

Application of Hierarchical Access Control to Sensor-derived Medical Information Systems

Min-Yuan Ho,¹ Bi-Huei Tsai,^{1*} Tzer-Shyong Chen,²
Yu-Fang Chung,³ and Chia-Yi Lin²

¹Department of Management Science, National Yang Ming Chiao Tung University
No. 1001, University Road, Hsinchu 30010, Taiwan

²Department of Information Management, Tunghai University
No. 1727, Sec. 4, Taiwan Blvd, Xitun District, Taichung City 407224, Taiwan

³Department of Electrical Engineering, Tunghai University
No. 1727, Sec. 4, Taiwan Blvd, Xitun District, Taichung City 407224, Taiwan

(Received September 11, 2025; accepted September 24, 2026)

Keywords: cloud-based medical information system, personal health record secure access control mechanism, data integration, sensor-derived medical information

With the rapid development of information technology and changes in social structure, the public awareness of health issues has continued to increase, driving the evolution of electronic medical records (EMRs). Traditional EMRs can store and retrieve medical records, but they are often confined to individual hospital database environments and cannot fully meet the demand for cross-hospital data sharing in urgent clinical situations. To address this problem, the concept of the personal health record (PHR) emerged, allowing personal medical data to be stored digitally in the cloud. This makes data available in real time and shareable across institutions while strengthening information security and reducing the risks of data theft and impersonation. In this study, we propose a secure access control mechanism for cloud-based medical information systems, which combines Newton interpolation polynomials, matrix operations, and timestamping to strengthen the stability, timeliness, and flexibility of access control. The mechanism distinguishes user privileges across authorization levels, supports multi-role and hierarchical resource access, and updates authorization information and access status in real time when permissions change to maintain data integrity and confidentiality. The system can also incorporate alerts triggered by abnormal values in continuous sensor readings, so that an elevated risk level, such as a sudden drop in blood pressure or an abnormal glucose level, can prompt immediate privilege adjustments or the routing of data to appropriate medical personnel. By incorporating sensor-derived physiological data from medical devices such as wearable health monitors and IoT-enabled bedside diagnostics, the mechanism can dynamically adjust access privileges and information pathways in response to heart rate, blood oxygen saturation, body temperature, and other physiological measurements. Access control is therefore context-aware: system decisions reflect both user roles and sensor-derived clinical urgency. Through this mechanism, patient data can be shared securely and rapidly across healthcare institutions,

*Corresponding author: e-mail: arden@thu.edu.tw
<https://doi.org/10.18494/SAM5936>

potentially reducing data transmission time and enabling real-time access to key medical information during emergencies to support rapid and accurate diagnosis and appropriate treatment planning.

1. Introduction

1.1 Research background

As the public awareness of health issues increases and the frequency of medical visits rises, the efficient access to and the management of medical information have become central concerns in modern healthcare. Traditional electronic medical records (EMRs) preserve the basic functions of paper-based documentation and provide comprehensive records of medical decisions, but most EMRs remain fragmented across individual healthcare institutions. This lack of integration hinders the coordination of medical resources and reduces the efficiency of healthcare services.

Advances in information and network technologies allow healthcare providers to update and share patient information in real time, and support the development of precision medicine and continuous health monitoring. In 2012, the Ministry of Health and Welfare partnered with Microsoft to integrate user-centric personal health records (PHRs) with cloud-based systems. Building on this national effort and the growing public emphasis on patient-centered care, we focus on cross-institutional data sharing and access control to improve diagnostic precision and healthcare efficiency.

Users manage their PHRs and retain complete control over data-sharing preferences.⁽¹⁾ Centralized access to a patient's health information allows physicians to retrieve consultation records and medical instructions from cloud databases during clinical encounters, and improves diagnostic accuracy and the quality and efficiency of healthcare delivery.⁽²⁾

1.2 Research motivation and objectives

The widespread adoption of wireless networks and information technologies has accelerated the development of cloud computing and access control mechanisms. PHR systems are increasingly being migrated to cloud platforms to meet the growing demand for flexible services, multi-user sharing, and real-time updates. Resource monitoring mechanisms are also needed to track cloud operations and virtual machines in real time. When anomalies occur, the system should issue automatic alerts to maintain the stability and security of cloud-based healthcare services. In this study, we propose a patient-centered, cloud-based medical record management framework that integrates time-restricted access control and encryption key management to support concurrent multi-user document access. Newton interpolation increases the flexibility of key management, whereas public-key infrastructure supports access authentication. The framework prevents unauthorized access and gives patients autonomous control over their health data. This control builds trust and strengthens the security of medical information exchange.

Unlike conventional access control mechanisms based mainly on user identity, security level, or predefined roles, the proposed framework integrates hierarchical access control, dynamic

permission management, temporal restrictions, and sensor-derived physiological data within a cloud-based PHR environment. Physiological data, including heart rate, blood oxygen saturation, blood pressure, body temperature, and glucose levels, can be collected from wearable health monitors and IoT-enabled medical devices. When abnormal physiological conditions are detected, the sensor-derived data can serve as contextual information for adjusting access privileges or routing relevant medical information to authorized healthcare personnel. The contribution of this study is the integration of Newton-interpolation-based access control with dynamic, time-restricted, and sensor-aware medical information management.

2. Literature Review

2.1 Overview of access control mechanisms

Access control has long been a major research topic in both academia and industry. The most common access control mechanisms are discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC). DAC grants access privileges based on a user's identity and request context, and allows users to manage access permissions for the objects they own. This mechanism is particularly advantageous in environments that emphasize data sharing and permission flexibility.⁽³⁾ Under MAC, system administrators assign access privileges by comparing user clearance levels with resource classifications, and access is allowed only when the user's security level matches or exceeds the classification of the resource.⁽⁴⁾ Because of its strong security guarantees, MAC has been widely adopted in highly confidential environments such as the military, government, and finance, and it is often integrated with authentication, encryption, and auditing mechanisms to strengthen system security.⁽⁵⁾ RBAC introduces an abstraction layer called "roles" between users and permissions, allowing users to access specific resources according to their assigned roles. This model simplifies permission management, improves administrative efficiency, and reduces security risks arising from errors or misuse.⁽⁶⁾ In sensor-based healthcare systems, contextual access decisions can be refined further using sensor-derived physiological data.

As data confidentiality has become more important, encryption and key management have become essential for safeguarding sensitive information. Encryption transforms data into ciphertext so that only authorized users can decrypt and access it. Key management covers key generation, storage, and rotation, and supports system security and stability. In mobile agent environments, integrating encryption with key management strengthens access control mechanisms.⁽⁷⁾ Before a mobile agent is granted network access, its access path must first be determined, after which its confidential files are encrypted using a symmetric encryption algorithm such as the advanced encryption standard (AES) or the data encryption standard (DES). Each file is encrypted with a unique key, and a hierarchical access control policy is implemented in accordance with the designed strategy.⁽⁸⁾ As cloud computing has matured, technologies such as elliptic curve cryptography (ECC),⁽⁹⁾ bilinear pairing,⁽¹⁰⁾ and identity authentication⁽¹¹⁾ have been introduced to improve encryption and verification efficiency. These advances provide fine-grained and flexible access control, and are particularly suitable for highly dynamic, multi-user environments such as medical information systems. Sensor-derived

physiological data can also provide contextual inputs for access-control decisions when abnormal conditions are detected.

