Applying UTXO-Based Privacy-Preserving Technologies to An Account-Based CBDC Architecture

To-Min Ok, Jun-Seok Shin, Jun-Sung Kim, Dae-Seon Choi*

Department of Software, Soongsil University, South Korea
tommyjade@ssu.ac.kr, james9907@soongsil.ac.kr, jikukjs@soongsil.ac.kr, sunchoi@ssu.ac.kr

Abstract

As central banks worldwide explore the development of Central Bank Digital Currencies (CBDCs), a critical design challenge lies in reconciling user privacy with regulatory oversight [1-2]. Traditional account-based systems inherently expose transaction details, while fully anonymous approaches may conflict with anti-money laundering (AML) and counter-terrorism financing (CTF) requirements. To address this challenge, we propose a privacy-preserving CBDC transaction framework that combines a UTXO-based model with Ring Confidential Transactions (RingCT), stealth addresses, and elliptic curve cryptography. The system is implemented on Hyperledger Besu, an enterprise grade Ethereum compatible platform designed for permissioned blockchain environments. We evaluate the proposed framework through intra-bank and inter-bank transaction scenarios, analyzing throughput, latency, and resource utilization. Our results show that while privacy-enhancing mechanisms introduce cryptographic overhead, the system maintains stable performance and acceptable latency. Furthermore, we introduce a dual-scenario privacy model that enables selective transparency, allowing authorized entities—such as auditors—to access encrypted transaction metadata without compromising individual privacy. This research demonstrates the practical viability of embedding strong privacy guarantees within CBDC architectures while preserving auditability and operational efficiency. The proposed model provides a scalable and adaptable foundation for future digital currency infrastructures.

Keywords: Central Bank Digital Currency (CBDC), Unspent Transaction Output (UTXO), Ring signature, Stealth address, Ring Confidential Transaction (RingCT)

1 Introduction

The global financial landscape is undergoing a profound transformation with the rapid emergence of Central Bank Digital Currencies (CBDCs) [3]. As central banks around the world pursue the digitalization of sovereign money, a key design challenge has surfaced: how to ensure transaction privacy while maintaining regulatory oversight. In traditional fiat systems, a certain degree of privacy is preserved through regulated intermediaries. However, digital currencies—especially those based on distributed ledger technologies—tend to expose sensitive transactional metadata unless enhanced with appropriate privacy-preserving mechanisms.

CBDC systems face a fundamental tension between two competing objectives: protecting user privacy and ensuring traceability for anti-money laundering (AML), counter-terrorism financing (CTF), and regulatory compliance. Fully transparent systems risk undermining individual confidentiality, while fully anonymous systems can enable illicit activities and reduce oversight capabilities. An effective CBDC design must therefore adopt selective transparency mechanisms, enabling authorized entities (such as auditors or regulators) to access sensitive data without compromising user-level confidentiality.

Several privacy-preserving techniques have been proposed in the context of public cryptocurrencies. Monero [4-5], for example, utilizes RingCT [6-7] and stealth addresses [8] to obscure the sender, recipient, and transaction amount. Zcash employs zero-knowledge proofs to achieve full transaction anonymity. While these methods offer strong privacy guarantees, they are designed for decentralized, public networks and often lack support for selective disclosure—a feature crucial in CBDC implementations. Moreover, their integration into enterprise-grade, permissioned blockchain environments is limited due to performance overhead and architectural complexity.

To bridge this gap, we propose a privacy-preserving CBDC transaction model that leverages the UTXO-based structure combined with RingCT, stealth addresses, and elliptic curve cryptography. The system is implemented on Hyperledger Besu [9], an enterprise-oriented Ethereum- compatible platform, enabling integration with permissioned networks. The architecture supports both intra-bank and inter-bank transfers, ensuring that sensitive transaction details remain confidential from unauthorized entities while remaining auditable by designated authorities.

The key contributions of this work are as follows:

- We design and implement a modular CBDC transaction framework that integrates privacy-enhancing cryptographic techniques into a permissioned blockchain architecture.
- We propose two realistic privacy scenarios with entity-specific visibility models to support both

*Corresponding Author: To-Min Ok; Email: tommyjade@ssu.ac.kr
DOI: https://doi.org/10.70003/160792642025092605001

user privacy and regulatory needs.
- We conduct extensive performance evaluations using a JMeter-based simulation environment, measuring latency, throughput, and resource usage under realistic transaction loads.

The remainder of this paper is organized as follows: Section 2 reviews related work on privacy-preserving mechanisms in digital currency systems. Section 3 details the proposed system architecture and cryptographic components. Section 4 outlines the implementation and use case scenarios. Section 5 presents performance evaluation results. Finally, Section 6 concludes the paper and discusses directions for future research.

2 Related Works

Numerous efforts have been made to enhance privacy in digital currency systems, ranging from permissionless cryptocurrencies to permissioned, institution-backed infrastructures. This section reviews key privacy-preserving technologies, particularly in the context of CBDCs, and highlights their capabilities and limitations.

2.1 Privacy Techniques in Cryptocurrencies

Privacy-focused cryptocurrencies such as Monero, Zcash, and Dash have pioneered advanced cryptographic methods to conceal transaction details. Monero employs RingCT, which utilize ring signatures and Pedersen commitments to obfuscate the sender, receiver, and transaction amount. Zcash implements zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) to offer full transaction anonymity, providing cryptographic proof without revealing underlying data.

While these technologies achieve high levels of privacy, they are tailored for fully decentralized, permissionless environments and often lack support for selective disclosure - a feature critical for CBDC applications where regulatory access is necessary. Moreover, their integration into enterprise-grade platforms is hindered by computational complexity, significant performance overhead, and incompatibility with permissioned blockchain architectures.

2.2 Privacy in CBDC Prototypes

Several central banks and research initiatives have explored privacy features within CBDC prototypes. For example, the European Central Bank's Digital Euro initiative and Project mBridge - led by the BIS Innovation Hub in collaboration with regional central banks - have emphasized the importance of balancing privacy with traceability in cross-border payment infrastructures.

Many of these systems adopt tiered or role-based privacy models, allowing users to remain pseudonymous to commercial banks while being identifiable to central authorities when necessary. However, such approaches typically rely on institutional-level controls rather than end-to-end cryptographic guarantees, raising concerns over centralized data exposure and potential misuse.

2.3 Privacy in Enterprise Blockchains

Enterprise blockchain platforms such as Hyperledger Fabric, Hyperledger Besu, and Corda have explored various privacy features, including private data collections, channels, and zero-knowledge proof integrations. While these mechanisms are effective for enforcing access control, they generally do not guarantee cryptographic anonymity at the transaction level.

Hyperledger Besu, in particular, supports Ethereum-compatible smart contracts and is increasingly used in financial infrastructure pilots. However, it lacks native support for privacy-enhancing features such as RingCT or stealth addresses, necessitating custom extensions to support transaction confidentiality and unlinkability.

2.4 Research Gap and Motivation

Despite the progress in privacy-preserving mechanisms, there remains a clear gap in designing a CBDC architecture that enables transaction-level privacy while preserving selective auditability within a permissioned setting. Existing solutions either focus exclusively on privacy (as in Monero or Zcash) or rely heavily on organizational control, without cryptographic enforcement of privacy policies.

This research aims to address this gap by:
- Integrating RingCT and stealth address mechanisms into a UTXO-based framework,
- Implementing the system on Hyperledger Besu to evaluate performance and compatibility,
- Proposing entity-level access models that support selective transparency for central banks, commercial banks, users, and auditors,
- Conducting empirical evaluations in both intra-bank and inter-bank transaction scenarios.

2.5 Transaction Confidentiality Technologies

Transaction confidentiality refers to mechanisms—both technical and institutional—that protect sensitive data such as sender identity, recipient, and transaction amount from unauthorized access [10-11]. In CBDCs, confidentiality is essential for preserving user privacy and maintaining trust in centrally issued digital currencies.

Traditional financial systems rely on centralized databases managed by banks and payment service providers. While internal protections exist, these systems remain susceptible to data breaches, surveillance, or misuse by authorities. Blockchain-based CBDCs, by contrast, operate on transparent infrastructures where confidentiality depends heavily on system design and access control [12-14].

Account-based CBDCs often report all transaction details to a central authority in real time, enabling large-scale data collection on user behavior. This raises concerns about financial surveillance and potential infringements on civil liberties—issues that extend beyond technology to the realm of governance.

To address these risks, recent research has focused on applying cryptographic techniques to preserve transaction confidentiality in CBDCs. The following sections examine key approaches under consideration.

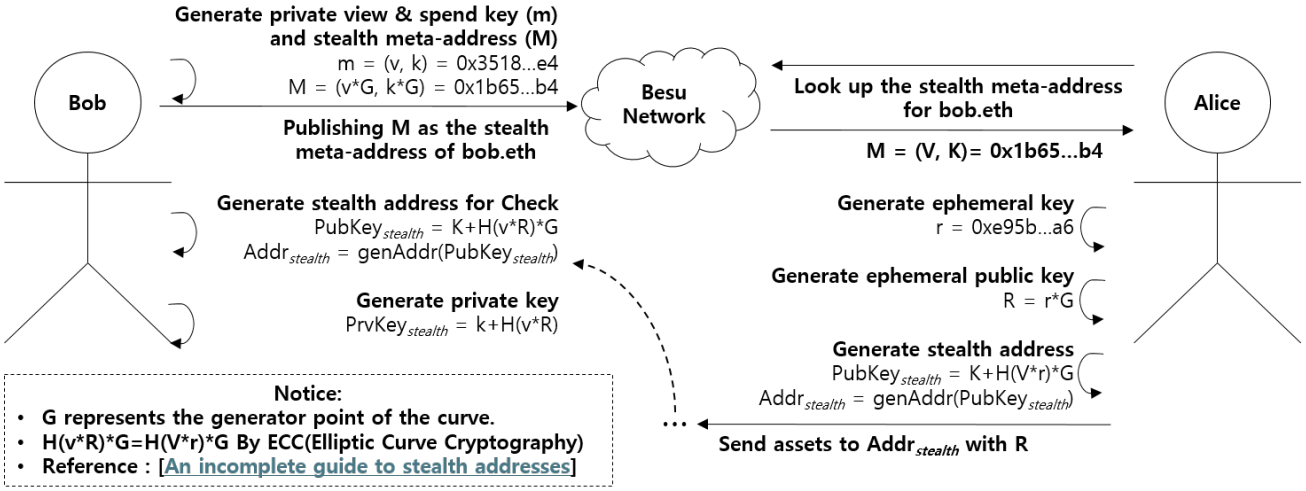

Figure 1. Stealth address system with elliptic curve cryptography [15-16]

2.5.1 Stealth Address

The Stealth Address is a cryptographic mechanism that generates a unique one-time address for each transaction based on the recipient's public key information. This design ensures that third parties are unable to link multiple transactions to the same recipient. Even if a single user receives multiple transactions, external observers cannot recognize them as belonging to the same entity. Only the recipient can identify and decrypt the transaction addressed to them.

As shown in [Figure 1], The procedure for generating and verifying a Stealth Address is as follows [8].

Stealth Meta Address Registration:

- Bob (the receiver) generates a Stealth Meta Address.
- Bob publishes the generated Stealth Meta Address to the Blockchain Network.

Stealth Address Generation from Stealth Meta Address:

- Alice (the sender) retrieves Bob's Stealth Meta Address from the Besu Network: M = (V, K)
- Alice generates a random ephemeral key (r) known only to herself and used only once.
- Alice computes a shared secret by multiplying her ephemeral key with Bob's View Public Key, and then hashes the result to obtain h(s).
- Bob's Stealth Public Key is computed as: Bob's Spend Public Key + h(s) * G.
- Bob's one-time Stealth Address is: pubkeyToAddress (Stealth Public Key)

Stealth Tx Creation:

- A transaction (Tx) is created and sent, containing the Stealth Address, a reference value for the amount, an ephemeral public key, and an encrypted message.

Check Stealth Address:

- The receiver multiplies the ephemeral public key by their View Private Key to generate a shared secret, which is then hashed to produce h(s).
- The Stealth Public Key is computed as: Spend Public Key + h(s) * G
- If pubkeyToAddress (Stealth Public Key) equals the Stealth Address stored in the Stealth Tx, the receiver can confirm that the Stealth Address belongs to them.

Monero applies this mechanism in practice, enabling all transaction outputs to appear unrelated on the blockchain—even when sent to the same recipient. This makes it impossible for third parties to track recipients, senders, or amounts, thereby significantly enhancing on-chain privacy [17].

