ABDM: Anonymity-Based Big Data Management for Protecting Healthcare Data from Privacy Breach

Jinyong Kim, Jaehoon Jeong, Jeonghyeon Kim, and Joomin Kim

Abstract

As the healthcare industry has continued to develop and the medical data becomes digitized, the risk of data leakage from attacks grows. All medical data is encrypted and stored to prevent data leakage, but from big data perspective, it is inefficient because the encryption of this medical data lowers the big data processing performance. To address this problem, this paper proposes an Anonymity-based Big Data Management (ABDM) for protecting healthcare data from privacy breach without compromising on performance. The idea of ABDM is to separate and store identity data and healthcare data in databases on different cloud servers. By storing the identity data and healthcare data separately in databases on different cloud servers, hackers are unable to identify whose healthcare data is obtained until they obtain both the identity data and healthcare data. Through experiments in a public cloud, ABDM outperforms existing healthcare data management systems by two times speed.

Introduction

Recently the importance of medical services is increasing due to the aging population and the increase in numbers single-person households. In addition, the healthcare industry, which combines medical care and information technology (IT) for personal health, is growing rapidly with the advancement of medical-related technologies and equipment [1]. Using data collected through healthcare equipment, healthcare provides diagnosis, treatment, and follow-up regardless of time and place, as shown in Fig. 1.

However, with the development of the smart healthcare industry and the digitalization of medical data, the risk of attacks and data leakage increases [2]. In the existing cyberattack area, there have been many forms of financial damage by targeting data such as personal computers (PC) or servers, but cyber-attacks in the digital medical field can even threaten human lives. In other words, data in digital medicine is very sensitive information that can endanger human life, and therefore can never be exposed. To prevent data leakage, medical data can be encrypted and stored in databases. However, from the big data perspective, encrypting and storing all medical data is very inefficient as big data processing performance is degraded by encrypting that medical data.

To address this problem, we propose an Anonymity-based Big Data Management (ABDM) for protecting healthcare data from privacy breach without any performance degradation. The core idea behind ABDM is to separate and store identity data and healthcare data in databases on different cloud servers. By storing the identity data and healthcare data separately in databases on different cloud servers, a user can only get meaningful information if they have both the identity data and healthcare data. In other words, if they steal only identity data or healthcare data, the hacker cannot get meaningful information from the leaked data. To obtain meaningful information, the hacker should steal both identity data and healthcare data.

The proposed ABDM system consists of an identity database, a healthcare database and a key database. The identity database is a database that stores information about the patient's identity. A healthcare database is a database in which healthcare information of a patient is stored. A key database is a database that stores a key that can link identity and healthcare data. For a hacker to steal meaningful information, they have to steal data from all databases on different cloud servers in order to obtain the relevant information.

The novelty is the introduction of a key database to make the healthcare database more secure against a hacker's intrusion attack to the healthcare database. The number of key database tables can be controlled according to the required security level. That is, the more key database tables means the more secure against a hacker's intrusion attack.

The main contributions to this paper are summarized as follows:

- **Data Protection by Using Anonymity Without Data Encryption for Confidentiality:** In order to protect data, we propose a healthcare data management architecture by using anonymity instead of data encryption for confidentiality (see the section "Overview and Design of ABDM"). Note that to the best of our knowledge, ABDM is the first

Digital Object Identifier: 10.1109/MNET.2024.3476380
Date of Current Version: 14 January 2025
Date of Publication: 8 October 2024

Jinyong Kim, Jaehoon Jeong (corresponding author), and Jeonghyeon Kim are with the Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, Gyeonggi-do 16419, Republic of Korea; Joomin Kim is with Huinno Company Ltd., Seoul 06011, Republic of Korea.

0890-8044/24©2024IEEE

attempt to protect data by using anonymity rather than encrypting data to maintain confidentiality.

- **No Threat of a Brute-Force Attack:** By utilizing anonymity for confidentiality, there is no threat of a brute-force attack. Because even if some of our databases are breached by a brute force attack, it is safe as there is no information that could identify healthcare data. Our healthcare databases can only be attacked by stealing data from all databases (see the section "Overview and Design of ABDM").
- **Data Storage Without Encryption:** This healthcare data management scheme stores healthcare data as plain text without using an encryption algorithm to maintain confidentiality. Due to the storage of healthcare data in plain text without encryption, there are no encryption and decryption processes when storing and retrieving the data (see the section "Workflow of ABDM"). As a result, ABDM is faster and lighter than other data management schemes that use encrypted data storage (see the section "Performance Evaluation").
- **Implementation and Feasibility Evaluation of the Proposed Healthcare Data Management:** To show the feasibility of the proposed healthcare data management, we implemented the system in a real network environment using the cloud-hosted MongoDB service (i.e., MongoDB Atlas) [3] on Amazon Web Service (AWS) [4]. The performance of ABDM is evaluated as per convention in healthcare data management systems using data encryption in experiments (see the section "Performance Evaluation")

The rest of this paper is organized as follows. The background and related work on healthcare data management against privacy violations are provided in the section "Related Work." The section "Overview and Design of ABDM" describes the architecture of the proposed healthcare data management using anonymity against privacy breach. The section "Workflow of ABDM" explains the workflow of ABDM against privacy information breach. The section "Performance Evaluation" describes the implementation and feasibility of ABDM and evaluates the performance of the ABDM through comparisons with other baseline systems. The section "Research Challenges" discusses research challenges for ABDM. Finally, the section "Conclusion" concludes this paper along with future work.

