applied sciences

MDPI

Article

A Multi-User Ciphertext Policy Attribute-Based Encryption Scheme with Keyword Search for Medical Cloud System

Han-Yu Lin * and Yan-Ru Jiang

Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan; 10657031@mail.ntou.edu.tw

* Correspondence: hanyu@mail.ntou.edu.tw

Abstract: Population aging is currently a tough problem of many countries. How to utilize modern technologies (including both information and medical technologies) to improve the service quality of health information is an important issue. Personal Health Record (PHR) could be regarded as a kind of health information records of individuals. A ciphertext policy attribute-based encryption (CP-ABE) is a cryptographic primitive for fine-grained access control of outsourced data in clouds. In order to enable patients to effectively store his medical records and PHR data in medical clouds, we propose an improved multi-user CP-ABE scheme with the functionality of keyword search which enables data users to seek for specific ciphertext in the cloud server by using a specific keyword. Additionally, we adopt an independent proxy server in the proposed system architecture to isolate the communication between clients and the cloud server, so as to prevent cloud servers from suffering direct attacks and also reduce the computational loading of cloud servers. Compared with the previous approach, the proposed encryption algorithm takes less running time and the ciphertext length is also relatively short. Moreover, the procedures of re-encryption and pre-decryption only require one exponentiation computation, respectively.

Keywords: attribute-based encryption; ciphertext policy; medical cloud; keyword search; proxy re-encryption

Citation: Lin, H.; Jiang, Y. A Multi-User Ciphertext Policy Attribute-Based Encryption Scheme with Keyword Search for Medical Cloud System. *Appl. Sci.* **2021**, *11*, 63. https://dx.doi.org/10.3390/app11010063

Received: 14 November 2020
Accepted: 18 December 2020
Published: 23 December 2020

1. Introduction

With the evolution of wireless technology and mobile application devices, mobile commerce and computing have received much attention and sharing data on cloud servers equally becomes more convenient. Nevertheless, several potential security risks like the impersonation attacks and data disclosure should be noted. To protect sensitive data before being uploaded to cloud servers, data encryption is a commonly employed way of preventing the risk of data disclosure, i.e., a data owner encrypts data with his/her own key before uploading data to cloud centers such that only authorized data users could decrypt and recover original messages.

Personal health information is an important factor in the digitization of medical treatment. In addition to the electronic medical records of hospitalization and related surgery, everyone can independently check his/her own health information such as heartbeat, height, blood pressure, blood sugar and diet content. Such health information can be used as a basis for self-health monitoring. Generally speaking, current health information could be classified into personal health record (PHR) [1,2], electronic health record (EHR) [3] and electronic medical record (EMR) [4,5], which are explained as follows.

Person health record (PHR): This record is measured by the personal health sensing equipment. It can be used to measure the wearer's daily physiological data including everyday diet, exercise habits, heartbeat, blood sugar, medication, doctor's diagnosis records and other related information. Such health records maintained by the wearer could be provided to the physician for evaluation at the time of consultation. As for the long-term treatment, it can also save unnecessary medical expense.

Electronic health record (EHR): This record contains medical history, medication records, prescription, medication allergies, etc. It can be accessed by specific hospitals with the the patient's consent, so that medical personnel can easily search the relevant information about the patient. In addition, the record can be updated by patients and medical institutions at any time. With various information, medical staff are able to make more accurate and efficient diagnoses.

Electronic medical record (EMR): Nowadays, the electronic medical records used by medical institutions, the process of patient diagnosis and treatment are all kept in the medical cloud system by the physicians. The access right of the electronic medical records is owned by the physician, meaning that these data could be accessed by hospitals through the medical clouds.

In medical cloud environments, there will be three main roles: a medical data owner, a cloud server and a medical user. The medical data owner will store his/her own health data on the cloud server and then authorize specific users such as patients and medical staffs to access the data. However, since cloud servers are not completely trustworthy, there will be many security misgivings such as data leakage and data loss. When the data are stored in the clouds with the form of ciphertext, it would be difficult to perform the functionality of keyword search. Consequently, a suitable cryptographic mechanism which could deal with the above scenarios is quite important. In this work, we elaborate on the merits of attribute-based encryption mechanisms and come up with a multi-user ciphertext policy attribute-based encryption scheme with the functionality of keyword search for medical cloud systems.

1.1. Related Work

In 1998, Blaze et al. [6] presented the first proxy-re encryption (PRE) in which a semi-trusted agent could obtain one re-encryption key to transform the ciphertext intended for the user A to the ciphertext intended for the user B, so that the user B could decrypt the ciphertext with his/her private key. In order to ensure the confidentiality of the ciphertext during the re-encryption process, the agent should not be capable of learning any information related to encrypted messages. In 2005, Ateniese et al. [7] introduced the PRE mechanism based on bilinear pairings and proved that their scheme could withstand chosen-ciphertext attacks. In 2007, Canetti and Hohenberger [8] addressed a multiple-hop PRE scheme which could also resist the chosen-ciphertext attack. Meanwhile, they proved the security of their mechanism in the standard security proof model. Thinking of identity-based cryptosystems, in 2016, Wang et al. [9] presented two ID-based PRE variants which are suitable for secure data sharing in the cloud. Considering secure cloud storage, in 2018, Zeng and Choo [10] addressed a so-called conditional PRE scheme with efficient computation. Nowadays, the PRE mechanism has many practical applications [6,11–16] like the forwarding of e-mails, the distribution of private keys and the management of key escrow.

To further provide the ciphertext with the search function, Boneh et al. [17] introduced the mechanism of public key encryptions with the characteristic of keyword search (PEKS) utilizing the assumption of Decision Diffie–Hellman Problem (DDHP). In their scheme, a data owner can use the public key to encrypt data along with keywords such that data users could request the keyword search on stored ciphertexts and decrypt it. In 2016, Cui et al. [18] realized an expressive PEKS system in prime order groups. Their scheme supports the multi-keyword search using the expressive Boolean formula. They also proved the security of their scheme in the standard model. To date, lots of methods for PEKS have been presented. The PEKS is a security mechanism that is often applied to cloud storage and could be utilized to seek for encrypted data. Unfortunately, the scheme is inherently subject to the attacks of keyword guessing. To solve this security flaw, in 2016, Chen et al. [19] presented a sever-aided public key encryption with a keyword search (SA-PEKS). In their construction, a semi-trust keyword server (KS) will be independent from the storage server and the data user must be authenticated by the KS to obtain

credentials. Their scheme is different from the PEKS as users must request the KS to run an authentication protocol for generating a KS-derived keyword in SA-PEKS. Consequently, it is secure against the offline keyword-guessing attacks.

