Data Storage Protection of Community Medical Internet of Things

Ziyang Zhang, Fulong Chen(✉), Heping Ye, Junru Zhu, Cheng Zhang, and Chao Liu

Anhui Provincial Key Laboratory of Network and Information Security,
Anhui Normal University, 189 Jiuhua South Road, Wuhu 241002,
Anhui, People's Republic of China
long005@ahnu.edu.cn

Abstract. With the improvement of people's living standard, people have put forward higher requirements on medical services. The effective combination of the traditional community medical systems and the modern Internet of things technologies can help to build a community medical Internet of things, which involves a large number of important information for health care and patient staff, and these information face the risk of privacy disclosure and information damage. From the point of view of data storage, we proposes a data storage protection method for preserving privacy data in the community medical Internet of things. Through analyzing the data integrity and security of the practical scheme, it is proved that the medical data can be protected effectively in the process of storage.

Keywords: Community medical care · Internet of things · Privacy data · Storage protection

1 Introduction

The rapid economic development has led to the deterioration of the natural environment upon which the survival of people's health under unprecedented threat. Various non-predictability of diseases have sprung up on the patients so that the patient's illness makes it painful bring the demand for medical services growing. However limited traditional medical service resources and uncertainty treatment time urge people to begin to look for better health service to make up for the lacking of available resources.

In Community Medical Internet of Things (CMIoT), due to the huge amount of heterogeneous medical data, extensive medical data sources, and various identification information which involve user privacy, once medical data loses or tampers, some privacy leakages resulting in catastrophic loss will occur [1]. [2,3] have presented that tags would be scanned while users were not aware of what readers would do, it would easily bring into the destruction of personal privacy, and it would cause the items of information suffering from attacking between local servers and remote servers.

J. Wan et al. (Eds.): CloudComp 2016, SPNCE 2016, LNICST 197, pp. 102–110, 2018.
https://doi.org/10.1007/978-3-319-69605-8_10

Data storage faces a paradox: encryption data cannot be efficiently processed, and the security and privacy of non encrypted data can not be guaranteed. Therefore, it is urgent to need a kind of effective privacy protection method to ensure the safety of medical data storage in the controllable range. Mni [4] proposed various models of medical data from production to storage. Ateniese [5] proposed a distributed data secure storage scheme in which data is encrypted using the symmetric keys, and the symmetric keys are encrypted using public key. However, there exists the risk of collusion between the malicious server and malicious users, leading to the disclosure of the file encryption key. Vimercati [6] proposed a method for secure storage of data by a non trusted server key derivation method, in which each file is encrypted with a symmetric key, each user has a private key, and in order to authorize, data owners create public tokens for users so that authorized users can use their private key to derive the decryption key of the specified file from the tokens. The key number of the scheme is too large and the complexity of the operation is linear with the number of users so as unable to effectively extend. Kamara and Lauter [7] studied a kind of abstract public cloud storage encryption framework composed of data processing module, data verification module, token generation module and credential generation module, in which the storage data controlled by the owner is authorized to be accessed via token generated by the token generator and to be decrypted through credentials generated by credential generator, and their security is controlled by the password mechanism. The data protection technology based on VMM is proposed in [8] where the operating system and the distributed file system are isolated to protect data security by using the Daoli virtual security monitoring system and the SSL secure transmission module. A kind of homomorphic encryption algorithm [9] is designed to realize data encryption and decryption with mixed operations of cector and matrix operation, supports for fuzzy retrieval of encrypted data, and can be better to perform the homomorphic addition and subtraction operation. The downside of this approach is the low efficiency in cipher text retrieval and homomorphic multiplication/division.

Wang [10] studied and proposed a secure storage of outsourced data in the cloud environment. In the method, the storage efficiency is improved by dividing the file into blocks and the data security is ensured for each data block using a different key encryption. Because of the need to spend a lot of cost data encryption and key management, the scheme has a lot of problems. A reliable data protection and destruction method with the help of a trusted platform was proposed by Zhang [11]. He designed a virtual monitor as the trusted third party responsible for monitoring and protecting the user's privacy data, and destroying user data in accordance with user requirements, even if the cloud server's super administrator can not bypass the protection of user privacy data. It is obvious that the method is too high requirements for the reliability of the hardware and software, and the actual situation is difficult to meet. A storage model of cloud computing was designed in [12] where the trusted third party server is responsible for the isolation of user privacy data and general data, and thereby realizes the protection of user's privacy information. However, in this scheme, when the data is stored, the two times of data partitions and matrix operation make the storage efficiency low so that it is difficult to use and expand on a large scale.