Related Work

Security Management for Healthcare Data: There are various approaches to securely managing healthcare data based on a cloud system [5]. Fabian et al. [6] proposed a multi cloud-based framework that can securely share secret cryptographic data and encrypted healthcare data between organizations. Our framework utilizes a hash function instead of encryption algorithms to protect healthcare data, which speeds up data handling.

Among the cloud-based approaches, the latest approaches apply a blockchain technology [7] to healthcare data management systems. Azaria et al. [8] proposed a MedRec framework which uses a log of medical history as a block unit in a blockchain system and allows system administrators to reach a consensus on storing new data or accessing stored data. Omar et al. [9] also proposed the MediBchain framework, which encrypts all medical data and builds a block unit to store the encrypted medical data. However, as mentioned in [7], while it is imperative to apply blockchain technology to existing management systems, practical challenges such as the difference in characteristics between transaction data and healthcare data need be considered to fit the blockchain system to healthcare data storage.

The core idea behind ABDM is to separate and store identity data and healthcare data in databases on different cloud servers.

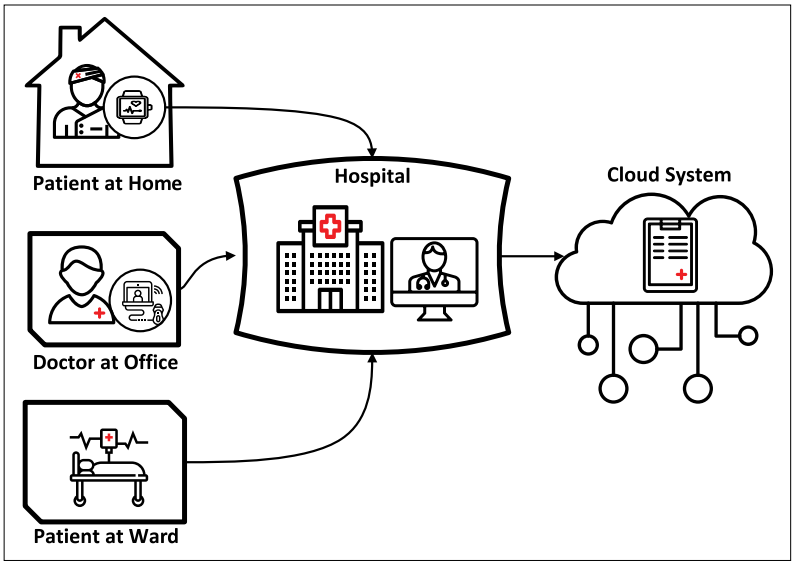

FIGURE 1. Online healthcare services through cloud computing.

K-Anonymity and Data Partitioning: As ABDM splits the entire healthcare database and saves the database fragments in each storage (e.g., cloud storage in our case), this technique appears to be similar to data management techniques (e.g., K-anonymity [10] and data partitioning [11], [12]) which are proposed for the protection of the healthcare data privacy or for efficient data management. However, rather than anonymizing the data or effectively managing the database, the purpose of our approach is to ensure data confidentiality and prevent an attacker from obtaining meaningful information unless the attacker has access to all parts of the database. Also, ABDM supports the restoration of a user's original data, but K-anonymity does not support this restoration.

K-anonymity [10] is a model that re-identifies the original database in order to keep the data anonymous and also not be able to distinguish individual data from others. K-anonymity is similar to our framework in which K-anonymity creates at least k numbers of each data to prevent a linkage attack [13] and ABDM splits the database into k database parts and stores them in each store so

The goal of ABDM is to protect healthcare data through anonymity from breaches of privacy information without any performance degradation.

that attacker cannot obtain significant information without accessing all k parts of the database.

Data Partitioning is a popular technique to split and store the database in order to efficiently manage and access data. Among the data partitioning techniques (e.g., horizontal partitioning [12]), vertical partitioning [11] is similar to our framework because we also split the database into the data of each column. However, in order to access data efficiently as a data partitioning technique, each part of the database contains identification data, which means that each of the data can be identified. On the other hand, in ABDM, each separate database in each store does not contain any identity information, except for the identity database. In other words, the attacker cannot obtain any meaningful information from a single separate database.

Ciriani et al. proposes a combined scheme of fragmentation and encryption to protect privacy in data storage [14]. The idea of the proposed scheme is to split a database table into multiple database tables such that a subset of fields cannot expose privacy data (e.g., illness) for a specific person (e.g., patient). It uses salt (as a random value) and encryption value (i.e., the encryption result of other fields' values) for privacy-preserving query processing. It is not efficient for time series data for a patient since the salts and encryption values need to be computed in the case of the addition of a new record per patient. On the other hand, our ABDM does not require the computation of hash values for the addition of a new healthcare data record per patient.