This study explores the feasibility of adapting such Stealth Address techniques to CBDC environments, aiming to strike a balance between technological privacy protection and institutional auditability.

2.5.2 Ring Signature

Ring Signature [18-19] is a digital signature scheme designed to preserve the sender's anonymity by making the actual signer indistinguishable from other participants in the ring. This structure ensures that a transaction verifier cannot determine which public key in the ring was used to generate the signature—only that the signer is one of the participants in the ring.

A Ring Signature consists of the following elements:

- $\{P1, P2, \ldots, Pn\}$: A set of n public keys forming the ring.
- si: A signature component derived from the signer's private key.
- I: A key image, which acts as a cryptographic identifier used to prevent double-spending.
- m: The message to be signed (e.g., transaction data).

In the case of Monero, the Multilayered Linkable Spontaneous Anonymous Group (MLSAG) Signature algorithm is used to implement Ring Signatures. This approach allows multiple transaction inputs to be signed simultaneously while generating a separate key image for each input. The result is a compact signature structure suitable for practical blockchain applications.

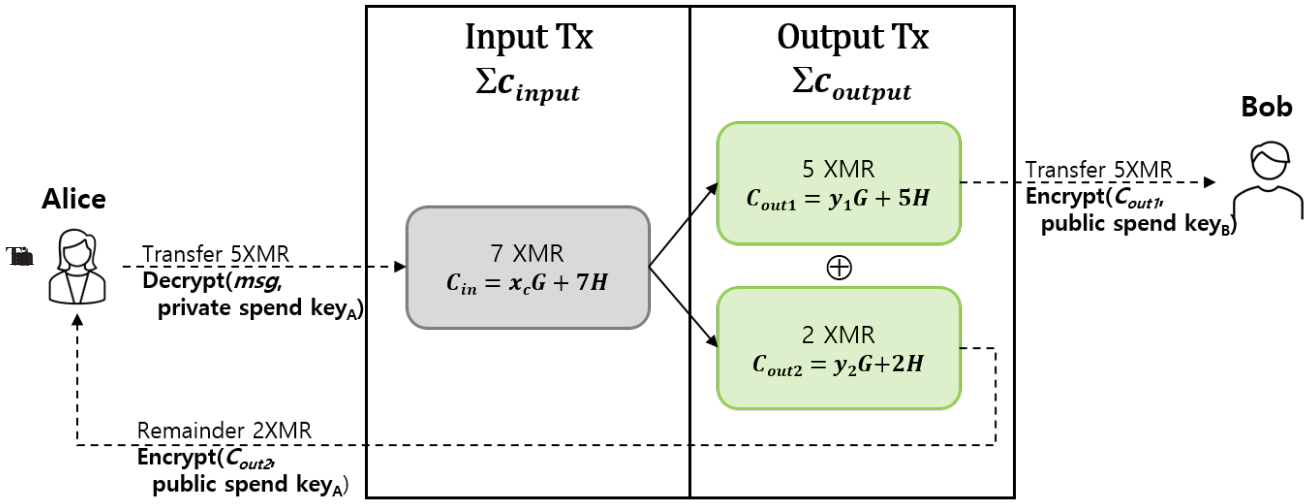

Figure 2. Concept of Ring Confidential Transaction

Signature Generation Procedure:

- The sender selects $n-1$ random public keys along with their own public key Ps, constructing a ring $\{P1, ..., Pn\}$.
- For each public key, the sender generates a random value ri and calculates a series of hash chains.
- Only at their actual position ss, the sender uses their private key to compute ss, which is then included in the signature.
- The final signature takes the form $(\{ci,ri\},I)$, which can be validated by cycling through the ring cryptographically, without revealing which Pi corresponds to the actual signer.

Key Image:

- The key image is computed as:

$$I = x \cdot Hp(P)$$

Where x is the sender's private key, P is the corresponding public key, and Hp is a hash function mapped to an elliptic curve point. If the same key image I is used more than once, the network can detect it and block double-spending. However, it remains computationally infeasible to recover the private key from I, thereby preserving privacy.

Anonymity Mechanism:

- The larger the ring size n, the stronger the anonymity of the sender.
- All public keys except the actual signer act as decoys, and the signature does not require knowledge of their private keys.
- While the signature can be verified for correctness, the identity of the signer remains cryptographically untraceable, ensuring both untraceability and unlinkability.

Monero applies such Ring Signature structures to all of its transactions, typically using 11 to 16 members per ring by default. As a result, it becomes virtually impossible to trace the sender—even using real-time blockchain analysis tools or forensic explorers.

2.5.3 Ring Confidential Transaction

RingCT [6-7, 20] encrypts the transaction amount within the blockchain while using mathematical proofs—specifically, Pedersen Commitments and Range Proofs—to verify the validity of the amounts. This advanced confidentiality technique ensures that while the amount is hidden, the integrity and correctness of the transaction are still maintained.

The Pedersen Commitment scheme used in RingCT is defined as follows:

$$C = r \cdot G + v \cdot H$$

where:

- v is the transaction amount,
- r is a randomly chosen blinding factor,
- G and H are independent elliptic curve generators.

This commitment C conceals the value v while still enabling validation of value consistency in transactions. Specifically, the network can verify that the sum of inputs equals the sum of outputs without revealing the actual amounts:

$$\sum Cin = \sum Cout$$

To prevent double-spending, RingCT employs a key image mechanism. The key image is derived from the sender's private key and uniquely represents the use of a specific output. If the same key image appears more than once, the network identifies it as a double-spend attempt.

The Ring Signature structure used in RingCT can be expressed as:

$$\sigma = \text{RingSign}(m, \{P1, ..., Pn\}, s)$$

where:

- m is the message being signed (e.g., transaction data),
- $\{Pi\}$ are the public keys included in the ring,
- s is the secret key of the actual signer.

Through this structure, RingCT ensures that the transaction amount is confidential, while still allowing verifiers to ensure that no tokens are created or lost, and that the transaction is mathematically valid. This approach, pioneered by Monero, enables untraceable and unlinkable transactions, which are essential for strong privacy guarantees on public blockchains.

As shown in [Figure 2] RingCT Process Overview,

Input Transaction: Ensuring Anonymity – Ring Signature:

- To send 5 XMR, Alice uses a 7 XMR UTXO to generate the transaction.
- Fact that she is the actual owner of this 7 XMR is concealed using a ring signature.
- Alice's input is combined with multiple other users' inputs to form a ring, making it impossible for external observers to identify which input belongs to the true sender.

Output Transaction: Hiding Amounts – RingCT:

- The output of the transaction consists of 5 XMR sent to Bob and 2 XMR returned to Alice as change.
- All output amounts are encrypted using Pedersen Commitments, ensuring that the actual amounts are not visible to outsiders.
- The validity of the total amount can be proven in a blinded manner, thereby preventing tampering.

Amount Balance Verification: Recipient Identification – Stealth Address:

- Bob scans the blockchain with his view key and detects that the transaction is addressed to him.
- He then uses his spend private key to decrypt and receive the 5 XMR.
- Alice also receives 2 XMR through a stealth address that was generated based on her own public key.

These technologies have proven to be effective tools for achieving privacy protection within fully transparent blockchain environments. When applied to CBDCs, they are regarded as foundational technologies that can help strike a balance between privacy preservation and regulatory compliance.

2.6 Comparative Analysis of Related Technologies and Previous Studies

Privacy-preserving mechanisms in CBDC systems typically follow two primary paradigms: UTXO-based and ZKP-based models [21-24]. UTXO-based designs, exemplified by Monero, use stealth addresses, ring signatures, and RingCT to conceal the sender, receiver, and transaction amount. These methods leverage the decentralized structure of UTXO transactions to provide strong on-chain privacy. However, they often lack institutional traceability, making them less suited for regulated environments governed by Anti-Money Laundering (AML) and Counter-Terrorism Financing (CTF) requirements.

ZKP-based approaches—such as zk-SNARKs [25-28], zk-STARKs [29-31], and Bulletproofs [32]—achieve privacy by mathematically proving transaction validity without revealing underlying data. While these offer excellent confidentiality and cryptographic assurance, they impose significant computational burdens, such as long proof generation times, large memory usage, and limited smart contract compatibility, which constrain real-time deployment in high-frequency CBDC systems.

Some initiatives, like the Digital Euro, have explored ZKP applications, though implementation specifics and performance data remain scarce. In contrast, China's e-CNY employs a tiered privacy model allowing partial anonymity while preserving auditability for large transactions [33]. To reconcile privacy with oversight, this paper proposes a viewing key–based conditional audit mechanism, offering a practical middle ground between full anonymity and unrestricted transparency.

To clarify the trade-offs between these approaches, [Table 1] summarizes key comparative dimensions:

In summary, the proposed architecture combines the technical strengths of UTXO-based models with institutional auditability, offering a scalable and policy-aligned solution for national CBDC systems. It avoids the steep integration cost of ZKPs while maintaining essential privacy controls.

Table 1. UTXO-based vs. ZKP-based privacy architectures

Feature	UTXO-Based	ZKP-Based
Privacy Mechanism	Obfuscation + encryption	Zero-knowledge proofs
Auditability	Via Viewing Keys	Requires custom proof circuits
Computational Cost	Moderate	High
Proof Size	Small–Moderate	Small (SNARK), Large (STARK)
Trusted Setup	No	SNARK: Yes, STARK: No
Quantum Resistance	No	STARK: Yes
Smart Contract Support	High (Ethereum-compatible)	Limited
Real-time Suitability	High for most scenarios	Challenging for high-throughput
Implementation Complexity	Moderate	High
Deployment Maturity	Proven (e.g., Monero, Besu)	Emerging

3 Proposed Architecture and Confidentiality Application Model

3.1 System Architecture and Flow Diagram

This study presents a hybrid architecture that integrates UTXO-based privacy-preserving technologies into a CBDC system built on an account-based infrastructure [Figure 3]. The core objective of this architecture is to simultaneously uphold user privacy and regulatory auditability by employing key cryptographic primitives—including Stealth Addresses, Ring Signatures, and RingCT—to ensure transaction confidentiality.

The proposed system comprises four major components:

3.1.1 Privacy System (API Server Layer)

The Privacy System of the API server processes transaction requests from financial institutions and central authorities, and is responsible for performing cryptographic

computations and key management throughout the transaction generation and validation phases. It interfaces with institutional keystores to manage Viewing Keys and Spending Keys, which facilitate the generation of receiver addresses, sender signing, and encryption of transaction amounts.

The Spending Key Comprises:

- The sender's Spend Private Key, used to reference the sender's UTXO in the input transaction (Input Tx), and the Spend Public Key, used to generate the receiver's UTXO in the output transaction (Output Tx).

The Viewing Key:

- Uses the View Public Key to generate a one-time Stealth Address for the recipient during transaction creation.
- Enables the recipient to detect incoming transactions addressed to them by scanning the ledger.
- Can be optionally shared with authorized third parties (e.g., regulators) to enable conditional visibility of transaction metadata without granting control over or access to the underlying UTXO.

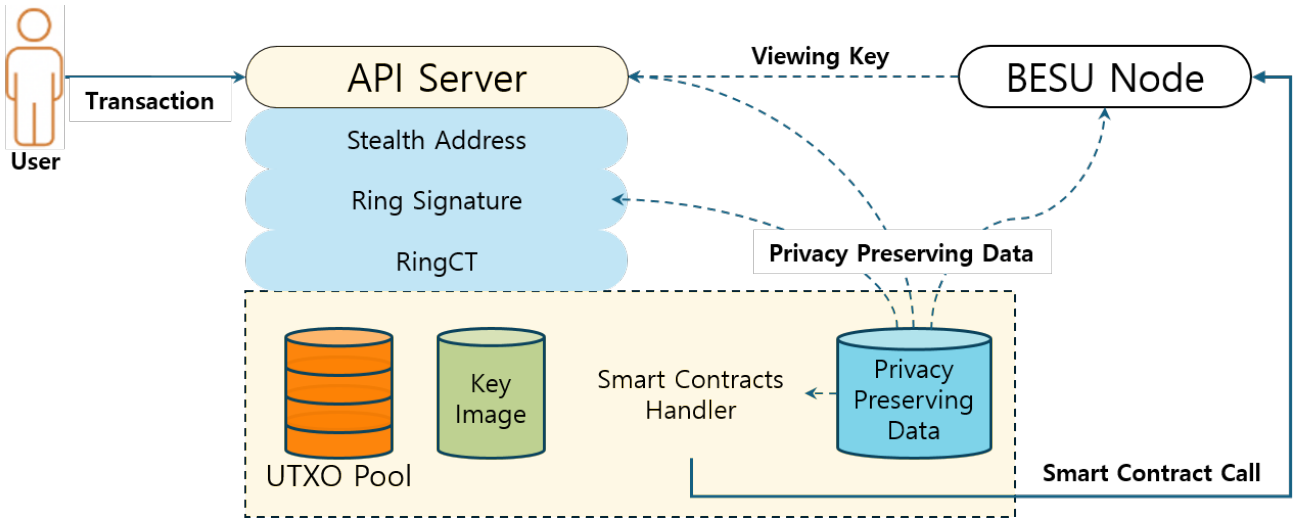

Figure 3. UTXO-based privacy technology application structure

3.1.2 UTXO Pool

The UTXO Pool is implemented as a memory cache-based structure that categorizes the state of each transaction into one of three statuses: Pending, Unspent, or Spent (see Figure 4). This structure is optimized for parallel processing and designed to prevent double spending. It is also used in the UTXO sampling process during RingCT construction to select decoy inputs (mix-ins), thereby enhancing input anonymity.