In order to share a ciphertext with many persons, Sahai and Water [20] presented an extension of IBE, called an attribute-based encryption (ABE) scheme. ABE is a promising cryptographic primitive which offers reliable and dynamic data-sharing. The notion of ABE utilizes the attributes of users to match an access policy associated with the ciphertext or the private key. More precisely, the ABE schemes could be classified into two categories, namely Ciphertext-Policy ABE (CP-ABE) [21] and Key-Policy ABE (KP-ABE) [22]. In 2007, Bethencourt et al. [21] introduced the CP-ABE. In a CP-ABE cloud storage system, a data owner encrypts data into ciphertext and then specifies an access policy before storing the ciphertext in the cloud server. The data user's private key is correlated with a set of attributes based on his/her identity. On the other hand, in a KP-ABE cloud storage system, a data owner's ciphertext is associated with a set of attributes, while the data user's private key is correlated with an access policy. In 2017, Lin et al. [23] proposed a collaborative key management mechanism using CP-ABE to share data in clouds. In their scheme, a key authority will be responsible for generating, issuing and storing private keys. Using the attribute group key to perform the private key update, their system could achieve immediate attribute revocation. In 2019, Sethia et al. [24] proposed a constant-size CP-ABE scheme with scalable revocation for resource-constrained IoT devices. They not only successfully implemented the proposed scheme, but also showed that their work is chosen-ciphertext attack (CCA)-secure. In 2020, Zhou et al. [25] proposed a multi-authority CP-ABE access model in multicloud. In particular, they introduced an attribute mapping method to handle both problems of data-sharing security and policy conflict in multicloud storage systems (MCSS).

1.2. Contributions

In this work, the authors aim at the access control and data-sharing of cloud health information. Using attribute-based encryption schemes, we further combine the techniques of ciphertext keyword search, linear secret-sharing and proxy re-encryption. In addition to being applicable to medical cloud systems, the proposed approach could also be applied to the general cloud environments. Specifically, we improve Wang et al.'s system [26] by reducing the ciphertext length and the computational complexity of encryption algorithm, and further introducing the role of proxy server to enhance the communication security. The shorter ciphertext length of our system also benefits the savings of communication overheads. Technically speaking, we combine users' attribute private keys with the public key of the proxy server, so that a legal trapdoor generated by the data user must be further converted by the proxy server. Besides, the ciphertext sent by the data owner is also re-encrypted by the proxy server before it is stored in the cloud server. Consequently, when a data user requests an encrypted message from the cloud server, this message should be pre-decrypted by the proxy server before it can be correctly recovered by the data user.

2. Preliminaries

We review the operation of bilinear pairings [27,28] and the technique of the linear secret-sharing scheme (LSSS) [29] in this section.

Bilinear Pairing

Let the symbols of G_1 and G_2 be two multiplicative cyclic groups in which the order is a prime number p and g is the generator of the group G_1. A bilinear map is expressed as e: $G_1 \times G_1 \rightarrow G_2$ with the following properties:

(i) ***Bilinearity:***

For all elements $g, h \in G_1$ and $a, b \in_R Z_p$, we have $e(g^a, h^b) = e(g, h)^{ab}$;

(ii) ***Non-degeneracy:***

Let a generator in the group of G_1 be p. Then the value $e(p, p)$ would also be a generator in the group G_2;

(iii) ***Computability:***

There is an efficient algorithm to compute e(g, h) for all elements $g, h \in G_1$.

Linear Secret Sharing Scheme (LSSS) **[29]**

Let $M_{l\times n}$ be a matrix with the row number of l and the column number of n, M_i the i-th row of the matrix and a mapping function ρ: $\{1, 2, \ldots, l\}\rightarrow$**P** which converts a row to a label of party. We say that a secret-sharing scheme Π for the access structure $\mathbb{A}$ over a set of parties **P** could be denoted as a linear secret-sharing method (LSSS) from Z_p. It is also represented as $(M_{l\times n}, \rho)$, which consists of the following two effective algorithms:

(i) **Share($(M_{l\times n}, \rho)$, s):** The share algorithm takes a secret-sharing value $s \in_R Z_p$ as an input, randomly chooses $y_2, \ldots, y_n \in_R Z_p$ and defines $v = (s, y_2, \ldots, y_n)$. At last, it outputs $M_{l\times n} \cdot v$ as the vectors of l shares. That is, the shared value obtained by the participant $\rho(i)$ is $\lambda_i = \langle M_{l\times n} \cdot v \rangle$;

(ii) **Recon($(M_{l\times n}, \rho)$, D):** The reconstruction algorithm takes an access set $D \in \mathbb{A}$ as the input. Let $\mathbf{I} = \{i \mid \rho(i) \in D\}$. It will return a set of constants $\{\mu_i\}_{i \in \mathbf{I}}$ fulfilling that $\Sigma_{i\in\mathbf{I}}\ \mu_i \cdot \lambda_i = s$.

3. The Proposed Scheme

We introduce an improved CP-ABE scheme with the functionality of keyword search based on Wang et al.'s construction [26]. In particular, we add a proxy server to the proposed system architecture for protecting the medical cloud server from direct attacks. Although two added algorithms, i.e., Re-Encrypt and Pre-Decrypt, would slightly increase the overall computational costs, we believe that it is a worthy tradeoff to obtain a higher security level.

3.1. System Model

Figure 1 shows the system model of the proposed scheme. There are five main parties in the proposed system: trusted authority (TA), proxy server (PS), medical cloud server (CS), data owner (DO) and the data user (DU).

