Mathematical Biosciences and Engineering

MBE, 19(9): 9200–9219.
DOI: 10.3934/mbe.2022427
Received: 21 March 2022
Revised: 28 May 2022
Accepted: 05 June 2022
Published: 22 June 2022

http://www.aimspress.com/journal/MBE

Research article

Novel Lagrange interpolation polynomials for dynamic access control in a healthcare cloud system

Te-Wei Chiang[1], Dai-Lun Chiang[2], Tzer-Shyong Chen[3], Frank Yeong-Sung Lin[1], Victor R. L. Shen[4,5,*] and Min-Chien Wang[3]

[1] Department of Information Management, National Taiwan University, Taipei City 106, Taiwan
[2] Financial Technology Applications Program, Ming Chuan University, Taoyuan City 330, Taiwan
[3] Department of Information Management, Tunghai University, Taichung City 407, Taiwan
[4] Department of Computer Science and Information Engineering, National Taipei University, Sanxia District, New Taipei City 237, Taiwan
[5] Department of Information Management, Chaoyang University of Technology, 168 Jifeng E. Rd., Wufeng District, Taichung City 413, Taiwan

* **Correspondence:** Email: rlshen@mail.ntpu.edu.tw, victor.rlshen@msa.hinet.net.

Abstract: The authority of user personal health records (PHRs) is usually determined by the owner of a cloud computing system. When a PHR file is accessed, a dynamic access control algorithm must be used to authenticate the users. The proposed dynamic access control algorithm is based on a novel Lagrange interpolation polynomial with timestamps, mainly functioning to authenticate the users with key information. Moreover, the inclusion of timestamps allows user access within an approved time slot to enhance the security of the healthcare cloud system. According to the security analysis results, this healthcare cloud system can effectively resist common attacks, including external attacks, internal attacks, collaborative attacks and equation-based attacks. Furthermore, the overall computational complexity of establishing and updating the polynomials is O(n*m* (log m)^2), which is a promising result, where m denotes the degree of $polynomial\ G(x, y)$ and n denotes the number of secure users in the hierarchy.

Keywords: dynamic access control; key management algorithm; Lagrange interpolation polynomial; personal health records; timestamp

1. Introduction

Patient-centered treatment has become a future trend to reduce unnecessary healthcare expenditure, and to enhance healthcare management quality and efficiency. The current applications contain electronic medical records, healthcare management, security control, telecare, personal health information cloud services, mobile health systems and wearable devices [1]. Healthcare staff members can conveniently access patient healthcare and physiological information through the combination of a healthcare cloud platform, terminal equipment and communication technology.

Big data analysis with artificial intelligence techniques can be applied to compare the personalized health records and data bank using cross-analysis. Also, the application of information technology analyzes the risks of underlying diseases and enables preventive treatment [2]. Especially, in emergency situations, the immediate acquisition of all patients' healthcare records and physiological information through the cloud system allows the healthcare staff members to make a fast and precise medical judgment.

This study stresses the security of healthcare information systems. The proposed algorithm allows us to update users' private keys for adding or removing any user. With the application of cloud platforms, the healthcare staff members with authority can access patients' past information by immediately logging into the personal health record (PHR) cloud system to ensure data availability among different medical institutions [2,3]. All related PHRs are stored in the personal health files. The users' security and privacy are therefore the most important issues when using the PHR cloud system in medical institutions.

File access in the healthcare environment should be controlled based on its personal identity. The authority limit for access control is therefore established in the system to ensure its data security. The hierarchically structured division can clearly define the security levels of files and user identities. Liu et al. proposed a hierarchy-based encryption mechanism and dynamic strategy for implementing the dynamic access control process introduced into the PHR cloud systems [3]. Therefore, key management becomes a primary issue in the access control process. In this case, a secure healthcare system with less computation should be built to achieve high efficiency and security.

***Problem statement*:** When personal health information is uploaded to the healthcare cloud system, the PHRs must ensure the security of health information [4]. This approach can reduce some of the computational load, and it is practically applied to the medical information system at National Taiwan University Hospital. Illegal users might attempt to steal the information in the access control process with the goal of information loss or misappropriation [5]. Therefore, a modified Lagrange interpolation polynomial-based access control algorithm with timestamps is proposed to create an effective and secure healthcare system so as to avoid the cloud system from being hacked.

Thus, the main purposes of this research are summarized as follows:

1) To propose a secure access mechanism, which was established in the cloud system, to maintain the users' security and information.
2) To integrate various types of healthcare information files so that users can access images, examination files and abstracts of health information.
3) To resist the most common attacks, namely, external attacks, internal attacks, collaborative attacks and equation-based attacks.
4) To enable dynamic adjustment of the personal authority via the access control matrix, which can effectively manage several users' authority.

2. Literature review

In this section, existing PHRs, key management mechanisms and related works are presented.

2.1. Personal health records

Since PHR are protected by the Personal Data Protection Act, they encourage medical staff members to think about innovation and answers to the unsolved clinical problems. PHRs not only bring more research works, but also help users make clinical decisions. In the research work by Flaumenhaft and Ben-Assuli [5], they reported that the privacy concerns related to individual rights were significant issues.

The valuable potential of PHRs is based on the adoption of those technologies by consumers and the active participation in the use of those technologies by multiple healthcare delivery constituents, such as hospitals, labs, pharmacies, insurance companies and government agencies. Consequently, the effective deployment and adoption of PHRs can result in a variety of benefits for these constituents [6].

Although the amount of interest in PHRs has been increasing gradually, their adoption remains low [7,8]. One of the oft-cited reasons is related to privacy and security due to an increasing trend of health information breaches [9,10]. PHRs are employed in many areas, such as in emergency departments [11], for high-risk women [12] and for diabetes management [13]. PHRs can be used to demonstrate values by providing a single view of patients' history, creating one source of the truth and bringing together potentially divergent documentation from different sources. This ensures that all healthcare professionals have the right information at the right time to reduce duplication, to enable a more preventative approach and to perform appropriate decision-making [14].

2.2. Key management mechanism

Since the quantity of decrypted data is large, the security management of private keys becomes an important issue. Key management aims to assist users in the process of dynamic access control. There are three key points required for key management, namely, security, stability and scalability.

2.3. Related works

Edemacu et al. [15] constructed a novel, high-efficiency and real-time-expression access control mechanism, which, for the decisional bilinear Diffie-Hellman assumption, was used for adjusting PHRs in the cloud system. Zhang et al. [16] proposed an access control mechanism with a password; it was applied to a PHR cloud system to securely share PHR information and access control transformation. Zahid et al. [17] proposed a framework for the practice of blockchain technology in the healthcare field of electronic health records (EHRs). The framework solved the extensibility problem encountered in blockchain technology with access recording systems. The blockchain-based framework provided an extensible, secure and complete solution for EHR systems. Madine et al. proposed a blockchain-based PHR structure, which was implemented by employing multi-party authorization and threshold encryption with a smart contract to automatically fulfill the access requirements of secure and reliable healthcare information in a PHR system [16,18].

