sensors

MDPI

Article

A Secure and Efficient Digital-Data-Sharing System for Cloud Environments

Zhen-Yu Wu

Department of Information Management, National Penghu University of Science and Technology, Penghu 880, Taiwan; zywu@gms.npu.edu.tw; Tel.: +886-06-926-4115 (ext. 5211 or 5422)

Received: 25 April 2019; Accepted: 20 June 2019; Published: 24 June 2019

Abstract: "Education Cloud" is a cloud-computing application used in educational contexts to facilitate the use of comprehensive digital technologies and establish data-based learning environments. The immense amount of digital resources, data, and teaching materials involved in these environments must be stored in robust data-access systems. These systems must be equipped with effective security mechanisms to guarantee confidentiality and ensure the integrity of the cloud-computing environment. To minimize the potential risk of privacy exposure, digital sharing service providers must encrypt their digital resources, data, and teaching materials, and digital-resource owners must have complete control over what data or materials they share. In addition, the data in these systems must be accessible to e-learners. In other words, data-access systems should not only encrypt data, but also provide access control mechanisms by which users may access the data. In cloud environments, digital sharing systems no longer target single users, and the access control by numerous users may overload a system and increase management burden and complexity. This study addressed these challenges to create a system that preserves the benefits of combining digital sharing systems and cloud computing. A cloud-based and learner-centered access control mechanism suitable for multi-user digital sharing was developed. The proposed mechanism resolves the problems concerning multi-user access requests in cloud environments and dynamic updating in digital-sharing systems, thereby reducing the complexity of security management.

Keywords: Education Cloud; confidentiality; privacy exposure; access control; digital sharing

1. Introduction

As the name suggests, e-learning involves learning through digital media [1]. E-learning can be traced back to the Programmed Logic for Automatic Teaching Operations (PLATO) introduced by Prof. Patrick Suppes of Stanford University and Prof. Don Bitzer of the University of Illinois in 1960 [2]. The PLATO system was capable of teaching students reading and writing skills through courses involving computer-assisted instruction (CAI). E-learning has gradually evolved from standalone CAI to online learning platforms that cater to mass users [3]. For example, the online courses offered through the University of Phoenix's online platform and through OpenCourseWare, operated by the Massachusetts Institution of Technology, provide learning opportunities for people all over the world through digital sharing.

Amidst advancements of network technologies, the prevalence of digital-multimedia-information use has accelerated. For example, application of digital multimedia information has become a mainstream in teaching. This approach not only overcomes the spatial and temporal constraints of learning but also contributes to more engaging, interactive, and immediate learning experiences, which motivate learners and stimulate their interests. These features support users' comprehension of new content, and reinforce previously learned concepts [3–6]. E-learning achieved steady development

worldwide after the National Institute of Standards and Technology (NIST) announced endorsement of cloud computing and the application of cloud computing in Internet applications in 2009 [7].

"Education Cloud" refers to the application of cloud computing [8] to establish data-based learning environments and use of comprehensive digital technologies in educational contexts. Related provisions are as follows:

(1) Create a learning environment centered on digital technology and data, provide teachers and students with an online education portal, and ensure that teachers are able to apply the various e-learning tools provided to them to teach their students.
(2) Establish a comprehensive online school network that offers wireless Internet and roaming mechanisms to facilitate teaching and learning at the school. Create a data application environment free of spatial constraints, and apply new technologies such as voice over Internet protocol to reduce administration cost and enhance interconnection efficiency
(3) Reduce digital gaps and create balanced digital environments to enable future students to develop new ways of learning through the novel teaching approach, thereby fostering their independent thinking and problem-solving abilities, which constitute the advantages of the new generations.
(4) Consolidate various cloud learning content and services and meet learner-centered resource demands; learn and identify suitable resources through the cloud to achieve cloud-based learning. Satisfy environmental demands for learning prosperity, autonomy, and convenience, thereby facilitating customized and autonomous learning.

In cloud-sharing environments, immense amounts of digital resources, data, and teaching materials must be stored in robust data-access systems. These systems must be equipped with effective security mechanisms to guarantee confidentiality and ensure the integrity of the cloud-computing environment [9–11]. To minimize the risk of privacy exposure, digital sharing service providers should provide encryptions for the digital resources, data, and teaching materials. The data-access systems must also offer editors and authors control over what data or materials they share. In addition, the data in these systems must be accessible to e-learners. In other words, data-access systems should allow conventional service providers to encrypt their data and should also provide control mechanisms by which users may access the data [12].

Access control is a form of access protection for data systems that prevents unauthorized users from corrupting, deleting, or modifying data. Specifically, it is a protection mechanism that governs what items are accessible to system users and the extent of the items' accessibility. Therefore, access control is a key component in the field of network communication and data security. When a user wants to retrieve a document stored in a data system, the user's identity is authenticated by a username and password to verify that the user possesses the necessary authorization to add, delete, modify, or access specific documents [13].

Cloud-based digital sharing systems are formatted to enable use by numerous users; however, provision of simultaneous access control to numerous users may overload a system's computational capacity or increase the requirement for physical management of the system [12,14]. Authorized users and users requesting authorization may access the system from different channels. Generally, demands from existing users and new user requests are immense and unpredictable. If all users are able to manage their subscriber base directly, security management (management of encryption and decryption keys and certifications) will be complicated by the excessive number of users [15].

By directly storing digital resources in the cloud, authorized users can manage these resources from any location and at any time without depending on an online administrator to approve their request. In other words, the accessibility and usability of digital sharing systems are unrestricted. However, following the continual additions and modifications of content in digital sharing systems, the digital information stored in the cloud may originate from various cities and counties, and authorized users may submit requests to the cloud server at any time to retrieve the latest resources or teaching

materials. Therefore, the dynamic update function of digital sharing systems must be implemented in the provision of cloud services.

Although previous studies have proposed various encryption techniques to prevent the unauthorized data access, most of these techniques are targeted at single-user systems. In cloud environments, digital sharing systems do not target single users; rather, they provide secure and efficient access control mechanisms that allocate different permissions to different users. To overcome the challenges of non-single-user applications and to harness the many benefits of combining digital sharing systems with cloud computing, we developed a cloud-based and learner-centered access control mechanism suitable for multi-user digital sharing. The mechanism resolves the problems related to multi-user access requests in cloud environments and dynamic updating in digital sharing systems, thereby reducing the complexity of security management.

The rest of this paper is organized as follows. Section 2 offers a review of related works. Section 3 proposes the digital-data-sharing system. Section 4 demonstrates the security of the proposed system and evaluates system performance. Finally, Section 5 concludes this paper.

2. Related Works

2.1. Access Control Mechanisms

Access control has been extensively studied, and scholars have proposed a variety of access control mechanisms, such as access control matrices, access control lists, capability lists, and role-based access control (RBAC). Access control matrices are the simplest of these mechanisms to use in management of system resource-access [16]. When a user submits a request to access system resources, the system uses the user's position relative to the requested object in an access control matrix to determine the legitimacy of the request.

The concepts behind access control lists and capability lists are similar. Both mechanisms compile authorization logs into lists. In an access control list, authorization logs are compiled into columns in the access control matrix, where system resources constitute the base matrix and users are represented in a linked list. In a capability list, permission logs are compiled into rows in the access control matrix, where users constitute the base matrix, and system resources are represented in a linked list. Access control lists facilitate the management of system resource requests. However, searching for a specific user in an access control list is time intensive. By comparison, the advantages of capability lists correspond with the disadvantages of access control lists.

In addition to the aforementioned access control mechanisms, several access control models have been proposed. For example, task-based access controls (TBACs) authenticate access requests and periods based on task requirements [17], temporal RBACs (TRBAC) authenticate role permissions based on the changes in time intervals [18], rule set-based access controls authenticate permissions based on the system's security strategy [19], and spatial RBAC (SRBAC) authenticates role permissions based on changes in spatial locations [20]. These access control models can be employed independently or combined with other control models. For instance, TBACs can be used simultaneously with RBACs in access systems [17], or a hybrid RBAC can be adopted as the access control mechanism in large and complex organizations [21]. Furthermore, access control mechanisms can be incorporated into other system structures, such as RBAC online-payment systems [22].

At present, commonly applied access controls can be categorized into three types: discretionary access control (DAC), mandatory access control (MAC), and RBAC.

In DAC mechanisms, access is granted based on user identity and the specific action related to the access request. Users are able to manage the access permission of the objects they own without intervention from system administrators. In sum, DAC enable the transfer of object authorities and is suitable for developing environments for data sharing and autonomous application of authority [23].

Although DAC models provide flexible access control mechanisms, they cannot guarantee the integrity of data after authorization [24]. For example, users that are authorized to access a particular

document may download the document onto a storage device and then transfer the document to others without the authorization of the owner. Thus, in DAC systems, document owners are unable to track the authorized users' transfer of their documents, and document receivers cannot determine whether the rights of the document belong to the document provider or whether the document was merely transferred from another source.

MACs prohibit users from freely allocating access permissions, and permission allocation rights belong exclusively to a system administrator [25]. MACs assign security levels and category labels to all subjects and objects in a system. When a user requests access to an object, the system compares the labels of the user and the object. If the users' access permissions correspond with or exceed the confidentiality level of the object, the request of the user is authorized; otherwise, the request is rejected [24]. For example, assume the system sets the security clearance of User A at 2 and that of User B at 4, and that the security levels of Documents A, B, and C are 1, 3, and 6, respectively. User A is able to view Document A, which is categorized under a security level lower than his or her clearance, whereas User B is authorized to view Documents A and B. The security level of Document C is higher than the clearance of both users; therefore, they cannot access Document C. The application of MAC is more complicated than that of DAC, making MAC-based systems suitable for environments with stringent security requirements, such as national defense departments.

