sensors

MDPI

Article

An Improved ID-Based Data Storage Scheme for Fog-Enabled IoT Environments

Han-Yu Lin *, Tung-Tso Tsai, Pei-Yih Ting and Ching-Chung Chen

Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan; tttsai@mail.ntou.edu.tw (T.-T.T.); pyting@mail.ntou.edu.tw (P.-Y.T.); 10857027@email.ntou.edu.tw (C.-C.C.)
* Correspondence: hanyu@mail.ntou.edu.tw

Abstract: In a fog-enabled IoT environment, a fog node is regarded as the proxy between end users and cloud servers to reduce the latency of data transmission, so as to fulfill the requirement of more real-time applications. A data storage scheme utilizing fog computing architecture allows a user to share cloud data with other users via the assistance of fog nodes. In particular, a fog node obtaining a re-encryption key of the data owner is able to convert a cloud ciphertext into the one which is decryptable by another designated user. In such a scheme, a proxy should not learn any information about the plaintext during the transmission and re-encryption processes. In 2020, an ID-based data storage scheme utilizing anonymous key generation in fog computing was proposed by some researchers. Although their protocol is provably secure in a proof model of random oracles, we will point out that there are some security flaws inherited in their protocol. On the basis of their work, we further present an improved variant, which not only eliminates their security weaknesses, but also preserves the functionalities of anonymous key generation and user revocation mechanism. Additionally, under the Decisional Bilinear Diffie–Hellman (DBDH) assumption, we demonstrate that our enhanced construction is also provably secure in the security notion of IND-PrID-CPA.

Keywords: ID-based; data storage; proxy re-encryption; fog computing; IoT

Citation: Lin, H.-Y.; Tsai, T.-T.; Ting, P.-Y.; Chen, C.-C. An Improved ID-Based Data Storage Scheme for Fog-Enabled IoT Environments. *Sensors* **2022**, *22*, 4223. https://doi.org/10.3390/s22114223

Academic Editor: Raffaele Bruno

Received: 5 May 2022
Accepted: 30 May 2022
Published: 1 June 2022

1. Introduction

According to the concept of cloud computing addressed by computer scientist John McCarthy [1] in 1992, the computing capability of computers will someday become a kind of public utility like telephone systems. Essentially, the cloud computing is an innovative computing concept, rather than a brand-new technique. It utilizes the network to provide the service of data computing, transmitting and sharing. Moreover, it allows lots of computers to simultaneously share the same computing task for not only improving the efficiency, but also solving the plight of insufficient hardware resources in a single computer. From the perspective of end users, they only need to focus on the required resources and service types. Generally speaking, there are three types of service models described as follows:

(i) Software as a Service (SaaS): Users can utilize the browser of information devices such as computers, cell phones, tablets and so on to access the resources and services of cloud providers and execute the required software and applications in highly malleable cloud infrastructures.
(ii) Platform as a Service (PaaS): The cloud service provider offers the platform for application development and the supported programming language with development tools so that users can deploy or purchase the required application services by themselves.
(iii) Infrastructure as a Service (IaaS): In the cloud infrastructure, cloud service providers offer all kinds of resources, including network, storage, analysis and computing, etc., so that users process tasks as if they were on the local machine without maintaining and managing the backend hardware structure.

Though the notion of cloud computing has greatly changed the traditional way of information utilization, sharing and storage, its high latency caused by the Internet transmission and the centralized processing burden of cloud systems are still major challenges in current IoT-enabled cloud applications. Owing to this reason, the model of fog computing has thus come up. It can be regarded as a technique expanded from the cloud computing. Moreover, it is more like a computing mode that is close to the end-users. Therefore, we could say that a fog is a kind of cloud approaching the ground. As compared with clouds, though fogs have a less powerful computing power, they could reduce response time, gain more energy savings and decrease the utilization of bandwidths. So far, fog computing approaches have been used in harsh operational environments such as shipping [2] and aviation [3].

In a fog-enabled IoT environment, fog nodes are core components that could be either physical devices or virtual equipment, and tightly coupled with intelligent terminals or the access network to provide computing resources. These fog nodes can forward received data to clouds and help with the downloading of user data. Like cloud databases, the fog layer has its own data storage and maintains the local database. Although the architecture of fog computing extended from that of cloud computing can increase mobility and reduce transmission latency, it is still vulnerable to many security threats summarized by Patwary et al. [4] in Figure 1.

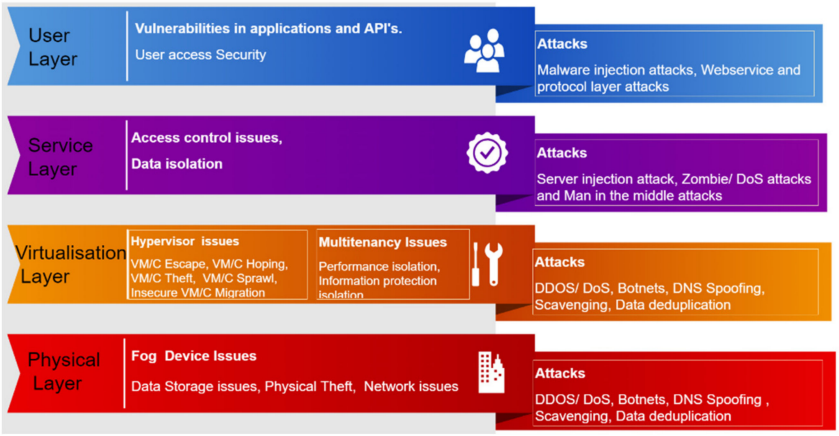

Figure 1. Security threats of fog computing [4].

When it comes to sharing confidential data in fog computing, the proxy re-encryption (abbreviated to PRE) scheme addressed by Blaze et al. [5] is a relevant alternative, since it can maintain data confidentiality during the transformation of ciphertexts. Specifically, a data owner can first encrypt the data and then upload the ciphertext to clouds for future access. As the data are transmitted and stored in the encrypted form, anyone including the honest-but-curious cloud server is unable to decrypt it without knowing the corresponding private key. When a data owner attempts to grant another user the access right of his cloud ciphertext, he could authorize a semi-trusted fog node (viewed as the proxy) to perform the procedure of ciphertext re-encryption. In such a way, an original cloud ciphertext is converted into the one that is decryptable by the designated data user. A major advantage of the technique of PREs is that the ciphertext remains undecrypted during the conversion process. Consequently, the proxy will learn nothing about the ciphertext. Up to the present, it has been found in many of the PRE applications [5–8] such as data sharing, data outsourcing, data storage in clouds, e-mail forwarding, etc.

1.1. Related Works

Under the Decisional Bilinear Diffie–Hellman (DBDH) assumption, in 2005, Ateniese et al. [6,7] presented an improved PRE scheme following the work of Dodis and Ivan [9]. They demonstrated that PRE schemes are useful in the access control of secure file

systems and could be realized efficiently in practice. In 2007, Caneti and Hohenberger [10] proposed the definition of chosen-ciphertext attacks (CCA) for PRE schemes and gave a concrete construction to satisfy the definition in the standard model. The definition that they introduced includes both game-based and simulation-based ones. The underlying security of their scheme is also the DBDH assumption.

Seeing that previous PRE schemes mainly convert the ciphertext from one public key to another, Green and Ateniese [11] introduced identity-based PRE (abbreviated to IB-PRE) schemes to deal with the problem of transforming the ciphertext from one identity to another. In addition, their schemes are non-interactive and provably secure in the random oracle models. One of their works also exhibits the property of multi-hop, i.e., a ciphertext could be converted from one identity to another more than once, which gives the practical applications more feasibility. Using standard model proofs, Chu and Tzeng [12] presented two identity-based PRE mechanisms. They showed that their first scheme has better efficiency in computational costs and the ciphertext length while the other one achieves CCA security. The two mechanisms are unidirectional and non-interactive. Nevertheless, in 2009, Shao and Cao [13] pointed out that the Chu-Tzeng scheme is not truly CCA-secure, as its re-encrypted ciphertext could be further converted into a well-formed ciphertext. Using the Decisional Diffie–Hellman (DDH) assumption and the integer factorization assumption, they also addressed a new PRE scheme that could withstand both the chosen-ciphertext attack and the collusion attack in the random oracle models.

In 2012, Fang et al. [14] presented the so-called conditional PRE with keyword search (C-PRES), which is a combination of conditional PRE (C-PRE) and public key encryption with keyword search (PEKS). They defined the security of chosen-ciphertext attacks for C-PRES and proved that their construction fulfills this definition. Wang et al. [15] also devoted their attention to the research of PRE with keyword search and hence introduced a new primitive named constrained single-hop unidirectional PRE supporting conjunctive keywords search (CPRE-CKS). Based on Park et al.'s work [16], Wang et al.'s scheme only allows the ciphertext containing the corresponding keywords to be re-encrypted by a proxy. Under the decisional q-parallel bilinear Diffie–Hellman exponent assumption, in 2013, Liang et al. [17] extended the traditional PRE into the CP-ABPRE, i.e., ciphertext policy-attribute-based PRE. In such a protocol, a proxy has the ability to re-encrypt the ciphertext from one access policy to another. Their protocol can be applied to any monotonic access structure and is proved secure in the random oracle models.

