Article | Received 15 December 2024; Accepted 26 March 2025; Published 18 April 2025
https://doi.org/10.55092/blockchain20250007

A hybrid blockchain overlay for secure and compliant document management and tokenisation in public and enterprise systems

Benjamin Brooks[1,*], Luigi Lunardon[2] and Alessio Pagani[3]

[1] Emerging Technology Team, Teranode Group, London, United Kingdom
[2] Emerging Technology Team, Teranode Group, Zug, Switzerland
[3] Dept of Mathematical Sciences, University of Bath, Bath, United Kingdom

* Correspondence author; E-mail: b.brooks@teranode.group

Highlights:

- Hybrid overlay network for secure and transparent data management.
- Interoperable system enabling document sharing and digital transactions.

Abstract: this work presents SOvNet, a blockchain-agnostic hybrid solution that combines the privacy of proprietary systems with the resilience of public blockchains. SOvNet is a private network that publishes cryptographic fingerprints of its data on public blockchains, while aiming to simplify the management of personal data and transactions. The network is maintained by a group of nodes that updates the system status and provides cryptographic proofs of data processing to the network users. Designed for governments and enterprises, SOvNet supports services such as document management, real-world asset digitalisation, and digital payments. It also supports communication between SOvNet instances with minimal user-side overhead. The user interface to interact with SOvNet implements only lightweight operations, allowing for deployment on personal devices, including laptops and smartphones.

Keywords: blockchain; blockchain interoperability; blockchain services; Central Bank Digital Currencies (CBDCs); decentralised networks; digital identities; document management; overlay network; real-world assets; stablecoins; tokenisation; Web3.

1. Introduction

The secure design of blockchain is being recognised by government as a valuable asset to advance digitalisation without compromising security [1–3]. Enterprises share aligned interests and requirements with governments when it comes to potential blockchain technology adoption [4, 5]. Several attempts have been made to combine blockchain technology with sovereign systems. Efforts to integrate blockchain into governmental infrastructure initially focused on introducing digital currencies such as central bank digital currencies (CBDCs) and stablecoins [6, 7]. Research then expanded to other domains, including healthcare, agriculture, education, voting systems, and climate change research [8–15]. In some notable cases, experimentation has progressed beyond the prototyping stage, with government blockchain-based digital management projects in China, Australia, and the United Arab Emirates [16–18]. Corporate usage of blockchain focuses more on streamlining financial operations (e.g. payment

processing and cross-border transactions) and supply chain management [19–22]. Companies like IBM and JPMorgan explored using enterprise blockchains such as Hyperledger and Quorum [23, 24].

Balancing the benefits of blockchain technology with the need to safeguard data confidentiality remains a core challenge, and concerns about losing control over sensitive data persist [25–28]. Widespread adoption raises concerns about scalability, interoperability, and regulatory compliance. Additionally, integrating blockchain solutions into legacy systems while ensuring compliance with data protection laws, such as GDPR, presents significant challenges [29, 30]. Uncertainties regarding the volatility of the blockchain market and their environmental impact have also created concerns on the use of this technology.

In this paper, we introduce SOvNet, a private overlay network built upon a public blockchain. SOvNet combines the strengths of a private system running on private servers, with the immutability and auditability provided by blockchain technology. Publishing data on a public blockchain builds trust with users, while processing information on private servers maintains data confidentiality and minimises blockchain fees. Unlike other solutions leveraging blockchain technology, SOvNet does not require users to interact directly with the blockchain for day-to-day operations. Network maintainers and users can still benefit from the public blockchain to timestamp data, provide proof of processing, and resolve disputes.

Since data are managed on private servers, SOvNet architecture can be adapted to ensure compliance on local privacy regulations. The system supports both full and selective data disclosure to third parties by only publishing hash representations of data and signatures to ensure authenticity [31]. Methods like BlockShare support selective disclosure using zero-knowledge proofs [32], but proof size and generation time can be costly.

SOvNet integrates concepts from both private and public blockchain-based solutions to offer a hybrid approach. Existing solutions such as Blockcerts and uPort ID store private data on public blockchains and require users to interact with such blockchains [33, 34]. Public blockchain systems have the benefit of direct data query, using services such as VQL [35], but introduce security risks when publishing proprietary data. Private ledger-based systems such as TradeLens, ChromaWay Land Registration, and Exonum require private blockchains such as Hyperledger and R3 Corda for data management, lacking transparency [23, 36–40]. SOvNet builds on these systems by taking a blockchain-agnostic approach that does not depend on a private system. SOvNet combines the transparency and resilience of public blockchains with the security of private networks. It also further develops the concept of interoperability between multiple blockchains seen in Blockcerts and the Mastercard MTN network by enabling communication between independent SOvNets [41]. Figure 1 compares SOvNet with other blockchain services for enterprises and governments, highlighting its focus on document sharing, real-world asset digitisation, and digital payments.

The paper is organised as follows: Section 2 provides essential background. Section 3 describes the design of SOvNet, detailing the roles and responsibilities of the different components. Section 4 focuses on the data in SOvNet and their blockchain fingerprints, giving a formal definition, describing their life cycle and outlining the verification process. Section 5 describes a communication protocol between independent SOvNets. Finally, Section 6 provides a performance and security analysis of the system.

2. Background

2.1. *Blockchain*

Blockchain technology emerged as a digital solution to enable peer-to-peer payments [42]. A blockchain is a distributed ledger that stores transactions. Transactions are stored in blocks linked by cryptographic hashes. A network of nodes manages the blockchain ledger, and

	SOvNet	TradeLens Hyperledger [36]	ChromaWay Land Reg. [37]	Exonum Land Reg. [38]	Blockcerts [33]	uPort ID [34]	Mastercard MTN [41]
Does not require a private blockchain	✓	✗	✗	✗	✓	✓	✓
Is a blockchain's overlay network	✓	✗	✗	✓	✗	✗	✓
Private data are not on public blockchains	✓	✓	✓	✓	✗	✗	✓
Support multiple document types	✓	✓	✗	✗	✓	✗	✓
Supports digital payment solutions	✓	✓	✗	✗	✗	✗	✓
Supports RWAs digitalisation	✓	✓	✗	✗	✗	✗	✓
Users do not interact with the blockchain	✓	✗	✓	✓	✗	✗	✓
Interoperable and migratable	✓	✗	✗	✗	✓	✓	✓
Supports selective disclosure	✓	✓	✓	✓	✓	✓	✗

Figure 1. A comparison between SOvNet and other well-established blockchain-based solutions for document management and digital money. TradeLens is not the only project based on Hyperledger, but all such projects share the same profile for the purpose of our analysis.

the validity of a block is determined using a consensus algorithm [43, 44]. UTXO-based blockchains, such as Bitcoin, are a type of blockchain whose atomic structural component is a UTXO (Unspent Transaction Output). A UTXO consists of an amount and a puzzle (the locking script). A UTXO is spent by providing a solution to the puzzle (the unlocking script), usually a digital signature. Transactions consume a set of UTXOs to generate a new group of UTXOs. Merkle trees enable efficient inclusion proofs often called Simplified Payment Verification (SPV) proofs [45]. Our implementation is for UTXO-based blockchains but can also be adapted to account-based blockchains.

