

Data Sharing Control with Blockchain-based Self-Sovereign Identity Management System

Engin Zeydan

Centre Tecnològic de Telecomunicacions de Catalunya, Barcelona, Spain, 08860

Josep Mangues

Centre Tecnològic de Telecomunicacions de Catalunya, Barcelona, Spain, 08860

Suayb S. Arslan

Massachusetts Institute of Technology, MA, USA, 02139

Yekta Turk

Independent Researcher, Istanbul, Turkiye, 34396

■ **CLOUD COMPUTING** is often praised for its excellence in facilitating data sharing and manipulation, which enables various parties to collaborate and exchange information within the same technical domain [1]. Although the benefit of sharing data is massively effective due to an unlimited number of actors accessing the data simultaneously without it losing value or being altered, factors such as data protection, security and transparency in data management are also becoming equally important for data owners [2]. Therefore, efficient data handling, while taking into account both legal and ethical obligations, is an indispensable prerequisite for reaping the benefits of collaborative data sharing. In that, **Blockchain Network (BCN)** appears to be one of the most prominent technologies that enable distributed settings for data sharing and help overcome such legal and technical difficulties. Blockchain technology can also significantly improve the management of intellectual property (IP) by providing an immutable proof of ownership and automating transactions. Some possible examples include artists registering digital art

as **Non-Fungible Tokens (NFTs)** to ensure secure sales and royalties, inventors filing blockchain-based patents to prioritise and attract investors, or musicians using smart contracts to automate royalty payments and track distribution.

In the past, **BCN**-based data sharing has been investigated in various domains. For instance, the secure data sharing using blockchain was investigated in [3] within the context of drones utilizing **Attribute-Based Encryption (ABE)**. A blockchain-based platform, leveraging the benefits of **Interplanetary File System (IPFS)** is also presented in an early work [4].

BCN-based SSI

BCN-based **Self-Sovereign Identity (SSI)** technology is a novel paradigm that can be used for building systems that manage data access control due to its unique characteristics. These characteristics include decentralization, security, transparency, and tamper-proofing, which make it an ideal technology for managing identity access in the data sharing process. Implementing a **BCN**-based **SSI** system involves higher economic and computational costs compared to traditional identity management systems due to infrastructure, transaction fees, development and compli-

Digital Object Identifier 10.1109/MCE.YYYY.Doi Number

Date of publication DD MM YYYY; date of current version DD MM YYYY

ance requirements. While traditional systems are more cost-effective and scalable, BCN-based SSI systems offer enhanced security, user control and trust, making them suitable for scenarios where decentralization and immutability are critical. Many research studies have shown the effectiveness of BCN-based SSI in setting up identity access management systems for data access [5]. Previous work in [6] proposes a security mechanism based on data encryption to secure data in off-chain storage in blockchain-based identity access systems but the methodology lacks detailed numerical analysis and integration aspects with identity layers such as BCN-based SSI. Typically, BCN-based SSI systems involve data owners defining an agreement with potential data accessors, which outlines the terms and conditions of data access. This guarantees that the agreement remains unchanged or amended without the awareness and approval of all involved parties, thereby ensuring governance over secure and reliable data access [7]. Several approaches to identity access management utilizing BCN technology are proposed, exhibiting slight variations in their implementations [8]. However, the standard methodology is to record Decentralized Identifier (DID) on the BCN-based SSI system between actors who intend to share or access data. Moreover, we discuss the importance of securing data that exists beyond the BCN and proposed solutions to augment the existing methods. Although BCN-based SSI technology is effective for securely storing and managing data access control, it may not always be the best option for storing large amounts of data due to scalability issues and high transaction costs. As a solution, many organizations choose to combine BCN-based SSI with off-chain storage systems [9]. This involves using the BCN-based SSI to manage data access securely, while the actual data is stored in a separate off-chain system, just like found in [10] and [11].