In this study, only the authorized party achieving the threshold number can access the healthcare

information. Kibiwott et al. proposed the idea of healthcare traceable access control, aiming to target the granularity strategy for hiding and the traceable access control mechanism of mHealth, which can accurately identify malicious users and suspicious private keys in the identity form by searching linked identities [19]. The above attribute-based signcryption scheme provided confidentiality and unpredictability of PHR information [20]. Nevertheless, such schemes could not completely solve the problem of leaking users' privacy in a PHR cloud system. The security of hidden personal privacy in the access control of a PHR cloud system was still a critical challenge, e.g., regarding the prevention of various types of malicious attacks. The existing mechanisms cannot resist the chaining attacks. Thus, two authorized users could collaboratively generate a new and effective private key that was possessed by another legal user [21,22].

Dr. Knuth has proved this concept by focusing on the running time of the algorithm. Based on his research results, the time complexity is O(m* (log m)^2), where m denotes the degree of polynomial $G(x, y)$. Similarly, the overall computational complexity of establishing and updating the polynomials is O(n*m* (log m)^2), where n denotes the number of secure users in the hierarchy [23]. Consequently, the modified Lagrange interpolation polynomial with a time constraint was applied to prevent users' personal privacy and information from being leaked by allowing users to obtain different access authorities and supporting the appropriate dynamic access control.

3. Research methodology

The developing idea proposed for this study, and its practical application, are presented in this section. The practical application of our access control mechanism can be viewed as a communication platform between patients and medical institutions. Patients or people can upload the healthcare information to the cloud system through mobile phones, including medication records, medical images and exercise records. The integration of lifestyle information uploaded to the cloud system could be of benefit to the doctor-patient relationship by enabling coordination of the decision-making process. Patient medical records are usually connected and transmitted to the healthcare cloud system in the medical information management office, as shown in Figure 1. Users can access the cloud system as long as they pass the verification of the polynomial G(x, y). The permission in the process is given by judging users' legitimacy and the accessed files by using a validation mechanism. A hacker attempting to obtain illegal access might be able to crack the validation mechanism to acquire the file key or users' private keys. Thus, designing a method to resist hacker attacks was the main purpose of this study.

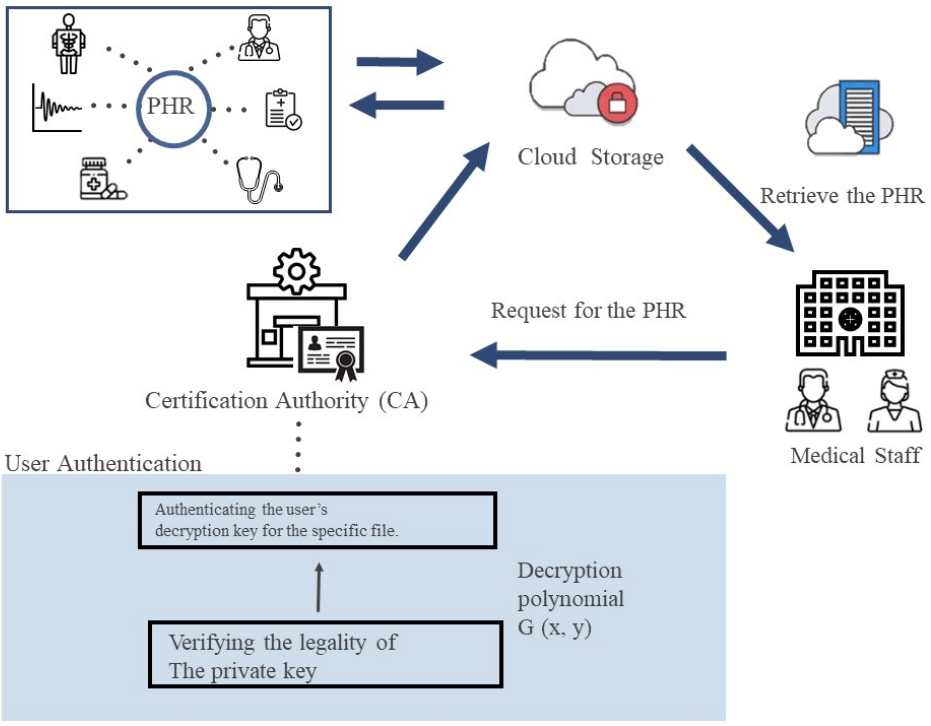

Figure 1. Structure of practical application in this study.

3.1. Personal health records in medical cloud system

Physiological information on the blood sugar, blood pressure and heart rate, as well as medical records and medical image information in the medical institutions and other relevant data records from the insurance companies and examination information in the research institutions or laboratories, can be collected through the wearable devices. Due to this, the medical staff members are allowed to collect important physical information and accelerate the understanding of patient health conditions to provide more appropriate diagnoses. All medical records, including past medical history and examination data, are included in the data bank of PHRs so that users can integrate various physiological information.

When logging in to this healthcare system, users can go through a two-stage validation procedure. The first stage of the polynomial $A_i(x)$ validates user legitimacy, and the second stage of the polynomial $F_{ij}(y)$ aims to validate whether users have the authority to decrypt the designated files. Thus, user authority is first validated; and, the cloud system will directly deny the log-in attempt of an illegal user. Its main purpose is to reduce unnecessary computational load and maintain the healthcare system effectively.

3.2. System initialization

Table 1 lists the definitions of various notations, where the private key H_i, file decryption key DK_j and polynomial $G(x, y)$, which is composed of various users' $A_i(x)$ and $F_{ij}(y)$, are the most important data items.

Table 1. Notation table.

Notation	Definition
S_i	The i-th user
H_i	Private key of the i-th user
DK_j	File decryption key
$file_j$	File, where j is the file number
$A_i(x)$	User authentication; x denotes the code for the user's private key input
$F_{ij}(y)$	Function to validate user file access
I_{H_i}	To judge whether the user's private key is in the list of legal private keys
$I_{J_i}(file_j)$	Indicator function to judge whether the user is authorized to access $file_j$
J_i	Files that the i-th user is authorized to access are put in the set J_i
p_{ij}	The corresponding random prime number for $file_j$
q_{ij}	The corresponding random prime number for H_i
TS	A set of access times
T	Time to access files
R	Constant
$G(x, y)$	Decryption polynomial

System initialization requires 10 steps of operation using the access control algorithm. Users need to have the private key, which is a polynomial $A_i(x)$, to log into the cloud system within a limited time. They are required to get the authority from the certificate authority (CA); otherwise, they cannot access the cloud system and are denied. The entire process is shown in Figure 2.

