Received April 25, 2022, accepted June 8, 2022, date of publication June 14, 2022, date of current version June 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3183092

CEV Framework: A Central Bank Digital Currency Evaluation and Verification Framework With a Focus on Consensus Algorithms and Operating Architectures

SI YUAN JIN AND YONG XIA
HSBC Laboratory, Guangzhou 510440, China
Corresponding author: Yong Xia (yong.xia@hsbc.com)

ABSTRACT We propose a Central Bank Digital Currency Evaluation and Verification (CEV) Framework for recommending and verifying technical solutions in the central bank digital currency (CBDC) system. We demonstrate two sub-frameworks: an evaluation sub-framework that provides consensus algorithm and operating architecture solutions and a verification sub-framework that validates the proposed solutions. Our framework offers a universal CBDC solution that is compatible with different national economic and regulatory regimes. The evaluation sub-framework generates customized solutions by splitting the consensus algorithms into several components and analyzing their impacts on CBDC systems. CBDC design involves a trade-off between system features - the consensus algorithm cannot achieve all system features simultaneously. However, we also improve the operating architectures to compensate for the weak system features. The verification sub-framework helps verify our proposed solution through empirical experiments and formal proof. Our framework offers CBDC designers the flexibility to iteratively tune the trade-off between CBDC system features for the desired solution. To the best of our knowledge, we are the first to propose a framework to recommend and verify CBDC technical solutions.

INDEX TERMS Central bank digital currency, evaluation sub-framework, consensus algorithm, operating architectures, verification sub-framework.

I. INTRODUCTION

The recent development in cryptography and distributed ledger technology (DLT) has seen a new form of currency known as central bank digital currency (CBDC) [1]. More than 85% of central banks worldwide have started CBDC research [2], [3]. However, notably fewer scientific papers consider CBDC technical implementations and verification approaches. The papers [4], [5] consider blockchain networks to provide CBDC services and propose a new consensus algorithm to satisfy CBDC system features. However, the proposed solutions cannot be applied to different scenarios.

The overall operating architecture [7] and consensus algorithms are core components of CBDC technical solutions. The operating architecture defines different CBDC networks, while the consensus algorithms define how the specific CBDC network functions and impacts many CBDC system features. Central banks have different requirements regarding the priority of CBDC system features [6], such as performance, privacy and security. Consensus algorithms are typically associated with blockchain networks, but they can be applied in any network to form data consistency. For example, China [10] does not use blockchain in its CBDC prototype. However, we believe that the CBDC networks use consensus algorithms to form data consistency. Furthermore, given the different economic and regulatory requirements across jurisdictions, central banks need customized consensus algorithms and operating architectures to satisfy their CBDC system features.

A. OUR CONTRIBUTION

Our paper reviews previous CBDC solutions and proposes a framework that provides an overall operating architecture

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

and customized consensus algorithms to satisfy different CBDC system features. Based on previous research [4], [5], [11], we contribute the following:

1) We propose a framework to recommend and verify CBDC related technical solutions in the form of consensus algorithms and operating architectures.
2) We are the first to split consensus algorithms into different components, significantly improving the efficiency of designing customized solutions.
3) We improve the CBDC operating architecture to solve the issues relating to business secrecy.

Specifically, we propose an evaluation sub-framework that provides holistic CBDC solutions covering CBDC technical specifications in Section III-A. We build a verification sub-framework to verify the feasibility and rationality of proposed solutions in Section III-B. Finally, we integrate both sub-frameworks into one framework, called the CBDC Evaluation and Verification (CEV) Framework.

The evaluation sub-framework involves the consensus algorithm and operating architecture. The consensus algorithm works by forming data consistency among participants [14]. Therefore, it directly impacts many CBDC system features, including performance, privacy and security. For example, Zhang *et al.* [30] studies how blockchain empowers CBDC and proposes a new consensus algorithm to improve CBDC performance. However, consensus algorithms have a complex impact on other CBDC system features, so we require a more systematic method to analyze the effects. Our solution is to split the consensus algorithms into their constituent components to derive the individual impacts before recombining them.

A trade-off between CBDC system features exists in implementing consensus algorithms [6], and we cannot achieve all features simultaneously. However, we can improve the operating architecture to compensate for some weak CBDC system features. For example, we use new operating architectures to solve the business secrecy issue (details in Section II-B).

The verification sub-framework can guide CBDC designers to verify proposed solutions. CBDC designers need to build a mathematical model for the solution and verify whether it meets initial expectations on diverse CBDC system features, like high performance. If an alternative trade-off is required, the evaluation sub-framework can be revisited to adjust the solution.

We then introduce a CBDC scenario to demonstrate the CEV framework in Section IV. We use the evaluation sub-framework to propose customized consensus algorithms and apply the verification sub-framework to develop a model to verify related CBDC system features. We use empirical experiments to test performance and leverage formal proof to verify security and privacy. Finally, the example offers a clear guide on satisfying CBDC system features for different national economic and regulatory conditions (details in Section III.A.1).

B. PAPER STRUCTURE

The remainder of this paper is organized as follows. Section II presents the research background and three CBDC system features. Section III introduces the CEV framework, including an evaluation sub-framework and a verification sub-framework. Section IV provides an example of leveraging the framework to develop a solution and verify it. Finally, we conclude the paper in Section V.

II. BACKGROUND

A. BLOCKCHAIN AND CONSENSUS ALGORITHM

Blockchain has shown many benefits among current CBDC projects worldwide [4], [11], [12], [16], [17]. For example, peer-to-peer payments could save liquidity and improve efficiency in cross-border transactions [18]. Research topics about blockchain frequently appear, especially on the topic of the performance [35]–[38]. However, the performance of blockchain, for example, bitcoin [33], cannot meet today's commercial needs.

Consensus algorithms play roles in many blockchains. Fabric [35] used Practical Byzantine Fault Tolerance (PBFT) [34] which provides fault tolerance while sacrificing performance. The Corda blockchain protocol aims to satisfy financial and regulatory requirements [36]. Transactions in the Corda platform are recorded only by participants rather than the entire network. This configuration achieves high performance and protects privacy but sacrifices security.

B. TIERED ARCHITECTURE AND BUSINESS SECRECY ISSUE

A tiered architecture [7] plays a role in many CBDC projects, including China's E-CNY [13], and Sweden's E-Krona [40].

Figure 1 shows a typical tiered CBDC architecture. A consensus algorithm runs in a consensus network. In a two-tier CBDC architecture, tier-1 institutions directly connect to the central bank (tier-0), and tier-2 institutions directly connect to tier-1 institutions. Tier-1 institutions take the responsibility of distribution[1] in a two-tier CBDC architecture. In most CBDC proposals, commercial banks become tier-1 institutions. However, it is not feasible to let all commercial institutions become tier-1 and be responsible for CBDC distribution and circulation[2] because the central bank does not have the capacity to connect too many banks simultaneously. Furthermore, the central bank risks a single point of failure and performance bottleneck in this configuration. Therefore, a select number of influential banks usually become tier-1 institutions in CBDC projects.

The more commercial institutions that circulate CBDC, the more areas CBDC services cover. Tier-2 institutions have to connect to tier-1 institutions to provide CBDC services. However, tier-2 institutions and tier-1 institutions are generally competitors, and they are reluctant to provide transaction and

[1]Distribution means that an institution helps the central bank issue CBDC and manage CBDC authentication work.

[2]Circulation means that an institution provides CBDC-related transfer services.

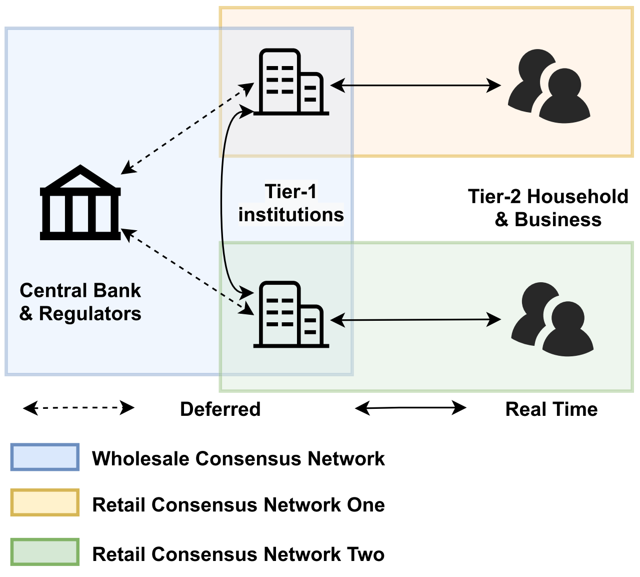
Tier-1 institutions
Tier-2 Household & Business
Central Bank & Regulators
Deferred
Real Time
Wholesale Consensus Network
Retail Consensus Network One
Retail Consensus Network Two

FIGURE 1. A two-tier architecture where consensus algorithms run separately in different consensus networks. The wholesale consensus network involves central banks and tier-1 institutions, and it handles wholesale transactions between tier-1 institutions and central banks; the retail consensus networks involve retail clients and tier-1 institutions, and it handles retail transactions between tier-2 households & business and tier-1 institutions.

customer information to each other. For example, if all tier-1 institutions are banks, the tier-2 banks may have concerns that tier-1 banks can monopolize their customer data. The business secrecy issue makes implementing two-tier architecture in a CBDC system challenging.

Current technical solutions to maintain business secrecy, such as homomorphic encryption [15], however, cannot satisfy CBDC system features because it significantly impacts performance. Therefore, we propose new operating architectures to improve CBDC business secrecy (details in Section III-A.2).

C. CBDC SYSTEM FEATURES

CBDC system features [6] measure CBDC-related considerations for designers and regulators. CBDC white papers [8], [9], [13] present many differences between jurisdictions regarding national conditions, and central banks focus on different CBDC system features. For example, Singapore's Ubin [8], and Canada's Jasper [9] focus on transaction settlement between different countries; China's E-CNY [13] emphasizes the volume of transactions per second in retail transactions. CBDC designers across different jurisdictions have varying approaches to satisfy CBDC system features.

Based on previous research [8], [9], [20]–[29], we summarize CBDC system features being used across different jurisdictions.

1) PERFORMANCE

Blockchain has many benefits and has been widely used in the wholesale CBDC, but seldom appears in retail CBDC projects [39]. One key factor is that its weak scalability cannot meet high performance. In CBDC scenarios, millions, even billions of customers may use CBDC, which requires a high performance to handle billions of requests. Therefore, we consider the following features to measure performance:

1) User Scalability: the cost of adding a new customer to a CBDC system.
2) Network Scalability: the capability to handle larger transaction volumes per second(TPS).
3) Latency: the time to complete one transaction.

We use empirical experiments to examine performance in the verification sub-framework and present an example in Section IV-B.2.

2) SECURITY

Central banks usually prioritize security. Security in a CBDC system involves various aspects, including cryptography, secure channels, key management, and prevention of double-spending attacks [32]. Specifically, the prevention of double-spending is one of the basic requirements in a CBDC system, playing a role in maintaining financial stability and reliability. CBDC has several security risks:

1) Cyber-Security: capability of protecting against outside attacks, especially double-spending attacks.
2) Resilience: capability of protecting against hardware issues, power or network outages, or cloud service interruption [28].

We use formal proof to verify potential security threats in the verification sub-framework and give an example in Section IV-B.4.

3) PRIVACY

We divide privacy into two aspects [28]:

1) Customer privacy protects customer data from being accessed by others.
2) Business secrecy prevents business data from leaking to business competitors.

