

Security and integrity of medical log records management in the cloud storage

Raed Wishah *

Princess Sumaya University

Amman

Jordan

Azmi Shawkat Abdulbaqi [†]

Renewable Energy Research Center

University of Anbar

Anbar

Iraq

Abstract

Logs of medicinal events in Cloud Computing management have become crucial, particularly for medical event logs. Medical auditing, medical analysis, medical evidence, forensic medicine, and other reasons need the safe storage of medical log data in any hospital or medical facility for a certain time. At each level of the medical record management process, including the log creation stage, the transmission phase, and the storage step, the integrity and security of the medical Log record data are maintained in this work. Furthermore, medical log records often include patient information that should not be made public (sensitive information). This paper uses a Secure Provisional Cloud Log Records Management Algorithm (SPCLRMA) approaches to ensure security while preserving medical transaction history in the cloud over time. Security for TSCMEL is ensured in this study by encrypting medical log data before it is saved in the cloud storage (CldStg). They're also kept in packs for convenient access. This project was created utilizing the Java programming language and Microsoft Visual Studio.

Subject Classification: 91G20, 90B05.

Keywords: Secure cloud log records management algorithm (SCLRMA) (Modified RSA algorithm), Secure provisional cloud log records management algorithm (SPCLRMA), Medical cloud, Cloud storage (CldStg).

* E-mail: r.wishah@psut.edu.jo (Corresponding Author)

[†] E-mail: azmi_msc@uoanbar.edu.iq

Introduction

The machine or transactional data is information produced by any software or the infrastructure that underpins it, including medical log records. Information is stored in medical logs in the cloud. Information may be found in cloud-based medical logs. User identification, system identifier, operation completed, and time are examples[1]. Furthermore, certain medical logs include security features such as restricting access and history utilization. Identifying deviations from the user's typical usage pattern is important to analyze the user's behaviour. Medical records are stored in database systems to facilitate transaction recovery. In this scenario, two types of logs are utilized: reverse logs and redo logs [2]. Existing recovery protocols have several security issues that make them susceptible during log message delivery, and there is no security mechanism to validate the data received by the syslog daemon. External auditors may occasionally require access to medical log records for security log auditing. In order to preserve the past, the log data must be kept briefly before being rotated or erased[3].

These log records are utilized to recover data lost due to transaction failures. Cloud-based logging is a low-cost alternative for hospitals and medical companies. Medical data integrity and confidentiality are the key issues with CldStg [4]. This paper addresses these issues by presenting encryption algorithms that provide accuracy, tamper resistance, secrecy, and integrity. New strategies for securing medical log records in the cloud are devised and applied in this paper. The medical log data is consolidated into packs, encrypted, and saved. The major benefit of encrypting medical log data is that unauthorized users cannot do user behaviour analysis[5].

1. Literature Related Works

Information about a person's health is very private and sensitive. The healthcare system's participants must communicate this information often[5]. The data might be revealed or altered throughout the operational process, making medical information sharing between participants very challenging. The electronic health record system is crucial for improving medical services and lowering the cost and effort associated with patient care [6]. In [7], a strong Blockchain-based healthcare system is described. EHD may have several benefits, including information exchange, advice sharing, disease surveillance, analysis of patient history, etc. To some degree, privacy and security concerns prevent the sharing of EHD.

Researchers have recently begun to focus more on the security and privacy concerns surrounding EHD [8]. A paradigm is known as SEHRTB was presented in 2019 [9], and it allows for the exchange of (Electronic Health Records) EHRs between patients, specialists, service providers, and other medical institutions. The technology decreases latency, but no system feasibility estimate was provided. To manage patient data, they introduced a system based on Blockchain technology [10]. Writers have utilized Elliptic curve cryptography to preserve pseudonymity (ECC). However, the plan had a flaw in that compatibility was not a concern. A secure paradigm for healthcare management was put up in 2019 by [11]. The approach utilized Ethereum's Blockchain to maintain its distributed ledger and symmetric encryption to ensure secure data access.

A model of a sharing mechanism for health data was put out in 2020 [12]. The Blockchain-based approach utilized proxy re-encryption technology. By utilizing proxy re-encryption, the model's participating institutions are guaranteed to be able to decipher the intermediate Ciphertext. The plan assures secrecy. However, the issue is that no optimization of the implementation process is provided. A strategy for safe EHR sharing integrating Blockchain technology and the Interplanetary File System (IPFS) was suggested in 2019 [13]. The model uses an access control system based on Ethereum smart contracts. The model could share data quickly and reliably via a mobile app but still needs effective cloud-based EHR management[14].