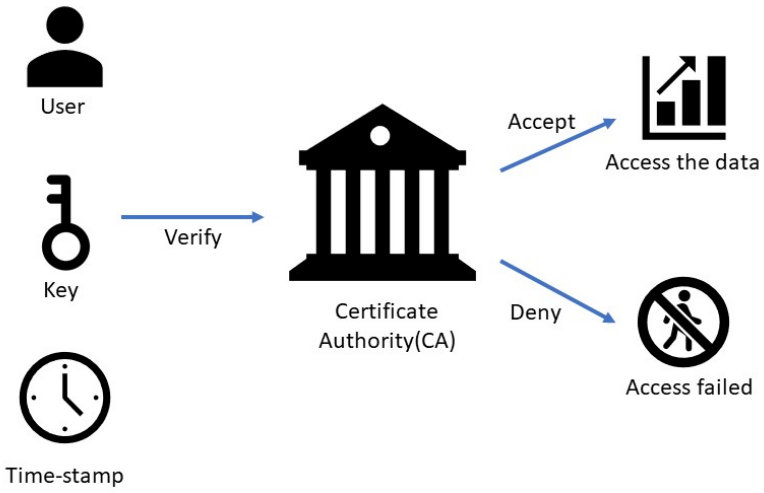

Figure 2. System initialization.

3.3. Access control algorithm

Input: *Personal Health Records (PHRs)*
Output: *Personal Health Records (PHRs)*
Procedure:

Step 1. Build a system user list.

$(S, \preccurlyeq)$ is a partially ordered set. The system uses partially ordered sets to build the access relationship. It is designed to give a set S, and the binary relationship "$\preccurlyeq$" with the characteristics of reflexivity, anti-symmetry and transitivity, and it is responsible for delivering binary data [20]. Different users are denoted as S_i in a set S; according to the identity of each user, the authority to access authorized files is set up; and, H_i denotes the user's private key. After giving sets of $S = \{S_1, S_2, \ldots, S_n\}$ and $H = \{H_1, H_2, \ldots, H_n\}$ to the cloud system, construct a user list with n users and their private keys.

Step 2. Build an incidence matrix between users and files.

The incidence matrix between users and files is built up. To encrypt all accessible files for users, the system is designed to construct a set of $file = \{file_1, file_2, \ldots, file_m\}$ with m files. There are decryption keys DK_j corresponding to the files $file_j$ (where $j = 1,2,\ldots,m$). The file encrypted by the key would not be accessed by illegal users without authority.

User authority is denoted as an access control matrix, where the number 1 denotes one with authority to access and the number 0 denotes one without authority. As shown in Figure 3, each confidential file, i.e., $file_1, file_2, file_3$ and $file_4$, has the respective decryption keys DK_1, DK_2, DK_3 and DK_4. To access a file, the decryption key for the file is needed.

$$
\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{array}
\begin{array}{c}
\begin{array}{cccc} file_1 & file_2 & file_3 & file_4 \end{array} \\
\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\end{array}
$$

Figure 3. Access control matrix.

Step 3. Build each user's file set J_i.

The CA is an impartial third-party unit in the system. Its function is to verify user identities and key issuances. It is also maintained as a secure information environment and mechanism with confidentiality, non-repudiation and availability. CA records the access authority of a user S_i in a set J_i. The access authority of User S_i is explained below.

$$J_i = \{J\text{: files that the } i\text{-th user is authorized to access are in the set}\}, \quad (1)$$

where $i = 1,2,\ldots,n$ and $n \in N$. For the set $(J, \preccurlyeq)$, $J_j \preccurlyeq J_i$ $(i, j \in N)$ reveals that the user S_i could acquire the decryption key with access authority for $file_j$; otherwise, it is not legally authorized. For instance, when $J_1 = \{2,3\}$ and $J_2 = \{1,2,3\}$, $\{2,3\} \preccurlyeq \{1,2,3\}$ such that $J_1 \preccurlyeq J_2$, we know that S_2 could acquire the decryption key of S_1 with the access authority to $file_2$ and $file_3$.

Step 4. The indicator function $I(x, y)$ is applied to judge the legal user, where the user's number is substituted into x. When it passes the validation, it is viewed as an internal user. However, an external user would be denied in this step. This step would reduce the computational load. Applying the example of User S_4 in Figure 3, User S_4 with a correct private key would receive a 1 for I_{H_4} to pass the validation. On the contrary, when the user inputs a wrong password of 123, I_{123} receives a 0.

$$I_x = \begin{cases} 1, & if x \in \{H_1, \ldots, H_n\} \\ 0, & o.w. \end{cases} \quad (2)$$

Step 5. Based on the access control matrix, the CA builds a polynomial $\mathrm{A}_i(x)$ to validate the private key for User S_i. In this step, each user is generated with a function for authentication.

$$\mathrm{A}_i(x) = \left\{\prod_{1\le k\le n, k\ne i}\left[\frac{x - H_k}{H_i - H_k} + (x - H_i)\right]\right\} \times I^{(x)}_{\{H_i\}}, \text{ for } i = 1, 2, \ldots, n \wedge x \in N \quad (3)$$

In Eq (3), $I^{(x)}_{\{H_i\}}$ is an indicator function. Before the system calculates $\mathrm{A}_i(x)$, the system will authenticate the user's identification to determine whether the logged in user is an internal one. If it is a legitimate internal user, the first half of the Lagrange interpolation polynomial will be calculated to verify whether the user has entered the correct private key. If it is not an internal user, the cloud system will automatically terminate the calculation in order to optimize system operations.

The file access function is provided in ***Steps 6–9***.

Step 6. The accessible files for each user are different. The CA sets q_{ij} and p_{ij} as the accessible files for each user.

Step 7. The indicator function $I_{J_i}(file_j)$ is applied to validate the user's authority to access the designated files. That is, $file_j \in J_i$, and the verification value for indicator function must be 1 to proceed to the successive step; otherwise, the verification value is set to 0 to stop the verification procedure.

$$I_{J_i}(file_j) = \begin{cases} 1, \text{ if } file_j \in J_i \\ 0, \quad \text{o.w.} \end{cases} \quad (4)$$

Step 8. A user's non-repeated private key H_i, $i = 1,2,\ldots,n$, where 24 hrs is the time unit, is applied; the t-th hour in a day is denoted as $TS(t)$, as shown in Eq (5).

$$y_{ij,t} = 24(q_{ij}\, mod\, p_{ij}) + TS(t) \quad (5)$$

$b(y_{ij})$ is the sole notation that allows User S_i to decrypt $file_j$, as shown in Eq (6).

$$b(y_{ij}) = 24(q_{ij}\, mod\, p_{ij}) \quad (6)$$

The parameter t indicates that the user can only log in or access the file within a limited time. Without the parameter t, it only calculates the decryption of the file. $b(y_{ij})$ is used to decrypt the file $file_j$ and verify the user's identity. So, it is not necessary to add timestamps to the calculation.