Our evaluation sub-framework improves the current operating architecture to protect business secrecy in Section III-A.2 since the consensus algorithm cannot simultaneously achieve all CBDC system features. We use formal proof to verify privacy protection in the verification sub-framework and present an example in Section IV-B.3.

4) OTHERS

Other CBDC system features do not conflict with those stated above. Examples include governance [28], functionality, interoperability and offline payments. We believe these requirements can be met or solved independently in the current financial system. Therefore we exclude them in the following analysis. In future work, we may involve more CBDC system features in the CEV framework if required.

III. CEV FRAMEWORK

The CBDC Evaluation and Verification (CEV) framework includes two sub-frameworks: an evaluation sub-framework

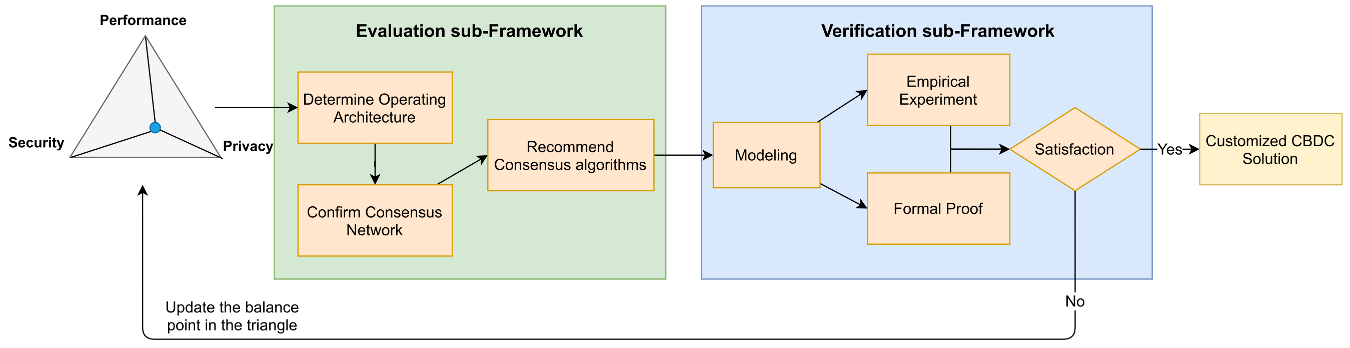
Performance
Security
Privacy
Evaluation sub-Framework
Determine Operating Architecture
Confirm Consensus Network
Recommend Consensus algorithms
Verification sub-Framework
Modeling
Empirical Experiment
Formal Proof
Satisfaction
Yes
Customized CBDC Solution
No
Update the balance point in the triangle

FIGURE 2. A closed-loop workflow of CEV Framework for CBDC designers.

that provides CBDC solutions and a verification sub-framework that proves the feasibility and rationality of recommended solutions.

Figure 2 shows the workflow of the CEV framework. First, CBDC designers determine the most important CBDC system features according to their economic and regulatory conditions. The evaluation sub-framework then determines the operating architecture to confirm the number of consensus networks and their relationship before recommending consensus algorithms for different consensus networks. Secondly, the verification sub-framework can guide CBDC designers to build a theoretical model for the solution and carry out experiments and proof to verify whether the solution meets the original CBDC system features. Finally, suppose the CBDC designers are not satisfied with the solution. In this case, they can return to adjust their preference on CBDC system features, leverage the evaluation sub-framework to update their solutions, and use the verification sub-framework to check the proposed solutions again.

A. EVALUATION SUB-FRAMEWORK

The evaluation sub-framework includes the consensus algorithm and the operating architecture. The consensus algorithm part splits consensus algorithms into different components, and the operating architecture part introduces the overall architecture that consensus algorithms run. We introduce how we recommend consensus algorithms and improve the operating architecture.

1) CONSENSUS ALGORITHM

Consensus networks implement consensus algorithms in diverse ways. For example, in the operating architecture (figure 1), the central bank can control the wholesale balance of issued CBDC rather than recording every retail transaction to avoid double-spending [32]. The central bank is only responsible for issuance and redemption transactions. If any issue exists in retail transactions, the corresponding tier-1 institutions should be accountable. Conversely, the central bank records every wholesale transaction to keep it safe. In this situation, both the wholesale and retail transactions are safe from double-spending. Overall, the consensus algorithm and operating architecture work together to form data consistency in this example.

Consensus algorithms can satisfy different CBDC system features with a trade-off [6]. As a result, they have direct but complex impacts on the CBDC system features (Section II).

We reviewed many consensus algorithms [14], [33], [34], [36], [44], [45] and found small variations between them that we need to take into account for a generalized solution. For example, IBFT [44] adopts a dynamic set of validators compared with a fixed set of validators in PBFT [34]. We split the consensus process into 6 different modules based on the variation identified. This allows us to better analyze the individual impacts on the CBDC system features. We then combine different components into one algorithm to assess the overall impact.

Figure 3 introduces the consensus algorithm components. The following steps describe the process:

1) Network - Election: a network requires one representative to lead the consensus process before a client sends a transaction request.
   a) Voting: the system votes for the leaders. Voting can involve extensive voting mechanisms, such as defining the percentage of votes to become a leader. RAFT [14] adopts an election timeout in the voting mechanism to determine the network's leader.
   b) Predetermination: the system predetermines the leaders. For example, the notary node in the Corda platform [36] is determined before network deployment.
   c) Round-robin: A group of nodes take turns as leaders in a certain order. PBFT [34] uses the round-robin approach to choose the primary (leader).
   d) Proposer: the system has no leader but proposers. For example, in some public blockchain systems, a proposer collects transactions from users and proposes them to the network via Proof of Work [33], Proof of Stake [45].
2) Client - Request: a client submits its payment request to the network.

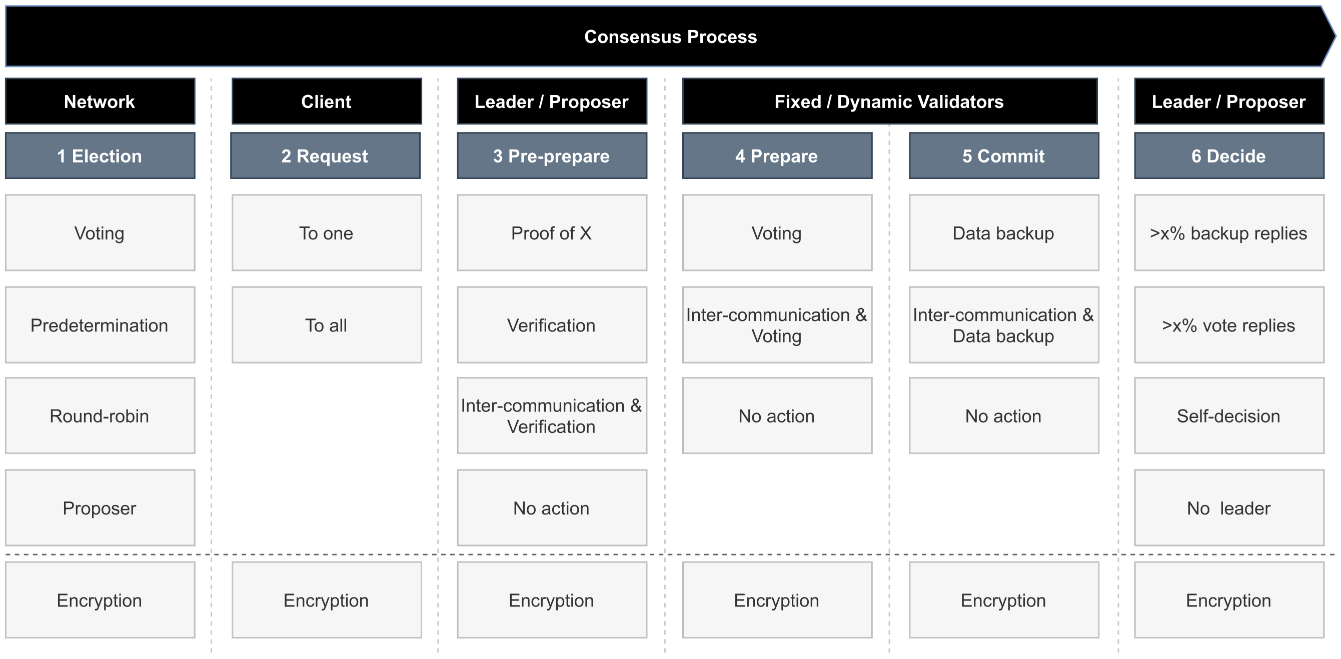

FIGURE 3. **Consensus Process: a CBDC client sends a request to the CBDC system. The system processes the request until reaching an agreement inside the network. We split the process into several components to better analyze each one.**

a) To one: a client sends the request to only one node. A node is a connection point in a communication network. For example, in RAFT, the client sends the request to one node and resends it to another node if no response is received. Furthermore, the "To one" option can leverage sharding [41], [42] to improve system performance. Sharding involves multiple nodes processing transactions in parallel and interacting with each other in a specific manner (discussed in the example of Section IV).

b) To all: a client sends the request to all nodes to ensure a node accepts the request in time.

3) Leader / Proposer - Pre-prepare: the leader node or proposer processes the request locally after receiving it.
   a) Proof of X: the leader or proposer leverages one of Proof of X, such as Proof of Work [33], to publish transactions.
   b) Verification: the leader verifies proposed transactions in a specific manner, like checking the input equals the output.
   c) Inter-communication & Verification: the leader communicates with other nodes and filters transactions. Filtration can reduce the illegal transactions.
4) Validators - Prepare: validators vote and communicate with others to verify the request from the leader node.
   a) Voting: validators vote for the request.
   b) Inter-communication & Voting: validators communicate with each other and vote for the request. PBFT [34] adopts this method to prevent malicious behavior.
   c) No action: validators do not contribute the decision. For example, validators (called followers) in RAFT [14] do not vote for any decision.
5) Validators - Commit: validators record transactions in their database.
   a) Data backup: validators make a backup in their local databases. RAFT [14] directly undertake a data backup after receiving it.
   b) Inter-communication & Data backup: validators communicate before backup. PBFT [34] adopts this method to ensure its safety.
6) Leader / Proposer - Decide: the leader finalizes the request.
   a) x% backup replies: the leader receives more than x% backup replies before responding. For example, in RAFT [14], the leader node must receive 50% replies.
   b) x% vote replies: the leader receives more than x% vote replies before it can finalize the transaction.
   c) Self-decision: the leader decides the transaction by itself. For example, the notary node in Corda [36] verifies proposed transactions by itself.

In addition to the above options, we include two further to improve the overall algorithm:

1) Fixed / Dynamic Validators: validators are non-leader nodes that can participate in consensus. Some consensus algorithms require all nodes to participate in consensus, while others choose selected or dynamic random nodes. For example, RAFT [14] and PBFT [34] require all nodes to participate data backup, while IBFT [44] adopts a dynamic set of validators.

TABLE 1. Impact of individual consensus components. The first column is consensus components. The first row is CBDC system features.