Overview and Design of ABDM

This section presents the purpose and overview of our ABDM for managing big data in cloud healthcare using the anonymity and describes the main components of ABDM.

Overview

In this subsection, we describe the goal and overview of ABDM. The goal of ABDM is to protect healthcare data through anonymity from breaches of privacy information without any performance degradation. Fig. 2 shows the architecture of our ABDM. The architecture of the databases of our ABDM is a collection of physically separated databases (i.e., a distributed database). This collection consists of three kinds of databases such as an identity database, multiple key databases, and a healthcare database. The identity database contains a patient's identity information such as name, gender, and birth date along with an identity hash value. This

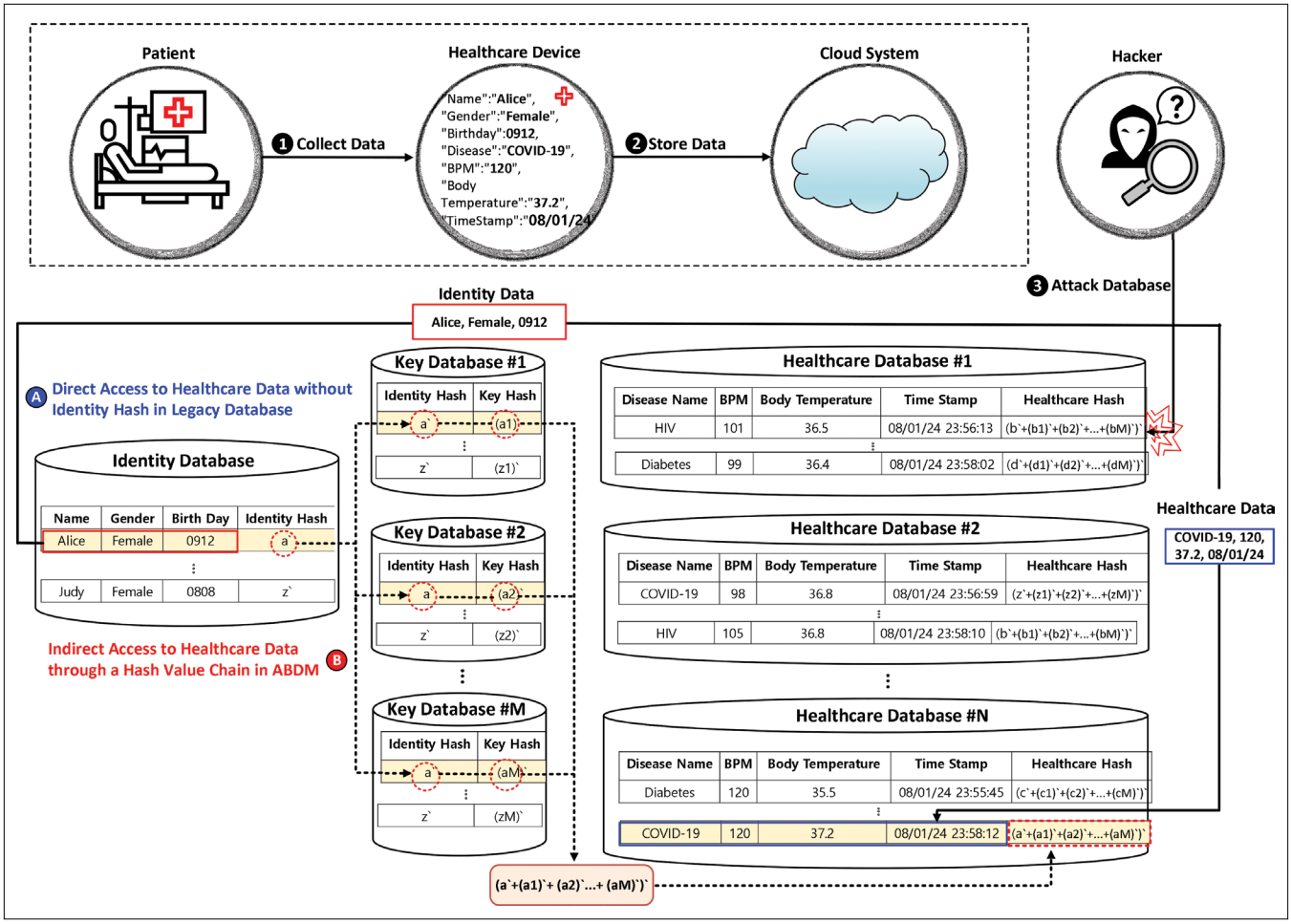

FIGURE 2. Cloud-based healthcare big data management using an anonymity architecture.

identity hash value is computed by a predefined hash function with a random string. Key databases contain an identity hash value from the identity database and a key hash value per data entry. Like the identity hash value, the key hash value is computed by a predefined hash function with a random string. A healthcare database contains a patient's healthcare information such as disease name, beats per minute (BPM), body temperature, time stamp along with a healthcare hash value. The healthcare hash value is a key used to retrieve a data entry and is computed by an identity hash value and its associated key hash values.

