Sensors and Materials, Vol. 37, No. 11 (2025) 5081–5097
MYU Tokyo

S & M 4234

A Time-based Secure Access Control Framework for Cloud Medical Sensor Information Systems

Min-Yuan Ho,[1] Bi-Huei Tsai,[1] Tzer-Shyong Chen,[2*]
Yu-Fang Chung,[3] and Dai-Lun Chiang[4]

[1]Department of Management Science, National Yang Ming Chiao Tung University
No. 1001, University Road, Hsinchu 30010, Taiwan
[2]Department of Information Management, Tunghai University
No. 1727, Sec. 4, Taiwan Blvd, Xitun District, Taichung City 407224, Taiwan
[3]Department of Electrical Engineering, Tunghai University
No. 1727, Sec. 4, Taiwan Blvd, Xitun District, Taichung City 407224, Taiwan
[4]Department of Computer Science, Tunghai University
No. 1727, Sec. 4, Taiwan Blvd, Xitun District, Taichung City 407224, Taiwan

(Received August 21, 2025; accepted October 30, 2025)

Keywords: personal health record, cloud healthcare, access control mechanism, time-based authentication, sensor-generated data

With the advancement of information technology and a shift toward healthcare focused on chronic diseases, health awareness has increased. This trend has driven the development of electronic health records (EHRs). However, traditional EHRs remain confined to individual hospital databases, restricting data sharing across institutions. Therefore, we adopt the concept of personal health records (PHRs), where personal medical records are securely stored in the cloud healthcare under the individual's ownership, allowing both individuals and authorized medical institutes to access and manage the data in real time. We propose an access control mechanism for cloud-based medical information systems. Our mechanism incorporates time-based authentication to increase access security. Leveraging sensor-generated data, and with the patient's permission, hospitals can access medical records when the patient goes to different medical institutions. This process ensures practical PHR data transfer within the healthcare network. Conversely, patients can also access their own PHRs with hospital authorization. This approach alleviates both time and costs associated with information transfer. Sensor-generated data further support real-time monitoring. This capability allows medical institutions to promptly access cases on the cloud medical information system for diagnosis and treatment in emergencies.

1. Introduction

The challenges in accessing medical information arise from the growing awareness of self-health management in society, which amplifies people's willingness to seek medical assistance. Patients and their family members autonomously maintain and manage their personal health

*Corresponding author: e-mail: arden@thu.edu.tw
https://doi.org/10.18494/SAM5908

ISSN 0914-4935

records (PHRs), including personal health maintenance behavior, medical records, and services derived from self-health management.(1) Sensor devices, such as smartwatches and home medical monitors, are playing an increasingly important role in capturing personal health behaviors, generating continuous data that supplement clinical records. Nonetheless, the majority of health records are stored separately within medical institutions. Such scattered data cannot be effectively integrated and delivered, and a lack of data integrity would lead to issues with data management. Accordingly, the Ministry of Health and Welfare, Taiwan, collaborated with Microsoft in 2012 to advance the integration of PHRs and cloud systems. The initiative integrated users' health records, including those from healthcare and insurance agencies, aiming to strengthen the public in autonomously managing personal health information while ensuring data security during access and delivery processes. In addition, the integration of sensor technologies—such as wearable health monitors—can enhance the continuity and accuracy of health data collected from daily life, supporting the development of more comprehensive and real-time PHRs.(2)

Wireless networks have affected various aspects of applications. Research on access control is maturing, gradually fueling the growth of cloud algorithms. In addition to the PHR system, many electronic systems strive to transition their platforms to the cloud. Cloud computing offers the advantages of timely provision of self-service, sharing resources with others, and rapid redeployment, catering to individual needs. While networks provide ease of use, risks associated with the changing environment cannot be neglected. High-efficiency secure access control is paramount. Therefore, matrix control is used as the control mechanism in this study to authenticate a user's file access rights.

Placing PHRs on the cloud aims to improve the efficiency of sharing medical information and reduce the wastage of medical resources. By using cloud-based PHRs, patients can independently manage their health care records, significantly lowering costs for medical institutions by utilizing PHRs on the cloud. Moreover, sensor-enabled devices can also feed real-time physiological data into the system, improving the accuracy and timeliness of cloud-based medical records. To efficiently handle many users, it is essential to integrate an access control mechanism that prevents excessive algorithmic burdens on physical systems. To support multiple users, we propose an efficient and secure access control mechanism that integrates PHRs with the cloud to provide additional benefits.(3) Specifically, we have designed a patient-centered PHR time-interval access control mechanism tailored for multi-user cloud applications. This design aims to simplify the complexity in key management and enhance security.

To implement a PHR system in a cloud environment, it is essential to ensure information security, with access control and personal privacy protection playing critical roles. For multi-user file access, Lagrange polynomials and public-key cryptography can be used for effective management, ensuring both security and confidentiality. The objective of this study is to develop a patient-centered medical record management framework with encryption and time control to block unauthorized access by insiders and outsiders. This is achieved through robust access control and key management mechanisms. By integrating real-time sensor data and blockchain encryption, the system ensures secure and anonymous data transmission, enhancing trust, efficiency, and data protection in healthcare environments.(4) As a result, patients benefit from

increased control over who accesses their data and enjoy greater peace of mind. The framework also enables patients to manage and selectively share their PHRs autonomously, contributing to the efficiency and reliability of PHR encryption. This gives patients the ability to choose when and with whom to share information, which increases their involvement and confidence in managing their health.