2. The Architecture of a System

The parts of the system built into this work are the medical log generators, The surveillance of the medical log, the client for logging data (the patient), and the logging cloud (specialist). Through a wired or wireless network, a medical log generator connects a logging client. Medical log surveillance connects directly to a medical logging client through a wired or wireless network. According to the status, the client (Patient) connects to the CldStg via a secure or anonymous channel. Figure 1 shows how the whole system is put together. Logging client is made up of servers and databases that effectively store data. With this proposed system, clients can efficiently store their medical information on the medical server. The logging cloud is a huge place to store data, and it also has many other useful features, such as access from anywhere at any time, better security for medical data, faster deployment, and lower costs. It also keeps medical log data in good shape over a long period. Only persons

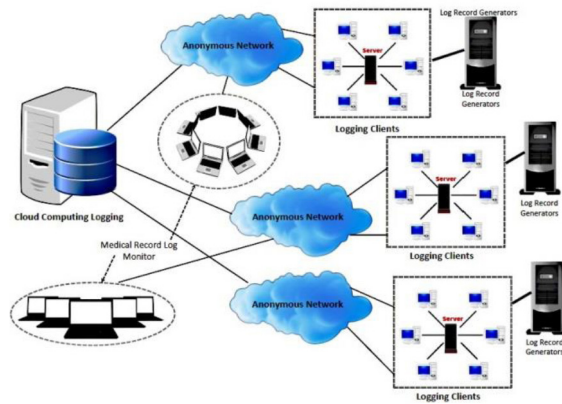


Figure 1
The System Architecture

who have paid for and signed up for CldStg services can upload medical records—having a lot of hospitals or medical centres as customers make it much cheaper for the service provider to run their business. Medical log Watch how the information in the medical log is being watched. This work uses medical log analysis to improve the security of medical data and transactions.

2.1 An Overview of SPCLRMA

The proposed SPCLRMA comprises four modules: preparing the medical log record, making anonymous upload tags, uploading anonymously, and deleting and rotating anonymously.

2.2 Preparing Medical Log Records

Before being transported to the cloud for storage, the medical log record is encrypted and message authentication codes are applied to each record to guarantee the integrity of the route. This work has three keys: two for keeping things honest and one for keeping things secret[16]. The Stamp of the Time, a special symbol of medical log initialization, and the pack size make up the first entry. The medical log close entry is the last one, showing that the pack is over. Based on [19], the following stages make up this algorithm:

1st stage: Create 3-master keys (Key-1 and Key-2) for integrity and Key-3 for confidentiality.

2nd stage: Create the 1st medical log entry $MLE = (\text{The Stamp of the Time, init_Mlog, } n)$.

3rd stage: Compute $Mac-0 = H_{Key-1} [E_{Key-3} [MLE]]$.

4th stage: Re-Calculate keys $A1 = H[Key-1]$, $X_1 = H[Key-2]$ and $K1 = H[Key-3]$.

5th stage: Eliminate Key-1, Key-2, Key-3.

6th stage: Produce a record $M1 = L1 \parallel H_{Key-1} [E_{Key-3} [MLE]]$. Calculate $Mac-1 = HA1[EK1[M1]]$.

7th stage: Compute collected $Mac-1 = H_{nX1}[Mac-0 \parallel Mac-1 \parallel n]$.

8th stage: Create a medical Log close entry $MLC = [EK_{n+1}[\text{The Stamp of the Time, } MLC \parallel Mac-n], HA_{n+1}[EK_{n+1}[\text{The Stamp of the Time, } MLC \parallel Mac-n]]$.

Pack -Uploading.

3. Modified Asymmetric Cryptography Algorithm (Modified RSA Algorithm)

A pseudocode representation of the Modified RSA Algorithm KeyGen algorithm is as follows [18]:

3.1 The Pseudocode of the KeyGen Algorithm

1st step: Randomly 4 prime numbers selecting p, q, r , and s . All primes must have the same length.

2nd step: Calculate n equal to $(p \text{ multiply } q)$, m equal to $(r \text{ multiply } s)$, ϕ equal to $(p \text{ minus } 1) \text{ multiply } (q \text{ minus } 1)$ and Z equal to $(r \text{ minus } 1) \text{ multiply } (s \text{ minus } 1)$.

3rd step: Select an integer e , $1 \text{ less than } e \text{ less than } \phi$, like $\gcd(e, \phi) \text{ equal to } 1$.

