sensors

MDPI

Article

Hidden Policy Attribute-Based Data Sharing with Direct Revocation and Keyword Search in Cloud Computing

Axin Wu [1,2], Dong Zheng [1,2], Yinghui Zhang [1,2,*] and Menglei Yang [1]

[1] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; waxinsec@163.com (A.W.); zhengdong@xupt.edu.cn (D.Z.); kmelly@163.com (M.Y.)
[2] Westone Cryptologic Research Center, Beijing 100070, China
* Correspondence: prrd2007@163.com; Tel.: +86-182-2053-2628

Received: 2 June 2018; Accepted: 3 July 2018; Published: 4 July 2018

Abstract: Attribute-based encryption can be used to realize fine-grained data sharing in open networks. However, in practical applications, we have to address further challenging issues, such as attribute revocation and data search. How do data users search for the data they need in massive amounts of data? When users leave the system, they lose the right to decrypt the shared data. In this case, how do we ensure that revoked users cannot decrypt shared data? In this paper, we successfully address these issues by proposing a hidden policy attribute-based data sharing scheme with direct revocation and keyword search. In the proposed scheme, the direct revocation of attributes does not need to update the private key of non-revoked users during revocation. In addition, a keyword search is realized in our scheme, and the search time is constant with the increase in attributes. In particular, the policy is hidden in our scheme, and hence, users' privacy is protected. Our security and performance analyses show that the proposed scheme can tackle the security and efficiency concerns in cloud computing.

Keywords: cloud computing; attribute-based encryption; direct revocation; keyword search; hidden policy

1. Introduction

With the application of intelligent terminals in our lives, a large amount of data can be generated quickly. These collected data are closely related to our lives. By analyzing personal data, one's behavior can be predicted, and by analyzing the enterprise data, a lot of business secrets can be obtained which can pose a serious threat to individuals [1] or enterprises [2]. Furthermore, there are many threats to data privacy during the processes of data processing [3,4], data transmission [5], data storage [6,7], data search [8], data confidentiality [9,10] and data access [11,12]. Among these security problems, we focus on security issues in cloud storage and cloud computing.

While the rapid development of cloud computing brings convenience to enterprises and individuals because of its storage services, computing services, scalability and so on, data security and user privacy are also a big problem [13,14] owing to data being exposed in open network environments [15]. Encrypting data before uploading data to the cloud server can solve data security and user privacy issues very well [16,17]. However, the encryption of data causes the loss of some characteristics of plaintext, and data sharing among numerous data users becomes another problem. Fortunately, attribute-based encryption (ABE) [18] provides a good solution to the data sharing and access control on cloud storage and cloud computing. After Sahai et al. proposed the notion of ABE, much work was done to improve the function and efficiency of the ABE. For example, Li et al. [19] proposed a multi-authority, fine-grained access control scheme. Zhang et al. [20] proposed an anonymous access control scheme for proxy re-encryption. Shen et al. [21] proposed a data sharing scheme

with anonymous tracking. These schemes are extensions of the ABE scheme and can be applied to some specific environments. In the scenario of data sharing, the Ciphertext Policy Attribute-Based Encryption (CP-ABE) [22] is more popular. In the CP-ABE system, the data owner specifies the access control structure related to ciphertexts. Only when the user attributes connected with the user secret key satisfy the access control structure, can the data be decrypted correctly. For example, Cai et al. [23] applied CP-ABE to the medical cloud which can help to improve the quality of medical services. Zhang et al. [24] applied CP-ABE to the mobile cloud computing, which makes it possible for resource-limited users to share data with others.

Although CP-ABE can bring a lot of convenience to our lives, there are still many problems to be considered in practical applications. For example, how can we ensure that revoked users cannot decrypt shared data? How do data users search for the data they want among massive amounts of data? In addition, in the CP-ABE system, the access control structure is also uploaded to the cloud server with the ciphertext, which may also leakage some sensitive information. In order to solve the above problems, the typical CP-ABE scheme is no longer suitable for the complex cloud computing environment. Therefore, searchable attribute-based encryption schemes (SABE) [25] and revocable attribute-based encryption schemes (RABE) [26] have been put forward.

In SABE [27,28], the data owner will upload the encrypted keyword index together with the ciphertext. When data users want to use data, he will generate a keyword trapdoor with his secret key, then uploads it to the cloud server. The cloud server checks whether the ciphertext containing the keyword index exists on the server without knowing the keyword. If it exists, the ciphertext will be returned to data users. Therefore, data users can retrieve the data they want based the keyword trapdoor. However, the search time in most searchable attribute-based encryption schemes increases with the number of attributes, which increases the burden on the server and reduces the user experience. In addition, when the access control structure is uploaded, it will also leakage some sensitive information.

The revocation scheme has practical application in dynamic networks and systems. For example, when a user leaves the system, the user identity is revoked in the system [29] which increases the security of the system. The RABE scheme can be divided into indirectly revocable attribute-based encryption (IRABE) schemes [30,31] and directly revocable attribute-based encryption (DRABE) schemes [32–35]. In the IRABE schemes, the revocation list is maintained by the authority center. When the user is removed from the system, the authority center updates the secret key of the non-revoked user. In DRABE schemes, the user's revocation list is held by the user. When a user is revoked, the user's private key does not need to be updated. Comparing the two schemes, the direct revocation scheme is more suitable for open network environments. In order to prevent the revoked users from decrypting the previous ciphertext, we can use the powerful computing power of cloud computing to update the ciphertext when the user is revoked from the system.

In order to make the data sharing scheme of CP-ABE more applicable to practical applications, it is necessary to propose a data sharing scheme with the functions of direct revocation and keyword search.

1.1. Our Contribution

In order to solve the problem described above and make the data sharing scheme of ABE more practical, we propose a hidden policy attribute-based data sharing scheme with direct revocation and keyword search (ABERS). Our scheme has the following advantages:

- Direct revocation of attributes: We use subset covering theorem to achieve the direct revocation of attributes. After revocation, there is no need to update the private key of the non-revoked user. In order to ensure that the users who have been revoked cannot decrypt the previous ciphertext, the ciphertext is updated.
- Fast keyword search: We use aggregation technology to achieve the fast search of keywords. Keyword search time is constant and will not increase with the numbers of attributes.

- Hidden policy: We use the AND gate access control structure to achieve the hidden policy. When the ciphertext is uploaded, the access control structure does not need to be uploaded. Thus, the function of the hidden policy can be realized.

1.2. Related Work

We review the work of the AND-gate attribute based encryption, the revocable attribute based encryption and the authorized keyword search in this section.

AND-gate attribute based encryption: Sahai and Waters [18] proposed the ABE scheme to solve data sharing and data access control. After Sahai et al. had proposed the ABE, much work was done to improve the function and efficiency of the ABE. In order to apply the function of the ABE scheme more flexibly, ABE was divided into key-policy ABE (KP-ABE) [36] and ciphertext-policy ABE (CP-ABE) [22,37]. In order to facilitate the application of terminal devices, some work was also done in references [38,39]. An AND-gate access structure ABE was introduced by Cheung and Newprot [40]. Unfortunately, there is no hidden access policy in this scheme. Due to the appearance of inner product encryption schemes, several other schemes [28,41,42] follow this structure, while hiding the access policy.

Revocable attribute based encryption: The RABE scheme is divided into IRABE schemes [30,31] and DRABE schemes [32–34]. In IRABE schemes, the revocation list is maintained by the authority center. When the user is removed from the system, the authority center updates the secret key of the non-revoked user. In DRABE schemes, the user's revocation list is held by the user. When a user is revoked, the user's private key does not need to be updated. In order to prevent revoked users from decrypting ciphertexts that existed before revocation, some work on the re-encryption proxy was done in reference [31] without interacting with data owners and in reference [43] without interacting with non-revoked users.

