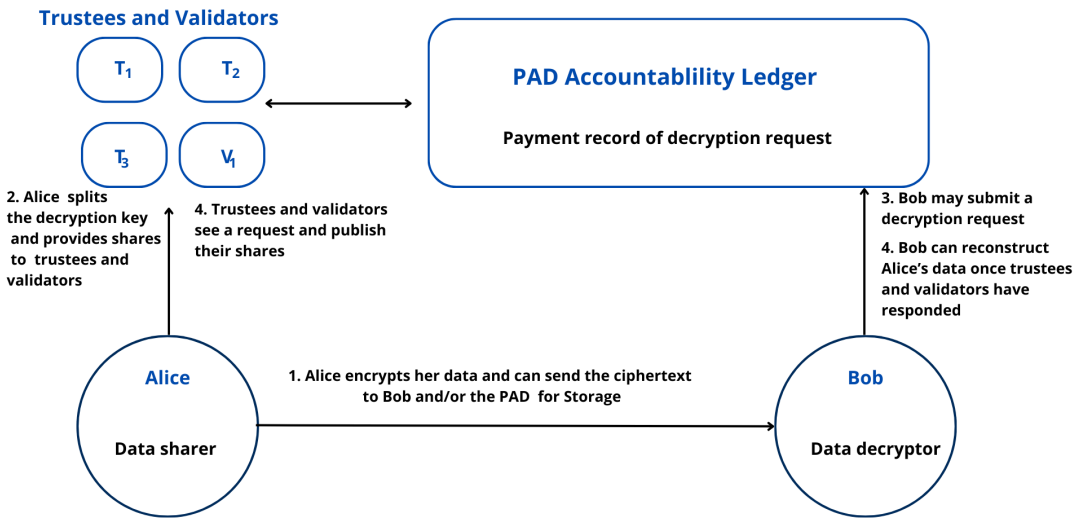


Figure 6.2: Simplified PAD schematic. See this link for more details.

6.3.4.6 PAD Model

The PAD model is a model of secure and private information flow based on the exchange and decryption of shared keys, as proposed by [118]. The use of PAD guarantees, on one hand, high availability, access, and integrity of information for a designated receiver without the aid of a trusted third party. On the other hand, it ensures that this information remains secure and private for any other non-designated entity. Finally, it guarantees transparency for all decryption operations and access to information in such a way that it is impossible for a dishonest user to deny being responsible for a malicious act.

The PAD system consists of three entities depicted in Fig. 6.2:

- One or more accountability ledgers with the properties of 'append-only', tamper-proof, and time-stamped.
- Several trustees who are third-party service providers whose role is to respond to valid decryption requests and publish messages in the accountability ledger that initiate decryption.
- Validators are service providers that ensure the maintenance and verification of the ledger.

For a secret that Alice wants to share with Bob, Alice first encrypts the secret with a symmetric key known only to her. Then she uses a cryptographic secret sharing scheme to fragment the secret key and distribute each encrypted part to the trustees and validators. It is impossible to decrypt the information related to the key unless there is a broad coalition of trustees.

To access Alice's secret, Bob can publish a request in the accountability ledger, to which the trustees and validators will respond with a decryption of the shared secret, also posted in the ledger. [119] proposed integrating PAD into blockchain to offer contextual privacy for CBDC payments. Our proposed solution is based on this intuition.

6.4 The Block-PAD Model

6.4.1 Security Model

Transaction system composition: The system involves three entities. Noticeably, A_{device}, a secure device operated by Alice (payer), B_{device}, a secure device operated by Bob (payee), and C_{account}, a pseudonymous account registered on the blockchain and associated with Carla (payee without a device).

CBDC architecture: The proposed system is based on a two-tier architecture. This involves the participation of commercial bank intermediaries for KYC operations for withdrawals and deposits.

TEE: We assume the system has a secure hardware module type TEE, discussed in the section 6.3.4.4. This protects against man-in-the-middle, replay/rollback attacks in offline mode. Indeed, this module secures fund transfer operations launched by the trusted application oriented towards FLOSS (Free/Libre and Open Source Software), notably by leveraging the advantages of data confidentiality and integrity in Alice.

Offline Block-PAD wallet: Alice does not have access to her private key, and it is not possible to bypass the offline Block-PAD wallet for offline transactions and online deposits (synchronization). Moreover, the device possesses an atomicity feature, implying that all required steps must be respected for any transaction, otherwise, the transaction will fail.

Communication Channel: We assume the exchange channel between Alice and Bob's devices via NFC or Bluetooth is entirely secure, as well as the online communications between different entities with a protocol type TLS.

Biometrics and Credentials: Access to the device for any operation is protected and controlled by passwords and fingerprints.

Cryptographic Methods: The system relies on correctly implemented cryptographic protocols, including blind signature as proposed in ecash2, zk-SNARK, asymmetric encryption with shared secret key, and CKKS.

6.4.2 System description

The global system is composed of:

The eIDWallet ecosystem based on:

- Attribute Provider, the entity in charge of providing official identification documents on the individual or legal entity.
- Identity Provider, which verifies and attests that the user of the eWallet is based on official identification documents. It has access to the user's real identity. It has read access to the list of PIDs with suspicious transactions.
- Personal Identification Data (PID) provider, which issues the PID for eID Wallet holders in compliance with the Identity Verification Provider's requirements.
- Qualified Trust Service Provider (qTSP), the certification authority in charge of issuing certificates or electronic attribute attestations.
- Wallet providers, the authority responsible for providing the digital wallet-PAD to end-users.

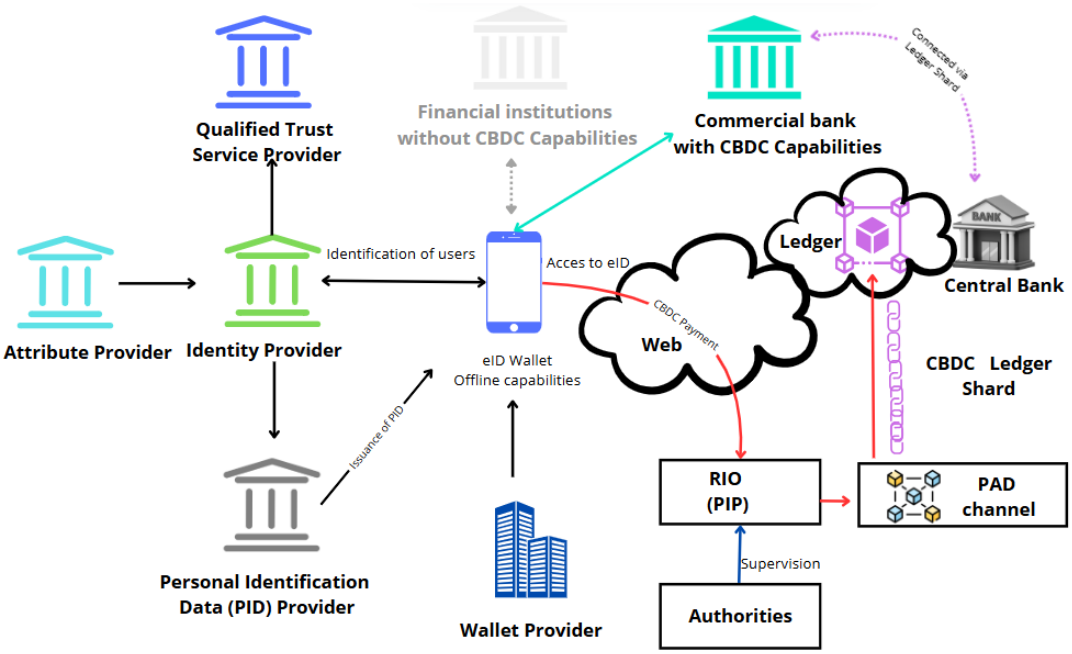


Figure 6.3: Overview of the proposed model. In addition to the client-facing flows, the commercial bank shard is connected to the central bank core ledger through a ledger shard link. This connection, consistent with the two-tier architecture in Fig.6.1, provides an implicit channel for transaction verification and audit.

- Regulatory Identity Overseer (RIO), here the Payment Interface Provider (PIP), which ensures the connection link between the PAD system and eID wallet for CBDC payment. It connects the pseudonymous identities of users with anonymous account(s) in a CBDC shard ledger. It generates pseudonymous identities with a user's eID without having access to the user's real identity. Before carrying out an online transaction, the RIO checks the validity of the user's eID. It also ensures that transactions are carried out in accordance with regulatory compliance. It also publishes a list of eID with suspicious transactions.
- PAD System described in the Section 6.3.4.6.
- Financial intermediaries (Commercial bank or Payment service), the intermediaries that handle their clients' KYC in a traditional manner, or alternatively, an onboarding based on eKYC relying on the client's PID, and manage distribution.
- CBDC ledger managed by the central bank, which issues and controls the currency.
- Regulatory Authority: it has read access to suspect PIDs and, if necessary, initiates a PID revocation operation with the Identity Verification Provider. Fig.6.3 shows the overview of the proposed model.

6.5 System Operations

For clarity of presentation, Table 6.2 summarizes the mapping between process steps, their corresponding algorithms, and the sections where they are described.

Table 6.2: Crosswalk of process steps, algorithms, and sections

Process step	Description	Algorithm(s)	Section
Onboarding (eID)	Identity creation and verification	Alg. 7, 8	5.1
Withdrawal	Withdrawal with blinding	– (procedure described)	5.2
Offline: Alice → Bob	Negotiation + verification + storage	Alg. 9, 10, 11	5.3
Offline: Bob → Carla	Consecutive payment (offline & hybrid modes)	Alg. 12	5.4
Online deposit / transfer (PAD)	Verification + PAD securing + update	Alg. 13, 14, 15, 16	5.5–5.6
Funding/Defunding	RIO claim & update	Alg. 17	5.7

6.5.1 Onboarding Phase

The onboarding phase begins with obtaining an eID wallet. This is a private and secure wallet, requiring FLOSS, open-source software. It is assumed that coins cannot be stolen due to enhanced biometric security. Alice obtains her eID wallet from the wallet provider. Each eID wallet corresponds to an account (individual or business). Each account is linked to a pair of cryptographic keys and Alice's credentials in the form of a digital certificate issued by the identifier provider. These credentials link Alice's real identity with the application. It is also assumed that Alice's digital identity can be revoked and updated. The term 'authorities' refers to the trustees and validators. The process unfolds as follows.

1. Alice downloads the eID wallet in a secure gallery application such as Google Play or App Store and obtains directly the public key of the ID provider. Then, she inserts her password and her personal information for the verification by the ID provider. The ID provider verifies the user's information with the Attribute Provider. Once the verification is completed and approved, the ID provider generates the credential structure follows the Algorithm 7.

2. Bank Onboarding: This phase assumes Alice already has an account with her bank. The eKYC process with her bank is as usual. The process of client registration protocol for CBDC account on the server S_v follows the Algorithm 8.

3. RIO Complete Onboarding Integration: $A_{\text{e_wallet}}$ sends to RIO its credentials for authentication. RIO can then generate the pseudonymous accounts associated with Alice's virtual identity. For each real public key certified $A_{\text{pk_cert}}$ associated with her virtual identity Alice can generate a set of short term asymmetric keys:

$$
\begin{aligned}
&(1) \quad (eID_A, A_{\text{pk}}, A_{\text{pvk}}, A_{\text{pk}_{\text{cert}}}) \rightarrow A_{\text{e_wallet}} \\
&(2) \quad (eID_B, B_{\text{pk}}, B_{\text{pvk}}, B_{\text{pk}_{\text{cert}}}) \rightarrow B_{\text{e_wallet}} \\
&(1) \rightarrow \phi(eID_A, A_{\text{pk}}, A_{\text{pk}_{\text{cert}}}, R_{\text{pk}}) \\
&(2) \rightarrow \phi(eID_B, B_{\text{pk}}, B_{\text{pk}_{\text{cert}}}, R_{\text{pk}}) \\
&(3) \quad \{(K_{A_{\text{priv}_i}}, K_{A_{\text{pub}_i}})\}_{i \in \{1 \ldots p\}} \leftarrow \text{PseudonymGen} \\
&(4) \quad \{(K_{B_{\text{priv}_i}}, K_{B_{\text{pub}_i}})\}_{i \in \{1 \ldots p\}} \leftarrow \text{PseudonymGen}
\end{aligned}
$$

Algorithm 7 Credential Structure and Key Generation

1: Generate credential structure with Identity Provider (IP) information and claims:

$$\Omega(\text{IP}, \mu) \rightarrow c_s$$

2: where c_s is the Credential structure with Identity Provider (IP) information and a set of μ claims.
3: Alice generates a pair of asymmetric keys and sends her public key to the ID Provider:

$$(A_{\text{pvk}}, A_{\text{pk}}) \leftarrow \text{KeyGen}()$$

4: The IP signs the credentials with its private key IP_{pvk} and associates them with Alice's public key, A_{pk}:

$$\Omega_E(\text{cs}, IP_{\text{pvk}}) \rightarrow (c_E, A_{\text{pk}})$$

5: where (c_E, A_{pk}) is the public key associated with the signed credential structure:

$$(c_E, A_{\text{pk}}) \rightarrow \text{eID}_A$$

6: Finally, the ID provider shares the hashed claims of the credential structure and its public key for verification to $A_{\text{e_wallet}}$:

$$H(c_c) \rightarrow H_c$$

7:

$$P(c_E, A_{\text{pk}}, cK, cV) \rightarrow A_{\text{e_wallet}}$$

8: where cK is the key of the claim to be verified, and cV is the value of the claim to be verified.

Algorithm 8 Client registration protocol

1: Alice's e_wallet generates a pair of zk-SNARK parameters (J_k, Z_k):

$$(J_k, Z_k) \leftarrow \text{zKeyGen}()$$

2: Then Alice's e_wallet shares V_c with the server S_v for verification:

$$V(c_E, A_{\text{pk}}, cK, cV) \rightarrow V_c$$

3: where V_c is the boolean result of the verification, true if both the claim and the signature are valid:

$$S_v \leftarrow P(V_c)$$

$$\Xi \leftarrow \text{Output}(V_c)$$

4: **if** $A_{\text{pk}}, \emptyset$ **then**
5: The server adds A_{pk} to the clients database:

$$\beta \leftarrow S_v(A_{\text{pk}})$$

6: **else**
7: It cancels the request.
8: **end if**
9: Then the server initializes Alice's current balance to 0:

$$A_{C_B} \leftarrow 0$$

10: Creates a A_{pk}_cert and sends it to $A_{\text{e_wallet}}$.

Table 6.3: Notation of RIO Onboarding

Variable	Description
eID_A	Alice's virtual identity
A_{pk}	Alice's virtual public key
$\{(K_{A\text{priv}_i}, K_{A\text{pub}_i})\}_{i\in\{1\ldots p\}}$	A set of Alice's short-term keys
eID_B	Bob's virtual identity
B_{pk}	Bob's virtual public key
$\{(K_{B\text{priv}_i}, K_{B\text{pub}_i})\}_{i\in\{1\ldots p\}}$	A set of Bob's short-term keys
eID_C	Carla's virtual identity
C_{pk}	Carla's public key
$\{(K_{C\text{priv}_i}, K_{C\text{pub}_i})\}_{i\in\{1\ldots p\}}$	A set of Carla's short-term keys

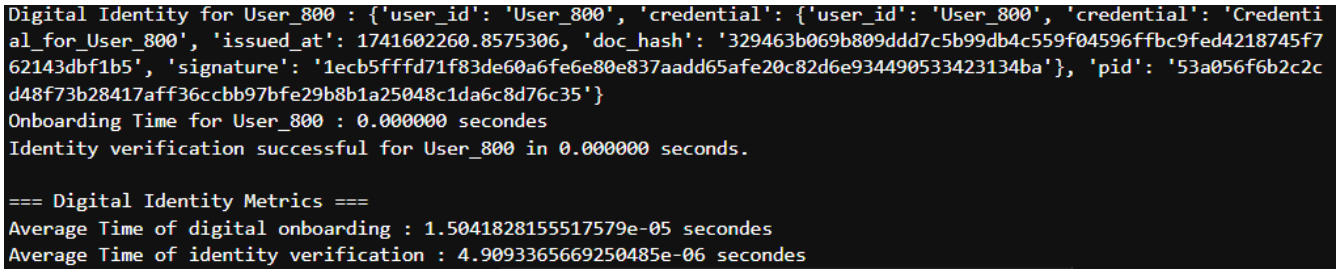


Figure 6.4: Simulation of users digital identity onboarding .

During online transactions, she uses the pseudonymous account information. The regulator, designated as Regulatory Identity Overseer (RIO), oversees regulatory compliance and identity verification. The RIO does not have access to Alice's real identity but only to her eID and Alice's certificate's public key. The central bank and the client's commercial bank do not have access to this second part of the onboarding. Table 6.3 showcases the notation of RIO Onboarding.

Fig.6.4 showcases the outcomes of simulation for 800 users in digital identity onboarding.

6.5.2 Withdrawal Phase

The withdrawal process usually follows the blind signature method determined in section 6.3.4.1 Alice withdraws coins from her online balance, which are signed by the central bank following the blind signature process.

The coins are then transferred to her offline wallet. The central bank blindly signs the coins using its private key. Knowing that Pk_b is its public key, we obtain finally as output an unblinded coin F_s.

Then Alice performs the offline transaction which is outlined by the following process unfolds as explained in [43] but with some modifications. Table 6.4 presents the notations of Offline Sealed Transaction.

6.5.3 Contract Negotiation

The first step of this offline transaction phase consists of contract negotiation between Alice and Bob. The algorithm 9 presents all the steps of the process, while the algorithms 10 and 11 showcase, respectively, the process of contract checking and coin storage.

Algorithm 9 Contract Negotiation

1: Alice and Bob agree on a contract for the payment.
2: Bob's device selects the attestation of the receiver:

$$R \leftarrow B_{\mathrm{pk}_{\mathrm{cert}}}$$

3: Bob's and Alice's devices establish a secure transaction via NFC or Bluetooth:

$$\sigma \leftarrow S(A_{\mathrm{device}}, B_{\mathrm{device}})$$

4: Bob initiates a payment request to Alice.

$$P_{\mathrm{req}} \leftarrow P(B \rightarrow A)$$

5: Alice inputs an e-coin corresponding to the requested amount:

$$F_s \leftarrow \varepsilon(A)$$

6: Alice signs the contract using the e-coin:

$$\Sigma_C \leftarrow \Sigma(F_s, C_n)$$

7: Generate the zk-SNARK proof of payment:

$$\Pi \leftarrow \Phi(F_s, J_k, W)$$

8: Encrypt the transaction data:

$$\Theta_{\mathrm{encr}} \leftarrow \varepsilon_T(\Pi, B_{\mathrm{pk}}, C_n, \Sigma_C, N, Z_k, A_{\mathrm{pvk}})$$

9: Send the encrypted data to Bob, associating it with Bob's public key:

$$\Gamma() \rightarrow (R, \sigma, P_{\mathrm{req}}, A_{\mathrm{pk}}, A_{\mathrm{pk}_{\mathrm{cert}}}, \Theta_{\mathrm{encr}})$$

Table 6.4: Notations of Offline Sealed Transaction

Notations	Description
$B_{\text{pk}}_\text{cert}$	Bob's certified public key
R	Receiver
σ	Secure transaction channel
Pk_b	Public key of the central bank
A_{device}	Alice's device
B_{device}	Bob's device
P_{req}	Payment request
Σ_C	Signed contract
C_n	Contract
W	Witness
Π	zk-SNARK proof of payment
ζ_1	Decision bit for Alice's public key checking
ζ_2	Decision bit for coin's authenticity
ζ_3	Decision bit for receiver transaction checking
ζ_4	Decision bit for zk-SNARK proof checking
σ	Coin's storage
N	Nonce
R_{pk}	RIO's public key
α	Receiver's pseudonymous identity
F_s	Unblinded coin

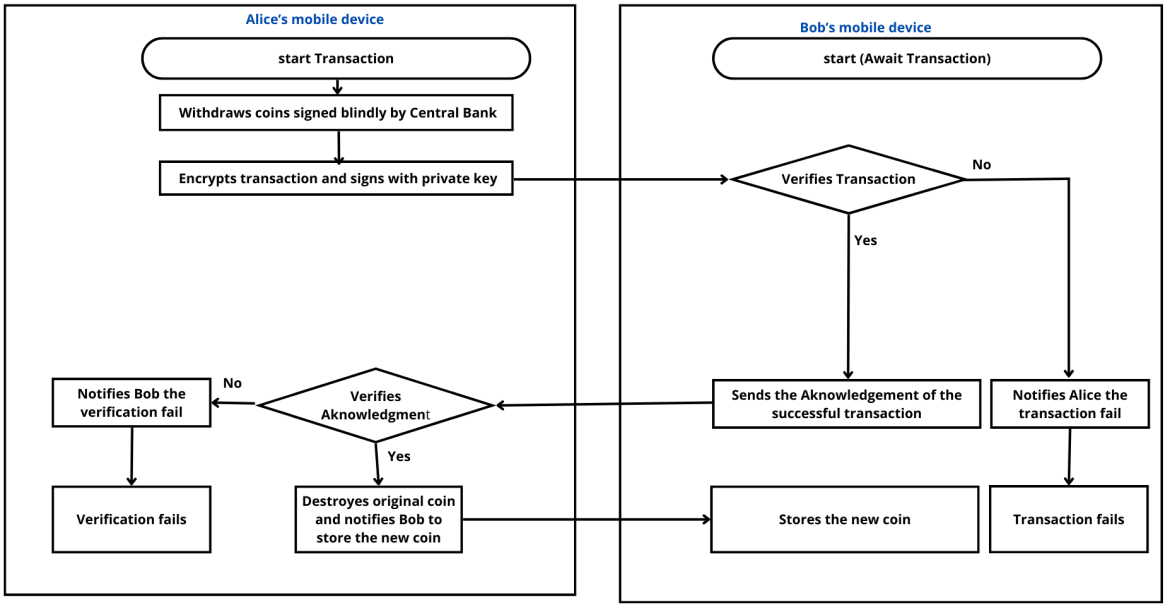


Figure 6.5: Offline transaction between Alice and Bob

6.5.4 Contract Checking: $V(\Gamma()) \rightarrow \zeta$

Fig. 6.5 depicts the main step of offline transaction between Alice and Bob while Fig. 6.6 depicts the result of transaction verification between Alice and Bob.

Algorithm 10 Contract Checking Procedure

Check Alice's public key:

$$\zeta_1 \leftarrow \Gamma_{A_{\text{pk}}}(A_{\text{pk}_{\text{cert}}}, A_{\text{pk}})$$

if $\zeta_1 = 0$ **then**
    Transaction fails.
else
    Check coin's authenticity:

$$\zeta_2 \leftarrow \Gamma_{F_s}(F_s, Pk_b)$$

    if $\zeta_2 = 0$ **then**
        Transaction fails.
    else
        Check receiver's transaction:

$$\zeta_3 \leftarrow \Gamma_{B_{\text{pk}}}(B_{\text{pk}})$$

        if $\zeta_3 = 0$ **then**
            Transaction aborts.
        else
            Check zk-SNARK proof:

$$\zeta_4 \leftarrow \Gamma_{\Pi}(\Pi, Z_k, F_s)$$

            if $\zeta_4 = 0$ **then**
                Transaction fails.
            else
                Bob's device sends an acknowledgment to Alice if the contract checking succeeds:

$$\Delta() \rightarrow \Lambda_1$$

                $\Lambda_1 = 1$
            end if
        end if
    end if
end if

Algorithm 11 Coin's Storage Procedure

Bob's device receives acknowledgment from Alice's device.
They store the coin:

$$\Delta() \rightarrow \Lambda_2$$

Confirm the coin has been stored:

$$\Lambda_2 = 1$$

Store the e-coin securely:

$$\sigma \leftarrow S(F_s)$$

```
Issuer's public key correctly loaded.
recipient_key correctly loaded in PublicKey.
Received recipient_key: PublicKey(166211014032968710836001713618267962628932866162019428266066986627770425548837551056046294260037822952539132036660894576234669379296720299963372416335230084481831866165356260424666820114849322143622066879835321717077552096717595917225263637241418367478352782161973886199150355176658692506997353364864907098166739492744755572890566419831175531579996547259168834734571224583338134350674853992478729516627766316302333812232013624145414580782487539105029964084569681545839256624991916934969418010183169426219099579543946799400184892470084342938321353231314010865996040515048333619353095322935199715859631860243993178318 7, 65537)
Type of recipient_key: <class 'rsa.key.PublicKey'>
Coin signature is valid.
Sender legitimacy verification successful: The sender is legitimate.
Transaction successfully logged.
zk-SNARK proof verification successful: The proof is valid.
Nonce is verified and unique, and the zk-SNARK proof is verified and unique.
Pièce stockée: {'coin_id': 'b506fd754b9be759a10717859d3bd3771e8c0c5afdbff91ceb4f026b9dcb539d', 'value': 50, 'signature': 'YnXLBbgKhfZ6b9Wuu1Skx9NRb75Q5SgK8TFpU5m3WzsWUest/gMJEt1uDztSF51q7l+2gkeUZDZ4nOUKRa0X+FEXItU84aPAVvT8OYAySr+ji6Rdbf/B/5vLZkbba3OuluRUyNflFk8RE4Wqb1iTGJ4md4Zyv8rpou2QGgMdgHWEWnFpThgt2rnqOXKO2zUKNxvC+SUEcASpKmZsAl/Dof7wJi69LK1W9jP20tTysPiK0+Ku7mn6Gppsr9HmbPt2Xv5o26uic+E2kjlPIm97CSdNNn6YYbMTR6QVv8kbC37OcWvivZsnP4mRo/f7+4GyuAANx5wkmHyIUIsQjMf6ZA==', 'nonce': 1, 'recipient': 'Bob'}
```


Figure 6.6: Overview of transaction verification result between Alice and Bob.

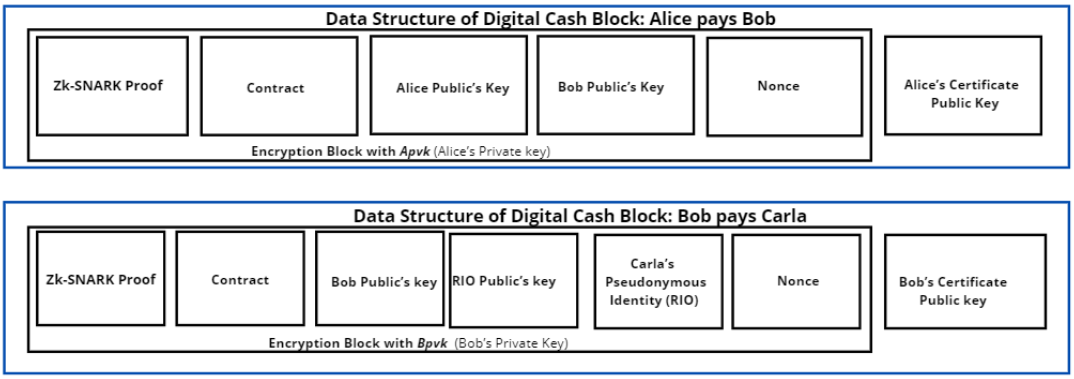


Figure 6.7: Data structure for CBDC consecutive offline transaction

6.5.5 Scenario of Bob Paying Carla

We now consider the case where Bob pays Carla, who does not have a secure device but has a pseudonymous account registered on the CBDC ledger. Table 6.5 presents notations for the Algorithm 12.The algorithm 12 showcases the steps of payment procedure in the case of this scenario.

Fig. 6.7 showcases the Data structure for consecutive offline transaction CBDC.

6.5.6 Online deposit/transfer via PAD

A user may deposit into their own account or transfer to another user's pseudonymous account. For instance, to make a deposit into Carla's account, Bob uses the *eIDWallet* application installed on his device. The deposit and transfer flows are managed via the RIO, which verifies Bob's credentials before enabling payment.

$$\Xi \leftarrow \text{Output}(V_c).$$

Algorithm 12 Payment Procedure from Bob to Carla

— **Offline Phase** —
Bob and Carla agree on a contract for the payment.
Bob's wallet requires him to select the attestation of the RIO:

$$R \leftarrow R_{\mathrm{pk}_{\mathrm{cert}}}$$

Carla provides Bob one of her public pseudonym identities.
Bob inputs an e-coin corresponding to the desired amount:

$$F_s \leftarrow \varepsilon(B)$$

Bob's wallet signs the contract using the e-coin:

$$\Sigma_C \leftarrow \Sigma(F_s, C_n)$$

Generate the zk-SNARK proof of payment:

$$\Pi \leftarrow \Phi(F_s, J_k, W)$$

Encrypt the transaction data:

$$\Theta_{\mathrm{encr}} \leftarrow \varepsilon_T(\Pi, R_{\mathrm{pk}}, C_n, \Sigma_C, N, Z_k, B_{\mathrm{pvk}}, \alpha)$$

— **Online Phase (synchronization with RIO)** —
Send the encrypted data to RIO, associating it with RIO's public key:

$$\Gamma() \rightarrow (R, B_{\mathrm{pk}}, R_{\mathrm{pk}}, B_{\mathrm{pk}_{\mathrm{cert}}}, \Theta_{\mathrm{encr}})$$

RIO: After successful execution of Algorithms 7–9, issue receipt λ bound to Carla's pseudonymous account α.
Carla: Observes λ and updated balance upon next connection.

Table 6.5: Notations for Algorithm 6

Notation	Description
D	Decision bit for double spending verification
L	Ledger Address
T	Token value
K_{sym}	Symmetric key
R_{sym}	Mask for symmetric key
$\{K_{\text{pubt}}^{(i)}\}$	Set of trustees' public keys
$\{K_{\text{pubv}}^{(j)}\}$	Set of validators' public keys
N_t	Number of trustees
R_{sf}	Retrieved secret function

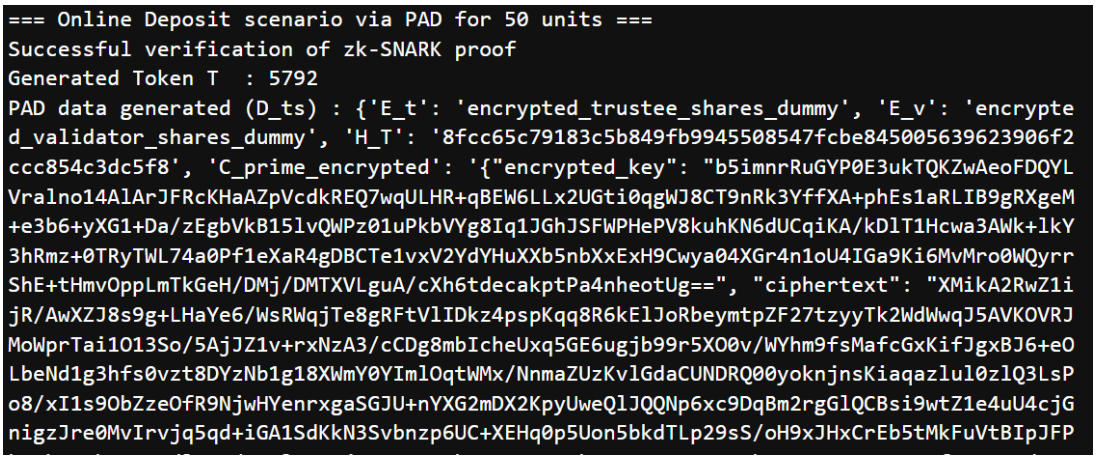


Figure 6.8: Online deposit simulation

If the credentials are successfully verified, the RIO initiates the corresponding payment or deposit. Fig. 6.8 depicts the result simulation of online deposit.

Step 0: Transaction Verification

RIO's wallet checks the transaction in the same way as a transaction between Alice and Bob. The difference is that the RIO has access to all the coins signed by the central bank and already distributed to users. Therefore, it can verify that the deposited coin has not already been used; if the coin is authentic and unique, it is followed by the verification of the zk-SNARK proof at the PAD level. The Algorithm 13 presents the process of transaction verification using zk-SNARK proof.