Our key idea is to use a chain of hash values (called "hash value chain"), which is a sequence of an identity hash value, multiple key hash values, and a healthcare hash value, so that a user (e.g., doctor and administrator) can access the healthcare data of a patient. To access the healthcare data of a patient, a hacker needs to retrieve all the hash values in different databases that are stored separately as a distributed database. Our ABDM can choose the number of key databases per patient elastically according to the required security level. That is, the more number of key databases means the stronger security level against hackers.

The assumption is that hackers can try to access any databases such as identity database, key database(s), and healthcare database. For the hackers to access healthcare data from a patient, they should be able to access all the corresponding key databases as intermediate databases to locate the relevant healthcare database records. If at least one of those key databases cannot be accessed by the hackers, they cannot locate the wanted records from the healthcare database. Thus, this key database provides a user with the "hash value chain" to link identity data and healthcare data to adjust the security level with the number of key databases per patient according to the user's security demand. Note that we can use multi-factor authentication (e.g., one time password) to prevent hackers from attacking in the same way as normal users. In the next subsection, we will describe the main components of ABDM to protect against privacy information breaches.

Components

In this subsection, we describe the main components of ABDM, a cloud-based healthcare big data management system using anonymity against privacy information breaches. As shown in Fig. 2, ABDM consists of the following main components: identity database, key database and healthcare database.

Identity Database: The identity database is a database that stores identity information (e.g., name, gender, and birthday) and a hash value (i.e., an identity hash value) of identity data. The identity information is data that can know the owner of the healthcare data. The identity hash value is the hash value of a randomly generated string (i.e., randomized strings of characters), and the string length and hash value can be determined by the ABDM system manager. With this generated hash value from the identity database, we can retrieve key data related to the identity data from the key database or healthcare data related to identity data from the healthcare database.

Key Database: The key database is the database that stores the identity hash value and the key hash value. The first hash value (i.e., identity hash value) is the hash value generated from the identity database. With this hash value, the identity of the key hash value can be obtained. The second hash value (i.e., key hash value) is the hash value of a randomly generated string, and the string length and the hash value can be determined by the ABDM system manager, as is the hash value generated in the identity database. With the identity hash value and the key hash value, we can retrieve the healthcare data in the healthcare database using the sum hash value of the identity hash value from the identity database and the key hash value generated from the key database, respectively. Note that we can increase the security level by increasing the number of key databases.

Healthcare Database: The healthcare database is a database that stores private and confidential healthcare information, which should not be disclosed, and the corresponding hash value (i.e., a healthcare hash value). Healthcare information is data such as disease names, beats per minute (BPM) and body temperature that are measured using various types of wearable devices. The healthcare hash value is the hash value of the sum of the hash values of an identity database and the key hash value of the keys. Thanks to the generated healthcare hash value, we can retrieve the healthcare data related to that identity. Note that we can apply anonymous methods (e.g., K-anonymity [10]) to our ABDM in order to protect private data against a hacker's prior knowledge (e.g., physical features and possessed devices) about healthcare data.

Workflow of ABDM

This section describes the cloud-based big data management workflow in healthcare using anonymity to prevent privacy information breaches. The ABDM workflow consists of data storage and data retrieval.

Healthcare Data Storage

This subsection describes the ABDM healthcare data storage workflow for managing big data in the cloud using anonymity. Fig. 3 shows the healthcare data storage workflow of ABDM. As shown in the figure, our ABDM consists of an identity database, key databases and healthcare databases. The numbers of identity databases, key databases, and healthcare databases are 1, M, and N, respectively. The identity database column contains name, gender, birth day and identity hash. The key databases column consists of the identity and the key hash. The healthcare databases column contains the disease name, beats per minute, body temperature, timestamp and healthcare hash. The detailed procedure for storing healthcare data is as follows:

1. A healthcare wearable device (e.g., watch, band and pad) generates and transmits patients' data (i.e., Judy, Female, 0808, COVID-19, 130, 37.6, and 2024/08/01…) to the server.