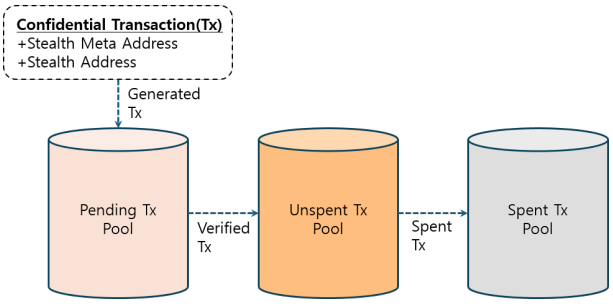

Figure 4. UTXO pool configuration and flow diagram

3.1.3 Smart Contracts (Privacy Enforcement Layer)

The privacy features of the system are enforced through a set of smart contracts deployed on Hyperledger Besu, enabling operations such as Stealth Address generation, Ring Signature validation, and transaction amount obfuscation. These contracts are organized into modular components: [34-35]

DCR Privacy Smart Contract:

Acts as the central contract for verifying confidential transactions across all institutions. It manages transaction records and supports deposit, withdrawal, and transfer operations involving Privacy Tokens. This contract ensures confidentiality of sender, receiver, and transaction amount during intra-institutional transactions.

Institution-Specific DCR Privacy Contracts:

Deployed independently by each participating institution, these contracts validate user-initiated confidential transactions, manage private balances, and track transaction status. They ensure internal privacy while maintaining verifiability of proper transaction execution. They also support auditability by authorized parties.

Exchange Transfer DCR Smart Contract:

Facilitates inter-bank transfers while maintaining confidentiality across institutions. It supports secure cross-ledger operations through Privacy Tokens, ensuring that sensitive metadata is not exposed during inter-institutional exchanges.

RingSigLibs:

A dedicated privacy-preserving cryptographic library supporting:

- Stealth Address verification: Ensures the one-time recipient address is valid.

- Ring Signature validation: Guarantees sender anonymity while validating transaction authenticity.
- RingCT verification: Confirms that the encrypted transaction amount has not been altered and lies within a valid range using Pedersen Commitments and Range Proofs.

This library is invoked directly by smart contracts, ensuring that transaction privacy and verifiability are enforced at the protocol level.

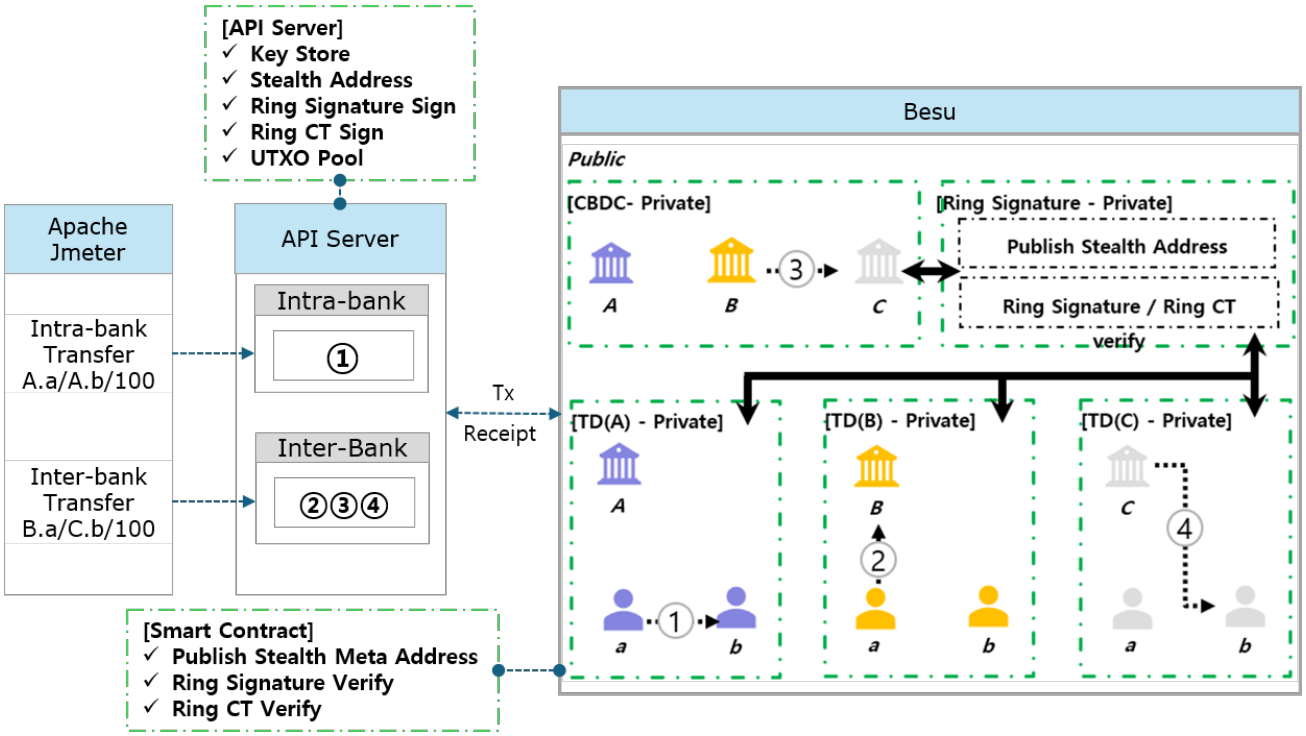

Figure 5. Intra-bank/Inter-bank transfer scenario

3.1.4 Blockchain Network (Hyperledger Besu Layer)

The underlying blockchain network is a private, Ethereum-compatible Hyperledger Besu consortium, comprising Validator Nodes and RPC Nodes operated by participating institutions. All transactions are validated and recorded on-chain using privacy-preserving mechanisms, with access to sensitive transaction metadata strictly controlled by the possession of the corresponding Viewing Key.

The overall system flow proceeds as follows:

- The sender generates a Stealth Address based on the receiver's public key and constructs the transaction.
- A Ring Signature is created to conceal the identity of the sender.
- The transaction amount is encrypted using RingCT, specifically through Pedersen Commitments and Range Proofs [36-37].
- The constructed transaction is broadcast to the blockchain network via the API server.
- On the blockchain, the transaction is verified and recorded by smart contracts.
- Regulatory or auditing institutions holding the Viewing Key can access the transaction content conditionally to ensure oversight without compromising full anonymity.

The proposed structure can be evaluated as an integrated architecture that maintains compatibility with existing account-based CBDC systems while incorporating the advantages of UTXO-based privacy-preserving technologies, thereby addressing both technical feasibility and policy acceptability.

3.2 Application of Privacy-preserving Technologies

This study applies key UTXO-based privacy-preserving technologies to ensure the confidentiality of CBDC transactions. The main technologies implemented are Stealth Address, Ring Signature, and RingCT — each performing a critical function to guarantee the confidentiality of the sender, receiver, and transaction amount, respectively.

First, As mentioned in [2.2 Overview of Transaction Privacy Technologies], Stealth Address is a privacy-preserving technique designed to make it impossible to trace the recipient's address on the blockchain. To apply the Stealth Address technology to the Ethereum-based Besu network, the API server was developed to implement logic for generating Stealth Meta Addresses and creating Stealth Address-based transactions. Additionally, a smart contract was implemented on the Besu network to verify Stealth Addresses and validate transactions based on Stealth Addresses.

Second, Ring Signature is a signature technique that conceals the identity of the sender while ensuring the validity of the transaction. The sender constructs a ring, or a set of public keys, by combining their own public

key with several randomly selected public keys. A valid signature is then generated within this set. External verifiers cannot determine which member of the ring created the signature, but they can confirm that it was generated by one of the valid keys within the ring.

This method protects the sender's identity while ensuring transaction integrity, and it also performs double-spending prevention through the use of a Key Image.

In this study, an extended version of the Ring Signature technique called MLSAG (Multilayer Linkable Spontaneous Anonymous Group) is applied. MLSAG enhances the security of standard Ring Signatures and supports signing for multiple input UTXOs. By improving the original Ring Signature structure, MLSAG prevents recipient tracing, further enhances the privacy of the sender, and ensures that UTXO amounts cannot be altered. Moreover, it guarantees the validity of the sender's signature and the transaction even when handling multiple UTXOs simultaneously.

Third, Ring Confidential Transaction is a technique that encrypts transaction amounts on the blockchain so that third parties cannot view them. RingCT uses the Pedersen Commitment method to encrypt the amount, and employs a Range Proof to verify that the amount falls within a valid range. This approach enables advanced privacy protection by ensuring transaction validity without revealing the actual value of the amount.

In this study, RingCT is combined with Ring Signature to achieve complete obfuscation of both the sender and the transaction amount.

These techniques have been used in existing blockchain-based cryptocurrencies. This study integrates them into an account-based CBDC architecture, implementing a system that satisfies both privacy protection and transaction verification integrity. Each technical component is executed in an automated manner via smart contracts, and is designed to work in conjunction with a Viewing Key-based conditional audit mechanism, enabling regulated information access for supervisory authorities.

4 Experimental Environment and Performance Evaluation

This chapter describes the experimental environment and results designed to verify the performance impact of the proposed UTXO-based privacy-preserving technologies on an account-based CBDC system. [Figure 5] The experiments were implemented on a Hyperledger Besu-based blockchain network and focused on two primary scenarios: intra-bank transfers and inter-bank transfers.

Performance evaluation was conducted based on key metrics such as transactions per second (TPS), latency, and gas costs. The comparison highlights the differences in system performance before and after the application of privacy-preserving technologies.

4.1 Experimental Setup and Repetition Conditions

In this study, an experimental environment was constructed using a Hyperledger Besu-based blockchain network to analyze the feasibility and performance impact of the proposed UTXO-based privacy-preserving technologies. The experiments were divided into intra-bank and inter-bank transfer scenarios, depending on whether privacy technologies were applied. Performance metrics included Transactions Per Second (TPS), average latency, and smart contract gas costs.

The experimental environment was configured as follows:

Each node was operated in a virtualized cloud environment, and the API server functioned as a core computation module responsible for Stealth Address generation, Ring Signature processing, and Ring Confidential Transaction handling. The blockchain network was built as a private blockchain based on Hyperledger Besu, consisting of six nodes (four validators and two RPC nodes) responsible for consensus. Transaction load generation was implemented using Apache JMeter, which automatically produced repeated requests for each test scenario.

Each experimental scenario was conducted under the following conditions:

- Intra-bank transfer: A fund transfer scenario between users within the same financial institution, characterized by relatively short transaction paths and simple verification processes.
- Inter-bank transfer: A fund transfer scenario between users of different financial institutions, involving more complex API server and smart contract processing flows, with conditional path branching based on Viewing Key access permissions.

Each scenario was repeated at least 10 times, and the average and standard deviation of the measured results were calculated to ensure the reliability of the data. Performance differences were quantitatively analyzed by comparing results with and without the application of privacy-preserving technologies.

Additionally, CPU usage, processing time, and memory consumption of the API server and smart contracts were separately monitored to analyze causes of load distribution and performance bottlenecks. This data was used to identify performance degradation factors and serve as a foundation for suggesting technical improvements.