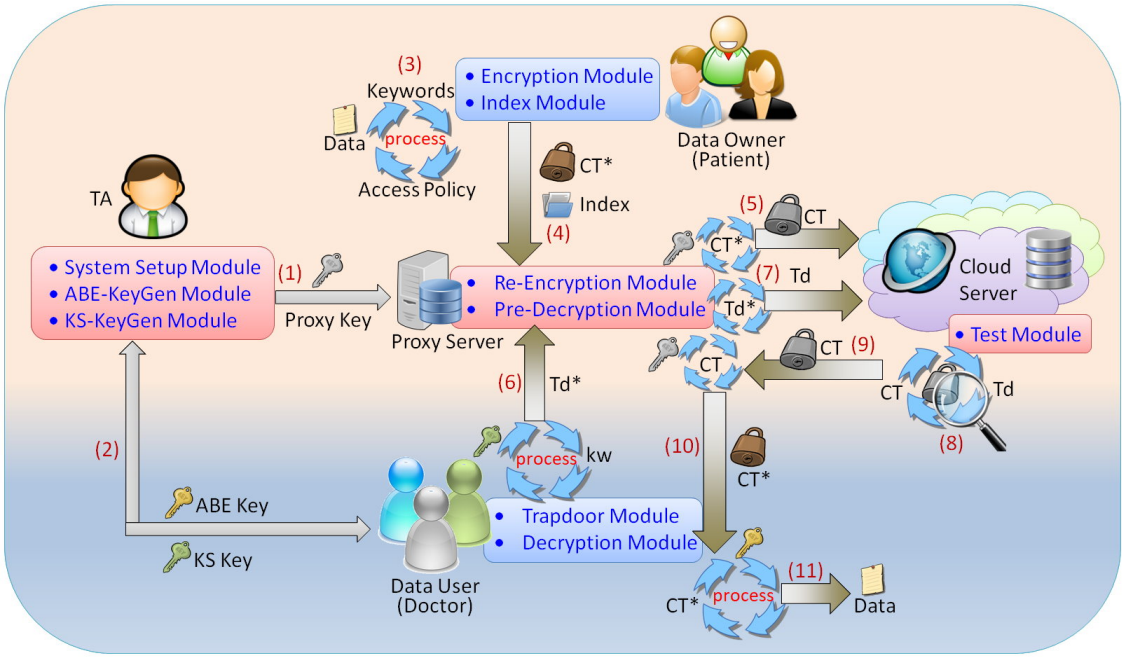

Figure 1. The architecture and data flow of the proposed system.

Trusted Authority (TA): This would be responsible for the system initialization and generating public parameters, distributing attribute keys (ABE-Keys) and keyword search keys (KS-Keys) associated with DU's attributes. Moreover, the TA also creates the key pair of proxy servers.

Data Owner (DO): The DO is a patient who defines an access policy for the PHR data or medical records and then encrypts the corresponding keywords into indexes. At last, the DO uploads the ciphertext along with indexes to the cloud.

Cloud Server (CS): The CS provides the services of storage and data retrieval. It will store encrypted data and indexes outsourced by the DO. After receiving requests from data users, it executes search procedures and transmits the matched ciphertext back.

Data User (DU): The DU is a doctor who can access patients' encrypted data from CS. Specifically, the DU owns the KS-Key and the ABE-key; the former is used to generate trapdoor information while the latter is used to decrypt ciphertexts.

Proxy Server (PS): The PS is an important role between the DO/DU and the CS. The PS is responsible for performing the proxy re-encryption process. After receiving the ciphertext uploaded by the DO, it will execute the re-encryption operation and then transmit the re-encrypted ciphertert to the CS. Similarly, after receiving the ciphertext downloaded by the DU, it will execute pre-decryption process and then return the ciphertext to the DU. Furthermore, when receiving trapdoors of keyword search requests, the PS also modifies it before transmitting it to the CS.

3.2. Algorithms

The proposed system could be divided into ten algorithms stated below.

Setup: Taking as input a security parameter k, the algorithm generates system's public parameters *params* and a master secret key *msk*.

ABE-KeyGen: The attribute key generation algorithm takes as input system parameters *params*, a master secret key *msk*, a user's identity *id* and the attribute set $\boldsymbol{w}$. It generates an attribute private key $d_{id,\,\boldsymbol{w}}$ associated with user's identity *id*.

KS-KeyGen: The keyword search key generation algorithm takes as input system parameters *params*, a master secret key *msk* and a user identity *id*. It generates a keyword search key q_{id} for *id*.

Encrypt: The encryption algorithm takes as input system parameters *params*, an LSSS access structure $\mathbb{A}$ and a message *msg*. It generates a ciphertext *ct*.

Re-Encrypt: The re-encryption algorithm takes as input system parameters *params*, a ciphertext *ct* and the private key of PS, say β. It generates a re-encrypted ciphertext *ct**.

Index: The index algorithm takes as input system parameters *params* and a keyword set *kw*. It produces a secure index *IX*(*kw*).

Trapdoor: The trapdoor algorithm takes as input system parameters *params*, a user's keyword search key q_{id}, a user's attribute private key $d_{id,\,\boldsymbol{w}}$, a keyword set *kw* and the private key of PS. It generates a trapdoor $T_{\boldsymbol{kw}}$ corresponding to the keyword set *kw*.

Test: The test algorithm takes as input system parameters *params*, a trapdoor $T_{\boldsymbol{kw}}$ and an index *IX*(*kw*). It will return either an intermediate result Q_{ct} or 0.

Pre-Decrypt: The pre-decryption algorithm takes as input system parameters *params*, a re-encrypted ciphertext *ct** and the private key of PS, say β. It outputs a ciphertext *ct*.

Decrypt: The input of this algorithm includes system parameters *params*, a ciphertext *ct* with the intermediate data $Q_{ct,}$ and the user's attribute private key $d_{id,\,\boldsymbol{w}}$. Finally, the algorithm will return an original message *msg* or an error symbol $\perp$.

3.3. Concrete Construction

Based on the previously defined algorithms, the authors will introduce a substantial formation of multi-user CP-ABE scheme with keyword search for medical cloud systems. Table 1 first summarizes several utilized notations. The operation of each algorithm is concretely described as follows.

Table 1. Symbol notations.

Notation	Description
p	a large prime
G_1, G_2	multiplicative cyclic groups of prime order p
g	a generator of G_1
e	bilinear pairing function satisfying e: $G_1 \times G_1 \rightarrow G_2$
H	one-way hash function
F	message authentication code (MAC)
d	the private key of TA
g^d, $e(g, g)^d$	the public keys of TA
β	the private key of PS
g^β	the public key of PS
Ω	universe of all attributes, i.e., $\Omega = \{attr_1, \ldots, attr_n\}$
w	the attribute set of DU, $w \subseteq \Omega$
Λ	the set of all keywords, i.e., $\Lambda = \{kw_1, \ldots, kw_l\}$
kw	a subset of keywords, i.e., $kw \subseteq \Lambda$

Setup: On inputting a security parameter k, the TA chooses a random numbers $d \in Z_p^*$ as the master secret key, and then generates essential parameters, as shown in Table 1.

ABE-KeyGen: Each DU can request his/her attribute key (ABE-Key) from the TA by sending the correlated identity id and attribute set $\boldsymbol{w}$. Utilizing the information $\boldsymbol{w}$ and id, the TA first selects $t \in_R Z_p^*$ to derive

$$K = (g^t)^d (g^d), \tag{1}$$

$$L = (g^\beta)^t, \tag{2}$$

$$\{K_x = H(x)^t\}_{x \in \boldsymbol{w}}. \tag{3}$$

Then, TA records (id, g^{dt}) in the user list and the ABE-Key $d_{id, \boldsymbol{w}} = (K, L, \{K_x\})$ would be returned to the DU through a secure channel.