The RBAC was introduced by Sandhu et al. in 1996 [13] It was subsequently incorporated and standardized by the NIST in 2011, and was renamed as NIST RBAC [26]. RBACs create "role" elements between the "user" and "access permission" elements in a system, enabling users to access documents through the "role" element.

As awareness of digital data confidentiality rose in 2004, Chen et al. proposed an access control mechanism that combined encryption and key management, and applied the mechanism in a mobile agent environment [27–29]. Before a mobile agent is approved for work on the Internet, the transmitting host decides which hosts will be visited by and what data is accessible to the agent. In addition, the owner of the mobile agent must first determine pathways and access strategy. The owner of the agent then encrypts his or her confidential files with separate keys using a symmetric encryption system, such as the Advanced Encryption Standard (AES), Data Encryption Standard (DES), or International Data Encryption Algorithm [30]. Finally, various access permissions are established based on the access control strategy, and a hierarchical structure is created based on the level of the access permissions. The owner of the agent provides a superkey to each host and publishes specific public parameters of the agent. The hosts then use their superkeys to access data from hosts with security levels below their clearance.

Thereafter, access mechanisms based on an elliptic curve, bilinear pairing, and ID authentication, and mechanisms with migration and time constraints were sequentially introduced [31–34]. Cloud environments matured by 2012 and Liu et al. proposed a dynamic access framework that achieved accurate access control of cloud data and logs in a multi-user setup. The framework was incorporated into a medical environment to maximize patients' control of their medical records. The system ensured privacy by only granting access to doctors, pharmacists, nurses, and researchers [14,15,35].

Recently, context-aware access control (CAAC) models have been developed, extending the basic RBAC authorization model, which determine whether users' requests to limit the access permissions for privacy data and information based on the dynamically changing contextual conditions, such as related resources, environments, user properties, software services [36,37]. For example, Schefer-Wenzl and Strembeck proposed the fuzzy model with an ontology-based approach that captures contextual conditions for mobile business processes [38]. Hosseinzadeh et al. used ontological techniques and Web Ontology Language (OWL) of modeling context-aware role-based access control scheme for smart spaces [39]. Trnka and Cerny proposed a CAAC scheme based on using security levels, which are granted to user based on his/her context [40]. Colombo and Ferrari proposed a roadmap to enhance the data protection functionalities of NoSQL datastore and then design a CAAC mechanism for MongoDB [41,42]. Kayes et al. developed several CAAC systems by considering a wide variety of

contextual conditions, the relationship context information utilizing the process of inferring implicit knowledge, and the purpose-oriented situation information based on the currently available context information [43–45].

2.2. Lagrange Interpolation Polynomial

Following, is a brief introduction to Lagrange interpolation polynomial [30], which we have adopted for encryption and decryption processes. In numerical analysis or other applications, many practical problems are represented through functions to express intrinsic relationships or regularity. However, the precise relationship between variable x and variable y of many functions are extremely complex, and cannot be determined through experiments. The method of Lagrange interpolation enables us to obtain a polynomial which passes through a finite set of points in the x-y plane. The polynomial obtained by this method is called the Lagrange polynomial. Mathematically, the Lagrange interpolation polynomial can obtain a polynomial function, which passes through known points of a two-dimensional plane. For example, in a x-y plane, given $n + 1$ are known points, (x_0, y_0), (x_1, y_1), ... , (x_n, y_n). The method of Lagrange interpolation provides a formula for constructing a unique polynomial of degree n which passes through these $n + 1$ points. Among them, the Lagrange fundamental polynomial, or interpolation basis function is expressed as follows:

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

The specific point of $l_j(x)$ is the derived value 1 from x_j. Values from other points x_i $(i \neq j)$ equals 0, the expression of which is as follows: $l_j(x) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$. The Lagrange polynomial is $L(x) = \sum_{j=0}^{n} y_j l_j(x)$.

That is the unique polynomial of degree n which passes through the points (x_0, y_0), (x_1, y_1), ... , (x_n, y_n). For example, the binomial that passes through (4, 1), (5, 5), and (6, 10) when expressed in Lagrange basic polynomial is as follows

$$l_1(x) = \left(\frac{x-5}{4-5}\right)\left(\frac{x-6}{4-6}\right),\ l_2(x) = \left(\frac{x-4}{5-4}\right)\left(\frac{x-6}{5-6}\right),\ l_3(x) = \left(\frac{x-4}{6-4}\right)\left(\frac{x-5}{6-5}\right).$$

By applying Lagrange interpolation polynomial, a single polynomial $L(x)$ can be obtained as expressed below:

$$\begin{aligned} L(x) &= f(4)l(1) + f(5)l(2) + f(6)l(3) \\ &= 1 \times \left(\tfrac{x-5}{4-5}\right)\left(\tfrac{x-6}{4-6}\right) + 5 \times \left(\tfrac{x-4}{5-4}\right)\left(\tfrac{x-6}{5-6}\right) + 10 \times \left(\tfrac{x-4}{6-4}\right)\left(\tfrac{x-5}{6-5}\right) \\ &= \tfrac{1}{2}x^2 - \tfrac{1}{2}x - 5 \end{aligned}$$

It can be inferred that $f(4) = 1, f(5) = 5, f(6) = 10$. By applying this formula, predicted values can be derived, for example: to derive $f(18)$, substitute $x = 18$ in $L(x)$, and $L(18) = f(18) = 148$ is derived.

3. The Proposed Mechanism

The foremost challenges when creating digital-sharing systems for cloud environments are managing large user bases and complex access relationships. Ensuring the confidentiality and integrity of users' cloud data represents an additional concern. In response to these challenges, we developed a dynamic multi-user access mechanism that can accurately access and control the digital resources and teaching materials stored in the cloud, as shown in Figure 1. The proposed systems indexes Lagrange interpolating polynomials to provide different users maximum control over their data and logs. The system also comprises an encryption technique to protect users' privacy, with unique keys

generated by Central Authority that can be freely used to share their digital data. The steps involved in developing the proposed access control system are explained in the following section. The definitions of the symbols used in the creation of the proposed system are tabulated in Table 1.

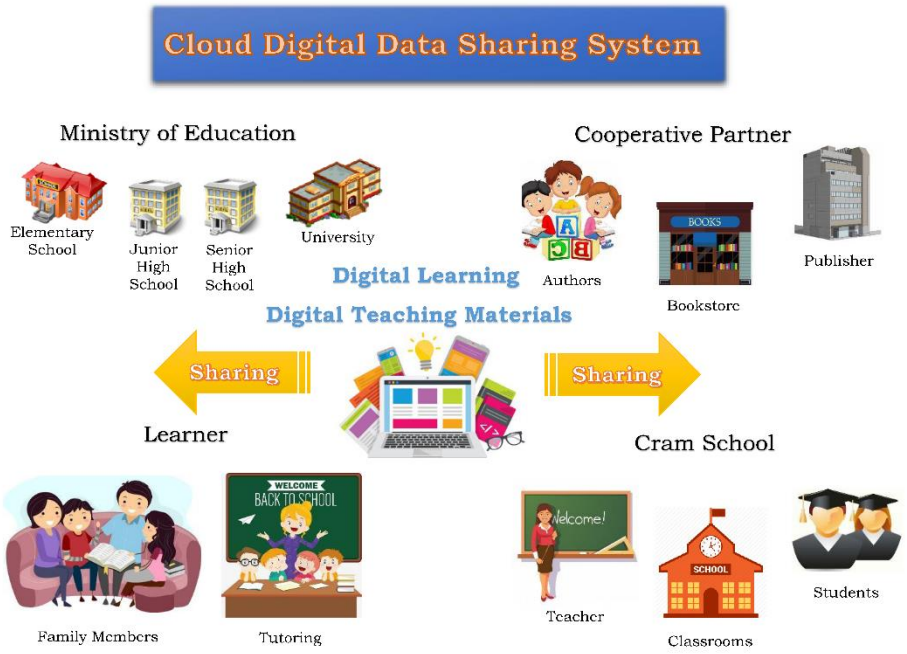

Figure 1. Access diagram for the cloud-based digital sharing system.

Table 1. Symbol definitions and parameters.