2.2 PHR

A PHR is an electronic health information system that integrates medical visit records, laboratory reports, prescriptions, and vaccination data. It is designed to help individuals monitor their own health status and serves as a practical tool for personal health management.⁽¹²⁾ A key feature of the PHR is that it allows individuals to store, manage, and share their health information as needed. It also enables healthcare professionals to access comprehensive medical histories in real time and improves diagnostic efficiency and the quality of medical services.⁽¹³⁾ Because PHRs contain highly sensitive personal health data, their security is a major concern. Ensuring the confidentiality and integrity of this information requires close collaboration between healthcare institutions and information and communication technology providers. This includes implementing encryption technologies, conducting regular security audits, and enforcing strict access control mechanisms to prevent unauthorized access or misuse and to safeguard the privacy and security of personal health information during storage and transmission.⁽¹⁴⁾ Sensor devices such as wearable monitors and IoT-connected tools can enrich PHRs with real-time physiological data. These data can improve health tracking but also require adaptive privacy safeguards.

2.3 Cloud-based encryption mechanisms

Cloud-based encryption mechanisms encrypt user data before it is stored on cloud servers so that only authenticated users with appropriate access rights can decrypt and retrieve the data. The primary objective of encryption is to protect data confidentiality during transmission and storage and to prevent unauthorized disclosure in cloud environments. Common encryption techniques are classified into two types: symmetric encryption and asymmetric encryption. Symmetric encryption algorithms, such as AES, use the same key for both encryption and decryption. These algorithms are computationally efficient and well suited to encrypting large volumes of data. In contrast, asymmetric encryption methods, such as Rivest-Shamir-Adleman (RSA), use a public–private key pair for encryption and decryption. Although computationally more intensive, asymmetric encryption addresses key distribution challenges and is widely used for key exchange and identity authentication.⁽¹⁵⁾

Cloud environments also typically include a key management system that generates, stores, distributes, and periodically rotates cryptographic keys to support secure and reliable key management.⁽¹⁶⁾ During data transmission, the sender encrypts plaintext with an encryption key before sending it to the cloud. The receiver must then hold the correct decryption key and the corresponding algorithm to restore the ciphertext to its original, readable form. This mechanism reinforces the protection of user data and strengthens the reliability of cloud services. In healthcare systems, validated sensor-derived physiological data may also serve as contextual input for access-control decisions.

3. Research Methodology

3.1 Research framework

In a cloud environment, storage systems must provide sufficient capacity and multilevel access control to handle large volumes of data securely. To maintain the confidentiality and integrity of documents on medical platforms, user operations must be authorized before execution. Time-restricted access and dynamic permission management, however, remain major challenges. In this study, therefore, we incorporate a time-based access mechanism and dynamic control into a cloud-based digital sharing framework to strengthen data owners' control over resources and maintain the security and privacy of medical records during real-time access. Figure 1 illustrates the cloud system architecture.

In the proposed framework, sensor-derived physiological data, including heart rate, blood oxygen saturation, blood pressure, body temperature, and glucose levels, are collected from wearable health monitors and IoT-enabled medical devices. These data are transmitted to the cloud-based PHR system and used as contextual inputs for dynamic authorization management. When abnormal values are detected, the system can update access privileges or route relevant medical information to authorized healthcare personnel. Identity verification, document authorization, decryption-key derivation, and time restrictions are governed by the proposed access-control functions.

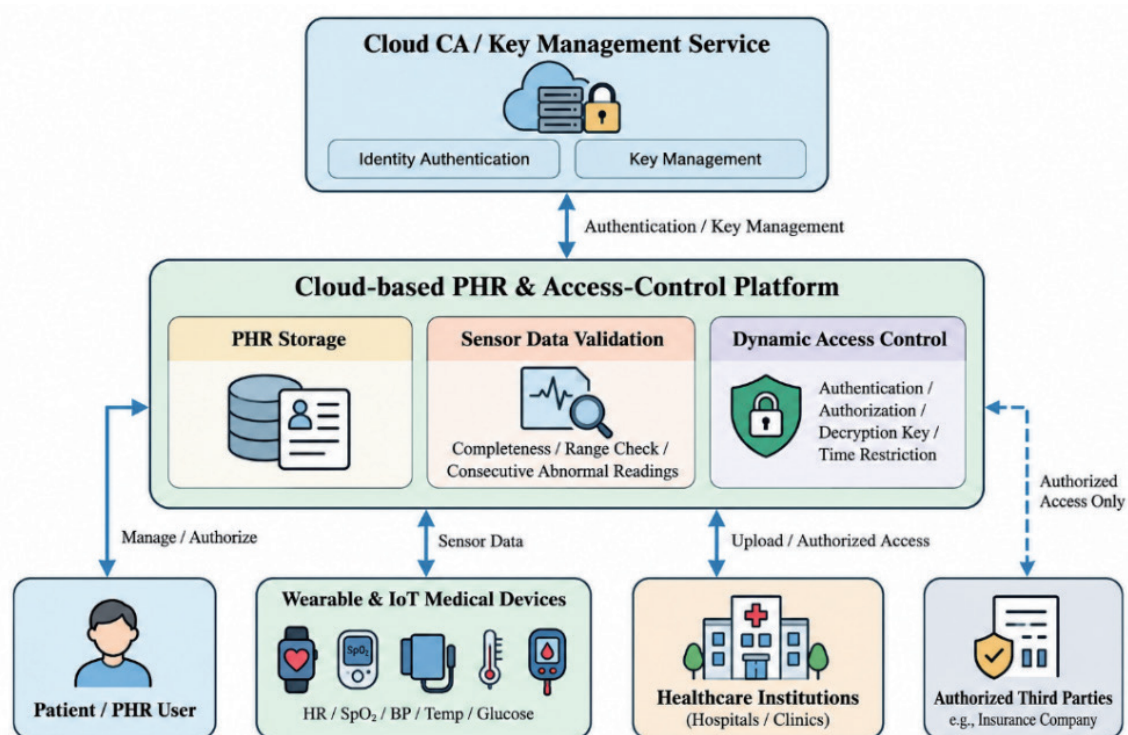


Fig. 1. (Color online) Conceptual application scenario of secure access control in a Taiwan healthcare context.

This study applies the Newton interpolation polynomial to determine a user's highest level of access to a document and to authorize operations such as data transmission, modification, or sharing with designated recipients.⁽¹³⁾ Newton interpolation addresses multi-constraint optimization problems and, when combined with the scalar and hash functions used in timestamping, strengthens document security. Each encryption key is unique and asymmetric, so unauthorized access is difficult within the allowable time window.⁽¹⁷⁾

To reduce false triggers caused by sensor noise, the framework applies a validation step before sensor-derived data are used for access-control adjustments. Sensor readings are first examined for completeness and acceptable ranges. An abnormal event is confirmed only when a measured value exceeds a predefined threshold and consecutive readings support the abnormal condition. Only validated abnormal events are used to trigger privilege adjustments or route relevant medical information to authorized healthcare personnel.

3.2 System initialization setup

In this section, we explain the system initialization process, including the creation of user identities and the authorization of access to individual files. The predefined access matrix is then used to compute each user's access polynomial, which is used to construct the corresponding decryption polynomial. Subsequent sections examine the security of the decryption polynomial and the operation of the dynamic user access mechanism, including changes to user and file permissions within specified time intervals.

The system also uses a partially ordered set S with the relation $\preceq$ to define access relationships, where $\preceq$ is a binary relation on S that is reflexive, antisymmetric, and transitive.⁽¹⁸⁾ Table 1 shows the parameters used in this study.