Contributions for Secure Data Access

The main contribution of this article is to introduce a protocol for secure data access control in off-chain storage, integrated with BCN-based SSI systems. Unlike previous works, we propose a security mechanism for off-chain storage in BCN-based SSI systems, which is based on the idea of shared data encryption. The proposed protocol employs Hybrid Public Key Encryption (HPKE) [12] (a solution that combines the performance of symmetric key encryption with public key encryption) to safeguard the data stored in off-

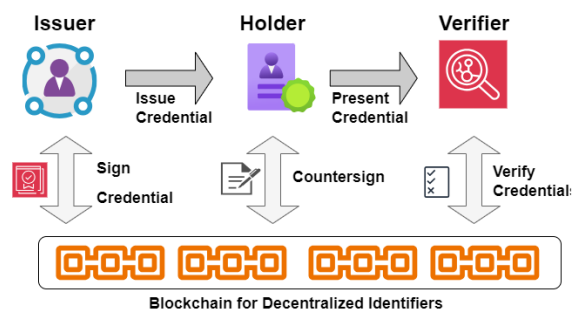


Figure 1: User centric approach for BCN-based SSI

chain storage.

One of the primary advantages of hybrid encryption is efficiency. By leveraging symmetric encryption for bulk data encryption, hybrid schemes optimize performance and reduce computational overhead compared to using asymmetric encryption alone, which is slower due to its computational complexity. Asymmetric encryption is used to securely exchange a randomly generated symmetric key between parties. This ensures that even if intercepted, the symmetric key remains confidential. Once the symmetric key is exchanged, it is used to encrypt the actual data, providing a fast and secure method for data protection. The flexibility of hybrid encryption extends to key management as well. Symmetric keys can be changed frequently for enhanced security, while asymmetric keys can be used for long-term identity verification and integrity checks.

The public key and data ID are stored in the BCN-based SSI registry. The symmetric key is used to decrypt the data in the off-chain data storage and provide access to authorized data users after validating the identity access agreement. In addition to existing methods, this mechanism adds an extra layer of security by combining encryption-based security mechanism in off-chain data storage with the BCN-based SSI identity access management process, safeguarding the data from unauthorized access by malicious actors and creating a resilient security system for data access. The performance of our approach was also tested on various relevant metrics through a test setup that involves the use of Hyperledger Indy for implementing SSI BCN and Quorum BCN for on-chain storage.

Architecture and Methodology

Existing DIM Solutions

Distributed Identity Management (DIM) solutions allow organizations to manage user identities and access to resources across different applications and

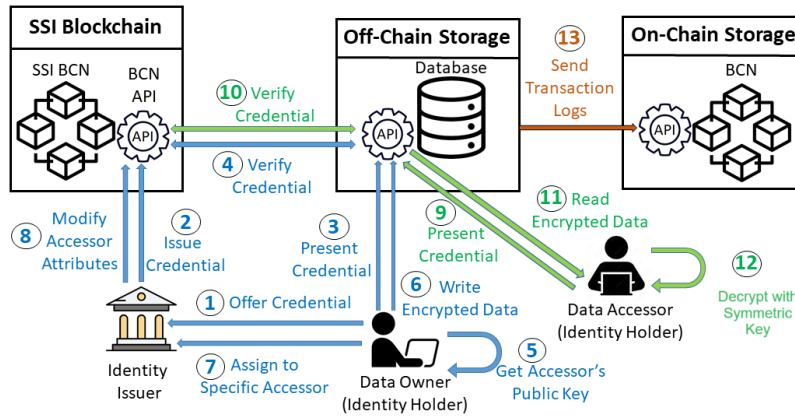


Figure 2: Secure data access with BCN-based SSI, off-chain and on-chain storage integration to allow data owners and accessors interact with smart contracts.

services in a decentralized way [13]. There are several approaches to implementing distributed identity management solutions, including: *Attribute-Based Access Control (ABAC)* by considering a wide range of attributes and conditions when making access decisions where users are granted access based on their attributes, such as title, location and role; *Federated Identity Management (FIM)*, which enables sharing of user identity information and authentication across multiple organizations or domains, where a trusted third party, called an Identity Provider (IdP), authenticates users and provides identity information to *Service Providers (SPs)* who need to verify the user's identity and *DID* which is a distributed identity management model where user identities are stored on a distributed ledger, such as a *BCN*. *BCN-based SSI* is a type of *DID*. Fig. 1 shows the user centric approach of digital identity with *BCN-based SSI* solution. In *BCN-based SSI*, users can create digital identities that includes only the information they want to share, such as shared data ID, data accessor's ID for shared data ID, data owner's ID and data access conditions. This information can be shared with others to prove the consent rights of their shared data. Note that *BCN-based SSI* systems can enhance the user's control over identity data while providing strong privacy protection, even when interacting with multiple systems or crossing jurisdictional boundaries. These mechanisms can also help to ensure that data is handled securely and in accordance with various legal requirements, which creates trust in the system.