Step 9. When the user's designated file number is included in legal access coverage, the decryption key DK_j is acquired after substituting y_{ij} into F_{ij} to decrypt the access polynomial. Otherwise, the maximal value is acquired. $l_{ij}()$ denotes the Lagrange interpolation polynomial that is built according to each user's accessible files, as shown in Eq (7). When Eq (8) receives a 1, it is converted into a 0 through $a\left(l_{ij}\left(b(y_{ij})\right)\right)$. On the contrary, Eq (9) yielding a 0 is viewed as an illegal access attempt so that it can be denoted as the original calculation value.

$$F_{ij}(y_{ij}) = b(y_{ij}) + DK_j - \left\{b(y_{ij})l_{ij}\left(b(y_{ij})\right) + \left[\prod_{file_j \in J_i} a\left(l_{ij}\left(b(y_{ij})\right)\right) + \min_{t \in TS}||(y_{ij,t} mod 24) - t||\right]R\right\} \quad (7)$$

$$l_{ij}\left(b(y_{ij})\right) = \prod_{t=1, t\ne i}^{m}\left(\frac{b(y) - b(y_{tj})}{b(y_{ij}) - b(y_{tj})}\right)$$

$$\left(\frac{b(y)-b(y_{i-1,j})}{b(y_{ij})-b(y_{i-1,j})}\right) \times \left(\frac{b(y)-b(y_{i+1,j})}{b(y_{ij})-b(y_{i+1,j})}\right) \ldots \left(\frac{b(y)-b(y_{mj})}{b(y_{ij})-b(y_{mj})}\right) \quad (8)$$

$$a\left(l_{ij}\left(b(y_{ij})\right)\right)=\begin{cases} l_{ij}\left(b(y_{ij})\right)-1, & 0 \\ l_{ij}\left(b(y_{ij})\right)\neq 1, & \text{otherwise} \end{cases} \tag{9}$$

R is not a fixed value; rather, it is a random integer generated by the cloud system. According to Eq (7), if the user's identity authentication is approved, the user can only obtain the decryption key DK_j; if the identity authentication fails, the user will get a meaningless number.

Step 10. When integrating the authentication (Eq (3)) and file authority (Eq (7)) to obtain $G(x,y)$, the CA builds a decryption equation that is employed in the internal surroundings, as shown in Eq (10).

$$G(x,y)=\sum_{i=1}^{n}A_i(x)F_{ij}(y)\wedge x,y\in R \tag{10}$$

In Eq (10), $G(x,y)$ is a public polynomial, $A_i(x)$ is used to verify the user's access authority and $F_{ij}(y)$is used to verify whether the user has the authority to access the file. The security of $G(x,y)$ is based on the two polynomials $A_i(x)$ and $F_{ij}(y)$. It has a certain level of difficulty, and it is not easy to crack $G(x,y)$ through brute-force attacks.

If the legal users obtain the authority from the CA, they can access the PHRs within the limited time. However, if they are not the legal users, the cloud system will deny their access and they cannot log into the cloud system, which is shown in Figure 4.

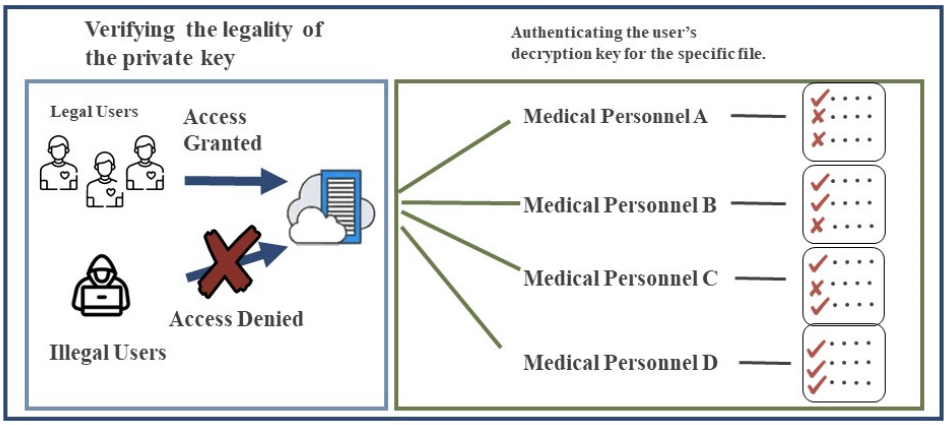

Figure 4. Decryption polynomial $G(x,y)$.

4. Dynamic access control scheme

In this section, security analysis and dynamic access control are discussed.

4.1. Security analysis

1) Users attempting to access files with incorrect decryption keys during a non-specified period

When accessing files within a non-accessible period, for example, User S_3, t = 9, subtracting t from TS, time control ≠ 0 and $l_{y_{S_3,file3}}\left(b(y_{S_3,file1})\right)\neq 1$, the cloud system multiplies a random constant R to obtain a maximal value such that DK_1or DK_3 cannot be obtained.

2) Users attempting to access files with an incorrect decryption key during the correct period

Assume that a pharmacist (S_3) attempts to access DK_3 with the term $y_{S_3,file_1}$ for the decryption key

DK. The pharmacist (S_3) selects two prime numbers q_{ij} = 73 and p_{ij} = 79 for $file_1$, as shown in Eq (11).

$$y_{S_3,file1,t} = 24(73\ mod\ 79) + TS(t) \tag{11}$$

The term $y_{S_3,file1}$ is substituted by $b(y_{ij})$, as shown in Eqs (12) and (13).

$$b(y_{S_3,file1}) = 24(73\ mod\ 79) \tag{12}$$

$$b(y_{ij}) = 24 \times 73 \tag{13}$$

Substituting $y_{S_3,file1}$ into $l_{y_{S_3,file3}}\left(b(y_{S_3,file3})\right)$, the result is not equal to 1, as shown in Eq (14).

$$l_{y_{S_3,file3}}\left(b(y_{S_3,file1})\right)$$

$$= \frac{(b(y)-11)(b(y)-37)(b(y)-107)(b(y)-137)}{(24\times 83-11)(24\times 83-37)(24\times 83-107)(24\times 83-137)}$$

$$= \frac{(24\times 73-11)(24\times 73-37)(24\times 73-107)(24\times 73-137)}{(24\times 83-11)(24\times 83-37)(24\times 83-107)(24\times 83-137)}$$

$$\neq 1 \tag{14}$$

Even though a user accesses files during the correct period, time control = 0 and $l_{y_{S_3,file3}}\left(b(y_{S_3,file1})\right) \neq 1$, so the system multiplies a random constant R to obtain a maximal value; then, DK_3 cannot be obtained.

4.2. Dynamic access control

When building a PHR file in a cloud system, dynamic access control is often encountered in the use cases, such as when adding or removing a user or modifying a file. Also, the owners of files and the system manager present authorities to update user information.

1) Adding a user

Any user intending to access some resources must be validated by the cloud system. When a new user S_{n+1} passes the system authentication, the personnel list in the system will be immediately updated with the data items and a private key H_{n+1} corresponding to the user, which is automatically generated according to S_{n+1}. Meanwhile, the cloud system writes the accessible file number $file_j$ of the new user into J_{n+1}. The term $file_j \in J_{n+1}$ reveals a new user S_{n+1} with authority to access $file_j$ and decrypts the key DK_j corresponding to $file_j$ by substituting the generated $y_{S_{n+1},j}$ and function $b'(y_{S_{n+1},j})$ into the decryption key polynomial $F_{ij}{}'(y_{S_{n+1},j})$ generated from the new user S_{n+1}, the decryption key DK_j can be decrypted, and the original public polynomial $G(x,y)$ is updated as $G'(x,y)$.

When adding a new user to the cloud system, the CA will update the user's access authority and old public polynomial by employing the following steps:

Step 1. To add a member S_{n+1}, the CA builds a private key H_{n+1}.