2 Architecture of Community Medical Internet of Things

The CMIoT is achieved in one community, as shown in Fig. 1. In the CMIoT, Data from a sensor is sent toward the nearest gateway belong to some place such as home, community public area, community health center or hospital, and then the data is transmitted to the nearest community router. Connection is built between the gateway and the database server of cloud data center through wireless network. In the end, the application server of data center provides the resolved data to users with mobile terminals or PC terminals. Data transmission integrates a variety of communication means. Sensors in one place establish communication via wireless self-organized network, and data in the gateway transmit through wireless local area network or mobile network.

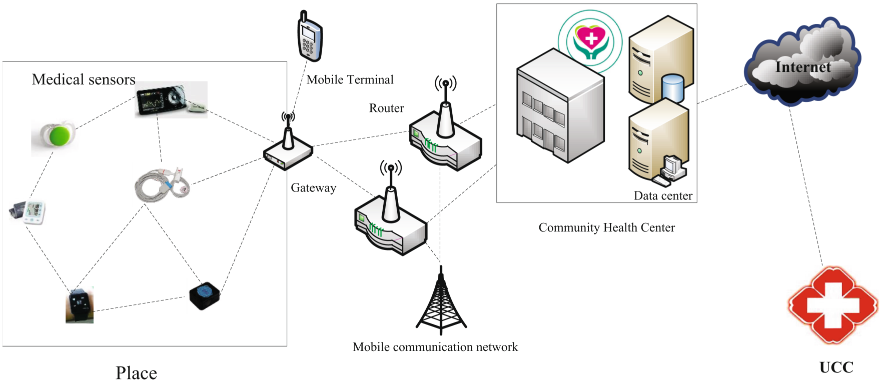

Fig. 1. Architecture of community medical Internet of things.

3 Storage Protection Model

The storage server is composed of the storage control center and the file system. After medical data generated by different systems and modules in CMIoT are transmitted to the server node, they firstly enter the storage buffer for unified processing of the control area in the storage module. The control center and the file system with a message queue exchange information through a message channel. If they communicate successfully and the current file system is free, the server can store the current data stream. Each file system independently enjoys and controls the communication link in order to achieve the purpose of distributed data storage.

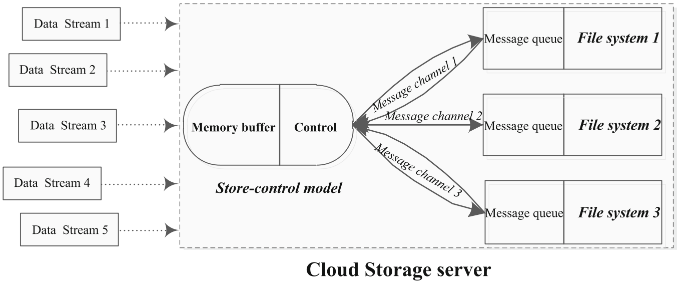

Fig. 2. Storage server model.

As shown in Fig. 2, at some point, there may be a large number of medical data to enter the storage server so that the control area can not handle them immediately. At this time, the server sends the data to the buffer storage buffer, and after the completion of the current data processing tasks in the control area, the data is extracted from the buffer and processed into the storage link. The control area immediately detects the current file system, and once the idle file system is detected, the control area will store in order the buffer data into the free file system through the message channel.

For each data stream, before coming to the storage server, it is signed by the data source sender with the private key, and then according to the public key

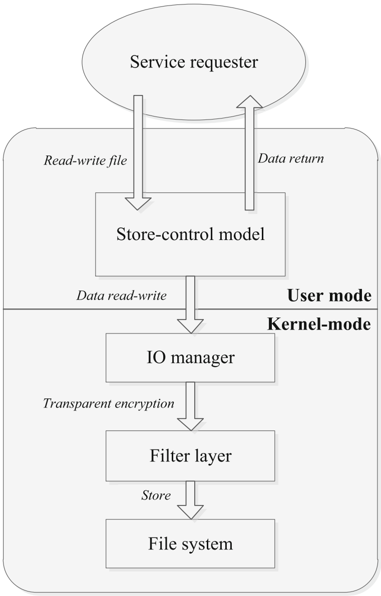

Fig. 3. Server security storage model.

of the data source sender, the control area decrypts it. If the data stream after decryption is detected without finding illegal operations, the storage control area uses its own symmetric key to encrypt the data stream, and store it to the file system. The storage control area has a specific process to process data streams so that the data stream is stored safely to the specified file system. Figure 3 is the server security storage model.