Symbol	Definition	Function
S_i	Security class, S_i = {u: u is the file id that S_i is authorized to access}, for i = 1, 2, ..., n	To distinguish the user's security class
H_i	Superkey H_i, for i = 1, 2, ..., n	To obtain the key for accessing $file_u$
DK_u	Decryption key, for u = 1, 2, ..., m	To decrypt the $file_u$ key
$file_u$	$File_u$, for u = 1, 2, ..., m	Represents the file encrypted with DK_u
$I_{\{H_1,...,H_n\}}(x)$	The indicate function of set {H_1, H_2, ..., H_n}	To determine whether H_i is present in the authentication list approved by the central authority (CA)
J_i	J_i = {u: 1 ≤ u ≤ m, u is the file id that S_i is authorized to access}	User-authorized file set
$I_{J_i}(x)$	The indicate function of set J_i	To determine whether the user has authorized the file set

3.1. Create a System User

In this study, we adopt the access relationships of a partially ordered set. A Central Authority (CA) (or a multiple number of CAs that are distributed by a single CA) builds the partially ordered set, which is a pair (S, ⪯), where ⪯ is a reflexive, antisymmetric, transitive binary relation in set S. In this paper, users are divided into disjoint sets labeled S_i, where S_i is a subset that corresponds with security classes, and each class is assigned clearance to access specific files. Therefore, the decryption key with permission to obtain the encrypted file can be expressed as S_i = {u: u is the file id that S_i is permitted to access} for i = 1, 2, ... , n, where $n \in N$ and '⪯' is a binary partial order relation over the set S = {S_1, S_2, ... , S_n}. For the set (S, ⪯), $S_j \preceq S_i$ (i, $j \in N$), indicating that a user in security class S_i can read or store data held by a user in security class S_j, but the opposite is not allowed. Each class possesses its

own cryptographic key; thus, if $S_j = \{1, 2\}$, $S_i = \{1, 2, 3\}$, $\{1, 2\} \preceq \{1, 2, 3\}$, then $S_j \preceq S_i$. In $S_j \preceq S_i$, S_i corresponds with the security class required to obtain the decryption key for S_j to retrieve $file_1$ and $file_2$.

The system may be accessed by users of a variety of identities, such as teaching material authors, partner vendors, authorities affiliated with the Ministry of Education, class teachers, students, and students' parents. In the proposed system, the security class of each user is expressed as S_i, and each user possesses a superkey (H_i), where $i = 1, 2, ..., n$. The CA creates a framework for these users. The system structure comprises n users in two sets, $S = \{S_1, S_2, ..., S_n\}$ and $H = \{H_1, H_2, ..., H_n\}$, which can be expressed in Table 2:

Table 2. Sets of security class and superkey.

S_1	S_2	...	S_i	...	S_n	
H_1	H_2	...	H_i	...	H_n	←secret and distinct

3.2. Establish an Associative Array and Function for System Users and Data Files

The digital-data-sharing system proposed in this study was developed specifically for teaching-material-related use. The proposed system stores data provided by publishers (partner vendors), the Ministry of Education, teaching material authors, and teachers. The system applies encryption keys to the data uploaded by the various users to generate encrypted files, which are then stored in the cloud server. The CA builds a structure in which m files form a set $file = \{file_1, file_2, ..., file_m\}$. Additionally, the CA creates decryption keys corresponding to $file_u$, where $u = 1, 2, ..., m$, protecting the encrypted files from random access. The decryption keys are expressed as DK_u, where $u = 1, 2, ..., m$. The relationship between the files and keys can be shown in Table 3.

Table 3. Decryption keys for corresponding encrypted files.

$file_1$	$file_2$	...	$file_u$	...	$file_m$	
1	2	...	u	...	m	file id, public
DK_1	DK_2	...	DK_u	...	DK_m	decryption keys, secret and distinct

The following adjacency matrix illustrates the access relationships. Assume the system structure comprises six security classes and four files; {security classes} × {files} data may be arranged in a two-dimensional array as follows:

$$\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{array} \begin{array}{c} \begin{array}{cccc} file_1 & file_2 & file_3 & file_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

We define the indicate function as $I(x, y)$. This function expresses that user i is permitted to obtain $file_u$ using DK_u. Variable x represents user's id i, and the variable y represents *file*'s id u. User i uses his or her secret superkey H_i to access row i. According to the construction, row i contains the set of $file_u$ that user i is authorized to access. For example, $I(3, 2) = 1$ because user 3 is authorized to access $file_2$. $I(6, 1) = 0$ because user 6 is not authorized to access $file_1$.

$$I(x, y) = \begin{cases} 1 & \text{, if user } x \text{ has access to file } y \\ 0 & \text{, otherwise} \end{cases} \quad (2)$$

For a flexible specification of access control policy, we combine the dynamically changing context by using the particular context database. It is mainly formed with 3w queries (who, what, and where), i.e., the sets of questions for judgement the specific people (1) whether he/she comes from the security class $\{S\}$; (2) what conditions that he/she needs to handle (specific information and resources to be obtained); and (3) what locations that he/she may exist (the determination of environments). Figure 2 shows the detailed structure of the database. Here, we require that before people, who have authority (i.e., $I(x, y) = 1$), access the file, he/she must firstly pass the queries from the context database.

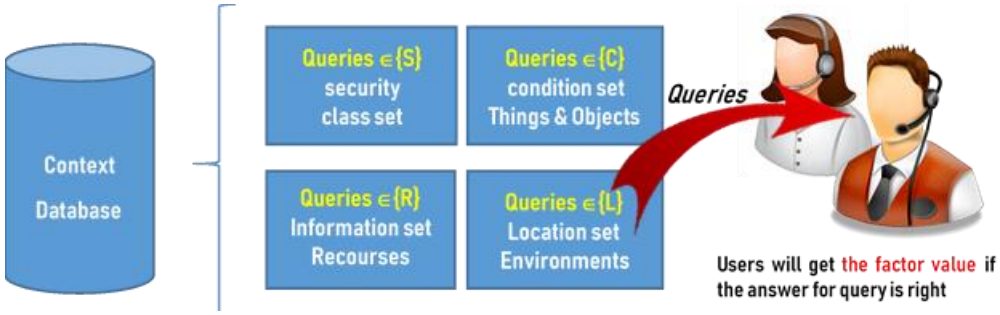

Figure 2. Context database and query set for the proposed digital sharing system.

Each query may equally come from different sets and the requested users need to send the right answers for the response queries, making the sum of value of identification factors be greater than, or equal to, one, i.e., $\{factor_{Q_n} \mid 0 \le factor_{Q_n} \le 1, \text{ for } i = 1, 2, \ldots, n\}$, where $factor_{Q_1} + factor_{Q_2} + \ldots + factor_{Q_n} \ge 1$. The system can determine that the user has indeed been authorized and owned the right to get secure key and access the file contents by the sum of the factor value reaching to one.

3.3. Establish the Correlation Functions to Derive Keys of System Users

To accurately derive *DK*s to access the desired file, numerous auxiliary polynomials and functions are stored in the system to assist in the processing of access control. First, the authenticity of users' keys must be defined. Therefore, the indicator function is defined as $I_{\{H_1,\ldots,H_n\}}(x) = \begin{cases} 1 & ,if \quad x \in \{H_1, \ldots, H_n\} \\ 0 & , o.w. \end{cases}$, where an output result of 1 represents an authentic key. For other outputs, the key is rejected.

Second, the clearance of the user must be determined for file access to be granted. Therefore, the function is defined as $J_i = \{u: 1 \le u \le m, u \text{ is the file id that } S_i \text{ has permission to access}\}$ (assuming that the system contains n users and m files for access). The preceding two auxiliary functions form the function $I_{J_i}(y)$, which expresses that the user's S_i is authorized to access the *DK*. The function can be expressed as follows:

$$I_{J_i}(y) = \begin{cases} 1 & ,if \quad y \in J_i \\ 0 & ,o.w. \end{cases} \tag{3}$$

Third, several correlation functions can be generated by applying the Lagrange interpolation polynomial as below steps:

Step 1: the CA establishes a unique superkey H_i, where $i = 1, 2, \ldots, n$, for user i in the $S = \{S_1, S_2, \ldots, S_n\}$ set. The H_i is confidential to the user i.

Step 2: the CA manages the H_i of all users and establishes an indicator function to authenticate the superkey: $I_{\{H_1,\ldots,H_n\}}(x) = \begin{cases} 1 & ,if \quad x \in \{H_1, \ldots, H_n\} \\ 0 & ,o.w. \end{cases}$. $I_{\{H_1, \ \ldots, H_n\}}(x)$ indicates that the indicator function of set $H = \{H_1, H_2, \ldots, H_n\}$. $I_{\{H_1,\ldots,H_n\}}(x)$ is used to verify the authenticity of H_i.

Step 3: the CA establishes the function $A_i(x)$ applying Lagrange interpolation polynomial for user i, where $A_i(x) = \left\{ \prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{(x-H_k)}{(H_i-H_k)} \right\} \times I_{\{H_1,\ldots,H_n\}}(x)$, for $i = 1, 2, \ldots, n$, $x \in R$.

Step 4: the CA selects nonrepeated random integers $\{DK_1, DK_2, \ldots, DK_m\}$ (supposing m confidential files exist) as the decryption key for encrypting and decrypting confidential files. The CA maintains the confidentiality of the DK_u and publishes the public parameter u.

Step 5: the CA defines $J_i = \{u: 1 \leq u \leq m, u$ is the file id that S_i has permission to access$\}$ when n users exist for $i = 1, 2, ..., n$ and m files for $u = 1, 2, ..., m$. J_i is the set of *file* id's that user i is authorized to visit.

Step 6: the CA defines $I_{J_i}(y) = \begin{cases} 1 & ,if \quad y \in J_i \\ 0 & ,o.w. \end{cases}$. This indication function expresses that user i is authorized to access the DK_u. The function $B_i(y)$ is established by applying Lagrange interpolation polynomial for each user i. Let $B_i(y) = \left\{ \sum_{u \in J_i} DK_u \left[\prod_{\substack{t=1 \\ t \neq u}}^{m} \frac{(y-t)}{(u-t)} \right] \right\} \times I_{J_i}(y)$, $y \in R$.