The timestamp is defined as

$$TR = \min_{T \in T_j(N)} |T \bmod 24 - N|. \quad (1)$$

In this study, all users are assigned the same access duration, that is, $q = 1$, and $T_j(N) \equiv T(N)$. The timestamp therefore simplifies to

$$TR = \min_{T \in T(N)} |T \bmod 24 - N|. \quad (2)$$

Newton interpolation is an established mathematical method. On the basis of this foundation, we developed the access-control functions and equations presented below as part of the proposed framework. To derive DK_k for the required files, the framework uses polynomial functions together with system storage mechanisms to manage access control. The following steps explain how user permissions are verified and accessible files are identified:

Step 1: In this phase, a two-level verification process is used to confirm the validity of the user's key.

(1) The function for the first level of verification is defined as I_H , where $H = \{H_1, H_2, \dots, H_n\}$.

When the set H contains the key of the user, the function outputs 1; if the key is not in the set,

Table 1
Parameter reference table.

Symbol	Definition	Function
S_i	Security level $i = \{1, \dots, n\}$, where the abbreviation for the i -th user n represents the total number of users	Differentiate users' security levels
I_H	Define the indicator function	Determine whether the user is included in the verification authority's approved list, that is, whether the key belongs to set
H_{0i}	Verify the key of user H_{0i} is one of the elements from H_1 to H_n $H_{0i} \in H$	User accesses the document using the decryption key
$I_{[g_i(x)=1]}$	$x \in H$	Determine whether the input key from the user has login permission
DK_k	Document decryption key $k = \{1, \dots, L\}$	Use the decryption key to decrypt the encrypted file k
$File_k$	File k accessible by User i	File k encrypted with DK_k
J_i	j_i represents the complete set of files accessible by user i : $J_i = \{file_{c_1}, \dots, file_{c_{l_c}}\}$. When user i belongs to user category c , l_c denotes the number of files accessible to users in category c . $file_{c_j}$ refers to the j -th file that the user is authorized to access within the set S_i .	Set of decryption keys for files accessible by user i
$File_k$	File k accessible by User i	File k encrypted with DK_k
$I_{J_i}(y)$	Indicator function of the set J_i	Determine whether the user has been authorized to access the file set
B_{kc}	$B_{kc} = \{y_{k,0c}, \dots, y_{k,l_c-1,c}\}$ verifies the key for user type c to access file k .	User i accesses the document using the key
$I_{[g_{kc}(y)=1]}$	$y \in J_i$	Verify whether the decryption key entered by the user is authorized to access the file
TR	Timestamp	Restrict the access time range for both users and files
N	t denotes the login time of the user. $N = [t]$, where $[]$ represents the floor function (Gauss symbol).	Indicate the time at which the user accesses the file
T	Accessible time	Users and files can only be accessed during specific time periods
$T_j(N)$	$j = 1, \dots, q$, where there are q types of usage time slots.	The accessible time set for users in the j -th time slot

Source: Proposed by the authors on the basis of this study.

the output is defined as 0 (i.e., not equal to 1). Accordingly, the indicator function is defined as

$$I_H(x) = \begin{cases} 1, & \text{if } x \in \{H_1, \dots, H_n\} \\ 0, & \text{o.w.} \end{cases} \quad (3)$$

(2) The second level of verification uses the partial function of the access polynomial $A_i(x)$, defined as

$$\frac{1}{f[H_{0i}]} \left(\sum_{m=0}^{n-1} f[H_{0i}, \dots, H_{mi}] \prod_{l=0}^{m-1} (x - H_{li}) \right). \quad (4)$$

As the main verification mechanism, when the user inputs a key such that $I_{[g_i(x)=1]}$ holds, the user has login authorization and the output is 1. If $I_{[g_i(x)=1]}$ does not hold, the user does not have login authorization and the output is defined as 0 (i.e., not equal to 1). The function can be expressed as

$$I_{[g_i(x)=1]} = \begin{cases} 1, & \text{if } g_i(x) = 1 \\ 0, & \text{o.w.} \end{cases}. \quad (5)$$

Step 2: In this stage, second-level verification is performed to confirm the user's access rights before granting access to the document.

The first-level verification function is defined as $J_i = \{file_{c_1}, \dots, file_{c_{l_c}}\}$. For an accessible file $File_k$ (assuming the system contains n users and l_c files to be accessed), the two auxiliary functions are combined to form the function $I_{J_i}(y)$, which indicates that user S_i is authorized by the system to access $File_k$.

If J_i contains the requested file, the user has access rights to the file and the output is 1; otherwise, the user does not have access rights and the output is 0.

If the key is not contained in J_i , the user does not have access rights to the file and the output is defined as 0 (i.e., not equal to 1).

(1) The function can be expressed as

$$I_{J_i}(y) = \begin{cases} 1, & \text{if } y \in J_i \\ 0, & \text{o.w.} \end{cases}. \quad (6)$$

(2) For the second-level verification, the partial function of the polynomial $B_c(y)$, corresponding to the set of files $j = \{1, \dots, l_c\}$ accessible to users in category c , is defined as

$$g_{kc}(y) = \frac{1}{f[y_{k,0c}]} \left(\sum_{k=0}^{l_c-1} f[y_{k,0c}, \dots, y_{k,l_c-1,c}] \prod_{q=0}^{l_c-1} (y - y_{k,qc}) \right). \quad (7)$$

This serves as the primary verification mechanism. When the file-related input supplied by user i satisfies the condition $I_{[g_{kc}(y)=1]}$, the user has access rights to the corresponding file, and the output is 1. If the condition $I_{[g_{kc}(y)=1]}$ is not satisfied, the user does not have access rights to the file and the output is defined as 0 (i.e., not equal to 1). The function can be expressed as

$$I_{[g_{kc}(y)=1]} = \begin{cases} 1, & \text{if } g_{kc}(y) = 1 \\ 0, & \text{o.w.} \end{cases}. \quad (8)$$

Step 3: Apply Newton's interpolation polynomial to construct the corresponding functional polynomial.

- (1) Through the Certificate Authority (CA), a unique secret key H_{0i} is generated for each user $S_i = \{S_1, S_2, \dots, S_n\}$, where $i = \{1, \dots, n\}$. This key is kept confidential from other users. The CA manages all users' H_{0i} keys and defines an indicator function for identity verification:

$$I_H(x) = \begin{cases} 1, & \text{if } x \in H \\ 0, & \text{o.w.} \end{cases} \quad (9)$$

Here, $H = \{H_1, H_2, \dots, H_n\}$ and $I_H(x)$ is used to verify whether access permission is granted. The CA constructs the function $A_i(x)$ for user i by applying a Newton interpolation polynomial to verify the user's identity.

$$\begin{aligned} A_i(x) &= \left\{ \left[\frac{1}{f[H_{0i}]} \left(\sum_{m=0}^{n-1} f[H_{0i}, \dots, H_{mi}] \prod_{l=0}^{m-1} (x - H_{li}) \right) \right] \times I_{[g_i(x)=1]} \right\} \times I_H(x) \\ &= \left\{ g_i(x) \times I_{[g_i(x)=1]} \right\} \times I_H(x), \text{ for } i = 1, 2, \dots, n, x \in R. \end{aligned} \quad (10)$$

- (2) The CA uses distinct random integers as decryption keys DK_k . Assuming there are confidential files, the keys are represented as $\{DK_1, DK_2, \dots, DK_l\}$. As these decryption keys are used for both encrypting and decrypting confidential files, the CA preserves the confidentiality of each DK_k while publishing the corresponding public parameter k .
- (3) When there are n users $i = 1, 2, \dots, n$ and l files $j = 1, 2, \dots, l$, the CA defines $J_i = \{file_{c_1}, \dots, file_{c_l}\}$, where user S_i is authorized to access file k . The set J_i indicates whether user i is granted access to the file.