Authorized keyword search: The search encryption scheme can be traced back to Perrig et al. [44]. Unfortunately, the scheme has a high computational cost. To accelerate the search, Lee et al. [45] implemented search encryption through hash tables. To make the application scene more flexible, a keyword search encryption scheme based on a public key was proposed in reference [46,47]. In order to make the search more secure, the authentication search encryption scheme was proposed in reference [48,49]. Further work was done in references [50,51]. In reference [50], the authorized keyword search was implemented through ABE technology with multi-keywords. The scheme presented in reference [51] can be applied to multi-user and multi-owner scenes; however, it is not suitable for dynamic network environments.

2. Preliminary

In this section, we mainly introduce the basic knowledge about attribute revocation and keyword search.

2.1. Access Control Structure

The "AND gate" access structure [52] is described as follows: Let $S = (x_1, x_2, \ldots, x_n)$ represent an attribute list of a user. Let $W = (w_1, w_2, \ldots, w_n)$ represent an access policy. The attributes satisfy the access control structure, if and only if $x_i = w_i, i \in [1, \ldots, n]$. Because the access control structure and attribute set have the same structure, when uploading ciphertext, there is no need to upload the control structure. So, the hidden policy can be achieved.

2.2. Multilinear Maps

The concept of multilinear mapping was first proposed by Boneh and Silverberg, as the following [53]. First, run $\Gamma(1^\lambda, n)$ to get $G = (<G_1>, \ldots, <G_n>)$, where λ is a security parameter. The description of the prime number group, G_i, whose order is $p > 2^\lambda$ and contains the generator, g_i,

of G_i, is expressed by $< G_i >$. A series of linear maps, $\{e_{i,j} : G_i \times G_j | i, j \geq 1; i + j \leq n\}$, are defined as follows:

$$e_{i,j}(g_i^u, g_j^v) = g_{i+j}^{uv} \quad \forall u, v \in Z_p.$$

We simplify the description as $e(g_i^u, g_j^v) = g_{i+j}^{uv}$.

2.3. Subset Cover

First, we introduce the full binary tree T of depth d, in which two functions $depth(x)$ and $path(x)$ are involved. Both $depth(x)$ and $path(x)$ take node x as the input. The function $depth(x)$ takes the depth of node x as the output. The function $path(x)$ takes the path from the roo,t $P_{x,0} = root$, to the node, $P_{x,denpth(x)} = x$, as the output. The use of subset cover theorem to solve the user revocation was referred to by Naor et al. [54]. Let the leaf node express the user in the system. For a set of revoked users, R, we can get all paths, $\{path(x)\}_{\forall x \in R}$, of the revocation node $x \in R$. The $cover(R)$ is the smallest set that can cover the unmarked nodes. For ease of understanding, we give a simple example, shown in Figure 1. Eight leaves $x_8, \ldots, x_{15}$ are contained in the full binary tree, T. If $R = \{x_8, x_{11}\}$ is a revocation set. The paths of nodes x_8 and x_{11} are $path(x_8) = (x_1, x_2, x_4, x_8)$ and $path(x_{11}) = (x_1, x_2, x_5, x_{11})$, respectively. The $cover(R)$ set is $\{x_3, x_9, x_{10}\}$. The non-revoked leaf nodes are covered by $cover(R)$.

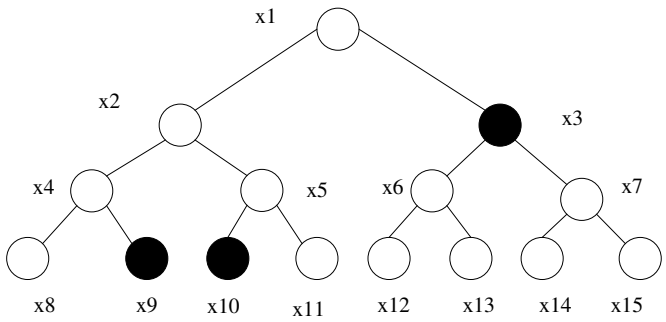

Figure 1. Subset cover.

Assumption 1. *n-Multi − linearDecisionalDiffie–Hellman (n-MDDH): Run* $\Gamma(1^\gamma, n)$ *to get* $G = (<G_1>, \ldots, <G_n>)$. *Select* $v_0, \ldots, v_n \in Z_p$. *Compute* $g_1^{v_0}, \ldots, g_1^{v_n}$. *For any poly-time algorithm, it is difficult with non-negligible advantage to tell* $g_n^{\prod_{j \in [0,..,n]} v_j}$ *from a random element in* G_n. *Please refer to [55] for more details.*

3. Definition

In this section, we mainly introduce the deployment of the model, the definition of the scheme and the security model of the scheme.

3.1. Deployment

ABERS can be applied to real environments. The data sharing system is shown in Figure 2. It involves four entities: data owner, data user, attribute authority and cloud server. Now, we will introduce their specific functions and functions.

- Data owner: The data owner is responsible for encrypting the data and generating the keyword index, I, and then uploading the ciphertext, CT, and keyword index, I. When the revocation list changes, the revocation list, R', is sent to the cloud server by the data owners.
- Data user: When data users want to download data, they should first use their own private keys to generate a keyword trapdoor, T, and then send T to the cloud server to check it. If the request is legal, then the desired data CT can be obtained.

- Attribute authority: The attribute authority is responsible for managing all users in the system, initializing the system, publishing the system's public parameters, PK, and generating the secret key, SK, for the user.
- Cloud server: The cloud server is responsible for storing the ciphertext of the data owner. When the data user sends the keyword trapdoor to the cloud server, the cloud server searches for it. If the file exists, it is returned to the data user. When the new revocation list is received from the data owner, the cloud server updates the ciphertext with the $Updata(R')$ algorithm.

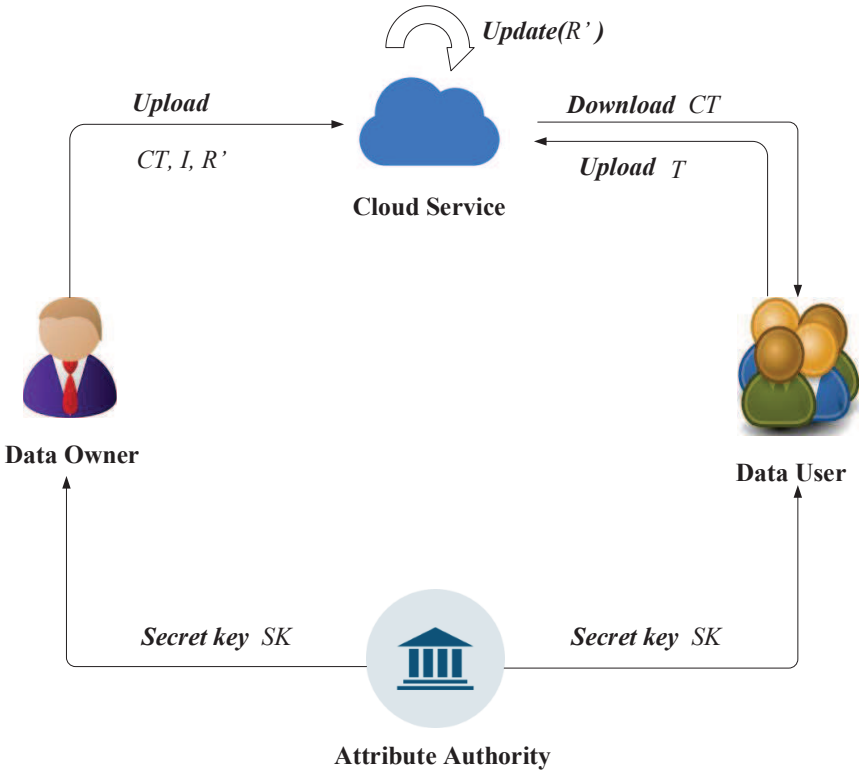

Figure 2. The data sharing system. CT is the ciphertext, I is the keyword index, R' is the revocation list, T is the keyword trapdoor and SK is the secret key.

3.2. Definition of the System Model

Our construction algorithm consists of the following eight algorithms.