Step 7: the CA establishes the key-deriving function $G(x, y) = \sum_{i=1}^{n} A_i(x)B_i(y)$, $x \in R$, $y \in R$. That is, $G(x, y) = A_1(x)B_1(y) + A_2(x)B_2(y) + \ldots + A_n(x)B_n(y)$ for $(x, y) \in R \times R$, and the CA declares it publicly.

Continually, user i can incorporate the owned superkey H_i and the file id u to $G(x, y)$ for deriving the DK_u, which is then used to decrypt $file_u$. The derivation process is described in the following steps:

Step 1: user i incorporates the superkey H_i into $I_{\{H_1,\ldots,H_n\}}(x) = \begin{cases} 1 & ,if \quad x \in \{H_1, ..., H_n\} \\ 0 & ,o.w. \end{cases}$. If user i's superkey H_i is present in the authentication list established by the CA, then $H_i \in \{H_1, ..., H_n\}$ and $I_{\{H_1,\ldots,H_n\}}(H_i) = 1$. If user i's superkey H_i is not present in the authentication list, then $I_{\{H_1,\ldots,H_n\}}(H_i) = 0$.

Step 2: user i incorporates the superkey H_i into $A_i(x) = \left\{ \prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{(x-H_k)}{(H_i-H_k)} \right\} \times I_{\{H_1,\ldots,H_n\}}(x)$. If user i uses his or her superkey H_i and the superkey H_i is present in the authentication list established by the CA, then $I_{\{H_1,\ldots,H_n\}}(x) = 1$ can be incorporated for calculation. In this instance, $A_i(H_i) = 1$ and $A_i(H_k) = 0$ for $k \neq i$.

Step 3: user i incorporates the id u of the desired $file_u$ into $I_{J_i}(y) = \begin{cases} 1 & ,if \quad y \in J_i \\ 0 & ,o.w. \end{cases}$, where $J_i = \{u:$ $1 \leq u \leq m$; u is the file id that S_i has permission to access $\}$. If user i is authorized to access DK_u, then $y \in J_i$ and $I_{J_i}(y) = 1$.

Step 4: user i incorporates the id u of the desired $file_u$ into $B_i(y) = \left\{ \sum_{u \in J_i} DK_u \left[\prod_{\substack{t=1 \\ t \neq u}}^{m} \frac{(y-t)}{(u-t)} \right] \right\} \times I_{J_i}(y)$. If user i is authorized to access DK_u, then $B_i(y) = DK_y$ if $y \in J_i$ or $B_i(y) = 0$ if $y \notin J_i$.

Step 5: user i calculates $G(x, y) = \sum_{i=1}^{n} A_i(x)B_i(y)$. If $x \in \{H_1, H_2, \ldots, H_n\}$ and $y \in J_{x.}$, then $G(x, y) = DK_y$. In this instance, the user can derive the decryption key; otherwise, $G(x, y) = 0$.

3.4. Change the Access Permissions of Users

The system users' membership system may be added or removed due to different events, or as time changes. Additionally, users' access permissions may change, and data may be added, modified, or deleted according to different access requirement. In this study, we developed an approach to resolve management problems related to system access security without sacrificing computing power and storage space.

The proposed system calculates the public function $G(x, y)$. The following goals may be achieved by updating the function and modifying the parameters: (1) add user; (2) remove user, and (3) update user permissions.

$$G(x, y) = \sum_{i=1}^{n} A_i(x)B_i(y), x \in R, y \in R \tag{4}$$

Further decomposition of $G(x, y)$ yields $A_1(x)B_1(y) + A_2(x)B_2(y) + \ldots + A_n(x)B_n(y)$ where $(x, y) \in R \times R$ and the subfunction $A_i(x)$ is correlated to the authentication of user data. The subfunction verifies whether H_i is present in a legitimate list in the system and whether the user can acquire a personal key for authentication. Additionally, subfunction $B_i(y)$ is correlated to data authentication. These subfunctions verify whether users can acquire DK_u to decrypt encrypted data files. $A_i(x)$ and $B_i(y)$ can be expressed as follows:

$$A_i(x) = \left\{ \prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{(x - H_k)}{(H_i - H_k)} \right\} \times I_{\{H_1, \ldots, H_n\}}(x), \text{ for } i = 1, 2, \ldots, n, x \in R \tag{5}$$

$$B_i(y) = \left\{ \sum_{u \in J_i} DK_u \left[\prod_{\substack{t=1 \\ t \neq u}}^{m} \frac{(y - t)}{(u - t)} \right] \right\} \times I_{J_i}(y), y \in R \tag{6}$$

(1) Add user: to add a user, the system merely update the indication functions $I_{\{H_1, \ldots, H_{n+1}\}}(x)$ and $I_{J_v}(y)$ and creates $A_v(x)$, $B_v(y)$, and J_v for the new user S_v, after which the data are updated to $G(x, y)$. In other words, $G'(x, y) = G(x, y) + A_v(x)B_v(y)$. Only simple additive operation is involved in the computation.

(2) Remove user: similar to the process for adding a user, the system removes the $A_v(x)$ and $B_v(y)$ parameters associated with member S_v from $G(x, y)$ to remove a user. Therefore, $G'(x, y) = G(x, y) - A_v(x)B_v(y)$. A subtraction algorithm is used in the computation.

(3) Update user access permissions: when a system user wishes to modify their access permissions, the system redefines $J_i{}' = \{u: 1 \leq u \leq m, u$ is the file id that S_i has permission to access $\}$, where J_i' represents the updated S_i permissions. $B_i(y)$ is then updated to $B_i'(y)$; that is, the function J_i related to $B_i(y)$ is replaced by J_i' to obtain $G'(x, y) = G(x, y) - A_i(x)B_i(y) + A_i(x)B_i'(y)$, reflecting the new permissions for the user. Addition and subtraction algorithms are used in the computation.

3.4.1. Adding a New Security Class

In case that S_v is a new security to be inserted into the user hierarchy; CA executes the procedure below for inserting the new security class S_v.

Step 1: CA distributes the secret parameter superkey H_v to a new security class S_v.

Step 2: CA establishes $A_v(x)$. $A_v(x)$ is identical to that of $A_i(x)$ except that n is replaced by $n + 1$, $A_v(x) = \prod_{\substack{v=1 \\ v \neq k}}^{n+1} \frac{x-H_k}{H_v-H_k}$. The index $I_{\{H_1,\ldots,H_{n+1}\}}(x) = \begin{cases} 1 & ,if \quad x \in \{H_1,\ldots,H_{n+1}\} \\ 0 & ,o.w. \end{cases}$ is updated.

Step 3: CA establishes the parameter $J_i = \{u\colon 1 \leq u \leq m, u$ is the file ID of authorized $S_i\}$ for S_v

Step 4: CA establishes $B_v(y)$, where $B_v(y) = \left\{ \sum_{u \in J_v} DK_u \left[\prod_{\substack{t=1 \\ t \neq u}}^{m} \frac{(y-t)}{(u-t)} \right] \right\} \times I_{J_v}(y)$.

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

Step 6: CA updates formula $G(x, y)$ in the original scheme that the new formula appears $G'(x, y) = G(x, y) + A_v(x)B_v(y)$

In the above process to append a user, CA simply updates the indices $I_{\{H_1,\ldots,H_{n+1}\}}(x)$ and $I_{J_v}(y)$ and establishes $A_v(x)$, $B_v(y)$, J_v for the new security class S_v. The information is updated to formula $G(x, y)$. Few costs are required for computing the new security class S_v, and merely addition is required for updating the entire scheme.

3.4.2. The Example of Adding a New Security Class

In this example, we assume that the digital-sharing mechanism contained the security classes S_1 through S_6 and the digital resources file$_1$ to file$_5$. A downstream bookstore joins the sharing mechanism as S_7 and the owner receives authorization to access Junior High School Year 1 English, Senior High School Year 2 Physics, and Senior High School Year 3 Chemistry (Table 4).

First, the CA assigns the superkey H_7 to the downstream bookstore and updates the indication functions to $I_{\{H_1,\ldots,H_{n+1}\}}(x)$ and $I_{J_v}(y)$ according to the digital-resource permissions of the bookstore. The CA defines $J_7 = \{1, 3, 4\}$ for S_7 and creates the following equation:

$$A_7(x) = \left\{ \frac{(x-H_1)(x-H_2)(x-H_3)(x-H_4)(x-H_5)(x-H_6)}{(H_7-H_1)(H_7-H_2)(H_7-H_3)(H_7-H_4)(H_7-H_5)(H_7-H_6)} \right\} \times I_{\{H_1,\ldots,H_7\}}(x)$$

$$B_7(y) = \left\{ DK_1 \times \frac{(y-2)(y-3)(y-4)(y-5)}{(1-2)(1-3)(1-4)(1-5)} + DK_3 \times \frac{(y-1)(y-2)(y-4)(y-5)}{(3-1)(3-2)(3-4)(3-5)} + DK_4 \times \frac{(y-1)(y-2)(y-3)(y-5)}{(4-1)(4-2)(4-3)(4-5)} \right\} \times I_{J_7}(y)$$

Finally, all parameters are updated into a new formula $G'(x, y) = G(x, y) + A_7(x)B_7(y)$

Table 4. The results after adding a new security class.