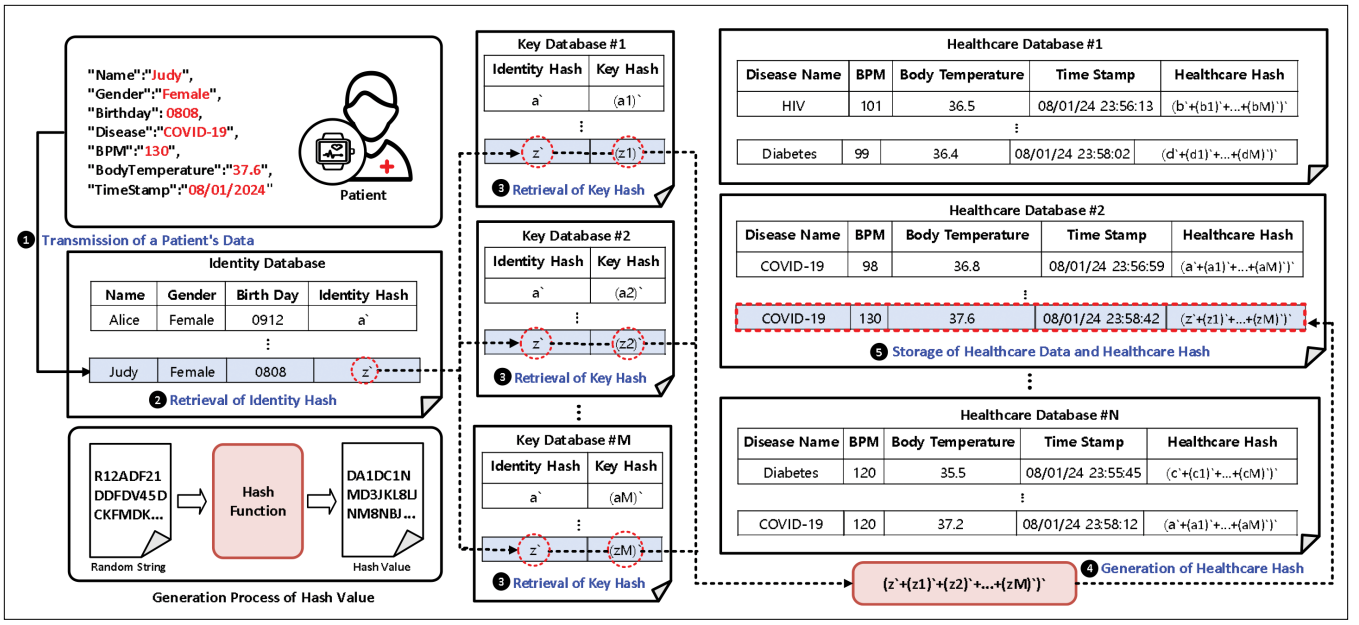

FIGURE 3. Healthcare data storage of cloud-based healthcare big data management using anonymity.

2. The ABDM system then retrieves the matching identity data (i.e., Judy, Female, and 0808) from the identity database based on the transmitted data.
   a. If the matching identity data is not retrieved from the identity database because the identity data is related to a new patient, the ABDM system generates the identity hash value (i.e., z′) of a randomly generated string related to the identity data and stores the transmitted identity data and generated identity hash value for the identity database. Similarly, the ABDM system generates the key hash values (i.e., (z1)′, (z2)′, …, and (zM)′) in the same way as the identity database and stores the identity hash value and key hash values in each key database.
   b. If matching identity data is retrieved from the identity database because the identity data is existing patient, the ABDM system retrieves the identity hash value (i.e., z′) along with the identity data. Then, based on the retrieved identity hash value, the ABDM system retrieves the key hash values (i.e., (z1)′, (z2)′, …, and (zM)′) matching the identity hash value from all key databases.
3. The ABDM system calculates the healthcare hash value (i.e., (z′+(z1)′+(z2)′+…+(zM)′)′) of the sum of the identity hash value and the key hash values.
4. Finally, the ABDM system stores both the transmitted healthcare data (i.e., COVID-19, 130, 37.6, and 2024/08/01 …) and the calculated healthcare hash value (i.e., (z′+(z1)′+(z2)′+…+(zM)′)′) into one randomly selected healthcare database among healthcare databases. Note that if there is a hash collision during all hash generation processes, the ABDM regenerates the collided hash value.

Healthcare Data Retrieval

This subsection describes the healthcare data retrieval workflow in cloud-based healthcare big data management using anonymity. Fig. 4 shows the healthcare data retrieval workflow in ABDM. The detailed procedure for retrieving healthcare data is as follows:

1. For retrieval of patient's healthcare data, we first retrieve the identity hash value (i.e., z′) matching the identity data (i.e., Judy, Female, and 0808) of the patient in the identity database.
2. Using the retrieved identity hash value, the ABDM system retrieves the key hash values (i.e., (z1)′, (z2)′, …, and (zM)′) matching the identity hash value from all key databases.
3. Using the identity hash value and the key hash values thus obtained, the ABDM system can calculate the healthcare hash value (i.e., ((z′+(z1)′+(z2)′+…+(zM)′)′) from the sum of the identity hash value and the key hash values.
4. Finally, with the calculated healthcare hash value, we can obtain the patient's healthcare data corresponding to the calculated healthcare hash value (i.e., (z′+(z1)′+(z2)′+…+(zM)′)′), and we can analyze the patient's condition.

In the case of data modification, as in the above method, you can easily modify it by searching for the data and changing only the relevant data. This is because the hash values such as identity hash value, key hash values, and healthcare hash value are generated by random strings rather than a patient's data.