2.2. Overlay network

An overlay network is a network built atop an existing network. In this paper, the base network is always a blockchain network. Often called layer 2 networks, overlay networks provide enhanced scalability, reduced costs, privacy, and control of data. A common technique to links overlay networks and underlying blockchains by publishing overlay data fingerprints on the blockchain. This ensures data integrity and timestamps data in the overlay network.

2.3. Public key infrastructure

A Public Key Infrastructure (PKI) is a combination of policies and procedures for digital certificate management [46, 47]. PKI binds public keys to user identities. Certificates are issued by a Certificate Authority (CA) and revoked by a Revocation Authority (RA). The CA can rely on a Registration Authority (RegA) to validate identities before issuing certificates.

PKI certificates can be managed using a blockchain [48]. The CA issues the certificate by signing a transaction that stores the public key of the user in an output. The certificate is valid if the transaction has been accepted by miners, and the RA is an owner of the UTXO. The RA revokes certificates by spending corresponding UTXOs. Figure 2 summarises this process.

2.4. Digital money

Digital money includes CBDCs and stablecoins. CBDCs are issued by central banks and integrated into national monetary systems. CBDCs require compliance with local laws and are not supported by all legislation [49]. Stablecoins are cryptocurrencies issued by private entities. While regulated, they operate outside official monetary systems. Stabilisation is

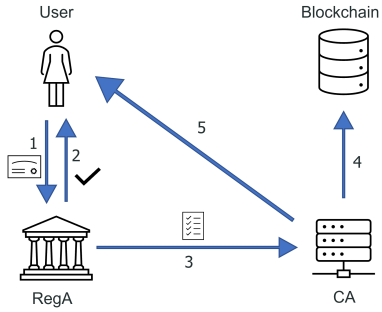

Figure 2. Issuance of user PKI certificates following the blockchain-based protocol described in [48]. The user sends the identification documents and a public key that they control to RegA (1). RegA checks the validity of the documents (2) and requests the CA to certify the public key (3). CA generates a blockchain transaction representing the PKI certificate (4) and shares the certificate and its SPV proof with the user (5).

achieved using collateral assets [50].

3. SOvNet overview

SOvNet is an overlay network maintained by a central authority and built atop a public blockchain, bridging the privacy offered by proprietary servers with the immutability provided by public blockchain networks. Data in SOvNet are mirrored by blockchain fingerprints, but the link is decoupled using cryptographic hashes. This prevents malicious observers from extracting confidential information from blockchain data.

Data circulates within SOvNet in the form of structured records containing users' personal information. These data records represent documents, tokenised assets, and digital payments. Sovereign nodes (SNs) form the infrastructure that supports SOvNet. SNs are servers maintaining a distributed database of processed data. Their functions include managing user registration, validating user identities, processing and storing data records, and publishing fingerprints of these data records on the blockchain through fingerprint transactions (see Section 4.2 for more details). SNs can specialise to provide a single function, such as acting as RegA, CA or RA in the user registration process. An overview of SOvNet's architecture is provided in Figure 3.

Users and enterprise systems generate data records. When registering, users are issued a PKI certificate by the CA of their SOvNet . User registration is described in Figure 2. Depending on the type of data record, users may require different levels of authorisation for creation. After sending a data record to the SNs, users receive a proof of acceptance. Users can share accepted data records with other users or governmental bodies at any time, with the proof of acceptance providing evidence of data validity and non-repudiation.

SNs can process data records even when momentarily disconnected from the blockchain network. The only inconvenience in such situations would be a delay in the creation of fingerprint transactions. SNs can also voluntarily delay the creation of fingerprint transactions if blockchain transaction fees increase due to temporary network congestion. SOvNet supports multiple blockchains, with the ability to switch between them. This requires porting data, but preserves the validity of existing fingerprint transactions published on the previous blockchain.

3.1. SOvNet servers

SNs are SOvNet servers that process, validate and store data records, providing users with verifiable proofs of data record processing and acceptance. SNs manage blockchain wallets to generate and fund blockchain fingerprint transactions.

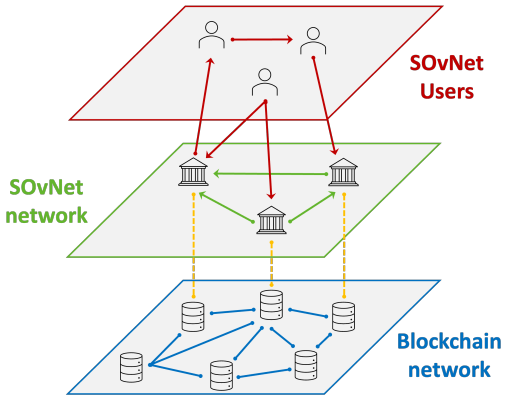

Figure 3. Overview of SOvNet's core structure. Users interact with both each other and sovereign nodes (SNs). These tightly interconnected servers publish user data fingerprints by generating transactions which are published on the selected public blockchain.

SNs store data in an internal distributed database with controlled access and redundancy. Storage policies can be updated to comply with local privacy regulations, such as GDPR, defining retention periods, access permissions, and data minimization rules. Depending on their role in SOvNet, SNs may have different read and write privileges. Stored data can be plaintext, encrypted, or hashed and may include the data record, its fingerprint transaction, and the acceptance proof.

Depending on their privileges, SNs offer different services. If SNs store full data records and supporting documents, they can provide processing and retrieval services. If SNs store fingerprints and proofs of acceptance, they can provide proof of validity services by answering queries on data records' legitimacy. If SNs opt-out of permanent storage, their role is limited to network maintainers: they process data records, but lack data verification and access capability.