	$file_1(DK_1)$ Junior High School Year 1 English	$file_2(DK_2)$ Junior High School Year 2 Math.	$file_3(DK_3)$ Senior High School Year 2 Physics	$file_4(DK_4)$ Senior High School Year 3 Chemistry	$file_5(DK_5)$ University Year 1 Chinese
$S_1(H_1)$: Teaching material author	1	1	1	1	1
$S_2(H_2)$: Publisher	1	1	1	1	0
$S_3(H_3)$: Teacher	1	0	0	1	0
$S_4(H_4)$: Student	0	0	0	1	0
$S_5(H_5)$: Cram school operator	0	0	0	0	1
$S_6(H_6)$: Student parents	1	0	0	0	0
$S_7(H_7)$: Downstream bookstore	1	0	1	1	0

3.4.3. Removing an Existing Security Class

Assuming that an existing security class S_v is to be removed from the digital-sharing mechanism, CA could precede the following Step 1 or Step 2.

Step 1: CA removes the relevant parameter $A_v(x)B_v(y)$ in the security class S_v from $G(x, y)$. $G'(x, y) = G(x, y) - A_v(x)B_v(y)$

Step 2: J_v is defined as the set of *file* ID's, which the user v is authorized to visit. Instinctively, CA updates J_v and deletes the authorization of the user: $J_v{}' = \phi$ = empty set

3.4.4. The Example of Removing an Existing Security Class

Assuming that S_7 downstream bookstore in the original scheme is no longer authorized, CA tends to remove S_7 from the scheme, as below (Table 5):

CA could choose one of the following methods to remove S_7; one is to update formula $G'(x, y) = G(x, y) - A_7(x)B_7(y)$ to remove the relevant parameters in S_7 and the other is to update $J_7{}' = \phi$ so that S_7 could not pass the authorization verification.

3.4.5. Updating a User Authorized

In the initial phase of the proposed scheme, CA would establish the access authority for the security class S_i. When a user is updated by the system authorization, CA would proceed the following steps.

Step 1: CA resets $J_i{}' = \{u: 1 \le u \le m, u$ is the file ID of authorized $S_i\}$. $J_i{}'$ presents the new authorization of S_i after update. When the authorization to the digital-sharing mechanism is changed, CA would re-calculate the adjacency matrix to generate a new set J_i.

Step 2: CA updates $B_i(y)$ to $B_i{}'(y)$, as J_i is replaced by $J_i{}'$ and the information of J_i is relevant with $B_i(y)$. Assuming that a new authorization of set $J_i{}'$ is given to user i, then $G'(x, y) = G(x, y) - A_i(x)B_i(y) + A_i(x)B_i{}'(y)$.

According to the above steps, the establishment of J_i could easily update the authorization of user i to access to digital data. When the user i does not present any authorization, $B_i(y)$ does not need to be updated, but just take $J_i{}' = \phi$ = empty set.

3.4.6. The Example of Updating a User Authorized

Assuming that S_4 student could access to $file_4$ Chemistry in the original scheme, but no longer could after the research project being changed, a new authorization allows to access to $file_2$ mathematics, as below (Table 6):

CA updates $J_4 = \{4\}$ to $J_4{}' = \{2\}$ and updates $B_4{}'(y)$. Then $G'(x, y) = G(x, y) - A_4(x)B_4(y) + A_4(x)B_4{}'(y)$

$$B_4{}'(y) = \left\{ DK_2 \times \frac{(y-1)(y-3)(y-4)(y-5)}{(2-1)(2-3)(2-4)(2-5)} \right\} \times I_{J_4}(y)$$

In this dynamic access section, the construction and updating of $G(x, y)$ involve only simple arithmetic calculations. These can be done on a fly for a system consisting of millions of servers and millions of files. This scheme is easy to operate as the user i just enters a pair of valid (H_i, u) to get the correct DK_u. The system administrator calculates and updates $G(x, y)$ in the background in real time. Every server follows exactly the same operational steps to retrieve the correct decryption key.

Table 5. The resulting after revoking the existing current security class.

	$file_1(DK_1)$ Junior High School Year 1 English	$file_2(DK_2)$ Junior High School Year 2 Math.	$file_3(DK_3)$ Senior High School Year 2 Physics	$file_4(DK_4)$ Senior High School Year 3 Chemistry	$file_5(DK_5)$ University Year 1 Chinese
$S_1(H_1)$: Teaching material author	1	1	1	1	1
$S_2(H_2)$: Publisher	1	1	1	1	0
$S_3(H_3)$: Teacher	1	0	0	1	0
$S_4(H_4)$: Student	0	0	0	1	0
$S_5(H_5)$: Cram school operator	0	0	0	0	1
$S_6(H_6)$: Student parents	1	0	0	0	0

Table 6. The resulting after updating of a user authorized.

	$file_1(DK_1)$ Junior High School Year 1 English	$file_2(DK_2)$ Junior High School Year 2 Math.	$file_3(DK_3)$ Senior High School Year 2 Physics	$file_4(DK_4)$ Senior High School Year 3 Chemistry	$file_5(DK_5)$ University Year 1 Chinese
$S_1(H_1)$: Teaching material author	1	1	1	1	1
$S_2(H_2)$: Publisher	1	1	1	1	0
$S_3(H_3)$: Teacher	1	0	0	1	0
$S_4(H_4)$: Student	0	1	0	0	0
$S_5(H_5)$: Cram school operator	0	0	0	0	1
$S_6(H_6)$: Student parents	1	0	0	0	0
$S_7(H_7)$: Downstream bookstore	1	1	1	1	1

4. Analysis of Security

In this section, a security analysis is performed to examine whether the proposed scheme is secure in practical applications. The analysis focuses on four types of attack that may affect the system's security.

4.1. Equation Attack

Equation attack: attackers attempt to use a public formula $G(\cdot)$ to crack polynomials using mathematical algorithms and obtain the DK_u.

Equation attacks occur during updates of users' permissions. When one user is being removed while the other users remain unchanged, attackers can subtract old public data $G(\cdot)$ with new public data $G'(\cdot)$, or $G'(\cdot) - G(\cdot) = 0$, to derive DK_u. The mechanism designed in this study can withstand equation attacks. In Section 3, we propose the following three dynamic update methods:

1. Addition of a new security class: $G'(x, y) = G(x, y) + A_v(x)B_v(y)$
2. Deletion of a security class: $G'(x, y) = G(x, y) - A_v(x)B_v(y)$
3. Update of a user's authorization $G'(x, y) = G(x, y) - A_i(x)B_i(y) + A_i(x)B_i'(y)$

In all three dynamic update methods, the public parameters $G(x, y)$ from before the update are subtracted from the updated public parameters $G'(x, y)$. Therefore, attackers can only derive $A_v(x)B_v(y)$ 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 polynomials created through Lagrange interpolation. Therefore, a multiplication algorithm must be applied to convert $A_v(x)$ and $B_v(y)$ into an $(n-1)(m-1)^{\text{th}}$ order polynomial with two unknowns.

$$A_v(x) = \left\{ \prod_{\substack{u=1 \\ u \neq v}}^{n} \frac{(x - H_u)}{(H_i - H_u)} \right\} \times I_{\{H_1,\ldots,H_n\}}(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} \tag{7}$$

$$B_v(y) = \left\{ \sum_{j \in J_v} DK_u \left[\prod_{\substack{t=1 \\ t \neq u}}^{m} \frac{(y - t)}{u - t} \right] \right\} \times I_{J_v}(y) = b_0 + b_1 y + \ldots + b_{m-1}y^{m-1} \tag{8}$$

$$A_v(x)B_v(y) = a_0b_0 + a_1b_0x + a_0b_1y + a_1b_1xy \ldots + a_{n-1}b_{m-1}x^{n-1}y^{m-1} \tag{9}$$

If the attacker incorporates $x = 0$ or $y = 0$ into the deduction, the returned polynomial messages of $A_v(x)B_v(y)$ would comprise a string of unstructured data. Therefore, our methods are not vulnerable to compromising attacks.

4.2. External Attack

External attack: external users attempt to use public parameters to gain access. They attempt to obtain DK_u or decrypt documents to acquire a digital resource stored in the cloud.

Digital teaching materials, data, and data sources acquired from the cloud can be sold at a low price, which not only infringes upon the authors' intellectual property rights, but also causes immense losses for publishers. For an unauthorized external user to access digital resources in the proposed digital sharing mechanism, the user must use public parameters to derive the decryption key and decrypt the files to acquire meaningful data.

The most critical known public parameter for external attackers is the public function $G(x, y)$, because this function contains the DK_u. Therefore, the equations based on this function must be

protected. In the proposed method, each security class S_i can be incorporated into private superkeys H_i using the public function $G(x, y)$ to derive the DK_u. If an external attacker attempts to obtain the DK_u, he or she must decrypt the Lagrange interpolating polynomials to obtain a secret key. For external attackers that can only obtain the public function $G(x, y)$ and file id u, the large number of unknown variables hinders them from reverse deriving the DK_u through mathematical computation. Therefore, attackers cannot unlawfully acquire digital teaching materials, data, or data sources through external attacks on the proposed system.

In the proposed method, the CA can choose any encryption method to generate the DK_u. For example, symmetrical key systems such as DES, Triple DES, or AES use diffusion and confusion to block hackers from cracking encryptions through statistical calculations. At present, these password systems remain challenging to crack. Therefore, attackers cannot directly extract meaningful content from encrypted documents in the proposed system.

4.3. Collaborative Attack

Collaborative attack: two or more authorized users collaborate and share superkeys H_i with each other in an attempt to derive a DK_j outside their authorization or other users' superkeys H_i.