Design Requirements

A detailed security analysis of a *BCN-based SSI* system requires an assessment of how it deals with

common security threats. The use of decentralized verification and reputation systems can help mitigate Sybil attacks, encryption and mutual authentication can protect against MitM attacks, robust consensus algorithms and smart contract audits can prevent blockchain-specific vulnerabilities, and advanced cryptographic techniques, access controls and session management can ensure privacy, confidentiality and resilience against replay, DoS and phishing attacks. Our proposed system for secure data access control in off-chain storage for consent platforms based on *BCN-based SSI* is illustrated in Fig. 2 (an extended version of generic architecture of the data control system based on data encryption in [6]). Due to the limitations of storing large data on the blockchain (e.g., on-chain *BCNs* may become slow and expensive due to the amount of data that needs to be stored and managed on the *BCN*), an off-chain storage system is utilized in our protocol. This approach can address the issues of transaction fees and scalability of *BCNs* by optimizing data management, reducing on-chain data overhead, improving transaction processing speed, and providing cost-effective and flexible storage solutions. In our protocol, big data is encrypted and then stored in the off-chain system, while the *BCN-based SSI* ledger stores user credentials, user public encryption keys (is associated with the user's *DID*) and data access consents in a secure manner using a combination of cryptographic techniques and smart contracts (to manage identity, access control, and consent enforcement, facilitate mapping of user *DIDs* to their public keys, and enforce data access policies based on the recorded consent statements). The *BCN-based SSI* is used to query the verification of the existence of consent for

data access through a combination of smart contracts, [DIDs](#) and cryptographic operations when data is requested from the off-chain system. The corresponding data for the requested dataset will then be retrieved and decrypted with the private key of data accessor. Since data is intended to be shared among multiple parties, an [HPKE](#) model is utilized. The use of [HPKE](#) can foster trust among multiple parties.

Data Storage Process

To begin the process of data sharing, the data owners/accessors (or users) generate their keys with [HPKE](#), consisting of a private key and a public key. Public keys of users are stored in the [BCN](#)-based [SSI](#) via a smart contract as part of the user's [DID](#) document and are associated with their unique [DID](#) and private keys of each users are stored on the user's device in a secure element, such as a hardware security module (HSM) which includes components such as secure element and key storage compartments, dedicated cryptographic processors, high-quality random number generators, multi-factor authentication, secure [Application Programming Interfaces \(APIs\)](#), compliance with industry and regulatory security standards. First, these users must register following the system's rules using [BCN](#)-based [SSI](#) before they can save data to be shared with other users of the system (step 1 in Fig. 2). Only after registration with [BCN](#)-based [SSI](#) system is completed, a user can log in to off-chain storage system and store encrypted data to be shared with the other data accessors. During the registration step, users offer credentials to an identity issuer to verify and authenticate their identity. The identity issuer (a trusted third party) then issues a verifiable credential that proves their identity and includes the necessary data into the [BCN](#)-based [SSI](#) system (step 2). Before completing the data storage process, the data owner selects specific data to grant approval to data accessor user to access it. To establish identity-based approval, the data owner must provide elements such as their [BCN](#)-[SSI](#) wallet address, data ID and data access conditions (time-based access, purpose-based access, role-based access, attribute-based access, geographical access and identity revocation). The attributes can be recorded in [BCN](#)-based [SSI](#) via a smart contract in an immutable manner. The smart contract executed on the [BCN](#)-based [SSI](#) provides a trusted and auditable record of the identity-access management agreement. Hence, parties with the appropriate permissions can verify the details of the agreement on the [BCN](#). Moreover,

the recorded identity approval attributes can serve as the basis for enforcing data access restrictions. The smart contract can autonomously execute access control based on the defined conditions. The verifiable credential, designed to be tamper-proof, is stored on the data owner and is shared only when the data owner wants to do so (step 3).