The CA defines $I_{J_i}(y) = \begin{cases} 1, & \text{if } y \in J_i \\ 0, & \text{o.w.} \end{cases}$. This indicator function identifies the users authorized to access DK_k and is applied in the $B_c(y)$ equation. If the user possesses a valid secret key, access to the file is granted.

$$\begin{aligned} B_c(y) &= \left\{ \sum_{k \in J_i} DK_k \left[\frac{1}{f[y_{k,0c}]} \left(\sum_{k=0}^{l_c-1} f[y_{k,0c}, \dots, y_{k,l_c-1,c}] \prod_{q=0}^{l_c-1} (y - y_{k,qc}) \right) \right] \times I_{[g_{kc}(y)=1]} \right\} \times I_{J_i}(y) \\ &= \left\{ \sum_{k \in J_i} DK_k \times g_{kc}(y) \times I_{[g_{kc}(y)=1]} \right\} \times I_{J_i}(y), \quad i = 1, 2, \dots, n, y \in R. \end{aligned} \quad (11)$$

- (4) A timestamp is incorporated into both $A_i(x)$ and $B_c(y)$ to enforce time-based access control:

$$TR = \min_{T \in T(N)} |T \bmod 24 - N|.$$

- (5) The $A_i(x)$ equation is updated to the following form:

$$A_i(x) = \left\{ \left[\frac{1}{f[H_{0i}]} \left(\sum_{m=0}^{n-1} f[H_{0i}, \dots, H_{mi}] \prod_{l=0}^{m-1} (x - H_{li}) \right) \right] \times I_{[g_i(x)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_H(x) \quad (12)$$

$$= \left\{ g_i(x) \times I_{[g_i(x)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_H(x), \text{ for } i = 1, 2, \dots, n, x \in R.$$

The $B_c(y)$ equation is updated to the following form:

$$B_c(y) = \left\{ \sum_{k \in J_i} DK_k \left[\frac{1}{f[y_{k,0c}]} \left(\sum_{k=0}^{l_c-1} f[y_{k,0c}, \dots, y_{k,l_c-1,c}] \prod_{q=0}^{l_c-1} (y - y_{k,qc}) \right) \right] \times I_{[g_{kc}(y)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_i}(y) \quad (13)$$

$$= \left\{ \sum_{k \in J_i} DK_k \times g_{kc}(y) \times I_{[g_{kc}(y)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_i}(y), \quad i = 1, 2, \dots, n, y \in R.$$

(6) The CA constructs a key generation function for users of category c .

Assuming there are C categories of users ($C = \{1, \dots, c\}$), a key generation function is constructed for user i in category c .

$$G_c(x, y) = \sum_{S_i \in c} A_i(x) B_c(y), \quad x \in H, y \in B_{kc}. \quad (14)$$

User i can combine their secret key H_{0i} with the required file $File_k$ into $G_c(x, y)$ to generate DK_k , which is then used to decrypt file k . The steps of the derivation process are described as follows:

User i incorporates their secret key H_{0i} into the function $I_H(x) = \begin{cases} 1, & \text{if } x \in H \\ 0, & \text{o.w.} \end{cases}$

A. If the secret key H_{0i} of user i is included in the approved key list H established by the CA, then $H_{0i} \in H$ and $I_H(H_{0i}) = 1$; otherwise, if the key is not in the list, then $I_H(H_{0i}) = 0$.

B. User i incorporates their secret key H_{0i} into the equation

$$A_i(x) = \left\{ \left[\frac{1}{f[H_{0i}]} \left(\sum_{m=0}^{n-1} f[H_{0i}, \dots, H_{mi}] \prod_{l=0}^{m-1} (x - H_{li}) \right) \right] \times I_{[g_i(x)=1]} \right\} \times I_H(x),$$

where $H_{0i} \in H$.

If the input $H_{0i'} \neq H_{0i} (i' \neq i)$, then $A_i(H_{0i'}) \neq f[H_{0i'}] / f[H_{0i}] \neq 1$; if $H_{0i'} \notin H$, then $A_i(H_{0i'}) = 0$; if H_{0i} is input, then $A_i(H_{0i}) = f[H_{0i}] / f[H_{0i}] = 1$.

C. User i incorporates the required file $File_k$ into the function $I_{ji}(y) = \begin{cases} 1, & \text{if } y \in J_i \\ 0, & \text{o.w.} \end{cases}$. If the file

is within the authorized access scope, user i is granted access to DK_k , which indicates that $x \in J_i$ and $I_{J_i}(x) = 1$.

D. In $B_c(y) = \left\{ \sum_{k \in J_i} DK_k \left[\frac{1}{f[y_{k,0c}]} \left(\sum_{k=0}^{l_c-1} f[y_{k,0c}, \dots, y_{k,l_c-1,c}] \prod_{q=0}^{l_c-1} (y - y_{k,qc}) \right) \right] \times I_{[g_{kc}(y)=1]} \right\} \times I_{J_i}(y)$,

user i in category c incorporates the desired file $File_k$ into the function. If user i is authorized to access DK_k , then, when $x \in J_i$, $B_c(y_{k,0c}) = DK_k$; otherwise, $B_c(y_{k,0c}) = 0$.

E. User i can compute $G_c(x, y) = \sum_{S_i \in c} A_i(x) B_c(y)$. If $x \in H$ and $y \in B_{kc}$, then $G_c(x, y) = DK_k$. In this case, the user can derive the decryption key; otherwise, $G_c(x, y) = 0$.

4. Authorization Settings for User and Document Access Permissions

System users may, under various circumstances, request that their access permissions be added, modified, or updated in the system, and access permissions must be adjusted to accommodate these changing needs. In this study, we propose a method for managing system access security without compromising computational efficiency. By utilizing a public function $G_c(x, y)$, the following objectives can be achieved through function updates and parameter modifications: (1) user addition, (2) user removal, and (3) user permission updates.

$$G_c(x, y) = \sum_{S_i \in C} A_i(x) B_c(y), x \in H, y \in B_{kc} \quad (15)$$

The function $A_i(x)$ is used to verify whether H_{0i} appears in the system's authorized record list, confirming whether the user is eligible to obtain the secret key required for identity authentication. The function $B_c(y)$ is related to document access. After computing $A_i(x)$ and $B_c(y)$, it can be determined whether the user can retrieve DK_k to decrypt the encrypted data file. $A_i(x)$ and $B_c(y)$ are defined as follows:

$$A_i(x) = \left\{ \left[\frac{1}{f[H_{0i}]} \left(\sum_{m=0}^{n-1} f[H_{0i}, \dots, H_{mi}] \prod_{l=0}^{m-1} (x - H_{li}) \right) \right] \times I_{[g_i(x)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_H(x), \quad (16)$$

for $i = 1, 2, \dots, n, x \in R$.

$$B_c(y) = \left\{ \sum_{k \in J_i} DK_k \left[\frac{1}{f[y_{k,0c}]} \left(\sum_{k=0}^{l_c-1} f[y_{k,0c}, \dots, y_{k,l_c-1,c}] \prod_{q=0}^{l_c-1} (y - y_{k,qc}) \right) \right] \times I_{[g_{kc}(y)=1]} \right. \\ \left. + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_i}(y), i = 1, 2, \dots, n, y \in R. \quad (17)$$

4.1 Security verification of the public function

The public function $G_c(x, y)$ is now applied to illustrate an example of user access permissions for confidential files. As shown in Table 2, suppose that the family member S_4 has legitimate access rights to both $File_3$ and $File_4$. In the first step, the user's secret key H_4 is substituted into $A_4(x)$, which yields the following result:

$$A_4(x) = \left\{ \left[\frac{1}{f[H_{04}]} \left(\sum_{m=0}^{n-1} f[H_{04}, \dots, H_{m4}] \prod_{l=0}^{m-1} (x - H_{l4}) \right) \right] \times I_{[g_4(x)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{H_4}(x) \\ = \left\{ \left[\frac{1}{f[4]} (f[4] + f[4,5](x-4) + f[1,4,5](x-4)(x-5) + f[1,2,4,5](x-1)(x-4)(x-5) \right. \right. \\ \left. \left. + f[1,2,3,4,5](x-1)(x-2)(x-4)(x-5)) \right] \times I_{[g_4(x)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{H_4}(x). \quad (18)$$

Table 2

User access permissions for confidential files.