3.2. Data records

Data records contain a combination of public and private information, and signatures from the involved parties. The way information is structured within the data record depends on the type of underlying data. Fingerprint transactions are blockchain transactions containing a fingerprint of the data record. At a high level, data records comprise four components (see Section 4.1 for more details):

- Template Flag: Identifies the template, defining its content, function, and parsing method.
- Salt Value: Enhances privacy by adding a layer of randomness to the data record fingerprint.
- Core Data: Contains the relevant private information, formatted according to the template.
- Signature: Provide data integrity and links the data record to the sender's identity.

3.2.1. Data record properties

Data records in SOvNet inherit some security properties from their link with blockchain transactions.

- **Verifiability:** data records include digital signatures from all entities involved in its creation, serving as proof of their participation in the generation process. The public keys used are registered and validated by SNs when they process the data record. Data records' fingerprints can be cross-referenced with blockchain records to guarantee that the packet remains unchanged. SNs generate proofs of acceptance for each data record, confirming their legitimacy.

Type of data record	Core data	
Driving license renewal request	Citizen's personal data	Old driving license
	Department of Transportation's certified public key	
Driving license	Merkle root	Department of Transportation's certified public key
	Citizen's certified public key	Department of Transportation's signature
Property purchase	New owner's certified public key	Previous owner's certified public key
	Property value	New owner's signature
	Previous owner's signature	Property land registry certificate
Tokenised vehicle	Proof of purchase	Owner's certified public key
	Vehicle insurance	Vehicle circulation documentation

Figure 4. The table shows four data record cores, including document renewal requests, personal documents, two-party contracts, and RWAs. In the second example, personal information is not stored in the data record, but users can still selectively disclose it.

- **Privacy:** SOvNet only publishes data record fingerprints on the public blockchain, ensuring that sensitive asset data remains in private, proprietary SOvNet databases. The security of these fingerprints is enhanced by using randomising salt values. SOvNet also supports selective disclosure of data record information.
- **Double-Spending Protection:** Double-spending is a critical issue in systems tokenising RWAs and supporting digital currencies. SOvNet inherits double-spending protection from the underlying blockchain: each data record is associated with a unique transaction. SNs update the RWA status (e.g., a change of ownership) by generating a new transaction that references the previous transaction.

3.2.2. Types of data records

Data records represent multiple types of data such as private documents, tokenised RWAs, and digital currencies. We discuss the information data records should contain in these use cases.

- **Documents:** SOvNet supports the creation of personal documents, which must contain both personal information and authorisation certificates. This information is stored in the core data field. We show examples of data record cores in Figure 4.
- **Real-World Assets:** SOvNet's data records can represent any type of RWA. Data records representing RWAs are structured similarly to document data records, the main difference being that their core data field contains both details on the RWA (such as ownership records, and asset value) and tokenisation records (e.g., tokenising entity).
- **Digital payments:** SOvNet's tokenization infrastructure can support both central bank digital currencies (CBDCs) and stablecoins. Both types of digital money can be implemented using the templates used for tokenized Real-World Asset (RWA), with the restriction that digital money is issued only by the central bank (for CBDCs) or an authorized entity (for stablecoins). Privacy of digital money solutions can be enhanced by using zero-knowledge proofs [51].

4. SOvNet specifications

This section presents the technical aspects of data records, providing their formal definition in Section 4.1, a discussion on fingerprint transactions in Section 4.2, the data record life cycle in Section 4.3, and data record acceptance proof generation and verification in Section 4.4.

Fingerprint transaction		
Inputs		
UTXO	Unlocking script	
A UTXO controlled by a SN	Signature of the SN controlling the UTXO	
Outputs		
Data record	Value	Locking script
Passport Data record	Dust	Requires: a signature of State Department Stores: fingerprint, checksum, template ID
Driving license data record	Dust	Requires: a signature from the Department of Transportation Stores: fingerprint, checksum, template ID
Fake data record	Dust	Requires: a signature of any revocation authority Stores: fake hash, non-compatible checksum, random template ID

Figure 5. Structure of a fingerprint transaction. The transaction consumes a UTXO owned by a SN and generates a set of outputs, some of which are linked to a data record. In this example, the first two outputs correspond to real data records, and are controlled by the corresponding RA. If there are multiple RAs, any of them controls the output (e.g. via locking scripts using 1-of-n multi-signature). The last output is not linked to a real data record; SNs can easily verify it by checking the checksum value.

4.1. Data record definition

SOvNet data records are represented by the 4-tuple $Data = (Templ, Salt, Core, Sig)$, where $Templ$ is the flag identifying the template, $Salt$ is a 256-bit random value used to randomise the fingerprint, $Core$ is the core information stored in the data record, and Sig is a confirmation signature generated by the owner of the data record. Let h be a cryptographic hash function and $\|$ denote concatenation. The message signed by Sig is $m = h(Salt) \,\|\, Templ \,\|\, Core$. The salt value is hashed so that it is not revealed when sharing the message m.

The field $Core$ can contain a plaintext version of the personal information, or only a reference to it. It is possible to generate $Core$ allowing for selective disclosure of information: the original data are fragmented, and the hash of each of these fragments labels a leaf in a Merkle tree, with $Core$ containing the root of the tree. Disclosing information requires the owner to disclose a Merkle path that links the disclosed data to a Merkle root stored in $Core$.

The output linked to $Data$ in a fingerprint transaction contains the cryptographic fingerprint $Fing$, $Templ$, and a checksum value $CkSum$, which enables SNs to detect errors or identify fake data records (see Section 4.2). The cryptographic fingerprint $Fing$ is defined as $Fing = h(h(Salt) \,\|\, Core \,\|\, Sig)$, and the checksum is $CkSum = h(\sigma \,\|\, Fing)$, where σ is a 256-bit secret controlled by the SNs. Since the SNs know the secret σ, they can easily verify $CkSum$, while the knowledge of $CkSum$ and $Fing$ does not disclose any information about σ.

4.2. Fingerprint transactions

Fingerprint transactions contain a cryptographic link to a data record. The link is created through the inclusion of a cryptographic fingerprint of the data record within one output of a blockchain transaction. Fingerprint transactions are generally dust transactions, meaning that the economic value associated with the outputs is the minimum amount allowed by the blockchain. This is different from the intrinsic value of the document or token they represent.

The cryptographic fingerprint of a data record is obtained by hashing its core together with the salt value. This approach guarantees the confidentiality of the information contained in the core of the data record and prevents brute-force attacks. Fingerprint transactions are valid only if they are broadcast by a SN using a certified public key. To reduce the number of fingerprint transactions and optimise cost, SNs can bundle multiple data records into a single fingerprint transaction, with each output corresponding to a different data record. Fingerprint transaction

generation policies can vary in each SOvNet, for example a single SN could be responsible for all fingerprint transactions, groups of SNs can be responsible for different templates, or each SN can fund and publish only the transactions it processes.