Step 2. The CA updates the secret polynomial $A_{n+1}(x)$ and indicator function $I_{H_{n+1}}^{(x)}$.

$$A_{n+1}(x) =$$

$$\left\{\prod_{\substack{1\le k\le n+1,\\ k\neq n+1}}\left[\frac{x-H_k}{H_{n+1}-H_k}+(x-H_{n+1})\right]\right\}\times I_{H_{n+1}} \tag{15}$$

Step 3. When $H_{(n+1)}$ is a legal private key, $A_{n+1}(H_{n+1}) = 1$; otherwise, it is 0.

Step 4. Aiming to target the accessible files for a new user, the file validation polynomial $F_{ij}(y_{S_{n+1},j})$ is updated. The new user decrypts $y_{S_{n+1},j,t}$ with the file decryption and timestamp, as shown in Eq (16).

$$y_{S_{n+1},j,t} = 24(q_{ij}\ mod\ p_{ij}) + TS(t) \tag{16}$$

$b'(y_{ij})$ is the sole term for the user S_{n+1} decrypting $file_j$, as shown in Eq (17); the polynomial to validate files is shown in Eq (17).

$$b'(y_{S_{n+1},j}) = 24(q_{ij}\ mod\ p_{ij}) \tag{17}$$

$$F_{ij}^{'(y_{S_{n+1},j})} = b'(y_{S_{n+1},j}) + DK_j - \left\{b'(y_{S_{n+1},j})l_{ij}\left(b'(y_{S_{n+1},j})\right) + \left[\prod_{file_j\in J_{i+1}} a\left(l_{ij}\left(b'(y_{S_{n+1},j})\right)\right) + \right.\right.$$

$$\left.\left.\min_{t\in TS}\left|\left|(y_{S_{n+1},j,t}\ mod\ 24) - t\right|\right|\right] R\right\} \tag{18}$$

Step 5. The original public polynomial $G(x,y)$ is updated as $G'(x,y)$, as shown in Eq (19).

$$G'(x,y) = G(x,y) + A_{n+1}(x)F_{ij}{}'(y_{S_{n+1},j}) \wedge x,y \in R \tag{19}$$

The process outlined in ***Steps 6–10*** depends on each user's authorities, and these steps are ignored for this particular task.

The change of i only adds or removes users, and the parameter I' is not used. The parameter j manages the access to the file, which is represented by $F_{ij}{}'(y_{S_{n+1},j})$, as shown in Eq (19). It represents the change of the user's access authority.

In Eq (19), if the attackers attempt to crack $G'(x,y)$, they must pass the verification of the indicator function $I_{H_{n+1}}$ before cracking $A_{n+1}(x)$. The attacker needs to face the difficulty of solving the polynomial before obtaining the user's private key c and the equation of $A_{n+1}(x)$. If the private key is incorrect, the attacker cannot deduce $A_{n+1}(x)$.

2) Removing a user

When the manager intends to remove a user from the healthcare access control system, they must simply remove the identity and access authority of User S_k. Besides, it clears the user's authority, with immediate validation, thus preventing them from being able to authorize any file $file_j$; the term $y_{S_i,j}$ is synchronously updated to completely remove the user's authority. It is assumed to have removed the member S_k. The CA removes the relevant notations $A_k(x)$ and $F_{ij}(y_{S_k,j})$ from the public polynomial to complete the removal update, as shown in Eq (20).

$$G"(x,y) = G(x,y) - A_k(x)F"_{ij}(y_{S_k,j}) \wedge x,y \in \mathcal{R} \tag{20}$$

3) Adding or removing the authority of a legal user's file in a specified period

To add the access authority of User S_i to $file_j$, the access authority of $file_j$ and the validation polynomial of F_{ij} are updated and the $term\ q_{ij} mod\ p_{ij}$ to decrypt $file_j$ is added to S_i. When removing the access authority of User S_i, the user's authority is thoroughly updated from the incidence matrix as ***Steps 7–10***. The process to add or remove the legal user's authority to access files in a specific time

period is shown below.

The CA updates the indicator function as shown in Eq (21).

$$I_{J_i}(file_j)' = \begin{cases} 1, \text{ if } file_j \in J_i \\ 0, \qquad \text{o.w.} \end{cases} \tag{21}$$

The CA adds the decryption notation $y(_{S\_i,j,t})$ with a timestamp, as shown in Eq (22).

$$y_{S_i,j,t} = 24(q_{ij} \bmod p_{ij}) + TS(t) \tag{22}$$

$\overline{b}(y_{S_i,j})$ is the sole term that allows User S_i to decrypt $file_j$, as shown in Eq (23); and, $\overline{b}(y_{S_i,j})$ is substituted into $F_{ij}(y_{S_i,j})$ as shown in Eq (24).

$$\overline{b}(y_{S_i,j}) = 24(q_{ij} \bmod p_{ij}) \tag{23}$$

$$F_{ij}(y_{S_i,j}) = \overline{b}(y_{S_i,j}) + DK_j - \left\{\overline{b}(y_{S_i,j}) l_{ij}\left(\overline{b}(y_{S_i,j})\right) + \left[\prod_{file_j \in J_i} a\left(l_{ij}\left(\overline{b}(y_{S_i,j})\right)\right) + \min_{t \in TS} \left|\left|(y_{S_i,j,t} \bmod 24) - t\right|\right|\right] R\right\} \tag{24}$$

The original polynomial $G(x,y)$ is updated as shown in Eqs (25) $\overline{G(x,y)}$ and (26) $\overline{\overline{G(x,y)}}$.

$$\overline{G(x,y)} = G(x,y) + A_i(x) F_{ij}(y_{S_i,j}) \wedge x, y \in R \tag{25}$$

$$\overline{\overline{G(x,y)}} = G(x,y) - A_i(x) F_{ij}(y_{S_i,j}) \wedge x, y \in R \tag{26}$$

4) Adding a file

When a patient's PHR $file_{j+1}$ is added to the cloud system, it provides the user with the term $q_{ij} \bmod p_{ij}$, which is composed of two unrepeated prime numbers p and q for $file_{j+1}$, to obtain the file key DK_{j+1}. For the user S_i with an access authority, the system updates the access authority; an example is provided in Table 2.

Table 2. Corresponding passwords to add new files.

User \ File No.	*file1*	*file2*	*file3*	*file4*	$file_{j+1}$
Doctor (S_1)	2	5	11	17	$q_1 \bmod p_{i,j+1}$
Head nurse (S_2)	29	NaN	37	41	$q_2 \bmod p_{i,j+1}$

Countinued on next page

User \ File No.	$file_1$	$file_2$	$file_3$	$file_4$	$file_{j+1}$
Pharmacist (S_3)	73	NaN	83	NaN	$q_3 \ mod \ p_{i,j+1}$
Patient (S_4)	101	103	NaN	NaN	$q_4 \ mod \ p_{i,j+1}$
Family member (S_5)	NaN	NaN	137	NaN	$q_5 \ mod \ p_{i,j+1}$

Note: NaN: Not a number.