	Medical order system <i>File</i> ₁	Examination results <i>File</i> ₂	Surgical consent form <i>File</i> ₃	Legal documents <i>File</i> ₄	Insurance documents <i>File</i> ₅
Patient <i>S</i> ₁	0	1	1	1	1
Doctor <i>S</i> ₂	1	1	1	1	1
Nurse <i>S</i> ₃	1	1	1	0	0
Family member <i>S</i> ₄	0	0	1	1	0
Insurance agent <i>S</i> ₅	0	0	0	1	1

Source: Proposed by the authors on the basis of this study.

Note: *File*₂ includes examination results and sensor-derived physiological data, such as heart rate, blood oxygen saturation, blood pressure, body temperature, and glucose measurements. The binary values in this table represent access permissions rather than sensor measurements.

The system verifies whether $I_{H_4} = 1$, and then whether $I_{[g_4(x)=1]}$ also holds. It then computes $A_4(H_{04}) = 1$, whereas $A_4(H_k)$, $k \in \{1, 2, 3, 5\}$, results in a series of random values. Because of the function $\Theta_i(x) = I_{\{1\}}^{(A_i(x))}$, however, the final computed value is 0, indicating that access is not granted. The polynomial $A_i(x)$ can therefore be used to verify whether the user is legitimate and determine whether the user's key H_{0i} exists in the CA's approved key list.

After confirming that the family member *S*₄ is a legitimate user, the system proceeds to verify whether their identity grants valid access permissions to *File*₃ and *File*₄. The set $J_4 = \{3, 4\}$ is then substituted into $B_4(y)$, resulting in the following computation:

$$\begin{aligned}
 B_4(y) &= \left\{ \left(DK_3 \left[\frac{1}{f[y_{3,04}]} \left(\sum_{k=0}^1 f[y_{3,04}, \dots, y_{3,1,4}] \prod_{q=0}^1 (y - y_{3,q4}) \right) \right] \right) \times I_{[g_{34}(y)=1]} \right. \\
 &\quad \left. + DK_4 \left[\frac{1}{f[y_{4,04}]} \left(\sum_{k=0}^1 f[y_{4,04}, \dots, y_{4,1,4}] \prod_{q=0}^1 (y - y_{4,q4}) \right) \right] \times I_{[g_{44}(y)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_4}(y) \quad (19) \\
 &= \left\{ DK_3 \times \left(\frac{1}{f[3]} (f[3] + f[3,4](y-3)) \right) \times I_{[g_{34}(y)=1]} + DK_4 \times \left(\frac{1}{f[4]} (f[4] + f[3,4](y-4)) \right) \times I_{[g_{44}(y)=1]} \right. \\
 &\quad \left. + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_4}(y).
 \end{aligned}$$

The system separately computes the following: $I_{J_4}(3) = 1$, $I_{J_4}(4) = 1$, $B_4(3) = DK_3$, and $B_4(4) = DK_4$. According to this, it is confirmed that the family member *S*₄ holds access rights to the confidential documents *File*₃ and *File*₄, and is authorized to retrieve their respective decryption keys. When the family member *S*₄ attempts to access the surgical consent form *File*₃, their secret key H_{04} and *File*₃ are substituted into the public function $G_c(x, y)$ for computation:

$$G_4(4, 3) = A_1(4)B_1(3) + A_2(4)B_2(3) + A_3(4)B_3(3) + A_4(4)B_4(3) + A_5(4)B_5(3). \quad (20)$$

The decryption key DK_3 for the surgical consent form *File*₃ is contained in $A_4(4)B_4(3)$.

$$\begin{aligned}
A_4(4) &= \left\{ \left[\frac{1}{f[4]} (f[4] + f[4,5](4-4) + f[1,4,5](4-4)(4-5) + f[1,2,4,5](4-1)(4-4)(4-5) \right. \right. \\
&\quad \left. \left. + f[1,2,3,4,5](4-1)(4-2)(4-4)(4-5)) \right] \times I_{[g_4(4)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{H_4}(4) \\
&= \left\{ \left[\frac{1}{f[4]} \times (f[4]) \right] \times I_{[g_4(4)=1]} + 0 \right\} \times I_{H_4}(4) = 1 + 0 = 1, \\
B_4(3) &= \left\{ DK_3 \times \left(\frac{1}{f[3]} (f[3] + f[3,4](3-3)) \right) \times I_{[g_{34}(3)=1]} \right. \\
&\quad \left. + DK_4 \times \left(\frac{1}{f[4]} (f[4] + f[3,4](3-4)) \right) \times I_{[g_{44}(3)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_4}(3) \\
&= [DK_3 \times 1 + DK_4 \times 0 + 0] = DK_3.
\end{aligned} \tag{21}$$

The remaining terms are evaluated to 0 due to insufficient parameter information. Specifically, $I_{H_1}(4)$, $I_{H_2}(4)$, $I_{H_3}(4)$, and $I_{H_5}(4)$ have not been authorized by the CA and thus fail the verification indicator I_H . As a result, these users do not have access rights, and their outputs are 0.

Through the final computation process shown below, the family member S_4 is able to successfully derive the decryption key DK_3 for the surgical consent form $File_3$.

$$\begin{aligned}
G_4(4,3) &= A_1(4)B_1(3) + A_2(4)B_2(3) + A_3(4)B_3(3) + A_4(4)B_4(3) + A_5(4)B_5(3) \\
&= 0 + 0 + 0 + 1 \times DK_3 + 0 = DK_3
\end{aligned} \tag{22}$$

4.2 Dynamic access control for users and documents

4.2.1 User addition

To add a new user S_v to the CA, the following steps are performed:

Step 1: The CA assigns the secret key H_v to a new security level corresponding to S_v .

Step 2: The CA constructs $A_v(x)$. This function is identical to $A_i(x)$, except that n is updated to $n + 1$.

$$\begin{aligned}
A_v(x) &= \left\{ \left[\frac{1}{f[H_{0v}]} \left(\sum_{m=0}^{n-1} f[H_{0v}, \dots, H_{mv}] \prod_{l=0}^{m-1} (x - H_{lv}) \right) \right] \times I_{[g_v(x)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{H_{n+1}}(x) \\
&= \left\{ g_v(x) \times I_{[g_v(x)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{H_{n+1}}(x), \text{ for } v = 1, 2, \dots, n, x \in R, \\
I_{H_{n+1}}(x) &= \begin{cases} 1, & \text{if } x \in \{H_1, \dots, H_{n+1}\} \\ 0, & \text{o.w.} \end{cases}
\end{aligned} \tag{23}$$

Step 3: The CA generates the parameter $J_i = \{file_c, \dots, file_{cl_c}\}$ for S_v .