2. Literature Review

The topic of access control has sparked intense debates. In past research, scholars have proposed several access control mechanisms, including access control matrices, access control lists, function lists, and role-based access control (RBAC). An access control matrix manages system resource access through straightforward mechanisms.[5] Various access control models have been presented, encompassing task-based access control (TBAC), which conducts access inquiries and verifications based on task requirements.[6] Additionally, temporary RBAC (TRBAC) validates role authority by considering changes in time intervals.[7] At the same time, spatial RBAC (SRBAC) authenticates role authority based on changes in the access control system's security strategy due to spatial location. With the increasing integration of sensor data, access control can further adapt to dynamic environmental and user-specific conditions, enhancing precision and security. For patients, these adaptive and robust access control mechanisms can help ensure that their sensitive data are accessed only by authorized individuals and only when necessary, potentially improving data privacy, personalized care, and overall safety. The aforementioned access control models can be operated independently or integrated with other control models. TBAC and RBAC can be used simultaneously in access systems or combined with RBAC to serve as the access control mechanism in complex organizations. Moreover, access control mechanisms can be incorporated into other system structures, such as the combination of RBAC with the financial industry to develop online payment systems.

The access right of the discretionary access control (DAC) mechanism was based on the user identity and relevant access inquiry. Users can autonomously manage their access rights without requiring system manager intervention. While the DAC model provided flexible access control mechanisms, it could not ensure the integrity of authorized data. In the mandatory access control (MAC) mechanism, the system evaluates user and object labels when a user requests access to an object. Authorization is granted if the user's authority matches or exceeds the confidentiality level of the object; otherwise, access is denied.[8]

Sandhu *et al.* promoted RBAC in 1996,[9] which was subsequently adopted by the National Institute of Standards and Technology (NIST) and standardized in 2011, later renamed NIST RBAC.[10] In this framework, RBAC introduced the "role" element, positioned between "user" and "access right" in the system, allowing users to access files via the intermediary "role" element. With the advancement of sensor technologies, RBAC systems can more effectively utilize real-time contextual data to refine role assignments and access decisions.

Along with the rising awareness of data confidentiality, Chen and Huang proposed an access control mechanism that combines encryption techniques and key management in 2005.[11] Building on this foundation, they subsequently applied this mechanism to the mobile agent

environment. In practice, before a mobile agent was permitted to operate on the Internet, the transfer host determined which hosts and data contents the agent accessed. In this case, an agent had to confirm the access path and use a single key to encrypt the secret document, utilizing symmetric encryption systems, such as the Advanced Encryption Standard (AES), Data Encryption Standard (DES), or International Data Encryption Algorithm.[12]

As the cloud environment matured, there was a promotion of elliptic curve cryptography, bilinear pairing, authentication, and access mechanisms, incorporating features such as transfer and time limits. Building on these developments, Liu *et al.* proposed a dynamic access framework in 2012 to implement accurate access control for cloud data and diaries in a multi-user device.[13] Specifically, this framework was introduced in the medical environment to enhance patients' control over their health records. As a result, the system authorized the access rights of doctors, pharmacists, nurses, and researchers while protecting patients' information privacy.

Relevant technologies, such as context-based authentication and access control (CAAC), have augmented the foundational RBAC authority model. This method verifies users' access rights to confidential information based on dynamically changing contexts, incorporating pertinent resources, environmental factors, user attributes, and software services.[14] Recent studies have proposed advanced access control and secure mechanisms in healthcare systems. Liu *et al.* presented several access control solutions for EHRs to improve data sharing and privacy protection.[15] Davis *et al.* indicated the function of blockchain in securing the internet of medical things (IoMT), ensuring trustworthy and tamper-resistant medical data exchange.[16] Evans *et al.* further emphasized implementing zero-trust security models in cloud environments to strengthen authentication and minimize threats and risks.[17] Lee *et al.* proposed a cross-domain access control model that inspired later context-aware approaches.[18] Colombo and Ferrari proposed a route map that reinforced the data protection function in Not only Structured Query Language (NoSQL) data repositories and established a CAAC mechanism for MongoDB.[19] Kayes *et al.* analyzed various context conditions and inferred the tacit knowledge process to discern the correlation between context messages and integrate existing context information,[20] thereby developing the CAAC system. Sensors play a vital role in providing dynamic, real-time environmental and user data that strengthen context-aware access decisions in these systems.

The AES is a symmetric block cipher algorithm that encrypts data in 128-bit blocks, dividing each block into 16 bytes.[21,22] Its predecessor, the DES, is also a block cipher that encrypts 64-bit blocks using a 56-bit key. However, DES became vulnerable to brute-force attacks because of its shorter key. To overcome this limitation, AES was introduced with longer keys and a more complex encryption structure. As a result, AES offers a significantly higher security than DES and is now one of the most widely adopted symmetric encryption standards.[23]

PHRs are electronic health record (EHR) systems. They empower individuals to store, manage, and share personal health information, including medical records, inspection reports, and prescription drugs, with medical personnel when necessary.[24] PHR systems can improve health care efficiency and quality, enabling individuals to actively participate in personal medical decisions and enhance self-health management.[25] Sensor technologies embedded

within PHR systems deliver continuous, accurate health monitoring data, enabling individuals to manage their personal health more effectively. PHR systems present advantages of convenience, traceability, right to know, and involvement in medical decisions.[26] Users govern their EHRs and randomly check and manage their personal health state. Additionally, they can share information with medical professionals to help them take better care of their health.