After data owner presents their credentials to the off-chain storage, the credentials are verified via [SSI](#)-based [BCN](#) (step 4). During the process of storing data in off-chain, [HPKE](#) is used by the users' devices. For data sharing, the data owner obtains each data accessors' public key from [BCN](#)-based [SSI](#) and uses it to derive a symmetric key for encryption. Then the data owner encrypts the data with a symmetric key (e.g. AES with Galois/Counter Mode (GCM) can be used here as a proven fast encryption mode for AES, which converts the block cipher into a stream cipher) (step 5). The encrypted data is then sent to the off-chain storage system (as shown in step 6), while data ID and specific accessor ID are sent to the [BCN](#)-based [SSI](#) via a smart contract (step 7) and modified (step 8). The smart contract in the [BCN](#)-based [SSI](#) system is modified to include the details of the data sharing transaction. Specifically, the data ID is associated with the shared data, ensuring clarity about the data being accessed, and the specific accessor ID is linked to the authorized data accessor, defining who is permitted to access the data. Note that for secure transmission of encrypted data to the off-chain storage system in step 6, some special security measures can be used (e.g., secure channels and [APIs](#), asymmetric/symmetric encryption, [Transport Layer Security \(TLS\)](#) encryption, certificate validation, secure key management, data integrity checks, data validation). Moreover, ensuring the security and integrity of encrypted data stored in an off-chain system is crucial for maintaining the trustworthiness of the entire blockchain-based system. Hence, data stored in the off-chain system is encrypted using strong encryption algorithms, has strict access controls to limit who can access the encrypted data, implements integrity checks, such as cryptographic hashes or checksums, to detect unauthorized alterations. This whole process allows the user to maintain control over the data to be shared and allows for secure encryption and sharing of data and credentials. The [BCN](#)-based [SSI](#) stores the public keys of data owners and accessors as well as the data ID, while the encrypted data is registered off-chain. Finally, the identity-based approval will be utilized by the data

accessor to access the user's data as described in next subsection, and all previous steps serve as the initial condition without which an actor (i.e. the data accessor) cannot access a data owners' data.

Data Access Process

When the authorized identity holder requests access to the data owner's data (step 9), the off-chain data storage system uses its API to request the BCN-based SSI system to check the access rights of the identity holder/data accessor (step 10). The BCN-based SSI is used to verify the existence of approval for data accessor regarding data of data owner and allows encrypted data to be read (step 11). In step 12, a smart contract on the BCN-based SSI manages that access, informs the off-chain data storage system about the data ID to be accessed by the data accessor. After receiving cipher data, data accessor drives symmetric key from her public key and DID within the BCN-based SSI system. With this key, they can decrypt data owner's data to obtain the plaintext data making it available to data accessor. This decryption happens off-chain. With successful decryption, the data accessor gains access to the data owner's data in a readable format and can use or process the data authorized. Finally, in step 13, the log transactions related users accessing off-chain storage, such as the timestamp of access, which user accessed what data, and other relevant information are periodically written to on-chain storage to maintain transparency and accountability.

The process proposed here combines identity-based data access management using BCN-based SSI and HPKE method for data which complements existing SSI-enabled approval mechanisms. The protocol leverages the identity-based data access process on the BCN-based SSI side to allow data access, while the HPKE mechanism protects the data and enables approved recipients to read it. This is particularly useful for implementing systems that manage large amounts of data and require a high level of data security in off-chain systems. It achieves this through identity-based data access management (with permission-based data access and user preference consideration), a complementary approach (with BCN-based SSI system handling identity and access control and HPKE securing the data) and the management of large amounts of data (by separating the identity management layer from the data protection layer), off-chain systems (to store data for scalability and performance reasons while keeping the data secure in a decentralized environment and only

accessible to authorized parties with dual process) as well as data protection and compliance (by combining identity management and data encryption).