The security class S_i adopted in the proposed method involves partially ordered relationships. When S_i is authorized to access S_j, the following formula $G(x, y)$ can be used:

$$G(x, y) = A_1(x)B_1(y) + A_2(x)B_2(y) + \ldots + A_n(x)B_n(y) \tag{10}$$

Therefore, we define a collaborative attack as a situation in which two or more authorized users target another authorized user. Two example scenarios are presented subsequently. In the first scenario, the collaborating attackers are in a partially ordered relationship with their target. In the second scenario, the collaborating attackers are not in a partially ordered relationship with their target.

Scenario 1: the collaborative attackers attempt to collect each other's privacy parameters superkey H_i and obtain the DK_u of another user in the system that the attackers do not have permission to access. Based on the above Section 3.4.2, the clearance of the collaborating attackers are $S_3 = \{1, 4\}$ and $S_4 = \{4\}$ and that of the target is $S_7 = \{1, 3, 4\}$. In contrast to S_3 and S_4, S_7 is authorized to access $file_3$. Therefore, in this example, S_3 and S_4 collaboratively launch an attack to acquire S_7 and DK_3. The data related to DK_3 is hidden in $A_7(x)B_7(y)$.

$$A_7(x) = \left\{ \frac{(x-H_1)(x-H_2)(x-H_3)(x-H_4)(x-H_5)(x-H_6)}{(H_7-H_1)(H_7-H_2)(H_7-H_3)(H_7-H_4)(H_7-H_5)(H_7-H_6)} \right\} \times I_{\{H_1,\ldots,H7\}}(x)$$

$$B_7(y) = \left\{ DK_1 \times \frac{(y-2)(y-3)(y-4)(y-5)}{(1-2)(1-3)(1-4)(1-5)} + DK_3 \times \frac{(y-1)(y-2)(y-4)(y-5)}{(3-1)(3-2)(3-4)(3-5)} + DK_4 \times \frac{(y-1)(y-2)(y-3)(y-5)}{(4-1)(4-2)(4-3)(4-5)} \right\} \times I_{J_7}(y)$$

However, S_3 and S_4 only possess H_3 and H_4, and these superkeys are unable to pass the $A_7(x)$ test. Lagrange interpolation calculations using these superkeys yield empty values. Therefore, $A_7(x)B_7(y) = 0 \times B_7(y) = 0$. In this instance, a collaborative attack is similar to an independent attack, and the attackers are not able to obtain additional data.

Scenario 2: the collaborating attackers are not in a partially ordered relationship with their target. The attackers collect each other's parameters to increase the probability of deriving DK_u. According to Section 4.1, the clearances of the collaborating attackers are $S_3 = \{1, 4\}$ and $S_4 = \{4\}$, and that of the target is $S_5 = \{5\}$. As described, no partially ordered relationship exists between S_5 and S_3 or S_4. To obtain S_5 and gain access to $file_5$, S_3 and S_4 must collaboratively launch an attack on S_5 to obtain DK_5. As in Scenario 1, S_3 and S_4 only possess the superkeys H_3 and H_4, which cannot be used to pass the $A_5(x)$ test, and calculations only yield empty values.

Thus, private superkeys H_i cannot be collected to derive a DK_u without authorization, regardless of the number of attackers or whether a partially ordered relationship exists between the attacker(s) and the target.

In addition to DK_u, attackers may also target superkeys H_i. In Scenario 1, the $A_7(x)$ results indicate that S_3 and S_4 only possess H_3 and H_4. Thus, these users lack sufficient data to obtain H_7 from the $A_7(x)$ results produced through Lagrange interpolation. Therefore, collaborative attacks are ineffective against the proposed system.

4.4. Reverse Attack

Reverse attack: an authorized user (attacker) uses a known public formula $G(x, y)$ and his or her private parameters to obtain the superkeys $H_i^{'}$ of other users.

Based on Section 3.4.2, users with S_6 and S_7 are generally able to derive DK_1 by applying $G(x, y)$. S_6 is in a partially ordered relationship with S_7. Specifically, $S_6 \preceq S_7$, where $S_6 = \{1\}$ and $S_7 = \{1, 3, 4\}$. In this scenario, the user that corresponds with S_6 is the attacker that attempts to use H_6 and $G(x, y)$ to derive the H_7 of S_7 and then use S_7 to obtain $file_3$ and $file_4$.

The proposed method only involves one public formula: $G(x, y) = A_1(x)B_1(y) + \ldots + A_6(x)B_6(y) + A_7(x)B_7(y)$. To use S_6 in the deduction process, point $(H_6, 1)$ is incorporated into the aforementioned polynomials for calculation. Subsequently, S_7 can be used to incorporate points $(H_7, 1)$, $(H_7, 3)$, and $(H_7, 4)$ into the calculations and thereby derive the key allocated by the CA. However, DK_3 and DK_4 of $file_3$ and $file_4$ cannot be obtained through incorporating S_6 to point $(H_6, 3)$ or point $(H_6, 4)$.

The user corresponding with S_6 attempts to acquire the DK_3 and DK_4 of S_7. Therefore, the target is H_7 or DK_3 and DK_4 related to $A_7(x)B_7(y)_7$. Because S_6 can be used to incorporate point $(H_6, 1)$ into $G(H_6, 1) = DK_1$, the user corresponding with S_6 may attempt the following calculations:

$G(H_6, 1) - DK_1 = 0$

$\Rightarrow A_1(H_6)B_1(1) + \ldots + A_6(H_6)B_6(1) + A_7(H_6)B_7(1) + \ldots + A_n(H_6)B_n(1) - DK_1 = 0$

$\Rightarrow c_0d_0 + c_1d_0x + c_0d_1y + c_1d_1xy + \ldots + c_{n-1}d_{m-1}x^{n-1}y^{m-1} - DK_1 = 0$

The equation demonstrates that $G(x, y)$ is an $(n-1)(m-1)^{\text{th}}$ order polynomial with two unknowns. The attacker cannot decipher the data of $A_7(x)B_7(y)$ from the polynomials. $G(x, y)$ is extremely simple and does not contain numerous parameters that attackers can manipulate. Even if the attacker gains a portion of $A_7(x)B_7(y)$, $A_7(x)$ and $B_7(y)$ are still protected by separate mechanisms.

H_7 data are hidden in $A_7(x)$, which is generated through Lagrange interpolation, expressed as follows:

$A_7(x) = \left\{ \frac{(x-H_1)(x-H_2)(x-H_3)(x-H_4)(x-H_5)(x-H_6)}{(H_7-H_1)(H_7-H_2)(H_7-H_3)(H_7-H_4)(H_7-H_5)(H_7-H_6)} \right\} \times I_{\{H_1,\ldots,H7\}}(x)$, A_7(x) verifies whether the H_i inputted by the user is present in the verification list approved by the CA. If the user is not approved by the CA, then the H_i is rejected from $I_{\{H_1,\ldots,H_n\}}(x)$. If the user uses a superkey other than H_7, Lagrange interpolation calculation yields a value of 0.

DK_3 and DK_4 data are hidden in B_7(y), which is generated through Lagrange interpolation.

$$B_7(y) = \left\{ DK_1 \times \frac{(y-2)(y-3)(y-4)(y-5)}{(1-2)(1-3)(1-4)(1-5)} + DK_3 \times \frac{(y-1)(y-2)(y-4)(y-5)}{(3-1)(3-2)(3-4)(3-5)} + DK_4 \times \frac{(y-1)(y-2)(y-3)(y-5)}{(4-1)(4-2)(4-3)(4-5)} \right\} \times I_{J_7}(y)$$

Users' file access clearance must be approved by the CA to pass authentication of $I_{J_i}(y)$, where $J_i = \{j: 1 \le j \le m\}$. Otherwise, the function yields an empty value.

Resources outside of individuals' authorization cannot be retrieved by reversing polynomials. In sum, the proposed method blocks equation attacks.

4.5. Proof of Lagrange Interpolation Theorem

In this subsection, we prove the used Lagrange interpolating polynomial is secure so that the above-mentioned malicious attacks, including equation attack, external attack, collaborative attack, and reverse attack are meaningless for our scheme. The proof is shown as follows:

Theorem 1 (Lagrange interpolation): given t distinct points (x_i, y_i) of the form $(x_i, f(x_i))$, where $f(x)$ is a polynomial of degree less that t, then $f(x)$ is determined by:

$$f(x) = \sum_{i=1}^{t} y_i \prod_{\substack{1 \le j \le t \\ i \ne j}} \frac{x - x_i}{x_i - x_j} \tag{11}$$

The scheme of Shamir [30] is defined for a secret $s \epsilon Z/pZ$ with p prime, by setting $a_0 = s$, and selecting $a_1, ..., a_{t-1}$ randomly in Z/pZ. The trusted party computes $f(i)$, where:

$$f(x) = \sum_{k=0}^{t-1} a_k x^k \tag{12}$$

For all $1 \le i \le n$. The shares $(i, f(i))$ are distributed to the n distinct parties. Due to the fact that the secret is the constant term $s = a_0 = f(0)$, the secret is regained from any t shares $(i, f(i))$, for $I \subset \{1, ..., n\}$ by $s = \sum_{i \in I} c_i f(i)$, where each $c_i = \prod_{\substack{j \in I \\ j \ne i}} \frac{i}{j-i}$.

Exercise: prove the formula for the secret to be accurate by replacing into the formula of Lagrange's interpolation theorem.