$Setup(1^{\lambda}, d, I, U) \rightarrow PK, MSK$: The algorithm takes the security parameters, λ, the depth of the tree, d, the set of the user identity, I, and the collection of attributes, U, as inputs with the common system parameters, PK, and the main secret key, MSK, as the outputs.

$Keygen(PK, MSK, S, id) \rightarrow SK_S$: The algorithm uses PK, MSK, the user attribute, S, and the user identity, id, as inputs, with SK_S as the output.

$Encrypt(PK, M, W, R, w) \rightarrow CT_{W,R}, I_{\omega}$: This algorithm uses PK, the message, M, an AND-gate access structure, W, a revocation list, R and the keyword, w, as inputs, with the ciphertext, $CT_{W,R}$, and keyword index, I_{ω}, as the outputs.

$Trapdoor(SK_S, w) \rightarrow t_{\omega}$: This algorithm takes the user's secret key, SK_S, and a keyword, w, as inputs with a trapdoor, t_{ω}, as the output.

$Test(I_{\omega}, t_{\omega}) \rightarrow 0$ *or* 1: This algorithm takes the keyword index, I_{ω}, and a trapdoor, t_{ω}, as inputs with a Boolean value, $\{0, 1\}$, as the output.

$Decryption(PK, CT_{W,R}, SK_S) \rightarrow m$ or $\perp$: This algorithm takes $PP, CT_{W,R}$ and SK_S as inputs, with m or $\perp$ as the output.

$Update(CT_{W,R}, R') \rightarrow CT_{W,R'}$: This algorithm takes $CT_{W,R}$ and R' as inputs with $CT_{W,R'}$ as the output.

3.3. Definition of System Security

The adversaries against the ABERS scheme include unauthorized data users and revoked data users. For unauthorized users, their attributes do not satisfy the access control structure. For revoked data users, their identities are in the revocation list. Both of them try their best to get the information of the ciphertext. Their behavior also includes a secret key recovery attack. They want to get a private key from a keyword trapdoor. The concrete models are as follows:

Indistinguishability against chosen plaintext attack (IND-CPA):
This security game is defined as follows:

- Init: The adversary, A, sends a revocation list, R^*, chosen by A to the challenger, B.
- Setup: B calls the algorithm $Setup(1^\lambda, d, I, U) \rightarrow PP, MSK$, and then sends PP to A.
- Phase 1: The adversary, A, is able to ask B about the private key of user (S, id).

  When $id \notin R^*$, the enquiry is aborted. Otherwise, B calls the algorithm $Keygen(PP, MSK, S, id) \rightarrow SK_S$ and then sends SK_S to A.
- Challenge: A sends two messages m_0^*, m_1^* ($|m_0^*| = |m_1^*|$) and a challenge access structure, W, to B. B randomly selects $b \in \{0, 1\}$ and then calls the algorithm $Encrypt(PP, m_b^*, W, R) \rightarrow CT_{W,R}$ and finally, sends $CT_{W,R}$ to A.
- Phase 2: A does the same inquiries as in Phase 1.
- Guess: A outputs the guess of b as $b' \in \{0, 1\}$.

In this game, the advantage of adversary A is defined as follows:

$$Pr_A = |Pr[b = b'] - \frac{1}{2}|$$

Definition 1. *If the advantage, Pr_A, of any polynomial-time adversary A is negligible, then the ABERS scheme is selectively indistinguishable under the $(d + 3)$-MDDH assumption.*

Indistinguishability against chosen keyword attack (IND-CKA):
This security game is defined as follows:

- Setup: B calls the algorithm $Setup(1^\lambda, d, I, U) \rightarrow PP, MSK$ and then sends PP to A.
- Phase 1: The adversary, A, is able to ask B about the private key of user (S, id). B calls the algorithm $Keygen(PP, MSK, S, id) \rightarrow SK_S$ and then sends SK_S to A.
- Challenge: A sends two messages, w_0^*, w_1^* ($|w_0^*| = |w_1^*|$), and a challenge access structure, W, to B. B randomly selects $b \in \{0, 1\}$ and then calls the algorithm $Encrypt(PP, M, W, R, w_b^*) \rightarrow CT_{W,R}, I_\omega$ and finally, sends I_ω to A.
- Phase 2: A does the same inquiries as in Phase 1.
- Guess: A outputs the guess of b as $b' \in \{0, 1\}$.

In this game, the advantage of adversary A is defined as follows:

$$Pr_A = |Pr[b = b'] - \frac{1}{2}|$$

Definition 2. *If the advantage, Pr_A, of any polynomial-time adversary, A, is negligible, then the ABERS scheme is indistinguishable against the chosen keyword attack.*

Selective security game on updated ciphertext:
This security game is defined as follows:

- Setup: The adversary, A, sends two revocation lists, R and R^*, and an attribute, S^*, that chosen by A to the challenger, B. B calls the algorithm $Setup(1^\lambda, d, I, U) \rightarrow PP, MSK$ and then sends PP to A.
- Phase 1: The adversary, A, is able to ask B about the private key of user (S^*, id). When $id \notin R^*$, the enquiry is aborted. Otherwise, B calls the algorithm $Keygen(PP, MSK, S, id) \rightarrow SK_S$ and then sends SK_S to A.
- Challenge: A sends two messages, m_0^*, m_1^* ($|m_0^*| = |m_1^*|$), and a challenge access structure, W, to B. B randomly selects $b \in \{0, 1\}$ and then calls the algorithm $Encrypt(PP, m_b^*, W, R, w) \rightarrow CT_{W,R}, I_\omega$ and $Update(CT_{W,R}, R') \rightarrow CT_{W,R'}$ and finally, sends $CT_{W,R'}$ to A.
- Phase 2: A does the same inquiries as in Phase 1.
- Guess: A outputs the guess of b as $b' \in \{0, 1\}$.

In this game, the advantage of adversary A is defined as follows:

$$Pr_A = |Pr[b = b'] - \frac{1}{2}|$$

Definition 3. *If the advantage, Pr_A, of any polynomial-time adversary, A, is negligible, then the ABERS scheme has selective security under the $(d + 3)$-MDDH assumption.*

4. Data Sharing System

In this section, we mainly introduce the concrete scheme, which contains the following seven algorithms *System initialization*, *User registration*, *Ciphertext upload*, *Trapdoor generation*, *Ciphertext retrieval*, *Ciphertext decryption* and *Ciphertext update*. The attribute authority executes the *System initialization* algorithm to generate public parameters and a master key for the system. Next, a secret key is generated by the attribute authority by running the *User registration* algorithm for each legitimate user based on their attributes. After that, ciphertext generated by the *Ciphertext upload* algorithm based on the access control structure can be uploaded to the cloud server to share data. If a data user wants to use data that is shared by a data owner, he first generates a keyword trapdoor with the *Trapdoor generation* algorithm based on his private key and keyword and uploads the keyword trapdoor to the server. After receiving the request, the cloud server checks whether the ciphertext containing the keyword trapdoor exists by calling the *Ciphertext retrieval* algorithm. If it exists, the ciphertext is returned to the data user. Then, the data user can decrypt the information with the *Ciphertext decryption* algorithm if his attributes satisfy the access control structure. In addition, when the cloud service receives the new revocation list from the data owner, the server updates the ciphertext with the *Ciphertext update* algorithm. The concrete implementation is as follows:

4.1. System Initialization

The attribute authority runs the *Setup* algorithm according to the system model definition. It runs the group generation algorithm to get $G = (< G_1 >, \ldots, < G_{d+3} >)$. Then, it selects a random number, $\alpha, \beta, a \in Z_p$ and a hash function, $H_1 : \{0, 1\}^* \rightarrow G_1$, $H_2 : \{0, 1\}^* \rightarrow G_1$. Finally, the PP and MSK are as follows:

$$PK = (G, g_{d+3}^\alpha, g_1^\beta, g_1^a, H_1, H_2).$$

$$MSK = (\alpha, \beta).$$

4.2. User Registration

At the user registration stage, the interaction between the attribute authority and the system user is as shown Figure 3—when the attribute authority receives the user's attributes, S, and identity, id, the *Keygen* algorithm is called and returns the secret key, SK, to the system user safely.