4th step: Calculate the secret exponent d , $1 \text{ less than } d \text{ less than } \phi$, like $(e \text{ multiply } d) \bmod \phi \text{ equal to } 1$.

5th step: Choose an integer g equal to $(m \text{ plus } 1)$.

6th step: Calculate the inverse of the modular multiplicative: μ equal to $Z^{-1} \bmod m$.

7th step: The *PublicKey*(encrypting) is (n, m, g, e) .

8th step: The *PrivateKey* (decrypting) is (d, Z, μ) .

Encrypting

1st step: Suppose m represents the encrypted message where Zero is less than $mesg$ less than n .

2nd step: Randomize r choosing, where r is less than m .

3rd step: Calculate Ciphertext as c equal to $(g \cdot mesg^e) \bmod (n \cdot multiply \cdot r^m) \bmod = m^2$.

Decrypting

1st step: Calculate messaged: m equal to $((c^Z \bmod m^2 \text{ minus } 1) / m) \cdot multiply \cdot \mu \bmod m)^d \bmod n$. Each tag upload is connected to a medical log record; Tag upload queries the cloud for medical log data.

4. Performance Analyzing

This work was completed on the abiCloud. In this publication, tables are utilized to describe the experimental findings. In five studies with varying client requests, Table 1 compares the number of requests ref utilized by SCLRMA and SPCLRMA. Genuine and fraudulent user requests were distributed 18:2 among the client (Patient) requests. Table 1 shows that the suggested SPCLRMA model outperforms the SCLRMA model in terms of limiting users and delivers accuracy in diagnosis and prevention of over 92%. This is owing to the model's suggested and implemented intelligent agents and successful key-sharing procedures. The level of security and the number of authorized users allowed by the SPCLRMA model is shown in Table 2.

Table 1

The SCLRMA model refused more requests for database access after confirming identification than the SPCLRMA model did

Trial's No	Order number tried	SCLRMA Order Rejections	SPCLRMA Order Rejections
1 st Experimenting	110	7	9
2 nd Experimenting	120	10	13
3 rd Experimenting	130	14	17
4 th Experimenting	140	17	21
5 th Experimenting	150	18	26

Table 2

Displays the number of requests for database access denied by SPCLRMA and SCLRMA after confirming identification.

Trial's No	Order number tried	SCLRMA Order Rejections	SPCLRMA Order Rejections
1 st Experimenting	110	9	10
2 nd Experimenting	120	11	15
3 rd Experimenting	130	14	22
4 th Experimenting	140	19	29
5 th Experimenting	150	23	36

Table 2 illustrates the number of requests for database access refused by the SCLRMA model after confirming identification and the SPCLRMA model. The memory requirements of the SCLRMA and SPCLRMA models are contrasted for various periods (t_1 , t_2). According to the implementation utilized in this model, a difference of 7% RAM is required in the recommended model because of the compression-created encryption.

5. Conclusion

The amount of medical log records has been constantly increasing in recent years. Locally storing and keeping medical log data involves a major financial and technical investment. Lower beginning costs, improved data security, continuous availability, decreased prices, more rapid deployment, and an extremely scalable and customized interface are all advantages of CldStg. In this project, a system that securely stores medical data utilizing appropriate medical log record archiving techniques was designed. The

flaws in present medical log management procedures were explored in this study. The researchers then presented a novel medical log management approach that addresses security and performance. Finally, the established secure logging mechanisms are shown to safely transport the created medical log records to the medical data repository. In the long run, it is intended to eliminate the need for external libraries and protocols by directly integrating the management of medical log records into the system. Additionally, I wish to do additional studies on secure key management techniques.

References

- [1] Liu, X., Xia, Y., Yang, W., & Yang, F. : Secure and efficient querying over personal health records in cloud computing. *Neurocomputing*, 274, 99–105 (2018). <https://doi.org/10.1016/j.neucom.2016.06.100>.
- [2] De Araújo, A. M. C., Times, V. C., & da Silva, M. U. : A Cloud Service for Graphical User Interfaces Generation and Electronic Health Record Storage. *Information Technology - New Generations*, 257–263 (2017). https://doi.org/10.1007/978-3-319-54978-1_36.
- [3] Alyami, M. A., Almotairi, M., Aikins, L., Yataco, A. R., & Song, Y.-T. : Managing personal health records using meta-data and cloud storage. 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS) (2017). <https://doi.org/10.1109/icis.2017.7960004>.
- [4] Belyaev, K., Sun, W., Ray, I., & Ray, I. : On the design and analysis of protocols for Personal Health Record storage on Personal Data Server devices. *Future Generation Computer Systems*, 80, 467–482 (2018). <https://doi.org/10.1016/j.future.2016.05.027>.
- [5] Gonçalves, G. D., Drago, I., Borges, A. V., Couto, A. P., & de Almeida, J. M. : Analyzing Costs and Benefits of Content Sharing in Cloud Storage. *Proceedings of the 2016 Workshop on Fostering Latin-American Research in Data Communication Networks* (2016). <https://doi.org/10.1145/2940116.2940128>.
- [6] Faisca, J. G., & Rogado, J. Q. : Personal cloud interoperability. 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM) (2016). <https://doi.org/10.1109/wowmom.2016.7523546>.