Properties: rhe features of Shamir's secret sharing scheme are as follows: (1) all hypotheses are under proof, (2) perfect - all information is well-protected by the shares, and (3) ideal - every share is of the same size p as the secret. Comparatively, almost all public key cryptosystems depend on some familiar problems (discrete logarithm problems, integer factorization) for hardness so that the safety can be assured.

Proof of Lagrange interpolation theorem: suppose $g(x)$ is the right-hand side of equation (11). For each x_i in we verify directly that $f(x_i) = g(x_i)$, so as we can get $f(x) - g(x)$ is divisible by $x - x_i$. It follows that:

$$\prod_{i=1}^{t} (x - x_i) | (f(x) - g(x)) \tag{13}$$

However, because $\deg(f(x) - g(x)) \le t$, the only polynomial of this degree satisfying Equation (13) is $f(x) - g(x) = 0$. □

4.6. Problems with Multi-User Access Requests

The proposed digital-data-sharing system is a user-centered structure that integrates all kinds of teaching materials from different users. The collected data are stored in cloud servers to achieve the purpose of digital information integration and resources share and exchange.

Cloud computing environments show the characteristics of easy expansion and resource share in which it presents several advantages to satisfy the integration, share and exchange of digital materials. In such digital-data-sharing system, the requirements of users to rapidly propose access request and receive permission from cloud service providers should be satisfied.

For this reason, dynamic access schemes need to be established completely to ensure providing instant and entire services of digital data. The key is the services provided by the sharing system being able to support distinct dynamic access demands so as to correspond to the data and user change in cloud computing environments.

The proposed scheme and method are flexible and could deal with all the security management problems of dynamic keys, such as adding a new security class, removing an existing security class, and updating a user authorized. The involved solutions are simple, mainly addition and deduction,

that it does not require enormous computation and storage space for parameter update. Regarding the key-deriving formula $G(x, y)$ in Section 3:

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

Function $A_i(x)$ is related to information verification for verifying the existence of H_i in the legal verification list of CA and the use of personal superkey for verification. Function $B_i(y)$ relates to the data verification for verifying the authorization of a user to obtain the decryption key DK_u to further decrypt the encrypted digital materials.

The dynamic access requirements of sharing system in cloud are considered from two aspects: users and material data. First, users are changeable. Unlike static access model, which could establish all user parameters in the beginning of access scheme, the constant increase or removal of material authors, students, parents, publishers, and various teachers could propose new requests to the user-centered sharing system. User parameters need to be continuous updated to the initial access scheme to correspond to the dynamic users.

Second, material files require appending and revision. The integration of digital data comes from the different users, units, and information sources. In addition to the author, authorized users with requests should be able to update the material records and revise the documents in the sharing system. For this reason, the data of the materials could be appended and removed with dynamic requests after the establishment of access scheme.

In regard to the above considerations, the established formula $G(x, y)$ is nimble and flexible, which could be easily updated and revise the parameters instantaneously.

4.7. Discussion

In this subsection, we discuss the computational overheads and storage required for use of the proposed system. Definitions of notations used in performance evaluation of the proposed scheme are presented in Table 7.

Table 7. Notation table.

Definition	Notation
n	Number of security classes
m	Number of files
v_i	Degree of the polynomial f(x) (the system involves N security classes, each with v_i predecessors)
$\|p\|$	Bit-length of an integer p
$T_{l()}$	Time required to calculate an interpolating polynomial
T_{mul}	Time required for a multiplication computation

Knuth demonstrated that the process of interpolating at $(n + 1)$ points requires $(n^2 + n)/2$ divisions and $(n^2 + n)$ subtractions by Newton's formula, where n is the degree of the interpolating polynomial [46].

With regard to the evaluation of the polynomial for the derivation of the successor's secret parameters, Knuth demonstrated that this scheme requires $(2n - 1)$ multiplications and $(2n)$ additions in addition to one modular operation performed by applying Horner's rule.

Therefore, as Table 8 shows, the proposed scheme requires $2nT_{l()} + nT_{mul}$ to create $G(x, y)$ in the process of key generation, where $T_{l()}$ is the computation time for the interpolating polynomial. As described, the required computations are as follows: $T_{l()}$ = $(2n - 1)$ multiplications + $(2n)$

additions + 1 modular operation, $\left(\sum_{1\le i\le n} v_i + n\right)T_{l()} + nT_{mul}$. Thus, in total, this process spends $\left(\sum_{1\le i\le n} v_i + 3n\right)T_{l()} + 2nT_{mul}$. In terms of storage, the public parameters $G(x, y)$, u in this study require $(m + 1)|p|$, and the storage for each security class of a private key H_i is $|p|$.

Table 8. Analysis of computation complexity.

	Key Generation/Derivation	Storage of Public Parameters	Storage of Private Keys for Each Security Class
Proposed scheme	$\left(\sum_{1\le i\le n} v_i + 3n\right)T_{l()} + 2nT_{mul}$	(m+1)$\|p\|$	$\|p\|$

4.8. Comparison

With the advent of the era of cloud computing, the values of access mechanisms lie in their compatibility with various Internet applications as well as their security and efficiency. In this subsection, we compare confidentiality, data integrity, correctness and completeness, time complexity, and whether the key encryption scheme is possessed with other presented schemes. As showed in Table 9, four schemes proposed by Chung et al. [31], Liu et al. [15], Hsiao et al. [47], and ours achieve privacy protection (using notation O to express) due to their owning the key mechanism for encryption data. Specially, Hsiao et al.'s [47] and our schemes also provide access control function and are thus suitable for cloud environments. In respect of time complexity, Chung et al.'s method [31] is based on an elliptic curve cryptosystem, Liu et al.'s scheme [15] is based on the bilinear pairing, and Hsiao et al.'s model [47] is based on the discrete logarithm problem, time taken is exponential time and the time complexity is O(2^N). The proposed scheme is based on the lagrange interpolation polynomial, time complexity is only O(N). Due to no key generation and derivation process for the control systems of Trnka and Cerny [40] and A. S. M. Kayes et al. [37], no discussion occurred here. In addition, in both schemes, there is no encryption function for access files. The user who wants to access files can only be determined by the access control, their confidentiality are thus partially achieved (using notation Δ to express). Finally, all schemes are correctly completed on all process designs, so they can achieve correctness and completeness.

Table 9. Comparison with security requirements.

	(1)	(2)	(3)	(4)	(5)
Chung et al. (2008) [31]	O	Cryptography	O	O(2^N)	Yes
Liu et al. (2013) [15]	O	Cryptography	O	O(2^N)	Yes
Trnka and Cerny (2016) [40]	Δ	Access control	O	-	No
Hsiao et al. (2018) [47]	O	Access control and cryptography	O	O(2^N)	Yes
A. S. M. Kayes et al. (2019) [37]	Δ	Access control	O	-	No
Our proposal (2019)	O	Access control and cryptography	O	O(N)	Yes

(1) Confidentiality; (2) data integrity; (3) correctness and completeness; (4) complexity; (5) privacy protection.

5. Conclusions and Future Works

Cloud-based education has been actively promoted in recent years. Amid these efforts, the promotion of digital sharing systems is essential to ensure user and data security. To harness the immense benefits of cloud computing, we developed a cloud-based and learner-centered access control mechanism suitable for multi-user applications. The mechanism resolves the problems of managing

numerous users and reduces the complexity of access relationships. It also ensures the confidentiality and integrity of user data stored in the cloud and prevents unauthorized individuals from randomly accessing or modifying digital data. The integrated learning feature of sharing systems prevents repeated investment and development, protects the natural environment, and enhances economic efficiency. Therefore, these systems may be used to facilitate the shift of mobile services to the cloud and stimulate developments in the software industry.