Akhil et al. [18] employed the technique of PRE to enhance the security of QR codes in 2016. A QR code is a machine-readable format that could be tampered with maliciously when being transmitted. Applying the technique of PREs to QR codes makes it easy and flexible to be shared and stored among different hosts. In 2018, Zeng and Choo [19] proposed a new kind of conditional PRE (C-PRE) scheme called sender-specified PRE (SS-PRE) in which a proxy can only convert the ciphertext from a specified sender to his/her delegatee. They also demonstrated that their SS-PRE scheme outperforms the conventional C-PRE ones.

Considering the communication security between the fog and the cloud, in 2018, Vohra and Dave [20] proposed an attribute-based access control and re-encryption system composed of two phases. In the first phase, the clouds will communicate with the proxy server and transmit encrypted data, which are then decrypted by the latter according to its attribute set and access policy. In the second phase, the proxy server will broadcast re-encrypted ciphertext to all fog nodes. Only the fog node that has the correct attributes can decrypt the ciphertext.

In 2020, Lian et al. [21] introduced a PRE model along with a concrete scheme that is suitable for complex access control factor description in hybrid clouds. A hybrid cloud has not only the advantage of more powerful computing resources in public clouds, but also that of easy management in private clouds. They showed that their construction is secure under the DBDH assumption and could be reliably deployed in hybrid clouds.

In 2021, Xiong et al. [22] proposed an adaptively secure puncturable identity-based PRE scheme for securing group messages. In their work, a message server is responsible for converting the ciphertext for each user, and thus the heavy computation load could be shifted from the user to the message server. They also prove the security of their system under the DBDH assumption. However, the message server would easily become the performance bottleneck, and the centralized architecture is not suitable for distributed application environments.

Considering the data sharing in clouds, Ge et al. [23] presented a verifiable and fair attribute-based PRE scheme in which the user is able to verify the correctness of the ciphertext converted by the cloud server. Moreover, the latter is also capable of claiming its honesty when being maliciously accused by the former. They conduct experiments to demonstrate the feasibility and the efficiency of their system in the realistic environments. Nevertheless, their mechanism does not deal with the revocation issue of either attributes or the user identity.

Recently, Zhang et al. [24] applied the technique of PRE to propose an ID-based data storage (DS) system utilizing anonymous key generation for the fog computing environment. A DS scheme for fog-enabled environments is a kind of data sharing technique utilizing fog nodes as the proxy to reduce response time and communication overheads. Specifically, the fog nodes can process the data gathered by the IoT sensors and forward them to the cloud. The fog node serving as a proxy between the cloud and the endpoints is able to transform the cloud ciphertext into another one, which is decryptable by the requested user, so as to achieve the purpose of data sharing in clouds. Using anonymous key generation, a malicious private key Generation (PKG) center is unable to learn the genuine private key of users. It has been formally proved that their protocol is indistinguishable against adaptively chosen identity and chosen plaintext attacks (abbreviated to IND-PrID-CPA) in random oracle models and secure against the PKG and the collusion attacks. However, in this work, we will show that their system has several security flaws. So far, lots of PRE-related cryptographic protocols [25–36] have been proposed.

1.2. Contributions

Since a secure DS scheme in fog-enabled IoT environments is the key to ensure data confidentiality and user privacy, we devote ourselves to the enhancement of current DS schemes.

In 2020, Zhang et al. [24] proposed a novel DS scheme supporting anonymous key generation, which is unnecessary to rely on a secure channel. Unfortunately, their system is vulnerable to several attacks. Motivated by Zhang et al.'s work [24], we present an improved DS scheme in the fog-enabled IoT environments. The novelty of our work is that we view fog nodes as semi-trusted entities rather than fully trusted ones in our system. Moreover, we introduce an additional random value in generating the proxy re-encryption keys, so as to prevent unauthorized decryption. The partial private key information will not be compromised during communication, which gains more protection of confidentiality in practical environments. Our work can strengthen the application security in fog-enabled environments. In particular, telemedicine has received much attention in recent years. The data confidentiality and user privacy are the most concerned. Improving existing schemes to withstand possible attacks is of utmost importance. The major contributions of this research are itemized below:

(i) We demonstrate some security vulnerabilities in relation to the illegal access privilege of Zhang et al.'s scheme [24], including the proxy attack, the outsider attack and the revoked user attack.

(ii) An enhanced DS variant on the basis of Zhang et al.'s system [24] is further proposed. In particular, we introduce an additional random value in the re-encryption key generation and modify the decryption algorithm.

(iii) We formally prove that the proposed construction is indistinguishable against adaptively chosen identity and chosen plaintext attacks (IND-PrID-CPA) in random oracle models.
(iv) The superior characteristics of anonymous key generation and user revocation are preserved in the proposed improvement.
(v) The confidentiality of partial private key information is well-protected during communication since it does not need to be transmitted with the communication data.

The organization of this paper is described as follows. In Section 2, the computational background and cryptographic assumptions are introduced. We review and analyze Zhang et al.'s protocol [24] in Section 3. A corresponding improvement is also presented. In Section 4, using random oracle models, we define the security notion of IND-PrID-CPA and formally prove the security of our improved variant. Finally, a conclusion is summarized in Section 5.

2. Preliminaries

We describe the property of bilinear pairing and a related computational assumption, which the proposed scheme is based on in this section.

Definition of Bilinear Pairing

We let both notions of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ be multiplicative groups and they have the same prime order p. A symmetric bilinear pairing is defined as e: $\boldsymbol{G}_1 \times \boldsymbol{G}_1 \rightarrow \boldsymbol{G}_2$. Some characteristics of bilinear pairings are stated below:

(i) Bilinearity

Letting P be an element of $\boldsymbol{G}_1$ and x, y be arbitrary integers of Z_p, the equality $e(P^x, P^y) = e(P, P)^{xy}$ will hold.

(ii) Non-degeneracy

There exists P, $W \in \boldsymbol{G}_1{}^2$, satisfying the inequality $e(P, W) \neq 1$.

(iii) Computability

There exists an algorithm that could efficiently compute $e(P, W)$, where P, $W \in \boldsymbol{G}_1{}^2$.

Decisional Bilinear Diffie-Hellman (DBDH) Problem

Given elements $(g, g^f, g^s, g^k, e(g, g)^{fsk}, \delta)$, where $g, g^f, g^s, g^k \in \boldsymbol{G}_1{}^4$ and $e(g, g)^{fsk}$, $\delta \in \boldsymbol{G}_2{}^2$, the DBDH problem is to determine whether $e(g, g)^{fsk}$ equals to δ or not.

Decisional Bilinear Diffie-Hellman (DBDH) Assumption

The DBDH assumption holds provided that the advantage for arbitrary probabilistic adversary running in polynomial time, and breaking the DBDH problem is negligible.

3. Proposed ID-Based Data Storage Scheme

We first formalize the algorithms of ID-based data storage (abbreviated as IB-DS) schemes and then review Zhang et al.'s work [24]. Several security weaknesses of their scheme will be demonstrated, and a corresponding enhanced variant shall be introduced later.

3.1. System Architecture

The system architecture of the IB-DS scheme is illustrated in Figure 2, which could be divided into three layers, i.e., the cloud, the fog and the user layers. There is also a trusted authority called the private key generation center (PKG), which is responsible for generating the private key of all involved entities. The cloud server of the cloud layer will store encrypted data gathered from the user layer. A data requester of the user layer can request the data access of the cloud ciphertext by the assistance of the fog layer. The fog nodes comprising the fog layer are viewed as a proxy between the cloud layer and the user layer. Whenever a data owner authorizes the access privilege of a cloud ciphertext to

another data user, the proxy (fog) would be granted a re-encryption key, which is able to transform the target cloud ciphertext into one decryptable by the desired data user.

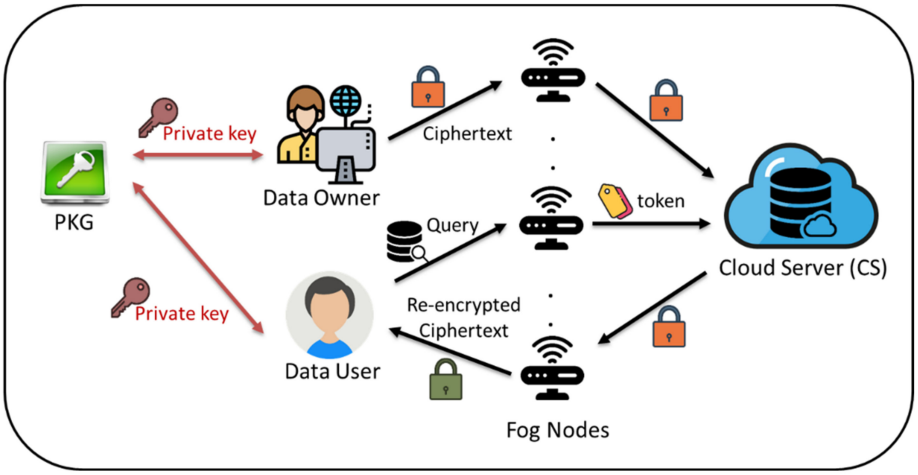

Figure 2. System architecture of IB-DS scheme.

