BlockEdge-based Private Virtual Storage for Personal Data Privacy Protection

Jung-Hyok Kwon
Smart Computing Laboratory
Hallym University
Chuncheon, South Korea
jhkwon@hallym.ac.kr

Sol-Bee Lee
Smart Computing Laboratory
Hallym University
Chuncheon, South Korea
thfqla3535@hallym.ac.kr

Eui-Jik Kim[*]
Division of Software
Hallym University
Chuncheon, South Korea
ejkim32@hallym.ac.kr

***Abstract*—This paper presents a BlockEdge-based private virtual storage, abbreviated BPVS, which aims to provide scalable and costless private virtual storage while protecting data privacy using a blockchain. The private virtual storage is built by reusing the remaining storage space of the existing Internet of Things (IoT) devices. Data access for private virtual storage is restricted only to authorized users whose authority is verified by a blockchain-based decentralized identity (DID). In BPVS, the BlockEdge is a special device responsible for storage and access management. To verify the feasibility of the BPVS, experimental implementation was conducted. The results demonstrated that BPVS is a practical solution for protecting personal data privacy.**

Keywords—blockchain, BlockEdge, personal data, privacy protection, private virtual storage

I. Introduction

Recently, demand for Internet of Things (IoT) services such as smart cities, intelligent transportation systems (ITSs), and digital healthcare has been sharply increasing [1, 2]. In these services, a number of IoT devices generate a tremendous amount of data by monitoring the surrounding environment. Therefore, there is an increasing interest in storage technology to store data efficiently. Traditionally, cloud systems (e.g., Google Drive, Apple iCloud, and Amazon Web Services) and network-attached storage (NAS) have been considered the as representative solutions for storing data [3–6]. With a cloud system, users store data in the cloud storage managed by a third-party service provider. On the other hand, with NAS, the user stores data on local storage servers installed at a personal site. In both solutions, authorized users can access data whenever an Internet connection is available.

However, the existing solutions may make it difficult to protect personal data privacy, including sensitive information [7]. Specifically, the cloud system may suffer from data loss and leakage caused by infrastructure failures and poor authority management. In addition, there is a possibility that internal employees of the cloud service provider may leak or corrupt data with malicious intent. In the case of NAS, insider threats can be mitigated since the owner has direct control over the security policies of the local storage server. However, it is vulnerable to physical theft and damage and may suffer from weak authentication and authorization policies if not properly configured. Moreover, this solution can be costly to install and expand, as it requires purchasing and maintaining additional storage servers.

In this paper, we propose a BlockEdge-based private virtual storage (BPVS) designed to protect personal data privacy [8]. To this end, we first develop private virtual storage by reusing the remaining storage space of IoT devices. Specifically, the storage space of the embedded devices is divided into private and sensing storage spaces, and then the private storage space of multiple IoT devices is logically integrated to form a single storage unit. To this end, in private virtual storage, the BlockEdge maintains the storage space of IoT devices and the location of stored data. Then, we develop a secure data access control scheme using a blockchain-based decentralized identity (DID) to prevent unauthorized access to personal data. Specifically, the client registration procedure and the authority verification procedure are developed to grant access authority to the client and verify the client's access authority, respectively. For this, BlockEdge directly interacts with IoT devices, clients, and blockchain networks to grant and verify access authority. We conducted the experimental implementation using a Raspberry Pi 4 Model B open-source hardware to verify the feasibility of the BPVS. The implementation results showed that BPVS is a practical solution for personal data privacy protection.

The remainder of this paper is organized as follows. In Sect. II, the design of BPVS is described in detail. Sect. III presents experimental implementation results. Finally, Sect. IV concludes this paper.

II. Design of BPVS

The BPVS is designed to protect personal data privacy by providing private virtual storage and secure data access control. In this section, we first present the detailed design of private virtual storage built based on multiple IoT devices. Then, we describe a secure data access control scheme using blockchain-based DID, which includes client registration and authority verification procedures.