Step 0.1: Verify zk-SNARK Proof for Double Spending

If the zk-SNARK proof verification is successful, the RIO's wallet sends a receipt to the wallet for the coin's destruction (exactly as in the offline payment process between Alice and Bob). The next step consists of securing the secret, which is a blinded coin F_s, by using the protocol proposed in Section 6.3.4.1 of [118]. The Algorithm 14 presents the process for securing the blinded coin.

Algorithm 13 Verify zk-SNARK proof to check for double spending at the PAD level

1: $D \leftarrow F(\Pi, Z_k)$
2: **if** $D = 0 \vee \lambda(\Pi, L)$ **then**
3: $\quad \psi(L, \text{"Alert"})$
4: **else**
5: $\quad \psi(L, \text{"Ok"})$
6: **end if**

Algorithm 14 Securing the Blinded Coin F_s

1: **if** $\psi(L, \text{"Ok"})$ **then**
2: Encrypt Secret and Token with symmetric key:

$$C \leftarrow E(T, F_s, K_{\text{sym}})$$

3: Sign and Encrypt Content with Bob's public key:

$$C' \leftarrow E(E(T, F_s, K_{\text{sym}}, K_{\text{Apriv}}), K_{\text{Bpub}})$$

4: Mask Symmetric Key and Share function for trustees:

$$S_m \leftarrow K_{\text{sym}} \oplus R_{\text{sym}}, \quad S_t \leftarrow P(S_m, N_t)$$

5: Generate Validator Shares and Encrypt Trustee Hashes:

$$V \leftarrow P(R, K_{\text{Bpub}}, N_v), \quad E_{hv} \leftarrow E(H(S_t), \{K_{\text{pubv}}^{(j)}\})$$

6: Encrypt Shares for Trustees and Validators:

$$E_t, E_v \leftarrow E(P(S_m, V, \{K_{\text{pubt}}^{(i)}\}, K_{\text{pubv}}^{(j)}))$$

7: Data Transmission and storage:

$$D_{ts} \leftarrow (E_t, E_v, H(T), C')$$

8: **end if**

Step 1: Securing the blinded coin F_s

Finally, the RIO sends Data to the PAD System with zk-SNARK Proof and provides Token value T and Verification Key K_{verif} to Bob.

Step 2: PIP Requesting a Decryption

RIO verifies the transaction threshold, creates the decryption request and posts it anonymously to the ledger.

$$\begin{aligned} &\text{If } T > \theta, \text{ then} \\ &\quad \phi \leftarrow G(H(T), K_{\text{Bpub}}) \end{aligned}$$

Step 3 : Bob Recovers and Decrypts Secret function

$$R_{sf} \leftarrow G(E^{-1}(E^{-1}(C', K_{\text{Bpriv}}), K_{\text{verif}}))$$

In this step, Bob has to retrieve and decode a secret. Bob uses his private key to decrypt the secret, and he uses Alice's verification key to confirm that it is authentic. After verification, he uses the decrypted content to reassemble the secret and retrieves

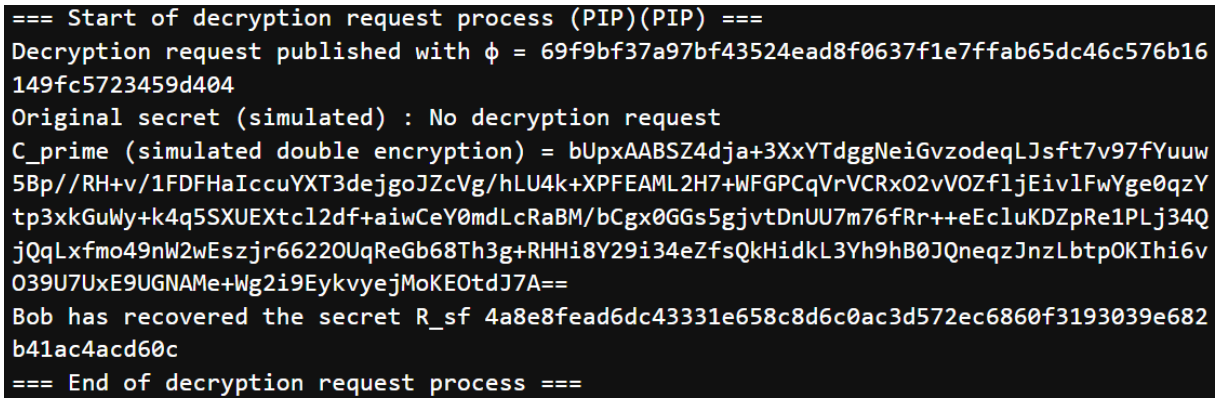


Figure 6.9: Result simulation of decryption process

Table 6.6: Notations for Algorithm 9.

Notation	Meaning
q	A large prime number
t	Another prime number for the message
n	Degree of polynomial
MAXB	Maximum allowed balance
ω	A random polynomial of degree n
e	A small noise
r, e_0, e_1, a	Random polynomials of degree n
$(R_{\text{pk}_0}, R_{\text{pk}_1})$	Pair of keys
R_{pk}	Public key of PIP
α	Recipient's pseudonym (Alice, Bob, or Carla)
op	Operation type
R_{pvk}	Random polynomials of degree n (RIO's private key)
T	Token value (coin's amount)
C_m	Current amount
C_b	Current balance

the original secret data. The first step in the PIP Decryption Request function is to look over the transaction data. A decryption request is created when the value of the token surpasses a predetermined threshold or when the transaction occurs frequently. After that, this request is made public for decryption, and the results are given back. No more action is taken if the transaction does not meet the requirements for a deeper examination. Fig. 6.9 depicts the result simulation of decryption process.

Step 4: Balance Updating

The PAD ledger encrypts the value token and sends it to the smart contract address shard ledger. Table 6.6 is the Notations for Algorithm 15 which is the process of Balance decryption Procedure. Fig. 6.10 provides a visual representation following the deployment of the smart contract for a transaction in the PAD Ledger, while the Fig. 6.11 illustrates the recording of the transaction in the blockchain PAD Ledger. The balance decryption procedure is described in Algorithm 16.

Algorithm 15 Balance Updating Procedure.

1: Key Generation:

$$\text{GenKey}() \rightarrow (R_{\text{pvk}}, (R_{\text{pk}_0}, R_{\text{pk}_1}))$$

$$R_{\text{pk}_0} = -a\omega + e$$

$$R_{\text{pk}_1} = a$$

2: Encryption:

$$\text{Enc}(T, R_{\text{pk}}) \rightarrow (c_0, c_1)$$

$$c_0 = R_{\text{pk}_0} \cdot r + e_0 + Tt$$

$$c_1 = R_{\text{pk}_1} \cdot r + e_1$$

3: Update Alice or Bob or Carla account:

$$\text{Upd}(c_B, T, \text{op}, R_{\text{pk}}, \alpha) \rightarrow c_{B'}$$

$$\alpha \in \{K_{A\text{pub}_i}, K_{B\text{pub}_i}, K_{C\text{pub}_i}\}_{i \in \{1 \ldots p\}}$$

$$c_T = \text{Enc}(T, R_{\text{pk}})$$

4: **if** op = "INC" **then**

$$c_{B'} = c_B + c_T$$

5: **if** $\text{Dec}(c_{B'}, R_{\text{pvk}}, t, q) >$ MAXB **then**
6: return error
7: **end if**
8: **return** $c_{B'}$
9: **else**
10: **if** op = "DEC" **then**
11: **return** $c_B + (-c_T)$
12: **end if**
13: **end if**

Algorithm 16 Balance Decryption Procedure

1: Decryption:

$$\text{Dec}(c, R_{\text{pvk}}, t, q) \rightarrow B'$$

$$B' = \text{round}((c_0 + c_1 \cdot R_{\text{pvk}})/t) \mod q$$

2: The smart contract performs the homomorphic operation and updates the balance:
3: **if** operationType is INC **then**
4: **if** balance does not exceed MAXB **then**
5: update Recipient's balance
6: **else**
7: return an error
8: **end if**
9: **else if** operationType is DEC **then**
10: update Recipient's balance
11: **end if**



Figure 6.10: Smart contract deployment for transaction in PAD Ledger

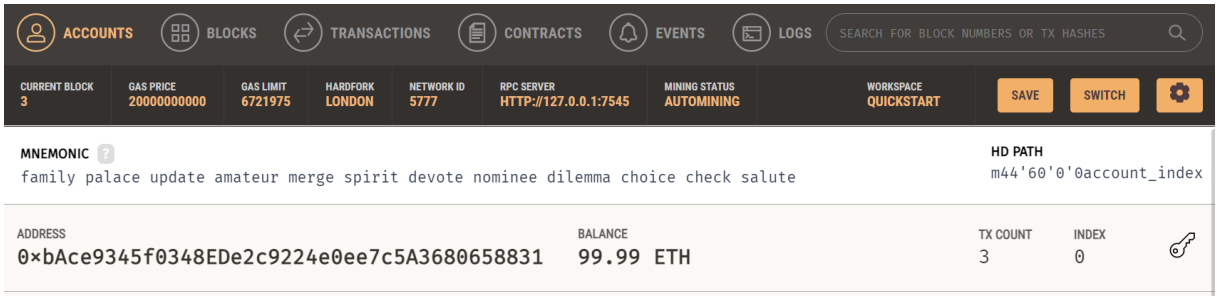


Figure 6.11: Overview of recording transaction in PAD Ledger

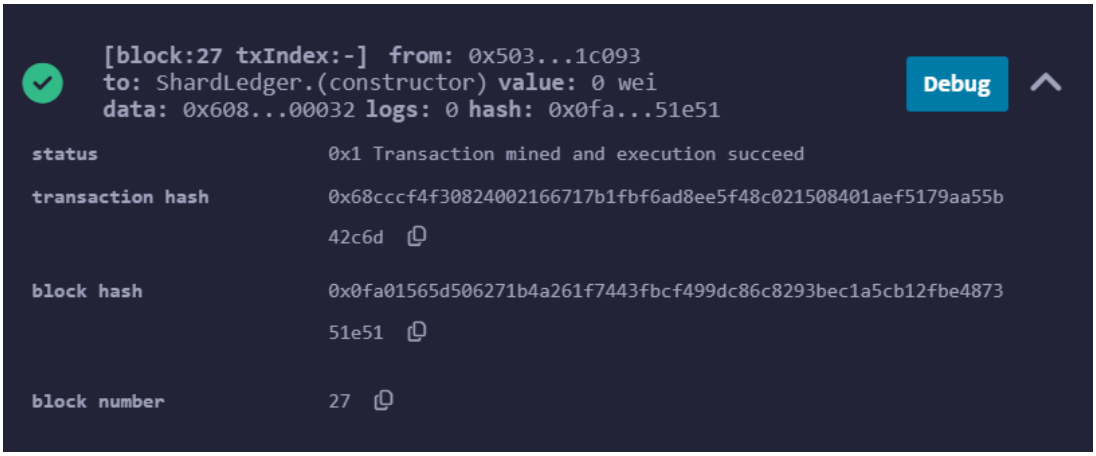


Figure 6.12: Funding transaction following the deployment of the smart contract.

6.5.7 Funding/Defunding Operation

The PAD adds a layer of privacy protection for transferring funds into a Central Bank Digital Currency (CBDC) account from a traditional bank account, and vice versa. When Alice transfers money from her regular fiat bank account to her CBDC account, the bank can see the connection between her personal identity and her unique CBDC account identifier. However, with the PAD integration, Alice first deposits funds into a special wallet on the CBDC network, a RIO wallet, and receives a cryptographic claim check, λ_A, which she can re-randomize to break the traceability to her original transaction. Table 6.7 presents the notations of the Algorithm 17 which showcases the implementation of these operations.

Fig. 6.12 illustrates the materialization of a funding transaction following the deployment of the smart contract, while Fig. 6.13 shows the recording of several funding transactions in the blockchain through the Ganache tool.

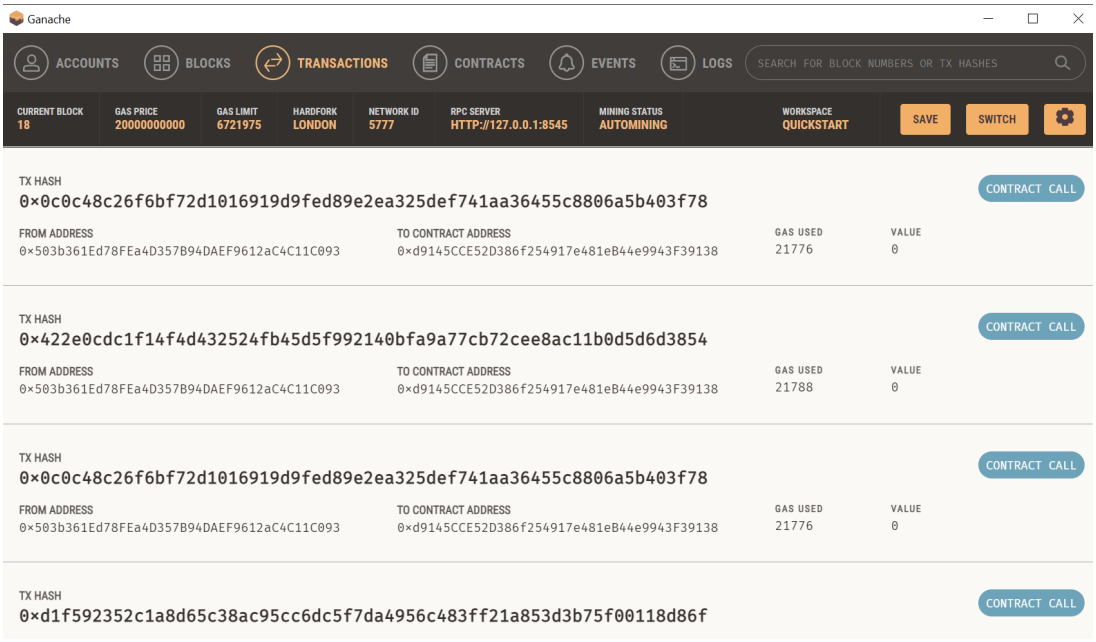


Figure 6.13: Funding transaction following the deployment of the smart contract.

Algorithm 17 Funding Process

1: Alice deposits an amount T from her bank account A_B to the RIO's wallet R_w with a PAD process:

$$P_T \leftarrow \kappa(A_B, T, R_{\text{pk}})$$

2: A cryptographic claim check is issued to Alice:

$$\lambda_A \leftarrow \eta(P_T)$$

3: Alice re-randomizes the claim check:

$$\lambda'_A \leftarrow \eta'(\lambda_A)$$

Table 6.7: Notations for Algorithm 11

Notation	Description
A_B	Alice's bank account
R_w	RIO's wallet
λ_A	Cryptographic claim check
λ'_A	Re-randomized claim check

Alice then broadcasts to the CBDC PAD chain, which includes trustees, validators, the PAD channel, and the CBDC shard ledger , to update her account.

Fig. 6.14 depicts the high-level functionnal map of PAD system centric Blockchain for CBDC payments.

6.6 Security Analysis

We propose to analyze the entirety of the system. By reassessing the risks associated with the PAD, previously presented in [118], with fresh insight and a new approach, we then evaluate the rest of the system and implement corrective measures.

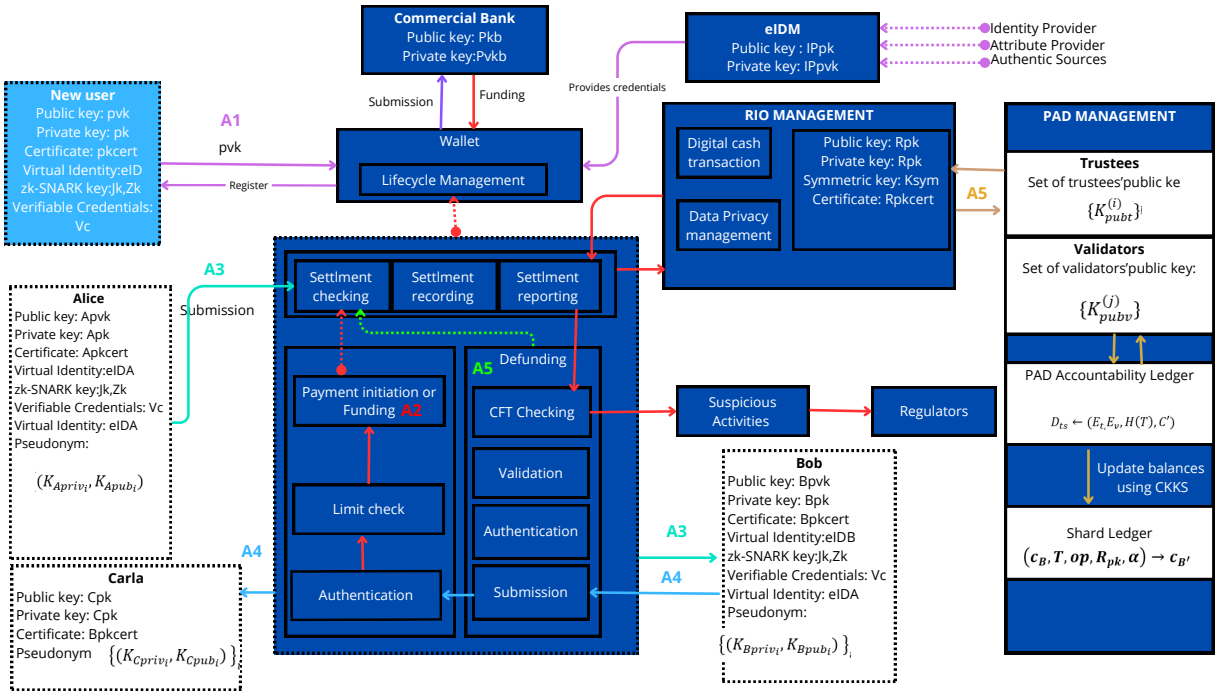


Figure 6.14: PAD system-centric Blockchain for CBDC payments. Annotations A1–A6 indicate the main process steps: A1 – Onboarding (Algs. 1–2), A2 – Withdrawal (Alg. 3), A3 – Offline Alice→Bob (Algs. 4–5), A4 – Offline Bob→Carla (Alg. 6), A5 – Online deposit/transfer via PAD (Algs. 7–10), A6 – Funding/Defunding (Alg. 11). See also Table 6.2 for a summary crosswalk.

6.6.1 Coalition Attacks Against Secret Sharing

The most imminent risk for Bob is that trustees and validators collaborate to discover Bob's secret "s". To preserve his secret, it is vital that Bob ensures the PIP or the recipient "Carla" can only decrypt the secret under certain conditions. We state the following theorem:

Theorem 1. *We assume the secret sharing thresholds for the trustees k_1 and validators k_2. For any coalition involving k_1 trustees and k_2 validators and Bob having enough information to decrypt "s", it is then possible to drastically reduce the probability of such a coalition by adjusting the ratios $\frac{k_1}{n_1}$ and $\frac{k_2}{n_2}$, where n_1 and n_2 represent the total number of trustees and validators, respectively.*

6.6.2 Coalition Attacks Against Carla's Identity

This attack would involve trustees pooling their actions to bias the process based on the identity of the requester. The identity should only be revealed by the RIO after the validators' calculations. Note that the boycott or publishing of invalid parts by trustees could slow down the process in the PAD system. Validators, in their role as guarantors, are tasked with identifying any invalid trustee part on the PAD blockchain channel.

6.6.3 Corrupted Trustees and Validators Risk

A corrupted authority is either an authority that leaves the system holding a valid secret part or a trustee who boycotts by not providing responses. To avoid Censorship Attacks that reduce the risk of insufficient responses from trustees, we state the following theorem:

Theorem 2: *For a number of corrupted authorities a present in the system, there exists an honest majority such that $n = 2a + 1$. Thus, for a number b of trustees (or validators) having left the system, we need that $a + b < t$, where t is the threshold.*

Theorem 3: *In a system where secrets are distributed, the protocol's progression is conditioned by the publication of at least k elements out of n by the trustees. Thus, blocking this process requires the collusion of at least $n - k + 1$ unscrupulous trustees. Ideally, k is chosen to be less than half of n to minimize the probability of such coalitions. This logic also extends to validators and their specific parameters k_0, n_0.*

6.6.4 Bob Fraud Risk: Double Incoming

In offline payment systems based on account synchronization on the ledger, there is a significant risk of Bob committing double incoming fraud. This involves using the same monetary unit to receive an amount of money in his offline account (for example, a CBDC smart card he owns) and in his online account. Even if the token itself is unusable due to cryptographic techniques, this often creates token synchronization problems. Currently, this is unlikely to happen. Indeed, in our protocol, the online deposit phase requires making at least one offline payment before making a deposit. The fact is that once the transaction is completed (with the receipt) by the user, he is deprived of his coin through a destruction operation. To enhance security, one can consider including a "Redeem Online" tag in the payment transaction frame, which will be taken into account during the online verification of the zk-SNARK proof.

6.6.5 DDoS Attacks Vulnerability Risk

Denial of service attacks capable of defeating the system can be orchestrated during the withdrawal and deposit phase. Although the withdrawal and deposit process does not occur on the same communication channel, the system may still be vulnerable to denial of service attacks, notably the RIO managing decryption and transaction recording operations. To mitigate this, one can limit the number of anonymous accounts and define a transaction threshold per user (here it refers to a pseudonymous account). Before each transaction, we can proceed with a threshold verification implemented with the following Algorithm 18: Let C_n be the counter initialized to 0, which increases by 1 every second. Let T be the time threshold for the first phase of command processing, and let N be the maximum number of commands to process before changing the protocol, with n, the command counter, initialized to 0. The system alternates between an active waiting phase for new commands and a command processing phase, adjusting its behavior based on the elapsed time and the number of processed commands.

6.6.6 Risks Related to Smart Contracts and Inherent to the Blockchain Network

Blockchain has various risks, including denial of service attacks and endpoint security [42]. This context focuses on smart contract vulnerabilities, which are crucial in verifying zk-SNARK payment proofs, recording PAD requests, and anonymously updating accounts in the CBDC ledger. Smart contract failures can have severe consequences, such as bad randomness, ether reception forcing, incorrect interfaces, integer overflow, race conditions, re-entrancy, unchecked external calls, unprotected functions, and variable shadowing. Therefore, thorough testing and auditing of smart contracts before deployment,

Algorithm 18 Command Processing Procedure

Initialization:
$K() \leftarrow (C_n \leftarrow 0,\ n \leftarrow 0,\ \text{Activate } C_n)$
Main loop:
while true **do**
    if $C_n < T$ **then**
        $\Phi_{\text{cmd}} \leftarrow$ CheckForCommands()
        if $\Phi_{\text{cmd}} = 1$ **then**
            ProcessCommands(*commands*) ▷ Process deposit or withdrawal
        else
            Wait for new commands
        end if
    else
        if $n > N$ **then**
            Proceed with the normal protocol
        end if
        if $\Phi_{\text{cmd}} = 1$ **then**
            ProcessCommands(*commands*) ▷ Process deposit or withdrawal
        end if
    end if
end while

Utilized functions (description only):
CheckForCommands(): returns 1 if deposit or withdrawal commands are available, 0 otherwise.
ProcessCommands(*commands*): executes the available deposit or withdrawal commands.

using dedicated tools or AI [120], and post-deployment anomaly detection monitoring are essential.

6.6.7 Security Properties Evaluation

6.6.7.1 Traceability

All actions and transactions in the system, such as decryption requests, trustee or validator responses, or a CBDC transaction, are recorded transparently and verifiably. This guarantees the unforgeability of coins. In the offline phase, the zk-SNARK proof is associated with a nonce at the recipient, ensuring that for each transaction from the same issuer, the payee's device must track the previously received nonce to make any replay attempt impossible.

6.6.7.2 Unlinkability

The real identities of Alice, Bob, or Carla are not revealed during transactions, which are conducted through completely anonymous accounts. Participants can have multiple anonymous accounts, and transactions cannot be linked to reveal their real identities. Identity disclosure is only possible in cases of proven suspicion and requires a process involving the Identity Provider, RIO, and auditors. Outside of this process, retrieving a user's identity is extremely difficult. Additionally, even if an anonymous transaction is visible in the ledger, its origin cannot be traced, ensuring the confidentiality of users and their transactions.

6.6.7.3 Undeniability

Attempts at double spending or double incoming must be detected and, if applicable, attributed only to their authors. Therefore, the system must be able to detect cases of double spending and double deposit (double income) independently of the number of users in the system.

6.6.7.4 Balance Integrity.

All accounts must be correctly updated during a financial transaction. This implies that the operations of incrementing and decrementing the balances at the honest issuer and recipient involved in the transaction must be performed safely.

6.7 Performance Evaluation

In this section, we discuss the experiments conducted to evaluate our end-to-end solution design. We assess the system's performance through benchmarking and analysis of different operational phases, including digital identity management, withdrawal operations, payment transactions, as well as deposit and funding operations. The performance metrics considered include throughput, latency, average onboarding time, identity verification time, zk-SNARK proof generation and verification time, computational load, and gas usage for blockchain-involved operations.

In our experimental evaluation, we aim to compare the performance of our CBDC payment solution under various configurations. The performance metrics (such as latency, throughput, and CPU load) are measured over multiple independent runs. Therefore, the proposed experimental design, which includes multiple runs with standard deviation evaluation for critical phases and a reduced number of runs for lower variability tasks, offers an efficient and robust method for assessing the performance and scalability of the CBDC payment solution. This approach not only ensures statistical reliability but also provides clear insights into the system's behavior under different load conditions.

6.7.1 Experimental Setup

To evaluate our solution, we implemented a simulator based on the CBDC SIM tool. This open-source solution models the classical operations of a CBDC infrastructure. However, to optimally test our system's operations, we substantially modified and customized our simulator. Overall, the objective is to prototype a secure system for issuing, transferring, and depositing tokens (coins) that guarantees privacy with an extensible payment lifecycle management framework.

Our prototype is implemented in Python 3.9.2 and runs on Windows 10 Professional, deployed on a computer equipped with an Intel(R) Core(TM) i7-5700HQ CPU @ 2.70GHz (64-bit, 16GB DDR3, 512GB SSD). We utilize the PyCryptodome library for cryptographic operations. Symmetric encryption is instantiated with SHA256 for indexing simulated proofs and AES in GCM mode, an authenticated mode ensuring confidentiality and integrity. The integration of cryptographic randomness provides the necessary entropy for generating symmetric keys, initialization vectors, etc. These symmetric keys are 256 bits in length.

For AES key transportation, we use hybrid encryption combining PKCS1_OAEP with RSA (2048-bit) to perform asymmetric encryption for secure symmetric key distri-

bution. For implementing hashing functions, we use Python's standard `hashlib` module, primarily for coin identifier derivation (`coin_id`, etc.) and simulated blind signatures (`sign_blinded_hash`).

For the on-chain implementation, we deployed smart contracts on Remix v0.61.1[121]. This platform, dedicated to Ethereum smart contract development and testing, is used in conjunction with Ganache[122]. Ganache allows deploying Ethereum smart contracts in a local blockchain environment, enabling transaction simulations.

For payment operations, we opted for the Ipv8[123] communication protocol. Ipv8 encapsulates and transmits encrypted transactions end-to-end in a peer-to-peer manner without relying on centralized servers, thereby enhancing privacy protection. The `pyipv8` package is integrated with NFC and Bluetooth protocols to enable autonomous transactions and ensure resilience in the absence of internet connectivity.

The implementation of our Block-PAD framework, along with all simulation notebooks and scripts, is publicly available in our GitHub repository [124], ensuring reproducibility of the reported results.

6.7.2 Results Analysis

6.7.2.1 Digital Identity Management

For user onboarding, we experimented with different user counts—10, 50, 100, 500, and 1000—and recorded digital identity metrics such as average onboarding time, standard deviation of onboarding time, average verification time, and total onboarding time. Each configuration was tested over thirty independent runs to reduce random bias. The results are summarized in Table 6.8 and visualized in Fig.6.15.

The average onboarding time per user remains extremely low, in the order of 10^{-5} to 10^{-4} seconds, across all scenarios. Fluctuations between runs are minimal, with slightly more pronounced variability for small cohorts (10 and 50 users) compared to larger groups. This suggests that variance is more sensitive to short simulation batches than to the overall scalability of the system.

The verification process is consistently faster than onboarding, with execution times in the order of 10^{-6} seconds. Results show almost negligible cost, with only minor fluctuations at 500 and 1000 users, likely due to background system activity rather than inherent inefficiency of the protocol.

The standard deviation of onboarding times increases slightly with the number of users, which is expected given that larger-scale simulations amplify natural variability. However, values remain low, confirming the overall stability of the process.

The total onboarding time increases linearly with the number of users, reaching approximately 3.4×10^{-2} seconds for 1000 users. This linear trend is expected since the system processes users sequentially, and it confirms that performance degradation is proportional to the workload without unexpected overhead.

Overall, the results indicate that the proposed digital identity framework is lightweight and scalable, with near-instantaneous onboarding and verification per user.We observed that these outcomes are consistent across all tested scales, suggesting that execution times remain negligible even when the number of users increases.

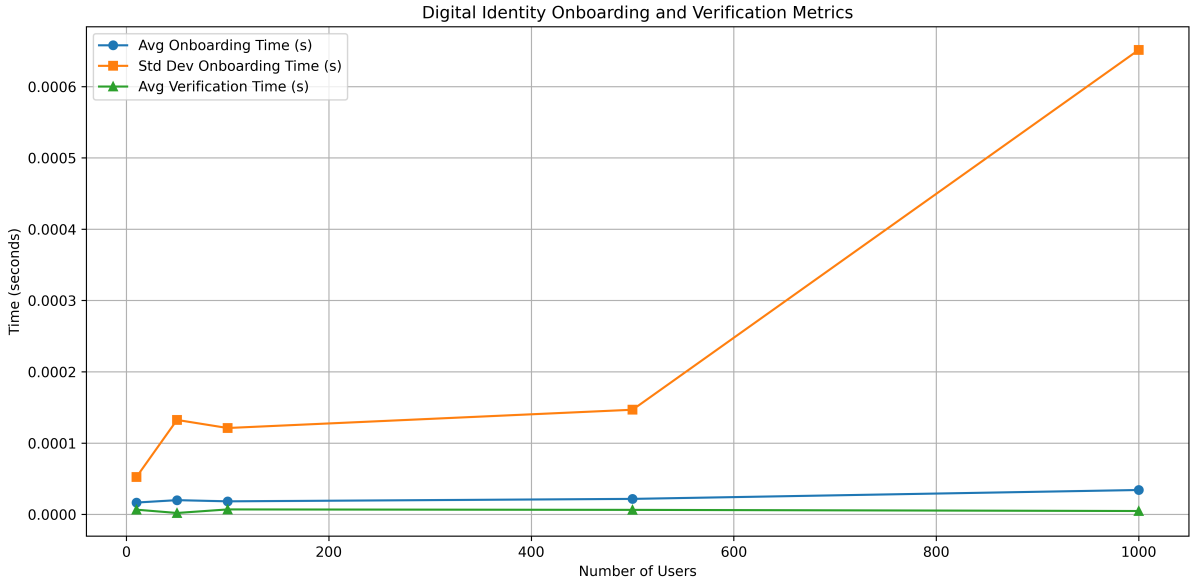
6.7.2.2 Withdrawal Transactions

The observed metrics in this phase include the evolution of load computing by run index, distribution of load computing, evolution of latency by run index, evolution of throughput

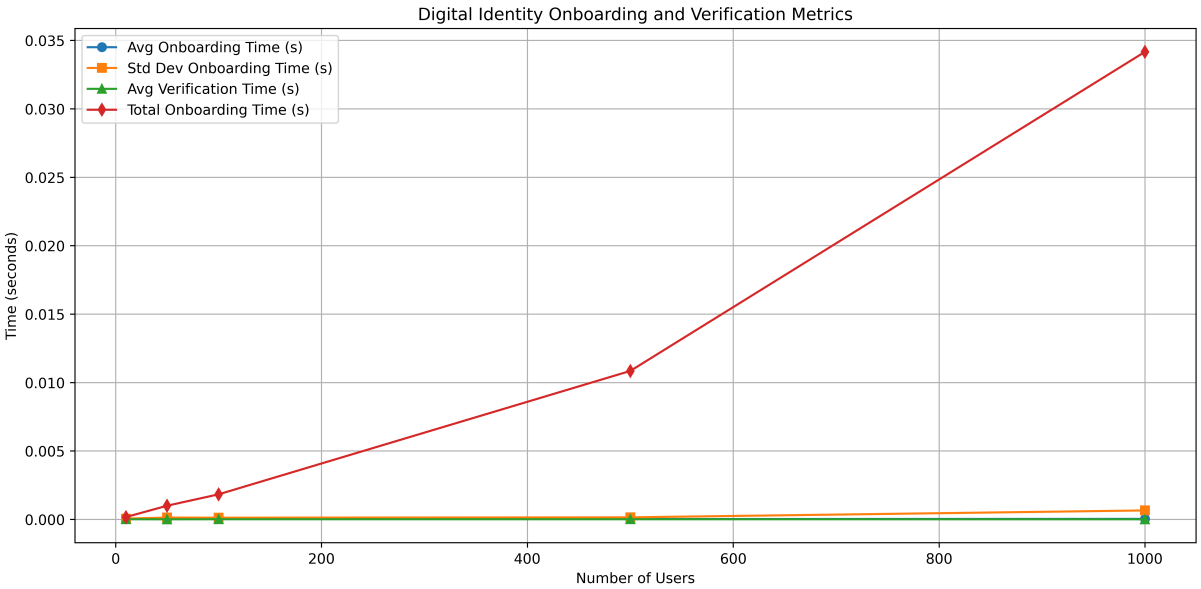
Table 6.8: Onboarding and Identity Verification Metrics (averaged over runs).