3.2. Algorithms

An IB-DS scheme consists of seven algorithms including Setup, Keygen, Encrypt, Query, Permission, Re-encrypt and Decrypt. The definitions of the above algorithms are described below:

- **Setup**(1^l): It accepts a security value l and then generates system public parameters PP and a master secret key Msk.
- **Keygen**(PP, Msk, ID): It takes the input of public parameters PP, the master secret key Msk and a user identity ID, and then outputs the private key SK_{ID} for the user ID via an interactive procedure.
- **Encrypt**(PP, ID, m, Y): This algorithm inputs system public parameters PP, a user identity ID, a message m and a symmetric key Y, and then outputs the ciphertext C of the message m.
- **Query**(PP, ID_u, SK_{IDu}, M_{cate}): It takes the input of system public parameters PP, a data user identity ID_u, a private key SK_{IDu} and a data category name M_{cate}, and then outputs a corresponding query token TK.
- **Permission**(PP, ID_u, SK_{IDo}, TK): It takes the input of system public parameters PP, a data user identity ID_u, the private key SK_{IDo} of the data owner and a query token TK, and then outputs either an invalid symbol $\perp$ or a re-encryption key RK.
- **Re-encrypt**(PP, ID_u, C, RK): It takes the input of system public parameters PP, a data user identity ID_u, a ciphertext C and a re-encryption key RK, and then outputs a corresponding re-encrypted ciphertext C'.
- **Decrypt**(PP, SK_{ID}, C or C'): It takes the input of system public parameters PP, a private key SK_{ID} and a ciphertext C (or C'), and then outputs a decrypted message m.

We summarize the input and the output parameters of each algorithm in Table 1.

Table 1. The input and output parameters of composed algorithms.

Algorithm \ Parameter	Input	Output
Setup	l	PP, Msk
Keygen	PP, Msk, ID	SK_{ID}
Encrypt	PP, ID, m, Y	C
Query	PP, ID_u, SK_{IDu}, M_{cate}	TK or $\perp$
Permission	PP, ID_u, SK_{IDo}, TK	$\perp$ or RK
Re-encrypt	PP, ID_u, C, RK	C'
Decrypt	PP, SK_{ID}, C or C'	m

3.3. Review and Security Analysis

This subsection reviews an IB-DS scheme proposed by Zhang et al. [24] in 2020. Although their protocol is provably secure in the random oracle models, there are still some security drawbacks, which will be pointed out later. The construction of their scheme is described below:

- **Setup:** Using a security value l, the PKG first decides two multiplicative groups $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$. Let p be the prime order of both groups and g a generator of $\boldsymbol{G}_1$. In the two groups, there is a symmetric pairing function e expressed as e: $\boldsymbol{G}_1 \times \boldsymbol{G}_1 \rightarrow \boldsymbol{G}_2$. The PKG then chooses integers a, $b \in Z_p^*$ as the Msk and computes the $Mpk = (P = g^a, Q = g^b)$. To maintain the membership of system users, the PKG also keeps a revocation list L. The system public parameter PP is composed of $\{\boldsymbol{G}_1, \boldsymbol{G}_2, e, g, p, Mpk, E(\cdot), D(\cdot), h_1, h_2\}$ where $E(\cdot)/D(\cdot)$ is a symmetric encryption/decryption function and (h_1, h_2) are two secure one-way hash functions that accept a variable-length input and generate a corresponding output in $\boldsymbol{G}_1$.
- **Keygen:** A user associated with the identity ID_i first chooses t_i, $z_i \in Z_p^*$ to compute

$$Z_i = g^{z_i}, \tag{1}$$

$$T'_i = Z_i \cdot h_1(ID_i \,||\, t_i), \tag{2}$$

and transmits (ID_i, T'_i) to the PKG who then chooses $d_i \in Z_p^*$ to compute

$$SK'_{i,1} = g^{ab}(T'_i \cdot h_2(ID_i \,||\, ID_{PKG}))^{d_i}, \tag{3}$$

$$SK'_{i,2} = g^{d_i}, \tag{4}$$

and delivers $(SK'_{i,1}, SK'_{i,2})$ to ID_i. In this way, ID_i could further set

$$SK_{i,1} = SK'_{i,1}/(SK'_{i,2})^{z_i} = g^{ab}(h_1(ID_i \,||\, t_i) \cdot h_2(ID_i \,||\, ID_{PKG}))^{d_i}, \tag{5}$$

$$SK_{i,2} = SK'_{i,2}. \tag{6}$$

Here, the full private key of ID_i is $SK_i = (SK_{i,1}, SK_{i,2})$. The correctness of the private key could be verified by the following equality:

$$e(SK_{i,1}, g) = e(P, Q)e(h_1(ID_i \,||\, t_i) \cdot h_2(ID_i \,||\, ID_{PKG}), SK_{i,2}). \tag{7}$$

- **Encrypt:** To encrypt the message $m = (m_1, m_2, \ldots, m_n)$, a data owner ID_o first selects $r \in_R Z_p^*$ and a symmetric key $Y \in \boldsymbol{G}_2$ to compute

$$\alpha = Y \cdot e(P, Q)^r, \tag{8}$$

$$\beta = g^r, \tag{9}$$

$$\theta = (h_1(ID_o \,||\, t_o) \cdot h_2(ID_o \,||\, ID_{PKG}))^r, \tag{10}$$

$$\tau = (E(Y, m_1), E(Y, m_2), \ldots, E(Y, m_n)). \tag{11}$$

Then, the ciphertext $C = (\alpha, \beta, \theta, \tau)$ along with (ID_o, M_{cate}), where M_{cate} represents the category name of data, are transmitted to the nearby fog (proxy), which will keep $(ID_o, M_{cate}, \alpha, \beta, \theta)$ in the local database of the fog layer and forward (ID_o, M_{cate}, τ) to the cloud server.

- **Query:** To request the data access of M_{cate}, a data user ID_u first chooses $w \in_R Z_p^*$ to compute

$$W = (SK_{u,1})^w, \tag{12}$$

and sends $(ID_u, M_{cate}, W, SK_{u,2})$ to the nearby proxy. Afterwards, the proxy utilizes M_{cate} to search for matched $(ID_o, M_{cate}, \alpha, \beta, \theta)$ in the local database and delivers the query token $TK = (ID_u, W, SK_{u,2}, \beta)$ to the corresponding data owner ID_o.

- **Permission:** When receiving the query token $TK = (ID_u, W, SK_{u,2}, \beta)$, the data owner sends $(ID_u, SK_{u,2})$ to the PKG, which will inspect whether ID_u is a revoked user or not according to its revocation list L and then return True/False to indicate that the membership of ID_u is valid/invalid. If False, the data owner submits an invalid symbol $\perp$ to the proxy. Otherwise, ID_o picks a random number $x \in Z_p^*$ to compute

$$RK_1 = (SK_{o,1})W^{-1}g^x, \tag{13}$$

$$RK_2 = \beta^x, \tag{14}$$

$$RK_3 = SK_{o,2}. \tag{15}$$

Then, the re-encryption key $RK = (RK_1, RK_2, RK_3)$ is transmitted to the proxy.

- **Re-encrypt:** Given the re-encryption key $RK = (RK_1, RK_2, RK_3)$, the proxy first uses the identity ID_o to retrieve τ from the cloud server and then computes

$$\alpha' = \alpha \cdot e(RK_2, g), \tag{16}$$

$$\eta = RK_1, \tag{17}$$

$$\rho = RK_3. \tag{18}$$

Finally, the resulting ciphertext $C' = (\alpha', \beta, \theta, \tau, \eta, \rho)$ would be returned to the requested data user ID_u.

- **Decrypt:** Given an original ciphertext $C = (\alpha, \beta, \theta, \tau)$, the data owner ID_o first computes

$$Y = \alpha \cdot \frac{e(SK_{o,2},\ \theta)}{e(SK_{o,1},\ \beta)} \tag{19}$$

and then recovers the message m as

$$m = (m_1, m_2, \ldots, m_n) = (D(Y, \tau_1), D(Y, \tau_2), \ldots, D(Y, \tau_n)). \tag{20}$$

We show that Equation (19) correctly derives the symmetric key Y. From the right-hand side of the equality, we have

$$\begin{aligned} \alpha \cdot \frac{e(SK_{o,2},\ \theta)}{e(SK_{o,1},\ \beta)} &= Ye(P,\ Q)^r \cdot \frac{e(g^{d_O},\ (h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^r)}{e(g^{ab}\,(h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^{d_O},\ g^r)} \\ &= Ye\left(g^{ab},\ g^r\right) \cdot \frac{e(g^{d_O},\ (h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^r)}{e(g^{ab}\,(h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^{d_O},\ g^r)} \\ &= Y \end{aligned}$$

When given a ciphertext $C' = (\alpha', \beta, \theta, \tau, \eta, \rho)$ of re-encrypted forms, ID_u computes a symmetric key

$$Y = \alpha' \cdot \frac{e(\theta,\ \rho)}{e\left((SK_{u,1})^w \eta,\ \beta\right)} \tag{21}$$

and then recovers the message m with Equation (20). The correctness of Equation (21) could be verified as follows. From the right-hand side of the equality, we find

$$\begin{aligned} \alpha' \cdot \frac{e(\theta,\ \rho)}{e((SK_{u,1})^w \eta,\ \beta)} &= Ye(P,\ Q)^r e(\beta^x,\ g) \cdot \frac{e(\,(h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^r,\ g^{d_O})}{e(SK_{u,1}{}^w (SK_{o,1})(SK_{u,1})^{-w} g^x,\ g^r)} \\ &= Ye\left(g^{ab},\ g^r\right) e(g^{rx},\ g) \cdot \frac{e(\,(h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^r,\ g^{d_O})}{e(g^{ab}\,(h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^{d_O} g^x,\ g^r)} \\ &= Ye\left(g^{ab},\ g^r\right) e(g^{rx},\ g) \cdot \frac{1}{e\left(g^{ab} g^x,\ g^r\right)} \\ &= Y \end{aligned}$$

Note that to revoke the membership of a user ID_i, the PKG will update its revocation list L as L' by adding the new entry $(ID_i, SK_{i,2})$, i.e., $L' = L \cup \{(ID_i, SK_{i,2})\}$. Unfortunately,

the authors find out that Zhang et al.'s scheme [24] has several security weaknesses stated as follows:

Weakness 1: A dishonest fog (proxy) is able to decrypt the ciphertext queried by a data user ID_u without having the knowledge of corresponding private key. According to Equation (12), the private key information $SK_{u,1}$ is further combined with a random integer w chosen by ID_u for computing the symmetric key Y. Although the dishonest proxy knows neither the private key $SK_{u,1}$ nor the secret w, it has obtained the combined value $W = (SK_{u,1})^w$ in the Query phase. Therefore, it can also successfully derive the symmetric key Y and decrypt the ciphertext queried by ID_u.