- [7] Li, J., Yan, H., & Zhang, Y. : Identity-Based Privacy Preserving Remote Data Integrity Checking for Cloud Storage. *IEEE Systems Journal*, 15(1), 577–585 (2021). <https://doi.org/10.1109/jsyst.2020.2978146>.
- [8] Wang, J. : Critical Factors for Personal Cloud Storage Adoption in China. *Journal of Data and Information Science*, 1(2), 60–74 (2016). <https://doi.org/10.20309/jdis.201614>.
- [9] Mishra, A. : Integrating with an Azure Storage Account. *Microsoft Azure for Java Developers*, 115–136 (2022). https://doi.org/10.1007/978-1-4842-8251-9_6.
- [10] Cheng, S., Lee, S.-J., & Choi, B. : An empirical investigation of users' voluntary switching intention for mobile personal cloud storage services based on the push-pull-mooring framework. *Computers in Human Behavior*, 92, 198–215 (2019). <https://doi.org/10.1016/j.chb.2018.10.035>.
- [11] Hou, B., Chen, F., Ou, Z., Wang, R., & Mesnier, M. : Understanding I/O Performance Behaviors of Cloud Storage from a Client's Perspective. *ACM Transactions on Storage*, 13(2), 1–36 (2017). <https://doi.org/10.1145/3078838>.
- [12] N. P. P. : Survey on Cloud Backup Services of Personal Storage. *International Journal of Research in Engineering and Technology*, 03(11), 380–382 (2014). <https://doi.org/10.15623/ijret.2014.0311063>.
- [13] Epilogue. From Personal Computing to Personal Computers. A People's History of Computing in the United States, 228–244 (2018). <https://doi.org/10.4159/9780674988538-009>.
- [14] Chaka, C. : Personal Mobile Cloud Computing Affordances for Higher Education. *Web-Based Services*, 1138–1161 (2016). <https://doi.org/10.4018/978-1-4666-9466-8.ch050>.
- [15] Ren, Z., Li, X., Xu, S., & Tong, Y. : Restricting the Number of Times That Data Can Be Accessed in Cloud Storage Using TrustZone. 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid) (2022). <https://doi.org/10.1109/ccgrid54584.2022.00038>.
- [16] Dai, D., Zheng, W., & Fan, T. : Evaluation of personal cloud storage products in China. *Industrial Management & Data Systems*, 117(1), 131–148 (2017). <https://doi.org/10.1108/imds-12-2015-0504>.
- [17] Heinsen, R. I., Lopez, C. P., Nguyen, T. D. T., & Huh, E.-N. : Cloud Storage Federation as a Service Reference Architecture. *Lecture Notes in*

- Electrical Engineering*, 668–675 (2017). https://doi.org/10.1007/978-981-10-4154-9_77.
- [18] Ho, R. : Cloud Computing and Enterprise Migration Strategies. *Cloud Technology*, 1732–1751 (2015). <https://doi.org/10.4018/978-1-4666-6539-2.ch080>.
- [19] Gracia-Tinedo, R., Sanchez Artigas, M., Moreno-Martinez, A., Cotes, C., & Garcia Lopez, P. : Actively Measuring Personal Cloud Storage. 2013 *IEEE Sixth International Conference on Cloud Computing* (2013). <https://doi.org/10.1109/cloud.2013.25>.
- [20] Hu, W., Yang, T., & Matthews, J. N. : The good, the bad and the ugly of consumer cloud storage. *ACM SIGOPS Operating Systems Review*, 44(3), 110–115 (2010). <https://doi.org/10.1145/1842733.1842751>.
- [21] Harvinder Singh, Anshu Bhasin & Parag Kaveri. SECURE : Efficient resource scheduling by swarm in cloud computing, *Journal of Discrete Mathematical Sciences and Cryptography*, 22:2, 127-137, DOI: 10.1080/09720529.2019.1576334.

Received April 2023