Users	Avg Onb (s)	Std Onb (s)	Avg Verif (s)	Total Onb (s)
10	1.663e-05	5.247e-05	6.633e-06	1.663e-04
50	2.000e-05	1.325e-04	2.000e-06	1.000e-03
100	1.833e-05	1.212e-04	7.000e-06	1.833e-03
500	2.177e-05	1.468e-04	6.367e-06	1.088e-02
1000	3.423e-05	6.515e-04	4.767e-06	3.423e-02

Legend: Avg Onb = Avg Onboarding Time (blue), Std Onb = Std Dev Onboarding Time (orange), Avg Verif = Avg Verification Time (green), Total Onb = Total Onboarding Time (red).



(a) Per-user onboarding and verification metrics.



(b) Overall onboarding time vs per-user metrics.

Figure 6.15: Digital identity onboarding and verification performance under varying user scales.

by run index, and transaction verification time. For better visualization and comprehension of the graphs, we limited our observations to the first 10 runs. Table 6.9 shows the results of the withdrawal simulation, and Fig. 6.16 shows the graphical evolution of the observed metrics.

Table 6.9: Results of the first 10 withdrawals for amounts of 50, 75, and 100.

Transaction ID	Amount	Latency (s)	Load Computing (s)	Throughput (tx/s)
0	50	0.0694	0.0696	13.2967
1	50	0.0671	0.0671	13.2967
2	50	0.0665	0.0665	13.2967
3	50	0.0738	0.0734	13.2967
4	50	0.0686	0.0685	13.2967
5	50	0.0671	0.0671	13.2967
6	50	0.0681	0.0680	13.2967
7	50	0.0689	0.0693	13.2967
8	50	0.0708	0.0703	13.2967
9	50	0.0666	0.0666	13.2967
0	75	0.1060	0.1037	13.2305
1	75	0.0823	0.0800	13.2305
2	75	0.0827	0.0818	13.2305
3	75	0.0885	0.0850	13.2305
4	75	0.0798	0.0793	13.2305
5	75	0.0814	0.0800	13.2305
6	75	0.0779	0.0773	13.2305
7	75	0.0756	0.0749	13.2305
8	75	0.0798	0.0771	13.2305
9	75	0.0767	0.0739	13.2305
0	100	0.0688	0.0688	13.4545
1	100	0.0676	0.0676	13.4545
2	100	0.0666	0.0666	13.4545
3	100	0.0701	0.0696	13.4545
4	100	0.0688	0.0690	13.4545
5	100	0.0669	0.0669	13.4545
6	100	0.0668	0.0670	13.4545
7	100	0.0718	0.0710	13.4545
8	100	0.0676	0.0679	13.4545
9	100	0.0677	0.0677	13.4545

The throughput over runs indicates that the 100-unit scenario (green line) achieves the highest throughput (about 13.45 TPS). The 50-unit scenario (blue) remains around 13.30 TPS, slightly below 100, while the 75-unit scenario (orange) stabilizes near 13.20 TPS, the lowest among the three. The relatively flat throughput curves suggest that the solution processes each run at a steady rate. The slightly higher throughput for 100 units suggests that the protocol may be optimized for larger transactions, or that internal overhead dominates over the transaction amount.

The latency evolution graph shows that the 75-unit scenario (orange line) consistently starts higher (0.105 s at run 0) and remains above the other two for most runs, stabilizing around 0.08–0.09 s. The 50-unit (blue) and 100-unit (green) scenarios generally stay within 0.065–0.075 s, with occasional dips below the 50-unit curve. The stability of latency after a few initial runs suggests rapid system stabilization. The higher latency for 75 units could be due to cryptographic overhead or internal parameter alignment.

Regarding load computing over runs, the 75-unit case exhibits slightly larger fluctuations, peaking near 0.085–0.090 s by run 4 before descending toward 0.08 s. The 50 and 100-unit cases remain within a compact 0.065–0.075 s range. This suggests that CPU overhead is the primary driver behind latency variations in the 75-unit case.

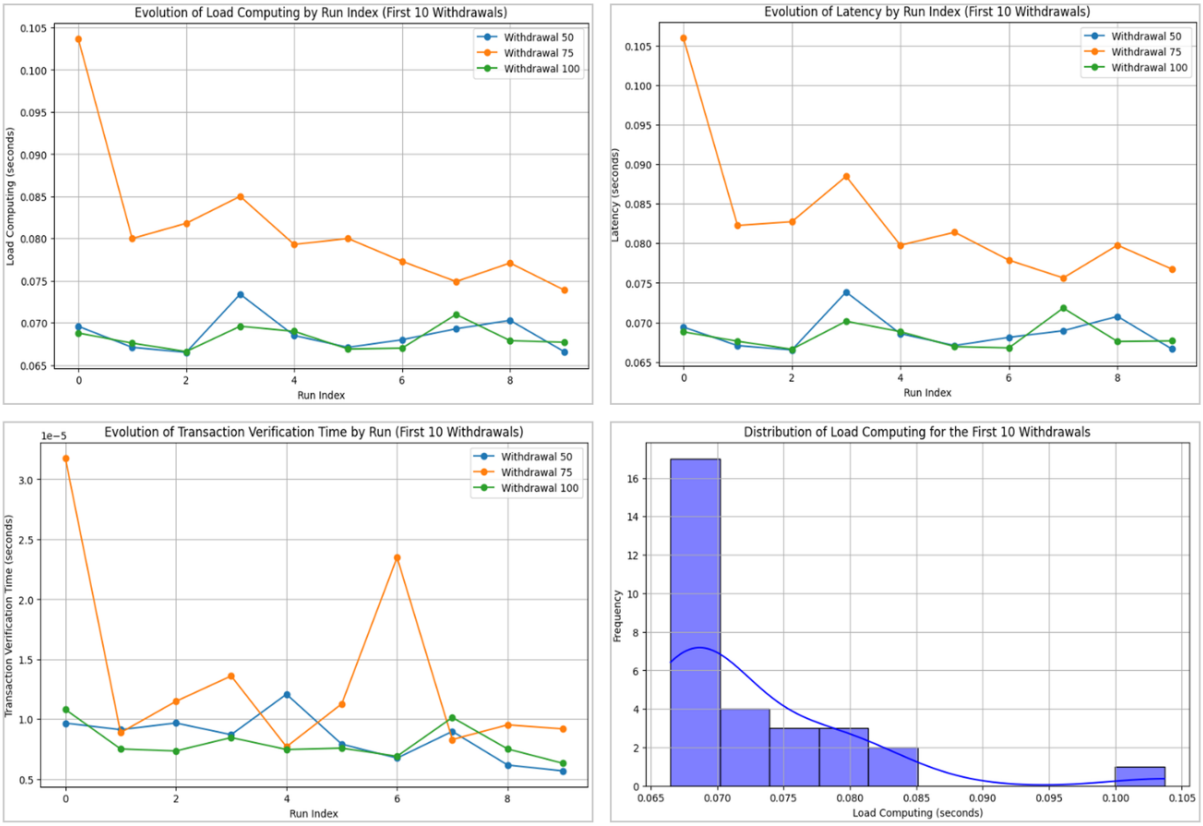


Figure 6.16: Withdrawal results

The distribution of load computing follows a right-skewed histogram, where most runs cluster around lower values (0.065–0.070 s), with a small tail extending toward higher values (0.08–0.10 s), likely due to sporadic cryptographic overhead.

For transaction verification time, times mostly range between 0.5×10^{-5} s and 3.0×10^{-5} s, with occasional spikes in the 75-unit case due to initialization or cryptographic warm-up. Despite these fluctuations, verification times remain consistently low, indicating efficient cryptographic handling.

6.7.2.3 Transaction Settlement

We consider two scenarios. First, Alice pays Bob, representing two users with secured devices. Then, Bob pays Carla. The graphs illustrate six distinct metrics: Latency, ZK Proof Generation Time, ZK Proof Verification Time, Transaction Verification Time, Load Computing, and Throughput.

Table 6.10 shows the results of the transaction settlement between Alice and Bob after 10 runs. Fig. 6.17 shows the graphics of the first five results of these transactions.

Table 6.10: Results of offline transaction simulation (Alice to Bob)

ID	Period	From	To	Amount	Latency	zk Proof Gen.	zk Proof Verif.	Trans. Verif.	Load Comp.	Throughput	Status
51	1740407553.531021	Alice	Bob	50	0.0411	0.0002	0.0001	0.00066	0.0412	23.5768	Success
52	1740407553.57149	Alice	Bob	50	0.0404	0.0001	0.0001	0.00066	0.0404	23.5768	Success
53	1740407553.6117558	Alice	Bob	50	0.0402	0.0001	0.0001	0.00064	0.0402	23.5768	Success
54	1740407553.6525247	Alice	Bob	50	0.0407	0.0001	0.0001	0.00063	0.0407	23.5768	Success
55	1740407553.7004604	Alice	Bob	50	0.0479	0.0001	0.0001	0.00108	0.0461	23.5768	Success
56	1740407553.7439003	Alice	Bob	50	0.0434	0.0001	0.0001	0.00064	0.0430	23.5768	Success
57	1740407553.7855418	Alice	Bob	50	0.0416	0.0001	0.0001	0.00064	0.0416	23.5768	Success
58	1740407553.8264637	Alice	Bob	50	0.0409	0.0001	0.0001	0.00063	0.0409	23.5768	Success
59	1740407553.8684583	Alice	Bob	50	0.0420	0.0001	0.0001	0.00064	0.0420	23.5768	Success
60	1740407553.9144616	Alice	Bob	50	0.0460	0.0001	0.0001	0.00086	0.0455	23.5768	Success

For latency, the observed range is 0.038 s to 0.048 s, with a peak near transaction 55. Minor spikes may arise from local CPU scheduling, ephemeral key generation, or short-range data transfer overhead. All times remain under 50 ms, making them suitable for near-real-time offline payments.

Regarding ZK proof generation time, on a microsecond scale, the values rarely exceed 1.8×10^{-4} s and eventually settle around 1.0×10^{-5} s, indicating a highly efficient ZK scheme. Early spikes may relate to cryptographic initialization.

Conversely, ZK proof verification time shows a steady plateau. The red curve remains between 9.6×10^{-5} s and 1.0×10^{-4} s. This low and stable verification cost ensures good scalability and minimal risk of performance bottlenecks.

For transaction verification time, microsecond-level overhead remains mostly under 1.0×10^{-4} s, with minor variations around runs 56 and 58. Even at peaks, sub-millisecond overhead ensures minimal user-perceived delay.

Regarding load computing, the range is approximately 0.039–0.046 s, with outliers near runs 53 and 58. Higher load computing often coincides with small latency spikes, underscoring CPU usage as a key driver of response time.

Finally, throughput remains steady at 23.5–24.0 TPS, indicating a robust capacity for offline transactions. This implies that the system can handle multiple rapid-fire payments, aligning well with real-world retail demands.

For the payment scenario between Bob and Carla, we focus on average latency, average load computing, and average throughput metrics. Table 6.11 shows the results of this simulation and Fig. 6.18, the corresponding graphic.

Table 6.11: Results of payment simulations between Bob and Carla

Period	From	To	Amount	Latency	Load Comp.	Throughput	Status
1740417109.1297	Bob	Carla	100	1.7949	1.7953	0.5571	Success
1740416652.6854	Bob	Carla	50	6.2916	6.2145	0.1589	Success
1740415986.8613	Bob	Carla	20	3.6511	3.6221	0.2739	Success
1740415751.8391	Bob	Carla	15	1.9801	1.9783	0.5050	Success
1740415405.1866	Bob	Carla	10	9.2675	9.2150	0.1079	Success

For average latency, the dark blue bar (10 units) exhibits the highest average latency, exceeding 8.0 s—a notably large value. Meanwhile, 15 units (teal) drops to roughly 3–4 s, 50 units (turquoise) sits around 5–6 s, and 100 units (lime green) showcases the lowest latency below 1 s. The significantly higher latency for 10 units is counterintuitive and may reflect cryptographic or initialization overhead triggered by small transactions. In contrast, the minimal latency for 100 units suggests beneficial batching or caching behaviors that scale better for larger sums.

Regarding average load computing, the bar for 10 units dominates at over 8 s, whereas 15 is closer to 3–4 s, 50 approaches 6–7 s, and 100 remains near 2 s. This correlates with the latency patterns: high load computing usually indicates longer transaction times. The extreme difference between 10 and 100 suggests that overhead is not strictly proportional to the payment amount but may hinge on internal or cryptographic routines.

Finally, average throughput increases with the amount of transactions. For 10 units (navy), throughput sits below 0.3 TPS; 15 is slightly higher at 0.4–0.45 TPS. The 50-unit bar ranges around 0.35–0.4 TPS, while 100 units soar above 0.65 TPS. The system handles larger payments more efficiently, indicating that repeating small transactions does not boost throughput. Instead, amortized cryptographic or ledger operations favor higher amounts.

These findings reinforce the importance of analyzing payment size effects in offline digital currency systems, guiding future refinements to harmonize performance across all

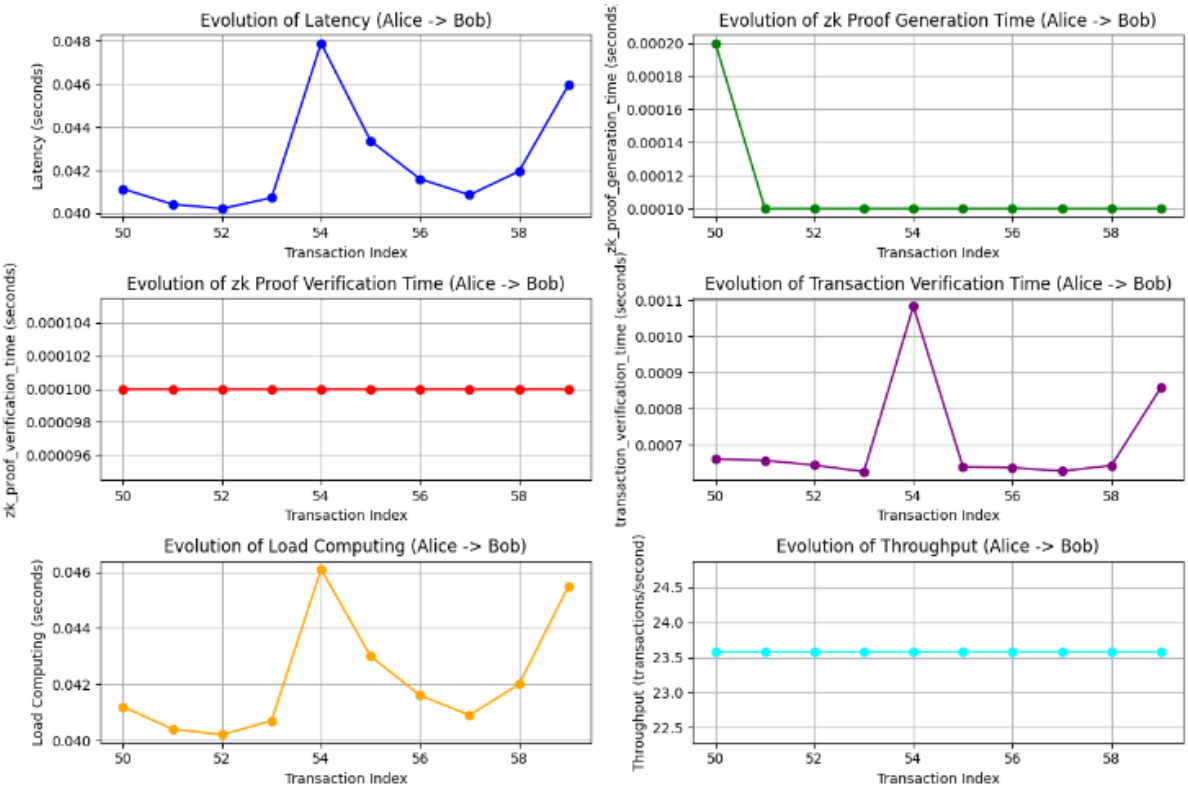


Figure 6.17: Offline transactions performance(Alice-Bob) results

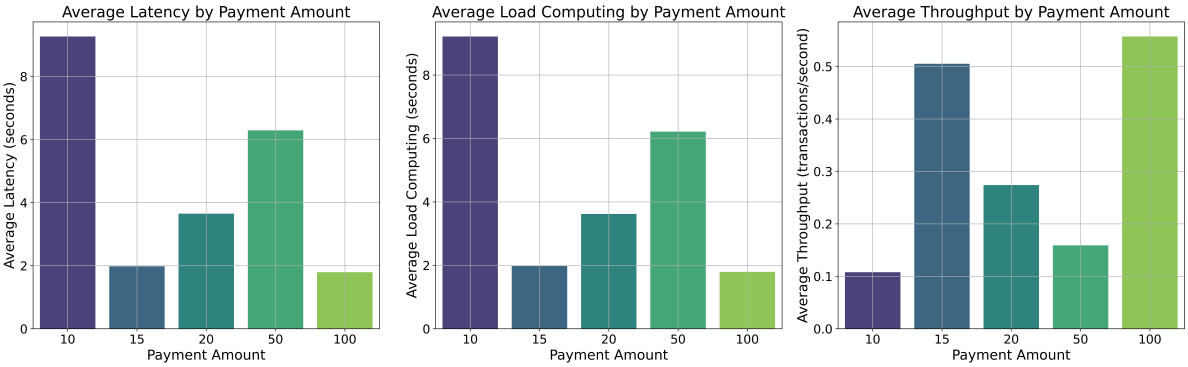


Figure 6.18: Offline transactions performance(Bob-Carla) results

denominations while maintaining robust cryptographic guarantees.

6.7.2.4 Funding Operation

To evaluate funding transactions, we limited our observations to five runs for three different amounts: 300, 700, and 1000. These patterns reveal important aspects of performance and scalability in a potential offline or blockchain-based funding scenario. Table 6.12 shows the results of the funding simulation and Fig. 6.19, the graphical representation of this simulation.

In terms of total time vs. number of runs, the 300-unit (blue) and 700-unit (orange) curves typically range from 0.015 s to 0.030 s, while the 1000-unit (green) curve peaks near 0.040 s at run 3 before abruptly dropping close to zero at run 4. In early runs, cryptographic or setup routines may be triggered, inflating total time for runs 1–3. The sharp dip for 1000 by run 4 might indicate cache priming or resource liberation, while background

Table 6.12: Results of Funding Simulation

Number of Runs	Amount	Total Time (s)	CPU Load	Gas Used
Amount = 300				
1	300	0.021000	0.015625	21776
2	300	0.018000	0.015625	21788
3	300	0.018000	0.015625	21776
4	300	0.017970	0.015625	21788
5	300	0.017000	0.000000	21776
Amount = 700				
1	700	0.024000	0.000000	21776
2	700	0.015996	0.000000	21788
3	700	0.018035	0.015625	21776
4	700	0.017892	0.000000	21788
5	700	0.020010	0.000000	21776
Amount = 1000				
1	1000	0.026010	0.015625	21776
2	1000	0.022995	0.015625	21788
3	1000	0.041524	0.015625	21776
4	1000	0.016106	0.000000	21788
5	1000	0.018000	0.000000	21776

processes or environment-driven tasks may account for minor spikes. Nonetheless, even the highest total time observed ~ 0.040 s remains relatively low for payment contexts, suggesting that the funding operation is not excessively time-consuming.

Regarding CPU load during funding transactions, we observe striking differences. The transaction for 1000 units (green line) stays near 0.016 for runs 1–3, then unexpectedly drops to 0.0 at runs 4–5. The transaction for 700 units (orange line) remains at zero for runs 1–2 before surging to ~ 0.014–0.016 for runs 3–5. The 300-unit transaction (blue line) hovers near zero after the initial runs, implying negligible CPU usage for subsequent iterations. These results indicate: - Resource Allocation or Scheduling: The significant swings from near-max to near-zero CPU usage show that certain runs handle cryptographic tasks or concurrency differently. - Protocol Phases: The transaction amount (300, 700, 1000) may influence concurrency or cryptographic load distribution. Some configurations perform heavy computation early and remain idle in later runs.

Another interesting performance metric is gas usage. All gas values stay within the range of 21776–21788, showing a cyclical pattern but minimal overall variation. The 1000-unit transaction (green) oscillates between ~ 21776 and 21788 across runs, notably at runs 2 and 4. The stable range implies a fairly uniform on-chain execution path, unaffected by run index or parameter changes. Additionally, minor cyclical spikes likely occur when ephemeral checks or ledger updates are triggered. From a cost standpoint, staying within 21780 ± 10 suggests a predictable computational overhead.

Finally, we analyzed transaction efficiency, defined as the ratio of Total Time to CPU Load. The 300-unit transaction (blue) is approximately 0.8-1.4, while the 700-unit transaction (orange) ranges from 0.2 to 2.6, and the 1000-unit transaction (green) peaks near 2.8 before dropping to approximately 0.3. A higher ratio suggests more waiting or I/O time compared to CPU time, implying a less CPU-bound process. Surges to 2.6–2.8 may indicate that the CPU finishes quickly, with the remaining time spent in idle or waiting states. Sharp changes suggest that some runs might be CPU-saturated while others rely on external or concurrency factors.

In conclusion, large efficiency swings highlight protocol phases that are either CPU-bound or I/O-bound. This behavior may be expected in protocols handling cryptographic verifications or ledger interactions in bursts.

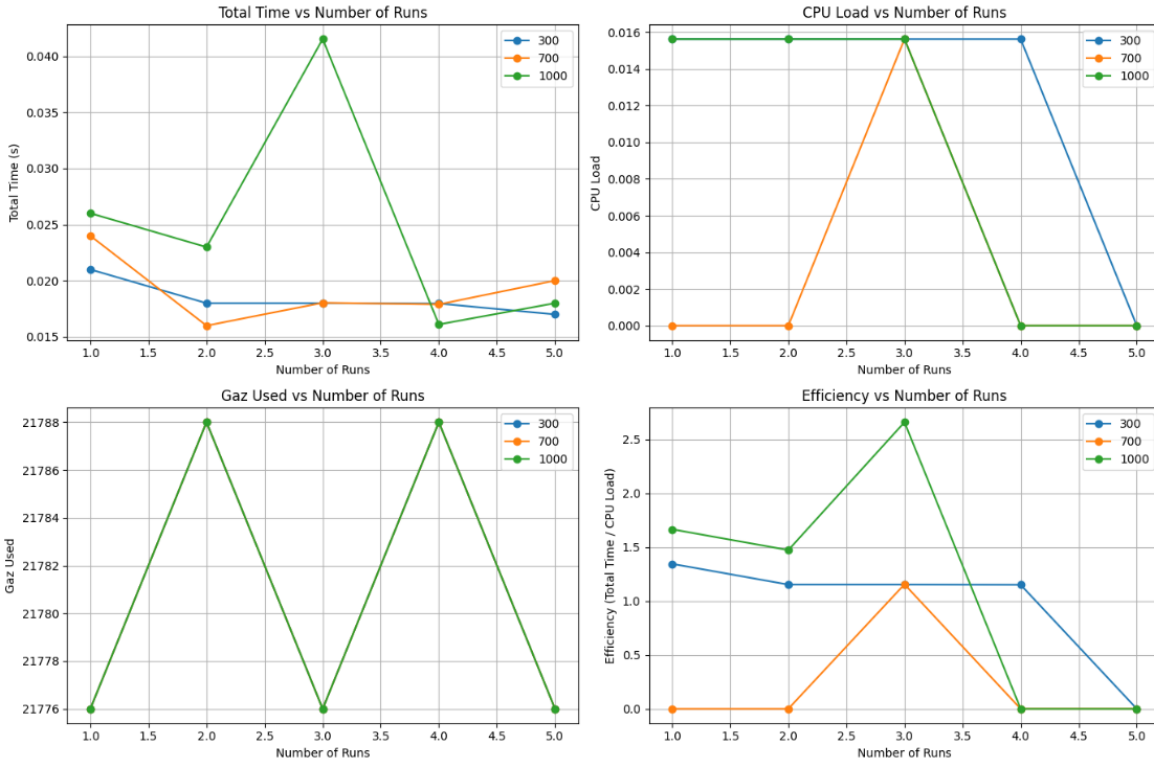


Figure 6.19: Funding transaction performance

6.8 Discussion

The empirical evaluation highlights the efficiency and scalability of the proposed digital identity onboarding and verification processes, as well as the feasibility of offline transaction execution. To further contextualize these results, it is essential to compare our solution with existing offline payment approaches reported in the literature. This comparative perspective provides additional insights into the strengths and limitations of our framework, and sets the stage for a broader discussion on its integration with regulatory and practical requirements.

6.8.1 Comparative Analysis

Table 6.13 provides a comparative view of our proposed Block-PAD solution against existing offline payment frameworks. Several solutions report transaction times in the order of hundreds of milliseconds to seconds (e.g., [49], [50]), which can be limiting in retail environments where low-latency payments are required. Other works such as [53] demonstrate the integration of advanced privacy-enhancing technologies but incur significant performance costs, with latencies around 210 ms per payment.

High-throughput protocols, notably [66], achieve tens to hundreds of transactions per second. However, these approaches rely on specific assumptions, such as access to trusted online services (e.g., timestamp servers), which limit their applicability in full offline deployments.

The DigiVault and MarkoPayChain solution [103] shows encouraging results, with latencies between 10–30 ms, throughput of 45–50 TPS, and proof operations (zk-SNARK) remaining lightweight (generation $\sim$0.011 s, verification $<$0.015 s). However, the observed success rate is around 88–90%, which highlights trade-offs between speed and reliability

in constrained environments.

Additional schemes such as Luo and Yang's EMV-compatible protocol [108] and Igboanusi's Pure Wallet architecture [44] highlight other trade-offs. EMV-compatible designs achieve latencies of about 100–150 ms but remain hardware dependent, while Pure Wallet emphasizes programmability, though blockchain-side synchronization constrains scalability in fully offline contexts.

In contrast, Block-PAD achieves a stable throughput of 23.57 TPS with a latency consistently around 40–47 ms and near-100% success rate. While the throughput is moderate compared to DigiVault/MarkoPayChain or highly optimized systems, the combination of low latency, adaptive privacy features, and genuine full offline feasibility makes our solution well-suited for retail and micro-payment use cases. This balance between performance and deployability highlights Block-PAD's practical value as a resilient offline payment framework.

Table 6.13: Comparative performance evaluation of offline payment solutions.

Solution	**Performance (Latency / TPS)**	**Limitations**	**Remarks**
Ref[49]	112–568 ms (comm.), up to 1590 ms (20 blocks / 60 nodes), $<$ 1 TPS	Weak privacy, secure element dependency	Transaction time exceeds typical retail requirements.
Ref[50]	Withdraw: $<$400 ms; Spend: 275–1600 ms	High receiver cost (bilinear pairing)	Not suitable for rapid consecutive offline payments.
Ref[53]	$\sim$210 ms/payment; $\sim$90 ms acceptance	Large cryptographic proofs	PET integration degrades performance.
Ref[66]	548 ms $\rightarrow$ 91.24 TPS (simulated)	Needs timestamp server; energy cost not detailed	High throughput, but not full offline.
Ref[103]	10–30 ms latency; 45–50 TPS; 88–90% success; zk-SNARK: gen. 0.011 s / verif. $<$0.015 s	Slightly reduced success rate under stress	Strong balance of speed and cryptographic robustness, but not perfect reliability.
Ref[108]	$\sim$100–150 ms/transaction (NFC)	EMV dependency, limited privacy	Good EMV compatibility, but slower than Block-PAD.
Ref[44]	Few tens of TPS (not precisely quantified)	Requires blockchain-side synchronization	Highlights programmability but limited in full offline context.
Our work (Block-PAD)	40–47 ms latency; 23.57 TPS (stable); $\sim$100% success	Throughput moderate vs. optimized systems	Balanced trade-off: low latency, stable throughput, adaptive privacy, full offline feasibility.

6.8.2 Full Consecutive Offline Payments Integrated Blockchain (Alice, Bob, Carla)

Offline payment in rCBDCs enhances resilience during technical failures and reduces network load, improving scalability by decreasing peak transaction volumes [74]. Intermittently offline payments that synchronize with the ledger are recommended for security. DLT systems like blockchain facilitate consecutive offline payments. For example, [43] detailed an offline method where Alice pays Bob with a secure device, and Bob can pay Carla, who connects to a blockchain-centric PAD without a device. The Block-PAD system enables unlimited consecutive offline payments with immediate finality and end-to-end privacy. However, integrating blockchain into CBDC protocols requires considering motivations.

6.8.3 Reconciling Privacy and Regulatory Constraints

Privacy is crucial for CBDC adoption but faces challenges due to security and regulatory requirements against fraud. Increasing security often reduces privacy, while enhancing privacy can hinder traceability for detecting illicit activities. We believe privacy and regulatory compliance can be reconciled through a secure, private, and legally compliant payment system based on specific principles. Privacy should be the default mode for all transactions, ensuring both payer and payee anonymity. This approach strengthens data protection by limiting exfiltration. Minimal user information should be retained during onboarding, transactions, and data access by system entities, as discussed in [125] and developed in [81]. In the Block-PAD, each entity has minimal access to user information, facilitated by digital identity, ensuring privacy from coin withdrawal onwards. Transactions remain anonymous unless there is a proven suspicion of illicit activity.

Private or public blockchains do not inherently guarantee privacy. As [126] warns about "the mistake of pseudonymity" and [127] highlights the risk of "de-anonymization on blockchains," privacy issues are significant. PETs play a crucial role in addressing these concerns while ensuring regulatory compliance. The Block PAD employs cryptographic tools such as blind homomorphic encryption and zero-knowledge proofs within a trusted execution environment to maintain high levels of privacy.