Ensuring the security of data accessed in a cloud environment is essential, and it is imperative to confirm that the data accessed by legally authorized personnel remain untampered and safeguarded against theft. In this case, the identification of asset value should match confidentiality, integrity, and availability.[27] In a cloud environment, data encryption strengthens security during transmission and storage, guaranteeing users' privacy and confidentiality.[19]

A Lagrange interpolation polynomial can accurately pass through multiple specified points on a 2D plane. On an x–y plane, a function that passes through a finite set of multipoint coordinates, yielding $n + 1$ points, results in a unique polynomial of order n or less. A polynomial is constructed such that it equals 1 at x_j but 0 at other data points, such as (x_1, y_1), (x_2, y_2), …, (x_n, y_n). Consequently, the interpolation basis function, denoted as $l_j(x)$, is utilized and expressed as

$$L(x) = \sum_{j=0}^{n} y_j l_j(x). \tag{1}$$

$l_j(x)$ is expressed as

$$l_j(x) = \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i} = \left(\frac{x - x_0}{x_j - x_0}\right) \cdots \left(\frac{x - x_{j-1}}{x_j - x_{j-1}}\right)\left(\frac{x - x_{j+1}}{x_j - x_{j+1}}\right) \cdots \left(\frac{x - x_n}{x_j - x_n}\right), \quad 1 \le j \le n. \tag{2}$$

$l_j(x)$ shows the value 1 on x_j, but 0 on other points x_i $(i \neq j)$, such that $l_j(x)$ reveals

$$l_j(x) = \begin{cases} 0, \ i \neq j \\ 1. i = j \end{cases}. \tag{3}$$

Mathematical functions are frequently used for design and analysis to represent standard operations, providing an effective means to verify performance. When multiple conditions exist, the relationship between x and y in the coordinate plane cannot be conclusively established. Lagrange interpolation is consequently applied to the design, through interpolation in the x–y coordinate, to acquire a Lagrange interpolation polynomial based on finite multipoint coordinates.

3. Research Methodology

3.1 Research structure

The Lagrange interpolation polynomial applied in this study allows users to transmit, edit, or

designate specific users for reading, while confirming the primary authority for each user's document access.[24] The Lagrange interpolation polynomial derived mathematically from multiple points can optimize solutions under various conditional constraints. Moreover, timestamp access control integrates a scalar function and a hash function to reinforce the security of document files. In the access mechanism, each encryption key exhibits uniqueness and an asymmetric relationship with each other. Attempting to crack files without authorization would be a challenging task within a reasonable computation time, as the process would need to exceed the timestamp-imposed limit.[28]

In our system, time-based authentication is integrated with a fog computing architecture to enhance access efficiency. Instead of routing all authentication and data access into the central cloud, fog nodes located closer to the data sources, including hospitals or medical devices, perform preliminary authentication and temporary data catching. This distributed design significantly reduces communication latency and alleviates computational load on the central server. By verifying user permissions and time-based tokens at the fog layer, the system shortens the response time for access requests. This enables a faster retrieval of medical records, especially in emergencies. Thus, while time-based authentication strengthens security, its combination with fog computing effectively improves access efficiency through reduced latency, localized processing, and optimized data transmission pathways. Moreover, a user can authorize the addition, modification, and removal of files, and has the autonomy to assign rights to specific users for their own digital assets. Leveraging sensor-based authentication data further fortifies the security framework, delivering the immediate and accurate verification of user identity and context-aware access management. The cloud system framework is shown in Fig. 1.

3.2 Construction of system initialization

Pre-system settings and building are introduced in this section, including establishing user identity and authorizing file access. According to the set access matrix, a user's access

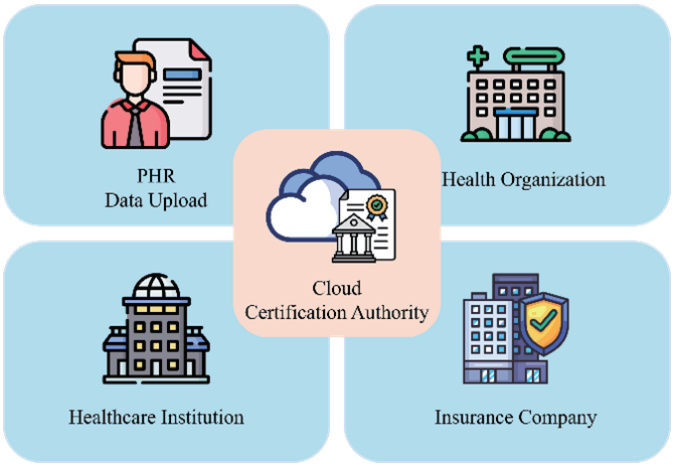


Fig. 1. (Color online) Application of cloud secure access control.

polynomial is calculated to construct the decryption polynomial. Besides, the system would establish an access relationship with a partially ordered set. (S, $\preccurlyeq$) symbolizes a partially ordered set; $\preccurlyeq$ represents reflexivity or anti-symmetry, delivering binary data in set S; partial sorting is defined by the binary relation "$\preccurlyeq$" on a set S, presenting reflexive, anti-symmetric, and transitive characteristics. The parameters used for equations in this paper are shown in Table 1.

Step 1: To accurately export DK_u to access the required file, both a polynomial and a system-saving function are necessary to manage access control. Initially, the authenticity of the user key has to be defined. Subsequently, the indicator function $I_{\{H_1,\ldots,H_n\}}(x)$ is used to verify whether the users have authority to access or not. It returns 1 if x belongs to the authorized key set $\{H_1, ..., H_n\}$ and 0 otherwise.

Step 2: User permission must be confirmed to authorize file access rights. It is suggested that the function $I_{J_i}(x)$ be defined for each file set $J_i = \{u: 1 \le u \le m\}$, where each file ID represents an authorized item. In other words, $I_{J_i}(x)$ equals 1 if x is included in the authorized set J_i; otherwise, it returns 0. This auxiliary function indicates whether user S_i is authorized to access a specific file.