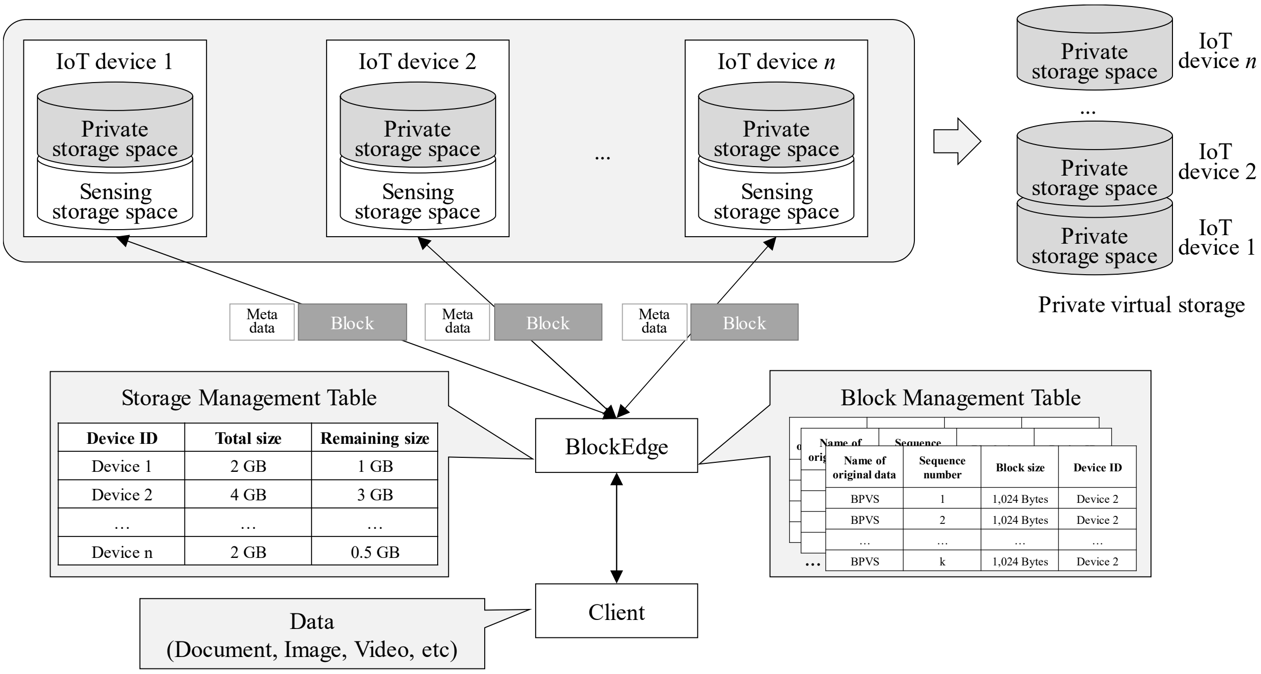
A. Private Virtual Storage

Figure 1 shows an architecture of private virtual storage that consists of a client, a BlockEdge, and multiple IoT devices. As shown in the figure, the storage space of each IoT device is divided into two types of spaces: private storage space and sensing storage space. The private storage space is an extra storage space that is rarely used by the IoT device. On the other hand, the sensing storage space is a dedicated storage space for IoT device operation (e.g., operating system, sensing application, sensing data collection). The size of each space is defined by the IoT device. The information for each space is informed to the BlockEdge when the connection is initiated. Upon receiving the storage space information, the BlockEdge lists the information on the storage management table. Table I shows an example of a storage management table that contains the device ID, the total size of private storage space, and the remaining size of private storage space. By using the storage management table, the BlockEdge builds the private virtual storage. Specifically, it logically integrates the private storage space of multiple IoT devices to form a single storage unit.

[*] Corresponding Author

2024 IEEE International Smart Cities Conference (ISC2) | 979-8-3503-6431-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISC260477.2024.11004207

979-8-3503-6431-6/24/$31.00 ©2024 IEEE



Device ID	Total size	Remaining size
Device 1	2 GB	1 GB
Device 2	4 GB	3 GB
…	…	…
Device n	2 GB	0.5 GB

Name of original data	Sequence number	Block size	Device ID
BPVS	1	1,024 Bytes	Device 2
BPVS	2	1,024 Bytes	Device 2
…	…	…	…
BPVS	k	1,024 Bytes	Device 2

Fig. 1. Architecture of private virtual storage.