6.8.4 Hybrid Paradigm Model Based on Regulatory Compliance

When designing a CBDC system, the choice of authentication method is key. A hybrid CBDC system can overcome the limitations of token-based and account-based systems by combining their strengths. This approach offers immediate finality and scalability, which account-based systems lack, and avoids the caps or limits associated with token-based systems. As discussed by [119] and described as "signed balance update" by [128], the Block-PAD system effectively integrates both authentication methods to optimize their benefits.

6.9 Conclusion

In this chapter, we introduced the Block-PAD framework, a blockchain-enabled architecture designed to support resilient and privacy-preserving CBDC transactions. Block-PAD addresses a broad spectrum of challenges related to CBDC deployments, including privacy, security, interoperability, regulatory compliance, and resilience in scenarios with

limited or intermittent connectivity. By integrating digital identity into the transaction workflow, the framework ensures that offline and hybrid payment flows can be executed reliably while remaining compliant with accountability requirements.

Our experimental evaluation, conducted over multiple independent runs, confirmed that Block-PAD achieves consistently low per-user onboarding and verification times, stable throughput (23.57 TPS), and latency in the range of 40–47 ms for autonomous device-to-device payments. These results demonstrate the practical feasibility of real-time retail payments in constrained offline environments. Furthermore, the system scales linearly in terms of total onboarding time while maintaining negligible per-user costs, thus confirming its robustness across increasing user volumes. In addition, the integration of privacy-enhancing technologies (zk-SNARKs and homomorphic encryption) proved efficient, with minimal computational overhead, allowing both privacy protection and regulatory auditability without compromising performance(6.12 and 6.7). To contextualize these results, we also provided a comparative analysis with existing CBDC offline payment solutions. While some state-of-the-art protocols achieve higher throughput under specific assumptions (e.g., reliance on timestamp servers or fixed user sets), Block-PAD balances low latency, adaptive privacy, and genuine offline feasibility, making it particularly suitable for retail and micro-payment use cases.

Despite the robustness of the protocol, fraud remains a critical dimension requiring in-depth analysis and complementary advanced tools. The following chapter directly addresses this challenge. It focuses on the systematic integration of advanced fraud protection technologies in offline payments. Noticeably by formulating a comprehensive framework designed to identify and block cases of fraud that cannot be detected cryptographically.

6.10 Chapter Summary

This chapter introduced the Block-PAD framework, a blockchain-enabled architecture designed to ensure resilient CBDC transactions in disconnected environments. First, it outlined the conceptual foundations of the system: role distribution, digital identity management, and local value circulation without immediate access to a ledger.

Block-PAD demonstrated that both interoperability and operational flexibility can be achieved under a hybrid mode full offline and staged offline. This hybridity rests on fortified cryptographic guarantees enabled by verifiable identities, transferable tokens, and on-chain synchronization protocols grounded in contextual privacy. Furthermore, the analysis highlights the security and privacy properties realized: resistance to double spending, contextual privacy, and robustness of the model against DDoS-style attacks. The Block-PAD framework thus convincingly shows that privacy and auditability can be combined within a secure offline CBDC ecosystem.

Chapter 7

Advanced Fraud Mitigation

7.1 Introduction

The preceding chapter demonstrated that fraud protection in CBDC offline payments can be ensured by a robust cryptographic architecture incorporating secure enclaves, blind signatures, zero-knowledge proofs, and PAD system. Nevertheless, even when security is guaranteed cryptographically, offline payments remain intrinsically vulnerable to specific attacks such as time manipulation or the use of compromised terminals. The absence of an immediate connection to the ledger creates a window of opportunity for fraud that may escape both cryptographic schemes and basic Markovian fraud detection models. To complement the security guarantees presented in earlier chapters it becomes necessary to address automatic and explainable detection of fraudulent behaviour in these disconnected environments. Hence, this chapter aims precisely at tackling this issue by introducing a lightweight, hybrid and explainable fraud-detection framework for CBDC offline payments. The objective is twofold: first, to design an approach capable of operating locally on constrained devices without connectivity; second, to provide decision transparency compliant with auditability requirements.

To that end, the proposed framework combines a deterministic rule engine with quantified machine learning models, coupled with a suite of post-hoc explainability tools (SHAP, LIME, etc.). The chapter is organised as follows. It first presents the methodological challenges posed by the lack of real data. Next, it introduces `Offsim_CBDC`, a simulator for offline transactions that yields an offline transaction dataset. Then it describes the adversary model and the taxonomy of frauds specific to offline payments. The central section details the framework architecture: rule engine, learning pipeline construction, feature engineering and model calibration. Finally, subsequent sections expose experimental results, comparative analysis, and discuss contributions, limitations and future directions. The remainder of this chapter is organized as follows: first, Section 7.2 presents the background and related work. Next, Section 7.3 highlights the problem formulation. Section 7.4 introduces *OffSim*: Agent-Based Simulation for Offline CBDC Fraud Detection. Section 7.5 presents the Hybrid Explainable Rule and Machine Learning Framework for Offline CBDC Fraud Detection. Section 7.6 introduces Hidden Markov chain Model-enhanced Fraud Detection for Offline CBDC. Section 7.7 relates to the discussion, while Section 7.8 and Section 7.9 present the conclusion and the chapter summary, respectively.

7.2 Background and Related Work

The problem of financial fraud is as old as the very existence of payment means. Fraudsters continually compete with ingenuity to counteract the security mechanisms of payment systems. These fraudulent operations often lead to significant financial losses, both for end-users and enterprises. At the beginning of 2025, the "Observatoire de la sécurité des moyens de paiement" published statistics on fraud in the first half of 2024[129]. The publication reveals that the number of fraudulent operations is rising, reaching 3.9 million. Alongside these accounting catastrophes lies the risk of reputational erosion for the entities responsible for ensuring transaction and wallet security. Within the realm of CBDC transactions, the specter of fraud is especially menacing. The new dimension of digital currencies and offline payment introduces radically novel threats. In this context, the need to identify anomalies and prevent fraud becomes imperative particularly because offline fraud detection must operate locally. Thus, the framework proposed in this chapter focuses on developing and evaluating an advanced offline fraud-detection solution for CBDC transactions. This approach aims to establish a significant milestone toward convergence between explainable artificial intelligence and financial technologies. Fraud detection in financial transactions remains a major concern. Alongside this, fraud detection techniques evolve at the pace of innovative operational strategies employed by fraudsters. This evolution is also marked by the transition from centralized detection models to those tailored for decentralized environments. On reflection, a comprehensive taxonomy of fraud detection methods highlights three main paradigms: Knowledge-based systems, Data-driven systems, and Adaptive hybrid systems.

Initially, the first systems relied on expert knowledge and deterministic rules. These aimed to identify behavioural deviations or suspicious patterns, and they offered the possibility of extracting interpretable rule sets from decision trees[130].Consequently, following business-logic by domain experts is greatly facilitated. Rule-based systems remain essential in the payment industry due to their transparency, low latency, and auditability. However, these static systems are fragile against adversarial behaviours and lack adaptability. Subsequently, data-oriented approaches emerged. They largely stem from statistical techniques, anomaly detection, and probabilistic models. Probabilistic sequential models such as Hidden Markov Models (HMMs) capture temporal dependencies in purchase behaviour[109], achieving roughly 80% accuracy on synthetic data. This ability to model complexity enhances robustness against evolving fraud strategies. Despite these advances, many of these models rely on coarse quantification of observations, continuous data streams, and unsupervised learning. More recently, some initiatives based on offline CBDC-oriented architectures have also incorporated Markov chains[103]. Yet those are strictly protocol-level and aim only at double-spending prevention by combining transaction categories (low/medium/high) with fraud/non-fraud states.

Furthermore, machine learning paradigms have proven effective in detecting financial fraud, particularly for credit-card and e-commerce fraud detection[131]. Supervised methods such as random forests, gradient boosting, and support vector machines are routinely applied to online commerce and finance systems[132]. These require large volumes of labelled data and continuous feature updates. Although unsupervised or semi-supervised approaches (e.g., isolation forests, autoencoders) mitigate the lack of labels, they often suffer from high false-positive rates. Moreover, these machine learning solutions typically remain tied to connected architectures and centralized real time streams, rendering them poorly transferable to offline transactions. The emergence of explainable AI (XAI) addresses transparency requirements in regulated environments that demand human over-

sight. It has led to hybrid models combining interpretability with adaptive learning[132], leveraging tools such as SHAP or LIME to explain black-box models. However, these explanatory models and tools have yet to be tested in an offline payment context. Recent research in TinyML and embedded AI shows significant progress[133, 134]. Thus, deploying models on constrained devices becomes feasible through quantisation, model reduction, and knowledge distillation[132]. Nonetheless, their application to offline scenarios remains insufficiently demonstrated.

In summary, analysis of existing work highlights four essential dimensions: connectivity, transparency, adaptability, and deployability. Deterministic systems are interpretable but static; machine learning oriented systems perform well yet demand centralized resources; embedded approaches enable on-device inference for low power applications but still rely on intermittent synchronisation. Consequently, no solution currently reconciles explainability, adaptability, and offline operation in secure CBDC ecosystems. Against this backdrop, the detection system proposed in this framework seeks to unify interpretability and adaptability under stringent resource constraints.

7.3 Problem Formulation

The problem formulation of this framework tackles the multidimensional challenge of deploying an offline CBDC fraud detection solution. The first dimension of this problem is inherent to scarcity. Indeed, every good detection model requires a complete, diverse and targeted dataset at its core. Given that central bank digital currencies are still in an embryonic development phase, there is no public dataset for offline CBDC transactions. The few datasets that are publicly available focus on online card payments and cannot capture peer-to-peer flows or dual-offline settlement semantics. This primary concern inevitably forces the consideration of a synthetically generated dataset via a simulator.

Thus, the second branch of this framework's problem concerns the creation of a simulator capable of synthetic realism. The custom simulator must emulate realistic agent behaviours (consumer, merchant, etc.) while embedding fraud scenarios. Moreover, it must be able to incorporate into the dataset the essential features of an offline transaction. Resource constraints are also a prime aspect that must be taken into account for a lightweight implementation. Indeed, detection logic must fit within smartphone grade CPUs and secure elements. On top of this dimension lies the need for explainability and calibration. Every decision must include an audit trail and produce calibrated probability scores to interface with offline risk throttles embedded in the wallet. In short, all these concerns must be accompanied and supported by two performance objectives: high accuracy and sequential robustness.

7.4 OffSim: Agent-Based Simulation for Offline CBDC Fraud Detection

Building a fraud detection system that incorporates a machine-learning model presupposes an appropriate dataset. In the offline payment context, such a dataset should be able to highlight the inherent or essential features of user interactions. However, at present there is virtually no dataset dedicated specifically to offline payments. Existing alternatives each have significant limitations. Anonymised banking datasets (e.g., credit card transaction records) are not suitable for this context because they are based on centralised

infrastructure. Cryptocurrency data such as Bitcoin or Ethereum transaction logs are too tied to blockchain-specific mechanisms and therefore irrelevant for modelling offline peer-to-peer transactions. E-commerce or retail datasets focus primarily on purchases rather than P2P transfers or balance management, making them of little use in an experimental environment. Given all these limitations and obstacles, it becomes necessary to obtain a realistic synthetic dataset via a simulator.

7.4.1 Model and Method

The simulator proposed in this framework is inspired by *Paysim*. Paysim is an open-source money-transfer simulator that incorporates various fraud behaviours. Its key strength lies in its ability to realistically simulate financial transactions within a configurable environment. In this line, data are not generated arbitrarily; they are derived from observations of Kenya's M-Pesa system [7]. This is a decentralised, mobile, peer-to-peer payment ecosystem. Consequently, transaction frequencies, user profiles, and even fraud patterns in Paysim are calibrated to reflect real world behaviours. Furthermore, the M-Pesa system operates without heavy banking infrastructure and without the need for an internet connection. This makes it a uniquely suitable payment mode, especially in rural areas where internet is scarce or absent.

Subsequently, because of this usage context and its ability to replicate the social and transactional diversity observed in reality, Paysim provides the essential building blocks for deploying an offline CBDC transaction simulator. Mobile-money payment systems such as M-Pesa or Orange Money have demonstrated operational effectiveness in low banking infrastructure contexts, particularly in Kenya. By targeting these same emerging and heavily underbanked markets, offline CBDCs reduce dependence on a single often inaccessible system. That said, the methodology adopted for designing the multi-agent simulator OffSim is built upon a delicate balance between fidelity to realistic behaviours, flexibility for experimentation, efficiency for potential deployment on constrained devices. This subsection highlights the three-step process that guided the construction of OffSim-CBDC.

7.4.1.0.1 Step 1: Fraud Patterns in Offline CBDC Contexts

The first step focuses on identifying fraud patterns that are specific to offline CBDC settings. This identification is made possible by an exhaustive literature review of the state of the art, which includes academic papers, central bank publications and industry reports outlining offline fraud scenarios. The analysis revealed fraud cases that do not appear in traditional mobile money systems or online banking.

7.4.1.0.2 Step 2: Mapping Fraud Relevant Features to Simulator Logic

After the fraud identification phase is complete, the second step translates those fraud schemata into observable features and internal agent states. For example, an instance of double spending fraud is manifested as a mismatch between a user's offline transaction volume and their actual balance. Likewise, device manipulation behaviour can be inferred from abnormal synchronization delays; such delays are typically quantified via statistical z scores across simulation steps. These correspondences guided the definition of the attributes tracked by each agent and the structure of transaction records.

7.4.1.0.3 Step 3: Extending the PaySim Schema with Backward Compatibility

The `Offsim` simulator reproduces the structure and logic of PaySim, yet it diverges considerably from it. The transaction frame is extended to include new fields such as `is_offline`, `device_timestamp`,`sync_delay_minutes`, `double_spend_flag`, and `is_fraud_off`. All new fields are integrated following the original CSV format and naming conventions produced by PaySim. This seamless integration guarantees backward compatibility with PaySim-based tools or benchmarks. Table 7.1 compares the features and logic of OffSim versus PaySim.

Table 7.1: OffSim-CBDC leveraging PaySim features and logic

Aspect	**PaySim**	**OffSim-CBDC**	**Comment**
Agent-Based Modeling	Yes	Yes	Both simulate individual agents' behavior over time.
Roles (customer, merchant, fraudster)	Yes	Yes	Roles are assigned and behaviors differ by role.
Transaction Types	Yes	Yes	Types like `CASH_IN`, `CASH_OUT`, `PAYMENT`, etc.
Offline Behavior Simulation	No	**Yes**	New in OffSim for offline CBDC modeling.
Fraud Injection	Yes	Yes	Both inject fraud under specific conditions.
Random Activation of Agents	Yes	Yes	Both randomly activate agents at each step.
Export to Dataset	Yes	Yes	Both output transaction logs to structured files.

7.4.2 Simulator Overview

The OffSim simulator follows a modular architecture. It relies on several core components: a JSON-based configuration loader, a rule-based fraud engine, and a persistent CSV writer. OffSim also maintains an internal state that captures the interactions among agents. Fig. 7.1 visualises the state-level flow of the simulation. It highlights real-time transaction generation via the `step()` call and the central role of the Simulation Core and its interactions with the agent layer. Fig 7.2 shows the class-level architecture. The diagram illustrates the class hierarchy that underpins the modelling of key actors in the system. Actors such as `Client`, `Fraudster`, `Mule` and `Merchant` inherit from a common parent class, `Client`. This inheritance unifies the properties shared by all agents while allowing specialised behaviours through overriding the `step()` method. Each subclass adapts this method to reflect its operational logic, whether legitimate or fraudulent. The architecture also incorporates composition via the `DynamicClientProfile` class, granting behavioural modularity to agents. Consequently, agents gain flexibility and scalability when faced with varied scenarios. In practice, this approach supports two complementary paradigms:behaviours driven by learning algorithms and stochastic fraud scenarios.

Fig. 7.3 presents a view of the fraud generation pipeline. At its core, the generator loads configuration parameters from a JSON file (`config.json`). The profiles of all agents (clients, fraudsters, mules, merchants and banks) are created by incorporating their respective fraud probabilities. Transaction generation proceeds step-by-step; the associated metadata are persistently stored in a CSV file. These metadata, including balance, offline

status, fraud labels, etc., enable explicit tracking of fraud indicators (double spending, device sync delay, reputation signals) in offline mode. Overall, the model illustrated by these visualisations demonstrates its ability to replicate the dynamics and interactions of real-world users in both offline and online payment modes. All of this is made possible thanks to the multi-agent framework `Mesa` [135].

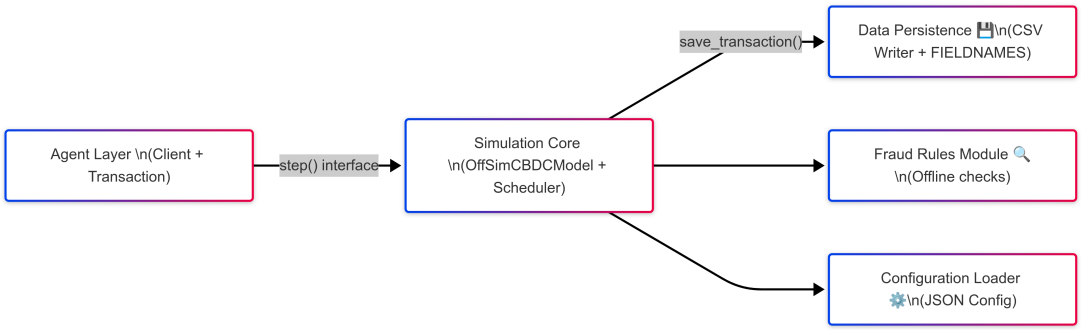


Figure 7.1: State-level architecture of OffSim-CBDC showing the main modules and interactions.

7.4.3 Parameters and Configuration

The simulator was implemented using the open-source Mesa 2.1 library. All source code, documentation and the generated dataset are publicly available on Kaggle[136]. The baseline configuration of the simulator was manually calibrated to reproduce the behavioural diversity and complexity of a realistic payment ecosystem over a 30 day horizon, i.e., 744 steps corresponding to 744 hours. This periodicity allows us to observe agent behaviour over a meaningful temporal window, to visualise daily fluctuations and emerging trends on a monthly scale. The simulated ecosystem contains a representative population of 30 000 clients, among which 3 000 (10%) are fraudsters and 1 000 merchants. Each client's balance evolves randomly within the interval [0, 100 000] units. Transactions cover five common operation types: `CASH-IN`, `CASH-OUT`, `DEBIT`, `PAYMENT` and `TRANSFER`. The latter is capped at 200 000 monetary units. These limits and transaction types were defined to constrain financial interactions. Fraud scenarios are triggered according to a probabilistic distribution: the online fraud probability is set to 10% while the offline fraud probability is 20%. The higher offline risk reflects the greater danger posed by attacks that exploit the lack of continuous network supervision. This configuration is deliberately simple and can be adjusted for different experimental scenarios.

7.4.4 Threat Model and System Assumptions

The proposed offline payment environment follows a two-tier architecture. User-level payments involve peer-to-peer or client-to-merchant transactions, orchestrated through secure mobile wallets that do not rely on constant connectivity to the central ledger. Each device stores locally a limited set of tokens and metadata unique identifiers, an operation history, and potentially sophisticated cryptographic proofs such as zero-knowledge signatures. We assume an attacker can obtain physical or software access to a client's or merchant's device. This enables the attacker to tamper with stored values or extract transaction metadata. Because the offline period is unsupervised by the central network, the attacker gains a critical temporal advantage and may attempt malicious transactions during this window. Moreover, the adversary possesses partial knowledge of fraud-detection

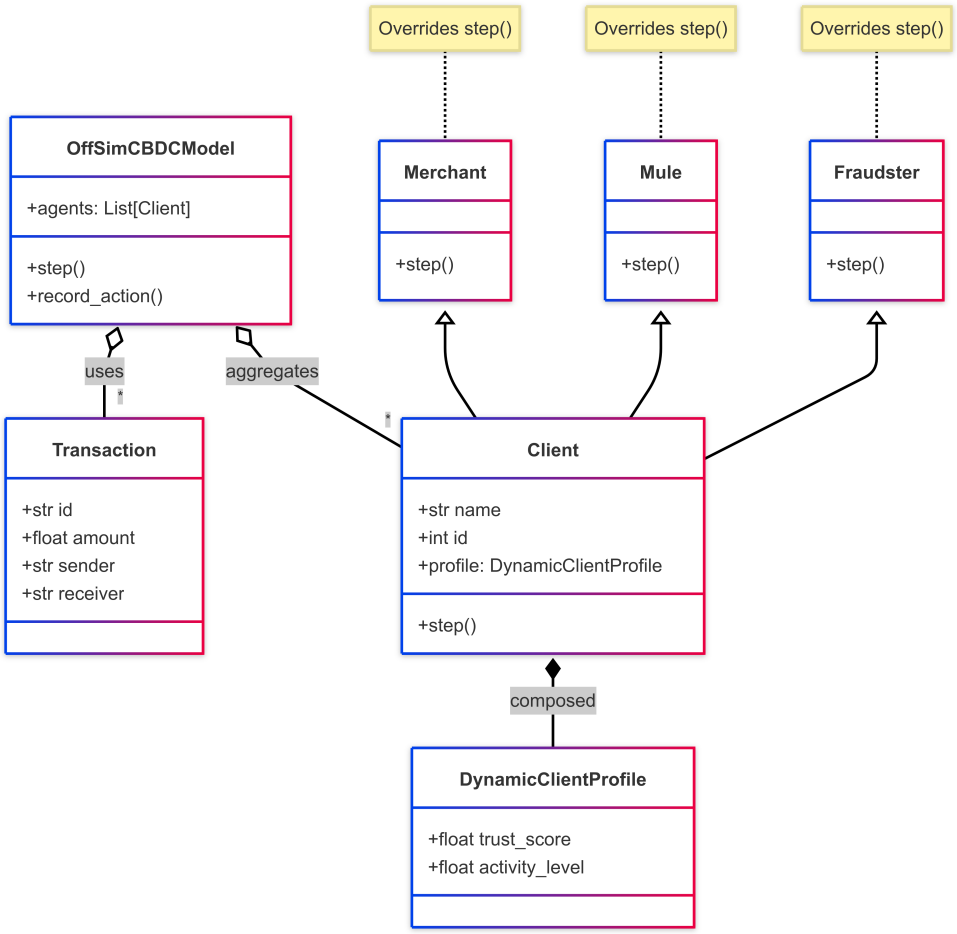


Figure 7.2: Class diagram of the OffSim-CBDC agent system.

rules or model characteristics, allowing replay attacks or cloning techniques that exploit stolen identifiers or tokens.

In short, the adversarial model can be formalised as follows. We consider an offline central bank digital currency system composed of two distinct sets of actors: the set of offline users denoted by $\mathcal{U} = u_1, u_2, \ldots, u_n$ and the set of merchants denoted by $\mathcal{M} = m_1, m_2, \ldots, m_k$. Each participant $x \in \mathcal{U} \cup \mathcal{M}$ is equipped with a secure device $\mathcal{D}x$ that locally maintains a wallet balance $B_x(t) \in \mathbb{R}_{\geq} 0$ at time t, a transaction log $\mathcal{T}_x = T_1^x, T_2^x, \ldots$, an optional set of cryptographic proofs $\Pi_x = \pi_1, \ldots, \pi_j$, and a local clock $\tau_x(t)$ that may drift from global time. The adversary $\mathcal{A}$ is modeled as a powerful entity capable of (i) tampering with $\mathcal{D}_x$ by modifying local state variables such as the balance or the clock, i.e., $B_x(t) \leftarrow B_x(t) + \delta$ or $\tau_x(t) \leftarrow \tau_x(t) + \Delta t$, thereby violating spending or timing constraints; (ii) mounting replay attacks by reusing a transaction $T = (x, y, a, t, \pi)$ to construct a forged transaction $T' = (x, y', a, t', \pi)$ with $y' \neq y$; (iii) cloning a legitimate device $\mathcal{D}_x$ into multiple physical instances $\mathcal{D}_x^{(1)}, \ldots, \mathcal{D}x^{(r)}$ in order to attempt double spending during an offline session; and (iv) performing knowledge-limited evasion attacks by exploiting partial knowledge of the fraud detection rule set $\mathcal{R}$ and feature space $\mathcal{F}$ to strategically craft transactions $T^* = \arg\min T \in \mathcal{T} \Pr[\text{Detection} \mid T, \mathcal{R}, \mathcal{F}]$.

The purpose of the fraud detection system $\mathcal{FDS}$ is to detect or deter such attacks during offline sessions, i.e., when $x \notin$ Connected, using only local features and without leaking sensitive user data. Formally, the detector operates over a local observation space $\mathcal{O}_x(t)$ and outputs a verdict $\mathcal{FDS}_x(\mathcal{O}_x(t)) \in \texttt{Legitimate}, \texttt{Suspicious}$ under strict

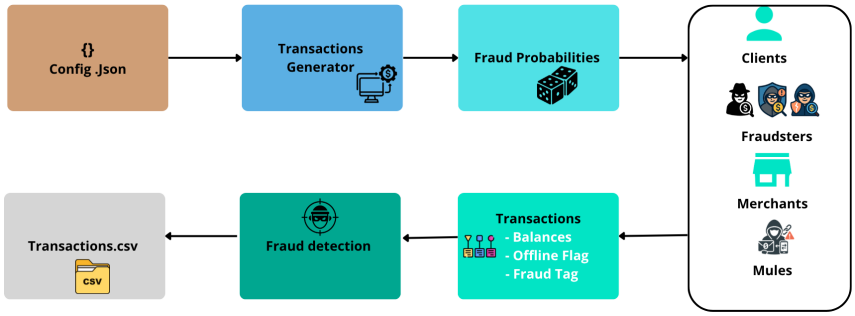


Figure 7.3: Fraud oriented transaction generation pipeline in OffSim-CBDC.

constraints on latency: $\mathcal{FDS}_x$ runs in $\leq \Delta t_{\max}$, privacy preservation: $\mathcal{O}_x(t)$ excludes personally identifiable information (PII), and auditability, ensuring that all alerts can be traced back to explicit rule activations or SHAP-based attribution.

This decision is part of a rigorous operational framework characterized by three major constraints. From a technical perspective, the system must ensure a rapid response. In other words, in terms of latency, the system $\mathcal{FDS}_x$ must execute within the time constraint $\leq \Delta t_{\max}$. Regarding privacy, $\mathcal{O}_x(t)$ *de facto* excludes all personally identifiable information (PII). Finally, in terms of regulatory compliance, the system integrates the dimension of auditability. Thus, all alerts are traceable and justifiable through rule correspondences or SHAP-based attribution.

7.4.5 Fraud Taxonomy in the Proposed Offline CBDC System

The threat model described earlier shows that offline payment introduces new fraud scenarios. These emerging frauds can be categorised and explained using the following nomenclature:

7.4.5.1 Double Spending

Double spending is the most immediate and damaging threat. It occurs when a user's wallet fails to mark a coin or token as spent, or when malicious users disable device level checks. To mitigate this risk we propose a robust detection strategy that combines pre-transaction indicators with contextual anomaly detection to flag suspicious transaction bursts (e.g., SHAP-explainable spikes).

7.4.5.2 Fake Merchant or Phantom Terminal

In this scenario the fraudster pretends to be a legitimate merchant in order to capture digital tokens or coins. Such attacks are common in offline peer-to-peer markets that do not require high inter-party trust. The detection strategy relies on rule-based validation of merchant reputation.

7.4.5.3 Timestamp Tampering

This attack targets the device hardware by manipulating the secure element's clock to bypass per-period spending limits. Countermeasures include a strict verification of device integrity and a hard threshold on allowable time-clock drift.

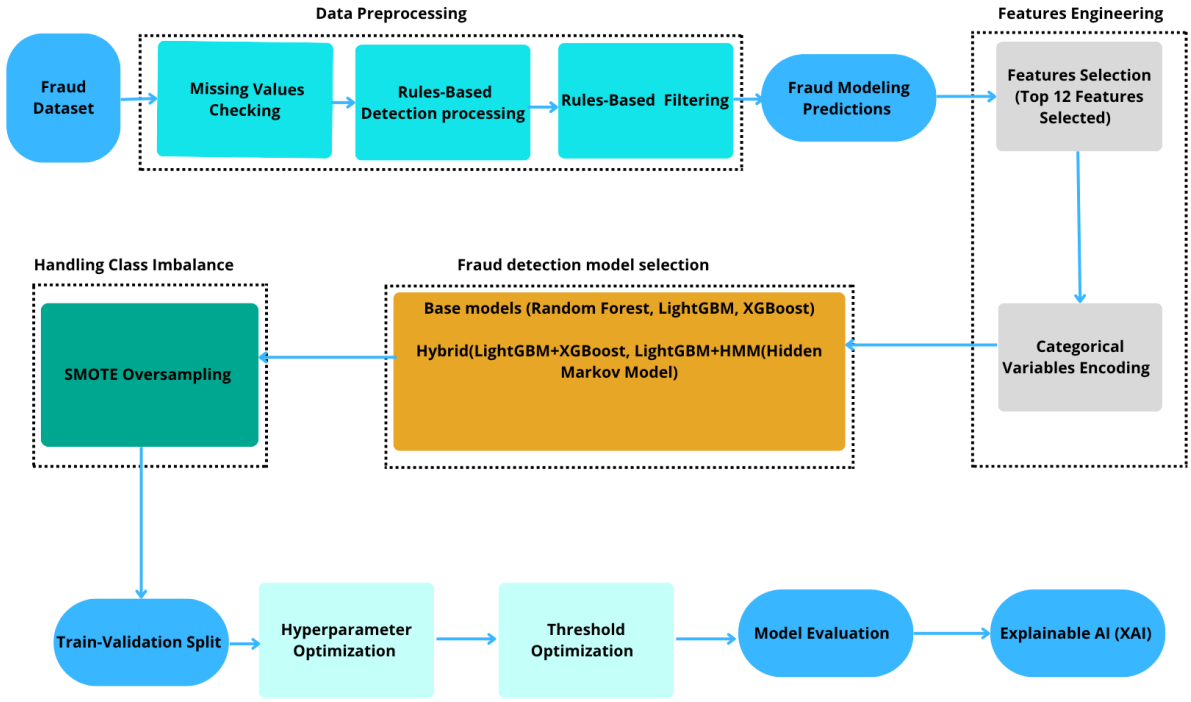


Figure 7.4: The proposed fraud detection framework

7.4.5.4 Micro-Top-Up Storms

The fraudster rapidly funds and drains offline balances to evade AML thresholds. The proposed system monitors the last 24 hours of peer-to-peer and top-up activity for each payer using two ring buffers, thereby detecting abnormal burst patterns.

7.5 Hybrid Explainable Rule and Machine Learning Framework for Offline CBDC Fraud Detection

7.5.1 Methodology

The proposed fraud-detection solution is organised around three dynamic axes: data pre-processing, model training, and system evaluation. In the preprocessing phase, we cover the following steps: data cleaning, exploratory analysis, rule filtering, feature engineering, and data pre-processing (the term "pre-processing" itself being used twice in the original text). The training phase focuses on selecting models that meet deployment criteria such as inference latency and explainability. Finally, the evaluation phase is concerned with identifying and evaluating key metrics of the system (precision, recall, F1-score, AUROC, etc.). Each of these stages will be detailed in the following subsections. Fig. 7.4 highlights the proposed fraud-detection framework.

7.5.2 Experimental Setup

The entire experiment was carried out on a standard laptop (no GPU accelerator) equipped with an Intel Core i7-1265U 12th-generation processor and 16 GB of RAM. This technical base was complemented by a modern, open source software stack. In particular, we used JupyterLab 4.1 running inside a `conda` distribution with the conda package manager

based on CPython 3.11. The main libraries employed throughout all phases of the experiment were were `numpy 1.26`, `pandas 2.2`, `scikit-learn 1.5` and `matplotlib 3.8`. For reproducibility, the notebooks and source code were archived on Kaggle [136].