Step 3: The Lagrange interpolation polynomial is applied to generate the corresponding function polynomial.

(1) To create the unique primary key H_i with certificate authority (CA), where $i = 1, 2, \ldots, n$, corresponding to $\{S_1, S_2, ..., S_n\}$ in S_i, the primary key is confidential to users.

(2) CA manages all users' H_i and constructs the primary key of the indicator function for authentication. CA manages all user keys $\{H_1, H_2, ..., H_n\}$ and constructs the primary key of the indicator function $I_{\{H_1,\ldots,H_n\}}(x)$ for verification. This indicator equals 1 when x matches any authorized H_i in the set and 0 otherwise.

$I_{\{H_1,\ldots,H_n\}}(x)$ refers to $H = \{H_1, H_2, ..., H_n\}$; the indicator function of $I_{\{H_1,\ldots,H_n\}}(x)$ is used for verification.

(3) CA establishes the function $A_i(x)$ for User i to apply the Lagrange interpolation polynomial, where

$$A_i(x) = \left\{ \prod_{k=1,k\neq i}^{n} \frac{(x - H_k)}{(H_i - H_k)} \right\} \times I_{\{H_1,\ldots,H_n\}}(x), \text{ for } i = 1,2,\ldots,n, x \in R. \quad (4)$$

Table 1
Parameters and notations used in the proposed access control algorithm.

Code	Definition	Function
S_i	Security level	Distinguish the user's security level
H_i	Key to verifying a person	Access file by key
DK_u	Decryption key	Interpret the encrypted file with the decryption key
$File_u$	File	File encrypted with DK
$I_{\{H_1,\ldots,H_n\}}$	Set the indicator function of $\{H_1, H_2, ..., H_n\}$	Judge a user being in the approved list of the certification center.
J_i	$J_i = \{u: 1 \le u \le m\}$, u is the file ID of S being authorized for access	Specific users could access the file set
$I_J(x)$	Indicating function of J_i set	Judge a user being authorized for the file set
TR	Time-based authentication	For limiting users and file access time
N	User's log-in time	Label the time of a user accessing file
T	Time for access	User and file are accessed at a specific time

(4) CA would select nonrepeating random integers as the decryption key DK_u. Assuming that there are m secret files, they are denoted $\{DK_1, DK_2, ..., DK_m\}$. The decryption key CA, used for encrypting and decrypting the secret file, maintains the confidentiality of DK_u and publicizes the public parameter u.

(5) When there are n users $i = 1, 2, ..., n$ and m files $u = 1, 2, ..., m$, CA defines $J_i = \{u: 1 \le u \le m\}$, S_i is authorized to access u, and J_i reveals whether User i is authorized to access files.

(6) CA defines $I_{J_i}(x) = \begin{cases} 1, & \text{if } x \in J_i \\ 0, & \text{o.w.} \end{cases}$. The indicator function shows the users authorized to access DK_u and is applied to the $B_i(y)$ equation. A user with a legal secret key could access the file.

$$B_i(y) = \left\{ \sum_{u \in J_i} DK_u \left[\prod_{t=1, t \ne u}^{m} \frac{(y-t)}{(u-t)} \right] \right\} \times I_{J_i}(x), x \in R, y \in R \tag{5}$$

(7) The time-based authentication $TR = \min_{TR(N)} |T \bmod 24 - N|$ is added to $A_i(x)$ and $B_i(y)$ so that $A_i(x)$ is updated as

$$A_i(x) = \left\{ \prod_{k=1, k \ne i}^{n} \frac{(x - H_k)}{(H_i - H_k)} + \min_{TR(N)} |T \bmod 24 - N| \right\} \times I_{\{H_1, \ldots, H_n\}}(x), \quad \text{for } i = 1, 2, \ldots, n, x \in R. \tag{6}$$

(8) The time-based authentication $TR = \min_{TR(N)} |T \bmod 24 - N|$ is added to $B_i(y)$ so that $B_i(y)$ is updated as

$$B_i(y) = \left\{ \sum_{u \in J_i} DK_u \left[\prod_{t=1, t \ne u}^{m} \frac{(y-t)}{(u-t)} \right] + \min_{TR(N)} |T \bmod 24 - N| \right\} \times I_{J_i}(x), x \in R, y \in R. \tag{7}$$

(9) CA establishes a key to generate the function.

$$G(x, y) = \sum_{i=1}^{n} A_i(x) B_i(y), x \in R, y \in R \tag{9}$$

When $(x, y) \in R \times R$, $G(x, y) = A_1(x)B_1(y) + A_2(x)B_2(y) + ... + A_n(x)B_n(y)$ and is publicized by CA.

To obtain the decryption key DK_u for file u, User i performs the following process based on the authorized key and file identifiers, which are described as follows.

A. User authentication

User i first verifies whether the personal key H_i is listed in the approved user set $\{H_1, ..., H_n\}$. If H_i exists in this set, the indicator function $I_{\{H_1, \ldots, H_n\}}(H_i)$ is assigned the value 1, meaning

that the user is authorized for further operations. Otherwise, it is 0.

B. Key combination

Once verified, User i constructs a local function $A_i(x)$ that integrates the valid key H_i with other authorized keys through a Lagrange interpolation structure. When H_i is included in the approved list, $A_i(H_i) = 1$, ensuring that only valid users can contribute to the subsequent computation process.

C. File authorization

The system then checks whether the requested file identifier ID_u belongs to the authorized file set J_i. If so, the corresponding indicator function $I_{J_i}(x) = 1$, which signifies that the user has legitimate access to the file u.