Components		Performance			Security		Privacy	
		User Scalability	Network Scalability	Latency	Resilience	Cyber-Security	Customer Privacy	Business Secrecy
1 Leader Election / Proposer	Voting	TBA	Medium	Medium	TBA	High	TBA	TBA
	Predetermination		High	High		TBA		High
	Round-robin		High	High			Low	Medium
	Proposer	High	High	Low			TBA	TBA
	Encryption	TBA	TBA	Low				
2 Request	To one	High	High	Low			Medium	
	To all	Low	Medium	High			Low	
	Encryption	TBA	TBA	Low			TBA	
3 Pre-Prepare	Proof of X			TBA				
	Verification	High	High	High		High		
	Inter-communication & Verification	Low	Low	Low		High	Low	Low
	No action	High	High	High	Medium	Low	Medium	Medium
	Encryption	TBA		Low	TBA	TBA	High	High
4 Prepare	Voting	Medium	Medium	Medium			Low	TBA
	Inter-communication & Verification	Low	Low	Low			Low	Low
	No action	TBA	TBA	TBA			TBA	TBA
	Encryption			Low				High
5 Commit	Data Backup	Medium	Medium	Medium	High			TBA
	Inter-communication & Verification	Low	Low	Low	High			
	No action	TBA	TBA	TBA	TBA			
	Encryption			Low	TBA			
6 Decide	>1/x Vote	Medium		TBA	High	High		Low
	>1/x Backup	Meidum			High	Medium		Medium
	Self-decision	High	High		Low	Low		High
	No leader	High	Low		High	High		Medium
Validators	Fixed	TBA	Low	Low	High	High		TBA
	Dynamic		High	High	Medium	Medium		

A dynamic set of validators can provide a higher performance due to a limited number of participants. In contrast, fixed validators are easier to implement and more secure.

2) Encryption: CBDC designers can use encryption in every step to secure transmitted information. Encryption can increase security but decrease performance. This option is independent of previous options. In most algorithms, consensus algorithms are more associated with achieving consistency and ignore encryption. However, in CBDC scenarios, encryption plays a vital role, so we include it in our consensus process.

Each component has many extensions, and components have constraints between each other. For example, "proof of X" in the third step is possibly connected with "proposer" in the first step. However, combining "proof of X" with other options is possible, but this needs to be verified in the verification sub-framework.

Table 1 shows how the above components influence the CBDC system features (Section II). We measure the impact through categories of High,[3] Medium[4] and Low.[5] TBA indicates the impact needs to be further analyzed. Each component has its impact on every CBDC dimension. We measure these impacts by empirical experiments and formal proof. Different components together can build one consensus algorithm with equal weighting. If needed, we can use a different weighting of impacts on CBDC system features.

[3]High means the component has a relatively positive impact on the system feature.

[4]Medium means the component has no impact or relatively medium impact on the system feature.

[5]Low means the component has a relatively negative impact on the system feature.

We have referenced RAFT several times. Here we use the Consensus Process to describe the RAFT consensus algorithm in figure 4. Then we leverage Table 2 to show that the RAFT consensus algorithm can provide CBDC systems with good fault-tolerance and performance while it takes no privacy protection measurements.

Overall, the evaluation sub-framework can guide CBDC designers to consider related factors, analyze different combinations, and find a suitable consensus algorithm. However, we require a method to judge whether the combination meets the expectations. Therefore, we utilize the verification sub-framework to ensure proposed consensus algorithms are valid and satisfactory.

2) OPERATING ARCHITECTURE

We propose two new operating architectures based on current CBDC architectures (figure 1). The operating architecture determines how the network functions at a high level. A trade-off [6] exists between CBDC system features that we cannot achieve excellent performance, security, and privacy simultaneously. However, we show it is possible to update the operating architecture to compensate for weak CBDC system

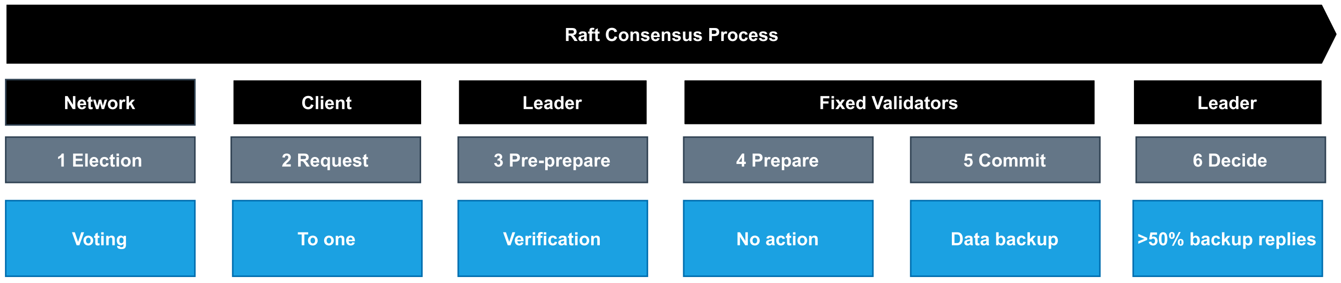

FIGURE 4. Consensus Process of RAFT starts from the voting and election timeout mechanism in the elections step. A client then sends a request to the leader in the network. The leader node then verifies the request in the pre-prepare step. After the leader's verification, the validators make a backup and respond to the leader node. This transaction will be valid if the leader receives more than half of the backup replies.

TABLE 2. Impact of RAFT on CBDC system features. The total row sums all values in the corresponding column with High (+1), Medium or TBA (0), and Low (−1).

Components		Performance			Security		Privacy	
		User Scalability	Network Scalability	Latency	Resilience	Cyber-Security	Customer Privacy	Business Secrecy
1 Leader Election	Voting	TBA	Medium	Medium	TBA	High	TBA	TBA
2 Request	To one	High	High	Low	TBA	TBA	Medium	TBA
3 Pre-Prepare	Verification	High	High	High	TBA	TBA	TBA	TBA
4 Prepare	No action	TBA	TBA	TBA	TBA	TBA	TBA	TBA
5 Commit	Data Backup	Medium	Medium	Medium	High	TBA	TBA	TBA
6 Decide	>50% Backup	Meidum	TBA	TBA	High	Medium	TBA	Medium
Validators	Fixed	TBA	Low	Low	High	High	TBA	TBA
Total		2	1	1	3	2	0	0
Average		**4/3**			**2.5**		**0**	

features. In the following example, we leverage our proposed architecture to improve business secrecy.

We conclude a three-tier CBDC architecture (Figure 5) to describe institutions that do not become tier-1 but want to provide CBDC services. The model describes how tier-1.5 institutions provide CBDC services to their customers in the current situation.

Tier-1.5 institutions have to provide transaction information to tier-1 institutions for bookkeeping because tier-1 institutions operate the ledgers. Once the tier-1 institution records the transaction in the ledger, the transaction becomes valid. However, tier-1.5 institutions will refuse to provide the customer data to tier-1 institutions because they are competitors subject to conflict of interest. Some commercial institutions even refuse to participate in CBDC-related services due to data privacy concerns.

To safeguard tier-1.5 institutions from the data monopoly of tier-1 institutions, we propose two operating architectures:

1) Use dynamic virtual addresses to keep the identities of participants secret from tier-1 institutions;
2) Use an independent operating organization that has no conflict of interest;

Figure 6 shows how tier-1.5 institutions can create virtual addresses for their customers in the tier-1 ledger. Privacy includes identity and transaction information. Virtual addresses ensure that tier-1 institutions have no access to the identity information of the payee and payer. Tier-1.5 institutions provide regulators with a mapping table between virtual addresses and real identities. Only regulators and

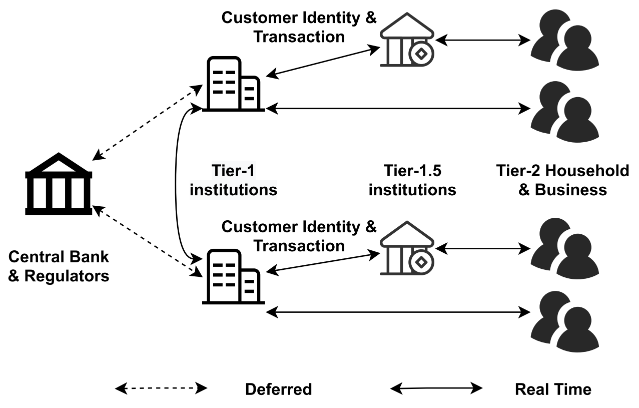
Customer Identity & Transaction
Central Bank & Regulators
Tier-1 institutions
Tier-1.5 institutions
Tier-2 Household & Business
Customer Identity & Transaction
Deferred
Real Time

FIGURE 5. Three-tier CBDC architecture.

tier-1.5 institutions will know the mapping relationship between virtual addresses and real customer identities. Tier-1.5 institutions only need to inform the regulators of the identity information. The regulators can then infer all transaction information by combining the identity mapping from tier-1.5 institutions and ledger transaction information from tier-1 institutions.

However, tier-1 institutions may be able to infer identity information by analyzing enough token flows and real-world events even though they only hold ledger transaction information. To further protect the business secrecy of tier-1.5 institutions, we can make virtual addresses dynamic such that tier-1.5 institutions create new virtual addresses to collect

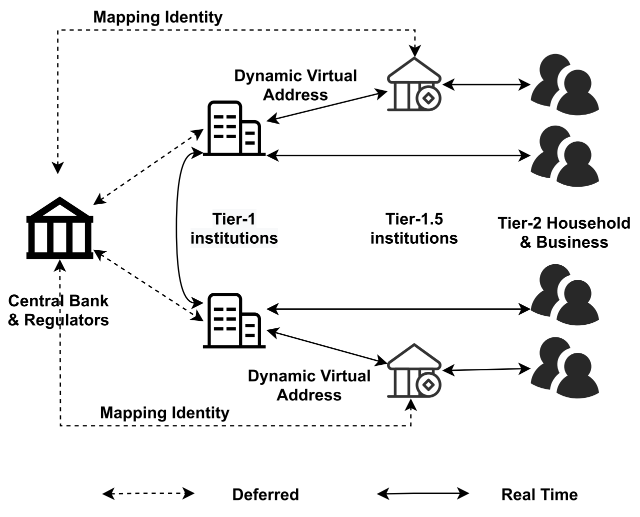

FIGURE 6. **Operating Architecture 1 leverage dynamic virtual address to avoid tier-1 institutions from knowing the real identities of customers.**

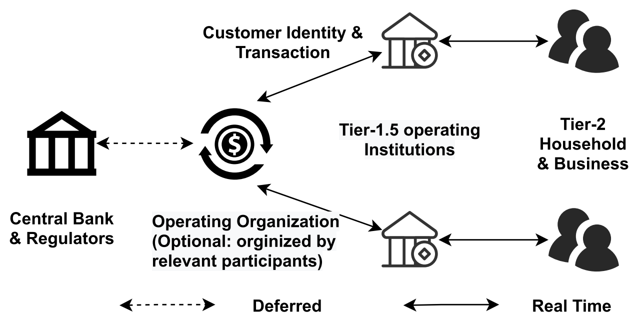
Customer Identity & Transaction
Tier-1.5 operating Institutions
Tier-2 Household & Business
Central Bank & Regulators
Operating Organization (Optional: orginized by relevant participants)
Deferred
Real Time

FIGURE 7. **Operating Architecture 2 builds an operating organization rather than tier-1 institutions to operate the CBDC system without interest conflict. The operating organization can know the transaction data of tier-1.5 institutions without business secrecy issues.**

change in transactions. This technical solution can prevent tier-1 institutions from accessing tier-1.5 institutions' customer data.

There are other methods to improve the operating architecture. For example, figure 7 presents a third-party organization in tier-1 to ensure no competition between tier-1 and tier-2. However, the operating organization needs a feasible business model to ensure enough money to support its stable operation.