This experimental design provides foundational data to quantitatively verify the technical feasibility of implementing such technologies in a real-world CBDC operational environment, and it can serve as a practical guideline for policy development and system optimization.

4.2 Experimental Scenarios (Intra-bank/Inter-bank Transfers)

To reflect real-world operational conditions of CBDC, this study's experiments were designed based on two representative transaction scenarios: Intra-bank Transfer

and Inter-bank Transfer. Each scenario differs in terms of transaction path, inter-institutional interaction, and the required level of privacy protection. Through these variations, the study aims to analyze the impact of privacy-preserving technologies on system performance across different operational conditions from multiple perspectives.

4.2.1 Intra-bank Transfer

The intra-bank transfer scenario involves the process of transferring digital currency between users within the same financial institution and features a relatively simple structure. Transfer requests are handled through the institution's API server, and transaction verification and state management are performed using a single UTXO Pool and smart contract.

This scenario has the following characteristics:

- The generation and verification of the recipient's Stealth Address are completed within the same institution.
- The Viewing Key is used exclusively for internal auditing, with no interaction with external institutions.
- Network-level communication latency is minimized, making this scenario suitable for analyzing performance changes caused specifically by cryptographic computations within the system.

The intra-bank transfer serves as a baseline for measuring the performance of the basic transaction processing flow, enabling a direct evaluation of the impact of increased computational complexity due to the application of privacy-preserving technologies on overall processing latency.

A total of 6 scenarios are defined for intra-bank transfers, as summarized in [Table 2].

Table 2. Use cases of scenario A

Category	Use Case	Entity	Check Item Description
Scenario A	A-1	Central Bank	Unable to identify all items
	A-2	Bank 1	Able to identify all items
	A-3	User A	Able to identify all items
	A-4	User B	Able to identify all items
	A-5	Bank2	Unable to identify all items
	A-6	Auditor	Able to identify all items

4.2.2 Inter-bank Transfer

The inter-bank transfer scenario involves the transfer of funds between users belonging to different financial institutions. It is a high-complexity scenario, where the communication paths between system components become more intricate, and the process flow diverges depending on whether Viewing Key linkage with regulatory authorities is required.

The main components of this scenario are as follows:

- The sender's and receiver's API servers are located at different institutions, and each institution has its own separate key management system.
- The Stealth Address and Ring Signature are generated at the sending institution, while verification by the receiver is performed independently at the receiving institution.
- Requests and verification procedures for Viewing Key access are executed in parallel through conditional logic within blockchain smart contracts.

The inter-bank transfer reflects a scenario likely to occur frequently in real operational environments and is well-suited for measuring the impact of inter-institutional compatibility and distributed transaction processing structures on system performance.

In particular, when privacy-preserving technologies are applied, key variables that influence performance include:

- Increased smart contract processing time,
- Verification delays due to Viewing Key access requests, and
- Complexities in synchronizing UTXO Pool states.

Table 3. Use cases of scenario B

Category	Use Case	Entity	Check Item Description
Scenario B	B-1	Central Bank	Sender & Receiver not identifiable
	B-2	Bank 1	Receiver not Identifiable
	B-3	Bank 2	Sender not Identifiable
	B-4	User A	Able to identify all items
	B-5	User B	Able to identify all items
	B-6	Bank 3	Unable to identify all items
	B-7	Auditor	Able to identify all items

A total of 7 scenarios are defined for inter-bank transfers, as summarized in [Table 3].

By ensuring experimental diversity in both conditions and transaction types, the two scenarios serve as meaningful comparative data for evaluating the effectiveness of privacy-preserving technologies and the overall scalability of the system.

4.3 Experimental Results and Analysis

This section presents a quantitative analysis of the impact that applying UTXO-based privacy-preserving technologies has on the performance of a CBDC system, based on the performance measurement results obtained from the previously described experimental setup and scenarios.

The experiments were conducted before and after the application of privacy technologies, comparing the following metrics for both intra-bank and inter-bank transfers:

- Transaction Processing Speed (TPS)
- Average Latency
- Smart Contract Gas Costs

4.3.1 Functional Analysis of Cryptographic Technologies in the Intra-bank Transfer Scenario

This study sets up a scenario of intra-bank transfer for CBDC, and conducts a functional analysis of four privacy-preserving cryptographic technologies — Homomorphic Encryption, zk-SNARK, zk-STARK, and Ring Signature — with a focus on the level of transaction information shared among participants and the concealment of information from non-participants. As summarized in

[Table 4], experiments were conducted to verify the level of confidentiality and the scope of information access for each technology.

Table 4. Overview of functional capabilities in intra-bank transfer scenarios

Category	Scenario	Entity	Requirement/Result		
			Sender	Receiver	Amount
Homomorphic Encryption	A-1	Central Bank	Hidden	Hidden	Hidden
	A-2	Bank 1			
	A-3	User A			
	A-4	User B			
	A-5	Bank2	Hidden	Hidden	Hidden
	A-6	Auditor			
zk-SNARK	A-1	Central Bank	Hidden	Hidden	Hidden
	A-2	Bank 1			
	A-3	User A			
	A-4	User B			
	A-5	Bank2	Hidden	Hidden	Hidden
	A-6	Auditor			
zk-STARK	A-1	Central Bank	Hidden	Hidden	Hidden
	A-2	Bank 1			
	A-3	User A			
	A-4	User B			
	A-5	Bank2	Hidden	Hidden	Hidden
	A-6	Auditor			
Ring Signature	A-1	Central Bank	Hidden	Hidden	Hidden
	A-2	Bank 1			
	A-3	User A			
	A-4	User B			
	A-5	Bank2	Hidden	Hidden	Hidden
	A-6	Auditor			

EC-ElGamal-based Homomorphic Encryption:

In the intra-bank transfer scenario applying homomorphic encryption, the following results were observed:

- Bank 1, User A (sender), and User B (receiver) were all able to access the information on the sender, receiver, and transferred amount.
- The auditor also had full access to all transaction information, satisfying the essential requirements for audit functionality.
- External entities such as the central bank and Bank 2 (participating institution 2) could not access information on the receiver or the transferred amount, as these were successfully concealed.
- However, the sender's information was not hidden and was identifiable by external entities.

This is due to the structural characteristic of homomorphic encryption, where the sender's public key must be used for withdrawal and smart contract execution. To hide the sender's identity, an additional anonymity technology such as Ring Signature needs to be combined. Thus, in the environment where homomorphic encryption was applied alone, sender privacy was deemed incomplete. In other words, while the receiver and amount were successfully hidden, the sender was not.

zk-SNARK-based Zero-Knowledge Proof:

In the intra-bank transfer experiment using zk-SNARK, the following outcomes were observed:

- Bank 1, User A, User B, and the auditor had access to all transaction information (sender, receiver, and amount).
- In contrast, the central bank and Bank 2 were unable to access any of these data fields, indicating complete confidentiality protection.

zk-SNARK has the advantage of concise proof generation and fast verification time, although it requires a trusted setup. However, in this experiment, a modified version was used to overcome this limitation, and EC-ElGamal-based homomorphic encryption (ECHC) was integrated for amount concealment, achieving privacy protection suitable for the scenario.

zk-STARK-based Zero-Knowledge Proof:

Applying a modified version of zk-STARK in the intra-bank transfer scenario yielded the same level of privacy protection as zk-SNARK:

- Bank 1, User A, User B, and the auditor were able to access all transaction-related information.
- The central bank and Bank 2 were denied access to sender, receiver, and amount data, thus successfully ensuring confidentiality.

zk-STARK offers the advantage of not requiring a trusted setup and is resistant to quantum attacks. In this experiment, an optimized version was used to improve proof size and verification speed.

Ring Signature-based Anonymity Technology:

The experiment results from applying Ring Signature in the intra-bank transfer scenario are as follows:

- All participating entities (Bank 1, User A, User B, and the auditor) were able to access the sender, receiver, and transferred amount, satisfying the functional requirements from both transaction parties and the audit perspective.
- External entities, namely the central bank and the second participating institution, were denied access to all transaction information (sender, receiver, and amount), thereby achieving confidentiality protection.

4.3.2 Functional Analysis of Cryptographic Technologies in the Inter-bank Transfer Scenario

In this study, an inter-bank transfer scenario (remittance across different institutions) for CBDC is established. The focus is on analyzing the level of transaction information sharing among participants and the concealment of information from non-participants using four privacy-preserving cryptographic technologies — Homomorphic Encryption, zk-SNARK, zk-STARK, and Ring Signature. As summarized in [Table 5], the experiments verify the confidentiality assurance level and the scope of information access for each technology.

EC-ElGamal-based Homomorphic Encryption:

In the inter-bank transfer scenario utilizing homomorphic encryption, the following results were observed:

- Bank 1, User A, User B, and the auditor were all able to access information on the sender, receiver, and transferred amount.
- The central bank (B-1), Bank 2 (B-3), and Bank 3 (B-6) were blocked from accessing the receiver and the amount; however, concealment of the sender information failed.

Table 5. Overview of functional capabilities in inter-bank transfer scenarios

Category	Scenario	Entity	Requirement/Result		
			Sender	Receiver	Amount
Homomorphic Encryption	B-1	Central Bank	~~Hidden~~	Hidden	
	B-2	Bank 1		Hidden	
	B-3	Bank 2	~~Hidden~~		
	B-4	User A			
	B-5	User B			
	B-6	Bank 3	~~Hidden~~	Hidden	Hidden
	B-7	Auditor			
zk-SNARK	B-1	Central Bank	Hidden	Hidden	
	B-2	Bank 1		Hidden	
	B-3	Bank 2	Hidden		
	B-4	User A			
	B-5	User B			
	B-6	Bank 3	Hidden	Hidden	Hidden
	B-7	Auditor			
zk-STARK	B-1	Central Bank	Hidden	Hidden	
	B-2	Bank 1		Hidden	
	B-3	Bank 2	Hidden		
	B-4	User A			
	B-5	User B			
	B-6	Bank 3	Hidden	Hidden	Hidden
	B-7	Auditor			
Ring Signature	B-1	Central Bank	Hidden	Hidden	
	B-2	Bank 1		Hidden	
	B-3	Bank 2	Hidden		
	B-4	User A			
	B-5	User B			
	B-6	Bank 3	Hidden	Hidden	Hidden
	B-7	Auditor			

zk-SNARK-based Zero-Knowledge Proof:

The experiment using zk-SNARK in the inter-bank transfer scenario yielded the following results:

- Bank 1, User A, User B, and the auditor were able to access all transaction information.
- The central bank, Bank 2, and Bank 3 were unable to access any of the data fields (sender, receiver, or amount), thus achieving complete confidentiality protection.
- The modified version of zk-SNARK was integrated with EC-ElGamal-based homomorphic encryption (ECHC), strengthening the linkage between value computations and proof, while its lightweight structure also improved transaction verification performance.

zk-STARK-based Zero-Knowledge Proof:

Applying a modified version of zk-STARK in the inter-bank transfer scenario produced the same level of privacy protection as zk-SNARK:

- Participants (Bank 1, User A, User B, and the auditor) were able to access all transaction information.
- The central bank, Bank 2, and Bank 3 were denied access to all information, which was successfully concealed.
- zk-STARK, a quantum-resistant technology that does not require a trusted setup, demonstrated reduced proof size and improved verification speed in the modified version, achieving a level of performance suitable for real-time transactions.

Ring Signature-based Anonymity Technology:

The experiment results from applying Ring Signature in the inter-bank transfer scenario are as follows:

- Bank 1, User A, User B, and the auditor were able to view all transaction information.
- The central bank, Bank 2, and Bank 3 were unable to access sender, receiver, or amount information.
- LSAG (Multilayered Linkable Spontaneous Anonymous Group Signature) ensured secure signing for multiple inputs and prevented double spending. Ring Confidential Transactions (Ring CT) used Pedersen Commitments combined with Range Proofs for amount concealment, thereby guaranteeing both confidentiality and integrity.

4.3.3 Performance Without Privacy-Preserving Technologies

In the performance evaluation conducted without the application of privacy-preserving technologies, both the intra-bank and inter-bank transfer scenarios demonstrated relatively high processing performance [Table 6], [Figure 6].