D. Decryption key generation

Using the verified user and file parameters, a second mapping $B_i(y)$ is generated. When both $x \in J_i$ and H_i are valid, $B_i(y)$ outputs the value corresponding to the decryption key DK_u; otherwise, the value is 0, indicating that access is not permitted.

E. Result

Finally, the overall function $G(x, y)$ combines these mappings by summing over all authorized users, expressed conceptually as $G(x, y) = \sum_i A_i(x)B_i(y)$. If both H_i and ID_u satisfy the authorization conditions, $G(x, y)$ yields the valid decryption key DK_u; otherwise, the value is 0.

This process ensures that medical sensor data can be securely accessed only by authorized healthcare personnel, effectively preventing data leakage and unauthorized decryption.

4. Dynamic Access Control of Users and Files

The method enhances address system access security management while maintaining computational efficiency. A public function $G(x, y)$ is used to achieve the following objectives. Updating the function and adjusting its parameters can facilitate the addition, removal, and modification of user authority.

$$G(x,y) = \sum_{i=1}^{n} A_i(x) B_i(y),\ x \in R,\ y \in R \tag{9}$$

$G(x, y)$ is further broken down to generate $A_1(x)B_1(y) + A_2(x)B_2(y) + ... + A_i(x)B_i(y)$, where $(x, y) \in R \times R$. The $A_i(x)$ equation is used to verify whether H_i is in the legitimate list of the system to confirm that the user is authenticated by the system, and $B_i(y)$ is related to file access. The computation of the equations $A_i(x)$ and $B_i(y)$ helps determine whether a user can obtain DK_u to decrypt the encrypted data file. $A_i(x)$ and $B_i(y)$ are expressed as below.

$$A_i(x) = \left\{ \prod_{k=1,k\neq i}^{n} \frac{(x - H_k)}{(H_i - H_k)} + \min_{TR(N)} \left| T \bmod 24 - N \right| \right\} \times I_{\{H_1,\ldots,H_n\}}(x), \quad \text{for } i = 1,2,\ldots,n,\ x \in R. \tag{10}$$

$$B_i(y)=\left\{\sum_{u\in J_i} DK_u\left[\prod_{t=1,t\neq u}^{m}\frac{(y-t)}{(u-t)}\right]+\min_{TR(N)}\left|T \bmod 24-N\right|\right\}\times I_{J_i}(x), x\in R \tag{11}$$

(1) Add a user: This process only requires updating the indicator functions $I_{\{H_1,\ldots,H_n\}}(x)$ and $I_j(x)$, The system then creates a new user S_v by generating $A_v(x)$, $B_v(y)$, and J_v, and finally updates the configuration $G(x, y)$. The updated form $G'(x, y) = G(x, y) + A_v(x)B_v(y)$ is fulfilled through a simple addition process.

(2) Delete a user: it resembles the process of adding a user. The system removes parameters associated with S_v in $A_v(x)$, $B_v(y)$, and $G(x, y)$ to delete a user. G is eventually updated to complete the deletion process. Similarly, the system deletes $A_v(x)B_v(y)$, and only subtraction is used in the calculation.

(3) Update user access right: When updating access right, the system redefines user access right J_i. $J_i = \{u: 1 \le u \le m\}$, and u is the file ID, which S_i is authorized to access, where J_i' represents the updated S_i authority. $B_i(y)$ is then updated to $B_i'(y)$; that is, J_i in the $B_i(y)$ equation is replaced by J_i' to acquire $G(x, y) = A_i(x)B_i(y) + A_i(x)B_i'(y)$. It reflects the user's new authority, and the calculation involves only addition and subtraction.

4.1 Adding new security level

The following procedure describes CA can add a new user S_v without redefining the entire structure of the access control system.

Step 1: CA allocates a new secret parameter key H_v for the security level S_v. This key represents the identity of the newly added user and will be used in later computations for authentication and decryption.

Step 2: CA constructs a new polynomial component $A_v(x)$, which follows the same structure as the existing $A_i(x)$ but is extended to include the new user parameter H_v. This adjustment allows $A_v(x)$ to integrate smoothly with the global access function while maintaining backward compatibility with existing users.

Step 3: CA defines a parameter set $J_i = \{u: 1 \le u \le m\}$, representing the file identifiers that the new security level S_v is authorized to access. Each element in this set corresponds to a medical data file or resource accessible to the user.

Step 4: CA establishes a corresponding polynomial $B_v(y)$ to describe the relationship between the user's authorization and file access rights. This function associates the new key H_v with the authorized file identifiers J_v, ensuring that the new user can decrypt only the appropriate data.

Step 5: CA updates the indicator I_{J_v}, which records whether a given file ID belongs to the approved access list for S_v. If the file is authorized, $I_{J_v}(x) = 1$; otherwise, $I_{J_v}(x) = 0$.

Step 6: CA updates the equation $G(x, y) = A_v(x)B_v(y)$ in the original program.

In summary, this procedure explains the process of adding a new security level. CA updates the indicator functions $I_{\{H_1,\ldots,H_n\}}(x)$ and $I_{J_i}(x)$, and constructs $A_v(x)$, $B_v(y)$, and J_v of the new security level S_v. This approach minimizes computational loading and allows the entire process to be updated efficiently through simple addition.

4.2 Removing existing security level

The following Step 1 or 2 can be executed when CA removes an existing security level S_v from the system.

Step 1: CA removes the parameters $A_v(x)B_v(y)$ associated with the security level S_v from the global configuration $G(x, y)$. After the removal, the system updates the structure to a new form $G'(x, y)$, where the authorization information of user v is no longer included. This operation allows CA to revoke a user's access rights efficiently without recalculating the entire key structure.