TABLE I. EXAMPLE OF STORAGE MANAGEMENT TABLE

Device ID	Total size	Remaining size
Device 1	2 GB	1 GB
Device 2	4 GB	3 GB
…	…	…
Device *n*	2 GB	0.5 GB

TABLE II. EXAMPLE OF BLOCK MANAGEMENT TABLE

Name of original data	Sequence number	Block size	Device ID
BPVS	1	1,024 Bytes	Device 2
BPVS	2	1,024 Bytes	Device 2
…	…	…	…
BPVS	*k*	1,024 Bytes	Device 2

In order to store and extract the data, the client transmits a request message to BlockEdge. When storing data, the client transmits both the request message for storing data and the original data. Note that the data type can be various (e.g., documents, images, and video). Upon receiving the data, the BlockEdge performs three major functional operations.: 1) data blocktization, 2) storage selection, 3) metadata creation. In the former, the BlockEdge divides the original data transmitted from the client into multiple blocks. Each block has a fixed size, and the size of the block can be determined depending on the maximum payload size of the application protocol. In storage selection, BlockEdge selects the IoT device to store the blocks. Specifically, the BlockEdge compares the remaining size of the private storage space listed in the storage management table. Then, one of the IoT devices that has the largest remaining size is selected. In the case that the total size of the block is greater than the remaining size of the selected IoT device, multiple IoT devices are selected to accommodate all the blocks. Before creating metadata, the BlockEdge generates a block management table for the requested data. Table II shows an example of a block management table containing the name of the original data, sequence number of block, block size, and selected IoT device ID. Then, it creates the metadata including the name of the original data and the sequence number of the block. Finally, the BlockEdge transmits both the block and metadata designated to the block to the selected IoT device and updates the storage management table.

If BlockEdge receives the extraction request message containing the name of the original data, it first checks the block management table to find the ID of IoT devices where blocks of the requested data are stored. Then, BlockEdge extracts the blocks from the IoT devices and integrates blocks in sequential number order to restore the original data. Finally, it transmits the restored data to the client.

B. Secure Data Access Control

The operation of secure data access control consists of two major procedures: 1) client registration and 2) authority verification. The first procedure grants access authority to the specific client, and the second procedure verifies the access authority of the client. For this, the blockchain-based DID is used to securely manage both procedures in a decentralized manner, reducing the risk of unauthorized access. The DID is which is represented by 'Scheme: Method:Method-Specific Identifier' as shown in Fig. 2 [9–11].

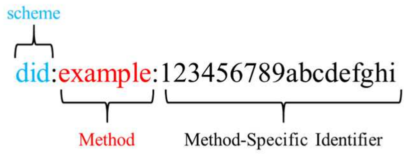


Fig. 2. Decentralized Identifier.

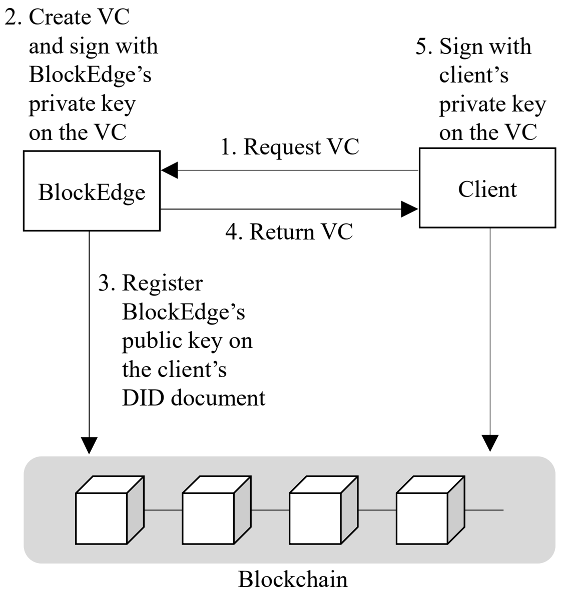

2. Create VC and sign with BlockEdge's private key on the VC
5. Sign with client's private key on the VC
1. Request VC
BlockEdge
Client
4. Return VC
3. Register BlockEdge's public key on the client's DID document
Blockchain

Fig. 3. Procedure of client registration.

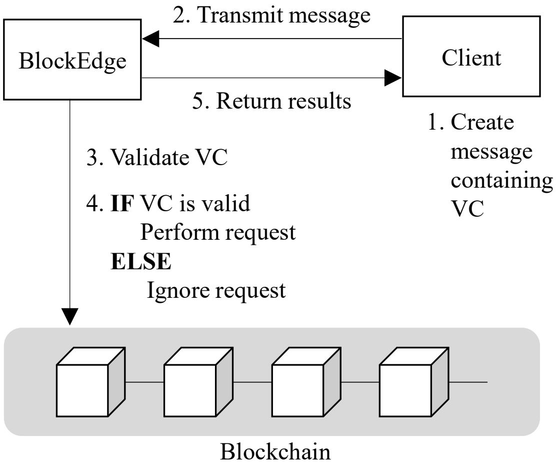


Fig. 4. Procedure of authority verification.