Table 6. General interval performance (for intra-800, inter-1,000 transfer)

	Scenario	Interval	From	To	In	Out	TPS	Latency(ms)
①	Intra-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*335.1*	*2250*
②		Send Tx	API Server	Besu	Sender, Receiver, Amount	Tx Hash	*N/A*	*2212*
③		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	
④	Inter-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*214.4*	*4636*
⑤		Send Tx	API Server	Besu	Sender, Receiver, Amount	Tx Hash	*N/A*	*4583*
⑥		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	

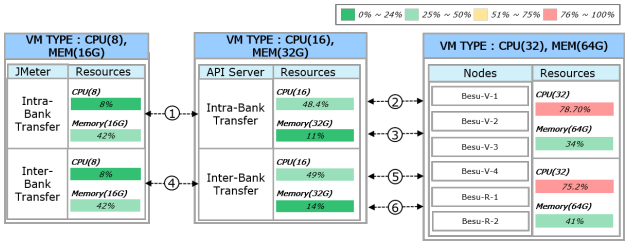

Figure 6. General (Confidentiality not applied) performance verification results

For intra-bank 15 transfers, the average TPS was measured at approximately 7.5, with an average latency of around 1,997ms, and a gas cost of approximately 42,000.

For inter-bank 15 transfers, the TPS also remained at 7.5, with an average latency of 2,001ms, and a gas cost of 128,000, indicating low resource consumption and fast response times.

These results are attributed to the absence of privacy-preserving technologies, which simplified the transaction processing flow and reduced computational complexity.

This set of results serves as a baseline for system performance without privacy features and provides a foundation for comparative analysis of performance changes following the application of privacy-enhancing technologies.

This result can be interpreted as the outcome of the API server and smart contracts performing relatively simple computations, without encrypting the address or amount information of the transacting parties, thereby achieving high processing efficiency.

4.3.4 Performance with Homomorphic Encryption Applied

As a result of performance testing on a digital currency transfer scenario using Homomorphic Encryption (HE) technology, experiments were conducted with 15 users, and the performance changes observed are summarized in [Table 7], [Figure 7].

Table 7. Homomorphic encryption interval performance (for intra-40, inter-15 transfer)

	Scenario	Interval	From	To	In	Out	TPS	Latency(ms)
①	Intra-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*2.9*	*5165*
②		Generate Transfer Data	API Server	HE Server	Sener, Receiver, Amount, Sender Bank's Pub-Key (ECHC), TD Balance(r)	Anonymity/TD Compare Set, Updater list	*N/A*	*626*
③		Send Tx	API Server	Besu	Anonymity/TD Compare Set	Tx Hash	*N/A*	*4356*
④		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	
⑤	Inter-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*1.8*	*8224*
⑥		Generate Transfer Data	API Server	HE Server	Sener, Receiver, Amount, Sender/Receiving Bank, Central Bank Pub-Key (ECHC), TD/CBDC Balance(r)	Anonymity/TD Compare Set, Updater list	*N/A*	*1434*
⑦		Send Tx	API Server	Besu	Anonymity Set, TD/CBDC Compare Set	Tx Hash	*N/A*	*6599*
⑧		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	

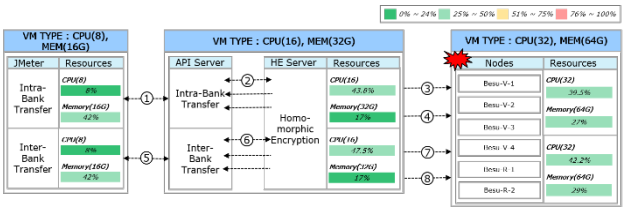

Figure 7. Homomorphic interval performance test architecture

In Scenario A, which involves intra-bank transfers, the application of homomorphic encryption led to a decrease in TPS from 7.5 to 2.9. The average latency increased from 1,997ms to 5,165ms, representing approximately a 2.6-fold increase. Furthermore, the GAS cost per transaction surged from 42,000 to 6,800,000, indicating about a 162-fold increase in computational cost. This demonstrates that homomorphic encryption significantly impacts performance, even in relatively simple intra-bank transfer scenarios.

In Scenario B, involving inter-bank transfers, TPS decreased from 7.5 to 1.8, and the average latency increased from 2,001ms to 8,224ms—approximately a 4.1-fold increase. The GAS cost also rose sharply from 128,000 to 10,900,000, indicating an approximately 85-fold increase in cost.

These results highlight the considerable computational overhead introduced by homomorphic encryption, which, while providing enhanced data confidentiality, presents challenges for real-time processing in blockchain-based digital currency systems.

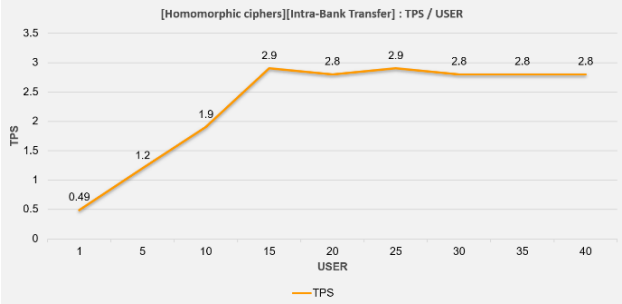

Figure 8. Homomorphic ciphers intra-bank transfer: TPS/USER

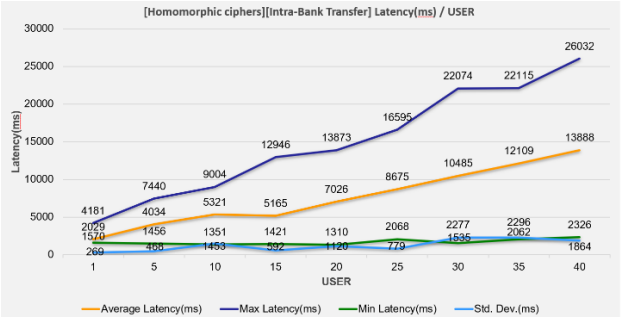

Figure 9. Homomorphic ciphers intra-bank transfer: Latency (ms)/USER

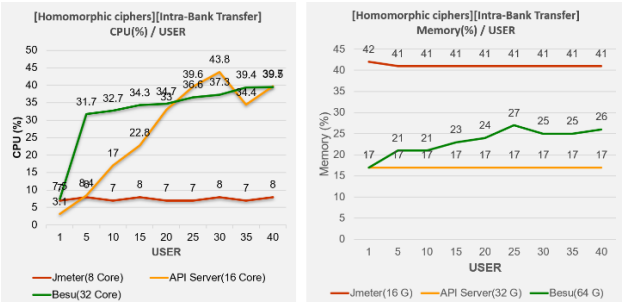

Figure 10. Homomorphic ciphers intra-bank transfer: CPU (%), Memory (%)/USER

As shown in [Figure 8], the TPS increased along with the number of users up to 15. However, beyond 15 concurrent users, the TPS plateaued, with the highest TPS recorded when the number of simultaneous users reached 15.

As shown in [Figure 9], the average latency exhibited a gradual upward trend overall, with the exception of the

10 to 15 user range. In the case of maximum latency, a sharp increase was observed as the number of users grew. While there was relatively little change in the 30 to 35 user range, another steep rise occurred at the 40-user mark. This suggests that once the system exceeds a certain threshold, response delays may worsen non-linearly.

As shown in [Figure 10], the CPU usage of the API server exhibited a gradual increase as the number of users grew, while the CPU usage of Besu remained relatively stable, ranging between approximately 30% and 40%. Additionally, memory usage for both the API server and Besu increased in tandem with the number of users, indicating that the transaction processing load has a tangible impact on system resources.

4.3.5 Performance with zk-SNARK Applied

The performance evaluation of a digital currency transfer scenario applying zk-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) technology was conducted under varying user and load conditions for both intra-bank transfers (Scenario A) and inter-bank transfers (Scenario B), as summarized in [Table 8], [Figure 11]. The experiment was carried out with 30 users for intra-bank transfers and 18 users for inter-bank transfers to quantitatively assess the impact of confidentiality-preserving technologies on system performance.

Table 8. zk-SNARK interval performance (for intra-40, inter-15 transfer)

	Scenario	Interval	From	To	In	Out	TPS	Latency(ms)
①	Intra-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*3.5*	*8489*
②		Balance Inquiry	API Server	Besu	Sender	Balance	*N/A*	*69*
③		Generate Transfer Data	API Server	HE Server	Sener, Receiver, Amount, Sender Bank's Pub-Key (ECHC), TD Balance, TD Balance Random Value (r)	Anonymity/TD Compare Set, Update (r) list	*N/A*	*948*
④		Send Tx	API Server	Besu	Anonymity/TD Compare Set, SNARK Proof	Tx Hash	*N/A*	*7252*
⑤		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	
⑥	Inter-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*1.6*	*11041*
⑦		Balance Inquiry	API Server	Besu	Sender	Balance	*N/A*	*81*
⑧		Generate Transfer Data	API Server	HE Server	Sener, Receiver, Amount, Sender/Receiving Bank, Central Bank Pub-Key (ECHC), TD/CBDC Balance, TD/CBDC Balance Random Value (r)	Anonymity/TD Compare Set, Update (r) list	*N/A*	*2327*
⑨		Send Tx	API Server	Besu	Anonymity Set, TD/CBDC Compare Set, SNARK Proof	Tx Hash	*N/A*	*8413*
Ⓐ		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	

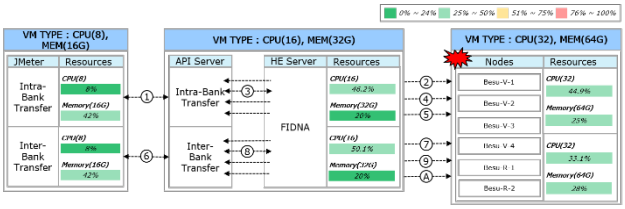

Figure 11. zk-SNARK interval performance test architecture

In the intra-bank transfer scenario, applying zk-SNARK resulted in a TPS of 3.5 and an average latency of 8,489ms. In contrast, for inter-bank transfers, the TPS dropped to 1.6, and the average latency increased to 11,041ms, reflecting more noticeable performance degradation due to the additional verification processes across multiple institutions.

To further clarify the effectiveness of zk-SNARK, a comparative experiment was conducted using 15 users, comparing environments with and without confidentiality features. In this comparison, the TPS for intra-bank transfers decreased from 7.5 to 2, while average latency increased from 1,997ms to 7,434ms. GAS costs also rose significantly from 42,000 to 6,250,000. This was primarily due to the computational load required for proof generation and the complexity of the verification logic within the smart contracts.

Similarly, for inter-bank transfers, the TPS declined from 7.5 to 1.5, and the average latency rose from 2,001ms to 9,474ms. GAS costs surged from 128,000 to 12,560,000—an increase of nearly 100 times. These results underscore the substantial performance trade-offs involved when incorporating zk-SNARK for privacy preservation. Detailed experimental results are provided below.

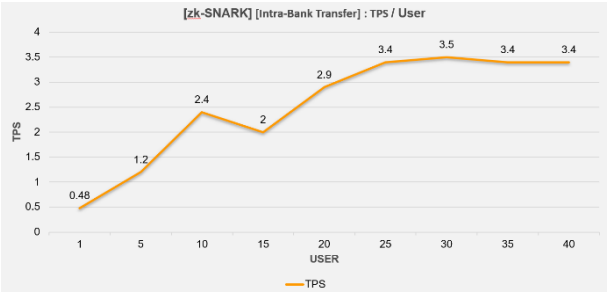

Figure 12. zk-SNARK intra-bank transfer: TPS/User

As shown in [Figure 12], the TPS increased along with the number of users up to 10. Between 10 and 15 users, the TPS plateaued, showing little to no growth. TPS began to rise again as the number of users increased up to 25, after which it stabilized and showed no significant changes beyond that point.

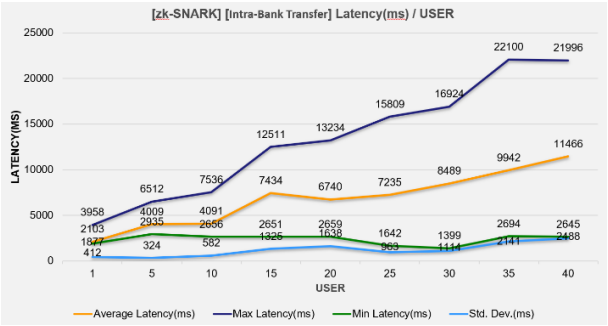

Figure 13. zk-SNARK intra-bank transfer latency (ms)/ USER

As shown in [Figure 13], the average latency remained relatively stable in the 5 to 10 user range and again between 15 and 25 users. Outside of these ranges, however, it showed a gradual upward trend as the number

of users increased. Meanwhile, maximum latency generally increased with the number of users, with particularly sharp spikes observed in the 10 to 15 user range and again between 30 and 35 users.