Step 2: Let J_v denote the set of file IDs that user v is authorized to access. When CA intends to revoke the user's authority, it can simply update J_v' to a null value. By doing so, all the files previously linked to that user become inaccessible, ensuring that the revocation process is immediate and computationally lightweight.

4.3 Updating authorized user

In the initialization stage, CA establishes access rights corresponding to each security level S_i. When an update of user authority is required, CA performs the following steps.

Step 2: CA resets the authorized file ID set $J_i' = \{u: 1 \leq u \leq m\}$, where each element u corresponds to a file ID associated with the updated authority of S_i. After the modification, J_i' represents the new set of accessible files. This procedure ensures that only the intended files remain within the user's authorized range after an update.

Once the new authority set J_i' is determined, CA updates the related parameters in $B_i(y)$ to form an updated expression $B_i'(y)$. The global function $G(x, y)$ then incorporates the new parameters through the combination $A_i(x)B_i(y) + A_i(x)B_i'(y)$, thus generating an updated configuration that reflects the latest user permissions.

If a user's authorization is revoked, CA only needs to clear the corresponding entries in J_i' instead of recalculating all parameters, which significantly reduces computational overhead and allows rapid policy adjustments.

5. Security Analysis

There are issues related to multi-user access rights, which determine the actions different users can perform in the cloud PHR system. As a result, the system introduces a function to modify and share resources stored in the cloud system. To ensure the security and stability of the cloud storage system, user access rights, security, and common attacks, including SQL injection, external, and coordinated attacks, are discussed and analyzed.

5.1 Equation attack

During the updating of a user's authority, there is a risk of an attacker attempting to crack the

polynomial and acquire DK_u through the function $G(x, y)$. When a user has been removed, and other users remain the same, an attacker can subtract old public data $G(x, y)$ from new public data $G'(x, y)$, i.e., $G'(x, y) - G(x, y) = 0$, to acquire DK_u. The mechanism designed in this study can resist an equation attack. The following dynamic updating methods are proposed in Sect. 3.

(1) Add a new security level: When a new user or role is added, the system introduces an additional security level to the global key structure. The new parameters corresponding to this security level are integrated into the existing configuration. This process ensures that newly authorized users can access the relevant data without impacting the validity of existing users' keys.

(2) Delete a security level: Conversely, when a user's authorization is revoked, the system removes the parameters related to that user's security level from the key configuration. This operation effectively invalidates the user's previous access rights while preserving the key relationships of other authorized users.

(3) Update user authority: Additionally, when an existing user's access rights change (for example, when they gain or lose permission for certain files), the system updates their access records and refreshes the associated key parameters. This update ensures that only the new authorization set remains valid, while all outdated key components are excluded from the computation.

(4) Among the three dynamic updating methods, the updated public parameter $G'(x, y)$ is subtracted from the public parameter before updating $G(x, y)$. In this case, an attacker can only acquire $A_v(x)B_v(y)$ or $A_i(x)B_i(y) + A_i(x)B_i'(y)$. In the proposed methods, both $A_v(x)$ and $B_v(y)$ are generated through the Lagrange interpolation polynomial.

$$A_v(x) = \left\{ \prod_{u=1,u\neq v}^{n} \frac{(x - H_u)}{(H_i - H_u)} \right\} \times I_{(H_1,\ldots,H_n)}(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} \tag{12}$$

$$B_v(y) = \left\{ \sum_{j\in J_v} DK_u \left[\prod_{t=1,t\neq u}^{m} \frac{(y - t)}{u - t} \right] \right\} \times I_{J_v}(x) = b_0 + b_1 y + \ldots + b_{m-1} y^{m-1} \tag{13}$$

$$A_v(x) B_v(y) = a_0 b_0 + a_1 b_0 x + a_0 b_1 y + a_1 b_1 xy + \ldots + a_{n-1} b_{m-1} x^{n-1} y^{m-1} \tag{14}$$

When an attacker includes $x = 0$ or $y = 0$ in the inference, the resulting polynomial from $A_v(x)B_v(y)$ would cover a series of unstructured data. This method therefore would not be affected by destructive attack.

5.2 External attack

The public function $G(x, y)$ serves as an important and singular public parameter for an external attacker since the function contains DK_u. Based on protecting the function equation, each security category S_i in the proposed method can use the public function $G(x, y)$ to

incorporate the private super key H_i to derive DK_u. External attackers attempting to acquire the key must decrypt the Lagrange interpolation polynomial to obtain the encryption key. An unknown number of variables would hinder external attackers, who could merely acquire the public function $G(x, y)$ and file ID_u, from reversing the derived DK_u through mathematical calculation. In this case, an attacker cannot illegally acquire any private information of the system through an external attack.

5.3 Coordinated attack

Coordinated attack: In a coordinated attack, two or more authorized users collaborate and share the super key H_i, attempting to derive DK_j beyond the authority or other users' super key H_i.

The security level S_i adopted in this proposal presents a partially ordered set. When S_i is authorized to access S_j, the equation $G(x, y) = A_1(x)B_1(y) + A_2(x)B_2(y) + ... + A_n(x)B_n(y)$, can be applied. A coordinated attack is defined as two or more authorized users collaborating to target another authorized user. The following condition presents as an example, as shown in Table 2.