Note that for the protection against a hacker's intrusion, our ABDM utilizes three kinds of separate databases (i.e., identity database, key databases, and healthcare database) using a hashing algorithm. To access each database, a user needs to know the host name (or IP address), ID, and password for it. So that hackers can know the

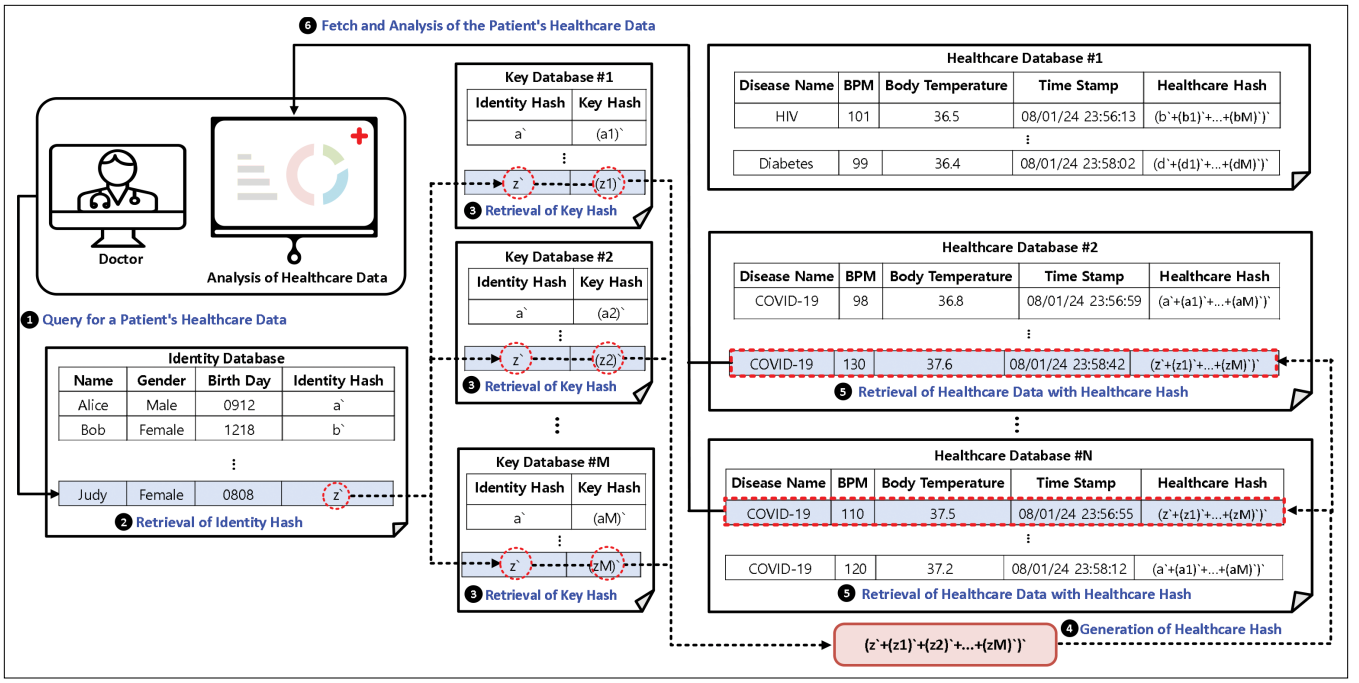

FIGURE 4. Healthcare data retrieval of cloud-based healthcare big data management using anonymity.

healthcare data with a user's identity, they should know such a login information per each database. Without this login information, they cannot access each database, so we can claim that our ABDM has at least the same security level as the encryption method. Moreover, our ABDM requires more than one key database from a user so that the user can generate a hash value with all the predefined key databases. Thus, our ABDM can provide a stronger security protection than a single database with an encryption method along with a user's ID and password.

Performance Evaluation

This section explains how to evaluate the ABDM's performance. A comparison of the ABDM system with other baseline encryption approaches (i.e., DES, 3DES, and AES256) and a plain-text approach was performed. All these encryption approaches above use the DES, 3DES and AES256 encryption algorithms, respectively, to encrypt and decrypt only healthcare data (i.e., heart rate).

The plain-text approach does not use any encryption algorithm. Those four baselines use a single table for user data and healthcare data. The ABDM system is the healthcare data management system that does not use an encryption algorithm, but uses a hash algorithm (i.e., SHA3-512) to securely process healthcare data. Note that instead of the encryption of healthcare data, our ABDM uses the keychain between the identity database and healthcare database through the key database(s), as discussed in the section "Overview", so that the hackers may not identify the user's healthcare data directly.

In order to evaluate the performance of our ABDM in a real network environment, we implemented it in a real network environment using a cloud-hosted MongoDB service (i.e., MongoDB Atlas) [3] on Amazon Web Service (AWS) [4]. Thus, this MongoDB Atlas handles the three kinds of databases such as identity, key, and healthcare databases.