7.5.3 Dataset Processing

The dataset used in this study originates from the Offsim simulator mentioned in Section 7.4. It contains roughly six million synthetically generated financial transactions that simulate interactions between clients and merchants in an offline digital currency environment. Reference [136] highlights its structural richness and versatility. Indeed, it is specifically designed to meet both the evaluation requirements of machine learning models and rule-based business process approaches. Consequently, this multidimensional organization allows a fine analysis of offline fraud patterns from multiple angles, thereby enabling comparison across heterogeneous methods. The essential target variable for fraud detection is `is_fraud_off`, which indicates whether a transaction is fraudulent in an offline context. Naturally, this target encapsulates anomalous behaviours such as double-spending attempts, suspicious validation delays, or transactions involving merchants with low credibility. To facilitate the exploitation of the Offsim dataset, Table 7.2 details the role and function of each column. Conversely, Table 7.5 presents the fraud proportions observed across the data.

Table 7.2: Dataset description

Variable Name	Description
`step`	Simulation time index of the transaction; each step corresponds to a discrete time unit (e.g., one hour).
`action`	Category of the financial operation, such as `TRANSFER`, `CASH-IN`, or `CASH-OUT`.
`amount`	Monetary value exchanged in the transaction, expressed in arbitrary local units.
`client_from`	Unique identifier of the originator of the transaction (which can be either a legitimate user or a fraudster).
`old_balance_orig`	Account balance of the sender prior to the transaction.
`new_balance_orig`	Account balance of the sender after the transaction is executed.
`client_to`	Unique identifier of the receiving party, often corresponding to a merchant.
`old_balance_dest`	Recipient's account balance before the incoming transfer.
`new_balance_dest`	Recipient's account balance after the incoming transfer.
`is_fraud`	Binary flag indicating whether the transaction is fraudulent in a general sense (e.g., initiated by a fraudster).
`is_flagged`	Shows whether the fraudulent actor has been marked as suspicious.
`is_offline`	Specifies if the transaction occurred in offline mode, without immediate network synchronization.
`device_timestamp`	Local timestamp recorded by the client's device when the transaction took place.
`sync_delay_minutes`	Time lag, in minutes, between the device's local timestamp and the moment of server synchronization.
`double_spend_flag`	Boolean variable marking suspected double-spending behavior (e.g., spending more than the available balance while offline).
`merchant_reputation`	Categorical attribute describing the merchant's trust level: `low`, `medium`, or `high`.
`is_fraud_off`	Final target label for offline fraud detection; set to true when the transaction is confirmed as an offline fraud case.

Table 7.3: Observed proportions for principal fraud-related binary attributes

Variable / Category	Count	Proportion (%)
`is_fraud`		
False	6 326 134	91.91
True	555 866	8.09
`is_flagged`		
False	6 882 000	100.00
True	0	0.00
`is_fraud_off`		
False	6 768 048	98.34
True	113 952	1.66
`double_spend_flag`		
False	6 761 041	98.35
True	120 959	1.65

7.5.3.1 Data Cleaning

At this stage we assess the consistency of the dataset. A global inspection shows that all variables are correctly typed: numerical fields such as `amount`, `old_balance_orig`, and `sync_delay_minutes` are stored as floating-point or integer values; categorical or identifier fields (e.g. `client_from`, `action`) are treated as strings; and boolean indicators (`is_fraud`, `is_offline`) are properly encoded. Furthermore, no missing values were found, which simplifies the modelling pipeline by obviating imputation or proxy variable strategies and reduces coverage bias. Moreover, a clean data set guarantees robust, interpretable results. Noticeably for label based analyses involving `is_fraud_off` or decision-layer features such as `double_spend_flag`.

7.5.3.2 Distribution Analysis

This crucial step consists of analysing and visualising the empirical distributions of numeric and categorical variables. Fig. 7.5 serves as the analytical framework for these various numeric metrics. The detailed dataset analysis reveals that enriching the data with behavioural metrics such as offline transaction history or periodicity (inter-arrival times) could enhance the model's sensitivity to offline fraud. The detailed statistics for numeric variables are available in Table 7.4. Fig. 7.6 presents an overview of the categorical-variable analysis for which the variables `is_fraud`, `is_flagged`, `is_fraud_off`, `double_spend_flag`, `merchant_reputation` and `action` are displayed. The categorical-variable analysis exposes a dual structural imbalance: a dominant majority class in both `is_fraud` and `is_fraud_off`, and under representation of certain modalities in `merchant_reputation` and `action`. Table 7.5 displays the category proportions for key binary/categorical fraud indicators.

7.5.3.3 Rules-Based Engine Implementation

7.5.3.3.1 Decision Path

Given the particular risks posed by offline payments, it is necessary to implement multiple security layers. These can be deterministic, cryptographic, or AI-based. In this study a

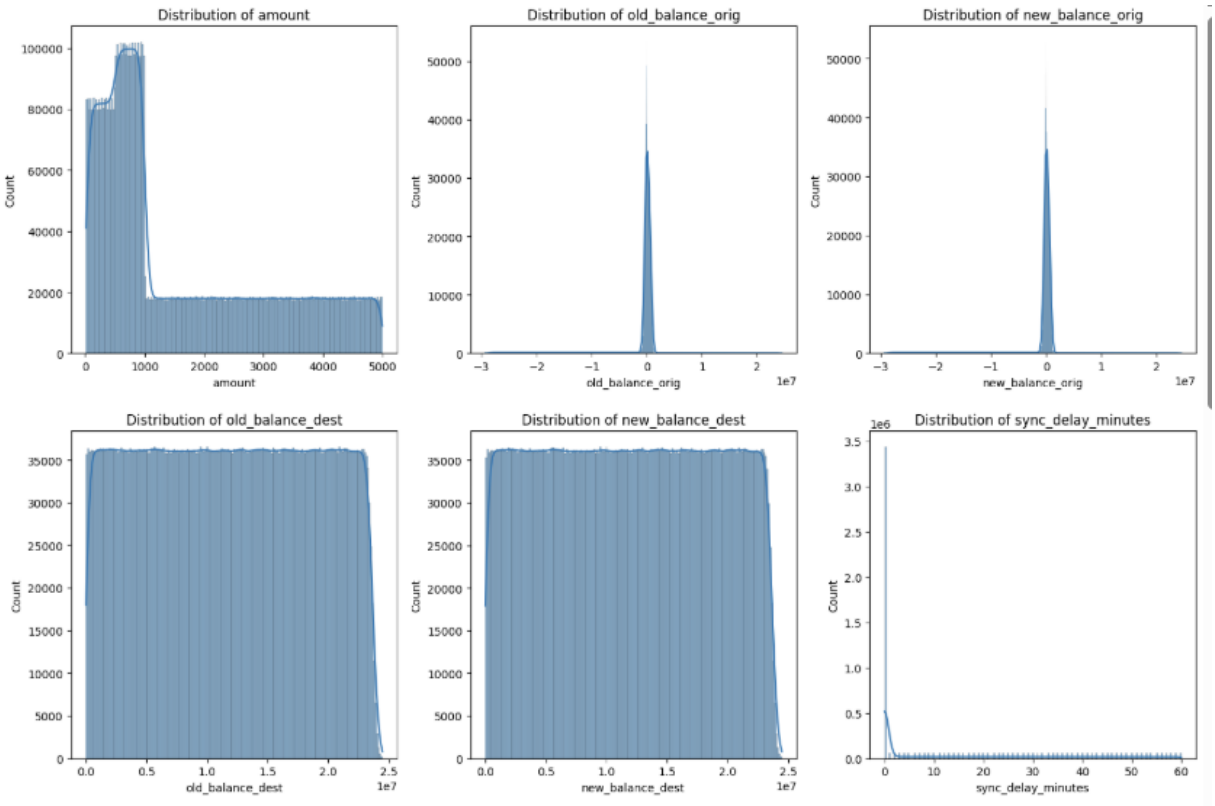


Figure 7.5: Univariate empirical distributions of the most relevant numerical attributes in the raw transaction log. Each panel displays an un–transformed histogram with `bins`=80 and a kernel–density *rug* for visual reference.

rule engine was built as the first line of defence. By embedding it directly on the device it locally verifies device and transaction integrity according to a decision tree. This lightweight, interpretable engine operates without an internet connection. The functional approach is initially grounded in a strict evaluation of conditions: payment amounts, device integrity, transaction frequency, balance consistency, cryptographic validity, etc. It also relies on threshold assessment for each executed transaction. After this dual evaluation, a categorical risk score (low, medium, high) is assigned and transmitted locally to the backend. Consequently, transactions flagged as High or potentially Medium are automatically blocked to eliminate the threat immediately. Conversely, transactions identified as Medium may be held pending further review by the embedded machine learning model. For auditability, process transparency is ensured through a clear decision logic. The algorithm 19 highlights the decision tree applied for each transaction and Table 19 presents the description of variables used in this algorithm.

All things considered, this rule engine offers several relevant advantages. First, it enhances resilience in disconnected environments.Second, it reduces device load through early filtering; indeed, the embedded AI solution is not invoked when a transaction is blocked. Third, it provides adaptability by allowing dynamic updates of rules to counter emerging fraud patterns. This deterministic approach combines local performance with clear decision-making. It can simultaneously block severe threats immediately and supplement subsequent advanced analyses.

Table 7.4: Summary statistics of numerical columns before transformation

Column	Count	Mean	Std	Min	25%	Max
step	6,882,000	186.5	107.4	1	9.38	372
amount	6,882,000	1,628	1,465	1	504	5,000
old_balance_orig	6,882,000	-7.08×10^5	5.17×10^6	-2.93×10^7	-2.19×10^5	2.45×10^7
new_balance_orig	6,882,000	-7.09×10^5	5.17×10^6	-2.93×10^7	-2.20×10^5	2.45×10^7
old_balance_dest	6,882,000	1.184×10^7	6.84×10^6	111	5.92×10^5	2.45×10^7
new_balance_dest	6,882,000	1.184×10^7	6.84×10^6	245	5.97×10^5	2.45×10^7
device_timestamp	6,882,000	2.51×10^4	3.22×10^4	1	160	100,000
sync_delay_min	6,882,000	15.3	19.6	0	0	60

Table 7.5: Category proportions for key binary/categorical fraud indicators

Variable / Category	Count	Proportion (%)
`is_fraud`		
False	6 326 134	91.91
True	555 866	8.09
`is_flagged`		
False	6 882 000	100.00
True	0	0.00
`is_fraud_off`		
False	6 768 048	98.34
True	113 952	1.66
`double_spend_flag`		
False	6 761 041	98.35
True	120 959	1.65

7.5.3.3.2 Initialisation and State Buffers

The rule engine's initialisation phase runs several processes. The first concerns the establishment of hard thresholds: in this experiment, we set a clock-drift bound to detect device tampering (`60 minutes`) and a sliding window of 15 micro-top-ups with an amount below 50€. Moreover, this stage also involves initializing two ring buffers. One manages the transaction time window, while the other handles the offline micro-top-up window. Coordinating these two buffers enables tracking of the last 24 hours of peer-to-peer (P2P) and top-up activity for every sender.

7.5.3.3.3 Offline Fraud Flagging via Rule-Based Preprocessing

After initialisation, each transaction must be marked once it passes through the rule engine. To this end, a new column is added to the dataset a binary "alert" flag. This flag indicates whether the transaction is deemed suspicious or legitimate according to the decision-tree logic. The updated dataset, including the alert flag, serves as input for the machine-learning phase.

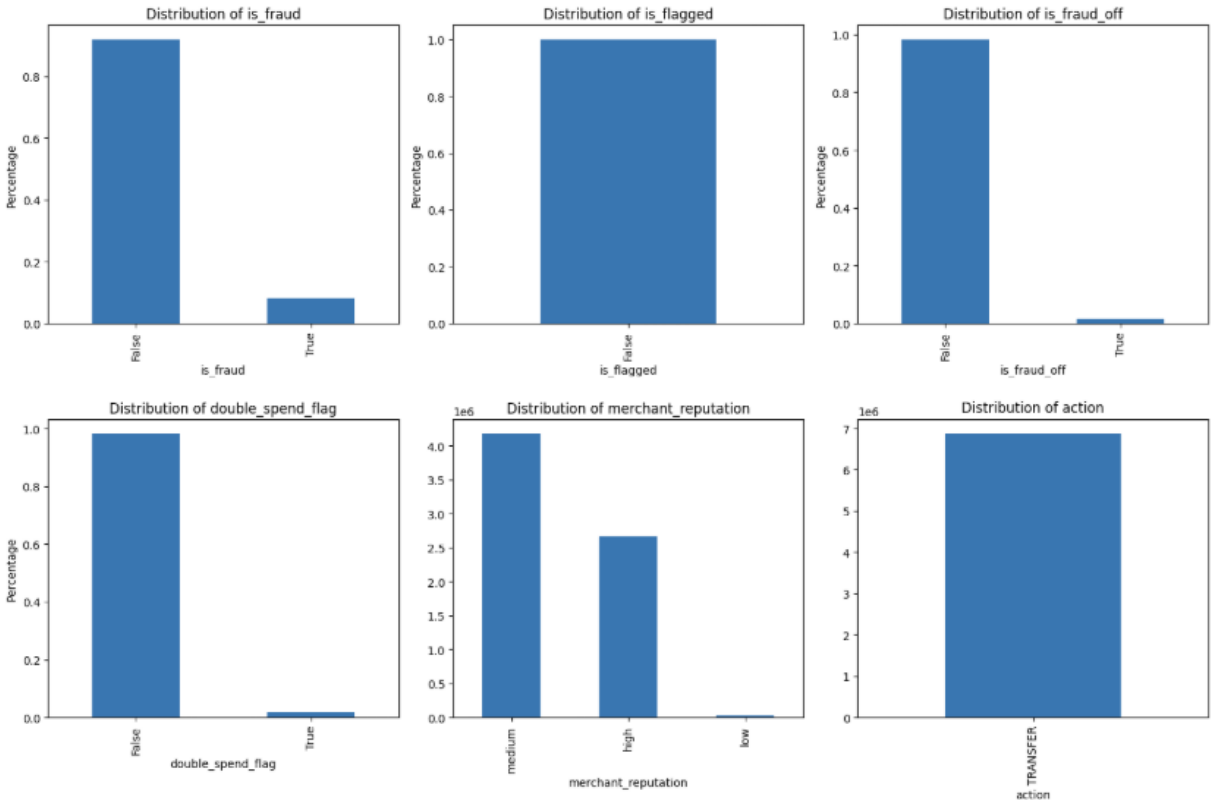


Figure 7.6: Empirical distributions of the principal *categorical / binary* variables. Each bar reports the relative frequency of the corresponding modality over the entire raw dataset.

7.5.3.4 Exploratory Analysis

7.5.3.4.1 Class Imbalance Assessment

The dataset obtained after rule-based preprocessing exhibits a pronounced class imbalance: only about 1.658065% of transactions are deemed fraudulent. At this stage, given such an imbalance, no reference model can realistically deliver meaningful and efficient performance. Fig. 7.7 highlights the minority class in the dataset.

7.5.3.4.2 Distribution analysis based numerical feature based post-rule-engine sample

This analysis allows us to observe the distribution of the various numeric variables after filtering. At first glance, online traffic was removed so that only the offline transaction flow remains. Likewise, only offline transactions reflecting a "High" or "Medium" risk were kept. Fig. 7.8 shows the empirical distributions for all these purely numerical columns. On this figure, the blue curves correspond to the surviving *legitimate* records (`is_fraud_off = False`, roughly 3.29 million rows), and the red curves to the *remaining offline fraud* (`is_fraud_off = True`, about 216,000 rows, i.e.6.6×10^{-2}%). Furthermore, it should be noted that the two classes differ by two orders of magnitude. Hence the density of legitimate transactions appears visually more pronounced. The shaded red area nevertheless reveals where the residual fraud still concentrates.

According to Fig. 7.8, the `step` variable (Time index) exhibits a saw-tooth distribution. This periodic oscillation reflects the simulator's regular reset of agent activities every 24 time units. It no longer contributes significantly to classification once obvious

Table 7.6: Description of variables used in algorithm

Variable	Description
device_jailbroken	Boolean, true if the device is rooted/jailbroken.
clock_drift_min	Time drift of device clock in minutes.
sender_authenticated	Boolean, indicates whether sender authentication succeeded.
is_offline	Boolean, true if the transaction is offline.
double_spend_flag	Boolean, true if double-spend indicator is triggered.
ledger_sync_age_h	Age of last ledger synchronization, in hours.
tx_type	Transaction type: P2P (peer-to-peer) or POS (point of sale).
amount	Transaction amount.
sender_mean_amount	Average amount historically sent by the sender.
offline_p2p_24h_count	Number of offline P2P transactions in the last 24h.
merchant_reputation	Reputation level of the merchant (LOW, MEDIUM, HIGH).
offline_topups_last_window_count	Number of offline top-ups within a sliding time window.
offline_topup_amount_range_ok	Boolean, true if top-up amounts fall within [3–50 EUR].
recipient_suspicious	Boolean, true if the recipient is flagged as suspicious.

anomalies have been removed. Moreover, analysis of the amount variable reveals a bimodal distribution. This observation empirically validates the hypothesis that fraudsters favour larger amounts in offline settings. The variables old_balance_orig, analogously new_balance_orig shows a tight distribution around zero. Fraudulent transactions cluster near zero, hinting at a double-spending behaviour. The variables (old_balance_dest and new_balance_dest) exhibit a uniform distribution up to about 22 million units. Consequently, this observation supports the rule-engine approach: it prioritises merchant confidence scores and behavioural indicators over raw balances to flag suspicious behaviour. The device_timestamp variable is densely populated between 1 and 3.6 million. This dense band corresponds to the number of simulated steps. Observations show that fraudulent trends mirror exactly the distribution of legitimate transactions. Finally, the sync_delay_minutes variable displays a pronounced contrast between classes. This feature emerges as one of the most discriminative numeric predictors after rule-engine application.

7.5.3.4.3 Correlation Analysis of Numerical Features

The post filtering correlation matrix offers a compact view of the fundamental numeric variables. It highlights latent redundancies and confirms that the rule engine has effectively eliminated the most trivial fraud signals. This step is crucial for observing interactions among core numerical features. We can thus see how these variables covary with one another and with the target is_fraud_off. Only cleaned and filtered transactions are subsequently introduced into the machine-learning pipeline. Fig. 7.9 presents Pearson coefficients computed on offline records not flagged by the rule layer ($n \approx 3,3$ M). This subset represents the "gray zone" that the classifier must still separate.

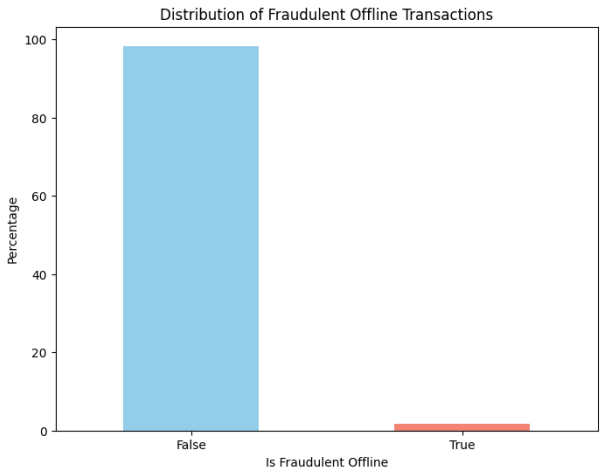


Figure 7.7: Class Imbalance visualization

7.5.3.5 Data Pipeline

The deterministic fraud-detection rules engine is important but not sufficient. While it can detect the most obvious frauds, it has intrinsic limitations. This limitations prevent it from keeping pace with evolving fraudulent strategies. For illustration, fraudsters may adjust their transaction amounts to stay just below defined detection thresholds. Thus, they can blend into the *LOW* category and evading predefined filters. This adversarial adaptability necessitates incorporating a lightweight machine-learning model. The goal is to identify complex, non-linear variable combinations that deterministic rules cannot capture.Subsequently, to fully exploit this complementarity, only offline transactions classified as *LOW* by the rule engine are retained. In other words, only those offline transactions where `alert == 0` after rule filtering. This strategic selection serves two purposes. The first objective is to exclude obvious fraud cases before machine learning training. The second objective is to build a relevant corpus for the machine-learning model. To achieve this, the engine is fed solely with offline transactions that escaped rule detection. Thus, a pivot variable `"alert"` is created in the dataset to facilitate filtering: it indicates a suspect transaction after rule filtering with `alert = 1`. Otherwise, it will indicate `alert = 0`. Initially, the `"alert"` column selects relevant transactions for model training. Subsequently, it serves as an explainable trace for transaction audit requirements. Table 7.7 summarizes the impact of the `alert` column on downstream ML.

Table 7.7: Impact of the `alert` column on downstream ML

Without `alert`	**With** `alert`
ML also re-learns trivial, rule-based patterns	ML focuses on borderline / complex cases only
Less control, limited explainability	Full traceability (rule vs. ML decision)
Risk of over-fitting on obvious signals	Better generalisation to subtle fraud patterns

7.5.3.6 Feature Engineering

In order to enhance classification performance and model generalisation, feature engineering is applied to every transaction. This routine of transforming raw data generates

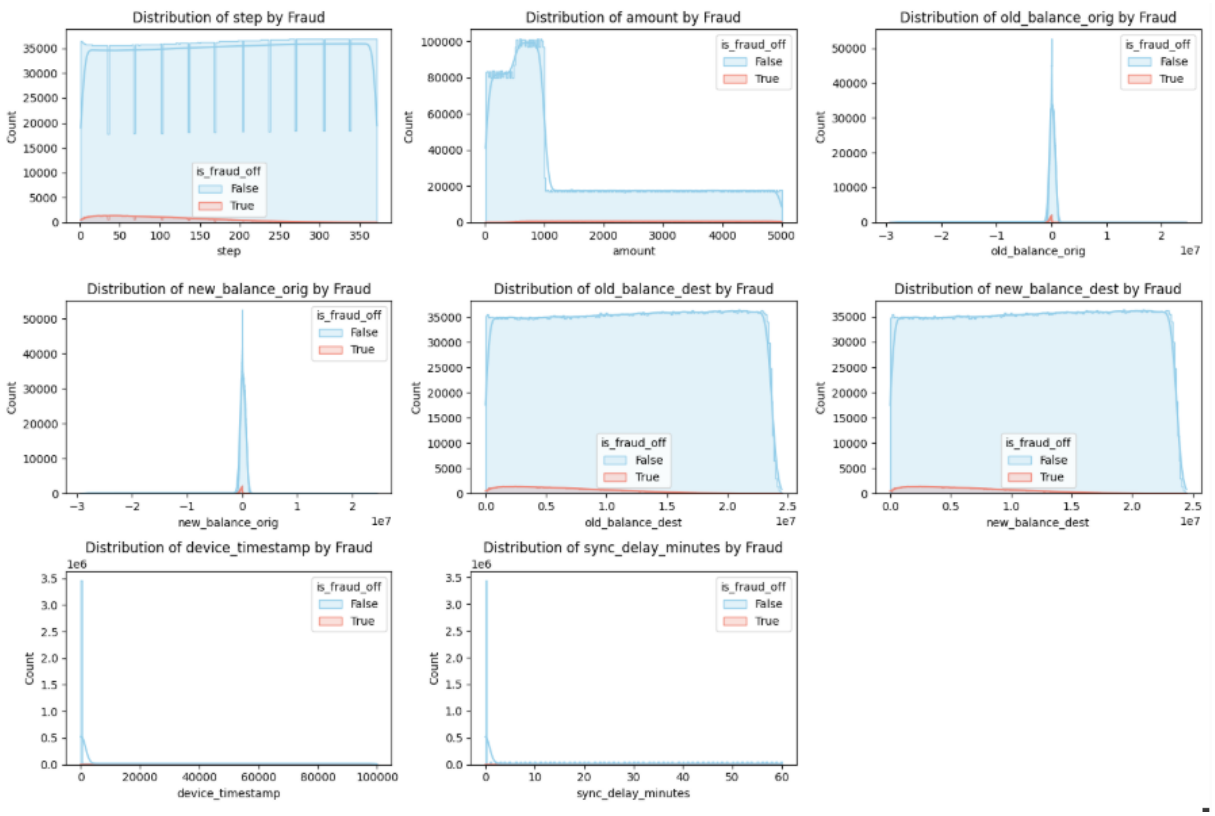


Figure 7.8: Distribution analysis after the rule engine process

both behavioural indicators and temporal markers. The key metrics `ratio_amount_orig` and `ratio_amount_dest` are computed to assess transactional proportionality relative to wallet balances. The feature `ratio_amount_orig` measures the fraction of the sender's balance consumed in a single transaction; by leveraging this metric, the model can spot anomalous behaviours such as attempts to drain an account instantaneously. In contrast, `ratio_amount_dest` evaluates the amount received against the recipient's initial balance, thereby facilitating detection of mule or proxy usage often characterised by accounts that are frequently credited and then quickly debited. Moreover, `net_delta_orig` is introduced to quantify the net financial impact of each transaction on the sender. It essentially measures the precise change in the origin balance. Unlike gross balances, this net differential reflects the true financial variation irrespective of account size, making it a more universal and informative indicator.

Temporal variables are also derived to capture behavioural anomalies over time. The features `hour_of_day` and `day_of_week` are extracted from the simulation index, assigning each transaction to human interpretable time slots. With these markers, it becomes possible to detect suspicious activity during off peak hours or according to atypical weekday/weekend patterns behaviours frequently associated with organised or opportunistic fraud.The feature `ledger_bucket` builds upon `sync_delay_minutes` to categorise the synchronization delay between the local device and the ledger into risk zones. This bucketization materialises the duration for which a transaction remained offline, an essential cue for identifying double spend fraud scenarios. Indeed, the longer the synchronization lag, the higher the likelihood of a double-spending attempt.

Finally, the boolean flag `p2p_flag` distinguishes peer-to-peer transfers from other transaction types. By incorporating this feature, the model can treat P2P operations differently from merchant payments. This specific processing logic recognise the inher-

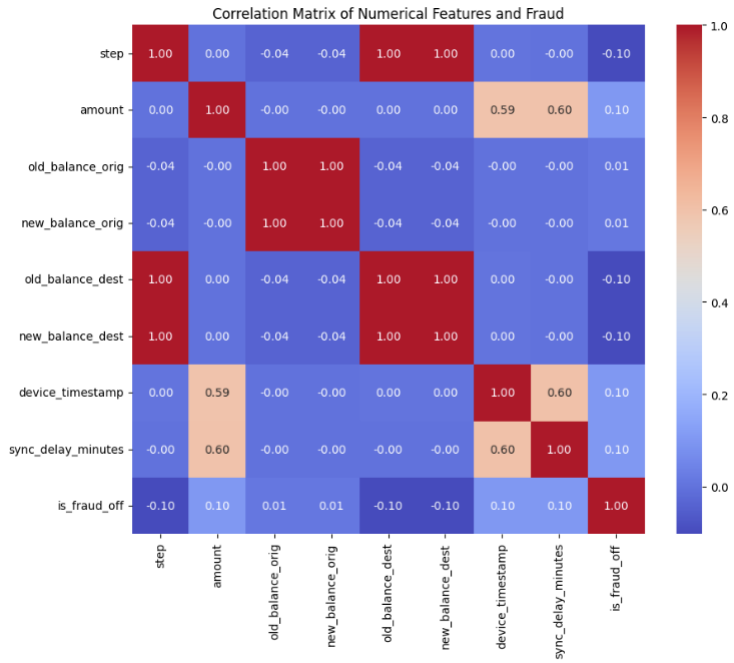


Figure 7.9: Pearson correlation matrix computed on the *post-rule engine* dataset (only offline transactions with `alert`=0). The diagonal elements are 1.0 by definition. Warm colors indicate positive correlations, while cool colors indicate negative correlations.

ent trust asymmetries between individual-to-individual transactions and those involving merchants. All in all, this suite of engineered features offers a compact yet expressive representation of each transaction. Noticeably by integrating both static and dynamic fraud indices in an economically efficient manner. Table showcases the summary of engineered features.

7.5.4 Benchmarking Tree-Based Ensemble Models

7.5.4.1 Candidate Models for On-Device Fraud Scoring

This step involves evaluating machine learning models that can be hybridised with a Rules-Based Engine. To this end, I surveyed the literature and identified AI fraud-detection models that satisfy three complementary requirements. The first requirement concerns the model's ability to achieve high predictive performance on data streams that are extremely under-represented. The second requirement is its capacity to orchestrate within a device with limited resources. Finally, the third requirement concerns the model's willingness to provide explainability. Consequently, three families of algorithms can meet this triple criterion:Generalised Linear Models, Single decision trees, Gradient–Boosted Trees, Logistic Regression, Bagging Methods, and others methods like Support vector machines and Tiny Neural Network. [134, 137–141]. After careful consideration, the selected test models are Random Forest, LightGBM, and XGBoost. The other families do not meet the necessary requirements. In particular, due to their lack of scalability, interpretability or difficulty in orchestrating quantisation for Tiny Neural Networks.

7.5.4.2 Implementation Plan

The various models will be tested using a supervised-learning pipeline that consists of three stages. First, we will perform numerical encoding of categorical features. This phase is

Table 7.8: Summary of Engineered Features

Feature	Type	Derived From	Purpose and Description
`ratio_amount_orig`	Float	`amount, old_balance_orig`	Quantifies the transaction amount as a proportion of the sender's available balance; essential for identifying excessive spending patterns or potential overdrafts.
`ratio_amount_dest`	Float	`amount, old_balance_dest`	Assesses the transaction amount in relation to the recipient's existing balance; critical for detecting irregular or suspicious transfer patterns.
`net_delta_orig`	Float	`old_balance_orig, new_balance_orig`	Measures the exact change in the sender's balance post-transaction; provides a scale-invariant metric for financial impact assessment.
`hour_of_day`	Categorical (0–23)	`step`	Represents the hour of the day when the transaction occurred; enables detection of unusual off-peak activities or automated fraud patterns.
`day_of_week`	Categorical (0–6)	`step`	Captures the day of the week for the transaction; helps identify behavioral anomalies between weekdays and weekends.
`ledger_bucket`	Categorical	`sync_delay_minutes`	Categorizes synchronization delays into risk-based intervals; facilitates modeling of reconciliation risk levels.
`p2p_flag`	Boolean	`transaction_type`	Indicates whether the transaction is peer-to-peer (vs. merchant payment); highlights differences in trust dynamics between transaction types.

essential to ensure that categorical variables such as transaction type are correctly split and interpreted during model training. Once this preprocessing step is completed, the next stage addresses the extreme class imbalance in the dataset. The rarity of fraudulent transactions (less than 1%) poses a challenge for many classifiers. Several techniques exist to mitigate this issue. Finally, the third operation concerns constructing the training dataset. At this stage, all records of the selected target features (10 numerical variables and two categorical ones) after feature-engineering are projected into a 12-feature vector. Table 7.9 describes these 12 features. Likewise, the binary target labels `is_fraud_off` (explicitly cast to integer format) are assembled into a label vector. In the end, the collected feature matrices and target vectors are vertically concatenated into the final matrix $X \in \mathbb{R}^{n \times 12}$ and label vector $y \in \{0, 1\}^n$.

Table 7.9: Input Features for Machine Learning Database

Type	Feature	Description
Categorical	`action`	Encoded transaction type (e.g., top-up).
	`merchant_reputation`	Discretized merchant risk score.
Numerical	`amount`	Transaction amount (€).
	`sync_delay_minutes`	Time since last sync (device isolation proxy).
	`double_spend_flag`	Binary double-spend indicator.
	`ratio_amount_orig`	Amount / origin account's prior balance.
	`ratio_amount_dest`	Amount / destination account's prior balance.
	`net_delta_orig`	Origin account's net balance change.
	`hour_of_day`	Transaction hour (0–23).
	`day_of_week`	Transaction day (0=Mon, 6=Sun).
	`p2p_flag`	P2P indicator (1 if P2P, else 0).
	`ledger_bucket`	Discrete historical transaction volume.