Coordinated attackers attempt to collect key H, targeting other users in the system with the intention of stealing the decryption key.

$$A_5(x) = \left\{ \frac{(x-H_1)(x-H_2)(x-H_3)(x-H_4)}{(H_5-H_1)(H_5-H_2)(H_5-H_3)(H_5-H_4)} \right\} \times I_{\{H_1,\ldots,H_5\}}(x) \tag{15}$$

$$B_3(y) = \left\{ DK_1 \times \frac{(y-2)(y-3)(y-4)(y-5)}{(1-2)(1-3)(1-4)(1-5)} + DK_4 \times \frac{(y-1)(y-2)(y-3)(y-5)}{(4-1)(4-2)(4-3)(4-5)} \right\} \times I_{J_3}(x) \tag{16}$$

Regarding attackers' security levels $S_3 = \{1, 4\}$ and $S_4 = \{4\}$ and the target security level $S_5 = \{5\}$, S_5 is authorized to visit $File_5$, while S_3 and S_4 do not have the authority to access. In this case, S_3 and S_4 collaboratively launch an attack, attempting to steal S_5 and DK_5. However, S_3 and S_4 simply have DK_3 and DK_4, which cannot pass the $A_5(x)$ test. When they forcibly use such keys for decryption, the result of the Lagrange algorithm appears null. As a result, the attackers cannot unable to obtain the necessary access to the data.

Table 2
User access privileges for each file in coordinated attack scenario.

	Doctor record (DK_1)	Check room record (DK_2)	Legal document (DK_3)	Letter of authority (DK_4)	Insurance company (DK_5)
Doctor (H_1)	1	1	1	1	1
Nurse (H_2)	1	1	0	0	0
Family member (H_3)	1	0	0	1	0
Insurance company (H_4)	0	0	1	1	1
Individual (H_5)	1	1	1	1	1

6. Discussion

In this section, we elaborate on the integration of time-based authentication, secure access control, and fog computing in the PHR cloud platform. This integration is achieved by applying the Lagrange interpolation polynomial, which allows the system to precisely define user access levels and dynamically respond to requests from users with varying privileges. This approach differs from typical methods that rely solely on static authentication or centralized cloud verification. According to previous studies, conventional access control frameworks predominantly functioned within centralized cloud environments, where authentication procedures and key management were entirely administered by a central server. This centralized architecture frequently introduced elevated latency and network congestion, particularly during the handling of large-scale medical datasets.

In contrast, our proposed framework incorporates fog computing nodes between the end devices and the cloud layer. These intermediary fog nodes are responsible for performing partial authentication, validating temporary cryptographic keys, and caching data in close proximity to the data source. By distributing these operations to nearby fog nodes, our architecture significantly reduces communication delays and alleviates the computational load on the central cloud.

Within this framework, partial authentication and data caching are executed by fog nodes positioned nearer to the data sources. Compared with conventional cloud-centric systems, the experimental evaluations demonstrate that the proposed fog-assisted framework achieves lower access latency and faster responses, while preserving an equivalent level of data confidentiality and integrity. This distributed design effectively minimizes communication latency and server workload.

This result is consistent with recent fog-based healthcare studies, which similarly indicate that local verification mechanisms enhance system reliability and reduce network latency. As a result, bringing together time-based authentication with fog computing not only improves access efficiency but also maintains data integrity and real-time system responsiveness. Furthermore, the multi-level user identity mechanism further strengthens the legitimacy of file operations, including adding, modifying, and deleting medical records. Through these combined features, our system effectively overcomes the limitations found in conventional single-layer access control systems and advances both the security and efficiency of EHR management.

7. Conclusions

We conclude that the proposed access control mechanism demonstrates high security, reliability, and efficiency, effectively countering common attacks such as equation, external, and coordinated attacks, which ensures robust protection in dynamic environments. Our approach improves information sharing among hospitals, streamlines security management, and strengthens the protection of personal health data. By combining fog computing with cloud architecture, the system delivers faster responses and lower latency through distributed data

processing at fog nodes, thus improving both access efficiency and system performance. These capabilities further support accurate diagnosis and personalized medical services while reducing unnecessary healthcare costs. Looking ahead, our future work will focus on improving system scalability and integrating advanced encryption techniques within a fog–cloud hybrid architecture to further enhance real-time performance and data security in large-scale healthcare environments.

Acknowledgments

This work was partially supported by the National Science Council of the Republic of China under Grant no. NSTC 114-2221-E-029-028.