When a confidential file is added to the cloud system, the CA gives each PHR user an access authority for the added file $file_{j+1}$ and resets the validation indicator I_{J_i} as $I^*_{J_i}$; and, the polynomial $F_{ij}(y_{S_i,j})$ is also updated as $F_{i,j+1}(y_{S_i,j+1})$. According to ***Steps 7–10***, the updating process is as shown in Eq (27).

Step 7. When adding the file $file_{j+1}$, the validation indicator I_{J_i} is updated and reset as $I^*_{J_i}$.

$$I^*_{J_i} = \begin{cases} 1, \text{if } file_{j+1} \in J^*_i \\ 0, \qquad \text{o.w.} \end{cases} \tag{27}$$

Step 8. When the target is a new file that users could access, the validation polynomial $F^*_{i,j+1}(y_{ij+1})$ is updated. The term $y_{S_i,j+1}$ with a timestamp for users decrypting a new file is shown in Eq (28).

$$y_{S_i,j+1} = 24(q_{ij} \ mod \ p_{ij}) + TS(k) \tag{28}$$

$b^*(y_{S_i,j+1})$ is the sole term that allows User S_i to decrypt $file_{j+1}$, as shown in Eq (29).

$$b^*(y_{S_i,j+1}) = 24(q_{ij} \ mod \ p_{i,j+1}) \tag{29}$$

Step 9. Substituting $b^*(y_{S_i,j+1})$ into $F^*_{i,j+1}(y_{i,j+1})$, the validation polynomial $F^*_{i,j+1}(y_{ij+1})$ is updated as shown in Eq (30).

$$F^*_{i,j+1}(y_{i,j+1}) = b^*(y_{S_i,j+1}) + DK_{j+1} - \left\{ b^*(y_{S_i,j+1}) l_{ij} \left(b^*(y_{S_i,j+1}) \right) + \left[\prod_{file_{j+1} \in J_i} a \left(l_{ij} \left(b^*(y_{S_i,j+1}) \right) \right) + \min_{t \in TS} \left| \left| (y_{S_i,j+1,t} \ mod \ 24) - t \right| \right| \right] R \right\} \tag{30}$$

Step 10. The original public polynomial $G(x,y)$ is updated as $G^*(x,y)$, as shown in Eq (31).

$$G^*(x,y) = \sum_{i=1}^{n} A_i(x) F^*_{i,j+1}(y_{ij+1}) \wedge x, y \in R \tag{31}$$

5) Removing a file

When a confidential file needs to be removed, simply remove the decryption key DK_W of the file $file_W$. The CA authorizes User S_i to modify the access authority of the removed file, and the validation indicator I_{J_i} is reset as $I^*_{J_i}$. The term $y_{S_i,w}$ with a timestamp for each user is synchronously updated; the polynomial $F_{ij}(y_{S_i,j})$ is updated as $F_{iw}(y_{S_i,w})$ to completely gain the authority of the confidential file.

The decryption polynomial update process is shown in Eq (32).

$$G^*(x,y) = G(x,y) - A_i(x)F_{iw}^*\left(y_{S_i,w}\right) \wedge x,y \in R \tag{32}$$

5. Avoidance of common attacks

In this section, how to avoid four common attacks on cloud systems, namely, external attacks, internal attacks, collaborative attacks and equation-based attacks, is interpreted; and, the security of each case is analyzed.

5.1. External attacks

External attacks are referred to as attacks through public decryption polynomials or any public information. To crack the validation indicator function and private key, an attacker starts from the application of the decryption polynomial to steal confidential files or private keys. The decryption polynomial is not publicized, as it merely allows internal users to prevent external users from acquiring user files and private keys.

5.2. Internal attacks

This type of attacker mainly consists of legal users with lower authority in the cloud system who are attempting to steal the private keys from those with higher authority in order to access files to which they have no access; they do this by attempting to crack through the decryption polynomial $G(x,y)$ and private key.

5.3. Collaborative attacks

Collaborative attacks refer to when two or more internal users are collaboratively attacking the cloud system.

From the access control matrix shown in Figure 3, the authority of Users S_3 and S_4 is $J_4 = \{file_1, file_2\}$ and $J_5 = \{file_3\}$, respectively; they intend to attack the files of User S_2, $\{file_1, file_3, \wedge file_4\}$. The decryption information related to the private keys DK_2 and DK_4 is hidden in $A_2(x)F_{S_4,file2}\left(y_{S_4,file2}\right)$ and $A_2(x)F_{DS_4,file4}\left(y_{S_4,file4}\right)$, respectively.

A patient (S_4) and their relative (S_5) respectively substitute the private keys H_4 and H_5 into a decryption polynomial $A_2(x)$; the computational process is described by Eqs (33) and (34).

$$A_2(H_4) =$$

$$\left\{\left[\frac{H_4-H_1}{H_2-H_1} + (H_4 - H_2)\right] \times \left[\frac{H_4-H_3}{H_2-H_3} + (H_4 - H_2)\right] \times \left[\frac{H_4-H_4}{H_2-H_4} + (H_4 - H_2)\right] \times \left[\frac{H_4-H_5}{H_2-H_5} + (H_4 - H_2)\right]\right\} \times I_{H_2} = 0 \tag{33}$$

$$A_2(H_5) =$$

$$\left\{\left[\frac{H_5-H_1}{H_2-H_1} + (H_5 - H_2)\right] \times \left[\frac{H_5-H_3}{H_2-H_3} + (H_5 - H_2)\right] \times \left[\frac{H_5-H_4}{H_2-H_4} + (H_5 - H_2)\right] \times \left[\frac{x-H_5}{H_2-H_5} + (H_5 - H_2)\right]\right\} \times I_{H_2} = 0 \tag{34}$$

The derivation results of Eqs (33) and (34) merely equal to 0; and, the attacker cannot force the

derivation nor re-derive the head nurse's (S_2) private key H_2 or any decryption key.

The patient (S_4) might attempt to attack the head nurse's (S_2) file $y_{S_2,file_4}$ to select two prime numbers q = 109 and p = 113, as shown in Eq (35).

$$y_{S_4,file_{4,t}} = 24(109\ mod\ 113) + TS(t) \tag{35}$$

The result of substituting $y_{S_4,file_4}$ into $b(y_{ij})$ is shown in Eq (36).

$$b(y_{S_4,file_4}) = 24(109\ mod\ 113) = 24 \times 109 \tag{36}$$

The result of substituting $b(y_{ij})$ into $l_{y_{S_2,file_4}}\left(b(y_{ij})\right)$ is shown in Eq (37).

$$l_{y_{S_2,file_4}}\left(b(y_{S_4,file_4})\right)$$

$$= \frac{(b(y)-17)(b(y)-89)(b(y)-109)(b(y)-139)}{(24\times 41-17)(24\times 41-89)(24\times 41-109)(24\times 41-139)}$$

$$= \frac{(24\times 109-17)(24\times 109-89)(24\times 109-109)(24\times 109-139)}{(24\times 41-17)(24\times 41-89)(24\times 41-109)(24\times 41-139)}$$

$$= \frac{2599\times 2527\times 2507\times 2477}{967\times 895\times 875\times 845}$$

$$= \frac{40{,}784{,}191{,}934{,}647}{639{,}903{,}184{,}375}$$

$$= 63.735 \tag{37}$$

The derivation result of Eq (37) simply equals a maximal value, which is converted to 0 during ***Step 9***, $asa\left(l_{ij}\left(b(y_{ij})\right)\right)$. The attacker cannot force the derivation nor re-derive the head nurse's (S_2) private key H_4 or any decryption key.