The concrete algorithms are as follows: Suppose that the path of id is $path(id) = \{p_{id,0}, \ldots, p_{id,d}\}$, where $p_{id,0} = root$ and $p_{id,d} = x$. The algorithm sets $P_{id,-1} = g_1^a$. Then, it calls the following recursive algorithm: $P_{id,j} = e(H_2(P_{id,j}), P_{id,j-1})$, for $j \in [0,d]$, $P_{id,j} \in path(id)$. Then, for $\forall x_i \in S$, it randomly selects $r_i \in Z_p$. In addition, a random number, $r' \in Z_p$, is selected. Finally, it calculates $r = \sum_{i=1}^{n} r_i$, $K_0 = g_{d+2}^{\frac{\alpha+r}{\beta}} \cdot P_{id,d}^{r'}$, $K_1 = g_1^{r'}$, $K_2 = g_1^{\beta r'}$, $k_3 = \prod_{i=1}^{n} H_1(x_i)^{\beta}$, $\{K_i = g_{d+2}^{r_i} \cdot H_1(x_i)^{r'}\}_{x_i \in S}$. The secret key is $SK_S = \{K_0, K_1, K_2, K_3, \{K_i\}_{x_i \in S}\}$.

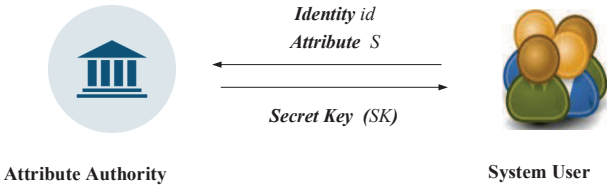

Figure 3. User registration.

4.3. Ciphertext Uploading

At the ciphertext uploading phase, the interaction between the cloud server and the data owner is as shown as Figure 4: The data owner calls the *Encryption* algorithm and then uploads the ciphertext, CT, and keyword index, I, to the cloud server.

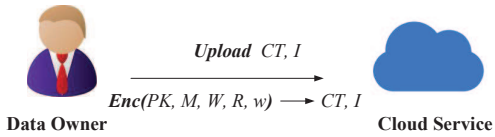

Figure 4. Ciphertext uploaded.

The concrete algorithms are as follows: The algorithm randomly selects $s, s' \leftarrow Z_p^*$ and then calculates $C_0 = M \cdot g_{d+3}^{\alpha \cdot s}$, $C_1 = g_1^s$, $C_2 = g_1^{\beta s}$, $C_{1,i} = H_1(W_i)^s$, $\tilde{C}_1 = e(g_1^{\beta}, g_1^{ws'})$, $\tilde{C}_2 = g_1^{ss'}$.

Suppose the path of element $x \in cover(R)$ is $path(x) = (p_{x,0}, \ldots, p_{x,depth(x)})$, where $p_{x,0}$ represents $root$ and $p_{x,depth(x)} = x$. Then, the algorithm sets $P_{x,-1} = g_1^a$. Finally, it calls the recursive algorithm $P_{x,j} = e(H(P_{x,j}), P_{x,j-1})$ for $j \in [0,d]$ and calculates $C_{2,i} = P_{x,depth(x)}^s$. The ciphertext and keyword index are as follows:

$$CT_{W,R} = (C_0, C_1, C_2, \{C_{1,i}, C_{2,i}\}).$$

$$I_w = (\tilde{C}_1, \tilde{C}_2, \{C_{1,i}\}).$$

4.4. Trapdoor Generation

At the trapdoor generation phase, the interaction between the cloud server and the data user is as shown as Figure 5: The data user calls the algorithm *Trapdoor*, and then uploads the keyword trapdoor, T, to the cloud server.

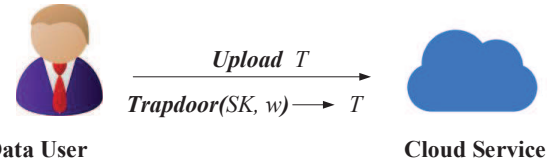

Figure 5. Trapdoor generation.

The data user generates the keyword trapdoor with the following formula:

$$t_w = e(K_3, g_1^w) = e(\prod_{i=1}^{n} H_1(x_i)^\beta, g_1^w).$$

No information about w can be obtained from t_ω.

4.5. Ciphertext Retrieval

The cloud server runs the *Test* algorithm according to the definition of the system model. It retrieves the file containing the keyword w with the following formula:

$$e(\tilde{C}_2, t_w) = e(\prod_{i=1}^{n} C_{1,i}, \tilde{C1}).$$

When the equation is correct, it returns 1. The file exists on the cloud server. When the equation is wrong, it returns 0. The file does not exist on the cloud server.

The correctness of the phase *Ciphertext retrieval* is verified as follows:

$$\begin{aligned} e(\tilde{C_2}, t_w) &= e(g_1^{ss'}, e(\prod_{i=1}^{n} H_1(x_i)^\beta, g_1^w)) \\ &= e(\prod_{i=1}^{n} H_1(x_i)^s, e(g_1^{s'w}, g_1^\beta)) \\ &= e(\prod_{i=1}^{n} C_{1,i}, \tilde{C_1}). \end{aligned}$$

4.6. Ciphertext Decryption

At the ciphertext decryption stage, the interaction between the cloud server and the data user is as shown as Figure 6: The data user calls the *Decrypt* algorithm. If the user is legal, the ciphertext will be deciphered.

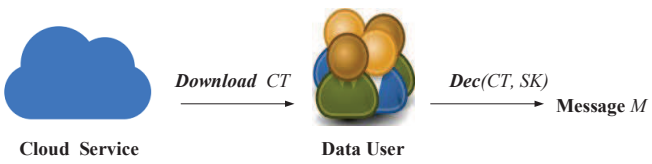

Figure 6. Ciphertext decryption.

The concrete algorithms are as follows: If $\exists i \in [1, \ldots, n]$ $x_i \neq W_i$, the attribute list S does not satisfy the access control structure. The algorithm returns $\perp$. When $id \in R$, The algorithm outputs $\perp$. Otherwise, it calculates the following process.

If id does not belong to R, there will be a node, $x \in (path(id) \cap cover(R))$, where $path(x) = (p_{x,0}, \ldots, p_{x,depth(x)})$ and $path(id) = (p_{id,0}, \ldots, p_{id,d})$. At the same time, there is $p_{id,j} = p_{x,j}$ for $j \in [0, \ldots, depth(x)]$.

The algorithm sets $P_{id,depth(x)}{}' = P_{x,depth(x)} = C_{2,x}$. Then, it calls the recursive algorithm $P_{id,j}{}' = e(H_2(P_{id,j}), P_{id,j-1}{}')$ for $j \in [depth(x_i) + 1, \ldots, d]$. The equation $P_{id,d}{}' = P_{id,d}^s$ can be obtained.

Then, it calculates

$$\frac{e(K_0, C_2)}{\prod_{i=1}^{n} \frac{e(K_i, C_1)}{K_1, C_{1,i}} \cdot e(P_{id,d}{}', K_2)} = g_{d+3}^{\alpha s}.$$

Finally, the following formula is used to get the plaintext:

$$M = \frac{C_0}{g_{d+3}^{\alpha s}}.$$

The correctness of the *Ciphertext decryption* phase is verified as follows:

$$\begin{aligned}\prod_{i=1}^{n} \frac{e(K_i, C_1)}{e(K_1, C_{1,i})} &= \prod_{i=1}^{n} \frac{e(g_{d+2}^{r_i} H_1(x_i) r', g_1^s)}{e(g_1^{r'}, H_1(w_i))^s} \\ &= \prod_{i=1}^{n} e(g_{d+2}^{r_i}, g_1^s) \\ &= g_{d+3}^{sr}.\end{aligned}$$

$$\begin{aligned}\frac{e(K_0, C_2)}{\prod_{i=1}^{n} \frac{e(K_i, C_1)}{e(K_1, C_{1,i})} \cdot e(p_{id,d}{}', K_2)} &= \frac{e(g_{d+2}^{\frac{\alpha+r}{\beta}} \cdot P_{id,d}^{r'}, g_1^{\beta s})}{g_{d+3}^{sr} \cdot e(p_{id,d}^{s}, g_1^{\beta r'})} \\ &= g_{d+3}^{\alpha s}.\end{aligned}$$

4.7. Ciphertext Update

When the revocation is changed, the ciphertext stored on the cloud server will be updated. The cloud server runs the *Update* algorithm according to the definition of the system model. It inputs a ciphertext, $CT_{W,R}$, and a new revocation list, R', where $R \subset R'$ outputs the updated ciphertext, $CT_{W,R'}$.