In the figure, 'did' means it will follow the DID schema, and 'example' denotes the name of the DID method. A DID method is a mechanism for creating, reading, updating, and deactivating DID documents related to a DID in a specific blockchain. Finally, '123456789abcdefghi' means a unique ID in the blockchain specified by DID method. It can be used to search the DID document registered in the blockchain. Note that DID document contains information associated with a DID such as id, public key, authentication, and services.

Figure 3 shows the procedure of the client registration. Firstly, the client transmits a request message containing a DID, serial number, manufacturer, data attributes to claim a verifiable credential (VC) to the BlockEdge. The VC is a document used to verify the client's authority for data access. It contains VC's DID Uniform Resource Locator (URL), type of VC, issuer's DID, issued date of VC, and a list of identity attributes included in the request message. Upon receiving the requests, the BlockEdge creates VC and signs the VC with its private key. Then, it registers the public key in the DID document of the client, which exists in the blockchain. Afterward, the BlockEdge returns the issued VC. Finally, the client signs with its private key.

Figure 4 shows the procedure of the authorized access. In this step, the client first creates a request message containing the VC. Then, it transmits the message to the BlockEdge. Upon receiving the message, the BlockEdge validates the VC transmitted from the client. To this end, the BlockEdge searches the DID document using client VC on the blockchain and obtains the client's public key. Then, it decrypts the digital signature using the public keys of the client and the BlockEdge. If the authority of the client is verified, the BlockEdge performs the request of the client. In other words, it conducts data storing or data extraction based on the request message. On the other hand, if the VC is not valid, the BlockEdge ignores the request message. Finally, it returns the results to the client.

III. Experimental Impelmentation

The experimental implementation was conducted to demonstrate the BVPS, as shown in Fig. 5. The Raspberry Pi 4 Model B open-source hardware platform running with the Linux-based Raspbian OS is used to develop the IoT devices and BlockEdge.

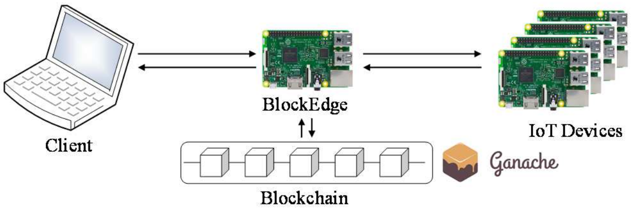


Fig. 5. Eperimental implementation.

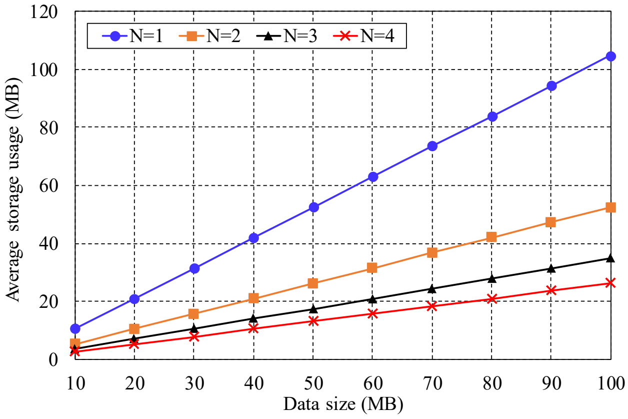


Fig. 6. Average storage usage for different numbers of IoT devices.

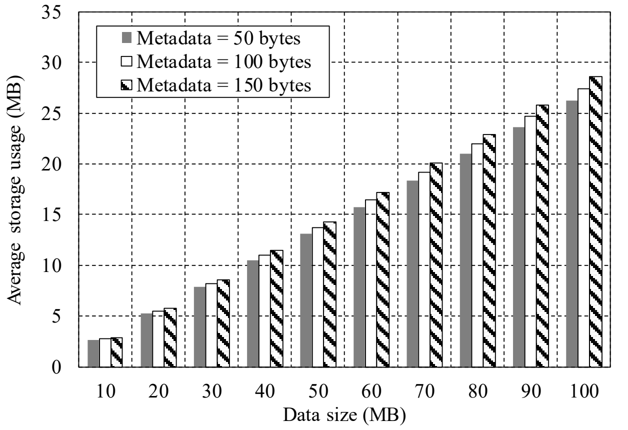


Fig. 7. Average storage usage for different sizes of metadata.