Weakness 2: An adversary is able to gain access to any cloud ciphertext without having the corresponding private key. More specifically, an adversary first randomly chooses $SK_{u,1}$, $w \in Z_p^*$ to compute $W' = (SK_{u,1})^w$ with respect to any M_{cate} he attempts to access. Since the adversary is not a revoked user in the revocation list L, he would receive a re-encrypted ciphertext. Then, based on the decryption equality, i.e., Equations (20) and (21), he could employ the value W' to recover the symmetric key Y and decrypt the received ciphertext, respectively.

Weakness 3: A revoked user can impersonate any legitimate user to gain access to any cloud ciphertext without having the corresponding private key. Assume that ID_u is a revoked user in the system. This means that the entry (ID_u, $SK_{u,2}$) has been stored in the revocation list L of the PKG. In order to request any ciphertext in the cloud server, ID_u could impersonate any non-revoked user, say ID_v, to issue a query. The procedure is similar to that mentioned in weakness 2. That is, he first randomly chooses $SK_{v,1}$, $w \in Z_p^*$ to compute $W'' = (SK_{v,1})^w$ in relation to any desired M_{cate}. As the impersonated identity ID_v is still a legitimate user, the attacker would receive a corresponding re-encrypted ciphertext, which is decryptable by his forged value W''.

Weakness 4: The partial information of the user's private key is compromised during communication. According to the procedures and data flows stated in the Query and the Permission phases, the partial private key $SK_{u,2}$ of the data user ID_u has to be transmitted via an open channel. This undoubtedly leaks the partial private key information out.

3.4. Construction of an Improved IB-DS Scheme

According to our cryptanalyses of Zhang et al.'s system [24], we find out that the private key of the requested data user is not properly hidden in the query algorithm, which makes the secret parameter able to be nullified by any malicious entity in the decryption process. Moreover, the decryption equation, i.e., Equation (21), does not integrate with the second private key of the data user, which is also a major problem that has led to previous attacks. To eliminate the security weaknesses of Zhang et al.'s scheme [24], the authors come up with an improved variant without modifying the system architecture and involved parties. In the Setup algorithm of our system, we additionally introduce a new hash function, i.e., h_3: $\boldsymbol{G}_2 \rightarrow \boldsymbol{G}_1$. Since the processes of Setup, Keygen and Encrypt algorithms are defined the same as those of Zhang et al.'s scheme [24], we formalize them as the following Algorithms 1–3:

Algorithm 1. Setup.

Input: A security value l
Output: PP, Msk
1: Decide groups ($\boldsymbol{G}_1$, $\boldsymbol{G}_2$)
2: Choose the prime order p and a generator g
3: Choose appropriate hash functions h_1, h_2 and h_3
4: Choose a symmetric encryption/decryption function $E(\cdot)/D(\cdot)$
5: Define the pairing function e: $\boldsymbol{G}_1 \times \boldsymbol{G}_1 \rightarrow \boldsymbol{G}_2$
6: $(a, b) \leftarrow Z_p{}^*$
7: $P = g^a$
8: $Q = g^b$
9: $Msk = (a, b)$
10: $Mpk = (P, Q)$
11: $PP = \{\boldsymbol{G}_1, \boldsymbol{G}_2, e, g, p, Mpk, E(\cdot), D(\cdot), h_1, h_2, h_3\}$
12: **return** (PP, Msk);

Algorithm 2. Keygen.

Input: PP, Msk, ID_i
Output: The full private key SK_i
1: $(t_i, z_i) \leftarrow Z_p{}^*$
2: $Z_i = g^{z_i}$
3: $T'_i = Z_i \cdot h_1(ID_i \,||\, t_i)$
4: $d_i \leftarrow Z_p{}^*$
5: $SK'_{i,1} = g^{ab}(T'_i \cdot h_2(ID_i \,||\, ID_{PKG}))^{d_i}$
6: $SK'_{i,2} = g^{d_i}$
7: $SK_{i,1} = SK'_{i,1}/(SK'_{i,2})^{z_i}$
8: $SK_{i,2} = SK'_{i,2}$
9: $SK_i = (SK_{i,1}, SK_{i,2})$
10: **return** SK_i;

Algorithm 3. Encrypt.

Input: PP, ID, m, Y
Output: A ciphertext C
1: $r \leftarrow Z_p{}^*$
2: $Y \leftarrow \boldsymbol{G}_2$
3: $\alpha = Y \cdot e(P, Q)^r$
4: $\beta = g^r$
5: $\theta = (h_1(ID_o \,||\, t_o) \cdot h_2(ID_o \,||\, ID_{PKG}))^r$
6: $\tau = (E(Y, m_1), E(Y, m_2), \ldots, E(Y, m_n))$
7: $C = (\alpha, \beta, \theta, \tau)$
8: Store $(ID_o, M_{cate}, \alpha, \beta, \theta)$ in the local database.
9: Send (ID_o, M_{cate}, τ) to the cloud server.
10: **return** C;

- **Query:** To request the data access of M_{cate}, a data user ID_u first chooses $w \in_R Z_p{}^*$ to compute

$$W = g^w, \tag{22}$$

and sends (ID_u, M_{cate}, W) to the nearby proxy. Afterwards, the proxy utilizes M_{cate} to search for matched $(ID_o, M_{cate}, \alpha, \beta, \theta)$ in the local database and delivers the query token $TK = (ID_u, W, \beta)$ to the corresponding data owner ID_o. The query processes are presented in Algorithm 4.

Algorithm 4. Query.

Input: PP, ID_u, SK_{IDu}, M_{cate}
Output: A query token TK or $\perp$
$w \leftarrow Z_p{}^*$
$W = (SK_{u,1})^w$
Send (ID_u, M_{cate}, W, $SK_{u,2}$) to the proxy.
if $M_{cate} = M_{cate}$ of the local database **then**
  $TK = (ID_u, W, SK_{u,2}, \beta)$
  return TK;
else
  return $\perp$;
end if

- **Permission:** When receiving the query token $TK = (ID_u, W, \beta)$, the data owner sends ID_u to the PKG, which will inspect whether ID_u is a revoked user or not according to its revocation list L and then return True/False to indicate that the membership of ID_u is valid/invalid. If False, the data owner submits an invalid symbol $\perp$ to the proxy. Otherwise, ID_o picks two random numbers x, $\pi \in Z_p{}^*$ to compute (RK_2, RK_3) as Equations (14) and (15), and (RK_1, RK_4) as

$$\mathrm{RK}_1 = \frac{SK_{o,1} g^x}{h_3(e(PW^\pi, Q))} \tag{23}$$

$$RK_4 = e(g, Q^\pi) \tag{24}$$

Then, $RK = (RK_1, RK_2, RK_3, RK_4)$ is the generated re-encryption key, which will be transmitted to the proxy. The permission processes are presented in Algorithm 5.

Algorithm 5. Permission.

Input: PP, ID_u, SK_{IDo}, TK
Output: $\perp$ or RK
if ID_u is revoked **then**
  return $\perp$;
else
  $x, \pi \in Z_p{}^*$
  $RK_1 = \frac{SK_{o,1} g^x}{h_3(e(PW^\pi, Q))}$
  $RK_2 = \beta^x$
  $RK_3 = SK_{o,2}$
  $RK_4 = e(g, Q^\pi)$
  $RK = (RK_1, RK_2, RK_3, RK_4)$
  return RK;
end if

- **Re-encrypt:** Given $RK = (RK_1, RK_2, RK_3, RK_4)$, the proxy first uses the identity ID_o to retrieve τ from the cloud server and then computes (α', β, θ, η, ρ) as Equations (9), (10), and (16)–(18), and Φ as

$$\Phi = RK_4. \tag{25}$$

Finally, the re-encrypted ciphertext $C' = (\alpha', \beta, \theta, \tau, \eta, \rho, \Phi)$ is returned to the data user ID_u. We illustrate the flow chart of query, permission and re-encryption algorithms in Figure 3. The re-encryption processes are presented in Algorithm 6.

Algorithm 6. Re-Encrypt.