Step 4: The CA constructs $B_e(y)$, where

$$B_e(y) = \left\{ \sum_{k \in J_v} DK_k \left[\frac{1}{f[y_{k,0e}]} \left(\sum_{k=0}^{l_e-1} f[y_{k,0e}, \dots, y_{k,l_e-1,e}] \prod_{q=0}^{l_e-1} (y - y_{k,qe}) \right) \right] \times I_{[g_{ke}(y)=1]} \right. \\ \left. + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_v}(y) \quad (24)$$

$$= \left\{ \sum_{k \in J_v} DK_k \times g_{ke}(y) \times I_{[g_{ke}(y)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_v}(y) \quad v=1, 2, \dots, n, y \in R,$$

and replaces the original indicator function $I_{J_i}(y)$ with the new indicator $I_{J_v}(y)$.

Step 5: The updated indicator function is defined as $I_{J_v}(y) = \begin{cases} 1, & \text{if } y \in J_v \\ 0, & \text{o.w.} \end{cases}$

Step 6: The CA updates the original scheme by modifying the function

$$G_c(x, y) = A_v(x) B_e(y). \quad (25)$$

In the user addition process described above, the CA updates only the indicators $I_{H_{n+1}}(x)$ and $I_{J_v}(y)$, and establishes the new security level S_v with $A_v(x)$, $B_e(y)$, and J_v . The computational cost required to generate the new security level S_v is significantly reduced, as updating the overall scheme requires only addition operations.

4.2.2 User revocation

To remove an existing security level S_v from the CA, the following steps can be performed:

Step 1: The CA removes the parameters related to the security level S_v from $G_c(x, y)$, that is, $\overline{G_c}(x, y) = G_c(x, y) - A_v(x) B_e(y)$.

Step 2: J_v denotes the set of files $File_k$ that user v is authorized to access. Therefore, the CA can revoke the user's authorization by updating J_v , effectively setting $\overline{J_v}$ into a null.

4.2.3 Modifying an authorized user

In the initial stage of the proposed scheme, the CA defines access rights for the security level S_i . When the system updates an authorized user, the CA performs the following steps:

Step 1: The CA resets $\overline{J_i} = \{file_c, \dots, file_{cl_c}\}$, where $\overline{J_i}$ represents the updated authorization for S_i . When modifying the authorization in the digital sharing mechanism, the CA recalculates the access matrix to generate a new J_i set.

Step 2: Since J_i has been replaced by $\overline{J_i}$, and the information in J_i is associated with $B_c(y)$, the CA updates $B_c(y)$ to $\overline{B_c}(y)$. Assuming the new authorization set $\overline{J_i}$ is granted to user i , when generating the new authorization, the CA adds $A_i(x) \overline{B_c}(y)$ while removing the old user information. The function $G_c(x, y)$ is then updated as

$$\overline{G}_c(x, y) = G_c(x, y) - A_i(x)B_c(y) + A_i(x)\overline{B}_c(y). \quad (26)$$

According to the steps above, constructing J_i allows for straightforward updates to user i 's authorization for accessing digital data. When user i has no authorization, $B_c(y)$ does not require updating. Only the value of $\overline{J}_i$ needs to be set to an empty set.

4.3 Document modification

4.3.1 Adding a document

When a new confidential document is added to the system, the CA assigns access permissions for the new document to each member of the digital sharing system. The verification indicator I_{J_i} is reset to $I_{\overline{J}_i}$, and the corresponding polynomial $B_c(y)$ must also be updated to $\overline{B}_c(y)$. Finally, the document addition is completed by updating the public function

$$\overline{G}_c(x, y) = \sum_{S_i \in c} A_i(x)\overline{B}_c(y), \quad x \in H, y \in B_{kc}. \quad (27)$$

4.3.2 Deleting a document

When an existing document needs to be removed from the original system, the CA modifies the access permissions for each member in the digital sharing system accordingly. The verification indicator I_{J_i} is reset to $I_{\overline{J}_i}$, and the corresponding polynomial $B_c(y)$ must also be updated to $\overline{B}_c(y)$. Finally, updating the public function completes the document removal process.

$$\overline{\overline{G}_c}(x, y) = \sum_{S_i \in c} A_i(x)\overline{\overline{B}_c}(y), \quad x \in H, y \in B_{kc} \quad (28)$$

4.4 Computational complexity and qualitative comparison

Let n denote the total number of users, n_c the number of users in category c , and l_c the number of files accessible to category c . The construction of the Newton interpolation coefficients for $A_i(x)$ requires $O(n^2)$ operations, whereas the construction of $B_c(y)$ requires $O(l_c^2)$. Once the interpolation coefficients are established, the corresponding polynomials can be evaluated in $O(n)$ and $O(l_c)$, respectively. The evaluation of $G_c(x|y)$ therefore has a computational complexity of $O(n_c(n + l_c))$. Moreover, dynamic operations such as user addition, revocation, and permission modification update only the affected functions and parameters rather than reconstructing the entire access-control scheme. Therefore, the computational cost increases polynomially with the numbers of users and files, while localized updates reduce the computational overhead of dynamic authorization management. The computational complexity of the major operations in the proposed access-control mechanism is summarized in Table 3.

Table 3

Computational complexity of the proposed access-control mechanism.

Operation	Computational complexity
User authentication $A_i(x)$	$O(n)$
Document authorization $B_c(y)$	$O(l_c)$
Access/key derivation $G_c(x, y)$	$O(n_c(n + l_c))$
Construction of $A_i(x)$	$O(n^2)$
Construction of $B_c(y)$	$O(l_c^2)$
User addition/removal	Local update without rebuilding the entire scheme
Permission update	Recompute only the affected authorization function

In addition to the computational complexity analysis, a qualitative comparison clarifies the functional differences between the proposed mechanism and conventional access-control approaches. Table 4 shows the comparison of the proposed mechanism with RBAC and attribute-based access control (ABAC) in terms of authorization characteristics, temporal restrictions, dynamic permission management, decryption-key derivation, and sensor-derived contextual information.

As shown in Table 4, RBAC manages access according to predefined roles, whereas ABAC can incorporate multiple users, resources, and contextual attributes. The proposed mechanism integrates hierarchical authorization, time-restricted access, dynamic permission updates, decryption-key derivation, and sensor-derived contextual information within the same access-control framework. This comparison highlights the functional characteristics of the proposed mechanism rather than empirical performance differences.

This study focuses on the mathematical formulation and theoretical verification of the proposed access-control mechanism. The computational complexity analysis and qualitative comparison above demonstrate its characteristics and theoretical feasibility. A full software prototype has not yet been developed, and empirical measurements of latency and throughput have not yet been conducted; these remain directions for future implementation and performance evaluation.

5. Security Analysis

In cloud-based PHR systems, multi-user access control is a major concern. To address this issue, the system supports the modification and sharing of resources stored in the cloud. To maintain the security and stability of cloud storage, we integrated public-key encryption, Newton interpolation polynomials, and access matrices, and used symmetric encryption to protect unpublished keys. Users' access requests must be evaluated against the access matrix to obtain authorization from the CA. Each user holds distinct access privileges, and accessing another user's data would require simultaneously compromising the access polynomial, the Newton interpolation polynomial, and the public-key cryptographic system. In this section, we examine user access rights, potential security issues, and five attack types, namely, external attacks, internal and privilege escalation attacks, collusion attacks, equation-based attacks, and replay attacks.

Table 4
Qualitative comparison of access-control mechanisms.