Authorization verification by BCN-based SSI serves as the primary security layer for data access in the protocol. Therefore, a person who has not received authorization from the data owner cannot access the data stored off-chain. As BCN-based SSI system operates without a central trusted third party, one cannot collaborate with a potential data accessor to provide them with the data. This is a significant benefit of utilizing a BCN-based SSI as a decentralized and secure ledger, functioning without the need for central control. Moreover, it eliminates the need for traditional data middlemen who might otherwise facilitate data exchanges. Additionally, employing HPKE of data provides an extra layer of security, in conjunction with the BCN-based SSI system. Even if a person tries to access the data stored off-chain, they cannot decrypt it without the specific private key of the authorized users.

Numerical Results

Test Setup

To ensure genuine decentralization and leverage the strong security features provided by BCNs, we utilize the Hyperledger Indy BCN as the underlying BCN-based SSI technology for our suggested model. Moreover, we also use an on-chain storage to store all transaction data of off-chain storage to provide further integrity and confidentiality. For this we have used Quorum BCN for on-chain storage. In our test setup, we used the OpenStack platform to create two virtual machines. One Virtual Machine (VM) has a REpresentational State Transfer (REST) API, and the other VM has a container-based DID system using Hyperledger Indy as the DID blockchain. The DID virtual machine has 8 GB RAM, 4 vCPUs, and 40 GB disk space. Four DID container nodes were created as issuers, and a Flask-based REST API was implemented for users to send requests to the BCN, facilitating various identity and credential management operations. Users can send HTTP requests to this API to request credentials, verify them, manage their identities, authenticate themselves securely, and control access to their identity information. The communication protocol used for making calls to the terms of is the DIDComm messaging

protocol from the Hyperledger Aries project ¹. Its special features are strong encryption, privacy-friendly attributes, and provenance verification, which greatly facilitates secure communication between nodes. This setup can be used to manage client credentials and identities for secure communications between nodes, as the DIDComm protocol is instrumental in ensuring the confidentiality, integrity, and authenticity of identity-related messages.

DID and BCN Metrics

We tested the DID implementation by making registration and verification requests through the REST API, and measured the average times for establishing connections and signing messages using the DIDComm protocol. To evaluate the system's performance, we examined the following key metrics: Transactions per Second (TPS) and latency. TPS refers to the number of transactions processed by the system per second, and average TPS values were obtained on the Quorum BCN for different data/log sizes sent as transaction logs in step 13 of Fig. 2. Data size is measured in kilobytes (KB), and the time to write is recorded in seconds (s) along with the corresponding TPS values. Latency is defined as the time taken for a transaction to be processed and confirmed on the blockchain, and it was measured for various operations, including credential presentation and verification, credential offer and issuance, DIDComm connection creation and signing, and credential revocation.

Table 1 presents the average time values for credential operations, such as *average credential presentation and verification time*, *average credential offer and issuer time*, *average DIDComm connection creation time*, *average DIDComm connection signing time*, and *average DIDComm revoke credential time*. These metrics evaluate the efficiency of various credential operations, secure communication channel establishment, and credential revocation. Table 2 presents the average time to write and TPS values obtained on Quorum BCN for different data/log sizes that are sent as transaction logs. The data size is measured in kilobytes (KB), and the time to write is given in seconds (s) along with the corresponding TPS values. The time to write is measured when submitting transactions per single line, per two lines, and for the whole lines. Corresponding TPS values

¹ Online: <https://github.com/hyperledger/aries>, Available: September 2023.

TABLE 1
AVERAGE DID IMPLEMENTATION TIME VALUES
FOR CREDENTIAL OPERATIONS.

KPIs (msec)	50K Requests	75K Requests	100K Requests
Credential presentation and verification	33	35	39
Credential offer and issuer	525	565	603
DIDComm connection creation	659	681	736
DIDComm connection signing	64	68	72
DIDComm revoke credential	212	256	277

are provided for different data sizes, highlighting the system's transaction throughput capabilities.