5.4. Equation-based attacks

Equation-based attacks is referred to attacks whereby an attacker attempts to obtain a private key by deriving a mathematical formula. The attack could easily happen when users change the access authority. As a result, the security of the authority change is the key point. The resistance to attacks during the process of adding or removing a user and updating authority is explained below.

1) Adding a user

$$G'(x,y) = G(x,y) + A_{n+1}(x)F_{ij}{}'(y_{S_{n+1},j}) \wedge x,y \in R \tag{38}$$

As shown in Eq (38), when the system needs to add a user, an attacker can deduce the updated decryption polynomial $G'(x,y)$ from the original decryption polynomial $G(x,y)$. Because $A_{n+1}(x)$ and $F_{ij}{}'(y_{S_{n+1},j})$ cannot be cracked, the attacker merely obtains the multiplied polynomial of $A_{n+1}(x)$ and $F_{ij}{}'(y_{S_{n+1},j})$; they cannot crack the polynomial $A_{n+1}(x)$ and decryption information of $F_{ij}{}'(y_{S_{n+1},j})$ or

obtain any file related to the user.

2) Removing a user

$$G''(x,y) = G(x,y) - A_k(x)F''_{ij}\left(y_{S_{k,j}}\right) \wedge x,y \in \mathcal{R} \tag{39}$$

As shown in Eq (39), like adding a user, any attacker can obtain the deduction of a public polynomial $G''(x,y)$ from the old public polynomial $G(x,y)$ when a user is removed from the cloud system. The attacker does cannot crack $A_k(x)F''_{ij}\left(y_{S_{k,j}}\right)$ so no file related to the user can be obtained.

3) Updating a user's file access authority in a specified period

$$\overline{G(x,y)} = G(x,y) + A_i(x)F_{ij}\left(y_{S_{i,j}}\right) \wedge x,y \in R \tag{40}$$

$$\overline{\overline{G(x,y)}} = G(x,y) - A_i(x)F_{ij}\left(y_{S_{i,j}}\right) \wedge x,y \in R \tag{41}$$

As shown in Eqs (40) and (41), in the process of adding or removing legal users' authority for file access in a specified period, any attacker can obtain the deduction of the public polynomial $\overline{G(x,y)}$ or $\overline{\overline{G(x,y)}}$ from the old public polynomial $G(x,y)$. The result is $A_i(x)F_{ij}\left(y_{S_{i,j}}\right)$ such that the attacker is unsuccessful.

In summary, the above four common attacks cannot crack this security mechanism. Meanwhile, attackers do not obtain any decrypted information. In this case, the security mechanism can effectively secure the healthcare cloud system against these attacks.

5.5. Functional comparison

To fully validate the claim that our proposed cloud system is more feasible and acceptable than other existing systems, we have made a functional comparison using different schemes. In Table 3, our proposed mechanism is compared with discretionary access control (DAC), attribute-based access control (ABAC) and role-based access control (RBAC).

DAC is an access method in which the owner is equipped with the authority to determine the permissions to other users. DAC is typically implemented using an access control list that allows the owner to easily set the user authorities. However, DAC is difficult to manage because of the scattered authority control. Besides, DAC cannot ensure the restricted use of information [24].

ABAC assigns attributes to an individual resource to determine the access authority [25]. These attributes may refer to a medical professional's team and individual roles in an institution and in their specific environment. To determine a user's authority to engage in each task, ABAC requires that information be accessed from multiple sources and evaluated with the specific access determination rules.

Sandhu et al. proposed the idea of an RBAC system in which roles are assigned to subjects and associated with permissions that define what kind of actions can be taken for different objects [26]. RBAC abridges the management of user access by assigning various roles to each user, and these roles define a set of permissions. RBAC also has its shortcomings. The cost and load are increased significantly as long as the number of users increases, because a role must be defined for each user. RBAC is a resource-intensive process for defining and structuring roles. It can only implement static and predefined policies. The high granularity of RBAC means that the system is vulnerable to internal cyberattacks [27].

Table 3. Results of functional comparison.

	DAC	ABAC	RBAC	Our mechanism
Flexibility	• Low flexibility: It lacks flexibility when an organization implements an authorization mechanism at multiple levels or involves multiple rights.	• High flexibility: It can define the detailed rules to protect data; therefore, access permissions can be changed by modifying the specific attributes.	• High flexibility: When changing the user's permission, only the role assigned to the user must be modified to change the user's access authority.	• High flexibility: Files can be added and modified in the system; moreover, personnel changes do not influence files. The effect of a file change on users is limited.
Strength	• An owner can conveniently set user's rights.	• During the revoking or adding of permissions, it is easy to modify attributes.	• RBAC provides fine-grained classification through which managers can automatically obtain all permissions related to their direct reports.	• Unnecessary verification is reduced. • Our system is secure enough to avoid internal and external cyberattacks.
Weakness	• Authority control is scattered and hard to manage. • Unauthorized rights may be granted.	• Numerous rules are required. • It tends to increase the overall cost.	• To establish granular policies, administrators must continually add more roles, which can lead to a considerable increase in the number of roles.	• When adding PHR files, the time complexity is proportional to the number of users.

6. Conclusions

The dynamic access control algorithm has been successfully built by employing the modified Lagrange interpolation polynomials with timestamps to enhance the security of healthcare cloud systems. To promote operational efficiency, user authentication and file authority verification have been applied to secure patient PHR files and private keys. In other words, it secures user files to effectively prevent illegal access by hackers. The access control matrix is used to manage users' keys and distinguish user identities and authorities. Moreover, the timestamp allows the dynamic access control process to become more feasible. The contributions of this novel access control algorithm are presented below.

1) Four common attacks, namely, external attacks, internal attacks, collaborative attacks and equation-based attacks, are disallowed for the purpose of proving the security of dynamic access control and the reliability of healthcare cloud systems.
2) A cloud system with higher security can optimize the service quality of healthcare information and enhance the capability of data sharing with each other among hospitals.
3) The significance is demonstrated by the ability to reduce the complexity of information management systems and effectively prevent hacker attacks.
4) It provides users with a complete and secure cloud-based healthcare information system.

In the future, PHR files will be used in various medical institutions. If a new cloud system is integrated with the proposed encryption polynomial and operated on healthcare websites, the healthcare staff members can immediately access individual medical data to accurately diagnose patients' diseases.

Acknowledgments

The authors are grateful to the anonymous reviewers for their constructive comments, which have improved the quality of this paper. Also, this work was supported by the Ministry of Science and Technology of Taiwan under grant number MOST 110-2221-E-029-011.

Conflict of interest

No conflict of interest is reported by the authors.