If $x \in Cover(R)$, $x = y$ for $y \in Cover(R')$. ${C_{2,i}}' = C_{2,i}$ is set.

For $x \in Cover(R)$, y is a child of x. Let $path(y) = path(x) \bigcup (p_{y,depth(x)+1}, \ldots, p_{y,depth(y)})$ and set ${p_{y,depth}}' = {p_{x,depth}}' = C_{2,i}$. Then, it calls the recursive algorithm ${P_{y,j}}' = e(H_2(P_{y,j}), P_{y,j-1})$ for $j \in [depth(x)+1, \ldots, depth(y)]$. Finally, it sets ${C_{2,i}}' = {P_{y,depth(y)}}'$, ${C_0}' = C_0$, ${C_1}' = C_1$, ${C_2}' = C_2$, and ${C_{1,i}}' = C_{1,i}$. The updated ciphertext is $CT_{W,R} = ({C_0}', {C_1}', {C_2}', \{{C_{1,i}}', {C_{2,i}}'\})$.

5. Security Proof

Theorem 1. *The ABERS scheme is the IND-CPA security under $(d+3)$-MDDH assumption in the random oracle model.*

If the adversary, A, can break through our scheme with an advantage that we cannot ignore, a simulator, B, can call the Adversary, A, to break the $(d+3)$-$MDDH$ assumption.

Simulator B inputs the group parameters, $(1^\gamma, n)$, and instantiates the $(d+3)$-$MDDH$ instance $(g_1, g_1^{a_0}, \ldots, g_2^{a_{d+3}}, Z)$. The game between the simulator B and the attacker A is as follows:

Setup: Adversary A selects a revocation list, R^*, and sends it to B. For each element, $id \in R^*$, in the revocation list, R^*, the simulator B sets $P_{R^*} = \{p_{id,i} \in path(id)\}_{id \in R^*, i \in [0,\ldots,d]}$ and the hash functions H_1, H_2 are simulated as followed:

- $\mathbb{O}_{H_1}$: When H_1 is called by the adversary, A (or B), a random number, $z_i \in Z_p$, is selected (unless it has already been done), and the simulator returns $g_1^{z_i}$ as a response to $H_1(x_i)$.
- $\mathbb{O}_{H_2}$: When $p_{id,i} \in P_{R^*}$, H_2 is called by the adversary, A (or B), and a random number, $v_{id,i} \in Z_p$, will be selected (if it has already been done, the same result will be returned), and the simulator returns $g_1^{a_i + v_{id,i}}$ as a response to $H_2(p_{id,i})$.
- When $p_{id,i} \notin P_{R^*}$, H_2 is called by the adversary, A (or B), a random number, $v_{id,i} \in Z_p$, will be selected (if it has already been done, the same result will be returned), and the simulator returns $g_1^{v_{id,i}}$ as a response to $H_2(p_{id,i})$.

The challenger, B, randomly selects the random number, $\alpha, \beta, a \leftarrow Z_p$, and calculates $g_{d+3}^{\alpha}, g_1^{\beta}, g_1^{a}$ and then returns $(G, g_{d+3}^{\alpha}, g_1^{\beta}, g_1^{a}, \mathbb{O}_{H_1}, \mathbb{O}_{H_2})$ to A.

*Phase*1&2: The adversary A makes the following enquiries to the challenger.

- When $id \notin R^*$, the enquiry is aborted.
- When $id \in R^*$, if A asks the challenger about the secret key of the user's identity, id, and attributes, $S = (x_1, x_2, \ldots, x_n)$, random numbers, $r'_j \in Z_p$ and $r_i^j \in Z_p \forall x_i \in S$, will be selected. Then, the simulator B calculates $r^j = \sum_{i=1}^{n} r_i^j$, $D = g_{d+2}^{\frac{\alpha + r^j}{\beta}}$, $K_1 = g_1^{r_j'}$, $K_2 = g_1^{\beta r_j'}$, $K_3 = \prod_{i=1}^{n} g_1^{z_i \beta}$ and $K_i = g_1^{r_i^j + z_i r_j'}$.
- The path of id is represented as $path(id) = (p_{id,0}, p_{id,d})$ and then $H_2(p_{id,i}) = g_1^{(a_i + v_{id,i})}$. After that, B computes $p_{id,d} = g_{d+2}^{a \prod_{i=0}^{d} (a_i + v_{id,i})}$ by calling multi-linear maps on $g_1^{b}, g_1^{a_0 + v_{id,d}}, \ldots, g_1^{a_d + v_{id,d}}$ and $K_0 = g_{d+2}^{\frac{\alpha + r^j}{\beta}} \cdot g_{d+2}^{b \prod_{i=0}^{d} (a_i + v_{id,i})}$.
- Finally, the secret key, $\{K_0, K_1, K_2, K_3, \{K_i\}_{x_i \in S}\}$, is returned to A.

Challenge: The adversary A sends two messages, m_0^*, m_1^* ($|m_0^*| = |m_1^*|$), and a challenge access structure, W, to B, B randomly selects $b \in \{0, 1\}$, and the encryption process is as follows: $C_0 = m_b^* \cdot Z$, ($C_1 = g_1^{a_{d+3}}$, $C_2 = g_1^{\beta a_{d+3}}$, $C_{1,i} = g_1^{t_i a_{d+3}}$. In addition, $P_{x,d}$ is generated according to the specified algorithm. $C_{2,i} = P_{x,d}^{a_{d+3}}$ is set. Finally, $(C_0, C_1, C_2, \{C_{1,i}, C_{2,i}\})$ is sent to A.

Guess: $b' \in \{0, 1\}$ is output by A.

When $Z = g_{d+3}^{\prod_{j \in [0, \ldots, d]} a_j}$, A plays the security game with B. When Z is a random number in a group, G_{d+3}, the information that C_0 contains m_b^* is lost. Therefore, the simulator, B, can call the A to break the $(d+3)$-$MDDH$ assumption. Because the assumption is difficult, our scheme is secure.

Theorem 2. *Suppose q is a bound on the total number of group elements in the INK-CKA security game. The advantage in this security game is $O(q^2/p)$.*

Simulator B inputs the group parameters $(1^{\gamma}, n)$ and instantiates the $(d+3)$-$MDDH$ instance $(g_1, g_1^{a_0}, \ldots, g_2^{a_{d+3}}, Z)$. The game between the simulator B and the attacker A is as follows:

Setup: The hash function, H_1, is simulated as follows:

$\mathbb{O}_{H_1}$: When H_1 is called by the adversary, A (or B), a random number, $z_i \in Z_p$, will be selected (unless it has already been done), and the simulator returns $g_1^{z_i}$ as a response to $H_1(x_i)$.

The challenger, B, randomly selects the random number, $\alpha, \beta, a \leftarrow Z_p$, and calculates $g_{d+3}^{\alpha}, g_1^{\beta}, g_1^{a}$ and then returns $(G, g_{d+3}^{\alpha}, g_1^{\beta}, g_1^{a}, \mathbb{O}_{H_1})$ to A.

*Phase*1: The adversary, A, makes the following enquiries to the challenger.