Input: PP, ID_u, C, $RK = (RK_1, RK_2, RK_3, RK_4)$
Output: A re-encrypted ciphertext C'
1: $\alpha' = \alpha \cdot e(RK_2, g)$
2: $\beta = g^r$
3: $\theta = (h_1(ID_o \,||\, t_o) \cdot h_2(ID_o \,||\, ID_{PKG}))^r$
4: $\eta = RK_1$
5: $\rho = RK_3$
6: $\Phi = RK_4$
7: $C' = (\alpha', \beta, \theta, \tau, \eta, \rho, \Phi)$
8: **return** C';

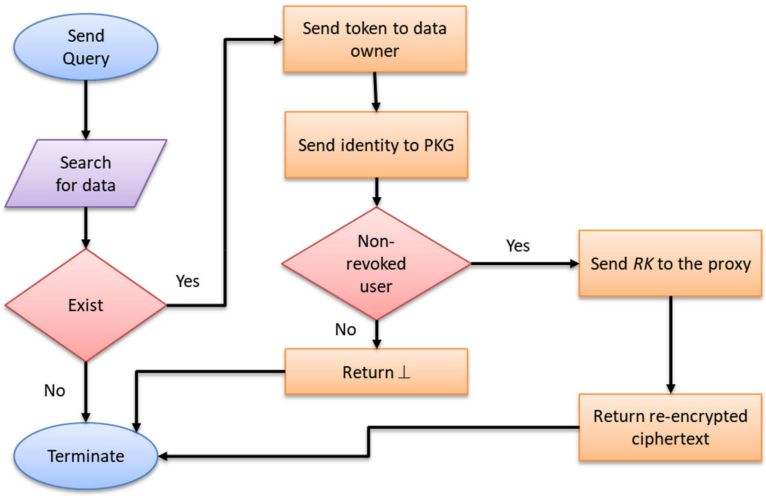

Figure 3. The flow chart of query, permission and re-encryption algorithms.

- **Decrypt:** Given an original ciphertext $C = (\alpha, \beta, \theta, \tau)$, the data owner ID_o first computes Y as Equation (19) and then recovers the message $m = (m_1, m_2, \ldots, m_n) = (D(Y, \tau_1), D(Y, \tau_2), \ldots, D(Y, \tau_n))$ by Equation (20). Still, when given a re-encrypted ciphertext $C' = (\alpha', \beta, \theta, \tau, \eta, \rho, \Phi)$, the data user ID_u first computes I and the symmetric key Y separately as

$$\begin{aligned} I &= \eta \cdot h_3\left(\frac{\Phi^w e(SK_{u,1}, g)}{e(h_1(ID_u||t_u)h_2(ID_u||ID_{PKG}), SK_{u,2})}\right) \\ &= \frac{SK_{o,1}g^x}{h_3(e(PW^\pi, Q))} h_3\left(\frac{\Phi^w e(P, Q) e(h_1(ID_u||t_u)h_2(ID_u||ID_{PKG}), SK_{u,2})}{e(h_1(ID_u||t_u)h_2(ID_u||ID_{PKG}), SK_{u,2})}\right) \\ &= \frac{SK_{o,1}g^x}{h_3(e(PW^\pi, Q))} h_3\left(e(g, Q^\pi)^w e(P, Q)\right) \\ &= SK_{o,1}g^x \end{aligned} \tag{26}$$

$$Y = \alpha' \cdot \frac{e(\theta, \rho)}{e(I, \beta)} \tag{27}$$

and then recovers the message m with Equation (20). The correctness of Equation (27) could be verified as follows. From the right-hand side of the equality, we find

$$\begin{aligned} &\alpha' \cdot \frac{e(\theta, \rho)}{e(I, \beta)} \\ &= Ye(P, Q)^r e(\beta^x, g) \cdot \frac{e((h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^r, g^{d_O})}{e((SK_{o,1})g^x, g^r)} \\ &= Ye\left(g^{ab}, g^r\right) e(g^{rx}, g) \cdot \frac{e((h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^r, g^{d_O})}{e(g^{ab}(h_1(ID_O||t_O)h_2(ID_O||ID_{PKG}))^{d_O} g^x, g^r)} \\ &= Ye\left(g^{ab}, g^r\right) e(g^{rx}, g) \cdot \frac{1}{e(g^{ab}g^x, g^r)} \\ &= Y \end{aligned}$$

Note that to revoke the membership of a user ID_i, the PKG will update its revocation list L as L' by adding the new entry ID_i, i.e., $L' = L \cup \{ID_i\}$. The decryption processes are presented in Algorithm 7.

Algorithm 7. Decrypt.

Input: PP, SK_{ID}, C
Output: The recovered message m
1: **if** $C = (\alpha, \beta, \theta, \tau)$ **then**
2: $\quad Y = \alpha \cdot \frac{e(SK_{o,2}, \theta)}{e(SK_{o,1}, \beta)}$
3: **elseif** $C = (\alpha', \beta, \theta, \tau, \eta, \rho, \Phi)$ **then**
4: $I = \eta \cdot h_3(\frac{\Phi^w e(SK_{u,1}, g)}{e(h_1(ID_u||t_u)h_2(ID_u||ID_{PKG}), SK_{u,2})})$
5: $\quad Y = \alpha' \cdot \frac{e(\theta, \rho)}{e(I, \beta)}$
6: **end if**
7: **for** $i = 1$ to n
8: $\quad m_i = D(Y, \tau_i)$
9: **next** i
10: $m = (m_1, m_2, \ldots, m_n)$
11: **return** m;

4. Security Model and Proof

To formally prove the security of our improved IB-DS scheme, the authors first present its security model and then give a completed security proof. Since the core building block of the IB-DS scheme is actually the IB-PRE scheme, the notion of a security model for the former also comes from that for the latter. Specifically, we will prove that our improved IB-DS construction is indistinguishable against the adaptively chosen identity and chosen plaintext attacks (IND-PrID-CPA). The security model of IND-PrID-CPA for the IB-DS scheme is defined as follows.

Definition 1. (IND-PrID-CPA). *An IB-DS scheme achieves the indistinguishability against adaptively chosen identity and chosen-plaintext attacks if in the following game, there is no probabilistic adversary A who is able to defeat a challenger B with non-negligible advantage in polynomial-time:*

Setup: In the beginning, the challenger B performs the Setup (1^l) algorithm to initialize the system public parameters PP and a master secret key Msk. Then, the parameters PP are sent to A.
Phase 1: The adversary A will make the following queries adaptively:

- *Private-key Queries:* In this query, the adversary A will provide an identity ID for the challenger B who then calls the Keygen (PP, Msk, ID) algorithm to get the corresponding private key SK_{ID} and returns it.
- *Permission Queries:* In this query, the adversary A will provide two identities (ID_o, ID_u) of non-revoked users and a data category name M_{cate} for the challenger B who first calls the Keygen (PP, Msk, ID) algorithm to gain the private keys SK_{IDo} and SK_{IDu}. Next, B performs the Query (PP, ID_u, SK_{IDu}, M_{cate}) and the Permission (PP, ID_u, SK_{IDo}, TK) algorithms to obtain the re-encryption key RK and returns it.

Challenge: The adversary A determines a target identity ID^*, a message $m^* = (m_1^*, m_2^*, \ldots, m_n^*)$ and two symmetric keys (Y_0, Y_1) of the same length. Next, the challenger B takes the input of (PP, ID^*, m^*, Y_λ) where $\lambda \in_R \{0, 1\}$ to produce a ciphertext $C^* = (\alpha^*, \beta^*, \theta^*, \tau^*)$ as the challenge for A.
Phase 2: After receiving the challenge, the adversary A can further make queries defined as those in phase 1, except for the following restrictions:

- A private-key query for the target identity ID^* is not allowed.
- Any permission query in relation to the identities of the form (ID^*, ID_u) or (ID_o, ID^*) is not allowed.

- The maximum number of times for the private key and the permission queries are bound by q_{pk} and q_{pr}.

Guess: When phase 2 terminates, the adversary A outputs a bit λ'. If $\lambda' = \lambda$, A is the winner of the game. Consequently, the advantage of A is defined as $Adv(A) = |\Pr[\lambda' = \lambda] - 1/2|$.

On the basis of a previously defined security model, we formally prove that our improved IB-DS construction is IND-PrID-CPA-secure in the proof model of random oracles below.

Theorem 1. (IND-PrID-CPA). *Let h_i (for i = 1 and 2) be random oracles. The proposed IB-DS system is indistinguishable against adaptively chosen identity and chosen-plaintext attacks (IND-PrID-CPA) under the DBDH assumption. In particular, if a probabilistic polynomial–time adversary A making at most q_{pk} and q_{pr} queries breaks the IND-PrID-CPA security of our IB-DS scheme with the non-negligible advantage ε, an algorithm B solving the DBDH problem can be constructed with the non-negligible advantage ε' where*

$$\varepsilon' \geq \frac{\varepsilon}{e(\sqrt{q_{pk} + q_{pr} + 1})}.$$

Proof. We depict the proof structure as Figure 4. Let $(g, g^f, g^s, g^k, e(g, g)^{fsk}, \delta)$ be a problem instance of DBDH for B whose purpose is to decide if $e(g, g)^{fsk}$ equals to δ or not by utilizing the advantage of A. In addition, the algorithm B also serves as a challenger responding to the queries that A makes in the following simulation game.

Setup: In the beginning, B performs the Setup(1^l) function to initialize public parameters $PP = \{\boldsymbol{G}_1, \boldsymbol{G}_2, e, g, p, Mpk, E(\cdot), D(\cdot), h_3(\cdot)\}$ where $Mpk = (P = g^f, Q = g^s)$. Note that the Msk of the PKG is implicitly defined as (f, s) which B does not know. Moreover, B chooses a random integer $rn \in Z_p^*$. Then, the parameters PP are sent to A.