7.5.4.2.1 LightGBM

LightGBM serves as a benchmark in the context of native categorical variable handling, training time, and memory footprint. To achieve this, LightGBM relies on *Gradient-Based One-Side Sampling* (GOSS) and *Exclusive Feature Bundling* (EFB) [137]. To handle the extreme class-imbalance problem, it proved necessary to deploy Synthetic Minority Over-Sampling Technique for Nominal and Continuous features (SMOTENC)[142]. This is a variant of the standard Synthetic Minority Over-Sampling Technique (SMOTE). SMOTENC offers the possibility to treat mixed data. In other words, SMOTENC can handle both categorical and numerical features of the dataset. Moreover, it helps avoid over-fitting that may arise from aggressive oversampling. Furthermore, by targeting a 1:2 ratio (0.5 sampling strategy), SMOTENC avoids the "all–positive" artefacts inherent to extreme weighting while ensuring the learner has enough minority mass to generalise. Consequently, SMOTENC generates synthetic minority samples by applying classical linear interpolation for numeric features (as in SMOTE) and a majority-vote technique. To ensure that the model is evaluated on imbalanced test data, the dataset is split into 80% training and 20% testing subsets. These resulting subsets are directly fed to the LightGBM trainer. The same stratified train–validation split is used for all other models tested in this experiment.

Moroever, Optimising parameters is crucial to maximise model performance. To achieve this, we used 9 parameters, namely `scikit_learn`, `objective`, `metric`, `learning_rate`, `num_leaves`, `num_leaves`, `max_depth`, `subsample`,and `colsample_bytree`. Table 7.10 provides their descriptions and the values used. Then early stopping strategy is applied in conjunction with the optimisation parameters. Its aim is to determine, adaptively and robustly, the number of training iterations. The performance metric `AUC` is monitored as a function of the number of rounds and the number of boosting iterations. In our experiments these values are set to 50 rounds and 800 boosting iterations respectively. Consequently, if after 50 rounds the `AUC` metric no longer improves, training stops before reaching the maximum of 800 iterations. Finnally, the Isotonic Calibration method is used to post-process the raw scores so that they become more probabilistic and interpretable.

Table 7.10: Selected LightGBM Hyperparameters and Their Descriptions

Parameter	Value	Description
`objective`	`binary`	Specifies the learning task type.
`metric`	`auc`	Evaluation metric during training.
`learning_rate`	0.05	Step size shrinkage for gradient descent.
`num_leaves`	63	Maximum number of leaves per tree.
`max_depth`	-1	Maximum tree depth (-1: unlimited, constrained by `num_leaves`).
`subsample`	0.8	Fraction of samples used per tree.
`colsample_bytree`	0.8	Fraction of features used per tree.
`min_data_in_leaf`	100	Minimum samples required in a leaf.
`verbose`	-1	Controls training log verbosity.

The performance of the model (and all models in this experiment) is evaluated using six relevant metrics:

- **Precision**. This metric represents the proportion of transactions flagged as fraudulent that are actually fraudulent.
- **Recall**. It highlights the model's sensitivity to correctly detect all fraud cases.

- **F1-Score**. This metric captures the maximum measure that balances precision and recall, especially when no threshold yields an efficient precision.
- **AUC-ROC**. The Area Under the Receiver Operating Characteristic Curve depicts the model's ability to rank fraudulent versus non-fraudulent transactions across all classifications.
- **PR-AUC**. This metric represents the area under the Precision–Recall curve. It is useful for plotting precision against recall in an imbalanced context. To this end, it focusing solely on positive predictions. Thus it determines the proportion of truly fraudulent transactions among all transactions detected as fraudulent (precision). Additionally, it determines the proportion of correctly detected frauds among all actual frauds.
- **Confusion Matrix**. This metric provides a summary of correct and incorrect predictions made by the model. By counting and categorising all predictions broken down by class (fraud vs. non-fraud), it offers an interpretability framework for the model.
- **Threshold Optimisation**. By varying the threshold, one can choose the value that maximises the F1-score for classifying transactions as fraud.

All these metrics provide a comprehensive evaluation framework for all other models.

7.5.4.2.2 Random Forest

The Random Forest algorithm aggregates predictions from multiple individual trees. In this way, the final prediction is either the class that maximises the average of all tree predictions or the one that receives the most votes [139].For the optimisation settings, seven parameters are defined. Noticeably,`n_estimators=600`, `max_depth=None`, `None`, `min_samples_leaf=2`, `max_features="sqrt"`, `n_jobs=-1`, `random_state=42`, and `class_weight=None`, and `n_estimators=600`.

Moroever the isotonic calibration and threshold selection step first transforms the raw Random Forest scores into reliable calibrated probabilities. Next, it finds the optimal decision threshold for making calls while avoiding any data leakage. In the experiment performed, the business goal is to ensure that every alert triggered by the model has at least a 55% chance of being a real fraud.

Table 7.11: Selected Random Forest Hyperparameters

Parameter	Value	Description
`n_estimators`	600	Number of trees in the forest.
`max_depth`	`None`	Maximum tree depth (unlimited).
`min_samples_leaf`	2	Minimum samples per leaf node.
`max_features`	`"sqrt"`	Features considered per split (total_features).
`n_jobs`	-1	CPU cores used (-1: all available).
`random_state`	42	Seed for reproducibility.
`class_weight`	`None`	Class rebalancing (disabled).

7.5.4.2.3 XGBoost

Like LightGBM, XGBoost is an implementation of Gradient Boosting. However, the two differ in how they grow trees. While LightGBM uses a "best-first" strategy, XGBoost employs balanced growth: all nodes at a given level are expanded simultaneously, whereas LightGBM expands only the leaf with the greatest gain. It is therefore interesting to see what fraud detection performance XGBoost will yield. In practice, XGBoost seeks to optimise an objective function using a gradient-boosting approach that combines several components. Notably, the model prediction, the loss function, and regularisation to prevent over-fitting [143].

The optimisation configuration concerns nine hyper-parameters. Noticeably, `objective = "binary:logistic"`, teval_metric, `max_depth: 8`, `subsample`, `colsample_bytree`, `lambda`, `alpha`, and `tree_method` . In line with LightGBM, early stopping is also applied to XGBoost to achieve the same objectives. Consequently, the number of rounds and iterations are set to 50 and 800 respectively. Likewise, the same calibration procedure used for LightGBM is applied to XGBoost.

Table 7.12: Selected XGBoost Hyperparameters

Parameter	Value	Description
`objective`	`"binary:logistic"`	Binary classification with logistic loss function.
`eval_metric`	–	Class separation assessment metric.
`eta`	0.05	Low learning rate for stable updates.
`max_depth`	8	Tree depth for complex interactions.
`subsample`	0.8	80% data subsampling per tree.
`colsample_bytree`	0.8	80% feature subsampling per tree.
`lambda`	1.0	L2 regularization strength.
`alpha`	0.0	L1 regularization (disabled).
`tree_method`	`"hist"`	Histogram-based optimization.

7.5.5 Results and Comparative Analysis

This section presents the experimental results obtained from our proposed offline CBDC fraud detection framework. First, it reports the Rule Engine results. Next, it evaluates each machine-learning model individually. Then it compares them to highlight each model's strengths and limits. Finally, it analyses the complementary performance of a LightGBM–XGBoost ensemble relative to each model's performance.

7.5.5.0.1 Rule Engine Performance Evaluation

The Rule Engine performance is analysed against three metrics. Notably, Precision, Recall and F1-score. The evaluation covered almost 6.5 million transactions. It reveals that the detection engine based solely on deterministic rules (without machine learning) achieves exceptional efficiency. Indeed, with a Precision of 85.2%, only 14.8% of generated alerts are false positives. This performance metrics ensuring high relevance of alerts. In contrast, with a Recall of 99.8%, virtually all frauds are identified. This has the effect of minimizing the risk of non-detection. Moreover, an F1-score of 91.9% confirms an optimal balance between detecting all fraud cases (Recall) and keeping false positives low (Precision). Table 7.13 summarises these results.

Table 7.13: Performance of the Rule Engine

#	Metric	Value	Interpretation
1	Precision	0.852	High precision indicates a low false-positive rate.
2	Recall	0.998	Captures 99.8 % of true offline fraud cases, showing near-perfect recall and very low false-negative rate.
3	F_1 score	0.919	Harmonic mean of precision and recall: $F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$. Score of 0.919 confirms excellent balance between fraud detection and limiting false alerts.

7.5.5.0.2 Class Distribution Results after SMOTENC

The obtained results show that the class distribution of fraudulent versus legitimate transactions was significantly improved. Consequently, the strong imbalance from the original dataset is reduced. Thus, the final training set after SMOTENC contains approximately 4.96 million transactions with 12 relevant features. The fraud ratio after re-balancing is 33.33%. In other words, synthetic oversampling led to a positive class ratio of one fraudulent transaction for every three transactions (0.33). Overall, this result offers a balanced environment for the boosting models. These are now capable of capturing fraud patterns effectively without bias toward the majority class. Table 7.14 presents the characteristics of the SMOTENC-balanced dataset.

Table 7.14: Characteristics of the SMOTENC-Balanced Dataset

Property	Value / Interpretation
Total samples	≈4.96M transactions
Features	12 (categorical + continuous)
Fraud-to-legit ratio	1:2 (positive class: ≈33%)
Balancing method	SMOTENC (synthetic oversampling)
Impact	Reduced class imbalance for fairer model training

7.5.5.0.3 Baseline Model Performance

LightGBM, Random Forest and XGBoost were primarily evaluated as baseline models. They are chosen because they are high calibre algorithms for structured financial data. Table 7.15 summarises their predictive performance on the SMOTENC-balanced dataset.

Precision and Recall: LightGBM reports a precision of 0.993, corresponding to only 0.33% false positives. Random Forest, in contrast, achieves a precision of 0.985, i.e. 0.78% false positives, more than twice the absolute number of errors on the legitimate class. Both LightGBM and Random Forest outperform XGBoost alone on this metric. Moreover, all models achieve a score of 1.000. This means that every fraud was detected in the validation set.

AUC & PR-AUC: Random Forest attains an AUC of 1.000, while LightGBM and XGBoost remain at 0.999985 and 0.99999, respectively. These almost perfect scores reflect extreme separability after feature engineering and SMOTENC. Although statistically equivalent, this may indicate slightly stronger regularisation in LightGBM and XGBoost. Conversely, the "perfect" metrics of Random Forest (AUC = 1.0, PR-AUC = 1.0) carry a higher risk of overfitting due to the algorithm's structure.

Table 7.15: Performance Comparison of Baseline Models

Metric	LightGBM	Random Forest	XGBoost
Thresholds and Scores			
Threshold (τ)	0.001	0.000	0.000
AUC Val.	∼0.9999	1.0000	0.99999
PR-AUC	1.000	1.000	1.000
Precision	0.993	0.985	0.985
Recall	1.000	1.000	1.000
Confusion Matrix			
True Negatives (TN)	659,364	656,406	654,307
False Positives (FP)	2,207	5,165	4,993
False Negatives (FN)	0	0	0
True Positives (TP)	330,786	330,786	330,786
Derived Metrics			
Specificity	0.9967	0.9922	0.9890
F1-Score	0.9965	0.9924	0.9894

Note: Specificity = TN / (TN + FP); F1-Score = 2 × (Precision × Recall) / (Precision + Recall).

Calibration and optimal threshold τ: The isotonic calibration of Random Forest yields a threshold $\tau = 0.000$. This indicates that the model predicts probabilities very close to 1 for fraud and near 0 for legitimate transactions. Consequently, any threshold > 0 is superfluous for maximising recall. XGBoost exhibits similar behaviour with an optimal threshold at $\tau = 0.000$. LightGBM, however, retains a slightly positive optimal threshold ($\tau = 0.001$). It shows a more nuanced calibrated probability distribution.

Confusion matrix: The confusion matrix highlights both the differences and trade-offs among the reference models. LightGBM achieves the most balanced result. Notably, while maintaining perfect recall, it substantially reduces false positives compared to Random Forest and XGBoost.

Overall, considering the various metrics, LightGBM and XGBoost best satisfy the deployment objectives. Indeed, Random Forest's risk of overfitting on patterns. Moreover, in a constrained environment characterized by offline operation, limited device computational resources, model size, and inference time become critical factors. However, Random Forest, with its size of 346 MiB (compared to 5.4 MB for LightGBM and 10.3 MB for XGBoost), can be more demanding to embed due to its large number of trees.

7.5.5.0.4 Comprehensive Benchmark of the proposed models

The baseline model performance revealed LightGBM and XGBoost as the models offering the best trade-off for deployment in constrained environments. Table 7.16 highlights the model artifacts and loading information for `XGBoost` and `LightGBM`. Moreover, Table 7.17 presents the inference-benchmark results for LightGBM and XGBoost. The comparative study between LightGBM and XGBoost is based on latency, throughput, and memory consumption metrics. It underscores significant performance differences. Identifying these indicators allows us to consider important practical implications of the models according to their use-case context. Fig. 7.10 synthesises the full benchmarking. It compares model performances across all inference and memory-consumption metrics.

Latency. When considering latency, the XGBoost model consistently shows a clear superiority over LightGBM across all tested batch sizes. The increased speed suggests that XGBoost is particularly suited to scenarios requiring rapid responses—such as real-time fraud detection over transaction streams.

Table 7.16: Model artifacts and loading information for LightGBM and XGBoost.

Model	Model file (size)	Calibration file (size)	Load time (ms)	RSS before (MB)	RSS after (MB)	Δ RSS (MB)
LightGBM	`lgbm_offline_fraud.txt` (5.4 MB)	`lgbm_calibration.pkl` (1.3 KB)	105.7	2121.7	2121.7	0.0
XGBoost	`xgb_offline_fraud.json` (10.3 MB)	`xgb_calibration.pkl` (1.2 KB)	444.7	3824.6	3756.9	-67.7

Table 7.17: Inference benchmark by batch size for LightGBM and XGBoost (dataset: 100,000 samples; calibration: True).

	LightGBM			XGBoost		
Batch	**mean (ms)**	**throughput (tx/s)**	**RSS (MB)**	**mean (ms)**	**throughput (tx/s)**	**RSS (MB)**
1	11.978	83.5	2121.7	1.769	565.4	3756.9
8	10.228	782.1	2121.7	2.622	3051.5	3756.9
32	5.722	5592.8	2121.7	3.575	8951.4	3756.9
128	20.676	6190.7	2121.7	8.571	14934.5	3756.9
512	66.953	7647.1	2121.7	14.696	34839.4	3756.9
2048	363.304	5637.2	2121.7	54.418	37634.9	3756.9

Note. Batch size = number of transactions processed per inference batch; Mean = average latency per batch; Throughput = transactions processed per second; RSS = resident set size, i.e., memory usage of the process.

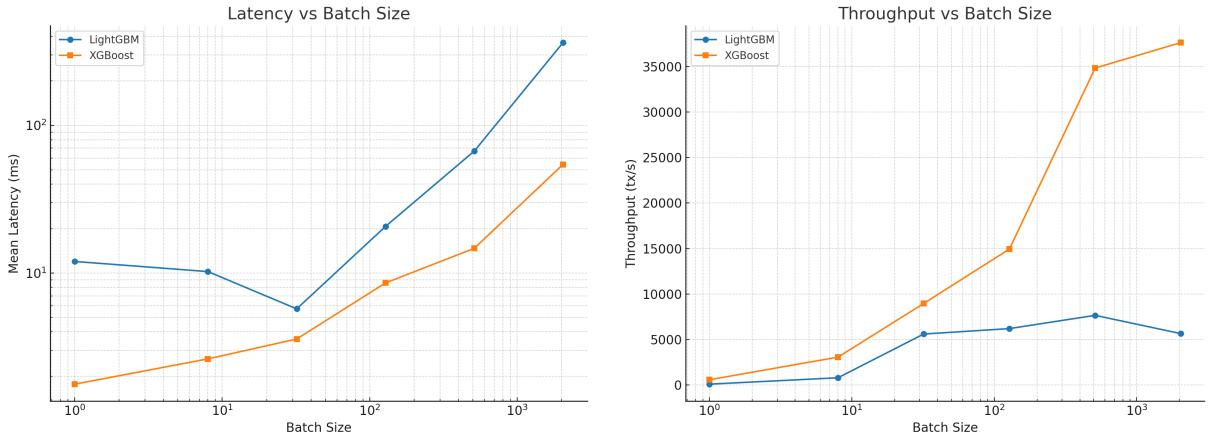
Throughput. In terms of throughput, XGBoost's superiority over LightGBM is equally evident. It already remains markedly higher even for small batches. As batch size grows, the gap widens further. This demonstrates that XGBoost better exploits parallelism and its internal prediction optimisations.

RSS. Regarding memory consumption (RSS), LightGBM maintains a remarkable advantage. In memory it occupies about **2.12 GB**, versus **3.76 GB** for XGBoost. This advantage is crucial in resource-constrained environments. Notably embedded devices or any other low-memory platform.

In summary, XGBoost is better suited to heavily loaded real-time applications. In contrast, LightGBM, though slower than XGBoost, offers lower memory consumption. Overall, LightGBM remains the best choice for the expected fraud-detection solution. Particularly because it strikes the sought compromise between execution speed, accuracy, and memory efficiency.

7.5.5.0.5 Explainability AI results

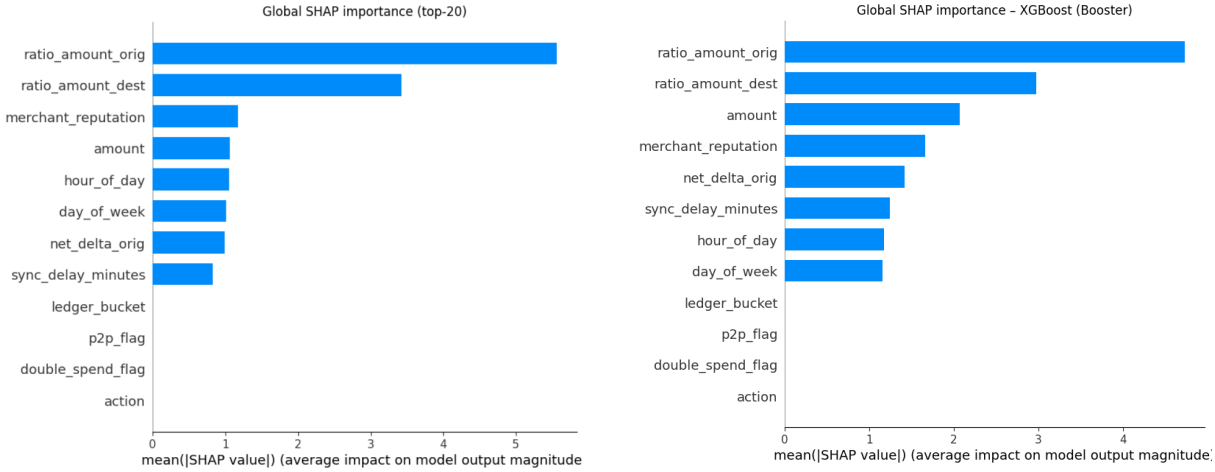
Global explainability (SHAP): The SHAP analyses reveal almost identical patterns across the three models. However, the magnitude of impact varies from one model to another. In the case of LightGBM, Fig. 7.11(a) shows that the variables `ratio_amount_orig`, `ratio_amount_dest` and `amount` consistently dominate the explanation of predictions. This observation underscores their central role in identifying amount anomalies. Notably unusually high debit or credit amounts. Such anomalies are frequently associated with double-spending attempts. Similarly, XGBoost, on its part, overemphasises transaction-ratio variables. Fig. 7.11(b) confirms that it is also highly effective at detecting fraud related to amount or balance. Furthermore, the variable `merchant_reputation` is a pivot feature in decision making. Acting as a trust barrier, it directly influences the prediction (Fig. 7.11b and 7.11a. Its major strength lies in its ability to act as a behavioural proxy, capturing signals that escape simple transactional characteristics (amount, ratios, etc.). In the context of offline transaction auditability, this variable of-



(a) Mean latency across batch sizes

(b) Throughput performance

Figure 7.10: Complete inference performance benchmarking of LightGBM and XGBoost (Tests conducted on 100,000 samples with calibrated probabilities).



(a) Global SHAP importance (LightGBM)

(b) Global SHAP importance (XGBoost)

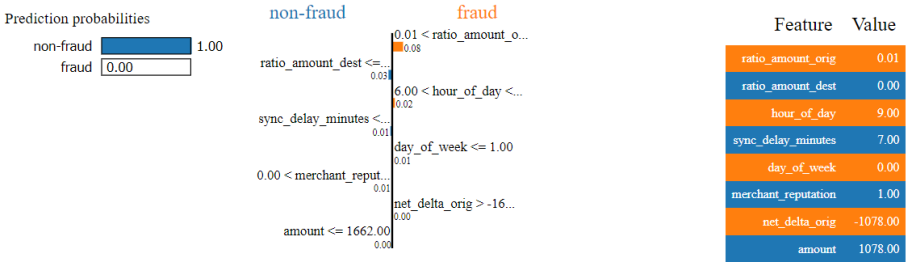
Figure 7.11: Global SHAP feature importance for LightGBM (a) and XGBoost (b).

fers an intuitive and easily justifiable indicator. The lower the merchant's reputation, the more naturally suspect the transaction becomes. Finally, temporal contextual variables, such as `sync_delay_minutes` and the block `hour_of_day/day_of_week`, also help highlight *atypical transaction patterns* in offline mode. notably during atypical hours (nighttime transactions) or on weekends.

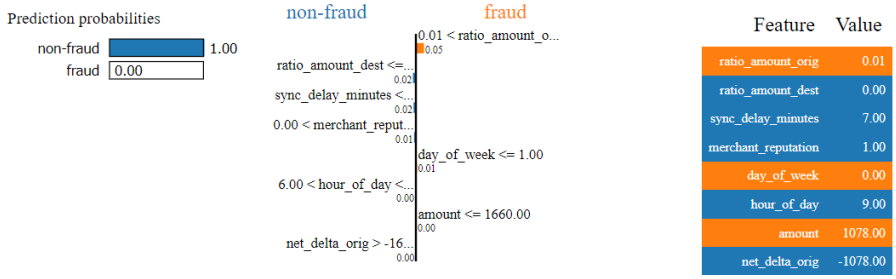
LIME: This tool is useful for auditability because it narrates the local decision in "rules" language. The LIME plot of LightGBM on Fig. 7.12a confirms that the model's local decisions are consistent with its SHAP trends. Indeed, the locally assigned weights highlight the same key variables. In short, LightGBM justifies its decision with intuitive and comprehensible signals such as a proportionate amount, a reliable merchant, and a "normal" time. Moreover, Fig. 7.12b of XGBoost shows the logic mirrors that of LightGBM, marked by the preeminence of the transactional ratio. Table 7.18 presents a synthesis of the comparative LIME study. Nevertheless, it should be noted that LIME is a local approximation and does not guarantee global stability.

Table 7.18: Comparative Analysis of LIME Explanations for Offline CBDC Fraud Detection

Model	Key LIME Features	Local Explanation Pattern	Auditability Relevance
LightGBM	`ratio_amount_orig/dest`, merchant reputation, temporal vars	Relies on 1-2 dominant vars (ratios + reputation)	Intuitive (low ratios + trusted merchant = legit). Auditor-friendly but may seem overly focused.
XGBoost	Ratios, sync delay, day of week, merchant reputation	Strong ratio emphasis + time/reputation signals	Similar auditability to LightGBM; risk of overweighting ratios (adversarial vulnerability).



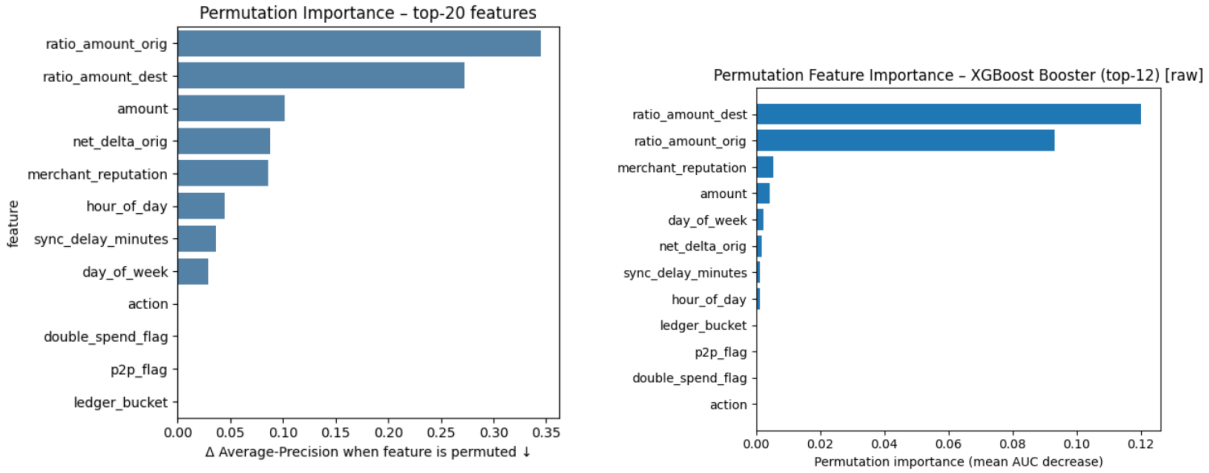
(a) LIME explanation — LightGBM



(b) LIME explanation — XGBoost

Figure 7.12: Comparison of LIME explanations for LightGBM and XGBoost in offline CBDC fraud detection.

Permutation Features Importance: The principle of feature-importance permutation relies on measuring the drop in model performance (e.g., AUC, F1) when a variable's values are randomly shuffled. Thus, if permuting a feature severely degrades performance, it is clear that the feature was crucial for the model's decision making. Fig. 7.13a shows that LightGBM places strong emphasis on transactional variables (`ratio_amount_orig` and `ratio_amount_dest`). This confirms once again that LightGBM focuses more on the consistency between amounts and their source and destination balances for fraud detection. Furthermore, Fig. 7.13b demonstrates that XGBoost's origin/destination transactional ratios also lie at the heart of its decision process. However, XGBoost exhibits an even stronger focus than LightGBM.



(a) Permutation importance — LightGBM (b) Permutation importance — XGBoost

Figure 7.13: Comparison of permutation feature importance for LightGBM, XGBoost, and the Hybrid model in offline CBDC fraud detection.

7.6 Hidden Markov Chain Model-enhanced Fraud Detection for Offline CBDC

7.6.1 Theoretical Foundations

In a non-connected CBDC wallet, transactions do not arrive at random and independently. They organise themselves into structured sequences marked by recurring temporal patterns. Especially, P2P bursts, repeated micro-recharges, clock drift, etc. Machine learning models such as `LightGBM`, `XGBoost` or a hybrid approach therefore enable one to estimate with precision the instantaneous probability $P(\text{fraud} \mid x_t)$ based on the attributes of a transaction x_t. However, these classifiers deliberately ignore inter-transaction dynamics. The challenge in incorporating a Hidden Markov Model (HMM) is to overcome this limitation by reintroducing the sequential dimension. The HMM component serves to smooth over time; it does not examine each transaction in isolation, but rather as part of a sequence. For example, a rule might read: "if you observe three suspicious transactions consecutively, the probability that the entire sequence is fraudulent increases inevitably, even if one transaction carries only a modest score." Thus, the HMM supplies an unsupervised complementary layer to the AI model. Nevertheless, the task remains to realise this HMM integration with negligible computational cost and highly transparent traceability. Fig. 7.15 illustrates the architecture of the HHM embedded within the AI model.

7.6.1.1 Hidden Markov Model Formulation

HMMs were first described in [144]. They are grounded in the interaction between a latent process, invisible to direct observation, and an observable process generated by hidden states. An HMM is globally characterised by the 3 elements: transition matrix, emission probability matrix and the vector of initial probabilities. Given the exponential complexity of this calculation, it is customary to employ optimised methods such as Rabiner's *Forward-Backward* algorithm to evaluate it efficiently

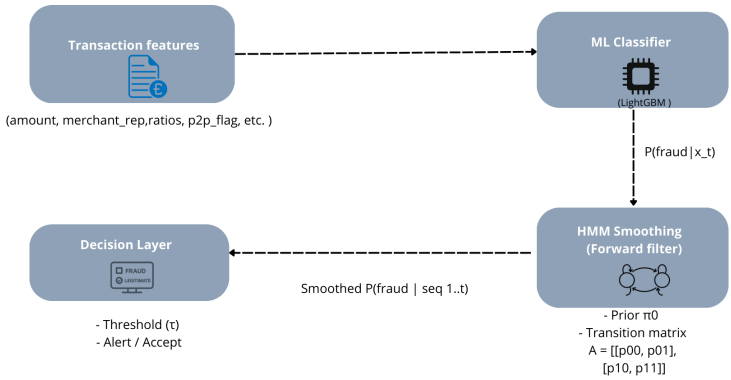


Figure 7.14: LightGBM-HMM Hybrid Architecture

7.6.1.2 GBDT-as-Emission: Integration with Machine Learning for offline fraud detection

A classical Markov chain applied to fraud detection works as long as it is possible to define the states clearly from the data (for instance, associating a `low amount` without fraud with a low score). However, when one wishes to generalise to a real dataset such as the one used in this chapter, there are no clear rules for slicing the state space. It is at this point that the HMM with AI emissions deploys its full insight. Accordingly, in a compact model (2 hidden states: fraud / legit), existing AI models are integrated as sensors producing probabilistic observations. Fig. 7.15 highlights the interactions between the HMM states and the GBT-as-Emission mechanism. Lightweight HMMs come with three main expectations. These concern robustness, offline efficiency, and auditability. Concerning robustness, the HMM must limit isolated false positives by re-weighting a suspicious transaction with an overall coherent history. Conversely, it will amplify detection of repeated fraud sequences, such as burst attacks.

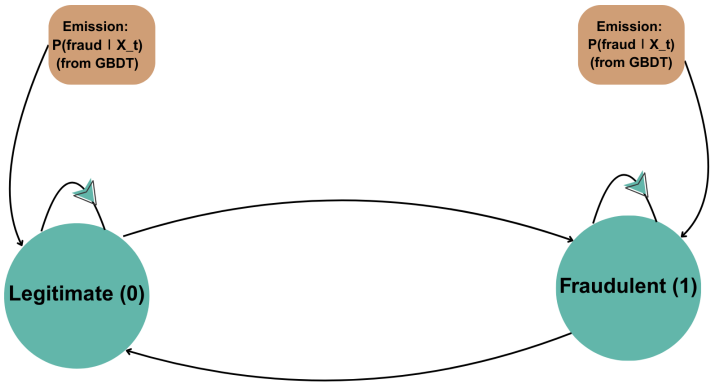


Figure 7.15: HMM with GBDT-as-Emission Hybrid

7.6.2 Implementation and Integration

7.6.2.1 Construction of Ordered Sequences

In this preparatory phase, the objective is to structure the sequential metadata so that it satisfies the HMM requirements. This structuring first involves creating a sequence identifier whose purpose is to group transactions belonging to the same wallet or device. Accordingly, based on the validation set, one must add an annotation column (1 for fraud,

0 for legitimate). Then one checks whether a sequence identifier already exists (e.g., `wallet_id`, `device_id`, etc.). If it does, it is used as the sequence column; otherwise, all transactions are combined into a single sequence. Furthermore, the second pillar of structuring is the temporal index. Its goal is to order events chronologically. Thus, if a timestamp is available (e.g., `timestamp`, `time`, `step`), it is used to sort transactions within each sequence. If not, an artificial index is generated to ensure strict ordering. Finally, every sequence is verified to confirm that it can be sorted chronologically. Consequently, the HMM will process transactions in the correct order and apply its sequential filtering mechanism (*Forward–Backward*).