The adversary A asks for the keyword, w, connected with $S = (x_1, x_2, \ldots, x_n)$ and the user's identity, id, for B. The random numbers $r'_j \in Z_p$ and $r_i^j \in Z_p \forall x_i \in S$ will be selected. Then, the simulator, B, calculates $r^j = \sum_{i=1}^{n} r_i^j$, $D = g_{d+2}^{\frac{\alpha + r^j}{\beta}}$, $K_1 = g_1^{r_j'}$, $K_2 = g_1^{\beta r_j'}$, $K_3 = \prod_{i=1}^{n} g_1^{z_i \beta}$ and $K_i = g_1^{r_i^j + z_i r_j'}$.

Finally, the simulator B produces trapdoor t_w as $t_w = e(K_3, g_1^{w}) = e(g_1^{\beta \sum_{i=1}^{n} z_i}, g_1^{w})$. After that, the trapdoor t_w is sent to A.

Challenge: The adversary, A, sends two keywords, w_0^*, w_1^* ($|w_0^*| = |w_1^*|$) to B. At the same time, the challenge access control structure, W, will also be sent. B randomly selects $s, s' \in Z_p$ and $b \in \{0, 1\}$, and the encryption process is as follows: $C_{1,i} = g_1^{z_i s}$, $\tilde{C_1} = e(g_1^{\beta}, g_1^{w_b^* s'})$, $\tilde{C_2} = g_1^{ss'}$. The challenge index, I_w^*, is sent to A.

Phase 2: This stage is the same as *Phase 1*, but there is the restriction that the trapdoors of generated attributes that satisfy the access control policy have not been queried before.

Guess: $b' \in \{0,1\}$ is output by A.

The Schwartz–Zipple lemma [56] points out that the probability of an "unexpected collision" occurring is, at most, $O(q^2/p)$.

Theorem 3. *The ABERS scheme achieves selective security on updated ciphertext under the* $(d+3)$*-MDDH assumption in the random oracle model.*

We can see that any polynomial time adversary can not learn any information from the original ciphertext under Theorem 1. The key to proving Theorem 3 is determining whether the original ciphertext is distinguishable from the updated ciphertext.

Now let us take a look at whether the original ciphertext and the updated ciphertext generated by the same message, the attribute set, S, and the revocation list, R', are uniformly distributed.

The original ciphertext generated by calling $Encrypt(PP, M, W, R', w)$ is

$$CT_{R'} = (C_0, C_1, C_2, \{C_{1,i}, C_{2,i}\}),$$

where $C_0 = M \cdot g_{d+3}^{\alpha \cdot s}$, $C_1 = g_1^s$, $C_2 = g_1^{\beta s}$, $C_{1,i} = H_1(W_i)^s$ and $C_{2,i} = P_{x,depth(x)}^s$.

The original ciphertext generated by calling $Encrypt(PP, M, W, R, w)$ is

$$CT_R = (C_0, C_1, C_2, \{C_{1,i}, C_{2,i}\}),$$

where $C_0 = M \cdot g_{d+3}^{\alpha \cdot s^*}$, $C_1 = g_1^{s^*}$, $C_2 = g_1^{\beta s}$, $C_{1,i} = H_1(W_i)^{s^*}$ and $C_{2,i} = P_{x,depth(x)}^{s^*}$.

The updated ciphertext generated by calling $Update(CT_R, R')$ is

$$CT_{R''} = (C_0', C_1', C_2', \{C_{1,i}', C_{2,i}'\}),$$

where $C_0 = M \cdot g_{d+3}^{\alpha \cdot s^*}$, $C_1 = g_1^{s^*}$, $C_2 = g_1^{\beta s}$, $C_{1,i} = H_1(W_i)^{s^*}$ and $C_{2,i} = P_{x,depth(x)}^{s^*}$ for $\forall id \in (Cover(R) \bigcap Cover(R'))$, $C_{2,i}' = C_{2,i} = P_{x,depth(x)}^{s^*}$, and $\forall id \in (Cover(R') - Cover(R))$, $C_{2,i}' = P_{x,depth(x)}^{s^*}$.

The original ciphertext and the updated ciphertext have the same terms, and each term is blinded by random numbers. Therefore, the original ciphertext and the updated ciphertext have the same distribution. At this point, similar to the analysis in [43], if the adversary, A, can break through our scheme, the simulator will be able to break the $(d+3)$*-MDDH* assumption.

6. Comparison

In this section, we compare our scheme with some related schemes. We have chosen several representative solutions related to the keyword search of ciphertext [42,52,57] and direct revocation [43,57,58]. The results of the comparison are shown in Table 1. Table 1 compares the functional differences between our schemes and related schemes from the perspective of keyword search, fast keyword search, direct revocation, hidden policy, communication overhead and storage overhead. Compared with other schemes, our scheme has better function. It is more accurate than the scheme [52]. The communication cost of the keyword trapdoor is the same, but the functioning of our scheme is greater. Compared with other schemes, the storage cost of our scheme is not very large.

Next, we compare the efficiency of the keyword search. In order to exclude other sources of interference and to make the result more accurate, we tested the schemes on the same platform, and the test results are shown in Figure 7. Figure 7 compares our scheme's search efficiency with refs. [42,52,57]. We can see that, compared with schemes [42] and [57], the keyword search efficiency in our scheme is very high. The search time cost does not increase linearly with the number of attributes in ciphertext policies, which is not enabled in [42,57]. This is because our search scheme uses aggregated search key technology without pairing the secret key components with the corresponding ciphertext components. In the process of keyword trapdoor generation, only one linear pair operation is needed. In the process

of ciphertext retrieval, by comparing whether the results of two pairs of linear pairs are equal, we can determine whether the required ciphertext exists. Although our scheme has the same efficiency in the search phase as that shown in reference [52], our scheme is more functional. From the point of view of function and efficiency, our scheme is more applicable to the practical environment.

Table 1. Feature comparison of our scheme and other typical schemes †.

Scheme	KS	FKS	DR	HP	CO	SO
[58]	×	×	√	×	−	$(\|L\|+\|C\|+2)\|G\|$
[43]	×	×	√	√	−	$(\|S\|+\|C\|+2)\|G\|$
[42]	√	×	×	√	$(2\|S\|+\|Z\|+1)\|G\|$	$(\|P\|+\|P\|\|W\|+2)\|G\|$
[52]	√	√	×	√	$\|G\|$	$(2\|S\|+5)\|G\|$
[57]	√	√	√	×	$(\|N\|+3\|L\|+\|I\|)\|G\|$	$(2\|S\|-2\|R\|+\|W\|+\|M\|+3)\|G\|$
Our scheme	√	√	√	√	$\|G\|$	$(2\|S\|+\|C\|+2)\|G\|$

† The symbol √ (resp. ×) represents the corresponding feature is (resp. is not) achieved in the scheme. KS means keyword search, FKS means fast keyword search, DR means direct revocation, HP means hidden policy, CO means communication overhead and SO means storage sverhead. |S| means the number of user attributes, |Z| means the bit length of an element of Z_p, |G| means the bit length of an element of G_i, |I| means the bit length of user ID, |L| means the number of rows of the access control matrix, |P| means the number of columns of the access control structure, |C| means the cardinality of cover(R), |R| means the cardinality of a revocation list, |M| means the maximum number of revoked users and |N| means the number of keywords.

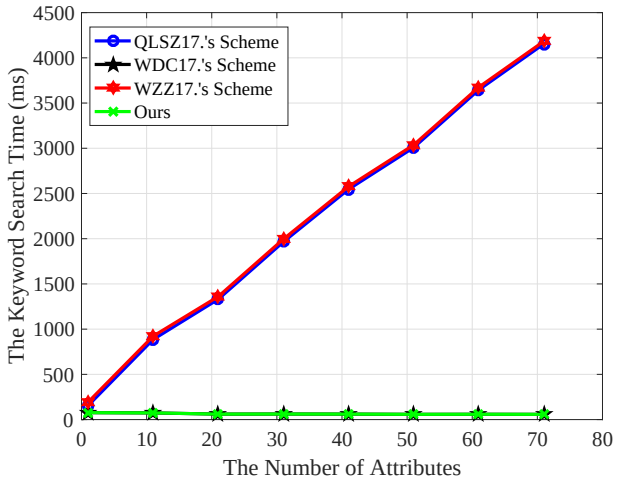

Figure 7. The comparison of keyword search performance.