Additional data that should be included in a fingerprint transaction are a template identifier for parsing purposes and a checksum value. The checksum values should be quick to generate and verify, while ensuring they do not reveal any information to external observers. Privacy can be further enhanced by publishing fake outputs (i.e. outputs not linked to real data records) within real fingerprint transactions, obfuscating the network traffic to external observers. We show the structure of fingerprint transactions in Figure 5.

The only information published on the blockchain is the salted hashes, meaning if the associated data is deleted from the SN databases, it becomes irretrievable. SOvNet is therefore inherently compliant with the GDPR right to be forgotten [29]. Moreover, SNs can delete all personal data from SN databases, while maintaining a proof of the data being processed.

The described implementation of fingerprint transactions maps data records to UTXOs bijectively, with the advantage of using the spending status of the UTXO to show the current status of the data record (e.g. active or expired). Further discussion on this topic is postponed to Section 4.3.

We conclude this section by presenting an alternative implementation of fingerprint transactions that reduces blockchain footprints. Instead of publishing multiple UTXOs, each fingerprint transaction contains a single UTXO labelled with the Merkle root of a Merkle tree representing multiple data records. This reduces blockchain bloating at the cost of requiring SNs to share Merkle proofs with users. Linking multiple data records to the same UTXO prevents the system from using the spending status of the UTXO to provide updates on the status of the corresponding data records.

4.3. Data record life cycle

Data records are created by users such as citizens processing payments, and administration issuing documents. The life cycle of a data record consists of creation, validation, proof of acceptance, and optionally expiration or revocation. Creation includes the collection of the core information formatted according to the template, and the inclusion of the template flag and salt value. Once the confirmation signature is computed, the data record is sent to the SNs.

Upon reception of data records, SNs verify their validity. This includes the following checks: adherence to a recognised template, inclusion of all the relevant supporting information, validity of the confirmation signature, and verification of the user's authorisation level. Data records passing these checks are added to the SN databases. A designated SN, using a certified public key, generates and publishes the relative fingerprint transaction on the blockchain. Figure 6 provides a summary of the steps in the creation of data records.

Once a new blockchain block is produced, all fingerprint transactions are extracted from the block, and all data records linked to a UTXO generated by those fingerprint transactions are processed. For each of these data records, SNs generate two proofs of acceptance, one based on the blockchain SPV proof, the other based on only SOvNet data. SNs provide both proofs to data records' owners. Generation of these proofs and how they can be used for user P2P interactions is described in Section 4.4.

Certain data records may require an expiration date or revocation, which SOvNet can provide based on the spending status of the output containing the data record fingerprint. A data record is valid while the corresponding output is unspent and expires when the output is spent. When fingerprint transactions are created, SNs choose the authority that can trigger the expiration of the data record, by linking the output to the public keys of the chosen entity. When a data record expires or is revoked, its SN database entry is updated to reflect the status.

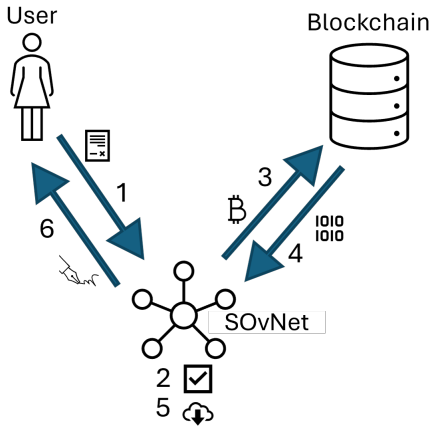

Figure 6. Data record life cycle. Users generate a data record and send it to the SNs (1). SNs check its correctness (2) and include it in a fingerprint transaction (3). When the fingerprint transaction is added to a blockchain block, SNs receive an SPV proof (4). The data record is stored in the SN databases (5), and the proofs of acceptance is shared with user (6).

4.4. P2P data record verification

In previous sections we described how SNs can verify data records. In some use-cases, data records may need to be verified by other users, such as citizens showing data records representing their driving license to police officers or buyers verifying the validity of a certificate before buying a property. This verification is called peer-to-peer (P2P) verification.

In a P2P verification, the verifier checks the data record integrity, validates its proof of acceptance, and verifies its current status. Checks include verification that the data record follows a proper template, confirmation of ownership, and signature validation. Ownership of the data record is verified using a challenge-response protocol, asking for a valid signature linked to the owner's certified public key. If some data are selectively disclosed, the Merkle paths of the disclosed information are verified. The current status of a data record is verified by checking the spending status of the output containing its fingerprint. Blockchain explorers commonly mark outputs as spent or unspent, making this process straightforward. If multiple data records correspond to the same output, this is not possible, and the status of the data record should be retrieved by querying the SNs.

We conclude this section by presenting two techniques to prove acceptance of data records, the first only requiring SOvNet data for verification, the second verified using blockchain data.

4.4.1. SOvNet data acceptance proofs

Verification based on SOvNet data minimises the overall system complexity by limiting the interaction with the blockchain. The data record acceptance proof using SOvNet data relies on signature validation, and the protocol to generate key pairs is provided below. This protocol provides key rotation by design (each blockchain block is associated with a different public key) and strengthens the link between a data record and the UTXO containing its fingerprint.

SNs publish the public key K^B_{pub} to SOvNet's users and provide the owner of the data record $Data$ with the signature $sig(K^B_{priv}, Fing)$. Notice that only SNs can compute the private key K^B_{priv}, since they are the only party knowing σ_B.

To verify a data record, the owner sends to a verifier the data record, the signature, and the public key. The verifier computes the fingerprint of the data record, validates the signature, and checks if the public key provided is one of those published by the SNs. The verifier does not require any interaction with the blockchain for this proof.

Algorithm 1 *Generation of the proof of acceptance using SOvNet data*. SNs initialise the private key to a secret and scan the block for fingerprint transactions. Fingerprints are used to update the private key. Once the new key is computed, fingerprints are signed by the updated key, and this signature works as proof of acceptance.