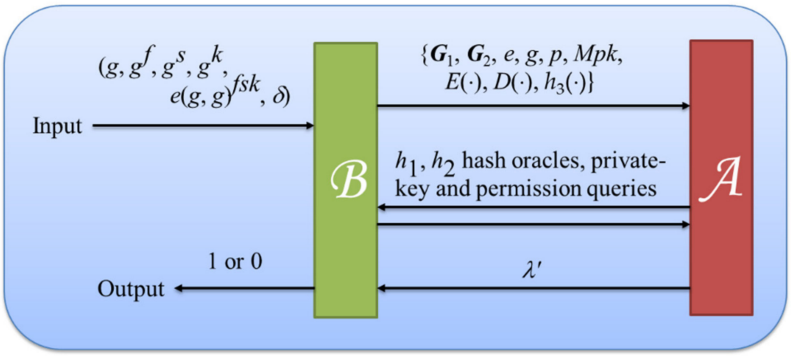

Figure 4. The simulation game between the adversary A and the algorithm B of Theorem 1.

Phase 1: The adversary A will make the following queries adaptively:

- $h_1(ID_i \,||\, t_i)$ *hash oracle:* For any $h_1(ID_i \,||\, t_i)$ query, B uses (ID_i, t_i) as the index to searches for a matched entry in the h_1-table named HT1. Otherwise, B first chooses a bit bt_1 with $\Pr[bt_1 = 0] = \psi$ where ψ will be determined later. When $bt_1 = 0$, B computes $HO_1 = P^{rn} g^{s1}$ where $s_1 \in Z_p^*$; else, B computes $HO_1 = g^{s1}$. Then, B updates HT1 as HT1 $\cup$ $\{(ID_i, t_i, bt_1, s_1, HO_1)\}$ and returns the value HO_1 to A.
- $h_2(ID_i \,||\, ID_{PKG})$ *hash oracle:* For any $h_2(ID_i \,||\, ID_{PKG})$ query, B uses the identity ID_i as an index to searches for a matched entry in the h_2-table named HT2. Otherwise, B first chooses a bit bt_2 with $\Pr[bt_2 = 0] = \psi$ where ψ will be determined later. When $bt_2 = 0$, B computes $HO_2 = P^{rn} g^{s2}$ where $s_2 \in Z_p^*$; else, B computes $HO_2 = g^{s2}$. Then, B updates HT2 as HT2 $\cup$ $\{(ID_i, ID_{PKG}, bt_2, s_2, HO_2)\}$ and returns the value HO_2 to A.
- *Private-key Queries:* For the private-key query of ID_i, B first uses the identity ID_i as an index to searches for matched entries $(ID_i, t_i, bt_1, s_1, HO_1)$ and $(ID_i, ID_{PKG}, bt_2, s_2, HO_2)$ in HT1 and HT2, respectively. (If no such entries exist, B will invoke h_1 and h_2

queries on behalf of A.) When both bt_1 and bt_2 equal to 1, B aborts. In cases where both bt_1 and bt_2 equal to 0, B selects $d_i' \in Z_p^*$ to compute

$$SK_{i,1} = Q^{\frac{-(s_1+s_2)}{2rn}} \left(P^{2rn} g^{s_1+s_2}\right)^{d_i'} \text{ and } SK_{i,2} = Q^{\frac{-1}{2rn}} g^{d_i'}.$$

In the remaining two cases where the values of bt_1 and bt_2 are reversed, B also chooses $d_i' \in Z_p^*$ to compute

$$SK_{i,1} = Q^{\frac{-(s_1+s_2)}{rn}} \left(P^{rn} g^{s_1+s_2}\right)^{d_i'} \text{ and } SK_{i,2} = Q^{\frac{-1}{rn}} g^{d_i'}.$$

As a matter of fact, in either of the above forms of private keys, $SK_{i,1}$ and $SK_{i,2}$ are well-formed, as shown below. To simplify the derivation, we let vx be $s_1 + s_2$ and rx be the value of either $2rn$ or rn.

$$\begin{aligned}
SK_{i,1} &= Q^{\frac{-vx}{rx}} (P^{rx} g^{vx})^{d_i} \\
&= Q^{f} (P^{rx} g^{vx})^{\frac{-s}{rx}} (P^{rx} g^{vx})^{d_i'} \\
&= Q^{f} (P^{rx} g^{vx})^{d_i' - \frac{s}{rx}} \\
&= g^{fs} (h_1(ID_i||t_i) h_2(ID_i||ID_{PKG}))^{d_i''} \text{ where } d_i'' = d_i' - \tfrac{s}{rx} \\
SK_{i,2} &= Q^{\frac{-1}{rx}} g^{d_i'} \\
&= g^{\frac{-s}{rx}} g^{d_i'} \\
&= g^{d_i''} \text{ where } d_i'' = d_i' - \tfrac{s}{rx}
\end{aligned}$$

Then, the computed private keys ($SK_{i,1}$, $SK_{i,2}$) are returned to A. It is evident to observe that the returned private keys have the identical distribution as those in the real scheme.

- *Permission Queries:* For any permission query of (ID_o, ID_u, M_{cate}) where $ID_o \neq ID_u$ and ID_u is not revoked, B first obtains their private keys SK_{IDo} and SK_{IDu} by the private-key queries and then finds out the corresponding information stored in HT1 and HT2. If any of ID_o and ID_u satisfies the condition that both of bt_1 and bt_2 equal to 1, B aborts. Otherwise, B selects w, x, $\pi \in Z_p^*$ to compute $W = g^w$, $RK_1 = \frac{SK_{o,1} g^x}{h_3(e(PW^\pi, Q))}$, $RK_2 = \beta^x$, $RK_3 = SK_{o,2}$, $RK_4 = e(g, Q^\pi)$ where β is the partial ciphertext with respect to M_{cate}. Here, $RK = (RK_1, RK_2, RK_3, RK_4)$ is the derived re-encryption key. Then, B returns RK to A.

Challenge: The adversary A determines a target identity ID^*, a message $m^* = (m_1^*, m_2^*, \ldots, m_n^*)$ and two symmetric keys (Y_0, Y_1) of the same length. Next, the challenger B takes the input of $(PP, ID^*, m^*, Y_\lambda)$ where $\lambda \in_R \{0, 1\}$ to produce a ciphertext $C^* = (\alpha^*, \beta^*, \theta^*, \tau^*)$ for A by the following steps:

Step 1: Without loss of generality, we assume that the corresponding hash queries for ID^* have been queried by A. If any of bt_1^* and bt_2^* equals to 0, B aborts;

Step 2: Otherwise, B computes

$$\alpha^* = Y_\lambda \cdot \delta,$$

$$\beta^* = g^k,$$

$$\theta^* = (g^k)^{(s1^* + s2^*)} = (h_1(ID^* \,||\, t^*) \cdot h_2(ID^* \,||\, ID_{PKG}))^k,$$

where (s_1^*, s_2^*) are the corresponding random values in relation to ID^* and stored in HT1 and HT2, respectively, and

$$\tau^* = (E(Y_\lambda, m_1^*), E(Y_\lambda, m_2^*), \ldots, E(Y_\lambda, m_n^*)).$$

At last, the ciphertext $C^* = (\alpha^*, \beta^*, \theta^*, \tau^*)$ is returned to A as a challenge ciphertext.

Phase 2: After receiving the challenge C^*, the adversary A can further make queries such as those in phase 1, except for the restrictions stated in Definition 1.

Guess: When phase 2 terminates, the adversary A outputs a bit λ'. If $\lambda' = \lambda$, B outputs 1; else, B outputs 0. The former stands for that $e(g, g)^{fsk}$ equals to δ while the latter does not.
Analysis: According to the steps in the challenge phase, if $e(g, g)^{fsk} = \delta$, the simulated C^* would be a valid ciphertext, and hence the advantage of A to break our construction is non-negligible, i.e., $Adv(A) = | \Pr[\lambda' = \lambda] - 1/2 | \geq \varepsilon$. On the contrary, if $e(g, g)^{fsk} \neq \delta$, the adversary A has no better advantage in guessing λ', meaning that $\Pr[\lambda' = \lambda] = 1/2$. Let Pr[Perfect] be the probability of the event that the entire simulation game is perfect without accidental termination. Consequently, we could express the advantage of B to break the DBDH problem as

$$\begin{aligned} & | \Pr[(g, g^f, g^s, g^k, e(g, g)^{fsk}) = 1] - \Pr[(g, g^f, g^s, g^k, \delta) = 1] | \\ \geq & | (1/2 + \varepsilon) - 1/2 | \cdot \Pr[\text{Perfect}] \\ = & \ \varepsilon \cdot \Pr[\text{Perfect}] \end{aligned}$$

To give a better estimation of Pr[Perfect], we first consider the probability that B does not abort in any query simulated above. For convenience, we define some probability events below:

Pr[¬PkQ]: the probability that all private-key queries are perfect without being aborted;
Pr[¬PrQ]: the probability that all permission queries are perfect without being aborted;
Pr[¬Ch]: the probability that the challenge phase is perfect without being aborted.