KS-KeyGen: Each DU could request his/her keyword search key (KS-Key) from the TA by sending the correlated identity id. Utilizing the information id, the TA first checks whether the user's identity is stored in the user list. Otherwise, the TA will deny this request. Then, DU chooses a random number $u \in Z_p^*$ to compute the following parameter for the TA.

$$q_u = g^u \tag{4}$$

After receiving q_u, the TA retrieves g^{dt} from the user list to compute

$$q_{id} = (g^{dt})(q_u{}^d). \tag{5}$$

The KS-Key q_{id} is sent back to the DU through a secure channel. DU also keeps the chosen random number u for decryption in the future.

Encrypt: When DO wants to encrypt his/her sensitive data msg for uploading to the CS, DO first defines an access policy $\mathbb{A}$ over a set of parties **P** who are able to decrypt his/her sensitive data msg. According to LSSS, the secret-sharing scheme for the access policy $\mathbb{A}$ could be expressed as $(M_{l \times n}, \rho)$ using a random vector $\boldsymbol{v} = (s, y_2, \ldots, y_n) \in_R Z_p^*$ and a mapping function ρ. Then, DO utilizes the share algorithm of LSSS to obtain l share values $\lambda_i = \langle \boldsymbol{M}_i, \boldsymbol{v} \rangle$ and computes the following parameters:

$$C = [e(g, g)^d]^s \cdot (msg), \tag{6}$$

$$C' = g^s, \tag{7}$$

$$\{C_i = g^{d\lambda_i} H(\rho(i))^{-s}\}, \tag{8}$$

Here, the ciphertext ct is composed of $(C, C', \{C_i\})$.

Index: For creating an index in relation to the keyword set $\boldsymbol{kw}$, the DO first picks an integer $t_i \in_R Z_p^*$ for each keyword $kw_i \in \boldsymbol{kw}$, and then computes

$$\{k_i = [e(g, g)^d]^s \cdot e(g, H(kw_i))^s\}_{kw_i \in \boldsymbol{kw}}, \tag{9}$$

$$IX(\boldsymbol{kw}) = \{(t_i, f_i = F(k_i, t_i))\}_{kw_i \in \boldsymbol{kw}}, \tag{10}$$

Next, the index set $IX(\boldsymbol{kw})$ along with the corresponding ciphertext ct and the access policy $(M_{l\times n}, \rho)$ are delivered to the PS.

Re-Encrypt: After receiving the ciphertext ct from the DO, the PS first utilizes its private key β to re-encrypt the parameter C in ciphertext as

$$C^* = C^{\beta}. \tag{11}$$

The re-encrypted ciphertext ct^* is formed by $(C^*, C', \{C_i\})$. Then the re-encrypted ciphertext ct^* along with the access policy $(M_{l\times n}, \rho)$ and index $IX(\boldsymbol{kw})$ will be stored by the CS.

Trapdoor: The DU can utilize the keyword search key q_{id} to generate a trapdoor for ciphertext search. Let kw be the searched keyword and $\boldsymbol{w}$ the attribute set of DU. First, the DU computes

$$T_q(kw) = (g^{H(kw)})({q_{id}}^{-u}), \tag{12}$$

$$L'' = L^{-u}, \tag{13}$$

$$\{{K'}_x = {K_x}^{-u}\}_{x\in \boldsymbol{w}}. \tag{14}$$

Then, the trapdoor $T'_{kw} = (T_q(kw), L'', \{{K'}_x\}_{x\in \boldsymbol{w}})$ and the attribute set w are transferred to the PS. When the PS receives $(T'_{kw}, \boldsymbol{w})$ from the DU, the PS computes the parameter L' as

$$L' = (L'')^{\beta^{-1}}. \tag{15}$$

Then, the modified trapdoor $T_{kw} = (T_q(kw), L', \{{K'}_x\}_{x\in \boldsymbol{w}})$ and the attribute set $\boldsymbol{w}$ are delivered to the CS.

Test: After receiving $(T_{kw}, \boldsymbol{w})$, the CS checks all ciphertext index data $(ct^*, IX(\boldsymbol{kw}))$ and the embedded access policy $(M_{l\times n}, \rho)$. Suppose that $\boldsymbol{w}$ fulfills the access policy $(M_{l\times n}, \rho)$ of some ciphertext ct^*. Let $\{\mu_i\}_{i\in \mathbf{I}}$ be a set of constants outputted by the Recon algorithm of LSSS. Then the CS further computes

$$Q_{ct} = \Pi_{i \in \mathbf{I}}[e(C_i, L')e(C', {K'}_{\rho(i)})]^{\mu_i}, \tag{16}$$

$$k_{kw} = e(C', T_q(kw))/Q_{ct}. \tag{17}$$

and verifies whether $IX(\boldsymbol{kw})$ satisfies

$$\{f_i = F(k_{kw}, t_i)\}_{f_i \in IX(\boldsymbol{kw})}. \tag{18}$$

If Equation (18) holds, we know that $kw \in \boldsymbol{kw}$ and the CS will transmit (ct^*, Q_{ct}) to the PS.

We can first derive Equation (16) as follows

$$\begin{aligned} Q_{ct} &= \Pi_{i \in \mathbf{I}}[e(C_i, L')e(C', {K'}_{\rho(i)})]^{\mu_i} \\ &= \Pi_{i \in \mathbf{I}}[e(g^{d\lambda_i}H(\rho(i))^{-s}, g^t)e(g^s, H(\rho(i))^t)]^{\mu_i/u} \\ &= \Pi_{i \in \mathbf{I}}[e(g^{d\lambda_i}, g^t)]^{\mu_i/u} \\ &= e(g, g)^{dts/u} \end{aligned}$$

Then, we further derive Equation (17) as follows

$$\begin{aligned} k_{kw} &= e(C', T_q(kw))/Q_{ct} \\ &= e(g^s, H(kw){q_{id}}^{-u})/e(g,g)^{dts/u} \\ &= e(g^s, H(kw)(g^{dt}{q_u}^d)^{-u})/e(g,g)^{dts/u} \\ &= e(g^s, H(kw)(g^{dt/u}g^d))/e(g,g)^{dts/u} \\ &= [e(g,g)^d]^s \cdot e(g, H(kw))^s \end{aligned}$$

Consequently, the correctness of Equation (18) could be easily derived below

$$\begin{aligned} &F(k_{kw}, t_i) \\ &= e\ (e(g, (g^b)^{sH(kw)}(g^a)^s), t_i) \\ &= e(k_i, t_i) \\ &= f_i \end{aligned}$$

Pre-Decrypt: After receiving (ct^*, Q_{ct}) where $ct^* = (C^*, C', \{C_i\})$, the PS runs the pre-decryption process by computing

$$C = (C^*)^{\beta^{-1}}. \tag{19}$$

The new ciphertext $ct = (C, C', \{C_i\})$ and Q_{ct} are transferred to the DU

Decrypt: When the DU receives $ct = (C, C', \{C_i\})$ from the PS, it can utilize its attribute key K to recover

$$msg = C \cdot {Q_{ct}}^u / e(C', K). \tag{20}$$

The correctness of Equation (20) could be confirmed below:

$$\begin{aligned} &C \cdot {Q_{ct}}^u / e(C', K) \\ &= msg \cdot [e(g,g)^d]^s \cdot e(g,g)^{dts} / e(g^s, g^{dt}g^d) \\ &= msg \cdot e(g,g)^{ds+dts} / e(g,g)^{s(dt+d)} \\ &= msg \end{aligned}$$

Note that in the practical usage, a data owner (patient) should utilize the system parameters generated by the TA rather than re-initializing the whole system parameters each time. The role of proxy server is semi-trusted, i.e., it might be curious about the received ciphertext, but will not deviate from the predefined process. When receiving an identical ciphertext, the cloud server might overwrite the existing one or just abort current process depending on the predefined rules. We display the pseudo-code of the proposed system in Table 2.