Input: Blockchain block B, a secret σ_B
Output: List of proofs Pfs, public key K^B_{pub}

Initialisation
$K^B_{priv} = \sigma_B$, $Pfs = [\,]$, $Fings = [\,]$.
Key generation
for transaction Tx in B **do**
  if Tx is a fingerprint transaction **then**
    for outputs in Tx **do**
      if matching *CkSum* **then**
        Extract $Fing$ from the output.
        K^B_{priv} += $Fing$.
        Append $Fing$ to $Fings$.
Compute the public key K^B_{pub}.
for $Fing$ in $Fings$ **do**
  Sign $Fing$ using K^B_{priv}.
  Append the signature to Pfs.
return K^B_{pub}, Pfs.

Algorithm 2 *Verification of a data record using SOvNet proofs*. The verifier checks the fingerprint, validates the signature, and ensures the public key was published by a SN.

Input: Data record D, signature Sig, public key K_{pub}
Output: Verification result V (valid or invalid)

Fingerprint computation
Compute $Fing_D$ from D.
Signature validation
Verify Sig using K_{pub} and $Fing_D$.
if Signature is invalid **then**
  return *Invalid*.
Public key verification
if K_{pub} is in the list of keys published by SNs **then**
  return *Valid*.
else
  return *Invalid*.

4.4.2. Blockchain data acceptance proofs

Using blockchain information allows for the verification of the data records by entities that are not SOvNet's users and thus cannot require validation by SNs. Data record acceptance is proved based on the blockchain SPV proof. The verifier needs to have access to the list of blockchain headers and to the list of certified public keys the SNs use to fund fingerprint transactions. The list of certified public keys can be stored on the blockchain.

Algorithm 3 *Verification of a fingerprint transaction using blockchain data.* The verifier ensures the transaction was funded by SNs, the fingerprint matches an output, and the SPV proof is valid.

Input: Data record D, fingerprint transaction Tx, SPV proof SPV
Output: Verification result V (valid or invalid)

Transaction validation
Check that Tx was funded by SNs.
if Tx is not funded by SNs **then**
    return *Invalid*.
Fingerprint check
Compute $Fing_D$ from D.
if $Fing_D$ is not in an output of Tx **then**
    return *Invalid*.
SPV proof validation
if SPV is available **then**
    Validate SPV.
    if SPV is invalid **then**
        return *Invalid*.
else
    Verify Tx directly on the blockchain.
return *Valid*.

A verifier receives a data record, a fingerprint transaction, and an SPV proof. The verifier checks that the transaction was funded by the SNs, that the data record's fingerprint is contained in one of the outputs, and that the SPV proof is valid. If an SPV proof is not available, the transaction validity can be verified by checking the blockchain.

If multiple data records correspond to the same UTXO, an additional step is required. The prover must also provide a Merkle path for the data record in the fingerprint transaction, which needs to be verified during the fingerprint check.

4.5. Digital payments

SOvNet supports digital payment solutions implemented either through account-based or UTXO-based models. In account-based systems, users are linked to wallets, and SNs track account balances. In UTXO-based models, SNs monitor collections of valid UTXOs to determine balances. Figure 7 shows how the core of the data records is structured in each of the two models. In both models, the spending status of a fingerprint transaction representing digital money transfer carries no information on the underlying data record.

In account-based models, data records representing digital money transfers contain the sending and receiving wallets and the transaction amount. Transactions requiring additional information are supported and rely on as-hoc templates. Transaction validation requires only verification of account balances. To increase user privacy, proof of funds can be implemented in zero-knowledge fashion using range proofs [52, 53]. Zero-knowledge range proofs can also be used to hide the payment amount while complying with cash-payment limitations [54].

In UTXO-based models, transaction data records contain a list of funding UTXOs and a list of generated UTXOs, each consisting of an amount and a cryptographic puzzle (e.g. a digital signature). To spend a UTXO, users provide a solution to the puzzle. SNs keep track of the set of valid UTXOs, but do not need to track their ownership, increasing users' privacy. Peer-to-peer token transfers are achieved through proof of ownership. A simple UTXO-based implementation is the chain-of-commitments model (e.g. Universal Blockchain Assets [55]).

5. Inter-SOvNet communication

If two users belong to different SOvNets, they are not able to use SOvNet data acceptance proofs to verify their data. Blockchain data acceptance proof allows for this verification (see

Digital payment model	Core data	
Account-based	Payer account ID	Payee account ID
	Payment amount	Payer signature
UTXO-based	List of input UTXO	List of output UTXO
	Unlocking signatures	

Figure 7. Information contained in a digital money transaction data record. This structure supports both CBDCs and stablecoins. The template identifier determines additional information on the transaction such as the currency used, and if supporting documents were required.

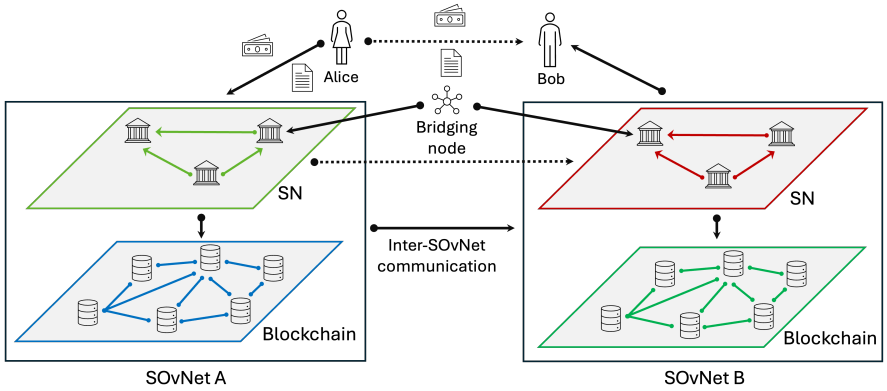

Figure 8. Inter-SOvNet communication. The bridging node is a SN of SOvNet B that is either a user or SN of SOvNet A. Users within SOvNet A communicate with users of SOvNet B through one of the bridging nodes without registering with the latter system.

Section 4.4.2), but at the cost of relying on an SPV node to verify relevant block headers. While this approach is practical for sporadic interaction, it requires trust in the underlying blockchain and does not scale. If the verifier is a user of both SOvNets, they can rely on the more efficient SOvNet data verification proof. However, this comes at the cost of multiple registration processes. To address these issues, we introduce the inter-SOvNet communication protocol, which allows data sharing between separate SOvNets without requiring users to communicate with multiple SOvNet instances. Selective disclosure of information can be integrated within this protocol to reduce the risk of data leaks between the two networks. Figure 8 summarises the protocol.