References

1. A. D. Salve, R. D. Pietro, P. Mori, L. Ricci, A logical key hierarchy-based approach to preserve content privacy in decentralized online social networks, *IEEE Trans. Dependable Secure Comput.*, **17** (2020), 2–21. https://doi.org/10.1109/TDSC.2017.2729553
2. M. A. Habib, M. Ahmad, S. Jabbar, S. Khalid, J. Chaudhry, K. Saleem, et al., Security and privacy-based access control model for internet of connected vehicles, *Future Gener. Comput. Syst.*, **97** (2019), 687–696. https://doi.org/10.1016/j.future.2019.02.029
3. X. H. Liu, Q. Liu, T. Peng, J. Wu, Dynamic access policy in cloud-based personal health record (PHR) systems, *Inf. Sci.*, **379** (2017), 62–81. https://doi.org/10.1016/j.ins.2016.06.035
4. Y. Xu, W. Gao, Q. Zeng, G. Wang, J. Ren, Y. Zhang, A feasible fuzzy-extended attribute-based access control technique, *Cyberspace Secur. Future Internet*, **2018** (2018), 1–11. https://doi.org/10.1155/2018/6476315
5. Y. Flaumenhaft, O. Ben-Assuli, Personal health records, global policy and regulation review, *Health Policy*, **122** (2018), 815–826. https://doi.org/10.1016/j.healthpol.2018.05.002
6. U. Ruhi, R. Chugh, Utility, value, and benefits of contemporary personal health records: Integrative review and conceptual synthesis, *J. Med. Internet Res.*, **23** (2021), e26877. https://doi.org/10.2196/26877
7. Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, Y. Zhang, An efficient privacy-enhanced attribute-based access control mechanism, *Concurrency Comput.: Pract. Exper.*, **32** (2020), 1–12. https://doi.org/10.1002/cpe.5556

8. A. Alanazi, Y. A. Anazi, The challenges in personal health record adoption, *J. Healthcare Manage.*, **64** (2019), 104–109. https://doi.org/10.1097/JHM-D-17-00191
9. M. M. Hossain, Y. A. Hong, Trends and characteristics of protected health information breaches in the United States, *Proc. AMIA Annu. Symp.*, **4** (2020), 1081–1090.
10. A. A. Abd-Alrazaq, B. M. Bewick, T. Farragher, P. Gardner, Factors that affect the use of electronic personal health records among patients: A systematic review, *Int. J. Med. Inf.*, **12** (2019), 164–175. https://doi.org/10.1016/j.ijmedinf.2019.03.014
11. S. Kim, T. Kim, W. Cha, J. Lee, I. Kwon, Y. Choi, et al., User experience of mobile personal health records for the emergency department: Mixed methods study, *JMIR mHealth uHealth*, **8** (2020), e24326. https://doi.org/10.2196/24326
12. H. Kim, A. Mahmood, E. Carlton, J. Goldsmith, C. Chang, S. Bhuyan, Access to personal health records and screening for breast and cervical cancer among women with a family history of cancer, *J. Cancer Educ.*, **35** (2020), 1128–1134. https://doi.org/10.1007/s13187-019-01568-5
13. D. Seo, Y. Park, Y. Lee, J. Kim, J. Park, J. Lee, The use of mobile personal health records for hemoglobin A1c regulation in patients with diabetes: Retrospective observational study, *J. Med. Internet Res.*, **22** (2020), e15372. https://doi.org/10.2196/15372
14. C. P. Subbe, N. Pearson, S. Wischhusen, R. Hibbs, S. Wright, M. Xenou, Scenario-based design for a hospital setting: An exploratory study of opportunities and barriers for personal health records usage, *Future Healthcare J.*, **7** (2020), 125–130. https://doi.org/10.7861/fhj.2019-0061
15. K. Edemacu, B. Jang, J. W. Kim, Efficient and expressive access control with revocation for privacy of PHR based on OBDD access structure, *IEEE Access*, **8** (2020). https://doi.org/10.1109/ACCESS.2020.2968078
16. C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, Y. Zhang, A blocktrain-based multi-cloud storage data auditing scheme to locate faults, *IEEE Trans. Cloud Comput.*, (2021), 1–12. https://doi.org/10.1109/TCC.2021.3057771
17. N. Zahid, A. H. Sodhro, U. R. Kamboh, A. Alkhayyat, L. Wang, AI-driven adaptive reliable and sustainable approach for internet of things enabled healthcare system, *Math. Biosci. Eng.*, **19** (2022), 3953–3971. https://doi.org/10.3934/mbe.2022182
18. M. M. Madine, K. Salah, R Jayaraman, I. Yaqoob, Y. Al-Hammadi, S. Ellahham, et al., Fully decentralized multi-party consent management for secure sharing of patient health records, *IEEE Access*, **8** (2020). https://doi.org/10.1109/ACCESS.2020.3045048
19. K. P. Kibiwott, Y. Zhao, J. Kogo, F. Zhang, Verifiable fully outsourced attribute-based signcryption system for IoT eHealth big data in cloud computing, *Math. Biosci. Eng.*, **16** (2019), 3561–3594. https://doi.org/10.3934/mbe.2019178
20. A. Shabbir, M. Shabbir, A. R. Javed, M. Rizwan, C. Iwendi, C. Chakraborty, Exploratory data analysis, classification, comparative analysis, case severity detection, and internet of things in COVID-19 telemonitoring for smart hospitals, *J. Exp. Theor. Artif. Intell.*, (2022), 1–28. https://doi.org/10.1080/0952813X.2021.1960634
21. G. Tripathi, K. Singh, D. K. Vishwakarma, Applied convolutional neural network framework for tagging healthcare systems in crowd protest environment, *Math. Biosci. Eng.*, **18** (2021), 8727–8757. https://doi.org/10.3934/mbe.2021431
22. L. K. Ramasamy, F. Khan, M. Shah, B. V. V. S. Prasad, C. Iwendi, C. Biamba, Secure smart wearable computing through artificial intelligence-enabled internet of thinggs and cyber-physical systems for hearlth monitoring, *Sensors*, **22** (2022), 1076. https://doi.org/10.3390/s22031076

23. D. E. Knuth, Seminumerical algorithms, in *The Art of Computer Programming*, **2** (1998), Addison-Wesley.
24. I. Indu, P. R. Anand, Hybrid authentication and authorization model for web-based applications, in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, *IEEE*, (2016), 1187–1191. https://doi.org/10.1109/WiSPNET.2016.7566324
25. D. R. Kuhn, E. J. Coyne, T. R. Weil, Adding attributes to role-based access control, *Computer*, **4** (2010), 79–81. https://doi.org/10.1109/MC.2010.155
26. R. Sandhu, D. Ferraiolo, R. Kuhn, The NIST model for role-based access control: towards a unified standard, in *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, (2000), 47–63. https://doi.org/10.1145/344287.344301
27. E. Chickowski, Healthcare unable to keep up with insider threats, *Dark Reading*, Available from: https://www.darkreading.com/vulnerabilities—threats/healthcare-unable-to-keep-up-with-insider-threats/d/d-id/1137610?. Accessed: May 12, 2018.

AIMS Press

©2022 author name, licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)