4 Storage Protection Scheme

After the storage server control area gets the data stream from the buffer, the server encrypts the data stream with its own private key and its own public key with the public key of the data source sender, and then generates a new encrypted data packet. Such data packet can only be decrypted using the sender's private key so as to get the public key of the storage server and decrypt the data stream for the reverse output of data stream. Using polling mode, the server queries whether the file system is idle or not, and stores the buffer data to the idle file system step by step. As shown in Table 1, some symbolic representations of data processing in the storage scheme are defined, the data storage process is shown as follows.

Step 1: The data stream sender encrypts the data stream plaintext P with the data stream encryption key K_C

$$C = E_{K_C}(P) \tag{1}$$

Step 2: The data stream sender encrypts the data stream encryption key K_C with the public key of the storage server P_{K-S}

$$K_K = E_{P_{K-S}}(K_C) \tag{2}$$

Table 1. The definitions of data storage symbols.

No.	Symbol	Definition
1	P_{K-R}	Public key of data source
2	S_{K-R}	Private key of data source
3	P_{K-S}	Public key of storage server
4	S_{K-S}	Private key of storage server
5	K_C	Encryption key of data stream
6	K_K	Encryption key of key
7	P	Plain text of data stream
8	C	Cipher text of data stream
9	$E_K(x)$	Encrypt data x with key K
10	$D_K(y)$	Decrypt data y with key K
11	$Sig_K(X)$	Sign data X with key K

Step 3: The data stream sender processes the plain text data with the hash function

$$P' = H(P) \tag{3}$$

and encapsulates the data stream plain text C, the hash value P' of the data stream plain text and the encryption key K_K of the key

$$D = C||P'||K_K \tag{4}$$

Step 4: Before the data is sent, the sender signs the encapsulated data D with its private key S_{K-S}

$$D' = Sig_{S_{K-S}}(D) \tag{5}$$

and sends it to the storage server.

Step 5: After the storage server receives the signed data D', it decrypts D' and the encryption key of the key with its own key S_{K-S}, and then uses the latter for decrypting P′

$$K_C = D_{S_{K-S}}(K_K) \tag{6}$$

$$P' = D_{S_{K-S}}(D') \tag{7}$$

$$P = D_{K_C}(P') \tag{8}$$

After the data stream enters the storage server, the control area of the storage server will decrypt the data packet for distributed storage. The data packet includes three parts such as header, body and remark as shown in Table 2. Afterwards, the control area encrypt the packaged data with the public key of the

Table 2. The symbol definitions of data package.

No.	Name	Definition
1	Header	Header of data file
	Data_Sequence	Sequence No. of data stream
	Data_FileNo	File system No. of data stream
	Data_type	Type of data stream
2	Body	Body of data file
	$Data_1$	Data body of data stream 1
	$Data_2$	Data body of data stream 2
	$Data_3$	Data body of data stream 3
3	Remark	Remark
	Data_Length	Length of data stream
	Data_En_Alg	Encryption algorithm

sender, and through the data channel established between the control area and the file system, using the data transmission mechanism based on transmission response, in other words, once the data transmission is interrupted, the data packet will be retransmitted, the control area stores all the data streams into distributed file systems as shown in Fig. 4.

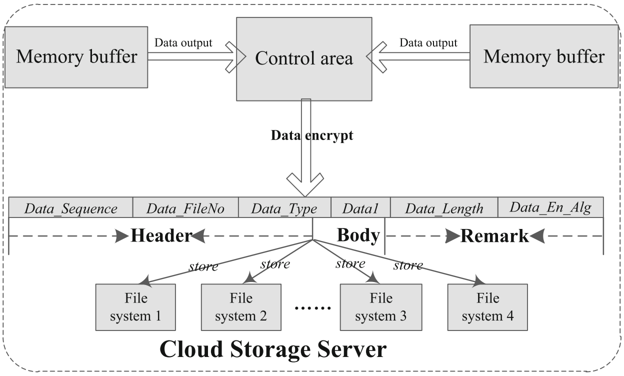

Fig. 4. Process of server storage.

5 Storage Security Analysis

5.1 Data Integrity

The scheme is provided with a control area module and a memory module in the storage server. The control area module is composed of a memory buffer and a control area. When data flows into a storage server, if the control area is processing some previous storage tasks, and unable to process incoming data stream in time, at this time the data stream will be stored in the memory buffer, so that the control area can complete the current tasks and turn to process current data storage. This ensures that a large amount of data can enter the storage server at the same time without being lost.