In the future, we will be strengthening the design for context-awareness to reach the perfect combination of CAAC and RABC with encryption function. In addition to a more stable use for cloud sharing, the scheme will be going further to a wider application in key managements, and access control for the area of Big Data, the Internet of Things, and AI that emphasize dynamic authentication, e.g., voice-command access control, biometric access control, intelligence-learning access control and key-configuration mechanism.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fosnot, C.T.; Perry, R.S. *Constructivism: A Psychological Theory of Learning. Constructivism: Theory, Perspectives, and Practice*, 2nd ed.; Teachers College Press: New York, NY, USA, 2005; pp. 8–33.
2. Woolley, D.R. PLATO: The emergence of on-line community. *Comput.-Mediated Commun. Mag.* **1994**, *1*, 5.
3. Pivec, M.; Dziabenko, O.; Schinnerl, I. Aspects of game-based learning. In Proceedings of the 3rd International Conference on Knowledge Management, Graz, Austria, 2–4 July 2003.
4. Ebner, M.; Böckle, M.; Schön, M. Game Based Learning in Secondary Education: Geographical Knowledgeof Austria. In Proceedings of the 2011 World Conference on Educational Multimedia, Hypermediaand Telecommunications, Lisbon, Portugal, 27 June–1 July 2011.
5. Moschini, E. Designing for the smart player: Usability design and user-centred design in game-based learning. *Digit. Creat.* **2006**, *17*, 140–147. [CrossRef]
6. Prensky, M. Digital game-based learning. *Comput. Entertain. (CIE)* **2003**, *1*, 21. [CrossRef]
7. Mell, P.; Grance, T. *The NIST Definition of Cloud Computing*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2011.
8. Brunette, G.; Mogull, R. *Security Guidance for Critical Areas of Focus in Cloud Computing V4.0*; Cloud Security Alliance: Seattle, WA, USA, 2017.
9. Gens, F. New IDC It Cloud Services Survey: Top Benefits and Challenges. 2009. Available online: http://blogs.idc.com/ie/?p=730 (accessed on 18 October 2018).
10. Gai, K.; Qiu, M. Blend Arithmetic Operations on Tensor-Based Fully Homomorphic Encryption over Real Numbers. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3590–3598. [CrossRef]
11. Gai, K.; Qiu, M.; Ming, Z.; Zhao, H.; Qiu, L. Spoofing-Jamming Attack Strategy Using Optimal Power Distributions in Wireless Smart Grid Networks. *IEEE Trans. Smart Grid* **2017**, *8*, 2431–2439. [CrossRef]
12. Carminati, B.; Colombo, P.; Ferrari, E.; Sagirlar, G. Enhancing User Control on Personal Data Usage in Internet of Things Ecosystems. In Proceedings of the 2016 IEEE International Conference on Services Computing (SCC), San Francisco, CA, USA, 27 June–2 July 2016.
13. Sandhu, R.S.; Coyne, E.J.; Feinstein, H.L.; Youman, C.E. Role-Based Access Control Models. *IEEE Comput.* **1996**, *29*, 38–47. [CrossRef]
14. Li, M.; Yu, S.; Ren, K.; Lou, W. Securing Personal Health Records in Cloud Computing: Patient-centric and Fine-grained Data Access Control in Multi-owner Settings. In Proceedings of the International Conference on Security and Privacy in Communication Networks, Singapore, Singapore, 7–10 September 2010; pp. 89–106.
15. Liu, C.-H.; Lin, F.-Q.; Chiang, D.-L.; Chen, T.-L.; Chen, C.-S.; Lin, H.-Y.; Chung, Y.-F.; Chen, T.-S. Secure PHR Access Control Scheme for Healthcare Application Clouds. In Proceedings of the 2013 42nd International Conference on Parallel Processing, Lyon, France, 1–4 October 2013; pp. 1067–1076.
16. Saunders, G.; Hitchens, M.; Varadharajan, V. Role-Based Access Control and the Access Control Matrix. *ACM SIGOPS Oper. Syst. Rev.* **2001**, *35*, 6–20. [CrossRef]

17. Coulouris, G.; Dollimore, J.; Roberts, M. Role and Task-Based Access Control in the PerDiS Groupware Platform. In Proceedings of the 3rd ACM Workshop on Role-Based Access, Fairfax, VA, USA, 22–23 October 1998.
18. Joshi, J.B.D.; Bertino, E.; Latif, U.; Ghafoor, A. A Generalized Temporal Role-Based Access Control Model. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 4–23. [CrossRef]
19. Ott, A.; Fischer-Hübner, S. The Rule Set Based Access Control (RSBAC) Framework for Linux. 2004. Available online: http://www.rsbac.org/documentation/ (accessed on 8 March 2019).
20. Hansen, F.; Oleshchuk, V. SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems. In Proceedings of the 7th Nordic Workshop on Secure IT Systems, Narke, Sweden, 7–8 November 2002.
21. Park, J.S.; Costello, K.P.; Neven, T.M.; Diosomito, J.A. A Composite RBAC Approach for Large, Complex Organizations. In Proceedings of the 9th ACM Symposium on Access Control Models and Technologies, Yorktown Heights, NY, USA, 2–4 June 2004.
22. Wang, H.; Cao, J.; Zhang, Y. A Flexible Payment Scheme and Its Role-Based Access Control. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 425–436. [CrossRef]
23. Sandhu, R.S.; Munawer, Q. The RRA97 Model for Role-Based Administration of Role Hierarchies. In Proceedings of the 14th Annual Computer Security Applications Conference, Scottsdale, AZ, USA, 7–11 December 1998; pp. 39–49.
24. Sandhu, R.S.; Samarati, P. Access control: Principle and practice. *IEEE Commun. Mag.* **1994**, *32*, 40–48. [CrossRef]
25. Osborn, S.L.; Sandhu, R.S.; Munawer, Q. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Trans. Inf. Syst. Secur.* **2000**, *3*, 85–106. [CrossRef]
26. Ferraiolo, D.F.; Sandhu, R.S.; Gavrila, S.I.; Kuhn, D.R. Ramaswamy Chandramouli: Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.* **2001**, *4*, 224–274. [CrossRef]
27. Chen, T.-S.; Chung, Y.-F. Hierarchical access control based on Chinese Remainder Theorem and symmetric algorithm. *Comput. Secur.* **2002**, *21*, 565–570. [CrossRef]
28. Chen, T.-S.; Chung, Y.-F.; Tian, C.-S. A Novel Key Management Scheme for Dynamic Access Control in a User Hierarchy. In Proceedings of the COMPSAC 2004, Hong Kong, China, 28–30 September 2004; pp. 396–397.
29. Pan, J.-Y.; Chen, T.-L.; Chen, T.-S. A Novel Key Management and Access Control Scheme for Mobile Agent. In Proceedings of the 2006 International Conference on Intelligent Computing, Kunming, China, 16–19 August 2006; pp. 334–345.
30. Stallings, W. *Cryptography and Network Security: Principles and Practice*, 7th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2016.
31. Chung, Y.-F.; Lee, H.-H.; Lai, F.; Chen, T.-S. Access control in user hierarchy based on elliptic curve cryptosystem. *Inf. Sci.* **2008**, *178*, 230–243. [CrossRef]
32. Huang, K.-H.; Chung, Y.-F.; Liu, C.-H.; Lai, F.; Chen, T.-S. Efficient migration for mobile computing in distributed networks. *Comput. Stand. Interfaces* **2009**, *31*, 40–47. [CrossRef]
33. Liu, C.-H.; Chung, Y.-F.; Chen, T.-S.; Wang, S.-D. Access Control and Key Management Scheme Based on Bilinear Pairings over Elliptic Curves for Mobile Agent. In Proceedings of the 2009 Third International Conference on Multimedia and Ubiquitous Engineering, Qingdao, China, 4–6 June 2009; pp. 189–196.
34. Liu, C.-H.; Chung, Y.-F.; Chen, T.-S.; Wang, S.-D. Mobile Agent Application and Integration in Electronic Anamnesis System. *J. Med. Syst.* **2012**, *36*, 1009–1020. [CrossRef]
35. Chen, T.-S.; Liu, C.-H.; Chen, T.-L.; Chen, C.-S.; Bau, J.-G.; Lin, T.-C. Secure Dynamic Access Control Scheme of PHR in Cloud Computing. *J. Med. Syst.* **2012**, *36*, 4005–4020. [CrossRef]
36. Kayes, A.S.M.; Rahayu, W.; Dillon, T.; Chang, E.; Han, J. Context-aware access control with imprecise context characterization for cloud-based data resources. *Future Gener. Comput. Syst.* **2019**, *93*, 237–255. [CrossRef]
37. Kayes, A.S.M.; Han, J.; Rahayu, W.; Dillon, T.; Islam, M.S.; Colman, A. A Policy Model and Framework for Context-Aware Access Control to Information Resources. *Comput. J.* **2019**, *62*, 670–705. [CrossRef]
38. Schefer-Wenzl, S.; Strembeck, M. Modelling context-aware RBAC models for mobile business processes. *Int. J. Wirel. Mob. Comput.* **2013**, *6*, 448–462. [CrossRef]
39. Hosseinzadeh, S.; Virtanen, S.; Rodríguez, N.D.; Lilius, J. A semantic security framework and context-aware role-based access control ontology for smart spaces. In Proceedings of the International Workshop on Semantic Big Data, San Francisco, CA, USA, 26 June–1 July 2016.

40. Trnka, M.; Cerny, T. On security level usage in context-aware role-based access control. In Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, 4–8 April 2016; pp. 1192–1195.
41. Colombo, P.; Ferrari, E. Towards Virtual Private NoSQL datastores. In Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland, 16–20 May 2016; pp. 193–204.
42. Colombo, P.; Ferrari, E. Enhancing NoSQL datastores with fine-grained context-aware access control: A preliminary study on MongoDB. *Int. J. Cloud Comput.* **2017**, *6*, 292–305. [CrossRef]
43. Kayes, A.S.M.; Han, J.; Colman, A. An ontology-based approach to context-aware access control for software services. In Proceedings of the International Conference on Web Information Systems Engineering, Nanjing, China, 13–15 October 2013; pp. 410–420.
44. Kayes, A.S.M.; Han, J.; Colman, A.; Islam, M.S. RelBOSS: A relationship-aware access control framework for software services. In Proceedings of the 2014 OTM Confederated International Conferences "On the Move to Meaningful Internet Systems", Amantea, Italy, 27–31 October 2014; pp. 258–276.
45. Kayes, A.S.M.; Han, J.; Colman, A. PO-SAAC: A purpose-oriented situation-aware access control framework for software services. In Proceedings of the 2014 International Conference on Advanced Information Systems Engineering, Thessaloniki, Greece, 16–20 June 2014; pp. 58–74.
46. Szidarovszky, F.; Yakowitz, S. *Principles and Procedures of Numerical Analysis*; Springer: Boston, MA, USA, 1978.
47. Hsiao, T.C.; Wu, Z.Y.; Chen, T.L.; Chung, Y.F.; Chen, T.S. A hierarchical access control scheme based on Lagrange interpolation for mobile agents. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1–7. [CrossRef]