The application for private virtual storage was developed the C/C++. The size of metadata was set from 50 bytes to 150 bytes, and the size of the block was fixed to 1024 bytes. The data rate between BlockEdge and IoT devices was set to 100 Mbps. The size of the original data was set from 10 MB to 100 MB. The size of private storage space for each IoT device was initialized to 1 GB. To implement a DID on the blockchain, we used visual studio code. The smart contract was developed using the Truffle Ethereum smart contract development framework and OpenZepplin smart contract library. In addition, to develop the DID, we used DID-JS and DID-ETHR libraries. The local Ethereum blockchain network was built using Ganache, a program that creates a virtual Ethereum network and allows users to execute smart contracts.

Figure 6 shows the average storage usage for different numbers of IoT devices. The average storage usage represents the average reduced size of private storage space for each IoT device by storing the client's data. In this experiment, we set the number of IoT devices (i.e., N) from one to four. In addition, the metadata size was fixed to 50 bytes. Overall, the average storage usage decreases as the number of IoT devices increases regardless of the size of the client's data. This is because private virtual storage is built by integrating the remaining storage space of all IoT devices. In other words, the greater the number of IoT devices, the larger the total size of the private virtual storage, thereby reducing average storage usage.

Figure 7 depicts the average storage usage for different sizes of metadata. In this experiment, the number of IoT devices was fixed to four. In the figure, when the size of metadata increases, the average storage usage increases. This is because the IoT device stores both block and metadata. when metadata increases to 50, 100, and 150 bytes, the average storage usage increases by 4.89%, 9.77%, and 14.65%, respectively, compared to the case where only blocks are stored. Accordingly, in order to efficiently use the storage space of BPVS, it is necessary to reduce the size of metadata.

IV. Conclusion

In this paper, we propose a BPVS to offer scalable and cost-free private storage while ensuring data privacy through blockchain technology. The BPVS utilizes the remaining storage space of existing IoT devices to create private virtual storage. Access to this private virtual storage is limited to authorized users, with their permissions validated by a DID system based on a blockchain. Within BPVS, the BlockEdge serves as a dedicated device that handles storage and access management. Experimental implementation was carried out to test the viability of BPVS. The results demonstrate that BPVS is a practical solution for safeguarding personal data privacy. In order to verify the possibility of widespread adoption, we will study the scalability of BPVS and discuss how the BPVS aligns with existing data protection regulations, in future work.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00353365). This research was supported by Basic Science Research Program through the NRF funded by the Ministry of Education (NRF-2020R1I1A3052733). This work was supported by the NRF grant funded by the Korea government (MSIT) (NRF-2021R1C1C2095696). Following are results of a study on the "Leaders in INdustry-university Cooperation 3.0" Project, supported by the Ministry of Education and National Research Foundation of Korea.

References

[1] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, January 2020.

[2] J. Hwang, L. Nkenyereye, N. Sung, J. Kim, and J. Song, "IoT Service Slicing and Task Offloading for Edge Computing," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11526–11547, July 2021.

[3] Google Drive: https://www.google.com/drive/ (accessed August 2024)

[4] iCloud: https://www.icloud.com (accessed August 2024)

[5] Amazon Web Services: https://aws.amazon.com/ (accessed August 2024)

[6] T.-C. Huang and D.-W. Chang, "TESA: a temporal and spatial information aware writeback policy for home network-attached storage devices," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 122–129, February 2013.

[7] L. D. Xu, Y. Lu, and L. Li, "Embedding Blockchain Technology Into IoT for Security: A Survey," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10452–10473, July 2021.

[8] T. Kumar, E. Harjula, M. Ejaz, A. Manzoor, P. Porambage, I. Ahmad, M. Liyanage, A. Braeken, and M. Ylianttila, "BlockEdge: Blockchain-Edge Framework for Industrial IoT Networks," *IEEE Access*, vol. 8, pp. 154166–154185, August 2020.

[9] R. Xiong, W. Ren, X. Hao, J. He, and K.-K. R. Choo, "BDIM: A Blockchain-Based Decentralized Identity Management Scheme for Large Scale Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 22581–22590, December 2023.

[10] M. Zhaofeng, M. Jialin, W. Jihui and S. Zhiguang, "Blockchain-Based Decentralized Authentication Modeling Scheme in Edge and IoT Environment," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2116–2123, 15 February 2021.

[11] M. R. Ahmed, A. K. M. M. Islam, S. Shatabda, and S. Islam, "Blockchain-Based Identity Management System and Self-Sovereign Identity Ecosystem: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 113436–113481, October 2022.