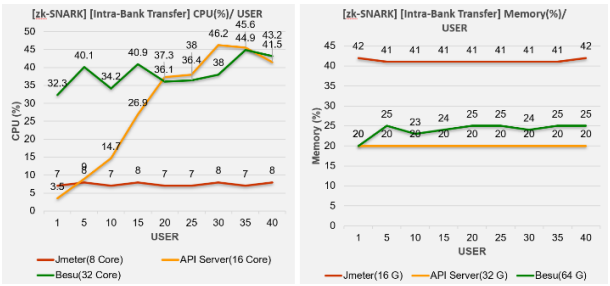

Figure 14. zk-SNARK intra-bank transfer CPU (%), Memory (%)/USER

As shown in [Figure 14], the resource usage analysis of the zk-SNARK-based system revealed that the CPU usage of the API server increased sharply as the number of users grew. In contrast, the CPU usage of Besu remained relatively stable, ranging between approximately 30% and 40%. Meanwhile, the memory usage of Besu remained consistent regardless of the number of users, suggesting that in this configuration, computational performance—rather than memory capacity—is the primary bottleneck.

4.3.6 Performance with zk-STARK Applied

The performance evaluation results of a digital currency transfer scenario applying zk-STARK (Zero-Knowledge Scalable Transparent Argument of Knowledge) technology are summarized in [Table 9]. This experiment measured performance under the assumption of 25 concurrent users for Scenario A (intra-bank transfer) and 12 concurrent users for Scenario B (inter-bank transfer). The effectiveness of zk-STARK was quantitatively analyzed through comparisons with a non-confidentiality environment [Figure 15].

Table 9. zk-STARK interval performance (for intra-40, inter-15 transfer)

	Scenario	Interval	From	To	In	Out	TPS	Latency(ms)
①	Intra-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*3.3*	*7539*
②		Balance Inquiry	API Server	Besu	Sender	Balance	*N/A*	*80*
③		Generate Transfer Data	API Server	HE Server	Sener, Receiver, Amount, Sender Bank's Pub-Key (ECHC), TD Balance, TD Balance Random Value (r)	Anonymity/TD Compare Set, Update (r) list	*N/A*	*905*
④		Send Tx	API Server	Besu	Anonymity/TD Compare Set, STARK Proof	Tx Hash	*N/A*	*6363*
⑤		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	
⑥	Inter-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*1.5*	*8060*
⑦		Balance Inquiry	API Server	Besu	Sender	Balance	*N/A*	*72*
⑧		Generate Transfer Data	API Server	HE Server	Sener, Receiver, Amount, Sender/Receiving Bank, Central Bank Pub-Key (ECHC), TD/CBDC Balance, TD/CBDC Balance Random Value (r)	Anonymity/TD Compare Set, Update (r) list	*N/A*	*2269*
⑨		Send Tx	API Server	Besu	Anonymity Set, TD/CBDC Compare Set, STARK Proof	Tx Hash	*N/A*	*5535*
Ⓐ		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	

In the intra-bank transfer scenario, with zk-STARK applied, the TPS was measured at 3.3, and the average latency was 7,539ms. These figures suggest that zk-STARK's transparency and quantum-resistance features contribute to increased system load. In the inter-bank transfer scenario, TPS was measured at 1.5, and the average latency at 8,060ms.

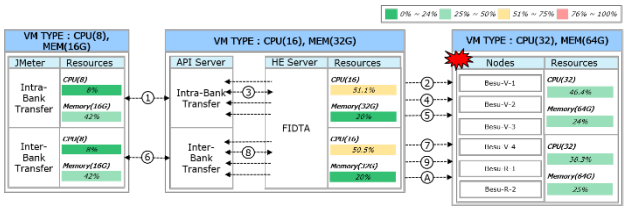

Figure 15. zk-STARK interval performance test architecture

Additionally, a performance comparison was conducted before and after applying zk-STARK, based on 15 concurrent users. For intra-bank transfers, the TPS decreased from 7.5 to 2.3, and the average latency increased from 1,997ms to 6,280ms—approximately a 3.1-fold increase. The GAS cost rose from 42,000 to 6,600,000.

In the case of inter-bank transfers, the TPS dropped from 7.5 to 1.5, and the average latency increased from 2,001ms to 9,759ms. The GAS cost increased from 128,000 to 12,900,000.

These results clearly indicate the performance trade-offs associated with zk-STARK implementation, especially in terms of computational cost and latency, despite its advantages in transparency and quantum security. Detailed experimental results are provided below.

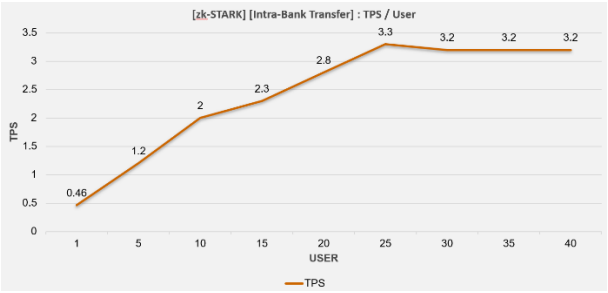

Figure 16. zk-STARK intra-bank transfer: TPS/USER

According to the results presented in [Figure 16], TPS (Transactions Per Second) showed a gradual increase proportional to the number of users in the range from 1 to 25 users. However, beyond 25 concurrent users, the growth in TPS began to slow, entering a plateau phase that indicates a saturation in processing performance. Notably, the maximum TPS was observed when the number of concurrent users reached 25, suggesting that the system's performance ceiling is reached at approximately 25 simultaneous users.

Average latency began at 2,170ms with a single user and gradually increased as the number of users grew,

reaching a peak of 12,078ms at 40 users. Meanwhile, maximum latency also showed a general upward trend proportional to the number of users. However, sharp spikes were observed in the 10–15 user range and again between 25 and 30 users. These results suggest that the system's latency performance can degrade non-linearly once specific user thresholds are exceeded [Figure 17].

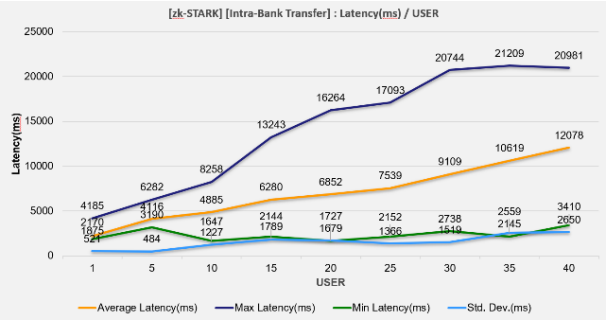

Figure 17. zk-STARK intra-bank transfer: Latency (ms)/ USER

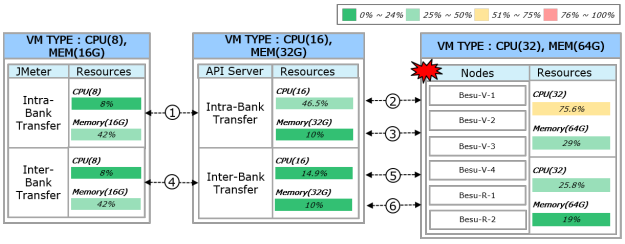

Figure 18. zk-STARK intra-bank transfer: CPU (%), Memory (%)/USER

As shown in [Figure 18], the CPU usage of the API server increased sharply with the number of users, whereas Besu's CPU usage remained relatively stable, ranging between approximately 30% and 45%. Despite the rise in user count, memory usage for both the API server and Besu stayed within the 18% to 25% range, without significant growth. This indicates that memory resources may be acting as a limiting bottleneck for processing performance in the system.

4.3.7 Performance with Ring Signature Applied

In this experiment, performance results with Ring Signature technology applied were compared to those without transaction privacy measures, based on scenarios involving 30 users. The comparison yielded the following results: [Table 10] [Figure 19]

Scenario A (Intra-bank Transfer), based on 30 users:

- TPS decreased from 7.5 to 5.0.
- Average latency increased from 1,997ms to 4,064ms.
- Gas cost increased from 42,000 to 590,000.

Scenario B (Inter-bank Transfer), based on 15 users:

- TPS dropped from 7.5 to 3.4.
- Average latency increased from 2,001ms to 4,056ms.
- Gas cost rose from 128,000 to 1,220,000.

The results indicate that the increase in smart contract processing costs and cryptographic computation bottlenecks on the API server were the primary contributing factors. Additionally, the Viewing Key-based conditional verification process was also found to contribute to the delay in transaction processing.

Table 10. RingCT interval performance (for intra-30, inter-15 transfer)

	Scenario	Interval	From	To	In	Out	TPS	Latency(ms)
①	Intra-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*5.0*	*4064*
②		Send Tx	API Server	Besu	TD : Stealth Address, Ephemeral Pub-Key, c-value, key image, ring sign	Tx Hash	*N/A*	*3457*
③		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	
④	Inter-Bank Transfer	Total	Jmeter	API Server	Sender, Receiver, Amount	Tx Receipt	*3.4*	*4056*
⑤		Send Tx	API Server	Besu	TD/CBDC : Stealth Address, Ephemeral Pub-Key, c-value, key image, ring sign	Tx Hash	*N/A*	*1652*
⑥		Get Tx Receipt	API Server	Besu	Transaction Hash	Tx Receipt	*N/A*	

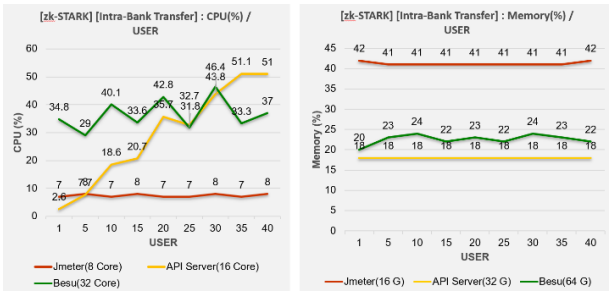

Figure 19. RingCT interval performance test architecture

Intra-Bank Transfer (TPS):

To evaluate the system's processing performance and stability, the number of concurrent users was gradually increased from 1 to 40 in increments of 5, and the Transactions Per Second (TPS) was measured. Due to a bottleneck in the API server, the TPS plateaued when the number of concurrent users exceeded 30. [Figure 20]

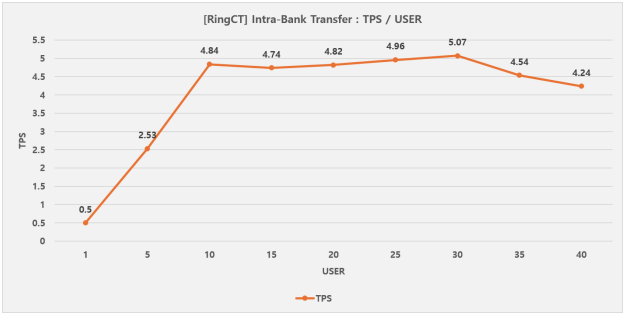

Figure 20. Ring Signature intra-bank transfer: TPS/USER

Intra-Bank Transfer (Latency):

To analyze the system's processing performance and stability, the number of concurrent users was increased from 1 to 40 in increments of 5, and the average (Average), maximum (Max), minimum (Min) latency, as well as the standard deviation (Std. Dev) were measured.

In the intra-bank transfer performance evaluation using Ring Signature, the latency showed a linear increase as the number of users grew. The average latency rose from 1,989ms with a single user to 6,813ms with 40 users, indicating that the system could not sufficiently support parallel processing, resulting in accumulated delays under higher user loads.

The maximum latency also increased noticeably with the number of users, from 2,086ms at 1 user to 9,426ms at 40 users. In contrast, the minimum latency remained relatively stable, ranging between 991ms and 3,140ms, regardless of the number of users.