Comparison Item	RBAC	ABAC	Proposed mechanism
Role-based authorization	✓	✓	✓
Attribute/context-based decision	Limited	✓	✓
Time-restricted access	Not inherent	Possible	✓
Dynamic permission update	Possible	Possible	✓
Decryption-key derivation	Not inherent	Not inherent	✓
Polynomial-based authorization	×	×	✓
Sensor-derived contextual input	Not inherent	Possible	✓

5.1 External attacks

External attacks refer to attempts by unauthorized users to exploit public parameters to gain access rights, with the intent of obtaining DK_u or decrypting files to access personal data stored in the cloud.

The most important public parameter for external attackers is the public function $G_c(x, y)$, as it contains DK_k . In the proposed scheme, this function must be securely protected. At each security level, S_i can use $G_c(x, y)$ in combination with its secret key H_{0i} to derive DK_k . If an external attacker attempts to acquire DK_k , they must compromise the Newton interpolation polynomial to obtain the decryption key. However, given that only $G_c(x, y)$ and $File_k$ are available to the attacker, the number of unknown variables makes it infeasible to reverse-engineer DK_k through mathematical computation. As a result, external attackers cannot illicitly obtain private information from the system through such attacks.

The CA can adopt any encryption method to generate DK_k in this scheme. Symmetric-key systems, for example, apply the principles of diffusion and confusion to prevent attackers from using statistical analysis to break the encryption. With current technology, such cryptographic systems remain highly resistant to compromise, so attackers cannot extract meaningful information directly from encrypted files. Even if attackers intercept sensor-derived physiological data, the system's encryption keeps these data protected during transit and storage.

5.2 Internal and privilege escalation attacks

This type of attack is more commonly observed among internal system members. It typically occurs when a legitimately authorized user with lower-level privileges, denoted as S_i , attempts to exploit the public function $G_c(x, y)$ along with their own secret key H_{0i} to illegally derive the secret keys of other authorized users with higher-level access. Through this method, the attacker seeks to obtain unauthorized access to confidential documents beyond their granted permissions.

As an illustrative example, suppose there are five categories of users ($C = 5$), each containing one user, specifically, S_1, S_2, S_3, S_4 , and S_5 . Among them, a nurse S_3 attempts to access legal documents $file_4$ and insurance documents $file_5$, which are only accessible to a doctor S_2 , as shown in Table 2.

Under normal circumstances, doctors possess higher access privileges than nurses $J_2 = \{1, 2, 3, 4, 5\}$, $l_2 = 5$; $S_3 = \{1, 2, 3\}$, $l_3 = 3$. In this example, from the doctor's S_2 perspective, the nurse

S_3 would be considered an attacker attempting to exploit their own secret key H_{03} by substituting it into the public function $G_c(x, y)$ in order to deduce the doctor's secret key H_{02} , and thus gain unauthorized access to legal documents $file_4$ and insurance documents $file_5$, which are only accessible to the doctor S_2 .

During the derivation process, the nurse S_3 can substitute $(H_{03}, 1)$, $(H_{03}, 2)$, and $(H_{03}, 3)$ into the public function $G_3(x, y)$ to obtain the decryption keys DK_1 , DK_2 , and DK_3 for the medical order system $file_1$, examination results $file_2$, and surgical consent form $file_3$, respectively. However, when attempting to input $(H_{03}, 4)$ and $(H_{03}, 5)$ into $G_3(x, y)$, the nurse cannot obtain the decryption keys for the legal documents $file_4$ and insurance documents $file_5$. In other words, the nurse S_3 is unable to access the decryption keys DK_4 and DK_5 , which are restricted to the doctor S_2 .

The above result can be generalized to other unauthorized privilege-escalation attempts. For an authenticated user S_i , access to a file $File_k$ is permitted only when the file belongs to the authorized set J_i . If $File_k \notin J_i$, the authorization indicator returns 0 and $B_c(y)$ cannot generate the corresponding decryption key DK_k . Consequently, $G_c(x|y)$ cannot return a valid decryption key for the unauthorized file. Therefore, even if a user successfully passes identity authentication, the user cannot obtain access privileges beyond the predefined authorization scope.

5.3 Collusion attacks

Collusion attacks refer to scenarios in which two or more authorized users collaborate by sharing their secret keys H_{0i} in an attempt to derive decryption keys DK_k beyond their authorized scope or those associated with other users' secret keys H_{0i} . In the proposed approach, the security levels S_i follow a partially ordered relationship. When S_i is authorized to access S_j , the corresponding public function can be expressed as

$$G_c(x, y) = A_1(x)B_1(y) + A_2(x)B_2(y) + \dots + A_n(x)B_n(y). \quad (29)$$

Collusion attacks thus involve multiple authorized users jointly targeting the decryption key of another authorized user. Even if such secret keys are forcibly applied, however, the Newton interpolation operation yields a null result, so the attackers cannot access the protected data.

5.4 Equation-based attacks

Equation-based attacks refer to attempts by adversaries to derive the decryption key DK_k by exploiting the formula $G_c(x, y)$ through mathematical algorithms. Such attacks are more likely to occur during access-right updates, such as the addition or removal of users or changes to file permissions. Attackers may attempt to analyze changes in $G_c(x, y)$ before and after updates to identify patterns or vulnerabilities in the polynomial structure.

To mitigate this threat, the proposed scheme introduces three dynamic update strategies to strengthen the security of access-right modifications.

- (1) Addition of new users: $\overline{G_c(x, y)} = G_c(x, y) + A_v(x)B_e(y)$

When a new user is added to the system, an attacker may attempt to subtract the original public function $G_c(x, y)$ from the updated version $G_c(x, y)$ in order to obtain $A_v(x)B_e(y)$. However, since $A_v(x)$ and $B_e(y)$ are not publicly disclosed, the attacker cannot decrypt any information contained within $A_v(x)$ and $B_e(y)$.

- (2) Removal of users: $\overline{G_c(x, y)} = G_c(x, y) - A_v(x)B_e(y)$

When a user is removed from the system, an attacker may attempt to subtract the updated public function $\overline{G_c(x, y)}$ from the original public function $G_c(x, y)$ to derive $A_v(x)B_e(y)$. However, the polynomial remains indecipherable and cannot be compromised using this method.

- (3) Updating user access authorization: $\overline{G_c(x, y)} = G_c(x, y) - A_i(x)B_c(y) + A_i(x)\overline{B_c(y)}$

When user authorization is updated, an attacker may attempt to subtract the original public function $G_c(x, y)$ from the updated public function $\overline{G_c(x, y)}$ to obtain $A_i(x)B_c(y) + A_i(x)\overline{B_c(y)}$. However, since $A_i(x)$, $B_c(y)$, and $\overline{B_c(y)}$ are not publicly disclosed, the attacker is unable to decipher the embedded information.