7.6.2.2 Transition Matrix Estimation in detail

This step aims to learn the transitions between hidden states (`legitimate` = 0, `fraud` = 1). To achieve this, a 2×2 matrix denoted by A was constructed . Each coefficient of the matrix represents the probability of moving from a given state (0 or 1) at a specific instant to another state (0 or 1) at a later instant. For every wallet sequence, the labels are traversed in chronological order. Whenever a transition is observed, the corresponding counter is incremented in the counting matrix. To avoid zero probabilities (for rarely observed transitions), Laplace smoothing was applied. Subsequently, each row of A is normalized so as to form a valid conditional probability distribution.

7.6.2.3 Forward Filtering on ML Outputs

In this phase, we use fraud probabilities produced by the calibrated GBDT model (LightGBM or XGBoost), the transition matrix, and apply the `forward algorithm`. Thus, at each step, we execute the following operations. Firstly, construction of the evidence (emission) vector. Then, perform propagation of the previous belief through the transitions. Upon this step, update with the evidence and performs normalization. Finally the result is the filtered probability that transaction is fraudulent, after smoothing by the HMM.

7.6.3 Experiments and Results

7.6.3.1 LightGBM + HMM

The experiments are carried out exclusively with the LightGBM model, which, according to the benchmark presented in the previous section, offered the best trade-off. The experimental procedure is implemented in exactly the same environment as described earlier. The experiment proceeds in three phases. In the first phase we perform all necessary imports for the test environment (packages, `DataFrame`) and load the artifacts (training data, calibrated LightGBM model, etc.). Moreover, during the second phase we deploy a piecewise monotonic calibrator and align the features of the `DataFrame` defined in the previous step so that they match the training schema of the LightGBM model. The piecewise monotonic calibrator transforms raw probabilities into calibrated ones based on threshold points (x, y). Furthermore, The third phase consists of computing fraud-calibrated probabilities from the LightGBM model. The fourth phase for its part, concerns the structuring of sequential metadata. Finally, in the fifth step we implement transition-matrix estimation and forward filtering. This experiment follows Algorithm 20. Table 7.19 lists the meaning of the variables used in this algorithm.

Table 7.19: Notations Used in Algorithm 1 (HMM with LightGBM-as-Emission)

Symbol	**Description**
$x_t \in \mathbb{R}^d$	Feature vector of transaction at time t (d features).
$y_t \in \{0, 1\}$	True label (0 = legitimate, 1 = fraud).
f_{LGB}	LightGBM classifier trained on (x, y).
C_{LGB}	Isotonic calibration function for LightGBM outputs.
p_t	Calibrated fraud probability: $p_t = C_{\text{LGB}}(f_{\text{LGB}}(x_t))$.
τ	Decision threshold applied on p_t.
τ^*	Optimal F1-maximizing threshold on validation.
$A = [a_{ij}]$	2×2 HMM transition matrix: $a_{ij} = P(q_{t+1} = j \mid q_t = i)$.
π_0	Initial state prior (e.g., $[0.999, 0.001]$).
π_t	Posterior state distribution at time t.
ℓ_t	Observation likelihood: $\ell_t = [1 - p_t, p_t]$.
$alert_t$	Binary fraud decision: $alert_t = \mathbb{I}\{\pi_t[\text{fraud}] \geq \tau^*\}$.

Algorithm 20 HMM with GBDT-as-Emission (Training & Inference)

Require: Train/val sequences $\{(x_t, y_t)\}$, validation threshold grid $\mathcal{T}$, blend weight grid $\mathcal{W}$

Ensure: Hybrid calibrated scorer p_t, transition matrix A, initial prior π_0, decision threshold τ^*

Train LightGBM on (x, y) to obtain f_{LGB}
Train XGBoost on (x, y) to obtain f_{XGB}
Fit isotonic calibrators C_{LGB}, C_{XGB} on validation data
Initialize $bestF1 \leftarrow -\infty$, $w^* \leftarrow 0$, $\tau^* \leftarrow 0$
for each $w \in \mathcal{W}$ **do**
    for each $\tau \in \mathcal{T}$ **do**
        For each validation sample x_t:
            $p_{\text{LGB}} \leftarrow C_{\text{LGB}}(f_{\text{LGB}}(x_t))$
            $p_{\text{XGB}} \leftarrow C_{\text{XGB}}(f_{\text{XGB}}(x_t))$
            $p_t \leftarrow w \cdot p_{\text{LGB}} + (1 - w) \cdot p_{\text{XGB}}$
        Compute $F1 \leftarrow F1_score(y_{val}, 1\{p_t \geq \tau\})$
        if $F1 > bestF1$ **then**
            $bestF1 \leftarrow F1$, $w^* \leftarrow w$, $\tau^* \leftarrow \tau$
        end if
    end for
end for
Estimate A (transition matrix) from labeled sequences using Laplace smoothing
Initialize prior π_0 (e.g., $\pi_0 = [0.999, 0.001]$)
Inference Phase:
for each sequence **do**
    $\pi \leftarrow \pi_0$
    for $t = 1 \ldots T$ **do**
        $p_{\text{LGB}} \leftarrow C_{\text{LGB}}(f_{\text{LGB}}(x_t))$
        $p_{\text{XGB}} \leftarrow C_{\text{XGB}}(f_{\text{XGB}}(x_t))$
        $p_t \leftarrow w^* \cdot p_{\text{LGB}} + (1 - w^*) \cdot p_{\text{XGB}}$
        $\ell_t \leftarrow [1 - p_t,\ p_t]$
        $prior \leftarrow A^\top \cdot \pi$
        $post \leftarrow \ell_t \odot prior$
        $\pi \leftarrow post / \sum(post)$
        $alert_t \leftarrow \mathbb{I}\{\pi[fraud] \geq \tau^*\}$
    end for
end for

Once the first two phases have been successfully completed, the third phase computing calibrated fraud probabilities from the LightGBM model produced the following output: `p_base min/max:` 1×10^{-9} `0.999999999`. This min/max value is a technical validation that confirms the probability vector is indeed calibrated and bounded within a coherent interval. The fourth phase, which involved structuring the sequential metadata, yielded the result shown in Table 7.20. That table demonstrates that all transactions are grouped into a single unique sequence (with `__seq__=0`). Their order is defined by a synthetic temporal index (`__t__=0,1,2,...`).

In practice, if identifiers such as `wallet_id` or timestamps (`timestamp`) were available, there would have been multiple independent sequences per user/account. This arrangement allows the construction of ordered sequences required to apply the HMM. Evidently,

this model relies on modeling transitions between hidden states over time. Furthermore, Fig. 7.16 presents the results of the transition-matrix estimation. This matrix describes two distinct latent states. Notably state 0 for a legitimate transaction and state 1 for a fraudulent one. By the way, the estimated probabilities reveal a particular pattern. Indeed, when a transaction is classified as legitimate, there is roughly a 67 % chance that the next transaction will also be legitimate and a 33 % chance it will be fraudulent. Conversely, when a transaction is predicted to be fraudulent, the probability that the following one becomes legitimate is again about 67 %, while the chance it remains fraudulent is 33 %. Subsequently, the immediate history does not significantly influence fraud prediction. Moreover, the transition matrix captures the distribution of post-calibration predictions rather than raw transaction statistics. This approach offers advantages in terms of robustness and auditability. In particular by avoiding strong temporal biases that could complicate decision interpretation. However, it also exposes a conceptual limitation. In its current form it cannot capture complex temporal dynamics.

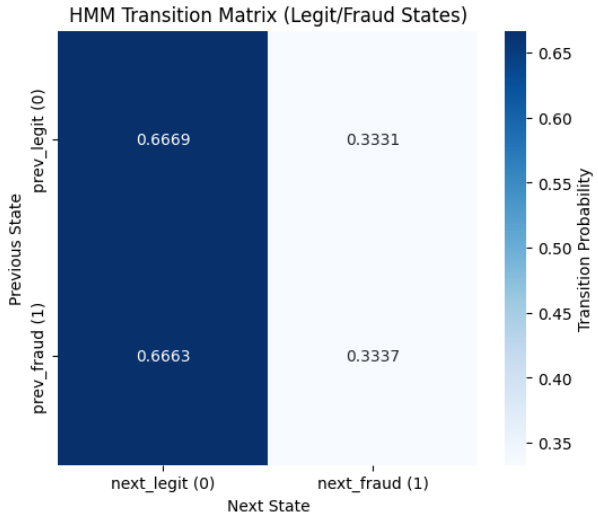


Figure 7.16: Transition matrix of the Hidden Markov Model (HMM) for legitimate and fraud states. The probabilities are as follows: from legitimate (0) to legitimate (0): 0.6669; from legitimate (0) to fraud (1): 0.3331; from fraud (1) to legitimate (0): 0.6663; and from fraud (1) to fraud (1): 0.3337. The matrix exhibits a marked symmetry in transitions, which indicates a weak temporal dependency between states.

Table 7.20: Build sequence metadata (seq_id, time)

SEQ_COL='_seq_'	**TIME_COL='_t_'**
0	0
0	1
0	2
0	3
0	4

7.6.3.2 Comparative Results

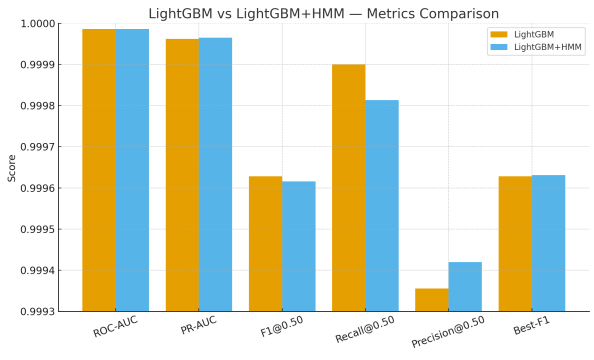
In order to assess the proposed model, a comparative study with the calibrated baseline LightGBM was conducted. Table 7.21 displays the performance metrics of both systems.

Evidently, the numerical analysis shows that even without the contribution of the HMM, the baseline model delivers outstanding results.

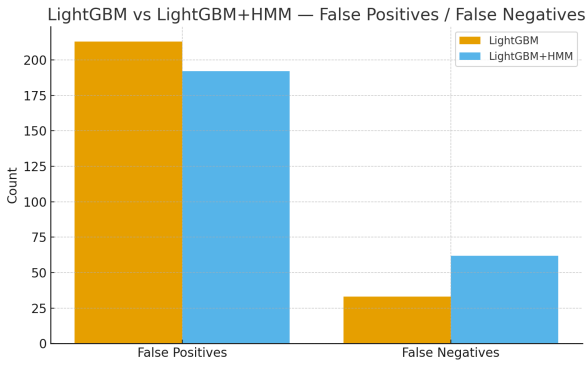
Fig. 7.17 demonstrates that the integration of the HMM does not alter the ROC-AUC. This confirms, with ease, that the relative ordering of predictions remains unchanged. Conversely, the PR-AUC improves slightly to **0.999965**. Moreover, at an identical threshold ($\tau = 0.5$), a clear reduction in false positives is observed (from 213 to 192). All things considered, within an operational setting, integrating the HMM stabilises the decisions of the baseline LightGBM model and markedly reduces false positives. Nevertheless, it should be noted that this benefit can only be fully realised by readjusting τ threshold after smoothing (rather than keeping it arbitrarily at 0.5). In an offline wallet where minimising false positives is crucial for ensuring a smooth user experience, the shift in the Precision/Recall trade-off proves particularly advantageous.

Table 7.21: Comparative Performance: LightGBM vs. LightGBM+HMM

Metric	**LightGBM**	**LightGBM+HMM**
ROC-AUC	0.999986	0.999986
PR-AUC	0.999962	0.999965
F1@0.50	0.999628	0.999616
Recall	0.999900	0.999813
Precision	0.999356	0.999420
Best-F1	0.999628 (τ=0.533)	0.999631 (τ=0.280)
FP at τ=0.5	213	192
FN at τ=0.5	33	62
Confusion Matrix at τ=0.5		
	LightGBM	**LightGBM+HMM**
Legitimate	991,787/213	991,808/192
Fraud	33/330,753	62/330,724



(a) Metrics Comparison between LightGBM and LightGBM + HMM



(b) Error Trade-off (FP vs FN)

Figure 7.17: Comparison between LightGBM and LightGBM + HMM: (a) Metrics and (b) Error Trade-off

7.7 Discussion

As CBDC initiatives evolve globally, offline payment capabilities are increasingly seen as a critical requirement. In order to satisfy the auditability requirement in financial trans-

Table 7.22: Unified Comparison of Fraud Detection Approaches in Literature and Our Work

Approach	Description	Reported Metrics
[145]	Stacking (XGBoost+LightGBM+CatBoost, optimized hyperparams)	Acc: 0.99, Prec: 0.99, Rec: 0.99, F1: 0.99, AUC: 0.99
[146]	LightGBM (XAI, optimized hyperparams)	Acc: 98.3%, F1: 0.70, ROC-AUC: 0.96
[147]	Stacking (RF+AdaBoost+XGBoost+LightGBM+ExtraTrees, LR meta)	Rec: 0.8246, AUC: 0.8146
[148]	Stacking (RF+DT, RF meta)+K-SMOTEENN+feature construction	F1: 0.92, Prec: 0.95, Rec: 0.88, ROC-AUC: 1.00
[149]	Optimized stacking (MLP+kNN+SVM, LR meta, SMOTE+GAN)	Acc: 99.86%, F1 improvement: 16%
[150]	Hybrid Ensemble (DT+RF+kNN+MLP, Grid Search+IHT-LR)	Not explicitly reported; outperforms baseline XGBoost
Our work	Rule-based filtering + calibrated GBDTs, extended with HMM smoothing. Variants: • Rule-based + LightGBM: rule filter + calibrated LGBM • RF baseline: reference tree ensemble • Rule-based + XGBoost: calibrated boosted trees • Rule-based+LGBM+HMM: GBDT-as-emission + 2-state HMM	• PR-AUC: 1.00, Prec: 0.993, Rec: 1.000, F1: 0.996 (LGBM) • ROC-AUC: 1.000, PR-AUC: 1.000, Prec: 0.985, Rec: 1.000 (RF) • ROC-AUC: 0.99999, PR-AUC: 1.000, Prec: 0.985, Rec: 1.000 (XGB) • ROC-AUC: 0.999986, PR-AUC: 0.999965, Prec: 0.99942, Rec: 0.999813, F1: 0.99963 (LGBM+HMM)

* F1 inferred from precision/recall when not explicitly provided.

actions, several solutions rely on probabilistic approaches [103, 109]. These approaches benefit from a low memory footprint for deployment. However, in terms of audibility, they suffer from a lack of local explainability. The transactions are only examined sequentially and not individually. In this context, fraud risks grow significantly due to the absence of real-time validation, making Explainable AI (XAI) an essential component for both proactive security and post-hoc auditability [151–154]. Within this catalogue of solutions, several proposed models offer limited interpretability in addition to being trained with datasets applied to credit card transactions [155]. Consequently, they are only relevant for online fraud detection and therefore cannot adequately address the challenges posed by offline fraud. In contrast, stacking ensemble models demonstrate remarkable performance and explainability [145, 156, 157]. Despite their resistance to over-fitting, their heavy memory footprint poses a barrier to deployment in resource-constrained environments. The benchmark performed on the various tested models contextualises this research by highlighting their ability to overcome the aforementioned limitations. Table 7.22 illustrates the comparison of the proposed models' performance with selected state-of-the-art works.

Thus, based on the integration of a rule engine, three variants were proposed. The first variant concerns the configuration of the rule engine with the LightGBM model. As

a result, the confidence biases traditionally associated with uncalibrated models are considerably reduced [146]. From an operational standpoint, the LightGBM variant stands out for its rapid loading time and low memory footprint. Similarly to LightGBM, the XGBoost variant also exhibits remarkable performance. Still, these results confirm observations in the literature where XGBoost is often praised for its robustness and generalisation ability [145, 157]. Consequently, even though this performance comes at a higher computational cost, XGBoost remains a relevant choice in production contexts. In addition, the LightGBM + HMM variant retains the same ROC-AUC as plain LightGBM while slightly improving PR-AUC. However, it is truly in the realm of explainability that HMM offers a clear advantage over other solutions proposed in the literature. HMM acts as an adaptive Bayesian filter capable of exploiting unlabelled sequences to enhance alert reliability. This semi-supervised feature is an additional asset for fraud detection in the offline CBDC context. Finally, the HMM variant stands out by integrating three components in a single pipeline: explainability, ML supervision, and sequential dynamics.

7.8 Conclusion

The work presented in this chapter has shown that fraud in a CBDC offline environment can be effectively detected. In this context, the proposed framework complements cryptographic fraud-detection schemes by means of a hybrid approach that combines deterministic rules and lightweight machine learning. By scrutinising the construction of the OffSim dataset, the adversary model, the taxonomy of attacks, and the overall architecture, the chapter delivers two major contributions. First, the solution is able to identify both overt frauds via the rule engine and subtler behaviours through quantified LightGBM and XGBoost models. Thus, the HMM adds sequential gain at a lightweight cost. The results of each model and the benchmarks performed confirm that this approach meets efficiently the constraints identified in the introduction namely the lack of connectivity, limited resources, and transparency requirements. Subsequently, these results have several important implications. They first show that explainability of AI models is not incompatible with deployment on constrained terminals. Next, they reveal that near-perfect performance can be achieved without resorting to expensive deep architectures. The next chapter will conclude this dissertation.

7.9 Chapter Summary

In response to subtle fraud cases escaping cryptographic schemes, this chapter presents a hybrid and explainable framework for fraud detection in CBDC offline payments. Thus, the system operating locally on constrained devices combines a deterministic rule engine with a quantified LightGBM/XGBoost model, to which are attached explainability tools such as SHAP, LIME, PDP, etc. Consequently, experimental results attest to exceptional performance both in precision and recall, corroborating the relevance of the approach for CBDC offline contexts. Reflecting on this, by offering an entirely explainable solution deployable on smartphones and tailored to offline environments, this chapter completes and extends the foundations laid in previous chapters. Notably it highlights the possibility of detecting varied fraud cases in a disconnected environment while providing transparency compatible with regulatory compliance; then, it demonstrates that advanced operational security for an offline CBDC system requires a coherent articulation of robust cryptographic protocols and explainable AI tools.

Algorithm 19 Rules-Based Engine Implementation

```
1: const LOW ← 1; MEDIUM ← 2; HIGH ← 3
2: risk ← LOW; fraud_flag ← false
3: if ctx.device_jailbroken ∨ ctx.clock_drift_min > 60 then
4:     return (HIGH, true)                                          ▷ device integrity
5: end if
6: if ¬ ctx.sender_authenticated then
7:     return (HIGH, true)                                          ▷ sender auth
8: end if
9: if ¬ ctx.is_offline then
10:    return (LOW, false)                                          ▷ online ⇒ LOW
11: end if
12: if ctx.double_spend_flag then
13:    return (HIGH, true)                                          ▷ double spend
14: end if
15: if ctx.ledger_sync_age_h < 24 then
16:    risk ← LOW
17: else if ctx.ledger_sync_age_h ≤ 72 then
18:    risk ← MEDIUM
19: else
20:    risk ← HIGH
21: end if
22: if ctx.tx_type = "P2P" then
23:    if ctx.amount > 4× ctx.sender_mean_amount then
24:        risk ← HIGH; fraud_flag ← true                           ▷ amount anomaly
25:    end if
26:    if ctx.offline_p2p_24h_count ≥ 10 then
27:        risk ← HIGH                                              ▷ high-frequency P2P
28:    end if
29: else if ctx.tx_type = "POS" then
30:    if ctx.merchant_reputation = LOW then
31:        risk ← HIGH; fraud_flag ← true                           ▷ fake merchant
32:    end if
33:    if ctx.offline_topups_last_window_count ≥ 15 ∧ ctx.offline_topup_amount_range_ok then
34:        if risk < MEDIUM then
35:            risk ← MEDIUM                                        ▷ micro top-up pattern
36:        end if
37:    end if
38: end if
39: if ctx.recipient_suspicious then
40:    risk ← bump(risk)                                            ▷ recipient suspicious
41: end if
42: return (risk, fraud_flag)
```

Chapter 8

Conclusion

This thesis entitled "Offline Payments Requirements for Central Bank Digital Currency" focused on developing a coherent core framework that ensures the security, privacy protection, and trustworthiness of offline transactions while meeting hardware and auditability constraints. Consequently, the adjacent research work deepened theoretical foundations, architectures, and operational mechanisms necessary to implement CBDC payment systems capable of reliable offline operation. Subsequently it addressed cardinal challenges of resilience, service continuity, financial inclusion, and broad accessibility inherent to CBDC transactions. Building on the theoretical and taxonomical contribution, presented at the conference paper [158], I proposed a multi-faceted technical contribution structured around three complementary axes.

8.1 Summary of Technical Contributions

Each contribution proposes a CBDC offline payment framework tailored to a specific context but built upon a tripartite structure: the securisation of the offline transaction protocol, the privacy of transaction data, and the anonymity of digital funds together with their synchronisation to the central bank ledger. In the first thesis contribution we introduced PrivTEE-Pay. PrivTEE-Pay is an offline payment framework designed for a two–tier single ledger infrastructure. We demonstrated that PrivTEE-Pay enables a conventional payment architecture to meet the CBDC offline requirements described earlier. Achieving this alignment required deploying a panel of technologies. Cryptographically, we employed blind signatures based on David Chaum's *eCash 1* model to guarantee both authenticity and anonymity of ecoins. The generation and verification of zk-SNARK proofs, coupled with the use of digital certificates, enabled offline transaction validation, particularly in detecting double-spending attempts by the recipient without violating transaction privacy. On the hardware side, integrating a Trusted Execution Environment (TEE) secured private keys, cryptographic operations, and protected against replay attacks. The theoretical foundations of this contribution were first presented in a conference paper [43] before validating our PrivTEE-Pay prototype through rigorous testing and analysis.

The second thesis contribution centres on integrating a smart card Digivaultcard and a Markovian fraud detection system MarkoPaychain into a conventional two–tier architecture. We showed that this substantially modified conventional architecture mitigates security concerns associated with mobile-phone TEEs. By relocating the secure element to the Digivaultcard, the offline wallet is protected against physical attacks that could compromise key and balance management. Complementing this security dimension, we

introduced MarkoPaychain, which detects suspicious fraud cases by analysing transaction histories and computing transition probabilities of various model states based on historical parameters. This time, we capitalised on Chaum's blind *ecash2* signatures, digital signatures, and zk-SNARK proof generation and verification. Consequently, we ensured full untraceability of ecoins, authenticity verification, and offline transaction validation in a privacy-preserving manner that satisfies auditability requirements. Finally, we presented this contribution in a conference paper [103], where we demonstrated the framework's viability through detailed result analysis.

In the third contribution of this dissertation, we proposed Block-PAD. Block-PAD is a framework built on an extended two-tier architecture that incorporates a distributed ledger. This framework operationalises offline payment under a bimodal scheme: full offline and staged offline. Interoperability is achieved through the integration of digital identity, which allows a conventional infrastructure to interconnect with a CBDC-compatible one. Consequently, funding and defunding operations can now be carried out between systems. Blok-PAD marks a significant milestone in privacy protection. End-to-end privacy (online/offline) is ensured through a mix of technologies: digital identity, PAD technology, RIO, and shared ledgers. This maximisation of privacy aligns with auditability requirements: the RIO supervisory authority can reveal the user's digital identity in case of suspicious transactions. Blok-PAD therefore implements contextual privacy. On the cryptographic side, it relies on blind signatures, zk-SNARKs, a secret key-sharing protocol and transaction proof between validators and trustees (entities of the PAD model), as well as the CKKS algorithm that allows users' balances to be operated upon anonymously. We empirically validated the Blok-PAD concept in a journal article through simulations and detailed results in [65].

For the fourth contribution, we introduced a hybrid framework for advanced fraud protection. This framework complements the cryptographic schemes presented earlier by mitigating residual fraud cases such as Phantom Terminal micro-top-up storms, timestamp tampering, and other micro-recharge anomalies. It couples a rule engine with a machine-learning model. The rule engine acts as a filter to detect manifest fraud scenarios, while the AI system identifies sophisticated fraud cases. We tested and evaluated several AI models in combination with the rule engine: LightGBM, XGBoost, and a hidden Markov chain integration. The results were compared against various stacking ensemble methods. Upon reflection, our results are comparable to or even surpass current state-of-the-art approaches that focus on online transactions. Finally, the training of the machine-learning models was made possible by the simulator Offsim-CBDC we developed, which generated the training dataset for the experiments[136].

8.2 Perspectives

This section marks the conclusion of our thesis by outlining the future directions we envision to extend our contributions.

8.2.1 Extended Requirements

The work presented in this dissertation has highlighted several essential features for deploying offline payment in central-bank digital currencies. Nevertheless, there remain unexplored features that could enrich our various contributions from a prospective standpoint. Below we list some of the key prospects identified. First, programmability is paramount. Offline programmability represents a stepping-stone toward fully offline smart

contracts; it enables time-locked emergency stipends or geo-fenced corporate per diem schemes. Second, post-quantum cryptography must also be considered. We need to adopt a crypto-agile mindset and prepare for the "Q-day." Future research should compress lattice keys below smart-card RAM budgets and design post-quantum blind signatures that preserve cash-like privacy.

8.2.2 PrivTEE-Pay

The primary goal of `PrivTEE-Pay` is to deliver an offline payment system based on a conventional ledger. While the system offers many contributions in terms of privacy, security, and auditability, financial inclusion for the non-banked remains an open question. Technically, this translates into enabling offline payments even when no bank account exists. A possible future direction would be to adopt a non-custodial wallet.

8.2.3 Digivault-card MarkoPayChain

Like `PrivTEE-Pay`, this framework could also be enhanced so that it does not necessarily require a bank account for usage. Furthermore, payment options using the DV-card could be extended to support secure machine-to-machine transactions in IoT environments. Cloud computing offers scalability and reliability features that `MarkoPayChain` can leverage. In our solution, a cloud-based Trusted Certified Authority (TCA) will be employed by the central bank to blindly sign tokens. Additionally, the cloud environment can synchronize offline transactions.

8.2.4 Block-PAD

Regarding `Block-PAD`, several performance-oriented enhancements can be pursued. These include cryptographic acceleration via optimized proof systems, pipeline processing, micro-batching strategies to further increase throughput and lightweight reconciliation mechanisms. Exploring these directions, together with tighter integration into decentralized digital identity standards, will strengthen the scalability, efficiency, and deployability of `Block-PAD` in real-world CBDC environments.

8.2.5 Advanced Fraud Mitigation

For LightGBM + HMM model, an extension could better exploit the sequential structure of the data. Noticeably by incorporating contextual transitions making probabilities depend not only on the previous state but also on other explanatory features. Especially amount, frequency, user profile, etc. Furthermore, despite the high performance of various models, it should be noted that an approach guaranteeing the privacy of training data has not been explored in this work. Future research could investigate methods for preserving training data integrity, akin to those proposed in machine learning

Bibliography

[1] BIS. Project polaris: A handbook for offline payments with cbdc, 2023.

[2] J. Knoerich. China's new digital currency: implications for renminbi internationalization and the us dollar, the (near) future of central bank digital currencies. risks and opportunities for the global economy and society. *Global Politics and Security*, 7(1):45–166, 2021.

[3] A. Aysan and F. Kayani. China's transition to a digital currency does it threaten dollarization? *Asia and the Global Economy*, 2(1):1–6, 2022.

[4] Diners Club International. History of diners club international. https://www.dinersclub.com/about-us/history/, 2025. Accessed: 2025-01-18.

[5] WhisperCash. *WhisperCash Brochure*, 2021. Accessed: 2025-01-18.

[6] F. Stalder and A. Clement. Exploring policy issues of electronic cash: The mondex case. *Canadian Journal of Communication*, 24(2), 1999. doi: 10.22230/cjc.1999v24n2a1099. Accessed: 2025-12-09.

[7] Nick Hughes and Susie Lonie. M-pesa: Mobile money for the "unbanked" — turning cellphones into 24-hour tellers in kenya. *Innovations: Technology, Governance, Globalization*, 2(1–2):63–81, 2007. doi: 10.1162/itgg.2007.2.1-2.63. URL https://direct.mit.edu/itgg/article/2/1-2/63/9485/M-PESA-Mobile-Money-for-the-Unbanked-Turning.

[8] Usman W. Chohan. The double spending problem and cryptocurrencies. *SSRN Electronic Journal*, 2021. doi: 10.2139/ssrn.3090174. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3090174. SSRN ID: 3090174; preprint first posted 23 Dec 2017, last revised 26 Jan 2021.

[9] I. V. Krsul, J. C. Mudge, and A. J. Demers. Method of electronic payments that prevents double-spending, 1998. URL https://patents.google.com/patent/US5839119A/en. U.S. Patent.

[10] H. Armelius, C. A. Claussen, and I. Hull. On the possibility of a cash-like cbdc. Staff memo, Sveriges Riksbank, 02 2021. URL https://www.riksbank.se/globalassets/media/rapporter/staff-memo/engelska/2021/on-the-possibility-of-a-cash-like-cbdc.pdf. Accessed: 2025-12-09.

[11] Jr. R. E. Mckinney, L. P. Shao, Jr. D. C. Rosenlieb, and D. H. Shao. Counterfeiting in cryptocurrency: An emerging problem. In *Handbook of Digital Currency*, pages 173–187. Academic Press, Cambridge, MA, USA, 2015.

[12] M. R. Dorsala, V. N. Sastry, and S. Chapram. Fair protocols for verifiable computations using bitcoin and ethereum. In *Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, July 2018.

[13] Ripjar. Everything you need to know about central bank digital currencies (cbdcs) and what it means for financial institutions. https://ripjar.com/blog/cbdcs-central-bank-digital-currencies/, 10 2021. Accessed: 2025-12-09.

[14] A. McCullagh and W. Caeli. Non-repudiation in the digital environment. *First Monday*, 5(8), August 2000. doi: 10.5210/fm.v5i8.778.

[15] Steve Kremer, Olivier Markowitch, and Jianying Zhou. An intensive survey of fair non-repudiation protocols. *Computer Communications*, 25(17):1606–1621, November 2002. URL http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/comcom02.pdf.

[16] European Central Bank. Eurosystem report on the public consultation on a digital euro. https://www.ecb.europa.eu/pub/pdf/other/Eurosystem_report_on_the_public_consultation_on_a_digital_euro~539fa8cd8d.en.pdf. Accessed: 2024-11-11.

[17] N. Atako. Privacy beyond possession: Solving the access conundrum. *Vanderbilt Journal of Entertainment and Technology Law*, 23(4), 2021. doi: 10.2139/ssrn.3798838.

[18] Digital Euro Association. Privacy and cbdcs. https://blog.digital-euro-association.de/privacy-and-cbdcs-dea-working-group. Accessed: 2024-11-11.

[19] R. Kozakiewicz. Technologies des stablecoins et monnaies numériques de banque centrales. *Revue Économique et Financière*, (149):107–118, 2023. doi: 10.3917/ecofi.149.0107.

[20] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf. Accessed: 2024-11-11.

[21] D. Chaum and S. Brands. Minting electronic cash. *IEEE Spectrum*, 34(2):30–34, February 1997. doi: 10.1109/6.570825.

[22] J. Camenisch, U. Maurer, and M. Stadler. Digital payment systems with passive anonymity-revoking trustees. *Journal of Computer Security*, 5(1):69–89, January 1997. doi: 10.3233/JCS-1997-5104.

[23] Z. Eslami and M. Talebi. A new untraceable off-line electronic cash system. *Electronic Commerce Research Application*, 10(1):59–66, jan-feb 2011. doi: 10.1016/j.elerap.2010.08.002.

[24] C. Wang, H. Sun, H. Zhang, and Z. Jin. An improved off-line electronic cash. In *Proceedings of the International Conference on Computational and Information Sciences (IEEE)*, pages 21–23, June 2013. doi: 10.1109/ICCIS.2013.122.