Since all the above probability events are independent, we know that Pr[Perfect] could be further expressed as $\Pr[\text{Perfect}] = \Pr[\neg\text{PkQ}] \cdot \Pr[\neg\text{PrQ}] \cdot \Pr[\neg\text{Ch}]$. In a private-key query, B aborts when both bt_1 and bt_2 are related to ID_i equal to 1, i.e., $\Pr[\neg\text{PkQ}] \geq (1 - (1 - \psi)^2)^{qpk}$. Similarly, in a permission query, B aborts when both bt_1 and bt_2 are related to ID_o or ID_u equal to 1. Hence, we obtain $\Pr[\neg\text{PrQ}] \geq (1 - (1 - \psi)^2)^{qpr}$. Still, in the challenge phase, B aborts if any of $bt_1{}^*$ and $bt_2{}^*$ corresponding to ID^* equals to 0. That is, $\Pr[\neg\text{Ch}] \geq (1 - \psi)^2$. Combining all of these probability events, we have

$$\begin{aligned} \Pr[\text{Perfect}] &\geq [(1 - (1 - \psi)^2)^{q_{pk}}][(1 - (1 - \psi)^2)^{q_{pr}}](1 - \psi)^2 \\ &= [(1 - (1 - \psi)^2)^{q_{pk} + q_{pr}}](1 - \psi)^2 \end{aligned}$$

When $\psi = 1 - \frac{1}{\sqrt{q_{pk}+q_{pr}+1}}$, the probability of Pr[Perfect] achieves the maximum value $\frac{1}{e(\sqrt{q_{pk}+q_{pr}+1})}$ where e is the base of natural logarithm. Accordingly, the advantage for B to solve the DBDH problem is calculated as $\varepsilon' \geq \frac{\varepsilon}{e(\sqrt{q_{pk}+q_{pr}+1})}$. □

Theorem 2. *The proposed construction is secure against the dishonest fog (proxy) that attempts to learn the plaintext from the ciphertext requested by a data user.*

Proof. In the permission phase of our scheme, a dishonest proxy can obtain the re-encryption key RK composed of four subkeys in which $RK_1 = \frac{SK_{o,1} g^x}{h_3(e(PW^\pi, Q))}$. If the proxy tries to derive the private key $SK_{o,1}$ for decrypting the original ciphertext $C = (\alpha, \beta, \theta, \tau)$, he has to know the two random numbers (x, π), which are chosen by the data owner. Consequently, he cannot successfully derive $SK_{o,1}$ from the re-encryption key RK_1. Furthermore, if he attempts to learn the plaintext from the re-encrypted ciphertext $C' = (\alpha', \beta, \theta, \tau, \eta, \rho, \Phi)$, he will face the difficulty in computing the parameter I owing to the lack of the data user's private keys. □

Theorem 3. *The proposed construction is secure against the malicious or compromised PKG who attempts to gain the access to any cloud ciphertext without having the corresponding private key.*

Proof. In the keygen algorithm of the proposed system, we also adopt the technique of anonymous key generation to issue each user's private keys. Although the private key $SK_{i,2}$ is controlled by the PKG, it cannot derive the private key $SK_{i,1} = g^{ab}(h_1(ID_i \,||\, t_i) \cdot$

$h_2(ID_i \,||\, ID_{PKG}))^{di}$ without knowing the secret value t_i chosen by the user. As for directly computing $h_1(ID_i \,||\, t_i)$ from the received parameter T'_i, the PKG also does not have the correct knowledge of $Z_i = g^{zi}$. Without having the full private keys of any user, a malicious or compromised PKG is impossible to decrypt either an original or a re-encrypted ciphertext with Equations (19) and (26). □

Theorem 4. *The proposed construction is secure against any revoked user who attempts to impersonate a legitimate user to gain access to any cloud ciphertext without having the corresponding private key.*

Proof. According to the query algorithm in our scheme, a revoked user ID_u impersonating a non-revoked user ID_v first chooses a random number $w \in_R Z_p^*$ to compute $W = g^w$, and sends (ID_v, M_{cate}, W) to the nearby proxy. Since the transmitted identity is ID_v, it will not be rejected by the PKG in the permission phase. Finally, ID_u will receive a re-encrypted ciphertext $C' = (\alpha', \beta, \theta, \tau, \eta, \rho, \Phi)$. However, to decrypt the ciphertext, he needs to know the correct private keys of ID_v in addition to his chosen random number w. Without the former, he cannot successfully recover the original message. □

We compare the functionality and security of our improved construction with some previous ones including Han et al.'s (HSM for short) [33], Tang et al.'s (THJ for short) [34], Wang et al.'s (WWM) [35], Matsuo's (Mat for short) [36] and Zhang et al.'s (ZBW for short) [24] in Table 2. From the table, it is obvious that only the schemes of ZBW and ours support anonymous key generation and user revocation. The schemes of THJ and Mat cannot resist the collusion attack plotted by the proxy and the data user. Except for the proposed construction, all the other compared ones are subject to the malicious (compromised) PKG attack. To sum up, our improved DS variant has better functionality and security among all compared mechanisms.

Table 2. Comparison of functionality and security.

Item \ Scheme	HSH	THJ	WWM	Mat	ZBW	Ours
Support anonymous key generation	No	No	No	No	Yes	Yes
Support user revocation	No	No	No	No	Yes	Yes
Withstand the malicious (compromised) PKG	No	No	No	No	No	Yes
Withstand the collusion attack	Yes	No	Yes	No	Yes	Yes
Withstand the dishonest proxy	Yes	Yes	Yes	Yes	No	Yes
Withstand the revoked user	n.a.	n.a.	n.a.	n.a.	No	Yes

Remark: The term of n.a. stands for not applicable.

Since only Han et al.'s [33] and Zhang et al.'s schemes [24] have similar structures to ours, we further make the efficiency comparison below. We consider the computation of bilinear pairing and exponentiation in our improved algorithms. The detailed results are shown in Table 3. Note that we use the symbols of "B", "E1" and "E2" to separately represent a bilinear pairing, an exponentiation computation over G_1 and an exponentiation computation over G_2. The numerical comparisons are also illustrated in Figure 5 by using the hardware of Intel Core 2 Duo 2.10 Ghz CPU and 2 GB RAM. The software platform is the Ubuntu 9.10 operating system and the PBC library [37]. The estimated running times of B, E1 and E2 computation are approximately 5.883, 0.736 and 0.142 ms, respectively. Although the proposed algorithms incur higher computational costs in the decryption phase, it could be regarded a reasonable trade-off to obtain a higher security.

Table 3. Comparison of computational complexity.

Item \ Scheme	HSM	ZBW	Ours
Cost of setup algorithm	6E1	2E1	2E1
Cost of keygen algorithm	5B + 5E1	5E1	5E1
Cost of encrypt algorithm	3B + 3E1 + E2	B + 2E1 + E2	B + 2E1 + E2
Cost of query algorithm	2E1	E1	E1
Cost of permission algorithm	5B + 4E1 + E2	2E1	2B + 3E1 + E2
Cost of re-encryption algorithm	0	B	B
Cost of decryption algorithm by ID_o	2B	2B	2B
Cost of decryption algorithm by ID_u	2B + 2E1	2B + E1	4B + E2

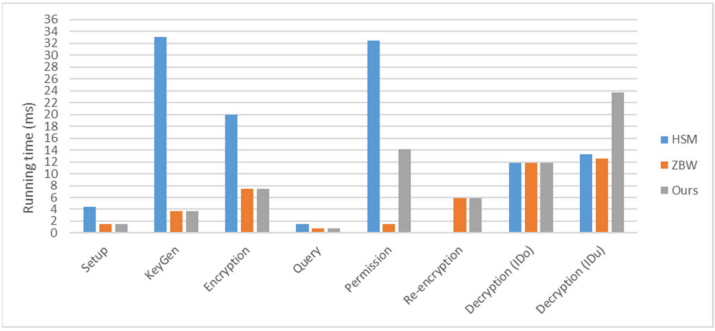

Figure 5. Numerical comparison of computational complexity.

The communication overheads are evaluated in terms of the length of the query token, the ciphertext and the re-encrypted ciphertext. For simplicity, the length of identity and that of the data category name are ignored in the comparison. Assume that the output length of symmetric encryption is $|\mathrm{SE}|$. The detailed results are shown in Table 4. It is evident that the query token length of our scheme is shorter than that of ZBW by $|G_1|$. Yet, when transmitting the re-encrypted ciphertext, our scheme has to send an extra element of G_2. We claim that the extra element is crucial for protecting the ciphertext from unauthorized decryption. The numerical comparisons are also illustrated in Figure 6 by using the elliptic curve of embedding degree 2. Hence, the size of order p is 160 bits and that of a field element is 512 bits.

Table 4. Comparison of communication overheads.

Item \ Scheme	ZBW	Ours
Query token length	$3\|G_1\|$	$2\|G_1\|$
Ciphertext length	$2\|G_1\| + \|G_2\| + n\|\mathrm{SE}\|$	$2\|G_1\| + \|G_2\| + n\|\mathrm{SE}\|$
Re-encryption ciphertext length	$4\|G_1\| + \|G_2\| + n\|\mathrm{SE}\|$	$4\|G_1\| + 2\|G_2\| + n\|\mathrm{SE}\|$

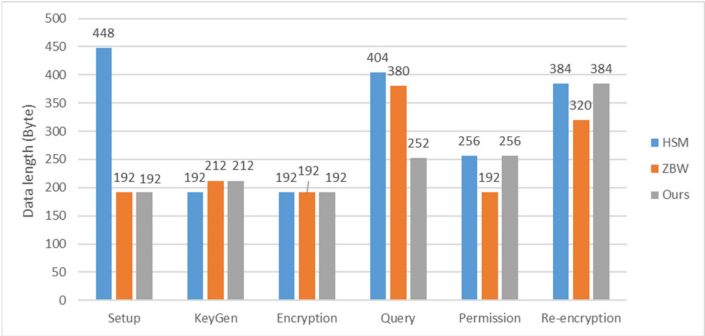

Figure 6. Numerical comparison of communication overheads.