References

1 M. S. Huang and Y. C. Lin: Understanding Bitcoin (Dong-Hua Publishing, Taipei, 2014) 5th ed.
2 A. Hosseini, H. Emami, Y. Sadat, and S. Paydar: BMC Med. Inform. Decis. Mak. **23** (2023) 116. https://doi.org/10.1186/s12911-023-02225-0
3 A. M. Tawfik, A. Al-Ahwal, A. S. Tag Eldien, and H. H. Zayed: Sci. Rep. **15** (2025) 16696. https://doi.org/10.1038/s41598-025-97234-9
4 S. B. Othman and M. Getahun: Sci. Rep. **15** (2025) 12358. https://doi.org/10.1038/s41598-025-95531-8
5 G. Saunders, M. Hitchens, and V. Varadharajan: Syst. Rev. **35** (2001) 6. https://doi.org/10.1145/506359.506362
6 G. Coulouris, J. Dollimore, and M. Roberts: Proc. 3rd ACM Workshop on Role-Based Access Control (ACM, 1998) 115–121. https://doi.org/10.1145/286884.286901
7 J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor: IEEE Trans. Knowl. Data Eng. **17** (2005) 4. https://doi.org/10.1109/TKDE.2005.1
8 R. S. Sandhu and P. Samarati: IEEE Commun. Mag. **32** (1994) 40. https://doi.org/10.1109/35.312842
9 R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman: Computer **29** (1996) 38. https://doi.org/10.1109/2.485845
10 D. F. Ferraiolo, R. S. Sandhu, S. I. Gavrila, and D. R. Kuhn: ACM Trans. Inf. Syst. Secur. **4** (2001) 224. https://doi.org/10.1145/501978.501980
11 T. S. Chen and J. Y. Huang: Appl. Math. Comput. **162** (2005) 339. https://doi.org/10.1016/j.amc.2003.12.118
12 P. Kumar, S. S. Rawat, K. Banerjee, A. O. Salau, G. Kumar, and N. Singhal: PLoS One **20** (2025) e0325950. https://doi.org/10.1371/journal.pone.0325950
13 C. H. Liu, Y. F. Chung, T. S. Chen, and S. D. Wang: J. Med. Syst. **36** (2012) 1009. https://doi.org/10.1007/s10916-010-9573-9
14 A. S. M. Kayes, W. Rahayu, T. Dillon, E. Chang, and J. Han: Future Gener. Comput. Syst. **93** (2019) 237. https://doi.org/10.1016/j.future.2018.10.047
15 Y. Liu, S. Wang, and J. Zhang: J. Med. Syst. **48** (2024) 45. https://doi.org/10.1016/j.jmedsys.2024.112345
16 J. Davis, P. Miller, and B. Thompson: IEEE Internet Things J. **11** (2024) 2054. https://doi.org/10.1109/JIOT.2024.39117650
17 T. Evans, M. Clark, and J. Garcia: World J. Adv. Res. Rev. **3** (2024) 122. https://doi.org/10.1234/wjarr.2024.3500
18 H. Lee, D. Kim, and S. Park: IEEE Access **12** (2024) 12345. https://doi.org/10.1109/ACCESS.2024.1234567
19 P. Colombo and E. Ferrari: Int. J. Cloud Comput. **6** (2017) 292. https://doi.org/10.1504/IJCC.2017.086061
20 A. S. M. Kayes, J. Han, and A. Colman: Proc. Web Information Systems Engineering (WISE) 2013, Part I (Springer, 2013) 410–420. https://doi.org/10.1007/978-3-642-41230-1_36
21 S. Mangard, M. J. Aigner, and S. Dominikus: IEEE Trans. Comput. **52** (2003) 483. https://doi.org/10.1109/TC.2003.1183950
22 Y. H. Su: Implementation of AES encryption on CAN bus system using FPGA. M.S. thesis, Dept. of Electronic Engineering, Chung Yuan Christian University, Taiwan (2022).
23 C. J. Pai: Design and implementation of an AES processor architecture. M.S. thesis, Dept. of Electronic Engineering, National Taiwan University of Science and Technology, Taiwan (2019).
24 Z. Y. Wu: Sensors **19** (2019) 2817. https://doi.org/10.3390/s19122817
25 C. C. Yang: Key acceptance factors of cloud-based PHR: A case study of HealthVault. M.S. thesis, Inst. of

Healthcare Information Management, National Chung Cheng University, Taiwan (2015).
26 C. T. Lin: Secure access control for electronic medical records in encrypted cloud databases using attribute-based encryption. M.S. thesis, Inst. of Computer Science and Engineering, National Chiao Tung University, Taiwan (2015).
27 R. Zhao, Y. Chen, and K. Wang: Sci. Rep. **15** (2025) 90908. https://doi.org/10.1038/s41598-025-90908-1
28 C. H. Wang: An authorization management infrastructure based on attribute credentials. M.S. thesis, Dept. of Information Management, Executive MBA Program, College of Management, National Chiao Tung University, Taiwan (2004).

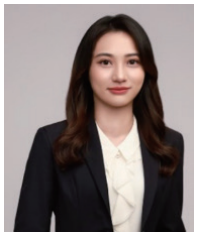
About the Authors

Min-Yuan Ho received his master's degree in information management from Tunghai University in 2023. He is currently a Ph.D. student in the Department of Management Science at National Yang Ming Chiao Tung University, focusing on information management, financial management, deep learning, and information security.

Bi-Huei Tsai received her Ph.D. degree in accounting from the Graduate Institute of Accounting, National Taiwan University, Taiwan, in 2002. She is currently a professor in the Department of Management Science, National Yang Ming Chiao Tung University, Taiwan. She teaches courses in accounting, economics, financial management, and intermediate accounting. Her research interests include environmental economics, event study, and technology forecasting.

Tzer-Shyong Chen received his Ph.D. degree in computer science from the Department of Electrical Engineering, National Taiwan University, Taiwan, in 1996. He is currently a professor in the Department of Information Management at Tunghai University, Taiwan. He has served on the evaluation committee of the Institute of Electrical and Electronics Engineers Taiwan and is also a member of IEEE. He has authored/co-authored over 100 refereed publications. His main research interests are in information security, cryptography, and network security.

Yu-Fang Chung received her B.A. degree in English language, literature, and linguistics from Providence University in 1994, her M.S. degree in computer science from Dayeh University in 2003, and her Ph.D. degree in computer science from National Taiwan University, Taiwan, in 2007. She is currently a professor in the Department of Electronic Engineering and Information Management at Tunghai University, where she is involved in research on information security and cryptography.



Dai-Lun Chiang received her M.S. degree in information management from Tunghai University, Taiwan, in 2015, and her Ph.D. degree in biomedical electronics and bioinformatics from National Taiwan University, Taiwan, in 2020. She is currently an associate professor in the Department of Computer Science and Engineering at Tunghai University, Taiwan. Her research interests include information security, medical information security, big data analytics, and artificial intelligence applications.