7. Conclusions and Future Work

In this article, we have put forward a hidden policy attribute-based data sharing scheme with direct revocation and keyword search. The scheme has the following advantages. First, it uses subset covering theorem to achieve the direct revocation of attributes. After revocation, there is no need to update the private key of a non-revoked user. In order to ensure that the users who have been revoked cannot decrypt the previous ciphertext, the ciphertext is updated. In this way, some secret keys do not match some ciphertext, and users who are revoked can not decrypt the previous ciphertext. In addition, when there is a user leaving the system, we just need to send the revocation list to the cloud server and let the cloud server update the ciphertext. Then, the private key of the non-revoked user does not need to be updated. Second, we use aggregation technology to achieve the fast search of keywords. In the process of keyword trapdoor generation, only one linear pair operation is needed. In the process of ciphertext retrieval, by comparing whether the results of two pairs of linear pairs are equal, we can determine whether the required ciphertext exists. So, the keyword search time is constant and does not increase with the number of attributes. Third, the AND gate access control structure is used to

achieve the hidden policy. When ciphertext is uploaded, the access control structure does not need to be uploaded. Thus, the function of the hidden policy can be achieved. In brief, when a lot of data is being shared, our solution can provide a good solution.

When a user leaves the system, the user needs to interact with the cloud server, and then, the server updates the ciphertext. This not only increases the cost of communication and computing, but the revoked user can decrypt the former ciphertext before the ciphertext is updated which is a threat to the security of the system. If there is no need to update the ciphertext, the revoked user will not be able to decrypt the ciphertext at the moment of revocation. So, in future work, we will solve the problem of how to ensure that the user can not decrypt the previous ciphertext without updating the ciphertext.

Author Contributions: Formal analysis, Y.Z., A.W., D.Z. and M.Y.; Methodology, A.W. and D.Z.; Supervision, D.Z.; Writing—original draft, A.W. and M.Y.; Writing—review & editing, Y.Z.

Funding: This research was funded by National Key R&D Program of China under grant number 2017YFB0802000; National Natural Science Foundation of China under grant number 61772418, 61472472 and 61402366; Natural Science Basic Research Plan in Shaanxi Province of China under grant number 2018JZ6001 and 2015JQ6236. Yinghui Zhang is supported by New Star Team of Xi'an University of Posts and Telecommunications under grant number 2016-02.

Acknowledgments: We are grateful to the editors and referees for their invaluable suggestions for improving the paper.

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

References

1. Shen, J.; Wang, C.; Li, T.; Chen, X.; Huang, X.; Zhan, Z.H. Secure Data Uploading Scheme for a Smart Home System. *Inf. Sci.* **2018**, *453*, 186–197.10.1016/j.ins.2018.04.048. [CrossRef]
2. Jhaveri, R.H.; Patel, N.M.; Zhong, Y.; Sangaiah, A.K. Sensitivity Analysis of an Attack-Pattern Discovery based Trusted Routing Scheme for Mobile Ad-Hoc Networks in Industrial IoT. *IEEE Access* **2018**, *6*, 20085–20103.10.1109/ACCESS.2018.2822945. [CrossRef]
3. Zhang, X.; Chen, X.; Wang, J.; Zhan, Z.; Li, J. Verifiable privacy-preserving single-layer perceptron training scheme in cloud computing. *Soft Comput.* **2018**, 1–14.10.1007/s00500-018-3233-7. [CrossRef]
4. Li, P.; Li, T.; Ye, H.; Li, J.; Chen, X.; Xiang, Y. Privacy-preserving machine learning with multiple data providers. *Future Gener. Comput. Syst.* **2018**.10.1016/j.future.2018.04.076. [CrossRef]
5. Zhang, X.; Tan, Y.A.; Liang, C.; Li, Y.; Li, J. A Covert Channel over VoLTE via Adjusting Silence Periods. *IEEE Access* **2018**, *6*, 9292–9302. [CrossRef]
6. Liu, Z.; Huang, Y.; Li, J.; Cheng, X.; Shen, C. DivORAM: Towards a Practical Oblivious RAM with Variable Block Size. *Inf. Sci.* **2018**, *447*, 1–11. [CrossRef]
7. Li, J.; Chen, X.; Huang, X.; Tang, S.; Xiang, Y.; Hassan, M.M.; Alelaiwi, A. Secure Distributed Deduplication Systems with Improved Reliability. *IEEE Trans. Comput.* **2015**, *64*, 3569–3579. [CrossRef]
8. Zhang, Y.; Deng, R.H.; Shu, J.; Yang, K.; Zheng, D. TKSE: Trustworthy Keyword Search over Encrypted Data with Two-side Verifiability via Blockchain. *IEEE Access* **2018**, *6*, 31077–31087.10.1109/ACCESS.2018.2844400. [CrossRef]
9. Xu, J.; Wei, L.; Zhang, Y.; Wang, A.; Zhou, F.; Gao, C.Z. Dynamic Fully Homomorphic encryption-based Merkle Tree for lightweight streaming authenticated data structures. *J. Netw. Comput. Appl.* **2018**, *107*, 113–124. [CrossRef]
10. Gao, C.Z.; Cheng, Q.; He, P.; Susilo, W.; Li, J. Privacy-Preserving Naive Bayes Classifiers Secure against the Substitution-then-Comparison Attack. *Inf. Sci.* **2018**, *444*, 72–88. [CrossRef]
11. Shen, J.; Gui, Z.; Ji, S.; Shen, J.; Tan, H.; Tang, Y. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *J. Netw. Comput. Appl.* **2018**, *106*, 117–123. [CrossRef]
12. Lin, Q.; Yan, H.; Huang, Z.; Chen, W.; Shen, J.; Tang, Y. An ID-based linearly homomorphic signature scheme and its application in blockchain. *IEEE Access* **2018**, *6*, 20632–20640. [CrossRef]
13. Wei, L.; Zhu, H.; Cao, Z.; Dong, X.; Jia, W.; Chen, Y.; Vasilakos, A.V. Security and privacy for storage and computation in cloud computing. *Inf. Sci.* **2014**, *258*, 371–386. [CrossRef]