B. VERIFICATION SUB-FRAMEWORK

Through the evaluation sub-framework, CBDC designers can develop a customized solution. The verification sub-framework works to ensure the proposed solution's feasibility and rationality.

The verification sub-framework contains diverse methods to verify proposed solutions. We divide these methods into two categories: empirical experiments and formal proof. For empirical experiments, we can simulate a real scenario and test models. For formal proof, we can build a mathematical model to infer related theories and find whether they meet initial expectations.

The verification sub-framework works in the following way:

1) Model proposed solutions for further verification. The verification sub-framework can guide CBDC designers to describe the proposed solution mathematically.
2) Conduct empirical experiments to examine performance.
3) Conduct formal proof to validate security and privacy.

IV. CBDC SCENARIO EXAMPLE

We present an example of using the CEV framework. We assume a country with a large population and well-developed communication technology. The CBDC designers focus on privacy and performance, especially network scalability, latency, and business secrecy. Finally, we leverage the CEV framework to propose a customized solution for this virtual country.

A. EVALUATION

We leverage the evaluation sub-framework to choose the operating architecture and propose consensus algorithms. In this example, we choose the virtual address operating architecture (figure 6). Then the evaluation sub-framework can propose consensus algorithms for different consensus networks.

Since the example emphasizes performance among the three CBDC system features, we leverage table 1 to determine the combinations with relatively high scores in performance for both wholesale and retail consensus networks.

The recommended wholesale network consensus algorithm follows figure 8, and the retail consensus networks follow figure 9. table 3 and table 4 are derived from table 1. Table 3 shows that the recommended wholesale network consensus algorithm has a good performance, especially network scalability, but the algorithm may bring a potential security issue. Similarly, table 4 shows that the retail network consensus algorithm has a good performance and a potential security issue. Both tables present that the system has relatively good privacy. The impact table only provides a rough analysis to CBDC designers, so we need to further verify the proposed solution in the verification sub-framework and make sure the proposed solutions are objective and reasonable.

Specifically, encryption in the third step protects business secrecy from validators because they only backup encrypted data for tampering with proof. Additionally, "To one / Sharding" in the "client-request" step can improve the system's performance by a token-based sharding method. Furthermore, "Data backup" can increase the resilience of CBDC systems. The impact table is derived from the verification sub-framework and works to recommend consensus algorithms in the evaluation sub-framework.

B. VERIFICATION

We use the verification sub-framework to verify the proposed solution by modelling the solution with empirical experiments and formal proof.

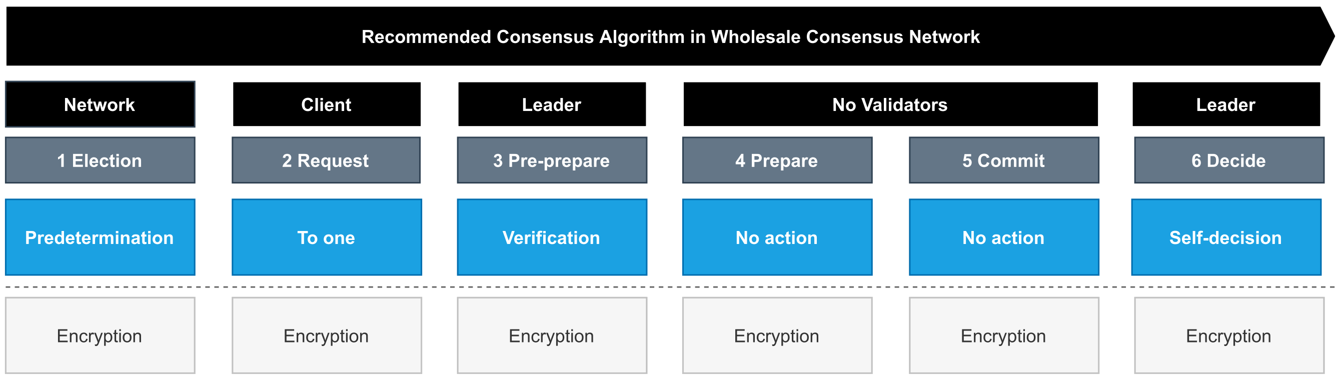

FIGURE 8. Recommended consensus algorithm in the wholesale consensus network with the following steps: 1. The wholesale consensus network predetermines the leader (central bank). 2. A client sends a cross-shard transaction request to its retail consensus network leader (tier-1 institution), and then the tier-1 institution forward it to the wholesale consensus network leader (central bank). 3. The central bank verifies the transaction and finishes the related issuance and redemption transactions in different retail consensus networks. Finally, the transaction is marked as successful.

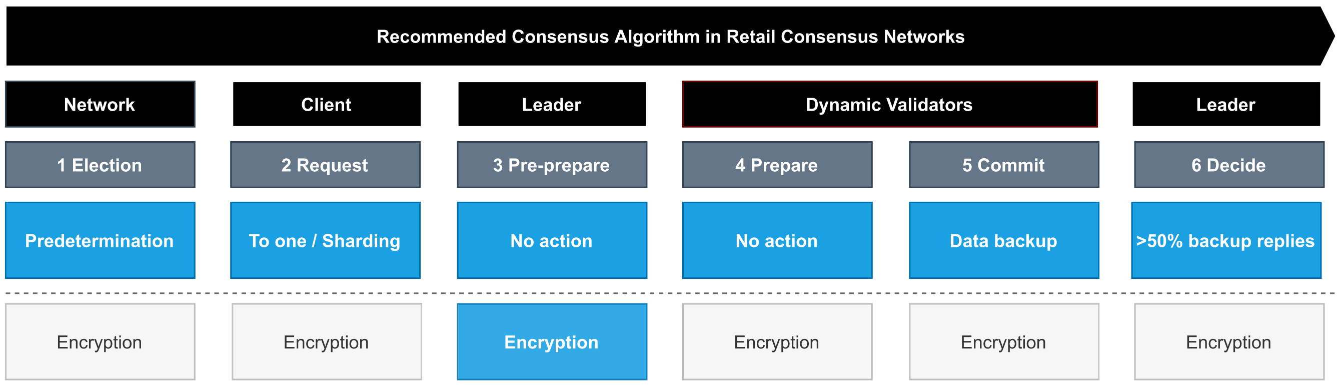

FIGURE 9. Recommended consensus algorithm in the retail consensus networks with the following steps: 1. The network predetermines the leaders (tier-1 institutions). 2. A client sends a transaction to its network leader. 3. The leader node encrypts the transaction before sending it to the validators in the network (possible competitors). 4. The network leverages a dynamic set of validators to encrypt backup data. 5. This transaction will be regarded as valid if the leader receives more than half of the backup replies.

TABLE 3. Impact of the recommended consensus algorithm in the wholesale consensus network. The total row sums all values in the table with High (+1), Medium or TBA (0), and Low (−1).

Components		Performance			Security		Privacy	
		User Scalability	Network Scalability	Latency	Resilience	Cyber-Security	Customer Privacy	Business Secrecy
1 Leader Election	Predetermination	TBA	High	High	TBA	TBA	TBA	High
2 Request	To one	High	High	Low	TBA	TBA	Medium	TBA
3 Pre-Prepare	Verification	High	High	High	TBA	TBA	TBA	TBA
4 Prepare	No action	TBA	TBA	TBA	TBA	TBA	TBA	TBA
5 Commit	No action	TBA	TBA	TBA	TBA	TBA	TBA	TBA
6 Decide	Self-decision	High	High	TBA	Low	Low	TBA	High
Validators	Fixed	TBA	Low	Low	High	High	TBA	TBA
Total		3	3	0	0	0	0	2
Average		**2**			**0**		**1**	

We first introduce sharding before our algorithm. Previous work [41]–[43] discusses account-based sharding and token-based sharding. We present them in a CBDC scenario with two types of transactions involved (shown in figure 10).

Account-based sharding divides users by accounts. In a two-tier CBDC architecture, tier-1 institutions have different wallets. For example, Tier-2 A sends transaction requests to Tier-1 A[6] when using CBDC because Tier-2 A created its account from Tier-1 A. As a result, a cross-shard transaction occurs if Tier-2 A transfers money to Tier-1 B's customer Tier-2 B. The cross-shard transaction needs the currency issuer (the central bank) to redeem a token in one ledger

[6]Tier-1 A is an example tier-1 institution.

TABLE 4. Impact of the recommended consensus algorithm in the retail consensus network. The total row sums all values in the table with High (+1), Medium or TBA (0), and Low (−1).

Components		Performance			Security		Privacy	
		User Scalability	Network Scalability	Latency	Resilience	Cyber-Security	Customer Privacy	Business Secrecy
1 Election	Predetermination	TBA	High	High	TBA	TBA	TBA	High
2 Request	To one	High	High	Low	TBA	TBA	Medium	TBA
3 Pre-prepare	No action	High	High	High	Medium	Low	Medium	Medium
	Encryption	TBA	TBA	Low	TBA	TBA	High	High
4 Prepare	No action	TBA	TBA	TBA	TBA	TBA	TBA	TBA
5 Commit	Data Backup	Medium	Medium	Medium	High	TBA	TBA	TBA
6 Decide	Self-decision	High	High	TBA	Low	Low	TBA	High
Validators	Dynamic	TBA	High	High	Medium	Medium	TBA	TBA
Total		3	5	1	0	-2	1	3
Average			**3**		**-1**		**2**	

TABLE 5. Table of notations.

Notation	Definition
x, y, z	time
xRy	x before y
τ	Token
V	Token Set
$p(\tau)$	The leader has recorded τ
$Tx(\tau, x)$	Transaction with input τ at time x
$H(\tau, x)$	τ has been spent before x
$F(\tau, x)$	τ can be spent after x
$r(\tau)$	Received token in a transaction using τ
$c(\tau)$	Change token in a transaction using τ
$f(\tau)$	Received token in a cross-shard transaction using τ
$d_G^+(\tau)$	The out-degree of vertex v
$d_G^+(\tau, x)$	The out-degree of vertex v at time x
H	Leaders execute transactions by state machine flow
EE	Existential Elimination
AE	And Elimination
AI	And Introduction
UE	Universal Elimination
UI	Universal Introduction
IE	Implication Elimination
II	Implication Introduction
BE	Biconditional Elimination
Ind	Induction
AS	Assumption

and issue a new token in another ledger, which significantly decreases the system's performance compared with single-shard transactions [43].

Token-based sharding divides users by tokens. For example, if Tier-1 A issues a token to Tier-2 A, Tier-2 A has to send their transaction requests to Tier-1 A to spend the token because Tier-1 A is the token operator. Tier-1 A can directly change the owner of the tokens, and this can avoid cross-shard transactions. Afterwards, Tier-2 B has to send a transaction request to Tier-1 A when spending the received token. Token-based sharding needs the central bank and operating institutions to provide a uniform interface to distribute transactions to the corresponding token operator.

The proposed solution uses token-based sharding in the retail consensus networks.

1) MODEL

Figure 12 shows a ledger state machine. The model provides a formal description of a finite state machine M = (S, V, t).