Results and Discussions

Table 1 shows the measurement of all KPIs described above for three different scenarios, with 50K, 75K, and 100K requests. The results indicate that as the number of requests increases, the performance of the DID management system degrades, but not proportionally to the increase in requests. The credential presentation time and DIDcomm signing time are relatively fast, while the credential offer time and DIDcomm connection creation time are slower. The revoke credential time may be more affected by the number of requests. For 75K requests, the credential offer time and DIDcomm connection creation time have increased significantly compared to the previous value for 50K requests. For 100k requests, all KPIs have also shown an increase in their values. To sum up, Table 1 shows that read operations are faster than write operations. Considering that many data owners and accessors can be simultaneously online, increasing the possibility of read operations during data access planning can be important for quick query responses. The DIDcomm connection implementation is not easy to apply, while TLS can be used for fast secure implementation thanks to its features such as cipher suits, Online Certificate Status Protocol (OCSP) stapling, hardware acceleration, various key exchange mechanisms, certificate handling, session resumption and protocol configurations.

In Table 2, comparing the TPS values, we can observe that the TPS decreases as the data size increases. For example, the TPS for data size 1 KB is 64, which is higher than the TPS for data size 75 KB, which is 21. This can be attributed to the fact that larger data sizes require more computational resources and time to process and verify. Additionally, comparing the time

TABLE 2
AVERAGE TIME TO WRITE AND TPS VALUES ON
QUORUM BCN FOR DIFFERENT LOG SIZES.

Data size (KB)	Time to Write(s) /TPS —per line—	Time to Write(s) /TPS —per two lines—	Time to Write(s) /TPS —whole lines—
1	0.063 / 64	0.048 / 42	0.048 / 21
2	0.284 / 64	0.119 / 84	0.048 / 21
4	0.979 / 64	0.41 / 75	0.048 / 21
6	1.247 / 64	0.522 / 77	0.048 / 21
13	5.101 / 71	2.137 / 75	0.048 / 21
39	16.74 / 64	7.013 / 76	0.048 / 21
75	27.226 / 60	11.406 / 77	0.048 / 21

to write for different data sizes, we can see that the time to write increases as the data size increases. This is expected since larger data sizes require more time to write to the BCN. Finally, comparing the time to write per line, per two lines, and for the whole lines, we can see that the time to write per line is the highest, followed by the time to write for two lines and the whole lines. This can be attributed to the fact that writing data line-by-line incurs more overhead in terms of BCN communication and verification compared to writing data for multiple lines at once.

Note that the strategies for improving TPS and reducing the time to write for larger data sets, the following common approaches can be further evaluated: (i) Using consensus mechanisms that are better suited for high throughput, such as Practical Byzantine Fault Tolerance (PBFT) or HoneyBadgerBFT. (ii) Grouping multiple transactions into a single batch before submitting them to the blockchain reduces the effort required to process individual transactions. (iii) Offloading some processing tasks, especially compute-intensive ones, to off-chain solutions such as state channels or sidechains can process transactions quickly and settle them periodically on the main blockchain. (iv) Implementation of data compression techniques can reduce the size of transactions so that smaller transactions can be processed more quickly. (v) Using caching mechanisms to store frequently accessed data in memory to significantly reduce the time required to retrieve data, especially in high read access cases. (vi) Exploring scaling solutions such as Lightning Network (for Bitcoin), Plasma (for Ethereum), or similar solutions for other blockchains can increase TPS by enabling off-chain or Layer 2 transactions. (vii) The implementation of parallel processing for simultaneous processing of multiple transactions can be achieved by multi-threading or distributed processing. (viii) Reviewing and optimizing smart contracts can make them more efficient in terms of gas usage and execution time. (ix) Considering upgrading hardware resources,

including faster processors and larger RAM can handle larger data volumes and processing loads.

In our protocol, there are also challenges associated with on-chain storage in terms of speed and cost. To ensure a balance between data security and operational efficiency, a dual-process approach (off-chain and on-chain storage) is employed. By adopting this hybrid approach, we strike a balance between the speed and cost benefits of off-chain storage and the security and transparency provided by on-chain recording. This way, we leverage the strengths of both storage methods to create a robust and efficient data access control protocol. At the same time, when integrating an BCN-based SSI system into existing data management and security infrastructures, especially in environments with legacy systems, strategic approaches must be taken to ensure seamless integration while maintaining data security and compliance. The use of middleware and API gateways for communication, data interoperability with ETL processes and standard protocols, federated identity management and credential bridging for IAM integration as well as adapters & incremental integration are some possible measures to gradually adapt with legacy systems. .