When the control area processes the new coming data streams, it will detect whether the storage modules of the file systems are idle or not, and once it finds a file system idle, it will transfer the data stream in a timely manner through the dedicated message channel. This avoids the situation that data can not be stored and may be lost due to unknowing whether the file system is busy or not, and the data integrity is guaranteed.

When the control area detects whether the file systems are idle or not, it will communicate with the file system in the form of a message queue. The communication channel between the control area and the file systems will not be blocked due to a lot of communication in a short time. It is very good to ensure the timely arrival of the feedback message and the integrity of the feedback information.

5.2 Data Security

Before entering the storage server, the data stream is signed by the private key, and then the control area uses the data source public key to decrypt the data and verifies its integrity. After that, the data is encrypted with a symmetric key and stored into the corresponding file system. This ensures the security of data in the process of arriving at the server and the file systems.

When a data stream is stored in a file system, it is interacted between the user state and the kernel state. Therefore, data storage is completed with to I/O manager and processed by a transparent encryption method. Once the data in the user state is requested to access, the kernel will receive that request, conduct an access request processing by verifying the role properties and finally complete data transmission in a transparent decryption method. The whole data request and feedback process is transmitted through the data encryption method. This also can protect the security of data.

6 Conclusions

Aiming at the problem of data security storage in the field of medical Internet of things, we design a secure data storage protection method. Through the design of a storage server model and secure storage model, we gives a complete data secure storage scheme. There are many security issues for data in the medical Internet of things, e.g., the secure transmission of medical Internet data and the classification of medical data privacy issues are the focus of research. The next step will be to explore and research the classification of privacy protection for medical data.

Acknowledgments. The authors would like to thank our colleagues and students in Engineering Technology Research Center of Network and Information Security at Anhui Normal University. We thank National Natural Science Foundation of China under Grant Nos. 61572036 and 61402014, University Natural Science Research Project of Anhui Province under Grant No. KJ2014A084, Wuhu City Science and Technology Project under Grant No. 2014cxy04 for support of this research.

References

1. Ye, H., Yang, J., Zhu, J., Zhang, Z., Huang, Y., Chen, F.: A secure privacy data transmission method for medical internet of things. In: Wan, J., Humar, I., Zhang, D. (eds.) Industrial IoT 2016. LNICSSITE, vol. 173, pp. 144–154. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44350-8_15
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
3. Medaglia, C.M., Serbanati, A.: An overview of privacy and security issues in the internet of things. In: Giusto, D., Iera, A., Morabito, G., Atzori, L. (eds.) The Internet of Things. Springer, New York, NY (2010). https://doi.org/10.1007/978-1-4419-1674-7_38

4. Mni, L., Zhang, Q., Tan, H.Y., et al.: Smart healthcare: from IoT to cloud computing. Sci. Sinica **43**(4), 515–528 (2013)
5. Ateniese, G., Fu, K., Green, M., et al.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur. **9**(1), 29–43 (2006)
6. Vimercati, S., Foresti, S., Jajodia, S., et al.: Over-encryption: management of access control evolution on outsourced data. In: Proceedings of the 33rd International Conference on Very Large Data Base, pp. 123–134 (2007)
7. Kamara, S., Lauter, K.: Cryptographic cloud storage. In: Sion, R., Curtmola, R., Dietrich, S., Kiayias, A., Miret, J.M., Sako, K., Sebé, F. (eds.) FC 2010. LNCS, vol. 6054, pp. 136–149. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14992-4_13
8. Hou, Q.H., Wu, Y.W., Zheng, W.M.: A method on protection of user data privacy in cloud storage platform. J. Comput. Res. Dev. **48**(7), 1146–1154 (2011)
9. HuangX, R.W., Gui, L., Yu, S., et al.: Privacy-preserving computable encryption scheme of cloud computing. Chin. J. Comput. **34**(12), 2391–2402 (2011)
10. Wang, W., Li, Z., Owens, R., et al.: Secure and efficient access to outsourced data. In: Proceedings of the 2009 ACM Workshop on Cloud computing security, pp. 55–66 November 2009
11. Zhang, F.Z., Chen, J., Chen, H.B., et al.: Lifetime privacy and self-destruction of data in the cloud. J. Comput. Res. Devel. **48**(7), 1155–1167 (2011)
12. Mao, J., Li, K., Xu, X.: Privacy protection scheme for cloud computing. J. Tsinghua Univ. (Sci. Tech.) **51**(10), 1357–1362 (2011)