The standard deviation grew from 112ms at 1 user to 2,190ms at 40 users, reflecting increased variability and fluctuation in latency under higher loads. These results indicate that, due to performance bottlenecks, the Ring Signature-based intra-bank transfer system exhibits increasing latency in proportion to user concurrency. [Figure 21]

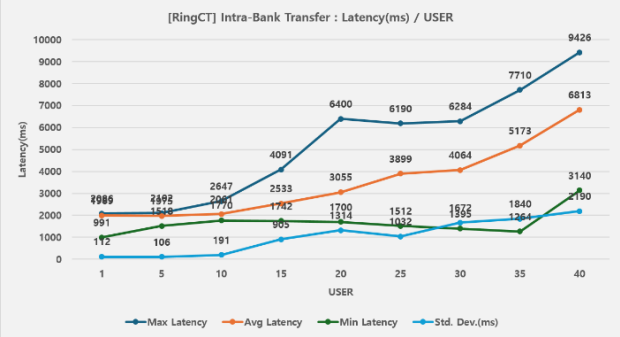

Figure 21. Ring signature intra-bank tansfer: Latency (ms)/ USER

Intra-Bank Transfer (CPU/Memory):

The CPU utilization of the Besu node increased as the number of concurrent users grew, rising from an initial value of 14% to a peak of 65.6%, before slightly dropping to 54.7% at 40 users. This trend indicates that the Besu node experienced a significant processing load as user concurrency increased.

In contrast, the CPU utilization of JMeter remained relatively stable, maintaining a consistent range between 7% and 8%, regardless of the number of users. The API server's CPU utilization increased from 6% with a single user to 67% at 40 users, reflecting the system's attempt to handle increased user requests through parallel processing.

In terms of memory utilization, JMeter maintained a stable usage level between 41% and 42%, showing no significant variation in response to increasing user load. Similarly, the Besu node's memory usage remained steady between 16% and 18%. The API server's memory utilization was also consistent, holding at approximately 8% even as the number of users increased. These findings suggest that no excessive memory stress occurred across system components, and the observed performance bottlenecks were primarily CPU-bound rather than memory-related. [Figure 22]

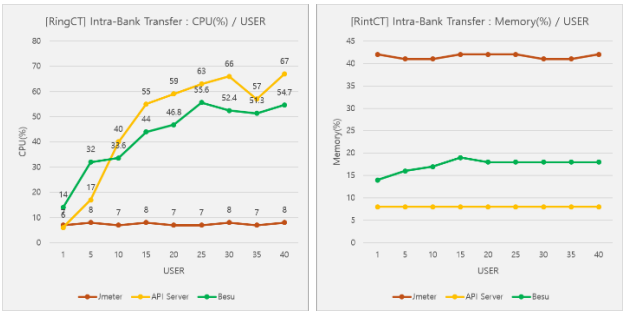

Figure 22. Ring signature intra-bank transfer: CPU (%), Memory (%)/USER

Inter-Bank Transfer:

To evaluate the system's processing performance and stability, the number of concurrent users was gradually increased from 1 to 15 in increments of 3, and Transactions Per Second (TPS) was measured.

The experimental results showed that despite the increase in the number of users, the TPS remained constant between 0.49 and 0.50, indicating a clear performance plateau. This suggests that a system bottleneck occurred as concurrency increased, and that parallel processing was not effectively achieved.

The main reason for the stagnation in TPS lies in the lock mechanism applied to CBDC accounts during inter-bank transfers. In this process, each transfer request must be processed sequentially, because CBDC balances are maintained separately by each institution, and the transfer requires synchronous state updates between two different financial institutions. As a result, simultaneous execution of multiple transfer requests is restricted, limiting system throughput even as the number of users increases.

4.3.8 Scalability Beyond 40 Concurrent Users

While performance evaluation in this study was conducted under up to 40 concurrent users, observable trends indicate a saturation point beyond which system throughput (TPS) plateaus and latency increases sharply. This behavior highlights several concurrency bottlenecks:

API Server Cryptographic Bottleneck:

The API server is responsible for generating stealth addresses, constructing ring signatures, and producing Pedersen Commitments. As user concurrency increases, these operations, particularly signature validation and key image generation, become CPU-intensive, leading to serialized processing delays.

Smart Contract Execution Limits:

Smart contracts that validate RingCT and perform range proofs incur high gas costs. With increasing users, contract execution approaches per-block gas limits, restricting the number of transactions that can be finalized per block, thereby throttling TPS.

Sequential Locking in Inter-bank Transfers:

The current architecture requires state synchronization between institutions (e.g., A Bank and B Bank) for inter-bank transactions. Locking mechanisms on institutional UTXO pools introduce sequential processing dependencies, making parallel execution infeasible under high load.

4.3.9 Analysis and Implications

The experimental results demonstrate that the application of privacy-preserving technologies is effective in enhancing system security and privacy levels. However, from the perspective of real-time performance and processing efficiency, the following limitations were identified:

Reduced Processing Speed:

Increased cryptographic computation per transaction significantly lowered TPS (transactions per second). In particular, as the size of the ring grows, the time required for signature generation and verification increases proportionally.

Increased Latency:

In addition to cryptographic overhead, other factors such as Viewing Key access checks and branching logic in smart contract verification contributed to overall processing delays.

Rising Costs:

In some cases, gas consumption increased more than tenfold, potentially leading to higher operational costs and decreased user acceptance.

These findings clearly illustrate both the structural advantages and the practical performance challenges of UTXO-based privacy technologies. They provide valuable empirical evidence for guiding future system optimization efforts and policy design for practical implementation.

5 Technical Considerations and Policy Consensus

This chapter provides a comprehensive examination of the proposed UTXO-based privacy-preserving technology as applied to an account-based CBDC system. The discussion focuses on its technical benefits and limitations, comparative analysis with existing privacy technologies, and its suitability from policy and regulatory perspectives.

5.1 Advantages and Limitations of Privacy-Preserving Technologies

The UTXO-based privacy-preserving architecture proposed in this study addresses a fundamental challenge in CBDC design: ensuring transaction confidentiality while preserving system integrity. Unlike traditional account-based models, this structure validates transactions without exposing sensitive data—such as sender, receiver, or transaction amount—on the blockchain. This design offers several key technical advantages:

First, the architecture significantly enhances privacy protection.

Technologies such as stealth addresses, ring signatures, and RingCT anonymize each transaction component. The receiver uses a one-time address, while the sender signs anonymously within a ring, ensuring that transaction participants remain unlinkable. This approach aligns with rising societal demands for data privacy and strengthens user trust.

Second, the system maintains data integrity and prevents double spending.

Mechanisms such as key image–based duplicate detection and range proofs verify transaction validity without compromising confidentiality.

Third, the stateless and decoupled nature of UTXO transactions supports scalability and parallelism.

Because each transaction is self-contained, the system handles concurrent processing more efficiently than account-based models.

However, the proposed approach also presents the following limitations:

- High computational overhead: Cryptographic operations such as signature generation and verification introduce significant latency and reduce throughput (TPS).
- Excessive smart contract costs: RingCT-related computations, performed on-chain, lead to elevated gas consumption.
- Governance complexity: Managing multiple key types (e.g., viewing keys, spend keys, key images) increases both user burden and operational challenges for institutions.

These limitations suggest that further optimization—such as lightweight cryptographic designs, off-chain computation, and better key governance—is essential for real-world deployment. Ultimately, the system's success hinges on its ability to balance privacy protection with regulatory compliance through technically sound and policy-aligned implementation.

5.2 Comparison with Existing Technologies

Privacy-preserving mechanisms for CBDCs draw heavily from cryptographic innovations in cryptocurrencies. The UTXO-based architecture proposed in this study occupies a distinct position between two dominant models: Monero-style full anonymity and ZKP–based confidentiality.

Monero, a privacy-centric cryptocurrency, integrates stealth addresses, ring signatures, and RingCT to fully conceal sender, receiver, and transaction amount. While effective in anonymizing transactions, Monero offers no means of regulatory access or traceability—rendering it incompatible with the auditability requirements of CBDC systems.

ZKP-based models (e.g., zk-SNARKs, zk-STARKs, Bulletproofs) mathematically prove transaction validity without disclosing data. While offering strong privacy and minimal reliance on trusted authorities, they introduce significant complexity and resource demands. High computation time, large proof sizes, and costly on-chain verification hinder their scalability in real-time, high-volume environments.

In contrast, this study's architecture incorporates viewing key–based conditional auditability, enabling selective access to encrypted transaction details under predefined conditions. This design offers a pragmatic compromise: preserving privacy while facilitating institutional oversight. Adjustable privacy levels, time-bounded audits, and role-based access control make the system more flexible and policy-compatible than existing alternatives.

In summary, the proposed model combines the strengths of Monero and ZKP-based approaches while mitigating their drawbacks—offering a technically feasible and regulator-friendly solution for national-level CBDC deployment.

5.3 Mitigation Strategies for Concurrency Scaling

To address these constraints and improve scalability beyond the current thresholds, the following technical strategies are proposed:

Batch Verification and Preprocessing:

Aggregate ring signature verifications and range proofs at the API server before submission to the blockchain. This amortizes computational cost across multiple transactions.

Off-chain Computation for Heavy Crypto:

Move cryptographic operations (e.g., MLSAG proof generation, range proof generation) off-chain and submit only verification results or zero-knowledge succinct proofs (e.g., zkRollup-style approach).

Parallel Ring Construction and Caching:

Use pre-generated ring member caches for mix-in selection, enabling parallel signature construction and reducing live UTXO sampling delays.

Asynchronous State Channel Design:

For inter-bank transfers, decouple the lockstep transaction confirmation flow using state channels or atomic commit protocols to improve concurrency without compromising consistency.

Dynamic Ring Size Adjustment:

Under high load, dynamically reduce ring size (n) while maintaining minimal anonymity guarantees to trade off cryptographic overhead for performance.

Horizontal Scaling of API Servers:

Deploy multiple load-balanced API server instances with stateless cryptographic modules to absorb peak traffic and distribute transaction load.

By applying these strategies, the proposed architecture can be extended to support retail-level transaction volumes, such as those anticipated in real-world CBDC deployments, without sacrificing privacy guarantees or system integrity.

5.4 Improvement Directions for Practical Adoption

Although the proposed UTXO-based architecture demonstrates high levels of privacy and security, several challenges remain in achieving real-time performance, processing efficiency, and operational viability. To support broader adoption, the following improvement directions are identified:

Optimize cryptographic performance:

High server load during stealth address generation, ring signature verification, and RingCT construction necessitates optimization. Solutions include precomputation, batch processing, and parallel cryptographic operations.

Modularize and streamline smart contracts:

Current contracts encapsulate complex cryptographic logic, resulting in high gas costs. Offloading computations to off-chain modules or modularizing smart contract functions can alleviate cost burdens.

Establish governance for viewing key management:

A policy framework for viewing key lifecycle—generation, delegation, revocation, and audit logging—is essential. Centralized or federated key management models can improve institutional operability and compliance.

Integrate lightweight privacy technologies:

Hybrid models combining RingCT with efficient ZKPs (e.g., Bulletproofs, zk-Rollups) can enhance scalability without sacrificing confidentiality, making the system viable for high-frequency transaction environments.

Improve user experience and developer accessibility:

Abstracting complex processes—such as key handling and transaction composition—through intuitive interfaces and SDKs will enable broader adoption among users and developers.

Institutionalization of Viewing Key Governance:

To ensure operational integrity and regulatory compliance, viewing key management should be institutionalized under a tiered governance model. In this framework, a central authority—such as the central bank—acts as the root of trust, issuing and certifying master keys to licensed financial institutions (e.g., commercial banks or payment providers). These institutions, in turn, are responsible for securely generating, distributing, and managing user-level viewing keys via institutional keystores. Key delegation and access rights can be enforced using Merkleized access control lists, while on-chain revocation registries enable tamper-evident tracking of expired or invalidated keys. Such a design ensures that only authorized auditors or regulatory entities can access encrypted transaction metadata, aligning with privacy-by-design and selective transparency principles. Integration with national digital identity infrastructure (e.g., eIDAS in the EU or K-Cert in Korea) further supports identity binding, audit traceability, and policy-driven key lifecycle governance.

By advancing these technical, governance, and usability components, the proposed architecture can evolve into a practical foundation for deploying scalable, privacy-preserving CBDC systems in real-world financial infrastructures.

6 Conclusion and Future Work

This study proposed a privacy-enhancing architecture for account-based CBDCs by integrating UTXO-based privacy-preserving technologies. Leveraging cryptographic primitives such as stealth addresses, ring signatures, and RingCT, the system effectively conceals the identities of transaction participants and the transferred amounts. Furthermore, the inclusion of a viewing key–based conditional audit mechanism allows for selective transparency, balancing privacy protection with regulatory compliance.

Experimental results confirm that the proposed architecture achieves strong transaction confidentiality. However, it also incurs performance trade-offs, including reduced throughput, increased latency, and higher gas

costs. In particular, computational bottlenecks in the API server—caused by cryptographic operations—and the cost-intensive structure of smart contract execution present challenges for real-time adoption in high-frequency financial systems.