Conclusions and Future Work

Our study has shown that BCN-based SSI system is effective in controlling data access. However, when BCN-based SSI is combined with off-chain systems, additional security mechanisms are required to safeguard the data. To solve this, we introduced a protocol that improves data security through the combination of BCN-based SSI and HPKE mechanisms. This strategy provides a multi-level security system for data stored in off-chain systems that are linked to a BCN-based SSI system. By utilizing a BCN-based SSI and off-chain storage system with hybrid asymmetric and symmetric keys, data access control can be efficiently managed to guarantee privacy, security, and data integrity. We have also validated our approach in a test setup with SSI BCN implemented using Hyperledger Indy and on-chain storage with Quorum BCN on various relevant metrics. Future work can focus on ensuring compliance with international data protection regulations such as General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA) by integrating advanced cryptographic techniques, robust consent management, and rigorous audit mechanisms into the blockchain-based system.

Detailed testing of HPKE to validate its performance and investigating the specific performance metrics of a blockchain-based SSI system under different network conditions, system load and different attack vectors are also of interest.

Acknowledgment

This work was partially funded by Grant PID2021-126431OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”, Generalitat de Catalunya grant 2021 SGR 00770 and Spanish MINECO - Program UNICO I+D (grants TSI-063000-2021-54 and -55)

REFERENCES

1. S. D. Okegbile *et al.*, “Performance analysis of blockchain-enabled data-sharing scheme in cloud-edge computing-based IoT networks,” *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21520–21536, 2022.
2. E. Zeydan *et al.*, “Post-quantum blockchain-based data sharing for IoT service providers,” *IEEE Internet of Things Magazine*, vol. 5, no. 2, pp. 96–101, 2022.
3. C. Feng *et al.*, “Efficient and secure data sharing for 5G flying drones: a blockchain-enabled approach,” *IEEE Network*, vol. 35, no. 1, pp. 130–137, 2021.
4. M. Naz *et al.*, “A secure data sharing platform using blockchain and interplanetary file system,” *Sustainability*, vol. 11, no. 24, p. 7054, 2019.
5. M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, “In search of self-sovereign identity leveraging blockchain technology,” *IEEE Access*, vol. 7, pp. 103059–103079, 2019.
6. M. Goint, C. Bertelle, and C. Duvallet, “Secure access control to data in off-chain storage in blockchain-based consent systems,” *Mathematics*, vol. 11, no. 7, p. 1592, 2023.
7. A. Preukschat and D. Reed, *Self-sovereign identity*. Manning Publications, 2021.
8. K. Rantos *et al.*, “Advocate: a consent management platform for personal data processing in the IoT using blockchain technology,” in *Innovative Security Solutions for Information Technology and Communications, Revised Selected Papers 11*, pp. 300–313, Springer, 2019.
9. J. Jayabalan and N. Jeyanthi, “Scalable blockchain model using off-chain ipfs storage for healthcare data security and privacy,” *Journal of Parallel and Distributed Computing*, vol. 164, pp. 152–167, 2022.
10. C. Liu *et al.*, “Extending on-chain trust to off-chain – trustworthy blockchain data collection using trusted execution environment (tee),” *IEEE Transactions on Computers*, vol. 71, no. 12, pp. 3268–3280, 2022.
11. S. S. Arslan and T. Goker, “Compress-store on blockchain: a decentralized data processing and immutable storage for multimedia streaming,” *Cluster Computing*, vol. 25, no. 3, pp. 1957–1968, 2022.
12. R. Barnes, K. Bhargavan, B. Lipp, and C. Wood, “Rfc 9180: Hybrid public key encryption,” 2022.
13. E. Maler and D. Reed, “The venn of identity: Options and issues in federated identity management,” *IEEE Security & Privacy*, vol. 6, no. 2, pp. 16–23, 2008.

Engin Zeydan is currently a Senior Researcher at CTTC.

Josep Mangués-Bafalluy is Senior Researcher and Head of the Services as Network Research Unit of the CTTC.

Suayb S. Arslan is currently visiting Massachusetts Institute of Technology, Boston, MA, USA as a faculty member.

Yekta Turk is an independent researcher.