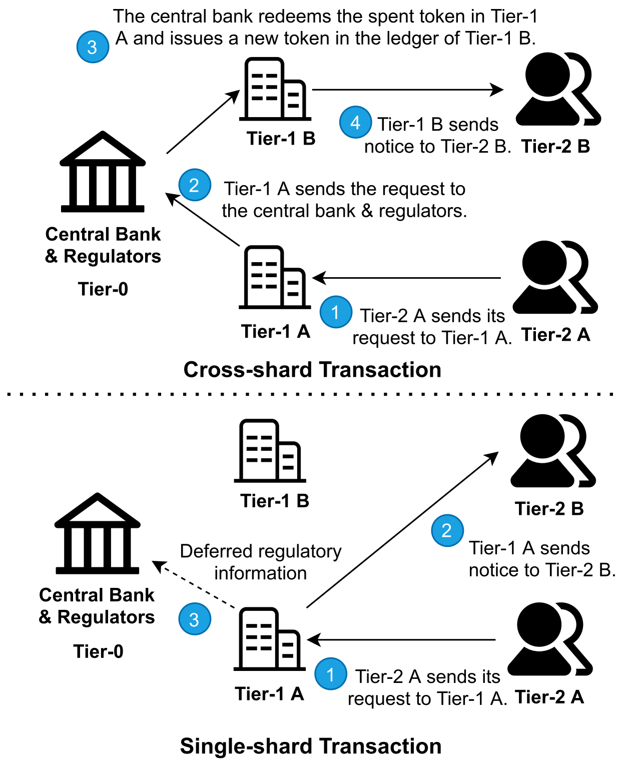
3 The central bank redeems the spent token in Tier-1 A and issues a new token in the ledger of Tier-1 B.
Tier-1 B
4 Tier-1 B sends notice to Tier-2 B.
Tier-2 B
2 Tier-1 A sends the request to the central bank & regulators.
Central Bank & Regulators
Tier-0
Tier-1 A
1 Tier-2 A sends its request to Tier-1 A.
Tier-2 A
Cross-shard Transaction
Tier-1 B
Tier-2 B
Deferred regulatory information
2 Tier-1 A sends notice to Tier-2 B.
Central Bank & Regulators
Tier-0
3
Tier-1 A
1 Tier-2 A sends its request to Tier-1 A.
Tier-2 A
Single-shard Transaction

FIGURE 10. Cross-shard transaction describes a transaction with two ledgers involved; single-shard transaction describes a transaction within one ledger.

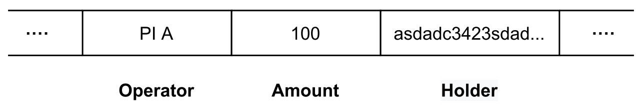

FIGURE 11. Holder is the identifier to determine which tier-1 institutions that tier-2 customers belong to.

The state machine can describe the overall status of the CBDC model $s \in S$. It can evolve to the next state by transition t(s, τ).

Figure 12 shows how a leader node records a token in the ledger before responding to the client. However, it only covers operations from the leader node and not from validator

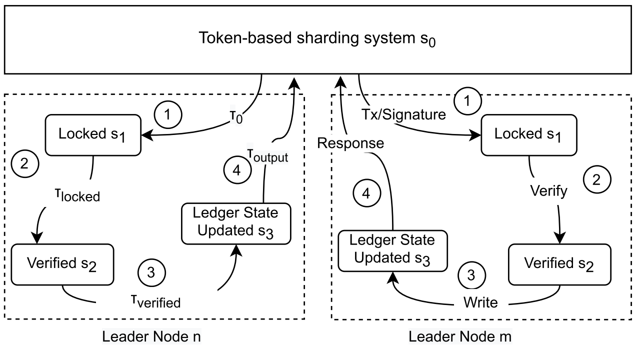

FIGURE 12. State Machine describes how the network executes a transaction, including token's states on the left side and operations on the right side: in the initial machine state s_0, a token-based sharding CBDC system distributes transactions to different sharded leaders (tier-1 institutions). Then the leader checks transaction signatures and input tokens τ_0. After the leader verifies the signature, the tokens become locked. If no leader has previously locked the tokens, the output becomes τ_{locked} and the machine moves to the locked state s_1. Otherwise, it would exit (a rolled-back transaction to the initial state). Next, the leader verifies whether τ_{locked} is available. If so, the output token becomes $\tau_{verified}$, and the machine moves to the verified state s_2. Otherwise, it would exit. In this step, the leader node can send the transaction to other validators to verify. Finally, the leader writes the transaction with inputs and outputs. If successful, the output would be τ_{output}, and the machine moves into the state s_3. Finally, the state machine moves back to s_0 and responds to a successful transaction.

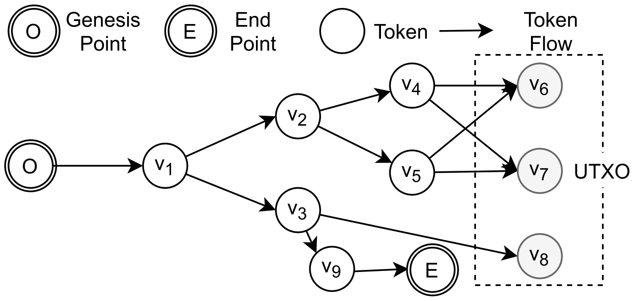

FIGURE 13. Ledger State. v are vertices (tokens), belonging to V, edge represents token flow. On the right side, UTXO includes all tokens without edge coming from.

nodes. The validators in the consensus algorithm undertake a data backup between state s_2 and s_3, which helps to reduce malicious behaviour. We discuss this further in Section-IV.4.

Figure 13 shows data model of a leader ledger. The transaction must be recorded in the ledger for it to be valid. Once a leader updates its ledger successfully, the transaction becomes legal and immutable.

Definition 1 (Ledger State): The ledger state of the CBDC is defined as a directed graph D=<V(D), E(D), φ> that the elements of V(D) are vertices (tokens) and the elements of E(D) are edges (token flow). φ is ordered mapping from token set V to token flow set E.

Definition 2 (UTXO): $UTXO = \{\tau | \tau \in V(D) \wedge d_G^+(\tau) = 0\}$. *UTXO is an unspent transaction output set. An unspent token means there is no edge coming from it (the out-degree of it is 0).*

Definition 3 (Transaction Graph): A transaction graph is a directed graph TD=<V(TD), E(TD), φ>. The in-degree of an input token and the out-degree of an output token are both 0 in any transaction graph. The leader updates the ledger when the state machine finishes a transaction ($\forall x.\ \forall \tau.(Tx(\tau,x) \Rightarrow D = D + TD)$).

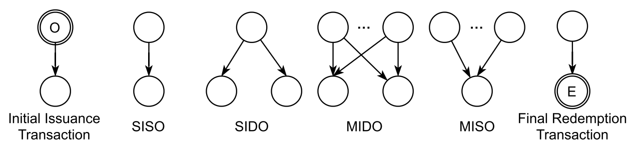

FIGURE 14. Different types of transaction graphs. SI refers to single-input. MI refers to multi-input. SO refers to single-output. DO refers to double-output.

Figure 14 shows all types of transaction graph. Only the currency issuer (Central Bank) can initiate the Initial Issuance Transaction, generating a new token from the genesis point. The issuer must also check the Final Redemption Transaction as the transaction receiver. Both types of transactions need the central bank to carry out real-time operations.

The following mathematical expressions present the token flows rather than the transactions. Here we add an assumption that the leader nodes are non-faulty (H) and follow the model procedures. Faulty nodes may behave arbitrarily and be vulnerable to inside and outside attacks. With non-faulty nodes, we can ensure leader nodes record tokens in the ledger in every transaction:

1) $\forall \tau.(H \wedge p(\tau) \Rightarrow p(r(\tau)) \wedge p(c(\tau)))$
2) $\forall \tau.(H \wedge p(\tau) \Rightarrow p(f(\tau)))$

$\forall \tau.(H \wedge p(\tau) \Rightarrow p(r(\tau))$ represents that if the ledger has recorded input τ, a non-faulty leader(H) always adds a valid transaction graph to the ledger with inputs of τ and outputs of the received token $r(\tau)$ and the change token $c(\tau)$. If the change is 0 ($c(\tau) = null$), p(c(τ)) means no token need to be recorded.

$\forall \tau.(H \wedge p(\tau) \Rightarrow p(f(\tau)))$ represents the central banks and non-faulty leaders(H) ensure that the receiving ledger records the output tokens in a cross-shard transaction (figure 15).

Cross-shard transactions impact the latency of the model. However, cross-shard transactions are required in some scenarios. For example, if CBDC users request to pay tokens in different ledgers, they first need to transfer them into one same ledger and then pay them to achieve an atomic transaction. In a CBDC system, the central bank is responsible for issuing and redeeming tokens, regulating both transactions and ensuring that the new shard records the output tokens.

2) PERFORMANCE

Empirical experiments test user scalability, network scalability, and latency. To ensure experiments are close to reality, we randomly initiate user transactions by different payment methods, including face-to-face transfer, collecting, etc.

We leverage AWS EC2 to deploy CBDC networks and carry out the empirical experiments shown in figure 16. Unfortunately, since the Corda open-source version has limitations on transaction volume, we could not demon-

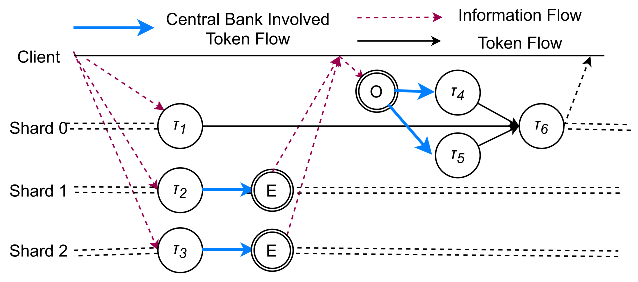

FIGURE 15. Cross-shard Transaction: a client has allocated different tokens in three different shards and used them to initiate a transaction. The transaction first turns the input tokens to the endpoint via the Final Redemption Transaction and issues new tokens in the new shard via the Initial Issuance Transaction.

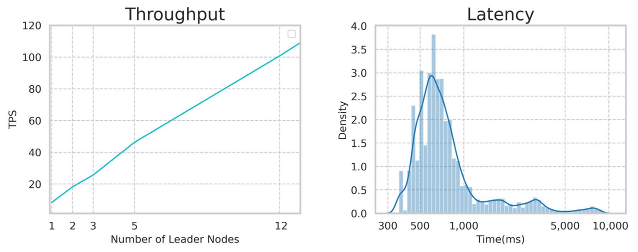

FIGURE 16. The left figure shows a linearly increasing TPS (Transaction per second); The right figure shows an acceptable level of latency for most of the transactions.

strate an extremely large transaction per second (TPS) in the experiment due to cost control. However, the experiment demonstrates performance improvement in a CBDC system.

Sharding improves network scalability and user scalability to a CBDC network. Commercial institutions, including tier-1 and tier-1.5, could become leader nodes or validator nodes and undertake customer due diligence in retail networks.

If CBDC is non-fungible, token-based sharding can map every non-fungible token to one leader node. Then different ledgers can circulate tokens efficiently in parallel, increasing the performance. However, CBDC is more like a fungible token, and most transactions produce a change. As a result, the token owner may use several small tokens, which causes additional concurrent transactions. Moreover, if the token owner pays two tokens that circulate in different ledgers simultaneously, it must first initiate a cross-shard transaction and then follow a single-shard transaction.

The recommended consensus algorithm uses sharding to improve performance by parallel running ledgers because cross-shard transactions being less frequent in the token-based sharding method than in the account-based method. However, more shards may cause reduced performance if each transaction in the retail consensus network requires verification from all parties in the wholesale network. Therefore, our proposed algorithm selects a dynamic set of validators that only need a backup, ensuring relatively high performance.