We used a cluster of three MongoDB servers (i.e., vCPU: 2, Memory: 2GB, Storage: SSD, and Network performance: up to 5Gbps). The evaluation was performed by experimenting with the healthcare data management systems implemented in Python on the cloud computing platform, the confidence interval of the experiments was 95%. The evaluation settings (i.e., performance metrics and parameters) are as follows:

- **Datasets:** We used the healthcare data (i.e., heart rate data at 250 beats per second) actually used by the HUINNO company (https://huinno.com/en/) for performance evaluation.
- **Performance Metrics:** In order to evaluate the algorithm performance, the data storage time and the data retrieval time were used as the performance metrics.
- **Parameters:** In order to evaluate the algorithm performance in various environments, the impact of the number of data (i.e., 100,000, 200,000, 300,000, 400,000, 500,000, 600,000, and 700,000) was investigated.

Note that the source codes of our implementation is available at https://github.com/jaehoonpauljeong/ABDM.

Impact of the Amount of Data

This section investigates the impact of the amount of data (i.e., the number of data entries) to be stored or retrieved for performance evaluation. Fig. 5 shows the data retrieval time depending on the amount of data using the baseline systems and ABDM using three key databases. As expected, the time and the memory usage for data retrieval for all four systems

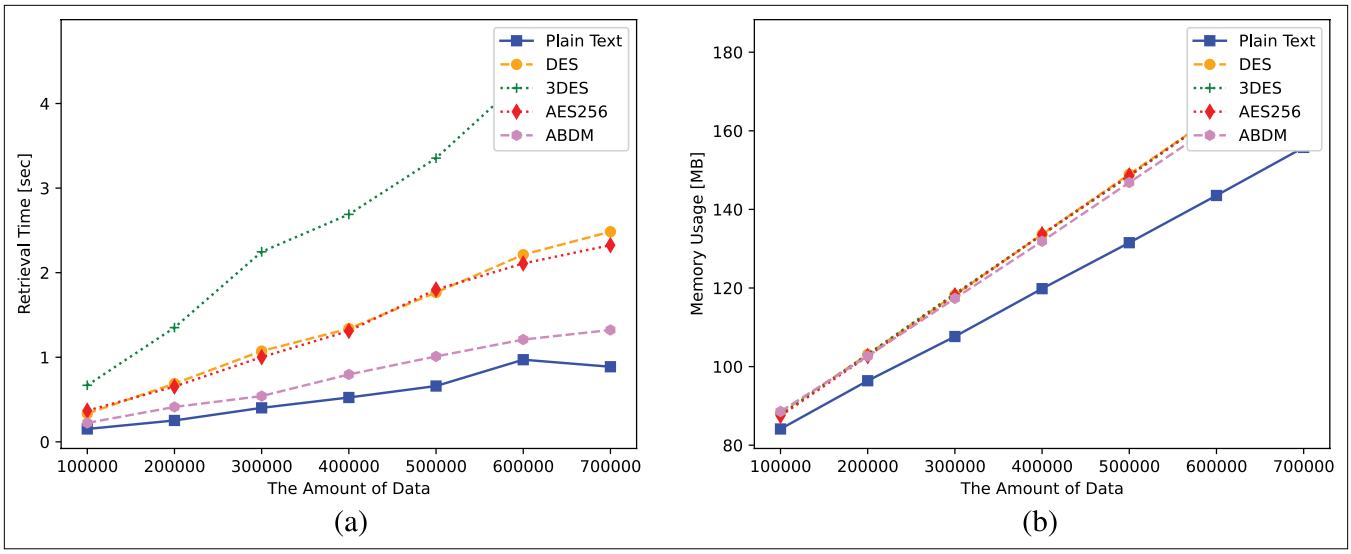

FIGURE 5. Impact of the amount of data. a) Data retrieval time versus the amount of data. b) Data retrieval memory usage versus the amount of data.

ABDM provides how to protect healthcare data by using anonymity, rather than encrypting that data to maintain confidentiality.

increased as the amount of data increased. The time and the memory usage for the data retrieval time tend to increase with increasing data volume; this is because more data requires more time to retrieve the healthcare data from a database. As shown in Fig. 5(a) and (b), our ABDM system outperforms other data encryption management systems (i.e., DES, 3DES, and AES256) and has almost no performance degradation compared to the plain-text approach in terms of data retrieval time and memory usage according to the number of data entries, respectively.

The reason our ABDM and the plain-text approach have almost the same performance is that the network delay to access the identity database and key databases is relatively smaller than the retrieval delay in the healthcare database. Note that the MongoDB Atlas automatically allocates the key databases to multiple servers in the cluster system. It should be noted that the number of the key databases does not affect the retrieval performance because the access to them is performed in parallel.

Data encryption management systems need to take more time and memory usage for data retrieval than our data management system because the systems need to encrypt and decrypt the healthcare data in order to process it securely. These data encryption and decryption require more time and memory for data retrieval than using three kinds of databases (i.e., an identity database, key databases, and a healthcare database) in our ABDM in line with the increasing amount of data.

Research Challenges

This section discusses some of the research challenges facing an ABDM to be deployed in the industry.