Inter-SOvNet communication opens a range of applications in public administration and enterprise. As a guiding example, consider cross-network data exchange in border control. Alice, a citizen of country A, wishes to enter country B. These countries process their information within SOvNets A and B respectively. To cross, she submits her digital passport and an identity statement to a bridging node linking the two SOvNets. This node verifies the data record, creates a confirmation data record in SOvNet B, and once confirmed, shares this with the immigration officers to confirm Alice's identity and entry rights.

5.1. Linking two SOvNets

A risk-assessment phase is required for two entities to link their networks. During this phase, each entity evaluates the reliability and integrity of the other's SOvNet. If both assessments are successful, the entities mutually agree to register Bridging Nodes (BNs) on each other's SOvNets. These BNs are SNs of their home SOvNet , and either SNs or users of the destination SOvNet . To ensure operational resilience, multiple BNs can be registered to avoid relying on a single point of failure.

User-level registration minimises the risk of data leaks from the source network. As a user, the BN gains read access to SOvNet A, but has access only to data directly shared with

Operation	Simulation on smartphone			Simulation on laptop		
	10 MB	100 MB	1 GB	10 MB	100 MB	1 GB
Confirmation signature generation	17 ms	174 ms	1.82 s	12 ms	117 s	1.21 s
Fingerprint transaction generation	24 ms	232 ms	2.47 s	16 ms	155 ms	1.59 s
P2P data record verification (SOvNet data)	22 ms	233 ms	2.32 s	16 ms	157 ms	1.55 s
P2P data record verification (Blockchain data)	23 ms	229 ms	2.34 s	14 ms	151 ms	1.48 s

Figure 9. Running time of different SOvNet's operations on a laptop and on a smartphone. When verifying acceptance using blockchain data, we assumed blockchain blocks containing 15,000 transactions.

it. SN-level participation is more suitable for networks representing different entities within the same organization (e.g. subsidiaries of a multinational company) and necessitates strict damage-control policies to prevent data leaks. As a SN of both networks, the BN gains access to both SN databases, enabling bilateral communication between the two networks.

5.2. Cross-network data record exchange

When Alice, a user of SOvNet A, wants to share a data record with Bob, a user of SOvNet B, she interacts with the BN of Bob's SOvNet within her own network. The data-sharing method depends on whether the BN is registered as a user or an SN in SOvNet A. If the BN is registered as a user, it must verify the validity of the data record using the SOvNet data acceptance proof provided by Alice. If the BN is registered as a SN of SOvNet A it can directly verify the data using the SN databases of SOvNet A.

Alice submits a data-sharing request to the BN, including the data record and Bob's identity as the recipient. The BN validates the data record either following the P2P verification or querying the SN databases. Once validated, the BN generates a verification data record in SOvNet B. This affirms the validity of the information Alice intends to prove and confirms that the record accurately represents the data. The verification data record is processed by SOvNet B. and the BN shares the proof of acceptance with Bob, who verifies the data and proof, and notifies Alice of successful receipt.

6. System evaluation

SOvNet's performance and estimated running cost is analysed in Section 6.1, Section 6.2 summarises the security features of the system.

6.1. Performance and cost

We conducted an evaluation of the computational performance of SOvNet to assess its viability for practical adoption. We have developed a prototype implementation to empirically measure the computational overhead incurred by the operations involved in SOvNet.

The operational burden was tested on a laptop (CPU Intel i7-1165G7 at 2.80 GHz, 48 GB of RAM) and on a smartphone (CPU Snapdragon 765G, 8 GB of RAM). The prototype was implemented in Python 3.11.9, using the cryptographic libraries `hashlib`, `secrets`, and `ecdsa` [56–58]. We used double SHA-256 as the hash function and ECDSA as the digital signature scheme. Each data record contained a 32-byte salt value and a 2-byte template flag.

For our performance evaluation we used the Bitcoin blockchain because of its popularity and wide adoption. We formatted fingerprint transactions to be compatible with Bitcoin transactions and published them on Bitcoin SV, an enterprise implementation of Bitcoin characterised by large block size and low transaction fees.

Data records per hour	Total transactions size per month	Bitcoin SV	Hyperledger Fabric (AMB)
10^4	0.7 GB	3.64 \$/month	643 \$/month
10^5	7 GB	36.4 \$/month	866 \$/month
10^6	70 GB	364 \$/month	1941 \$/month
10^7	700 GB	3640 \$/month	3811 \$/month

Figure 10. Monthly cost estimation of running SOvNet on different services. Bitcoin SV fees are estimated using the average transaction fees and transaction size data for April 2024 [59]. The cost of running SOvNet using AMB Hyperledger Fabric is estimated using the AMB pricing service [60]. Pricing is evaluated using a standard membership based in Europe with 3 peer node instances. Instance type is chosen according to the expected network traffic.

We ran a Monte Carlo simulation with 1000 data records (average core size of 100 MB, standard deviation of 3 MB), to assess the running cost of various operations in their life cycle. The results of the simulation are shown in Figure 9. Further testing with documents of different sizes (average core size of 10 MB and 1 GB, standard deviation of 300 KB and 30 MB) confirmed that the cost for operations manipulating the core increased linearly. These tests shows that SOvNet is suitable for deployment on lightweight devices, even when handling large documents. Data records are linked to UTXOs using 66-byte data payloads (32 bytes for the fingerprint and the checksum value, and 2 bytes for the template flag). The outputs of fingerprint transactions are locked by pay-to-public-key-hash scripts, with the data payload stored after an `OP_RETURN`; each output requires 92 bytes.

We considered different thresholds for the average number of data records produced per hour and estimated the cost of running SOvNet on Bitcoin SV. We then compared the cost of running a similar system using a private blockchain. To estimate this cost, we relied on the pricing of Amazon Managed Blockchain (AMB) services for Hyperledger Fabric, one of many choices when developing a private blockchain. Other approaches may incur different running costs. Figure 10 shows the cost comparison between the two infrastructures.

6.2. Security

Even if SOvNet is built atop a public blockchain, it guarantees a high level of confidentiality. An external observer cannot collect information about SOvNet or its users from blockchain data. Indeed, the only public information are the fingerprint transactions, but it is impossible to map an output to a data record without already knowing the data record (as the fingerprint is just a salted hash). If SNs databases are manipulated the public blockchain can be used to detect the corrupted information.

Users can request SNs to not store private information in the SN databases and to delete previously stored information. This makes SOvNet compliant with GDPR. Even if data are deleted from the SN databases, SNs can store the fingerprint of the data record as a reference to monitor its status and to prove that it was accepted.