Overall, we prove that our proposed algorithms can increase the system's TPS while not sacrificing response speed (latency).

3) PRIVACY

We provide two updated operating architectures to protect business secrecy. For figure 7, we designate one operating organization to distribute CBDC because it is not a competitor with other operating institutions such that their data are safe from monopoly.

For figure 6, the dynamic virtual addresses can prevent tier-1 institutions from knowing tier-1.5 institutions' customer identity. The method is similar to the bitcoin schema. In the bitcoin [33] system, essential facts exist: 1) transactions generate new addresses to collect change; 2) users could have many addresses. Bitcoin uses this method to protect customer privacy from leaking to the public, while our model protects tier-1.5 institutions' data from leaking to tier-1 institutions.

However, like the bitcoin schema [47], tier-1 institutions can still obtain secret information, depending on the transaction types. Appendix B proved that a transaction may leak information depending on transaction types. Unlike SISO transactions, SIDO, MIDO, and MISO transactions may expose relationships between inputs and outputs.

Therefore, a more aggressive method to protect user privacy is required. We introduce virtual entities in this operating architecture. In the architecture, tier-1.5 institutions process transaction data before sending it to tier-1 institutions. tier-1.5 institutions can create virtual entities with virtual addresses in the network and use these virtual entities to create SISO transactions for customers. For example, in a SIDO transaction, tier-1.5 institutions can use virtual entities as the receiver to avoid the connection between payer and payee. Once the transaction is complete, the tier-1.5 institution can send the token to the real receiver via a SISO transaction. Other types of transactions can also apply to virtual entities. Enough virtual entities can help tier-1.5 institutions to hide the direct relationship between the payees and payers.

4) SECURITY

Appendix A proved that double-spending is possible in single-shard transactions and impossible in cross-shard transactions. Our proof assumes that the leaders are non-faulty. The assumption is reasonable in the wholesale consensus network, given that we would not expect the central bank to act maliciously. However, in the retail consensus network, tier-1 institutions are responsible for their ledger and decide each transaction on its own without validation. There is no mechanism to ensure that they do not act maliciously. As a result, double-spending may occur in single-shard transactions.

The design of CBDC is a trade-off [6] between different CBDC system features, including performance, security and privacy. In this example, one institution decides all the transactions, ensuring a high performance but sacrificing security.

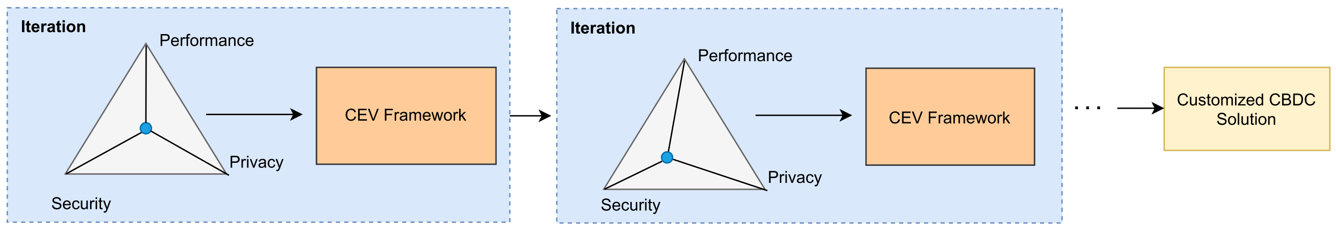

FIGURE 17. Iterations of the CEV Framework until an satisfactory trade-off.

Although we cannot avoid double-spending in real-time in the recommended consensus algorithms, we can increase the cost of malicious behaviour. For example, the recommended consensus algorithm in the retail consensus network chooses data backup from validators, and the leader sends encrypted transactions to validators. The validators will undertake a data backup. Once the leader node changes the original data, we can use the same encryption method to encrypt data and check data consistency. If the leader node behaves maliciously, it will be punished. Therefore, the leader nodes are discouraged from behaving maliciously because the validators are able to detect such behaviour easily. Furthermore, encryption can prevent the validators from accessing customer data, ensuring business secrecy in the CBDC system. In some cases, third-party auditors can run validator nodes, and it is not necessary to encrypt the data.

We can also ensure that extra verification will not overly influence latency because validators in the network are dynamic, and not all validators need to join the consensus process. Moreover, since the central bank controls issuance and redemption transactions, it knows the balance of money on each ledger so that no extra money is received from retail networks. Overall, the solution cannot avoid double-spending perfectly, but it uses extra mechanisms to reduce the possibility of malicious behaviour.

C. FRAMEWORK ITERATION

Figure 17 shows how we iterate the CEV framework. CBDC design involves many trade-offs [6] between different CBDC system features. We start from a country with a large population, focusing on performance and privacy. Then we use the evaluation sub-framework to propose solutions. Finally, we leverage the verification framework to measure performance, privacy, and security. The proposed solution presents an excellent performance and privacy, but double-spending is possible.

After verification, CBDC designers can return to CBDC system features to adjust the trade-off. For example, if they need to increase security, they can continually use the evaluation sub-framework to propose new solutions and use the verification sub-framework to verify them. In the experiment, if CBDC designers want to avoid double-spending perfectly, they can make validators vote for each transaction. However, it may potentially influence the system's performance. In this case, the CBDC designer can revisit the CBDC system features until finding a satisfactory trade-off. Our framework offers CBDC designers a flexible method to meet their CBDC requirements.

V. CONCLUSION

Our paper proposes a CBDC framework (CEV Framework), including an evaluation sub-framework and a verification sub-framework to design solutions for CBDC systems. Our work proposes an original approach and potentially promotes the evolution of CBDC. Furthermore, to the best of our knowledge, we are the first to propose a framework that offers a holistic solution for CBDC designers according to their economic and regulatory conditions.

Our paper analyzes CBDC technical solutions by splitting consensus algorithms into different components and improving the operating architectures to solve CBDC related issues. Most importantly, we build a verification sub-framework to prove the feasibility of the recommended algorithms and operating architectures with rigorous empirical experiments and formal proof. By using the CEV framework, central bank digital currency projects can better design the consensus algorithms and adopt reasonable operating architectures.

Future work could include considering more CBDC system features and solutions into the framework, updating the impact table for proposing solutions more accurately and improving the efficiency of the verification sub-framework. Finally, the CEV framework can also be used to propose stablecoin and other regulated cryptocurrency solutions.

.

APPENDIX A SECURITY PROOF

Here are some temporal logic proofs in the paper. Please see notations in Table 5 from Section IV-B.1. The premises below come from definitions in Section IV. The logic proofs [46] have been checked by the proof-editor from Stanford University [48].

Lemma 1: A non-issuance transaction changes the out-degrees of the input tokens to non-zero.

$$\forall x.\forall \tau.(Tx(\tau, x) \Rightarrow \forall y.(xRy \Rightarrow d_G^+(\tau, y)! = 0).$$

Proof: Since ($\forall$x. $\forall$ τ.(Tx(τ,x) $\Rightarrow$ D = D + TD)) in Definition 3, the leader adds a new transaction into the ledger where the out-degrees of input tokens keep same. A successful non-issuance transaction in definition 3 makes the out-degrees of input tokens become non-zero. □

Lemma 2: $\forall$x. $\forall$ τ.(Tx(τ,x) $\Rightarrow$ $\forall$y.(xRy $\Rightarrow$ H(τ, y)))

Proof: Lemma 1 shows $\forall$x. $\forall$ τ.(Tx(τ,x) $\Rightarrow$ $\forall$y.(xRy $\Rightarrow$ $d_G^+(\tau, y)! = 0)$.

1.	$\forall x.\forall\tau.(Tx(\tau,x) \Rightarrow \forall y.(xRy \Rightarrow d_G^+(\tau,y)! = 0))$	Premise
2.	$\forall y.\forall\tau.(H(\tau,y) \Leftrightarrow d_G^+(\tau,y)!{=}0)$	Premise
3.	$Tx(\tau,x) \Rightarrow \forall y.(xRy \Rightarrow d_G^+(\tau,y)!{=}0)$	UE: 1
4.	$H(\tau,y) \Leftrightarrow d_G^+(\tau,y)!{=}0$	UE: 2
5.	$Tx(\tau,x)$	AS
6.	$\forall y.(xRy \Rightarrow d_G^+(\tau,y)!{=}0)$	IE: 3,5
7.	$xRy \Rightarrow d_G^+(\tau,y)!{=}0$	UE: 6
8.	xRy	AS
9.	$d_G^+(\tau,y)!{=}0$	IE: 7,8
10.	$d_G^+(\tau,y)!{=}0 \Rightarrow H(\tau,y)$	BE: 4
11.	$H(\tau,y)$	IE: 9,10
12.	$xRy \Rightarrow H(\tau,y)$	II: 8,11
13.	$\forall y.(xRy \Rightarrow H(\tau,y))$	UI: 12
14.	$Tx(\tau,x) \Rightarrow \forall y.(xRy \Rightarrow H(\tau,y))$	II: 5,13
15.	$\forall x.\forall\tau.(Tx(\tau,x) \Rightarrow \forall y.(xRy \Rightarrow H(\tau,y)))$	UI: 14

□

Lemma 3:

$$\forall x.\forall\tau.(\mathrm{Tx}(\tau,x)) \Rightarrow \forall z.(zRx \Rightarrow F(\tau,z)))$$

Proof: In the model, one token τ keeps unspent status when it has not been involved in any transaction. For any $\tau \in V, F(\tau,y) \Leftrightarrow (\tau \in \mathrm{UTXO}$ at time y$) \Leftrightarrow d_G^+(\tau,y) = 0$. From the definition in transaction, we get $\forall$x. $\forall$ τ.(Tx(τ,x) $\Rightarrow$ $\forall$y.(yRx $\Rightarrow$ $d_G^+(\tau,y) = 0)$.

1.	$\forall x.\forall\tau.(Tx(\tau,x) \Rightarrow \forall y.(yRx \Rightarrow d_G^+(\tau,y) = 0))$	Premise
2.	$\forall y.\forall\tau.(F(\tau,y) \Leftrightarrow d_G^+(\tau,y){=}0)$	Premise
3.	$Tx(\tau,x) \Rightarrow \forall y.(yRx \Rightarrow d_G^+(\tau,y){=}0)$	UE: 1
4.	$F(\tau,y) \Leftrightarrow d_G^+(\tau,y){=}0$	UE: 2
5.	$Tx(\tau,x)$	AS
6.	$\forall y.(yRx \Rightarrow d_G^+(\tau,y){=}0)$	IE: 3,5
7.	$yRx \Rightarrow d_G^+(\tau,y){=}0$	UE: 6
8.	yRx	AS
9.	$d_G^+(\tau,y){=}0$	IE: 7,8
10.	$d_G^+(\tau,y){=}0 \Rightarrow F(\tau,y)$	BE: 4
11.	$F(\tau,y)$	IE: 9,10
12.	$yRx \Rightarrow F(\tau,y)$	II: 8,11
13.	$\forall y.(yRx \Rightarrow F(\tau,y))$	UI: 12
14.	$Tx(\tau,x) \Rightarrow \forall y.(yRx \Rightarrow F(\tau,y))$	II: 5,13
15.	$\forall x.\forall\tau.(Tx(\tau,x) \Rightarrow \forall y.(yRx \Rightarrow F(\tau,y)))$	UI: 14

□

Lemma 4:

$$\forall x.\forall\tau.(F(\tau,x) \Leftrightarrow \neg H(\tau,x))$$

Proof: $\neg H(\tau,x)$ means τ has not been spent before x. Therefore, F(τ,x) $\Leftrightarrow d_G^+(\tau,x)! = 0 \Leftrightarrow \neg H(\tau,x)$.