Table 2. The pseudo-code of the proposed system.

Setup(1^k)
Choose system parameters p, $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, g, e, H, F, d, β, $\Omega = \{attr_1, \ldots, attr_n\}$, $\boldsymbol{w}$, $\Lambda = \{kw_1, \ldots, kw_l\}$;
ABE-KeyGen$(id, \boldsymbol{w})$
Choose $t \in_R Z_p{}^*$; Compute $K = (g^t)^d(g^d)$, $L = (g^\beta)^t$, $\{K_x = H(x)^t\}_{x \in w}$; Record (id, g^{dt}) in the user list; return $d_{id,w} = (K, L, \{K_x\})$;
KS-KeyGen(id)
If (id is in the user list) then Compute $q_u = g^u$ where $u \in Z_p{}^*$; Retrieve g^{dt} from the user list; return $q_{id} = (g^{dt})(q_u{}^d)$;
Encrypt$(\mathbb{A}, msg)$
Choose a random vector $v = (s, y_2, \ldots, y_n) \in_R Z_p{}^*$; Compute $\lambda_i = \langle \boldsymbol{M}_i, v \rangle$, $C = [e(g,g)^d]^s \cdot (msg)$, $C' = g^s$, $\{C_i = g^{d\lambda_i} H(\rho(i))^{-s}\}$; return $ct = (C, C', \{C_i\})$;
Index$(\boldsymbol{kw})$
Choose $t_i \in_R Z_p{}^*$ for each $kw_i \in \boldsymbol{kw}$; Compute $\{k_i = [e(g,g)^d]^s \cdot e(g, H(kw_i))^s\}_{kw_i \in kw}$, $IX(\boldsymbol{kw}) = \{(t_i, f_i = F(k_i, t_i))\}_{kw_i \in \boldsymbol{kw}}$; return $IX(kw)$;
Re-Encrypt(ct, β)
Compute $C^* = C^\beta$ and return $ct^* = (C^*, C', \{C_i\})$;
Trapdoor$(q_{id}, kw, d_{id,w}, \boldsymbol{w}, \beta)$
Compute $T_q(kw) = (g^{H(kw)})(q_{id}{}^{-u})$, $L'' = L^{-u}$, $\{K'_x = K_x{}^{-u}\}_{x \in w}$, $L' = (L'')^{\beta^{-1}}$; return $T_{kw} = (T_q(kw), L', \{K'_x\}_{x \in w})$;
Test$(T_{kw}, \boldsymbol{w})$
Check all $(ct^*, IX(\boldsymbol{kw}))$ and embedded access policy $(M_{l \times n}, \rho)$; If (w fulfills the access policy $(M_{l \times n}, \rho)$ of some ciphertext ct^*) then Compute $Q_{ct} = \Pi_{i \in I}[e(C_i, L')e(C', K'_{\rho(i)})]^{\mu i}$, $k_{kw} = e(C', T_q(kw))/Q_{ct}$; If *(IX(kw)* satisfies $\{f_i = F(k_{kw}, t_i)\}_{f_i \in IX(kw)}$*) then return* (ct^*, Q_{ct}); return $\perp$;
Pre-Decrypt(ct^*, Q_{ct}, β)
Compute $C = (C^*)^{\beta^{-1}}$ and return Q_{ct} & $ct = (C, C', \{C_i\})$;
Decrypt$(ct, Q_{ct}, d_{id,w})$
return $msg = C \cdot Q_{ct}{}^u / e(C', K)$;

4. Security Proof and Efficiency

We first formally prove that the proposed system is secure against the adaptive chosen-ciphertext attacks (CCA2), assuming the hardness of Decisional Diffie–Hellman Problem (DDHP) in the random oracle model as Theorem 1.

Theorem 1. *The proposed multi-user CP-ABE with keyword search is secure against the adaptive chosen-ciphertext attacks (CCA2) in the random oracle model provided that there is no polynomial-time adversary having a non-negligible advantage to break the intractable DDHP.*

Proof. We will complete this security proofs by showing that if the ciphertext indistinguishability of the proposed system could be broken by a polynomial-time adversary $\mathcal{A}$, another DDH distinguisher $\mathcal{B}$ could also be built by calling $\mathcal{A}$ as a subroutine. The goal of $\mathcal{B}$ is to decide whether $g^c = g^{ab}$ or not by giving a DDHP instance (g, g^a, g^b, g^c). In the following simulation game, the distinguisher $\mathcal{B}$ is responsible for answering queries submitted by the adversary $\mathcal{A}$. □

Setup: Initially, $\mathcal{B}$ chooses $\beta \in Z_p^*$ and computes g^β as the key pair of PS. The public key of TA is set as $(g^c, e(g^c, g))$. Then $\mathcal{B}$ transmits necessary system parameters $\{p, \boldsymbol{G}_1, \boldsymbol{G}_2, g, e, H, F, g^\beta, g^c, e(g^c, g), \Omega, \boldsymbol{w}, \Lambda\}$ to $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ will interact with the distinguisher $\mathcal{B}$ below.

H oracle: When $\mathcal{A}$ submits an $H(x)$ query, $\mathcal{B}$ returns with an integer $v_1 \in_R Z_p^*$. The record (x, v_1) is also stored in the H-list.

F oracle: When $\mathcal{A}$ submits an $F(k_i, t_i)$ query, $\mathcal{B}$ responds with an integer $v_2 \in_R Z_p^*$. The record (k_i, t_i, v_2) is also stored in the F-list.

ABE-KeyGen oracle: When $\mathcal{A}$ submits an *ABE-KeyGen*$(id, \boldsymbol{w})$ query, $\mathcal{B}$ selects $t \in_R Z_p^*$ to derive $K = (g^c)^t(g^c)$, $L = (g^\beta)^t$, $\{K_x = H(x)^t\}_{x \in \boldsymbol{w}}$, and keeps (id, g^{ct}) in the user list. Finally, $\mathcal{B}$ returns the ABE-Key $d_{id, \boldsymbol{w}} = (K, L, \{K_x\})$.