In P2P interactions, users are not able to trick verifiers into accepting counterfeit data records. Security of the proof of acceptance using blockchain data is guaranteed by the fact that users cannot generate fingerprint transactions (as they are funded using SNs' keys). When relying on SOvNet data, security is guaranteed by the dependence of the private key on the secret σ_B. Even if all the users collude and reveal their data records and the corresponding UTXOs, they are still unable to deduce the private key without knowing σ_B.

7. Conclusion

In this paper, we introduce SOvNet, an overlay network designed specifically for governments, enterprises, and privacy-oriented institutions. SOvNet leverages features inherent to public

blockchains, such as integrity, timestamping and auditability, while safeguarding its internal confidentiality. By isolating private information in proprietary servers, SOvNet ensures compliance with GDPR and prevents external interception of sensitive data from blockchain transactions. The inter-SOvNet communication protocol allows for information sharing between different SOvNets without impacting the user experience.

SOvNet decouples blockchain transactions from private data using salted cryptographic hashes. SOvNet's reliance on the blockchain is not a dependency: normal network functions are maintained even when temporarily disconnected from the blockchain network. SOvNet is blockchain agnostic, in fact, it does not rely on particular blockchain features, such as smart contracts, and thus it supports multiple blockchain protocols and transition between them. Data can be ported to another blockchain without loss or leakage. Costs associated with SOvNet can be reduced by switching between blockchain networks. SOvNet's running costs are generally lower than alternative solutions implemented relying on private blockchains.

SOvNet's architecture is designed for versatility without compromising security, accommodating any network topology and a diverse range of data records. SOvNet relies on lightweight processes, allowing users to interact with it on personal devices such as smartphones.

Acknowledgments

This work was funded by nChain and Teranode Group.

Conflicts of Interests

Benjamin Brooks and Luigi Lunardon are currently employed by Teranode Group. Alessio Pagani is currently affiliated with the University of Bath. All authors have been employed by nChain while working on this paper. Some aspects of this paper are present in pending patent applications owned by nChain Licensing AG.

Authors' contribution

Conceptualization, L.L., A.P.; Investigation, L.L., A.P.; Formal analysis, B.B, L.L., A.P.; Visualisation, B.B, L.L., A.P.; Writing – original draft, L.L., A.P.; Writing – review & editing, B.B., L.L. All authors have read and agreed to the published version of the manuscript.

References

[1] Bhardwaj S. Hong Kong forms task force to foster Web3 development, 2023. Https://www.forbesindia.com/article/cryptocurrency/hong-kong-forms-task-force-to-foster-web3-development/86309/1 (accessed on 16–01–2024).

[2] Adejumo O. Nigeria national agency to authenticate government certificates using blockchain, 2023. Available: https://cryptoslate.com/nigeria-national-agency-to-authenticate-government-certificates-using-blockchain/ (accessed on 16–01–2024).

[3] Celdran C. Navigating the technological frontier: A Year in Review of blockchain in the Philippines, 2023. Https://coingeek.com/navigating-the-technological-frontier-a-year-in-review-of-blockchain-in-the-philippines-video/ (accessed on 16–01–2024).

[4] Gausdal AH, Czachorowski KV, Solesvik MZ. Applying blockchain technology: Evidence from Norwegian companies. *Sustainability* 2018, 10(6):1985.

[5] Yang R, Wakefield R, Lyu S, Jayasuriya S, Han Fea. Public and private blockchain in construction business process and information integration. *Autom. Constr.* 2020, 118:103276.

[6] Bindseil U. Tiered CBDC and the financial system. *Available at SSRN 3513422* 2020 .

[7] Arner DW, Auer R, Frost J. Stablecoins: risks, potential and regulation. *BIS working*

paper 2020 .

[8] Monrat AA, Schelén O, Andersson K. A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities. *IEEE Access* 2019, 7:117134–117151.

[9] Swan M. Blockchain: Blueprint for a new economy. *O'reily Media* , USA, 2015 .

[10] Azbeg K, Ouchetto O, Andaloussi SJ. BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security. *Egypt. Inform. J.* 2022, 23(2):329–343.

[11] Daniel J, Sargolzaei A, Abdelghani M, Sargolzaei S, Amaba B. Blockchain technology, cognitive computing, and healthcare innovations. *J. Adv. Inf. Technol* 2017, 8(3).

[12] Ur Rahman M, Baiardi F, Ricci L. Blockchain Smart Contract for Scalable Data Sharing in IoT: A Case Study of Smart Agriculture. In *2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*. Dubai, United Arab Emirates, December 12-16, 2020, pp. 1–7.

[13] Hjálmarsson FÞ, Hreiðarsson GK, Hamdaqa M, Hjálmtỳsson G. Blockchain-based e-voting system. In *2018 IEEE 11th international conference on cloud computing (CLOUD)*. San Francisco, CA, USA, July 2–7, 2018, pp. 983–986.

[14] Steiu MF. Blockchain in education: Opportunities, applications, and challenges. *First Monday* 2020 .

[15] Lopez ME, Murphy J, Pagani A. Bitcoin-Powered IoT Networks for Climate Change Research: a Peer-to-Peer Micro-Payment System to Enhance Data Collection. In *Proceedings of the 2023 8th International Conference on Cloud Computing and Internet of Things*. Okinawa, Japan, September 22–24, 2023, pp. 139–146.

[16] Hou H. The application of blockchain technology in E-government in China. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. Vancouver, Canada, July 31 – August 3, 2017, pp. 1–4.

[17] Lander L, Cooper N. Promoting public deliberation in low trust environments: Australian use cases. *Available at SSRN 3077474* 2017 .

[18] Khan S, Shael M, Majdalawieh M, Nizamuddin N, Nicho M. Blockchain for Governments: The Case of the Dubai Government. *Sustainability* 2022, 14(11):6576.

[19] Kimani D, Adams K, Attah-Boakye R, Ullah S, Frecknall-Hughes Jea. Blockchain, business and the fourth industrial revolution: Whence, whither, wherefore and how? *Technol. Forecast. Soc. Change* 2020, 161:120254.

[20] Kim HM, Laskowski M. Toward an ontology-driven blockchain design for supply-chain provenance. *Intell. Syst. Account. Finance Manag.* 2018, 25(1):18–27.

[21] Kamath R. Food traceability on blockchain: Walmart's pork and mango pilots with IBM. *J. Br. Blockchain Assoc.* 2018, 1(1):47–53.

[22] Jensen T, Hedman J, Henningsson S. How TradeLens delivers business value with blockchain technology. *MIS Q. Exec.* 2019, 18(4).