1.	$\forall x.\forall\tau.(F(\tau,x) \Leftrightarrow d_G^+(\tau,x)!{=}0)$	Premise
2.	$\forall x.\forall\tau.(H(\tau,x) \Leftrightarrow d_G^+(\tau,x)!{=}0)$	Premise
3.	$\forall\tau.(F(\tau,x) \Leftrightarrow d_G^+(\tau,x)!{=}0)$	UE:1
4.	$F(\tau,x) \Leftrightarrow d_G^+(\tau,x)!{=}0$	UE:3
5.	$\forall\tau.(H(\tau,x) \Leftrightarrow d_G^+(\tau,x)!{=}0)$	UE:2
6.	$H(\tau,x) \Leftrightarrow d_G^+(\tau,x)!{=}0$	UE:5
7.	$F(\tau,x) \Rightarrow d_G^+(\tau,x)!{=}0$	BE:4
8.	$d_G^+(\tau,x)!{=}0 \Rightarrow F(\tau,x)$	BE:4
9.	$H(\tau,x) \Rightarrow d_G^+(\tau,x)!{=}0$	BE:6
10.	$d_G^+(\tau,x)!{=}0 \Rightarrow H(\tau,x)$	BE:6
11.	$F(\tau,x)$	AS
12.	$d_G^+(\tau,x)!{=}0$	IE:7,11
13.	$H(\tau,x)$	IE: 10, 12
14.	$F(\tau,x) \Rightarrow H(\tau,x)$	II:11,13
15.	$H(\tau,x)$	AS
16.	$d_G^+(\tau,x)!{=}0$	IE:9,15
17.	$F(\tau,x)$	IE:8,16
18.	$H(\tau,x) \Rightarrow F(\tau,x)$	II:15,17
19.	$F(\tau,x) \Leftrightarrow H(\tau,x)$	BI:14,18
20.	$\forall\tau.(F(\tau,x) \Leftrightarrow H(\tau,x))$	UI:19
21.	$\forall x.\forall\tau.(F(\tau,x) \Leftrightarrow H(\tau,x))$	UI:20

□

Lemma 5:

$$\forall x.\forall y.(Tx(\tau_1,x) \wedge Tx(\tau_2,y) \wedge xRy \Rightarrow \tau_1 \neq \tau_2)$$

Proof: Time is continuous that there is always one timestamp existing between any two timestamps. We infer that a token cannot be spent twice in different timestamps by assuming a double-spending transaction possible as one premise and trying to find a contradiction with other premises.

1.	$\forall x.\forall\tau.(\mathrm{Tx}(\tau,x) \Rightarrow \forall y.(xRy \Rightarrow H(\tau,y)))$	Premise
2.	$\forall x.\forall\tau.(\mathrm{Tx}(\tau,x)) \Rightarrow \forall z.(zRx \Rightarrow F(\tau,z)))$	Premise
3.	$\forall x.\forall\tau.(F(\tau,x) \Leftrightarrow \neg H(\tau,x))$	Premise
4.	$\forall x\forall y.(xRy \Rightarrow (\exists z.(xRz \wedge zRy)))$	Premise
5.	$\exists x.\exists y.(xRy \wedge Tx(\tau,x), Tx(\tau,y))$	goal
6.	$\exists y.([x]Ry \wedge Tx(\tau,[x]), Tx(\tau,y))$	EE: 5
7.	$[x]R[y] \wedge Tx(\tau,[x]), Tx(\tau,[y])$	EE: 6
8.	$[x]R[y]$	AE: 7
9.	$Tx(\tau,[x])$	AE: 7
10.	$Tx(\tau,[y])$	AE: 7
11.	$\forall y.([x]Ry \Rightarrow (\exists z.([x]Rz \wedge zRy))$	UE: 4
12.	$[x]R[y] \Rightarrow (\exists z.([x]Rz \wedge zR[y]))$	UE: 11
13.	$\exists z.([x]Rz \wedge zR[y])$	IE: 8,12
14.	$[x]R[z] \wedge [z]R[y]$	EE: 13
15.	$[x]R[z]$	AE: 14
16.	$[z]R[y]$	AE: 14
17.	$\forall\tau.(\mathrm{Tx}(\tau,[x]) \Rightarrow \forall y.([x]Ry \Rightarrow H(\tau,y)))$	UE: 1
18.	$\forall\tau.(\mathrm{Tx}(\tau,[y]) \Rightarrow \forall z.(zR[y] \Rightarrow F(\tau,z)))$	UE: 2
19.	$\mathrm{Tx}(\tau,[x]) \Rightarrow \forall y.([x]Ry \Rightarrow H(\tau,y))$	UE: 17
20.	$\mathrm{Tx}(\tau,[y]) \Rightarrow \forall z.(zR[y] \Rightarrow F(\tau,z))$	UE: 18

21.	$\forall y.([x]Ry \Rightarrow H(\tau, y))$	IE:9, 19
22.	$\forall z.(zR[y] \Rightarrow F(\tau, z))$	IE:10, 20
23.	$[x]R[z] \Rightarrow H(\tau, [z])$	UE:21
24.	$[z]R[y] \Rightarrow F(\tau, [z])$	UE:22
25.	$H(\tau, [z]))$	IE:15, 23
26.	$F(\tau, [z]))$	IE:16, 24
27.	$\forall \tau.(F(\tau, [z]) \Leftrightarrow \neg H(\tau, [z])$	UE:3
28.	$F(\tau, [z]) \Leftrightarrow \neg H(\tau, [z])$	UE:27
29.	$F(\tau, [z]) \Rightarrow \neg H(\tau, [z])$	BE:28
30.	$\neg H(\tau, [z])$	IE:26, 29
31.	*Contradiction*	25, 30

With proof by contradiction above, we infer that a recorded token in the ledger cannot be spent twice in different transactions. □

Lemma 6: Assume a leader is non-faulty (H), double-spending will not occur in its shard.

Proof: In the ledger, the CBDC issuer creates and issues tokens (we use a to represent issued token) with the in-degree of 0. For any valid payment transaction, a non-faulty leader confirms that the received token and change token are recorded in the ledger.

1.	$p(a)$	Premise
2.	$\forall\tau.(H\wedge p(\tau)\Rightarrow p(r(\tau))\wedge p(c(\tau)))$	Premise
3.	$H\wedge p(\tau)\Rightarrow p(r(\tau))\wedge p(c(\tau))$	UE: 2
4.	H	AS
5.	$p(\tau)$	AS
6.	$H\wedge p(\tau)$	AI: 4,5
7.	$p(r(\tau))\wedge p(c(\tau))$	IE: 3,6
8.	$p(r(\tau))$	AE: 7
9.	$p(\tau)\Rightarrow p(r(\tau))$	II: 5,8
10.	$\forall\tau.(p(\tau)\Rightarrow p(r(\tau))$	UI: 9
11.	$H\Rightarrow(\forall\tau.(p(\tau)\Rightarrow p(r(\tau))))$	II: 4,10
12.	H	AS
13.	$p(\tau)$	AS
14.	$H\wedge p(\tau)$	AI: 12,13
15.	$p(r(\tau))\wedge p(c(\tau))$	IE: 3,12
16.	$p(c(\tau))$	AE: 15
17.	$p(\tau)\Rightarrow p(c(\tau))$	II: 13,16
18.	$\forall\tau.(p(\tau)\Rightarrow p(c(\tau))$	UI: 17
19.	$H\Rightarrow(\forall\tau.(p(\tau)\Rightarrow p(c(\tau))))$	II: 12,18
20.	H	AS
21.	$\forall\tau.(p(\tau)\Rightarrow p(r(\tau)))$	IE: 11,20
22.	$\forall\tau.(p(\tau)\Rightarrow p(c(\tau)))$	IE: 19,20
23.	$\forall\tau.(p(\tau))$	Ind: 1, 21, 22
24.	$H\Rightarrow(\forall\tau.p(\tau))$	II: 20,23

□

Lemma 7: If leaders are non-faulty(H), double-spending will not occur in cross-shard transactions.

Proof: The model shows that a non-faulty leader with the central bank ensures that the leader records new tokens in the ledger. Lemma 5 proves a recorded token has no double-spending problem. Since the leader node records the tokens in the ledger successfully, no double-spending exist in a cross-shard transaction. In our model, the currency issuer (Central Bank) secures the Initial Issuance Transaction and the Final Redemption Transaction and makes sure the token is recorded. □

Lemma 8: If leaders are non-faulty(H), double-spending will not occur in all transactions.

Proof: There are two kinds of transactions in the network: single-shard transactions and cross-shard transactions. Lemma 6 and 7 prove no double-spending with non-faulty leaders in these two kinds of transactions. Therefore, we conclude no double-spending problem in the network if leaders are non-faulty. □

APPENDIX B PRIVACY PROOF

Lemma 9: SISO transaction protects the identity relationship between payers and payees the most.

Proof: Here are all types of transactions and their privacy protection capability.

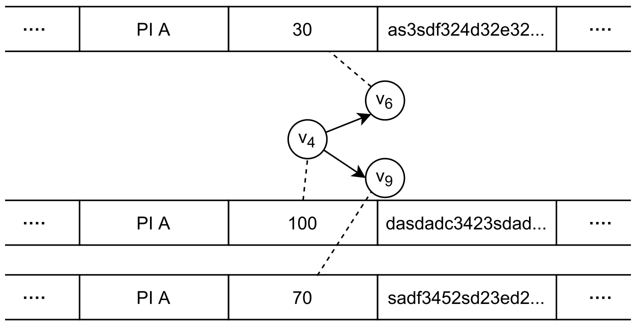

FIGURE 18. **One of v_6 and v_9 is the change money back to the owner of v_4. The relationship between payees and payers could be inferred when collecting enough extra data, like goods, transaction places.**

1) SIDO transaction: figure 18 shows a SIDO transaction, in which one of the output tokens should be the change token back to the payer.
2) MISO transaction: the payers pays several tokens from different virtual addresses. Then these virtual addresses possibly belong to one person.
3) MIDO transaction: the payers are usually the same person.
4) SISO transaction: the payees and payers usually are not the same people.

Overall, we concluded that SISO transaction can protect identity privacy the most. □

ACKNOWLEDGMENT

This article is partially based on their work in the Global CBDC Challenge [19], in which they were shortlisted as finalists. Based on the framework in this paper, they built an evaluation and recommendation platform that advocates consensus algorithms and operating architectures for different CBDC designers. This platform can satisfy different national economic and regulatory conditions. The authors wish to acknowledge the other two teammates, Mark Liu and Bing Qu. They also gratefully acknowledge the help of Bo Tong Xu in fruitful discussions. This paper is reviewed by Dr. Philip Intallura, Dr. Bing Zhu, and Dr. Ziyuan Li. They also wish to express great appreciation for their valuable input.

REFERENCES

[1] T. Adrian and T. M. Griffoli, *The Rise of Digital Money*. Washington, DC, USA: IMF, 2019. [Online]. Available: https://www.imf.org/en/Publications/fintech-notes/Issues/2019/07/12/The-Rise-of-Digital-Money-47097

[2] E. A. Opare and K. Kim, "A compendium of practices for central bank digital currencies for multinational financial infrastructures," *IEEE Access*, vol. 8, pp. 110810–110847, 2020, doi: 10.1109/ACCESS.2020.3001970.