KS-KeyGen oracle: When $\mathcal{A}$ submits a *KS-KeyGen*(id) query, $\mathcal{B}$ first checks if id is stored in the user list and then retrieves g^{ct} to compute $q_u = g^u$ and $q_{id} = (g^{ct})(g^{cu})$ where $u \in Z_p^*$. The value (q_{id}, u) is also returned to $\mathcal{A}$ as a result.

Encrypt oracle: When $\mathcal{A}$ submits an *Encrypt*$(\mathbb{A}, msg)$ query, $\mathcal{B}$ first chooses a random vector $\boldsymbol{v} = (s, y_2, \ldots, y_n) \in_R Z_p^*$ and computes $\lambda_i = \langle \boldsymbol{M}_i, \boldsymbol{v} \rangle$, $C = [e(g^c, g)]^s \cdot (msg)$, $C' = g^s$, $\{C_i = g^{c\lambda_i} H(\rho(i))^{-s}\}$. The ciphertext $ct = (C, C', \{C_i\})$ is returned to $\mathcal{A}$ as a result.

Index oracle: When $\mathcal{A}$ submits an *Index*$(\boldsymbol{kw})$ query, $\mathcal{B}$ chooses $t_i \in_R Z_p^*$ for each keyword $kw_i \in \boldsymbol{kw}$, and then computes $\{k_i = [e(g, g^c)]^s \cdot e(g, H(kw_i))^s\}_{kw_i \in \boldsymbol{kw}}$ and $IX(\boldsymbol{kw}) = \{(t_i, f_i = F(k_i, t_i))\}_{kw_i \in \boldsymbol{kw}}$. At last, the index set $IX(\boldsymbol{kw})$ is returned to $\mathcal{A}$ as a result.

Re-Encrypt oracle: When $\mathcal{A}$ submits a *Re-Encrypt*(ct) query, $\mathcal{B}$ returns $ct^* = (C^*, C', \{C_i\})$ where $C^* = C^\beta$.

Trapdoor oracle: When $\mathcal{A}$ submits a *Trapdoor*$(id, q_{id}, u, kw, d_{id, \boldsymbol{w}}, \boldsymbol{w})$ query, $\mathcal{B}$ runs the Trapdoor algorithm of the proposed system and then returns $T_{kw} = (T_q(kw), L', \{K'_x\}_{x \in \boldsymbol{w}})$ to $\mathcal{A}$.

Test oracle: When $\mathcal{A}$ submits a *Test*$(T_{kw}, \boldsymbol{w})$ query, $\mathcal{B}$ runs the Test algorithm of the proposed system and then returns the result to $\mathcal{A}$.

Pre-Decrypt oracle: When $\mathcal{A}$ submits a *Pre-Decrypt*(ct^*, Q_{ct}) query, $\mathcal{B}$ runs the Pre-Decrypt algorithm of the proposed system and then returns $ct = (C, C', \{C_i\})$ & Q_{ct} to $\mathcal{A}$.

Decrypt oracle: When $\mathcal{A}$ submits a *Decrypt*(id, ct^*, Q_{ct}) query, $\mathcal{B}$ first find out the ABE-Key $d_{id, \boldsymbol{w}}$ from previous history of ABE-KeyGen queries and then runs the Decrypt algorithm. Then the recovered msg is returned to $\mathcal{A}$.

Challenge: The adversary $\mathcal{A}$ would deliver $\mathcal{B}$ two messages, (msg_0, msg_1) of the same length and an access policy $\mathbb{A}$. Next, $\mathcal{B}$ decides m_λ using an internal flipped coin $\lambda \leftarrow \{0, 1\}$ and generates a corresponding ciphertext ct' according to the steps of previous Encrypt queries except that $C = [e(g^a, g^b)]^s \cdot (msg_\lambda)$. The ciphertext ct' is the target challenge for $\mathcal{A}$.

Phase 2: The adversary $\mathcal{A}$ could make new queries, such as those mentioned in Phase 1. However, it is not allowed to directly make a *Decrypt* query on the challenged ciphertext. At the end of this game, $\mathcal{A}$ has to output a bit λ'. If $\lambda' = \lambda$, we say that $\mathcal{A}$ wins this game.

Analysis of the game: In the above simulation game, $\mathcal{B}$ responds to each query made by $\mathcal{A}$ with an indistinguisbable response without termination. Therefore, we could claim that if the adversary $\mathcal{A}$ has the non-negligible advantage ε to break the ciphertext indistinguishability of the proposed scheme, the distinguisher $\mathcal{B}$ also has the non-negligible advantage ε to solve the given DDHP instance. Precisely specking, when $\lambda' = \lambda$, we could know that $g^c = g^{ab}$. Otherwise, $g^c \neq g^{ab}$.

Q.E.D.

Considering potential data risks in medical cloud systems, the authors analyze the essential security requirements with respect to the constructed system as follows.

(1) **User-controlled access control**

In our system, the encryption process uses an attribute-based encryption mechanism. That is, the DO can formulate a corresponding access structure for his/her ciphertext and authorize it to a specific DU through the access structure. In other words, when the DU wants to access the ciphertext stored in the cloud, the attribute set corresponding to the DU's private key must satisfy the access structure of desired ciphertext. Therefore, the proposed system has the characteristic of user-controlled access control.

(2) **Multi-user sharing**

Cloud data sharing mechanisms could be categorized into two types, i.e., single-user and multi-user. Traditionally, it is not secure to achieve multi-user data sharing by using the technique of private key sharing. In our system, the idea of multi-user sharing is realized by utilizing the technique of linear secret-sharing (LSSS) which randomly chooses a secret-sharing value s through the Share algorithm to achieve multi-user sharing and avoid the risk of private key leakage.

(3) **Confidentiality of ciphertexts**

According to the proposed Encrypt algorithm, the ciphertext ct generated by the DO includes parameters (C, C', $\{C_i\}$). However, only the parameter C is related to the plaintext msg. Since $C = msg \cdot e\ (g^a, g^s)$. Without the secret-sharing value s randomly chosen by the DO, any malicious attacker can only derive the original plaintext msg by guessing the correct s, and the success probability is only $(p-1)^{-1}$, which is negligible.

(4) **Confidentiality of searched keywords**

In the proposed trapdoor algorithm, the keywords sent by the DU will be encrypted by the PS, which utilizes the DU's KS-Key and an integer u chosen at random. Therefore, only the parameter $T_q(kw)$ of the trapdoor contains the plaintext information of the keyword kw searched by the DU. When the PS re-encrypts the parameter L'' of the trapdoor with its private key β, it learns nothing about the keyword searched by the DU. Moreover, a malicious TA only knows the KS-Key of the DU. Even if it colludes with the PS, they cannot successfully derive the keyword kw due to the unknown random number u chosen by the DU.