[23] Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis Kea. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*. Porto, Portugal, April 23–26, 2018, pp. 1–15.

[24] Consensys. Quorum Whitepaper v0.2. Available: https://github.com/ConsenSys/quorum/blob/master/docs/Quorum%20Whitepaper%20v0.2.pdf (accessed on 28–11–2024).

[25] Ølnes S, Ubacht J, Janssen M. Blockchain in government: Benefits and implications of distributed ledger technology for information sharing. *Gov. Inf. Q.* 2017, 34(3):355–364.

[26] Batubara FR, Ubacht J, Janssen M. Challenges of blockchain technology adoption for e-government: a systematic literature review. In *Proceedings of the 19th annual international conference on digital government research: governance in the data age*. Delft, Netherlands, May 30 – June 01, 2018, pp. 1–9.

[27] Clavin J, Duan S, Zhang H, Janeja VP, Joshi KPea. Blockchains for government: use cases and challenges. *Digit. Gov. Res. Pract.* 2020, 1(3):1–21.

[28] Elisa N, Yang L, Chao F, Cao Y. A framework of blockchain-based secure and privacy-preserving E-government system. *Wirel. Netw.* 2023, 29(3):1005–1015.
[29] European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), 2016. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj (accessed on 07-04-2025).
[30] Belen-Saglam R, Altuncu E, Lu Y, Li S. A systematic literature review of the tension between the GDPR and public blockchain systems. *Blockchain Res. Appl.* 2023, p. 100129.
[31] Saito K, Watanabe S. Lightweight selective disclosure for verifiable documents on blockchain. *ICT Express* 2021, 7(3):290–294.
[32] Peng Z, Xu J, Hu H, Chen L, Kong H. BlockShare: A Blockchain empowered system for privacy-preserving verifiable data sharing. *IEEE Data Eng. Bull.* 2022 45(2):14–24.
[33] Blockcerts. Blockchain Certificates, 2015. Available: https://github.com/blockchain-certificates (accessed on 07–04–2025).
[34] Lundkvist C, Heck R, Torstensson J, Mitton Z, Sena M. Uport: A platform for self-sovereign identity 2017 128:214. Available: https://whitepaper.uport.me/uPort_whitepaper_DRAFT20170221.pdf (accessed on 07–04–2025).
[35] Wu H, Peng Z, Guo S, Yang Y, Xiao B. VQL: Efficient and verifiable cloud query services for blockchain systems. *IEEE Trans. Parallel Distrib. Syst.* 2021, 33(6):1393–1406.
[36] Jensen T, Henningsson S, Hedman J. Delivering business value with blockchain technology: The long journey of TradeLens. *MIS Q. Exec.* 2019, 18(4):221–243.
[37] Kempe M. The land registry in the blockchain—Testbed: A development project with Lantmäteriet, Landshypothek Bank, SBAB, Telia company, ChromaWay and Kairos Future. *March. Stockholm: Kairos Future* 2017 .
[38] Yanovich Y, Ivashchenko I, Ostrovsky A, Shevchenko A, Sidorov Aea. Exonum: Byzantine fault tolerant protocol for blockchains. *bitfury. com* 2018 pp. 1–36.
[39] Brown RG, Carlyle J, Grigg I, Hearn M. Corda: an introduction. *R3 CEV, August* 2016, 1(15):14.
[40] Hearn M, Brown RG. Corda: A distributed ledger. *Corda Technical White Paper* 2016, 2016:6.
[41] Mastercard. Unlocking the potential of digital asset innovation: Building a Mastercard Multi-Token Network, 2023.
[42] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review* 2008 .
[43] Dwork C, Naor M. Pricing via processing or combatting junk mail. In *Annual international cryptology conference*. Santa Barbara, CA, USA, August 16–20, 1992, pp. 139–147.
[44] King S, Nadal S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August* 2012, 19(1).
[45] Merkle RC. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*. Santa Barbara, CA, USA, August 16–20, 1987, pp. 369–378.
[46] Ellis JH. The possibility of secure non-secret digital encryption. *UK Communications Electronics Security Group* 1970 8.
[47] Chokhani S, Ford W, Sabett R, Merrill C, Wu S. Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, 2003. Available: https://www.ietf.org/rfc/rfc3647.txt (accessed on 16–01–2024).
[48] Tartan C, Wright C, Pettit M, Zhang W. A Scalable Bitcoin-based Public Key Certificate Management System. *SECRYPT* 2021:548–559 .

[49] Bossu W, Itatani M, Margulis C, Rossi A, Weenink H, *et al.* Legal aspects of central bank digital currency: Central bank and monetary law considerations. *IMF working paper* 2020 .

[50] Arner DW, Auer R, Frost J. Stablecoins: risks, potential and regulation. *BIS working paper* 2020 .

[51] Van Saberhagen N. CryptoNote v 2.0, 2013. Available: https://web.archive.org/web/20201028121818/https://cryptonote.org/whitepaper.pdf (accessed on 07–04–2025).

[52] Bünz B, Bootle J, Boneh D, Poelstra A, Wuille Pea. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE symposium on security and privacy (SP)*. San Francisco, CA, USA, May 21–23, 2018, pp. 315–334.

[53] Eagen L, Kanjalkar S, Ruffing T, Nick J. Bulletproofs++: next generation confidential transactions via reciprocal set membership arguments. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Zurich, Switzerland, May 26–30, 2024, pp. 249–279.

[54] European Commission. Regulation (EU) 2024/1624 of the European Parliament and of the Council of 31 May 2024 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing (Text with EEA relevance), 2024. Available: https://eur-lex.europa.eu/eli/reg/2024/1624/oj (accessed on 07–04–2025).

[55] Vaughan O. Universal Blockchain Assets. Cryptology ePrint Archive, Paper 2024/784, 2024.

[56] Smith GP. Hashlib. https://github.com/python/cpython/blob/main/Lib/hashlib.py.

[57] D'Aprano S. Secrets, 2015. Available: https://github.com/python/cpython/blob/main/Lib/secrets.py (accessed on 07–04–2025).

[58] Hubert Kario BW Peter Pearson. Pure-Python ECDSA and ECDH, 2020. Available: https://github.com/tlsfuzzer/python-ecdsa/ (accessed on 07–04–2025).

[59] Taal. WhatsOnChain. Available: https://whatsonchain.com/block-stat/total_size (accessed on 07-04-2025).

[60] Amazon. Amazon Managed Blockchain Pricing. Available: https://aws.amazon.com/managed-blockchain/pricing/ (accessed on 07–04–2025).