[3] R. Auer, G. Cornelli, and J. Frost, "Rise of the central bank digital currencies: Drivers, approaches and technologies," Social Sci. Res. Netw., Rochester, NY, USA, BIS Working Papers 880, Oct. 2020. [Online]. Available: https://papers.ssrn.com/abstract=3723552

[4] B. Singh and S. Kumar, "Permission blockchain network based central bank digital currency," in *Proc. IEEE 4th Int. Conf. Comput., Power Commun. Technol. (GUCON)*, Sep. 2021, pp. 1–6, doi: 10.1109/GUCON50781.2021.9574020.

[5] N. Dashkevich, S. Counsell, and G. Destefanis, "Blockchain application for central banks: A systematic mapping study," *IEEE Access*, vol. 8, pp. 139918–139952, 2020, doi: 10.1109/ACCESS.2020.3012295.

[6] Group of Central Banks. (Oct. 2020). *Central Bank Digital Currencies: Foundational Principles and Core Features*. [Online]. Available: https://www.bis.org/publ/othp33.htm

[7] R. Auer and R. Boehme. (Jun. 2021). *Central Bank Digital Currency: The Quest for Minimally Invasive Technology*," [Online]. Available: https://www.bis.org/publ/work948.htm

[8] Monetary Authority of Singapore. (2017). *Project Ubin: Central Bank Digital Money Using Distributed Ledger Technology*. [Online]. Available: https://www.mas.gov.sg/-/media/MAS/ProjectUbin/Project-Ubin–SGD-on-Distributed-Ledger.pdf

[9] J. Chapman, "Project Jasper: Are distributed wholesale payment systems feasible yet," *Financial Syst.*, vol. 59, pp. 4–6, Jun. 2017.

[10] Y. Qian, "Experimental study on prototype system of central bank digital currency," (in Chinese), *J. Softw.*, vol. 29, no. 9, pp. 2716–2732, 2018.

[11] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 168:1–168:41, 2021, doi: 10.1145/3471140.

[12] X. Han, Y. Yuan, and F.-Y. Wang, "A blockchain-based framework for central bank digital currency," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Informat. (SOLI)*, Nov. 2019, pp. 263–268, doi: 10.1109/SOLI48380.2019.8955032.

[13] S. Allen, S. Capkun, I. Eyal, G. Fanti, B. A. Ford, J. Grimmelmann, A. Juels, K. Kostiainen, S. Meiklejohn, A. Miller, E. Prasad, K. Wüst, and F. Zhang, "Design choices for central bank digital currency: Policy and technical considerations," Nat. Bur. Econ. Res., Cambridge, MA, USA, BIS Working Papers 27634, Aug. 2020, doi: 10.3386/w27634.

[14] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 305–319. [Online]. Available: https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro

[15] S. Darbha and R. Arora. (Jun. 19, 2020). *Privacy in CBDC Technology*. [Online]. Available: https://www.bankofcanada.ca/2020/06/staff-analytical-note-2020-9/

[16] (2020). *Group of Thirty, Digital Currencies and Stablecoins: Risks, Opportunities, and Challenges Ahead*. [Online]. Available: https://group30.org/publications/detail/4761

[17] V. Sethaput and S. Innet, "Blockchain application for central bank digital currencies (CBDC)," in *Proc. 3rd Int. Conf. Blockchain Comput. Appl. (BCCA)*, Nov. 2021, pp. 3–10, doi: 10.1109/BCCA53669.2021.9657012.

[18] *Inthanon–Lionrock Leveraging Distributed Ledger Technology to Increase Efficiency in Cross-Border Payments*, Hong Kong Monetary Authority, Bank Thailand, Bangkok, Thailand, Jan. 2020.

[19] Monetary Authority of Singapore. (2021). *Global CBDC Challenge Problem Statements*. [Online]. Available: https://hackolosseum.apixplatform.com/hackathon/globalcbdcchallenge

[20] C. Boar and A. Wehrli. (Jan. 2021). *Ready, Steady, Go?—Results of the Third BIS Survey on Central Bank Digital Currency*. [Online]. Available: https://www.bis.org/publ/bppdf/bispap114.htm

[21] T. M. Griffoli, M. S. M. Peria, I. Agur, A. Ari, J. Kiff, A. Popescu, and C. Rochon, "*Casting Light on Central Bank Digital Currencies*. Washington, DC, USA: IMF. Accessed: Feb. 18, 2022. [Online]. Available: https://www.imf.org/en/Publications/Staff-Discussion-Notes/Issues/2018/11/13/Casting-Light-on-Central-Bank-Digital-Currencies-46233

[22] C. Boar, H. Holden, and A. Wadsworth. (Jan. 2020). *Impending Arrival—A Sequel to the Survey on Central Bank Digital Currency*. [Online]. Available: https://www.bis.org/publ/bppdf/bispap107.htm

[23] *Delivery Versus Payment on Distributed Ledger Technologies: Project Ubin*, Deloitte, Monetary Authority Singapore, Singapore Exchange, Singapore, Aug. 2018.

[24] J. Fernández-Villaverde, D. Sanches, L. Schilling, and H. Uhlig, "Central bank digital currency: Central banking for all?" *Rev. Econ. Dyn.*, vol. 41, pp. 225–242, Dec. 2020, doi: 10.1016/j.red.2020.12.004.

[25] P. Wierts, H. Boven, and I. van de Wiel, "Central bank digital currency: Objectives, preconditions and design choices," De Nederlandsche Bank, Amsterdam, The Netherlands, Tech. Rep., Apr. 2020. [Online]. Available: https://ideas.repec.org/p/dnb/dnbocs/20-01.html

[26] I. Agur, A. Ari, and G. Dell'Ariccia, "Designing central bank digital currencies," *J. Monetary Econ.*, vol. 125, pp. 62–79, Jan. 2022, doi: 10.1016/j.jmoneco.2021.05.002.

[27] R. Auer and R. Boehme. (Mar. 2020). *The Technology of Retail Central Bank Digital Currency*. [Online]. Available: https://www.bis.org/publ/qtrpdf/r_qt2003j.htm

[28] J. K. Zhou J. Alwazir, S. Davidovic, A. Farias, A. Khan, T. Khiaonarong, M. Malaika, H. K. Monroe, N. Sugimoto, H. Tourpe, and P. Zhou, *A Survey of Research on Retail Central Bank Digital Currency*. Washington, DC, USA: IMF, 2020. [Online]. Available: https://www.imf.org/en/Publications/WP/Issues/2020/06/26/A-Survey-of-Research-on-Retail-Central-Bank-Digital-Currency-49517

[29] A. Lannquist, S. Warren, and R. Samans, "Central bank digital currency policy-maker toolkit," World Econ. Forum, Geneva, Switzerland, Insight Rep., 2020. [Online]. Available: https://www3.weforum.org/docs/WEF_CBDC_Policymaker_Toolkit.pdf

[30] J. Zhang, R. Tian, Y. Cao, X. Yuan, Z. Yu, X. Yan, and X. Zhang, "A hybrid model for central bank digital currency based on blockchain," *IEEE Access*, vol. 9, pp. 53589–53601, 2021, doi: 10.1109/ACCESS.2021.3071033.

[31] G. Calle and D. Eidan. (Apr. 2020). Central bank digital currency: An innovation in payments. R3. [Online]. Available: https://www.r3.com/wp-content/uploads/2020/04/r3_CBDC_report.pdf

[32] A. D. Kshemkalyani and M. Singhal, *Distributed Computing: Principles, Algorithms, and Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[33] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, p. 21260, Oct. 2008. [Online]. Available: https://www.debr.io/article/21260.pdf

[34] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, 1999, vol. 99, no. 1999, pp. 173–186.

[35] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Proc. Workshop Distrib. Cryptocurrencies Consensus Ledgers*, 2016, vol. 310, no. 4, p. 5.

[36] R. G. Brown, *Corda: An Introduction*. New York, NY, USA: R3 CEV, August vol. 1, no. 2016, p. 15.

[37] I. Eyal, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. 13th USENIX Symp. Networked Syst. Design Implement. (NSDI)*, 2016, pp. 45–59.

[38] J. Poon and T. Dryja, "The Bitcoin lightning network: Scalable off-chain instant payments," White Paper, 2016. [Online]. Available: https://lightning.network/lightning-network-paper.pdf

[39] R. Auer, P. Haene, and H. Holden, "Multi-CBDC arrangements and the future of cross-border payments," BIS Paper 115, 2021.

[40] S. Riksbank, "The Riksbank to test technical solution for the e-krona," Sveriges Riksbank, Stockholm, Sweden, Feb. 2020.

[41] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 583–598, doi: 10.1109/SP.2018.000-5.

[42] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Toronto, ON, Canada, Oct. 2018, pp. 931–948, doi: 10.1145/3243734.3243853.

[43] M. Król, O. Ascigil, S. Rene, A. Sonnino, M. Al-Bassam, and E. Rivière, "Shard scheduler: Object placement and migration in sharded account-based blockchains," in *Proc. 3rd ACM Conf. Adv. Financial Technol.* New York, NY, USA: Association for Computing Machinery, Sep. 2021, pp. 43–56, doi: 10.1145/3479722.3480989.

[44] Y.-T. Lin. (Aug. 2017). *Istanbul Byzantine Fault Tolerance*. [Online]. Available: https://github.com/ethereum/EIPs/issues/650

[45] G. Wood, "Ethereum: A secure decentralized generalized transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[46] N. Rescher and A. Urquhart, *Temporal Logic*. vol. 3. Springer, 2012. [Online]. Available: https://books.google.com.hk/books?hl=zh-TW&lr=&id=ha7wCAAAQBAJ&oi=fnd&pg=PR11&dq=Temporal+logic&ots=P67r1pUQ47&sig=g15hB8gTPQD5sRajJ_h9kTVmdqQ&redir_esc=y#v=onepage&q=Temporal%20logic&f=false

[47] E. Androulaki, "Evaluating user privacy in Bitcoin," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2013, pp. 34–51.

[48] Standford University. (2021). *Logica Formal Logic Proof*. [Online]. Available: http://logica.stanford.edu/logica/homepage/index.php

SI YUAN JIN received the B.S. degree in financial technology from the South China University of Technology, in 2021. He is currently a Full Stack Engineer at HSBC Laboratory. His work focuses on central bank digital currency and quantum computing research. His research interests include central bank digital currency, blockchain, consensus algorithm, and quantum computing. He focuses on state-of-art technologies and enhances their potential applications in financial areas.

YONG XIA received the Ph.D. degree from the University of Zurich. He is currently the Global Head of HSBC Laboratory. He is also the Board Director of Corda Network Foundation, one of the five Council Members at the International Requirements Engineering Board and a Visiting Professor at the South China University of Technology. Before that, he was a member of the IBM Academy of Technology and a Senior Certified Consultant (at the Thought Leader level) at IBM, a Visiting Professor at Fudan University, an Enterprise Ph.D. Supervisor at Shanghai Jiao Tong University, and a Lead Architect at Credit Suisse. He is well known for many of his publications and international thought leader level consulting activities in the areas of cognitive computing, machine learning, deep learning, Fintech, blockchain, CBDC, distributed computing, and requirements engineering.