(5) **Unforgeability of trapdoors**

A legitimate trapdoor T'_{kw} generated by the DU includes ($T_q(kw)$, L', $\{K'_x\}_{x \in w}$), in which the parameter $T_q(kw)$ is generated by the KS-Key, the parameters L' and $\{K'_x\}_{x \in w}$ are both generated using a random number correlated to the KS-Key. In addition, the re-encrypted trapdoor created by the PS is $T_{kw} = (T_q(kw), L', \{K'_x\}_{x \in w})$, in which the parameter L' is computed with the PS's private key β. Therefore, without all of the above-mentioned information, no attackers could forge a legitimate trapdoor.

(6) **Resistance to malicious cloud servers**

Uploading sensitive data to the cloud server is risky. If the CS's behavior is malicious, the encrypted data and keywords uploaded by the user might be viewed. However, according to the proposed Encrypt, Re-Encrypt and Trapdoor algorithms, if a malicious CS lacks the information of correct user keys, the PS's private key and the utilized random number, it will encounter computational difficulty and fail to make it.

Since the proposed scheme is modified from Wang et al.'s mechanism [26], we will make a comparison of computational complexity. Table 3 shows the computational complexity analysis of the algorithms of the two systems. Note that we only consider the operations of bilinear pairing (T_{bp}) and exponential (T_{exp}), since they will take more computation time. From this table, it is evident that the Encrypt algorithm of our system is efficient than that of Wang et al.'s work. The ciphertext length of the proposed scheme is also shorter, which could benefit to the reduction in communication overheads and

cloud storage space. Although Wang et al.'s system has the lower computational complexity in the algorithm of Trapdoor, their system does not support the functionalities of re-encryption and pre-decryption.

Table 3. Comparison of computational costs.

Phase	Wang et al.	Ours
ABE-KeyGen	$(2 + x)T_{exp}$	$(2 + x)T_{exp}$
KS-KeyGen	$2T_{exp}$	$2T_{exp}$
Encrypt	$(2 + 3l)T_{exp}$	$(2 + 2l)T_{exp}$
Re-Encrypt	N.A.	T_{exp}
Index	$(1 + k)T_{exp} + kT_{bp}$	$(1 + k)T_{exp} + kT_{bp}$
Trapdoor	$(2 + x)T_{exp}$	$(3 + x)T_{exp}$
Test	$lT_{exp} + (2l + 1)T_{bp}$	$lT_{exp} + (2l + 1)T_{bp}$
Pre-Decrypt	N.A.	T_{exp}
Decrypt	$T_{exp} + T_{bp}$	$T_{exp} + T_{bp}$
Ciphertext length	$(2l + 1) \mid G_1 \mid + \mid G_2 \mid$	$(l + 1) \mid G_1 \mid + \mid G_2 \mid$

Remarks: x (size of attribute sets), l (number of shared users), k (number of keywords).

To make a practical simulation of the proposed system, we adopt the experimental results of [30] which utilized the hardware of Macbook Pro laptop with a 2.7 GHz Intel Core i5 processor and 8GB RAM. According to their results, shown in Table 4, the approximate running time of a bilinear pairing operation is about 10.243 milliseconds (ms), and that of an exponentiation computation is 1.266 ms.

Table 4. Approximate execution time of evaluated computation.

Item	Running Time
Bilinear pairing (T_{bp})	≈10.243 ms
Exponentiation (T_{exp})	≈1.266 ms

Figure 2 simulates the approximate running time of each participated party with respect to various $\{x, l, k\}$ combinations. From this figure, we can observe that the running time of the PS remains constant no matter how the combination changes. As for the DU, its running time is mainly affected by the size of attribute sets. The major factor to affect the running time of the CS is the number of shared users, since the CS must spend more time to perform the Test procedure. The TA is responsible for performing the ABE-KeyGen and KS-KeyGen algorithms, i.e., its running time would be affected by the size of attribute sets along with the number of shared users. The latter also affects the execution time of the DO when performing the encryption procedure. In addition, the Index algorithm carried out by the DO would also be influenced by the number of keywords.

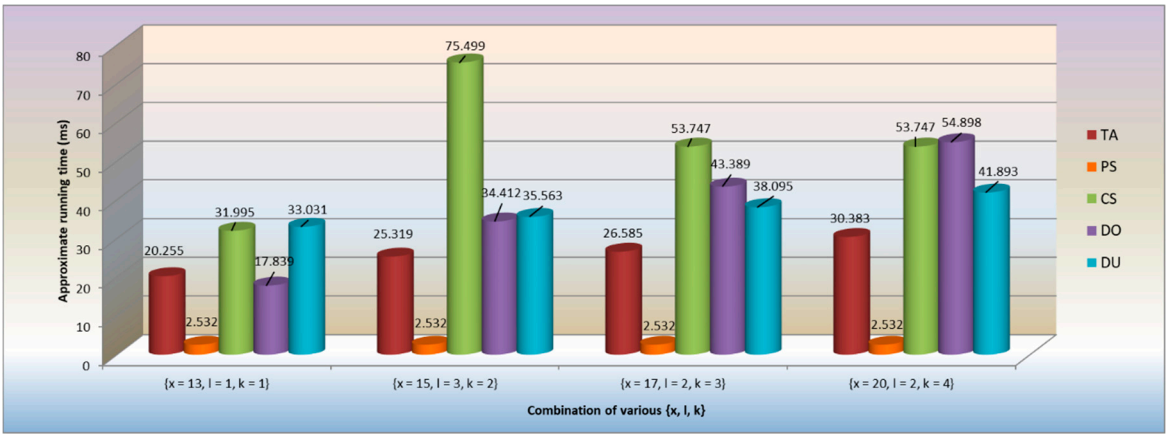

Figure 2. Comparison of various $\{x, l, k\}$ combinations.

5. Conclusions

The data-sharing of medical cloud systems is an important issue in the current world. How to ensure the security of health and medical data kept in cloud servers has become the critical topic of many researchers. Using attribute-based encryption mechanisms could obtain one-to-many data-sharing and fine-grained access control. In this work, the authors came up with an efficient multi-user, ciphertext policy attribute-based encryption scheme with keyword search for medical cloud systems. More precisely, we improved Wang et al.'s system by decreasing the computational complexity of Encrypt algorithm and further introduced the role of proxy server to not only release the load of the cloud server, but also protect the cloud server from direct attacks. The Re-Encrypt and Pre-Decrypt processes conducted by the proxy server only take one exponentiation computation. We believe that it would be a worthy tradeoff to obtain a better security level. Compared with Wang et al.'s system, the ciphertext length of our system is also shorter, which aids in the savings of communication costs.