[25] G. Van Damme, K. M. Wouters, H. Karahan, and B. Preneel. Offline nfc payments with electronic vouchers. In *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds*, August 2009. doi: 10.1145/1592606.1592613.

[26] F. Omondi. Nigeria's oyapay now integrates the use of bluetooth and qr payments for offline use. https://innov8tiv.com/nigerias-oyapay-now-integrates-the-use-of-bluetooth-and-qr-payments-for-offline-use/. Accessed: 2024-11-11.

[27] BIS. Central bank digital currencies: foundational principles and core features, rapport no1 in a series of collaborations from a group of central banks. https://www.bis.org/publ/othp33.htm, . Accessed: 2024-11-11.

[28] A. Opare and K. Kim. A compendium of practices for central bank digital currencies for multinational financial infrastructures. *IEEE Access*, 8:110810–110847, 2020.

[29] R. Auer, G. Cornelli, and J. Frost. Rise of the central bank digital currencies: Drivers, approaches and technologies. https://www.bis.org/publ/work880.htm. Accessed: 2024-11-11.

[30] Capgemini. *Building central bank digital currency growth opportunities ahead for central and commercial banks*. Capgemini, 2023.

[31] C. Berg. Interoperability as a critical design choice for central bank digital currencies. https://ssrn.com/abstract=4205405. Accessed: 2024-11-11.

[32] M. Bech, C. Boar, Daniel Eidan, Philipp Haene, Henry Holden, and Wee Kee Toh. Using cbdcs across borders: lessons from practical experiments. https://www.bis.org/publ/othp51.pdf. Accessed: 2024-11-11.

[33] BIS. Project aurora, the power of data, technology and collaboration to combat money laundering across institutions and borders. https://www.bis.org/about/bisih/topics/fmis/aurora.htm, . Accessed: 2024-11-11.

[34] BIS. Central bank digital currencies: ongoing policy perspectives. https://www.bis.org/publ/othp65.htm, . Accessed: 2024-11-11.

[35] B. Scollan and E. Darling. Designing digital currency wallets for broad adoption. *Journal of Payments Strategy & Systems*, 17(1):96–106, 2023.

[36] L. V. Schumacher. Decoding the digital euro, friend or foe, make up your mind, 2024.

[37] G. Samid. A level paying field: Cryptographic solutions towards social accountability and financial inclusion. *Cryptology ePrint Archive*, 2022. 2022/130.

[38] B. Brodsky, A. Dubey, and D. T. Lucas. *Enabling offline payments in an online world: A practical guide to offline payment design*. Lipis Advisors Gmbh, 2023.

[39] M. Christodorescu, W. C. Gu, R. Kumaresan, M. Minae, M. Ozdayi, B. Price, S. Raghuraman, M. Saad, C. Sheffield, M. Xu, and M. Zamani. Towards a two-tier hierarchical infrastructure: An offline payment system for central bank digital currencies. *arXiv Preprint*, arXiv:2012.08003, 2020.

[40] Ricky Lamberty, Daniel Kirste, Niclas Kannengiesser, and Ali Sunyaev. Hybcbdc: A design for central bank digital currency systems enabling digital cash. *IEEE Access*, 12:137712–137728, 2024. doi: 10.1109/ACCESS.2024.3458451. URL https://publikationen.bibliothek.kit.edu/1000174152/154642087. Accessed: 2025-12-10.

[41] R. Auer and R. Böhme. The technology of retail central bank digital currency. https://www.bis.org/publ/qtrpdf/r_qt2003j.pdf. Accessed: 2024-11-11.

[42] H. Guo and X. Yu. A survey on blockchain technology and its security. *Blockchain: Research and Applications*, 3(2):1–15, 2022.

[43] O. Atangana, M. Barbier, L. Khoukhi, and W. Royer. Securing privacy in offline payment for retail central bank digital currencies: A comprehensive framework. In *Proceedings of the Conference on Blockchain and Cryptocurrency (B2C'23)*, pages 25–32, October 2023.

[44] I. S. Igboanusi, K. P. Dirgantoro, J. M. Lee, and D. S. Kim. Blockchain side implementation of pure wallet(pw): An offline transaction architecture. *ICT Express*, 7 (3):327–334, 2021.

[45] R. S. Rasheed, K. H. A. Al-Shqeerat Ahmed, S. Ghorab, F. S. AbuOwaimer, and A. Ahmed AbuSamra. Blockchain mobile wallet with secure offline transactions. *Computers, Materials and Continua*, 75(2):2905–2919, 2023.

[46] W. Ha Yin. *Issues in Electronic Payment Systems: A New Offline Transferable E-Coin Scheme and a New Offline E-Check Scheme*. Ph.d. dissertation, Chinese University of Hong Kong, August 2001.

[47] C. Popescu. A secure and efficient off-line electronic transaction protocol. *Studies in Informatics and Control*, 19(1):27–34, 2010. ISSN 1220-1766. doi: 10.24846/v19i1y201003.

[48] W. Jie, W. Qiu, A. Koe, J. Li, Y. Wang, and Y. Wu. A secure and flexible blockchain-based offline payment protocol. *IEEE Transactions on Computers*, 73(2):408–421, February 2024. doi: 10.1109/TC.2023.3331823.

[49] B. Yang, D. Tong, and Y. Zhang. Dot-m: A dual offline transaction scheme of central bank digital currency for trusted mobile devices. In *Proceedings of the 16th International Conference on Network and System Security (NSS)*, pages 233–248, December 2022. doi: 10.1007/978-3-031-23020-2_13.

[50] B. Yang, D. Tong, Q. Yu, F. Wei, and Y. Zhang. A dual offline anonymous e-payment scheme for mobile devices based on tee and se. *Journal of Software*, 35(8), 2024. doi: 10.13328/j.cnki.jos.007115.

[51] A. Dogan, M. Takaoglu, T. Dursun, and E. Olcer. Smart card based offline payment system for central bank digital currencies. In *Proceedings of the Conference on Blockchain and Cryptocurrency (B2C'22)*, pages 114–127, November 2022.

[52] M. Aprile, L. Mainetti, E. Mele, and R. Vergallo. A sustainable approach to delivering programmable peer-to-peer offline payments. *Sensors*, 23(2):1336, 2023. doi: 10.3390/s23031336.

[53] C. Beer, S. Zingg, and K. Kostiainen. Payoff: A regulated central bank digital currency with private offline payments. *arXiv Preprint*, arXiv:2408.06956, August 2024. doi: 10.48550/arXiv.2408.06956.

[54] E. Benoist. Practical offline payments using one-time passcodes. *SUERF Policy Brief*, (622), 2023.

[55] M. Franklin and M. Yung. Secure and efficient off-line digital money (extended abstract). In A. Lingas, R. Karlsson, and S. Carlsson, editors, *Automata, Languages and Programming. ICALP 1993. Lecture Notes in Computer Science*, volume 700, pages 265–276. Springer, Berlin, Heidelberg, 1993. doi: 10.1007/3-540-56939-1_78.

[56] D. Chaum, C. Grothoff, and T. Moser. How to issue central bank digital currency? *arXiv Preprint*, arXiv:2103.00254v1 [econ.GN], 2021. doi: 10.48550/arXiv.2103.00254.

[57] D. Chaum and T. Moser. *eCash 2.0, Inalienably private and quantum-resistant to counterfeiting*. SNB Working Paper, 2022. Accessed: 2024-11-11.

[58] BIS. *Project Tourbillon, Exploring Privacy, Security and Scalability for CBDCs Final Report*, November 2023. Accessed: 2024-11-11.

[59] M. Adams, L. Boldrin, R. Ohlhausen, and E. Wagner. An integrated approach for electronic identification and central bank digital currencies. *Journal of Payments Strategy Systems*, 15(3):287–305, 2021. doi: 10.5281/zenodo.271482636.

[60] C. Garman, M. Green, and I. Miers. Accountable privacy for decentralized anonymous payments. In *International Conference on Financial Cryptography and Data Security*, 2017. doi: 10.1007/978-3-662-54970-4_5. Accessed: 2024-11-11.

[61] J. Lovejoy, C. Fields, M. Virza, T. Frederick, D. Urness, K. Karwaski, A. Brownworth, and N. Narula. A high performance payment processing system designed for central bank digital currencies. *Cryptology ePrint Archive*, (Paper 2022/163), 2022. Accessed: 2024-11-11.

[62] E. Androulaki, J. Camenisch, A. De Caro, M. Dubovitskaya, K. Elkhiyaoui, and B. Tackmann. Privacy-preserving auditable token payments in a permissioned blockchain system. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies (AFT '20)*, pages 255–267, 2020. doi: 10.1145/3419614.3423259.

[63] G. Goodell, D. R. Toliver, and Nakib H. D. A scalable architecture for electronic payments. In *Financial Cryptography and Data Security. FC 2022 International Workshops*, pages 645–678. 2023. doi: 10.1007/978-3-031-32415-4_38.

[64] M. Babel, A. Bechtel, J. Gross, B. Schellinger, and J. Sedlmeir. *Designing a Central Bank Digital Currency with Support for Cash-like Privacy*, 2021. Accessed: 2024-11-11.

[65] O. Atangana, M. Barbier, L. Khoukhi, and A. Kockam. Block-pad: A blockchain-enabled framework for resilient and flexible cbdc transactions leveraging digital identity. *Computer Networks*, 2025.

[66] A. Dmitrienko, D. Noack, and M. Yung. Secure wallet-assisted offline bitcoin payments with double-spender revocation. In *Proceedings of the Conference on Blockchain and Cryptocurrency*, pages 520–531, 2017. doi: 10.1145/3052973.3052980.

[67] T. Takahashi and A. Otsuka. Short paper: Secure offline in bitcoin. In *Financial Cryptography and Data Security*, volume 11599 of *Lecture Notes in Computer Science*, pages 12–20. Springer, Cham, 2020.

[68] Hackernoon. Educational byte: Exploring offline cryptocurrency transactions. https://hackernoon.com, 2024. Accessed: 2024-11-11.

[69] N. Ivanov and Q. Yan. System-wide security for offline payment terminals. *arXiv Preprint*, arXiv:2107.08490v1 [cs.DC], July 2021. doi: 10.48550/arXiv.2107.08490. Accessed: 2024-11-11.

[70] A. Rial and A. M. Piotrowska. Compact and divisible e-cash with threshold issuance. *Proceedings on Privacy Enhancing Technologies*, 2023(4):381–415, 2023. doi: 10.56553/popets-2023-0116.

[71] EMVCo. *EMV Integrated Circuit Card Specifications for Payment Systems Book 3 Application Specification*, 2022. Accessed: 2024-11-11.

[72] Cbdctracker. Cbdctracker. https://cbdctracker.org/, 2024. Accessed: 2024-11-11.

[73] T. T. Huynh, T. D. Nguyen, and H. Tan. A survey on security and privacy issues of blockchain technology. In *Proc. Int. Conf. System Science and Engineering (ICSSE)*, pages 362–367, 2019. doi: 10.1109/ICSSE.2019.8823094.

[74] B. Brodsky, A. Dubey, and D. T. Lucas. Enabling offline payments in an online world, scalability, 2023.

[75] B. Brodsky, A. Dubey, and D. T. Lucas. *Enabling offline payment in an online world, Privacy considerations*. Lipis Advisors Gmbh, 2023.

[76] B. Brodsky, A. Dubey, and D. T. Lucas. *Enabling offline payments in an online world, A practical guide to offline payment security*. Lipis Advisors Gmbh, 2023. Accessed: 2024-11-11.

[77] Y. Chu, J. Lee, S. Kim, H. Kim, Y. Yoon, and H. Chung. Review of offline payment function of cbdc considering security requirements. *Applied Sciences*, 12:4488, 2022. doi: 10.3390/app12094488.

[78] L. Aboulaiz, B. Akintade, H. Daud, M. Lansey, M. Rodden, L. Sawyer, and M. Yip. Offline payments: Implications for reliability and resilience in digital payment systems. https://www.federalreserve.gov/econres/notes/feds-notes/offline-payments-implications-for-reliability-and-resiliency-in-digital-payment-systems-20240816.html, 2024. Accessed: 2024-11-11.

[79] Sreekanth Malladi, Jim Alves-Foss, and Robert B. Heckendorn. On preventing replay attacks on security protocols. In *Proceedings of the International Conference on Security and Management*, pages 77–83, 2002. Accessed: 2025-12-18.

[80] F. Tronnier. Privacy in the age of central bank digital currency. In M. Friedewald, S. Schiffner, and S. Krenn, editors, *Privacy and Identity Management. Privacy and Identity 2020*, volume 619 of *IFIP Advances in Information and Communication Technology*, pages 96–114. Springer, Cham, April 2021. doi: 10.1007/978-3-030-72465-8_6.

[81] R. Auer, R. Böhme, J. Clark, and D. Demirag. Mapping the privacy landscape for central bank digital currencies. *Queue*, 20(4):16–38, August 2022. doi: 10.1145/3561796.

[82] A. Kiayias, M. Kohweiss, and A. Sarencheh. Peredi: Privacy-enhanced, regulated and distributed central bank digital currencies. *Blockchain: Research and Applications*, (100176), December 2023.

[83] Panagiotis Michalopoulos, Odunayo Olowookere, Nadia Pocher, Johannes Sedlmeir, Andreas Veneris, and Poonam Puri. Compliance design options for offline cbdcs: Balancing privacy and aml/cft. In *Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2024. URL https://orbilu.uni.lu/bitstream/10993/61334/1/Michalopoulos%20P%2C%20Olowookere%20O%2C%20Pocher%20N%2C%20Sedlmeir%20J%2C%20Veneris%20A%2C%20Puri%20P%20%282024%29%20Compliance%20Design%20Options%20for%20Offline%20CBDCs%20Balancing%20Privacy%20and%20AML-CFT.pdf. Accessed: 2025-12-10.

[84] Sebastian Elfors. White paper: Using fido for the eudi wallet. https://fidoalliance.org/white-paper-using-fido-for-the-eudi-wallet/, April 2023. Accessed: 2024-11-11.

[85] E. Androulaki, M. Brandenburger, A. De Caro, K. Elkhiyaoui, A. Filios, L. Funaro, Y. Manevich, and S. Natarajan. A framework for resilient, transparent, high-throughput, privacy-enabled central bank digital currencies. *Cryptology ePrint Archive, Paper 2023/1717*, 2023. https://eprint.iacr.org/2023/1717.

[86] OpenWallet Foundation. The openwallet foundation. https://openwallet.foundation/, 2024. Accessed: 2024-11-11.

[87] Ledger Insights. Indian cbdc pilot progresses to programmable, offline usage. https://www.ledgerinsights.com/india-cbdc-programmable-offline/, 2024. Accessed: 2024-11-11.

[88] European Central Bank. Exploring anonymity in central bank digital currencies. https://www.ecb.europa.eu/press/intro/publications/pdf/ecb.mipinfocus191217.en.pdf, 2019. In Focus, Issue no. 4, December 2019. Accessed: 2024-11-11.

[89] European Central Bank. Annex 1: Functional and non-functional requirements linked to the market research for a potential digital euro implementation. https://www.ecb.europa.eu/euro/digital_euro/timeline/profuse/shared/pdf/ecb.dedocs230113_Annex_1_Digital_euro_market_research.en.pdf, January 2023. Accessed: 2024-11-11.

[90] Bank of Ghana. The ecedi report. https://www.bog.gov.gh/news/the-ecedi-report/, 2024. Accessed: 2024-11-11.

[91] G+D Filia. Unplugged brings offline capability to digital payments. https://www.gi-de.com/en/group/press/press-releases/g-d-filia-unplugged-brings-offline-capability-to-digital-payments, 2024. Accessed: 2024-11-11.

[92] O. Caudevilla and H. M. Kim. The digital yuan and cross-border payments: China's rollout of its central bank digital currency. University of Hong Kong Faculty of Law Research Paper No. 2023/30, 2022. Accessed: 2024-11-11.

[93] People's Bank of China. Progress of research and development of e-cny in china. http://www.pbc.gov.cn/en/3688110/3688172/4157443/4293696/2021072014364791207.pdf, 2021. Accessed: 2024-11-11.

[94] Finextra. China's e-cny app launches offline payments. https://www.finextra.com/newsarticle/41600/chinas-e-cny-app-launches-offline-payments, 2024. Accessed: 2024-11-11.

[95] Sveriges Riksbank. E-krona pilot phase 4. https://www.riksbank.se/en-gb/payments--cash/e-krona/e-krona-reports/e-krona-pilot-phase-4/, 2024. Accessed: 2024-11-11.

[96] S. Lan and S. V. Kashani. Influence of e-krona and e-cny on individual privacy: A comparative study between sweden and china. Master's thesis, Master's Thesis, September 2021. Accessed: 2024-11-11.

[97] Idemia. Offline cbdc payments, 2023.

[98] GlobalPlatform. Globalplatform tee white paper. https://globalplatform.org/wpcontent/uploads/2018/04/GlobalPlatform_TEE_Whitepaper_2015.pdf, 2015.

[99] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proc. 3rd Conf. Innovations in Theoretical Computer Science (ITCS)*, pages 326–349, January 2012. doi: 10.1145/2090236.2090263.

[100] Matteo Bertaccini. *A Guide to Algorithms in Blockchain, Quantum Cryptography, Zero-Knowledge Protocols, and Homomorphic Encryption*. Packt Publishing, 2022.

[101] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. doi: 10.1109/TIT.1976.1055638. URL https://ee.stanford.edu/~hellman/publications/24.pdf. Accessed: 2025-12-11.

[102] GSA-Global. *CBDC Wallets and the Security Requirements*, 2024. Accessed: 2024-11-11.

[103] Olivier Atangana, Lyes Khoukhi, Morgan Barbier, and Joël Di Manno. Securing offline cbdc transactions: Digivault card and markopaychain with mobile phone integration. In *Proceedings of the 2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, pages 33–40, 2025. doi: 10.1109/ICIN64016.2025.10942801. URL https://doi.org/10.1109/ICIN64016.2025.10942801. Accessed: 2025-12-11.

[104] Houssem Moudoud, Soumaya Cherkaoui, and Lamia Khoukhi. Towards a scalable and trustworthy blockchain: Iot use case. In *Proceedings of the IEEE International Conference on Communications (ICC'21)*, pages 1–6, Montreal, Canada, June 2021.

[105] Zakaria A. El Houda, Abdelhakim Hafid, and Lamia Khoukhi. Brainchain: A machine learning approach for protecting blockchain applications using sdn. In *Proceedings of the IEEE International Conference on Communications (ICC'20)*, pages 1–6, Dublin, Ireland, June 2020.

[106] Tal Ashur and Samuel Dhooghe. Marvellous: a stark-friendly family of cryptographic primitives. *Cryptology ePrint Archive*, (2018/046), 2018. URL https://eprint.iacr.org/2018/046.

[107] Y. Kou, Ch-T. Lu, S. Sirwongwattana, and Y-P. Huang. Survey of fraud detection techniques. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, volume 2, pages 749–754, Taipei, Taiwan, 2004. doi: 10.1109/ICNSC.2004.1297040.

[108] J. N. Luo and M. H. Yang. Emv-compatible offline mobile payment protocol with mutual authentication. *Sensors*, 19(21), 2019. doi: 10.3390/s19214611.

[109] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar. Credit card fraud detection using hidden markov model. *IEEE Transactions on Dependable and Secure Computing*, 5(1):37–48, 2008. doi: 10.1109/TDSC.2007.70228.

[110] J. Schasfoort. Central bank digital currency transaction simulator. https://github.com/blackrhinoabm/CBDC-sim, 2024.

[111] Elli Androulaki, Angelo De Caro, Kaoutar El Khiyaoui, Romain Gay, Rebekah Mercer, and Alessandro Sorniotti. Secure and privacy-preserving cbdc offline payments using a secure element. Cryptology ePrint Archive, Paper 2024/1746, 2024. URL https://eprint.iacr.org/2024/1746. Accessed: 2025-12-17.

[112] Khaled Baqer, Ross Anderson, Lorna Mutegi, Jeunese Adrienne Payne, and Joseph Sevilla. Digitally: Piloting offline payments for phones. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 131–143, Santa Clara, CA, 2017. USENIX Association. ISBN 978-1-931971-39-3. URL https://www.usenix.org/conference/soups2017/technical-sessions/presentation/baqer.

[113] Nektarios Evangelou, Rowdy Chotkan, Bulat Nasrulin, and Jérémie Decouchant. Taming double-spending in offline payments with reputation-weighted loan networks. https://arxiv.org/abs/2504.05143, 2025. arXiv:2504.05143 [cs.CR]; Accessed: 2025-12-17.

[114] V. Buterin, J. Illum, M. Nadler, F. Schär, and A. Soleimani. Blockchain privacy and regulatory compliance: Towards a practical equilibrium. *Blockchain: Research and Applications*, (100176), December 2023.

[115] Y. C. Liao, Z. Liu, Y. Tseng, and R. Tso. Blockchain-based confidential payment system with controllable regulation. In *Proceedings of the 17th International Conference on Information Security Practice and Experience*, pages 39–56, Taipei, Taiwan, November 2022.

[116] J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic encryption for arithmetic of approximate numbers. In T. Takagi and T. Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, volume 10624 of *Lecture Notes in Computer Science*. Springer, Cham, 2017.

[117] TTC. Draft eu us ttc digital identity mapping exercise report, December 2023.

[118] SW7Group. Pad privacy-preserving accountable decryption. https://padtech.s3.eu-central-1.amazonaws.com/pad-whitepaper/white_paper_2021.pdf, October 2021.

[119] M. Rahmilevich. Towards a scalable privacy-preserving and regulatable cbdc framework. https://static.sched.com/hosted_files/hgf22/f6/Oracle_CBDC_Presentation_v6.pdf, 2022.

[120] I. David et al. Do you still need a manual smart contract audit?, 2023. arXiv:2306.12338v2.

[121] Ethereum Foundation. Remix ethereum ide. https://remix.ethereum.org. Accessed: 2025-03-25.

[122] Truffle Suite. Ganache: Personal blockchain for ethereum development. https://trufflesuite.com/ganache/. Accessed: 2025-03-25.

[123] Tribler. Ipv8: A secure peer-to-peer communication library. https://github.com/Tribler/py-ipv8. Accessed: 2025-03-25.

[124] Olivier Atangana, Lyes Khoukhi, Morgan Barbier, and Andre Kockam. Block-pad: Experimental implementation repository. https://github.com/Olivieratangana/Block-PAD, 2025. GitHub repository.

[125] D. Ballaschk and J. Paulick. The public, the private and the secret: Thoughts on privacy in central bank digital currencies. *Journal of Payments Strategy and Systems*, 15(3):277–286, 2023.

[126] R. Barresi. La confidentialité des paiements: Du xviiie siècle à l'euro numérique, les monnaies numériques et les cryptoactifs. *Revue Économique et Financière*, (149): 257–269, 2023.

[127] H. Jang et al. A survey on security and privacy in blockchain-based central bank digital currencies. *Journal of Internet Services and Information Security*, 11(3): 16–29, 2022.

[128] G. Fanti and K. Kostiainen. Missing key: The challenge of cybersecurity and central bank digital currency, June 2022.

[129] Observatoire de la sécurité des moyens de paiement. Rapport annuel 2024 de l'observatoire de la sécurité des moyens de paiement. Technical report, Banque de France, 2025. URL https://www.banque-france.fr/system/files/2025-09/OSMP-2024_VF.pdf. Accessed: 2025-12-11.

[130] João Lucas Martins, João Bravo, Ana Sofia Gomes, Carlos Soares, and Pedro Bizarro. Riff: Inducing rules for fraud detection from decision trees. https://arxiv.org/abs/2408.12989, 2024. arXiv:2408.12989 [cs.LG]; Accessed: 2025-12-11.

[131] Nuno Carneiro, Gonçalo Figueira, and Miguel Costa. A data mining based system for credit-card fraud detection in e-tail. *Decision Support Systems*, 95:91–101, 2017. doi: 10.1016/j.dss.2017.01.002. URL https://www.sciencedirect.com/science/article/pii/S0167923617300027. Accessed: 2025-12-12.

[132] Vinicius Facco Rodrigues, Lucas Micol Policarpo, Díorgenes Eugênio da Silveira, Rodrigo da Rosa Righi, Cristiano André da Costa, Jorge Luis Victória Barbosa, Rodolfo Stoffel Antunes, Rodrigo Scorsatto, and Tanuj Arcot. Fraud detection and prevention in e-commerce: A systematic literature review. *Electronic Commerce Research and Applications*, 56:101207, 2022. doi: 10.1016/j.elerap.2022.101207. URL https://www.sciencedirect.com/science/article/pii/S1567422322000904. Accessed: 2025-12-11.

[133] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Shlomi Regev, Rocky Rhodes, Tiezhen Wang, and Pete Warden. Tensorflow lite micro: Embedded machine learning on tinyml systems. https://arxiv.org/abs/2010.08678, 2020. URL https://arxiv.org/pdf/2010.08678. arXiv:2010.08678; Accessed: 2025-12-15.

[134] Pete Warden and Daniel Situnayake. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, 2019. ISBN 9781492052043. URL https://www.oreilly.com/library/view/tinyml/9781492052043/. Accessed: 2025-12-15.

[135] Ewout ter Hoeven, Jan Kwakkel, Vincent Hess, Thomas Pike, Boyu Wang, Rht, and Jackie Kazil. Mesa 3: Agent-based modeling with python in 2025. *Journal of Open Source Software*, 10(107):7668, 2025. doi: 10.21105/joss.07668. URL https://doi.org/10.21105/joss.07668. Accessed: 2025-12-16.

[136] Olivier Atangana. Offsim-cbdc and fraud detection ml. https://www.kaggle.com/code/olivieratangana/offsim-cbdc-and-fraud-detection-ml, 2025. Kaggle Notebook; Accessed: 2025-12-15.

[137] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pages 3149–3157, 2017. URL https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf. NeurIPS 2017, Long Beach, CA, USA.

[138] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, USA, 1984. ISBN 9780412048418.

[139] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. doi: 10.1023/A:1010933404324.

[140] Joseph Berkson. Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, 39(227):357–365, 1944. doi: 10.1080/01621459.1944.10501110.

[141] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. doi: 10.1007/BF00994018.

[142] Mimi Mukherjee and Matloob Khushi. Smote-enc: A novel smote-based method to generate synthetic data for nominal and continuous features. *Applied System Innovation*, 4(1):18, 2021. doi: 10.3390/asi4010018. URL https://www.mdpi.com/2571-5577/4/1/18. Accessed: 2025-12-15.

[143] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785. Accessed: 2025-12-15.

[144] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. doi: 10.1109/5.18626. URL https://doi.org/10.1109/5.18626. Accessed: 2025-12-15.

[145] C. V. Sai, D. Das, N. Elmitwally, O. Elezaj, and M. B. Islam. Explainable ai-driven financial transaction fraud detection using machine learning and deep neural networks. *SSRN Electronic Journal*, 2023. doi: 10.2139/ssrn.4439980. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4439980. Preprint; Accessed: 2025-12-15.

[146] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint*, 2017. doi: 10.48550/arXiv.1702.08608. URL https://arxiv.org/abs/1702.08608. Accessed: 2025-12-15.

[147] Shanshan Zhu, Haotian Wu, Eric W. T. Ngai, Jifan Ren, Daojing He, Tengyun Ma, and Yubin Li. A financial fraud prediction framework based on stacking ensemble learning. *Systems*, 12(12):588, 2024. doi: 10.3390/systems12120588. Accessed: 2025-12-15.

[148] N. Damanik and C-M. Liu. Advanced fraud detection: Leveraging k-smoteenn and stacking ensemble to tackle data imbalance and extract insights. *IEEE Access*, 13: 10356–10370, 2025. doi: 10.1109/ACCESS.2025.3528079. Accessed: 2025-12-15.

[149] K. C. Veigas, D. S. Regulagadda, and S. A. Kokatnoor. Optimized stacking ensemble (ose) for credit card fraud detection using synthetic minority oversampling model. *Indian Journal of Science and Technology*, 14(32):2607–2615, 2021. doi: 10.17485/IJST/v14i32.807. Accessed: 2025-12-15.

[150] A. Md, R. Hossen, M. A. Uddin, M. N. Uddin, and U. K. Acharjee. Securing transactions: A hybrid dependable ensemble machine learning model using iht-lr and grid search. *Cybersecurity*, 7(1):32, 2024. doi: 10.1186/s42400-024-00221-z. Accessed: 2025-12-15.

[151] M. Webb. 60 global credit card fraud statistics in 2025. https://www.techopedia.com/credit-card-fraud-statistics, 2025. Accessed: 2025-04-23.

[152] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html. Accessed: 2025-04-23.

[153] Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv Preprint*, 2018. doi: 10.48550/arXiv.1802.03888. URL https://arxiv.org/abs/1802.03888.

[154] Wanli Yang, Yong Zhang, Keqin Ye, Linfeng Li, and Cheng-Zhong Xu. Ffd: A federated learning based method for credit card fraud detection. In Kan Chen, Sudhir Seshadri, and Liang-Jie Zhang, editors, *Big Data – BigData 2019*, volume 11514 of *Lecture Notes in Computer Science*, pages 18–32. Springer, Cham, 2019. doi: 10.1007/978-3-030-23551-2_2.

[155] V. Jain, A. Balakrishnan, D. Beeram, M. Najana, and P. Chintale. Leveraging artificial intelligence for enhancing regulatory compliance in the financial sector. *International Journal of Computer Trends and Technology*, 72(5):124–140, May 2024. doi: 10.14445/22312803/IJCTT-V72I5P116.

[156] Z. Zhang, H. A. Hamadi, E. Damiani, C. Y. Yeun, and F. Taher. Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access*, 10:93104–93139, 2022. doi: 10.1109/ACCESS.2022.3204051. URL https://doi.org/10.1109/ACCESS.2022.3204051. Accessed: 2025-12-15.

[157] T. Martins, A. M. De Almeida, E. Cardoso, and L. Nunes. Explainable artificial intelligence (xai): A systematic literature review on taxonomies and applications in finance. *IEEE Access*, 12:618–629, 2024. doi: 10.1109/ACCESS.2023.3347028. URL https://doi.org/10.1109/ACCESS.2023.3347028. Accessed: 2025-12-15.

[158] O. Atangana, M. Barbier, L. Khoukhi, and A. Kokcam. Systematization of knowledge on offline-capable digital-currency protocols. In *Proceedings of the 30th International Conference on Financial Cryptography and Data Security (ICFCDS)*, Tokyo, Japan, April .

Appendix A

List of Publications

P1. **O. Atangana**, M. Barbier, L. Khoukhi, and A. Kokcam, *Block-PAD: A blockchain-enabled framework for resilient and flexible CBDC transactions leveraging digital identity.* Computer Networks, 2026, 275, https://doi.org/10.1016/j.comnet.2025.111805.

P2. **O. Atangana**, M. Barbier, L. Khoukhi, and A. Kokcam. Systematization of knowledge on offline-capable digital-currency protocols. In Proceedings of the 20th International Conference on Financial Cryptography and Data Security (ICFCDS),Tokyo, Japan, Avril 2026.

P3. **Olivier Atangana**, Lyes Khoukhi, Morgan Barbier, and Joël Di Manno. *Securing offline cbdc transactions: Digivault card and markopaychain with mobile phoneintegration.* In Proceedings of the 2025 28th Conference on Innovation in Clouds,Internet and Networks (ICIN), pages 33–40, 2025. doi: 10.1109/ICIN64016.2025.10942801. URL https://doi.org/10.1109/ICIN64016.2025.10942801. Accessed:2025-12-11.

P4. **O. Atangana**, M. Barbier, L. Khoukhi, W. Royer, *Securing Privacy in Offline Payment for Retail Central Bank Digital Currency: A Comprehensive Framework*, Blockchain and Cryptocurrency Conference (B2C), 2023.