In the aforementioned method, $A_v(x)$ and $B_e(y)$ are both polynomials constructed using Newton interpolation. Therefore, a multiplication operation must be applied to transform $A_v(x)$ and $B_e(y)$ into a polynomial of degree $(n-1)(m-1)$ th with two variables.

$$\begin{aligned}
 A_v(x) &= \left\{ \left[\frac{1}{f[H_{0v}]} \left(\sum_{m=0}^{n-1} f[H_{0v}, \dots, H_{mv}] \prod_{l=0}^{m-1} (x - H_{lv}) \right) \right] \times I_{[g_v(x)=1]} \right. \\
 &\quad \left. + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{H_{n+1}}(x) \\
 &= \left\{ g_v(x) \times I_{[g_v(x)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{H_{n+1}}(x) \\
 &= a_0 + a_1x + \dots + a_{n-1}x^{n-1}.
 \end{aligned} \tag{30}$$

$$\begin{aligned}
 B_e(y) &= \left\{ \sum_{k \in J_v} DK_k \left[\frac{1}{f[y_{k,0e}]} \left(\sum_{k=0}^{l_e-1} f[y_{k,0e}, \dots, y_{k,l_e-1,e}] \prod_{q=0}^{l_e-1} (y - y_{k,qe}) \right) \right] \times I_{[g_{ke}(y)=1]} \right. \\
 &\quad \left. + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_v}(y) \\
 &= \left\{ \sum_{k \in J_v} DK_k \times g_{ke}(y) \times I_{[g_{ke}(y)=1]} + \min_{T \in T(N)} |T \bmod 24 - N| \right\} \times I_{J_v}(y) \\
 &= b_0 + b_1y + \dots + b_{m-1}y^{m-1}
 \end{aligned} \tag{31}$$

$$A_v(x)B_e(y) = a_0b_0 + a_1b_0x + a_0b_1y + a_1b_1xy + \dots + a_{n-1}b_{m-1}x^{n-1}y^{m-1}. \tag{32}$$

If an attacker incorporates $x = 0$ or $y = 0$ into the inference process, the resulting polynomial message of $A_v(x)B_e(y)$ will consist of unstructured data. Therefore, the method proposed in this study is resistant to destructive attacks.

5.5 Replay attacks

The timestamp mechanism restricts each access request to a predefined valid time interval. A previously captured request retransmitted outside the permitted access period therefore cannot satisfy the time constraint and does not proceed to the subsequent authorization process. Replay attempts within the same valid time interval, however, are not explicitly distinguished by the current mechanism. Additional freshness mechanisms, such as unique request identifiers or nonces, may be incorporated in future implementations to strengthen replay protection.

6. Conclusions

In this study, we proposed a secure access control mechanism with time-based authentication for deployment in a cloud-based PHR system. The primary objective is to establish a rigorous and trustworthy resource access framework that enables users and healthcare institutions to efficiently track and manage health-related data. Using Newton interpolation polynomials, the system distinguishes access privileges among users with different authority levels and addresses multi-role and diverse access requests in cloud environments. Timestamping adds temporal control and flexibility to the access control mechanism. The framework can also incorporate sensor-derived physiological data from wearable health monitors and bedside diagnostic tools, allowing access privileges to be updated in response to real-time physiological conditions. Through identity-based privilege assignment, only users with sufficient authorization or ownership may add, modify, or delete medical records. When access rights change, the system updates permission information and access status in real time, supporting efficient access management while preserving data integrity and confidentiality.

The theoretical security analysis indicates that the proposed mechanism can resist the equation-based, external, and collusion attacks examined in this study. It supports the security and usability of the medical information system while enabling secure data exchange and sharing across institutional boundaries. The mechanism also simplifies security management and supports dynamic authorization updates. The encryption scheme is designed to support the secure deployment of PHRs across multiple medical institutions and the controlled sharing of health information, which may contribute to diagnostic precision and personalized healthcare services. By integrating sensor-derived physiological data into the cloud infrastructure, the platform supports continuous monitoring and adapts access control to each patient's real-time condition. It also supports individual health management and the broader development of secure, interoperable medical information systems.

Acknowledgments

This work was supported partially by the National Science Council of the Republic of China under Grant no. NSTC 114-2221-E-029-031.

References

- 1 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>
- 2 J. L. Lin, S. C. Chang, H. W. Tian: Integrated care project for promoting health and well-being technologies, Tech. Rep. MOHW111-IM-I-114-000001 (Ministry of Health and Welfare, Taiwan, 2022).
- 3 U. N. Cobrado, S. Sharief, N. G. Regahal, E. Zepka, M. Mamauag, and L. C. Velasco: Inform. Med. Unlocked **49** (2024) 101552. <https://doi.org/10.1016/j.imu.2024.101552>
- 4 R. S. Sandhu and P. Samarati: IEEE Commun. Mag. **32** (1994) 40. <https://doi.org/10.1109/35.312842>
- 5 C. P. Pfleeger and S. L. Pfleeger: Security in Computing (Prentice Hall PTR, Upper Saddle River, NJ, 2003) 3rd ed.
- 6 M. T. de Oliveira, Y. Verginadis, L. H. A. Reis, E. Psarra, I. Patiniotakis, and S. D. Olabarriaga: Expert Syst. Appl. **213** (2023) 119271. <https://doi.org/10.1016/j.eswa.2022.119271>
- 7 Y. Xiong, Y. Tan, J. Huang, and C. F. Lai: Inf. Sci. **346–347** (2016) 289.
- 8 T. S. Chen, Y. F. Chung, and C. S. Tian: Proc. 28th Annual Int. Computer Software and Applications Conf. (COMPSAC 2004) (IEEE, 2004) 396–397. <https://doi.org/10.1109/CMPSAC.2004.1342869>
- 9 K. Veerabadrappa, C. B. Naikodi, S. B. Venkataswamy, and H. K. Narayanaswamy: Int. J. Intell. Eng. Syst. **17** (2024) 814. <https://doi.org/10.22266/ijies2024.1231.62>
- 10 H. Dahshan, H. M. Eldeeb, and A. E.-D. R. Shehata: J. Inf. Hiding Multimed. Signal Process. **15** (2024) 144. <https://www.jihmsp.org/2024/vol15/N3/02.JIHMSP-240506%20%282%29.pdf>
- 11 G. Shi, M. Qi, Q. Zhong, N. Li, W. Gao, L. Zhang, and L. Gao: Sensors **25** (2025) 1857. <https://doi.org/10.3390/s25061857>
- 12 H. S. Chen, C. C. Liao, H. B. Hou, and W. C. Chao: Formosan J. Med. **22** (2018) 283. [https://doi.org/10.6320/FJM.201805_22\(3\).0008](https://doi.org/10.6320/FJM.201805_22(3).0008)
- 13 Y. T. Huang, D. L. Chiang, T. S. Chen, S. D. Wang, F. P. Lai, and Y. D. Lin: Knowl.-Based Syst. **236** (2022) 107679. <https://doi.org/10.1016/j.knosys.2021.107679>
- 14 E. Li, O. Lounsbury, J. Clarke, H. Ashrafian, A. Darzi, and A. L. Neves: BMC Med. Inform. Decis. Mak. **23** (2023) 158. <https://doi.org/10.1186/s12911-023-02255-8>
- 15 D. Clemente-Lopez, J. de J. Rangel Magdaleno, and J. M. Muñoz Pacheco: Internet Things **25** (2024) 101032. <https://doi.org/10.1016/j.iot.2023.101032>
- 16 D. Goyal, S. Jain, and S. K. Sood: Int. J. Adv. Comput. Sci. Appl. **10** (2019) 175.
- 17 C. J. Pai: Design and implementation of AES processor architecture, M.S. thesis, Dept. Electron. Eng., Nat. Taiwan Univ. of Sci. and Technol., Taiwan (2019) (unpublished).
- 18 B. A. Davey and H. A. Priestley: Introduction to Lattices and Order (Cambridge University Press, Cambridge, UK, 2002) 2nd ed.

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.



Chia-Yi Lin received her B.B.A. degree from the Department of Information Management, Tunghai University, Taiwan, in 2021, and her M.B.A. degree from the Graduate Institute of Information Management, Tunghai University, Taiwan, in 2023. Her research currently involves information security and management.