- **Combination of Relational Database and Non-Relational Database:** In the ABDM, we considered a non-relational database [15] for big data, not a relational one. We can use both relational database and non-relational database. For the efficient management of the identity database and key database, both the databases can be implemented by a relational database (e.g., MySQL). On the other hand, for the operational efficiency of big data such as healthcare data, the healthcare database can be implemented by a non-relational database (e.g., MongoDB).
- **Impact of the Number of Key Databases:** An administrator can add or remove key databases to increase security levels or speed-up processing. If the key databases are added or removed, all hash values corresponding to healthcare data in the healthcare databases must be computed again as an initialization process. However, each time the key databases are added or removed, it is inefficient to perform the initialization process for all hash values as the process may take a long time, depending on the amount of healthcare data. To mitigate this inefficiency, the administrator needs to select an optimal number of key databases.
- **Separating Sensitive Data:** ABDM can further be used to protect data, not only for healthcare data, but also for sensitive data (e.g., disease) from a user's collected data (e.g., a combination of a name, gender, birthday, and disease as shown in Fig. 2) that should not be exposed. However, in this case, it is difficult to properly separate identity data for the identity database from sensitive data for the healthcare database. In order for ABDM to be used not only for healthcare data but also for sensitive data, it is necessary to study how to properly separate identity data from sensitive data. Also, to avoid a background-knowledge

attack, the healthcare database should not have a record structure to allow for privacy information leakage, that is, the inference of a user's disease by the user's background knowledge such as the user's disease name, blood type, height, and weight.

Conclusion

This paper presented an Anonymity-based Big Data Management (called ABDM) against the breach of privacy information (e.g., healthcare data). ABDM provides how to protect healthcare data by using anonymity, rather than encrypting that data to maintain confidentiality. This paper demonstrated the feasibility of ABDM using anonymity by deploying a prototype to cloud servers in MongoDB Atlas in AWS. We showed that ABDM outperformed existing healthcare data management systems using the encryption algorithms (i.e., DES, 3DES, and AES256) while it has almost the same performance with the plain-text approach. In future work, we will improve ABDM in terms of performance and management efficiency, so that it can be more actively used in the healthcare industry.

Acknowledgment

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) in South Korea under Grant RS-2022-II221199 and Grant RS-2022-II221015.

References

[1] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.
[2] *Top 10 Biggest Healthcare Data Breaches of All Time*. Accessed: Oct. 6, 2024. [Online]. Available: https://digitalguardian.com/blog/top-10-biggest-healthcare-data-breaches-all-time
[3] *MongoDB Atlas*. Accessed: Oct. 6, 2024. [Online]. Available: https://www.mongodb.com/atlas/
[4] *Amazon Web Services*. Accessed: Oct. 6, 2024. [Online]. Available: https://aws.amazon.com/
[5] A. K. Pandey et al., "Key issues in healthcare data integrity: Analysis and recommendations," *IEEE Access*, vol. 8, pp. 40612–40628, 2020.
[6] B. Fabian, T. Ermakova, and P. Junghanns, "Collaborative and secure sharing of healthcare data in multi-clouds," *Inf. Syst.*, vol. 48, pp. 132–150, Mar. 2015.
[7] C. Esposito et al., "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan. 2018.
[8] A. Azaria et al., "MedRec: Using blockchain for medical data access and permission management," in *Proc. Int. Conf. Open Data (OBD)*, 2016, pp. 25–30.
[9] A. A. Omar et al., "Privacy-friendly platform for healthcare data in cloud based on blockchain environment," *Future Gener. Comput. Syst.*, vol. 95, pp. 511–521, Jun. 2019.
[10] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.
[11] S. Navathe, S. Ceri, G. Wiederhold, and J. Dou, "Vertical partitioning algorithms for database design," *ACM Trans. Database Syst.*, vol. 9, no. 4, pp. 680–710, Dec. 1984.
[12] S. Ceri, M. Negri, and G. Pelagatti, "Horizontal data partitioning in database design," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 1982, pp. 128–136.
[13] M. M. Merener, "Theoretical results on de-anonymization via linkage attacks," *Trans. Data Privacy*, vol. 5, no. 2, pp. 377–402, Aug. 2012.
[14] V. Ciriani et al., "Combining fragmentation and encryption to protect privacy in data storage," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 3, pp. 1–33, Jul. 2010, doi: 10.1145/1805974.1805978.
[15] A. Singh, "NoSQL: A new horizon in big data," *Int. J. Sci. Res. Sci., Eng. Technol.*, vol. 2, no. 2, pp. 1183–1188, Apr. 2016.

Biographies

Jinyong (Tim) Kim (Member, IEEE) (timkim@skku.edu) is a currently a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Sungkyunkwan University, Republic of Korea.

Jaehoon (Paul) Jeong (Member, IEEE) (pauljeong@skku.edu) is currently a Professor with the Department of Computer Science and Engineering, Sungkyunkwan University, Republic of Korea.

Jeonghyeon (Joshua) Kim (Student Member, IEEE) (jeonghyeon12@skku.edu) is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Sungkyunkwan University, Republic of Korea.

Joomin (Joshua) Kim (Member, IEEE) (joshua007k@gmail.com) has been a Chief Technology Officer at Huinno Company Ltd., Republic of Korea, since 2020.