Through both technical and institutional analysis, this research demonstrates the feasibility of a middle-ground CBDC model—offering greater privacy than conventional systems while avoiding the full anonymity of models like Monero or the heavy computational burden of ZKP–based systems. The viewing key framework is designed to align with legal and organizational requirements, underscoring its applicability in real-world deployment scenarios.

6.1 Future Deployment Roadmap

To translate the proposed architecture into real-world deployment, a phased integration strategy is essential. This roadmap involves the following stages:

Sandbox Testing Phase:

The system can first be deployed in a regulatory sandbox hosted by the central bank or a financial regulatory authority. This phase allows controlled testing of privacy guarantees, auditability mechanisms (via viewing keys), and system resilience under simulated transaction loads.

Pilot Program with Selected Institutions:

A limited-scale pilot involving a consortium of commercial banks, central bank nodes, and selected end users can validate the privacy-audit balance in intra- and inter-bank transactions. During this stage, performance under real transaction volume and institution-specific key management practices will be evaluated.

National Integration:

Based on pilot outcomes, the system can be integrated into a nationwide CBDC infrastructure, either as a modular privacy layer or as the core transaction engine. Integration with existing digital identity frameworks (e.g., eIDAS, K-Cert, or MyData platforms) will support role-based access to viewing keys, audit logs, and compliance enforcement.

This roadmap ensures that the architecture evolves from research prototype to policy-aligned, operationally viable digital currency infrastructure, bridging privacy preservation with institutional trust.

6.2 Future Research Directions Include:

- Optimization of cryptographic performance: Explore methods such as parallel signature verification, off-chain computation, and modular smart contracts to mitigate latency and resource consumption.
- Integration with lightweight zero-knowledge technologies: Investigate hybrid models combining RingCT with Bulletproofs, zk-Rollups, or other scalable privacy techniques to maintain confidentiality with improved efficiency.
- Governance frameworks for audit and key management: Develop institutional policies for viewing key access, delegation, revocation, and audit log transparency to support formal adoption.
- Enhancement of user experience and interface design: Design intuitive tools for key management and transaction processing to promote accessibility and broader public engagement.

In summary, this work serves as a practical demonstration of how privacy-enhancing technologies can be embedded into institutional CBDC infrastructures. It offers both a technical foundation and policy-aligned blueprint for the future design of privacy-preserving digital financial systems.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00565, Development of User identity certification and management technology for self-sovereign identity applications).

References

[1] A. Kosse, I. Mattei, *Gaining Momentum – Results of the 2021 BIS Survey on Central Bank Digital Currencies*, BIS Papers No. 125, May, 2022.

[2] U.S. Department of Homeland Security, *Combatting Illicit Activity Utilizing Financial Technologies and Cryptocurrencies Phase III*, DHS Report No. 2024-09, September, 2024. Available: https://www.dhs.gov/sites/default/files/2024-09/2024aepphaselllcombattingillicitactivityutilizingfinancial.pdf

[3] D. Li, W. E. Wong, S. Pan, L. S. Koh, M. Chau, Design Principles and Best Practices of Central Bank Digital Currency, *International Journal of Performability Engineering*, Vol. 17, No. 5, pp. 411–419, May, 2021. https://doi.org/10.23940/ijpe.21.05.p1.411421

[4] Monero Project, *Official Monero Website*, GetMonero.org. [Online]. Available: https://www.getmonero.org/, Accessed: February, 2015.

[5] K. M. Alonso, *Zero to Monero: First Edition*, Monero Publishing, 2018.

[6] S. Noether, A. Mackenzie, Ring Confidential Transactions, *Ledger*, Vol. 1, pp. 1–18, December, 2016. https://doi.org/10.5195/ledger.2016.34

[7] T. H. Yuen, S. F. Sun, J. K. Liu, M. H. Au, M. F. Esgin, Q. Zhang, D. Gu, RingCT 3.0 for Blockchain Confidential Transaction: Shorter Size and Stronger Security, *Financial Cryptography and Data Security: 24th International Conference, FC 2020*, Kota Kinabalu, Malaysia, 2020, pp. 464–483. https://doi.org/10.1007/978-3-030-51280-4_25

[8] Monero Project, *Stealth Address*, GetMonero.org. [Online]. Available: https://www.getmonero.org/resources/moneropedia/stealthaddress.html, Accessed: March 28, 2025.

[9] Hyperledger Foundation, *Hyperledger Besu Documentation*, 2024. [Online]. Available: https://besu.hyperledger.org/, Accessed: July, 2024.

[10] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, White Paper, https://bitcoin.org/bitcoin.pdf, October, 2008.

[11] D. Yaga, P. Mell, N. Roby, K. Scarfone, *Blockchain Technology Overview*, arXiv preprint arXiv:1906.11078,

June, 2019.
https://arxiv.org/abs/1906.11078

[12] Z. Zheng, S. Xie, H. N. Dai, X. Chen, H. Wang, Blockchain Challenges and Opportunities: A Survey, *International Journal of Web and Grid Services*, Vol. 14, No. 4, pp. 352–375, December, 2018.
http://dx.doi.org/10.1504/IJWGS.2018.095647

[13] M. Nofer, P. Gomber, O. Hinz, D. Schiereck, Blockchain, *Business & Information Systems Engineering*, Vol. 59, No. 3, pp. 183–187, June, 2017.
https://doi.org/10.1007/s12599-017-0467-3

[14] A. Sunyaev, Distributed Ledger Technology, in: *Internet Computing*, Springer, Cham, 2020, pp. 265–299.

[15] T. Wahrstätter, M. Solomon, B. DiFrancesco, V. Buterin, *Stealth Addresses, Ethereum Improvement Proposals*, ERC-5564, August, 2022. [Online]. Available: https://eips.ethereum.org/EIPS/eip-5564

[16] T. Wahrstätter, *Stealth Address Utilities*, Based on ERC-5564, March, 2025. [Online]. Available: https://nerolation.github.io/stealth-utils/

[17] H. Armelius, C. A. Claussen, I. Hull, *On the Possibility of a Cash-like CBDC*, Sveriges Riksbank Staff Memo No. 2, February, 2021. https://www.riksbank.se/globalassets/media/rapporter/staff-memo/engelska/2021/on-the-possibility-of-a-cash-like-cbdc.pdf

[18] Monero Project, *Ring Signature*, GetMonero.org. [Online]. Available: https://www.getmonero.org/resources/moneropedia/ringsignatures.html, Accessed: March 15, 2018.

[19] E. Fujisaki, K. Suzuki, Traceable Ring Signature, *International Workshop on Public Key Cryptography*, Beijing, China, 2007, pp. 181–200.

[20] A. Poelstra, A. Back, M. Friedenbach, G. Maxwell, P. Wuille, Confidential Assets, *Financial Cryptography and Data Security 2018, International Conference*, Nieuwpoort, Curaçao, 2018, pp. 43–63.
https://doi.org/10.1007/978-3-662-58820-8_4

[21] U. Fiege, A. Fiat, A. Shamir, Zero Knowledge Proofs of Identity, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 1987, pp. 210–217.
https://doi.org/10.1145/28395.28419

[22] O. Goldreich, Y. Oren, Definitions and Properties of Zero-Knowledge Proof Systems, *Journal of Cryptology*, Vol. 7, No. 1, pp. 1–32, December, 1994.
https://doi.org/10.1007/BF00195207

[23] A. De Santis, G. Persiano, Zero-Knowledge Proofs of Knowledge Without Interaction, *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, Pittsburgh, PA, USA, 1992, pp. 427–436.
https://doi.org/10.1109/SFCS.1992.267809

[24] O. Goldreich, H. Krawczyk, On the Composition of Zero-Knowledge Proof Systems, *SIAM Journal on Computing*, Vol. 25, No. 1, pp. 169–192, February, 1996.
https://doi.org/10.1137/S0097539791220688

[25] T. Chen, H. Lu, T. Kunpittaya, A. Luo, *A Review of zk-SNARKs*, arXiv preprint arXiv:2202.06877, February, 2022.
https://arxiv.org/abs/2202.06877

[26] A. Nitulescu, *zk-SNARKs: A Gentle Introduction*, Ecole Normale Superieure Technical Report No. ENS-2020-12, June, 2020.

[27] A. M. Pinto, An Introduction to the Use of zk-SNARKs in Blockchains, *Mathematical Research for Blockchain Economy: 1st International Conference MARBLE 2019*, Santorini, Greece, 2020, pp. 233–249.
https://doi.org/10.1007/978-3-030-37110-4_16

[28] A. Banerjee, M. Clear, H. Tewari, Demystifying the Role of zk-SNARKs in Zcash, *2020 IEEE Conference on Application, Information and Network Security (AINS)*, Kota Kinabalu, Malaysia, 2020, pp. 12–19.
https://doi.org/10.1109/AINS50155.2020.9315064

[29] S. T. Nassar, A. Hamdy, K. Nagaty, Hybrid zk-STARK and zk-SNARK Framework for Privacy-Preserving Smart Contract Data Feeds, *2024 International Conference on Computer and Applications (ICCA)*, Cairo, Egypt, 2024, pp. 55–62.
https://doi.org/10.1109/ICCA62237.2024.10928100

[30] A. E. Panait, R. F. Olimid, On Using zk-SNARKs and zk-STARKs in Blockchain-Based Identity Management, *13th International Conference on Innovative Security Solutions for Information Technology and Communications (SecITC 2020)*, Bucharest, Romania, 2020, pp. 130–145.
https://doi.org/10.1007/978-3-030-69255-1_9

[31] O. Fomenko, *ZK-STARKs Explained*, Technical Report, July, 2024.

[32] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, G. Maxwell, Bulletproofs: Short Proofs for Confidential Transactions and More, *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2018, pp. 315–334.
https://doi.org/10.1109/SP.2018.00020

[33] Digital Pound Foundation, *China Reports Rapid Growth for Digital Yuan: 180 Million Wallets and 7 Trillion Yuan in CBDC Transactions*, October 12, 2024. [Online]. Available: https://digitalpoundfoundation.com/china-reports-rapid-growth-for-digital-yuan-180-million-wallets-and-7-trillion-yuan-in-cbdc-transactions

[34] Z. Wang, H. Jin, W. Dai, K. K. R. Choo, D. Zou, Ethereum Smart Contract Security Research: Survey and Future Research Opportunities, *Frontiers of Computer Science*, Vol. 15, No. 2, Article No. 152802, April, 2021.
https://doi.org/10.1007/s11704-020-9284-9

[35] Ethereum Foundation, *Introduction to Smart Contracts*, 2024. [Online]. Available: https://ethereum.org/en/developers/docs/smart-contracts/, Accessed: July, 2024.

[36] E. Morais, T. Koens, C. Van Wijk, A. Koren, A Survey on Zero Knowledge Range Proofs and Applications, *SN Applied Sciences*, Vol. 1, No. 8, Article No. 946, August, 2019.
https://doi.org/10.1007/s42452-019-0989-z

[37] M. Christ, F. Baldimtsi, K. K. Chalkias, D. Maram, A. Roy, J. Wang, SoK: Zero-Knowledge Range Proofs, *Cryptology ePrint Archive*, Article No. 2024/430, 2024.
https://eprint.iacr.org/2024/430

Biographies

To-Min Ok received the B.S. degree in computer science from Dongguk University, South Korea, in 1998, the M.S. degree in computer engineering from Seoul National University (SNU), South Korea, in 2000. He is currently an Associate Professor with the Department of Software, Soongsil University, South Korea. His research interests include CBDC, AI Security and Multimedia & Networks

Jun-Seok Shin received the B.S. degree in software from Seowon University, South Korea, in 2023. He is currently pursuing the M.S. degree with the Department of Software, Soongsil University, South Korea. His research interests include user authentication, CBDC, DID, generative AI, and AI security.

Jun-Sung Kim is currently an undergraduate senior in the Department of AI Convergence at Soongsil University, South Korea. His research interests include CBDC, AI security, large language models (LLMs), and multimodal systems.

Dae-Seon Choi (Member, IEEE) received the B.S. degree in computer science from Dongguk University, South Korea, in 1995, the M.S. degree in computer science from Pohang Institute of Science and Technology, South Korea, in 1997, and the Ph.D. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2009. He was a Professor with the Department of Medical Information, Kongju National University, South Korea, from September 2015 to August 2020. He is currently a Professor with the Department of Software, Soongsil University, South Korea. His research interests include identity management and information security.