Author Contributions: H.-Y.L. wrote the original draft. Y.-R.J. made the performance evaluation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology of Republic of China under the contract number MOST 109-2221-E-019-052.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Ethical Approval: This article does not contain any studies with human participants or animals performed by the author.

References

1. Srinivasan, U.; Datta, G.; Hons, M.S.; Hons, B.E. Personal health record (PHR) in a talisman: An approach to providing continuity of care in developing countries using existing social customs. In Proceedings of the 2007 9th International Conference on e-Health Networking, Application and Services, Taipei, Taiwan, 19–22 June 2007; pp. 277–279.
2. Puustjärvi, J.; Puustjärvi, L. Exploiting personal health records in automating information therapy. In Proceedings of the 2010 Second International Conference on eHealth, Telemedicine, and Social Medicine, St. Maarten, The Netherlands, 10–16 February 2010; pp. 100–105.
3. Azhagiri, M.; Amrita, R.; Aparna, R.; Jashmitha, B. Secured electronic health record management system. In Proceedings of the 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 15–16 October 2018; pp. 915–919.
4. Zhu, H.; Hou, M. Research on an electronic medical record system based on the Internet. In Proceedings of the 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, 21–23 September 2018; pp. 537–540.
5. Yang, W.; Chen, J.; Chen, Y. An electronic medical record management system based on smart contracts. In Proceedings of the 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media), Bali, Indonesia, 6–9 August 2019; pp. 220–223.
6. Blaze, M.; Bleumer, G.; Strauss, M. Divertible Protocols and Atomic Proxy Cryptography. In *Advances in Cryptology—EUROCRYPT'98*; Springer: Berlin/Heidelberg, Germnay, 1998; pp. 127–144.

7. Ateniese, G.; Fu, K.; Green, M.; Hohenberger, S. Improved proxy re-encryption schemes with applications to secure distributed storage. In Proceedings of the 10th Network and Distributed System Security Symposium (NDSS'05), San Diego, CA, USA, 10–13 February 2005; pp. 29–43.
8. Canetti, R.; Hohenberger, S. Chosen-ciphertext secure proxy re-encryption. In Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS 2007), Alexandria, VA, USA, 29 October–2 November 2007; pp. 185–194.
9. Wang, X.A.; Xhafa, F.; Zheng, Z.; Nie, J. Identity based proxy re-encryption scheme (IBPRE+) for secure cloud data sharing. In Proceedings of the 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), Ostrava, Czech Republic, 7–9 September 2016; pp. 44–48.
10. Zeng, P.; Choo, K.R. A new kind of conditional proxy re-encryption for secure cloud storage. *IEEE Access* **2018**, *6*, 70017–70024. [CrossRef]
11. Talmy, A.; Dobzinski, O. Abuse freedom in access control schemes. In Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA 2006), Vienna, Austria, 18–20 April 2006; pp. 77–86.
12. Ateniese, G.; Fu, K.; Green, M.; Hohenberger, S. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.* **2006**, *9*, 1–30. [CrossRef]
13. Chiu, Y.P.; Lei, C.L.; Huang, C.Y. Secure multicast using proxy encryption. In Proceedings of the 7th International Conference on Information and Communications Security (ICICS), Beijing, China, 10–13 December 2005; pp. 280–290.
14. Khurana, H.; Hahm, H.S. Certified mailing lists. In Proceedings of the ACM Symposium on Communication, Information, Computer and Communication Security (ASIACCS'06), Taipei, Taiwan, 21 March 2006; pp. 46–58.
15. Khurana, H.; Slagell, A.; Bonilla, R. Sels: A secure e-mail list service. In Proceedings of the ACM Symposium on Applied Computing (SAC'05), Santa Fe, NM, USA, 13–17 March 2005; pp. 306–313.
16. Taban, G.; Cárdenas, A.A.; Gligor, V.D. Towards a secure and interoperable DRM architecture. In Proceedings of the 6th ACM Workshop on Digital Rights Management, Alexandria, VA, USA, 30 October–3 November 2006; pp. 69–78.
17. Boneh, D.; Crescenzo, G.D.; Ostrovsky, R.; Persiano, G. Public key encryption with keyword search. In *Advances in Cryptology—EUROCRYPT 2004*; Springer: Berlin/Heidelberg, Germnay, 2004; pp. 506–522.
18. Cui, X.H.; Wan, Z.; Deng, R.; Wang, G.; Li, Y. Efficient and expressive keyword search over encrypted data in cloud. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 409–422. [CrossRef]
19. Chen, R.; Mu, Y.; Yang, G.; Guo, F.; Huang, X.; Wang, X.; Wang, Y. Server-aided public key encryption with keyword search. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2833–2842. [CrossRef]
20. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 457–473.
21. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute based encryption. In Proceedings of the IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
22. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS'06), Taipei, Taiwan, 22 March 2006; pp. 89–98.
23. Lin, G.; Hong, H.; Sun, Z. A collaborative key management protocol in ciphertext policy attribute-based encryption for cloud data sharing. *IEEE Access* **2017**, *5*, 9464–9475. [CrossRef]
24. Sethia, D.; Shakya, A.; Aggarwal, R.; Bhayana, S. Constant size CP-ABE with Scalable revocation for resource-constrained IoT devices. In Proceedings of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York City, NY, USA, 10–12 October 2019; pp. 0951–0957.
25. Zhou, S.; Chen, G.; Huang, G.; Shi, J.; Kong, T. Research on multi-authority CP-ABE access control model in multicloud. *China Commun.* **2020**, *17*, 220–233. [CrossRef]
26. Wang, C.; Li, W.; Li, Y.; Xu, X. A ciphertext-policy attribute-based encryption scheme supporting keyword search function. In Proceedings of the International Symposium on Cyberspace Safety and Security (CSS 2013), Zhangjiajie, China, 13–15 November 2013; pp. 377–386.
27. Barreto, P.; Naehrig, M. Pairing-friendly elliptic curves of prime order. In Proceedings of the 12th International Workshop on Selected Areas in Cryptography, Kingston, ON, Canada, 11–12 August 2005; pp. 319–331.
28. Miller, V. The weil pairing and its efficient calculation. *J. Cryptol.* **2004**, *17*, 235–261. [CrossRef]
29. Zhao, J.; Gao, H. LSSS matrix-based attribute-based encryption on lattices. In Proceedings of the 2017 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, China, 15–18 December 2017; pp. 253–257.
30. Agrawal, S.; Chase, M. FAME: FAST attribute-based message encryption. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CSS'17), Dallas, TX, USA, 30 October–3 November 2017; pp. 665–682.