5. Conclusions

To enhance the security of more and more popular data applications in fog-enabled IoT environments, in this paper, we proposed an improved data storage scheme following Zhang et al.'s work [24]. We pointed out several security vulnerabilities in their scheme. Concretely speaking, their scheme fails to satisfy the basic access policy that only the user owning the correct private key can decrypt the corresponding cloud ciphertext. Hence, an adversary including the dishonest proxy, malicious PKG and revoked users can arbitrarily request a cloud ciphertext and decrypt it without having the knowledge of corresponding private key. To eliminate the above security flaws, we have modified some algorithms in our improved system. Moreover, we formally proved that our construction is IND-PrID-CPA-secure in the random oracle models. Overall, the advantages of anonymous key generation and user revocation are also preserved in the proposed variant with higher security. Our improved mechanism can provide better security protection for the applications in fog-enabled IoT environments. Although the computational complexity is increased, we believe that it is a worthy trade-off to gain a higher level of security. The limitation of our mechanism is that each user has to maintain an extra secret value chosen at the keygen phase. Such a value will be utilized in the decryption process. The aim of future work should be to combine attribute-based mechanisms for supporting more fine-grained access control policies.

Author Contributions: Writing—original draft, H.-Y.L.; Writing—review and editing, T.-T.T. and P.-Y.T.; Resources, C.-C.C. 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 numbers MOST 110-2221-E-019-041-MY3 and MOST 110-2222-E-019 -001-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. McCarthy, J. Reminiscences on the history of time sharing. *IEEE Ann. Hist. Comput.* **1992**, *14*, 19–24.
2. Christos, S.C.; Panagiotis, T.; Christos, G. Combined multi-layered big data and responsible AI techniques for enhanced decision support in Shipping. In Proceedings of the 2020 International Conference on Decision Aid Sciences and Application (DASA), Sakheer, Bahrain, 8–9 November 2020; pp. 669–673.
3. Spandonidis, C.; Sedikos, E.; Giannopoulos, F.; Petsa, A.; Theodoropoulos, P.; Chatzis, K.; Galiatsatos, N. A novel intelligent iot system for improving the safety and planning of air cargo operations. *Signals* **2022**, *3*, 95–112. [CrossRef]
4. Patwary, A.A.N.; Naha, R.K.; Garg, S.; Battula, S.K.; Patwary, M.A.K.; Aghasian, E.; Amin, M.B.; Mahanti, A.; Gong, M. Towards secure fog computing: A survey on trust management, privacy, authentication, threats and access control. *Electronics* **2021**, *10*, 1171. [CrossRef]
5. Blaze, M.; Bleumer, G.; Strauss, M. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology—EUROCRYPT'98*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 127–144.
6. 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, 23–26 February 2005; pp. 29–43.
7. 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]
8. 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, 21–24 March 2006; pp. 46–58.
9. Dodis, Y.; Ivan, A. Proxy cryptography revisited. In Proceedings of the 10th Network and Distributed System Security Symposium, San Diego, CA, USA, 6–7 February 2003.
10. 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.
11. Green, M.; Ateniese, G. Identity-based proxy re-encryption. In Proceedings of the Applied Cryptography and Network Security'07 (ACNS 2007), Zhuhai, China, 5–8 June 2007; pp. 288–306.

12. Chu, C.K.; Tzeng, W.G. Identity-based proxy re-encryption without random oracles. In Proceedings of the 10th Information Security Conference (ISC'07), Valparaiso, IN, USA, 9–12 October 2007; pp. 189–202.
13. Shao, J.; Cao, Z. CCA-Secure proxy re-encryption without pairings. In Proceedings of the Public Key Cryptography (PKC 2009), Irvine, CA, USA, 18–20 March 2009; pp. 357–376.
14. Fang, L.; Susilo, W.; Ge, C.; Wang, J. Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search. *Theor. Comput. Sci.* **2012**, *462*, 39–58. [CrossRef]
15. Wang, X.A.; Huang, X.; Yang, X.; Liu, L.; Wu, X. Further observation on proxy re-encryption with keyword search. *J. Syst. Softw.* **2012**, *85*, 643–654. [CrossRef]
16. Park, D.; Cha, J.; Lee, P. Searchable Keyword-Based Encryption. Cryptology ePrint Archive 2005, Report 2005/367. Available online: https://eprint.iacr.org/2005/367 (accessed on 22 April 2022).
17. Liang, K.; Fang, L.; Susilo, W.; Wong, D.S. A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security. In Proceedings of the IEEE 2013 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS), Xi'an City, China, 9–11 September 2013; pp. 552–559.
18. Akhil, N.V.; Vijay, A.; Kumar, D.S. QR code security using proxy re-encryption. In Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 18–19 March 2016; pp. 1–5.
19. Zeng, P.; Choo, K.R. A new kind of conditional proxy re-encryption for secure cloud storage. *IEEE Access* **2018**, *6*, 70017–70024. [CrossRef]
20. Vohra, K.; Dave, M. Securing fog and cloud communication using attribute based access control and re-encryption. In Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018), Coimbatore, India, 20–21 April 2018; pp. 307–312.
21. Chen, B.; He, D.; Kumar, N.; Wang, H.; Choo, K.K.R. A blockchain-based proxy re-encryption with equality test for vehicular communication systems. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2048–2059. [CrossRef]
22. Xiong, H.; Wang, L.; Zhou, Z.; Zhao, Z.; Huang, X.; Kumari, S. Burn after reading: Adaptively secure puncturable identity-based proxy re-encryption scheme for securing group message. *IEEE Internet Things J.* **2021**. [CrossRef]
23. Ge, C.; Susilo, W.; Baek, J.; Liu, Z.; Xia, J.; Fang, L. A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds. *IEEE Trans. Dependable Secur. Comput.* **2021**. [CrossRef]
24. Zhang, J.; Bai, W.; Wang, X. Identity-based data storage scheme with anonymous key generation in fog computing. *Soft Comput.* **2020**, *24*, 5561–5571. [CrossRef]
25. Ahene, E.; Qin, Z.; Adusei, A.K.; Li, F. Efficient signcryption with proxy re-encryption and its application in smart grid. *IEEE Internet Things J.* **2019**, *6*, 9722–9737. [CrossRef]
26. Rawal, B.S. A proxy re-encryption-based webmail and file sharing system for collaboration in cloud computing environment. In Proceedings of the 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 21–22 December 2018; pp. 213–218.
27. Ge, C.; Liu, Z.; Xia, J.; Fang, L. Revocable identity-based broadcast proxy re-encryption for data sharing in clouds. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 1214–1226. [CrossRef]
28. Chen, W.H.; Fan, C.I.; Tseng, Y.F. Efficient key-aggregate proxy re-encryption for secure data sharing in clouds. In Proceedings of the 2018 IEEE Conference on Dependable and Secure Computing (DSC), Kaohsiung, Taiwan, 10–13 August 2018; pp. 1–4.
29. Yang, C.C.; Tso, R.; Liu, Z.Y.; Hsu, J.C.; Tseng, Y.F. Improved proxy re-encryption scheme with equality test. In Proceedings of the 2021 16th Asia Joint Conference on Information Security (AsiaJCIS), Seoul, Korea, 19–20 August 2021; pp. 37–44.
30. Agyekum, K.O.B.O.; Xia, Q.; Sifah, E.B.; Cobblah, C.N.A.; Xia, H.; Gao, J. A proxy re-encryption approach to secure data sharing in the Internet of things based on blockchain. *IEEE Syst. J.* **2021**, *16*, 1685–1696. [CrossRef]
31. Nirmala, V.; Shanmugalakshmi, R. Hierarchical identity role based proxy re-encryption scheme for cloud computing. In Proceedings of the 2013 International Conference on Advanced Computing and Communication Systems, Coimbatore, India, 6–7 April 2013; pp. 1–4.
32. Yasumura, Y.; Imabayashi, H.; Yamana, H. Attribute-based proxy re-encryption method for revocation in cloud storage: Reduction of communication cost at re-encryption. In Proceedings of the 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China, 9–12 March 2018; pp. 312–318.
33. Han, J.; Susilo, W.; Mu, Y. Identity-based data storage in cloud computing. *Future Gener. Comput. Syst.* **2013**, *29*, 673–681. [CrossRef]
34. Tang, Q.; Hartel, P.; Jonker, W. Inter-domain identity-based proxy re-encryption. *Inf. Secur. Cryptol.* **2009**, *5487*, 332–347.
35. Wang, L.; Wang, L.; Mambo, M.; Okamoto, E. New identity-based proxy re-encryption schemes to prevent collusion attacks. In Proceedings of the 4th International Conference on Pairing-based Cryptography (Pairing'10), Yamanaka Hot Spring, Japan, 13–15 December 2010; pp. 327–346.
36. Matsuo, T. Proxy re-encryption systems for identity-based encryption. In Proceedings of the 1st International Conference on Pairing-based Cryptography (Pairing'07), Tokyo, Japan, 2–4 July 2007; pp. 247–267.
37. PBC Library, the Pairing-Based Cryptography Library. Available online: http://crypto.stanford.edu/pbc/ (accessed on 22 April 2022).