14. Zhang, Y.; Zheng, D.; Deng, R.H. Security and Privacy in Smart Health: Efficient Policy-Hiding Attribute-Based Access Control. *IEEE Int. Things J.* **2018**, *5*, 2130–2145.10.1109/JIOT.2018.2825289. [CrossRef]
15. Zhang, Y.; Deng, R.H.; Liu, X.; Zheng, D. Blockchain based efficient and robust fair payment for outsourcing services in cloud computing. *Inf. Sci.* **2018**, *462*, 262–277.10.1016/j.ins.2018.06.018. [CrossRef]
16. Yu, S.; Wang, C.; Ren, K.; Lou, W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In Proceedings of the 2010 IEEE INFOCOM, San Diego, CA, USA, 14–19 March 2010; pp. 1–9.
17. Li, M.; Yu, S.; Zheng, Y.; Ren, K.; Lou, W. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *24*, 131–143. [CrossRef]
18. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; pp. 457–473.
19. Li, J.; Chen, X.; Chow, S.S.M.; Huang, Q.; Wong, D.S.; Liu, Z. Multi-authority fine-grained access control with accountability and its application in cloud. *J. Netw. Comput. Appl.* **2018**, *112*, 89–96. [CrossRef]
20. Zhang, Y.; Li, J.; Chen, X.; Li, H. Anonymous attribute-based proxy re-encryption for access control in cloud computing. *Secur. Commun. Netw.* **2016**, *9*, 2397–2411. [CrossRef]
21. Shen, J.; Zhou, T.; Chen, X.; Li, J.; Susilo, W. Anonymous and Traceable Group Data Sharing in Cloud Computing. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 912–925. [CrossRef]
22. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
23. Cai, Z.; Yan, H.; Li, P.; Huang, Z.A.; Gao, C. Towards secure and flexible EHR sharing in mobile health cloud under static assumptions. *Cluster Comput.* **2017**, *20*, 2415–2422. [CrossRef]
24. Zhang, Y.; Zheng, D.; Li, Q.; Li, J.; Li, H. Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing. *Secur. Commun. Netw.* **2016**, *9*, 3688–3702. [CrossRef]
25. Zheng, Q.; Xu, S.; Ateniese, G. VABKS: Verifiable attribute-based keyword search over outsourced encrypted data. In Proceedings of the IEEE Conference on Computer Communications IEEE INFOCOM 2014, Toronto, ON, Canada, 27 April–2 May 2014; pp. 522–530.
26. Boldyreva, A.; Goyal, V.; Kumar, V. Identity-based encryption with efficient revocation. In Proceedings of the 15th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 27–31 October 2008; pp. 417–426.
27. Dan, B.; Crescenzo, G.D.; Ostrovsky, R.; Persiano, G. Public Key Encryption with Keyword Search. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; pp. 506–522.
28. Sun, W.; Yu, S.; Lou, W.; Hou, Y.T.; Li, H. Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. In Proceedings of the 2014 IEEE INFOCOM, Toronto, ON, Canada, 27 April–2 May 2014; pp. 226–234.
29. Li, J.; Li, J.; Chen, X.; Jia, C.; Lou, W. Identity-Based Encryption with Outsourced Revocation in Cloud Computing. *IEEE Trans. Comput.* **2015**, *64*, 425–437. [CrossRef]
30. Pirretti, M.; Traynor, P.; Mcdaniel, P.; Waters, B. Secure attribute-based systems. *J. Comput. Secur.* **2010**, *18*, 799–837. [CrossRef]
31. Sahai, A.; Seyalioglu, H.; Waters, B. Dynamic credentials and ciphertext delegation for attribute-based encryption. *Lect. Notes Comput. Sci.* **2012**, *7417*, 199–217.
32. Attrapadung, N.; Imai, H. Conjunctive Broadcast and Attribute-Based Encryption. In Proceedings of the 3rd International Conference on Pairing-Based Cryptography—Pairing 2009, Palo Alto, CA, USA, 12–14 August 2009; pp. 248–265.
33. Goyal, V.; Jain, A.; Pandey, O.; Sahai, A. Bounded Ciphertext Policy Attribute Based Encryption. In Proceedings of the 35th International Colloquium on Automata, Languages, and Programming (ICALP 2008), Reykjavik, Iceland, 7–11 July 2008; pp. 579–591.
34. Ostrovsky, R.; Sahai, A.; Waters, B. Attribute-based encryption with non-monotonic access structures. In Proceedings of the 14th ACM Conference on Computer & Communications Security, Alexandria, VA, USA; 29 October–2 November 2007; pp. 195–203.
35. Wang, H.; Zheng, Z.; Wu, L.; Li, P. New directly revocable attribute-based encryption scheme and its application in cloud storage environment. *Cluster Comput.* **2017**, *20*, 2385–2392. [CrossRef]

36. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 89–98.
37. Zhang, Y.; Wu, A.; Zheng, D. Efficient and privacy-aware attribute-based data sharing in mobile cloud computing. *J. Ambient Intell. Humaniz. Comput.* **2017**, 1–10, doi:10.1007/s12652-017-0509-1. [CrossRef]
38. Zhang, Y.; Chen, X.; Li, J.; Wong, D.S.; Li, H.; You, I. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. *Inf. Sci.* **2016**, *379*, 42–61. [CrossRef]
39. Li, J.; Zhang, Y.; Chen, X.; Xiang, Y. Secure attribute-based data sharing for resource-limited users in cloud computing. *Comput. Secur.* **2018**, *72*, 1–12.10.1016/j.cose.2017.08.007. [CrossRef]
40. Ling, C.; Newport, C. Provably secure ciphertext policy ABE. In Proceedings of the 14th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 29 October–2 November 2007; pp. 456–465.
41. Li, J.; Ren, K.; Zhu, B.; Wan, Z. Privacy-Aware Attribute-Based Encryption with User Accountability. In Proceedings of the 12th International Conference on Information Security (ISC 2009), Pisa, Italy, 7–9 September 2009; pp. 347–362.
42. Qiu, S.; Liu, J.; Shi, Y.; Zhang, R. Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack. *Sci. China (Inf. Sci.)* **2017**, *60*, 1-12. [CrossRef]
43. Shi, Y.; Zheng, Q.; Liu, J.; Han, Z. Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation. *Inf. Sci. Int. J.* **2015**, *295*, 221–231. [CrossRef]
44. Perrig, A.; Wagner, D.; Song, D.X. Practical techniques for searches on encrypted data. In Proceeding of the 2000 IEEE Symposium on Security and Privacy (S & P), Berkeley, CA, USA, 14–17 May 2000; pp. 44–55.
45. Lee, C.C.; Li, C.T.; Chen, C.L.; Chiu, S.T. A Searchable Hierarchical Conditional Proxy Re-encryption Scheme for Cloud Storage Services. *Inf. Technol. Control* **2016**, *45*, 289–299.10.5755/j01.itc.45.3.13224. [CrossRef]
46. Fang, L.; Susilo, W.; Ge, C.; Wang, J. Public key encryption with keyword search secure against keyword guessing attacks without random oracle. *Inf. Sci.* **2013**, *238*, 221–241. [CrossRef]
47. Golle, P.; Staddon, J.; Waters, B. Secure Conjunctive Keyword Search over Encrypted Data. *Lect. Notes Comput. Sci.* **2004**, *3089*, 31–45.
48. Bao, F.; Deng, R.H.; Ding, X.; Yang, Y. Private query on encrypted data in multi-user settings. In Proceeding of the International Conference on Information Security Practice and Experience (ISPEC 2008), Sydney, Australia, 21–23 April 2008; pp. 71–85.
49. Yang, Y.; Lu, H.; Weng, J. Multi-user private keyword search for cloud computing. In Proceeding of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science, Athens, Greece, 29 November–1 December 2011; pp. 758–759.
50. Li, H.; Liu, D.; Jia, K.; Lin, X. Achieving authorized and ranked multi-keyword search over encrypted cloud data. In Proceeding of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 7450–7455.
51. Cao, N.; Wang, C.; Li, M.; Ren, K.; Lou, W. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Trans. Parallel Distrib. Syst.* **2011**, *25*, 829–837.
52. Wang, H.; Dong, X.; Cao, Z. Multi-value-Independent Ciphertext-Policy Attribute Based Encryption with Fast Keyword Search. *IEEE Trans. Serv. Comput.* **2017**, *99*, 1.10.1109/TSC.2017.2753231. [CrossRef]
53. Boneh, D.; Silverberg, A. Applications of Multilinear Forms to Cryptography. *Contemp. Math.* **2003**, *324*, 71–90.
54. Naor, D.; Naor, M.; Lotspiech, J.B. Revocation and Tracing Schemes for Stateless Receivers. *Crypto* **2001**, *2001*, 41–62.
55. Freire, E.S.V.; Hofheinz, D.; Paterson, K.G.; Striecks, C. Programmable hash functions in the multilinear setting. In Proceedings of the 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2013; Volume 8042, pp. 513–530.
56. Schwartz, J.T. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM* **1980**, *27*, 701–717. [CrossRef]

57. Wang, S.; Zhao, D.; Zhang, Y. Searchable attribute-based encryption scheme with attribute revocation in cloud storage. *PLoS ONE* **2017**, *12*, e0183459.10.1371/journal.pone.0183459. [CrossRef] [PubMed]
58. Wang, H.; He, D.; Shen, J.; Zheng, Z.; Yang, X.; Man, H.A. Fuzzy matching and direct revocation: A new CP-ABE scheme from multilinear maps. *Soft Comput.* **2017**, *22*, 2267–2274. [CrossRef]