sensors

MDPI

Article

Two-Factor-Based Public Data Protection Scheme in Smart Ocean Management

Jian Shen [1,2,3], Xinzhao Jiang [1], Youngju Cho [4,*], Dengzhi Liu [1] and Tianqi Zhou [1,2]

[1] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, China; s_shenjian@126.com (J.S.); jxz942@163.com (X.J.); liudzdh@126.com (D.L.); tq_zhou@126.com (T.Z.)
[2] State Key Laboratory of Cryptology, Beijing 100878, China
[3] Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China
[4] SW Convergence Education Institute, Chosun University, Gwanju 61452, Korea
* Correspondence: csjyj@chosun.ac.kr; Tel.: +82-062-230-7754 (ext. 230-7509)

Received: 2 December 2018; Accepted: 26 December 2018; Published: 2 January 2019

check for updates

Abstract: Nowadays, two-factor data security protection has become a research hotspot in smart ocean management. With the increasing popularity of smart ocean management, how to achieve the two-factor protection of public data resources in smart ocean management is a serious problem to be tackled. Furthermore, how to achieve both security and revocation is also a challenge for two-factor protection. In this paper, we propose a two-factor-based protection scheme with factor revocation in smart ocean management. The proposed scheme allows data owners (DOs) to send encrypted messages to users through a shipboard server (SS). The DOs are required to formulate access policy and perform attribute-based encryption on messages. In order to decrypt, the users need to possess two factors. The first factor is the user's secret key. The second factor is security equipment, which is a sensor card in smart ocean system. The ciphertext can be decrypted if and only if the user gathers the key and the security equipment at the same time. What is more, once the security equipment is lost, the equipment can be revoked and a new one is redistributed to the users. The theoretical analysis and experiment results indeed indicate the security, efficiency, and practicality of our scheme.

Keywords: two-factor; public data protection; redistribution; attribute-based cryptography; smart ocean management

1. Introduction

The construction of smart ocean management has become an important trend of smart ocean field. With the increasing requirements for the quality of management, a large number of problems have emerged. On the one hand, it is very significant to structure a secure memory space to ensure the data security (nautical data , ship position data, and ocean resources data, etc.). On the other hand, it is a problem that how to access these data safely and efficiently to meet the needs of navigation department, radar department, electromechanical department, and other departments on the warship. Especially, it has become an urgent problem that access to public data, such as computer rooms, conference rooms, control rooms, and other public resources. However, it is difficult to solve such thorny problems with traditional data processing schemes. Therefore, information and communication technologies to tackle the problem of smart ocean management emerge at a historic moment [1,2].

Because smart ocean management involves a large amount of complex data, for instance, confidential navigation roadmaps, high-dimensional remote sensing satellite image, and marine resource distribution maps, therefore, cloud computing technology is usually applied to shipboard server (SS). Cloud computing is an innovative change in smart ocean management. Compared with traditional computing technology, cloud computing has a lot of advantages in terms of storage and calculation of ocean data, therefore, it has received extensive interest from the academic community [3–5]. By using cloud computing technology in smart ocean management, users can not only get limitless storage space, but also get limitless computing resources [6]. Server data storage [7–9] is a branch of distributed storage model, which is one of the most significant applications in smart ocean management. SS storage has many advantages, and the most extensive thing is data availability. However, the research on public data access is not too comprehensive, which refers to the situation where many users access public data. An occasion is smart ocean management as mentioned above. When acquiring important public resources, for example, crew and captain read confidential documents, security equipment and secret keys (two factors) are essential for users. Security equipment and secret keys are distributed by the central authority (CA) which communicates with the ship through a secure node transmission channel.

To prevent data from being stolen, researchers mostly design single public key encryption schemes [10–12]. In general public key encryption schemes, the secret key required for decryption is often related to the public key. The key is generally stored in a private device or a trusted third party (TTP). If there is no malicious attack, this kind of secret key storage scheme is secure enough. Unfortunately, it often goes against one's wishes. When being accessed through the Internet, private devices, and TTP are most likely to be attacked by illegal hackers, resulting in the secret key being stolen. But for all these, the users know nothing. In addition, please consider the following two real-life work occasions: Personal computers that store a user's secret key may be used by others in cockpits, laboratories, and offices, public computers that record users' login information will be shared by different users. Under these circumstances, the secret keys are able to be compromised by some malicious attackers who use technical or non technical means. Therefore, single secret key encryption scheme no longer satisfies certain security requirements in smart ocean management, and the two-factor data protection scheme arises at the historic moment.

We note that there are some other research works on two factors, such as [13]. Nevertheless, the two factors in their works are different from ours, they leverage different technologies to design different systems. Here, we will not compare them with our present scheme.

As far as we know, ref. [14], for the first time, provides two-factor data protection to support security device revocability. While the shceme in [14] is actually an identity-based scheme, and ciphertext can be decrypted by only one user rather than a group of users. Therefore, ref. [14] is only a solution to the storage problem of single user data, but it is not suitable for protecting public data in smart ocean management. In the era of shared economy, it is more worthwhile to study the privacy protection of public data.

1.1. Our Contribution

Inspired by [14], we propose a two-factor public data protection scheme in smart ocean management. Note that message validation is presented to verify the correctness of the message after decrypting. The contributions of our research are as follows:

- A practical attribute-based encryption (ABE) data protection scheme is proposed. In practical applications, public resources are more likely to be shared by many users. The security of these public resources is of great significance. In our scheme, we take advantage of an efficient ABE to address this problem. Ensure that only users who satisfy specific attributes can access public data.
- A two-factor-based data protection scheme that supports revocation in smart ocean management is proposed. We apply this kind of data protection scheme to smart ocean management for the first time. Ocean-related data is often highly confidential, however, a single factor scheme does not meet the security requirements of the application layer. Therefore, we design a two-factor-based data protection scheme. In addition, when performing missions in the ocean environment, the sensor cards of the crew will inevitably be lost, for example slipping into the sea. Here,

the revocation of the sensor cards is particularly important. Therefore, the revocation and redistribution of security equipment is also supported by our scheme.

- A scheme that security equipment is independent of secret key is proposed. In our scheme, the distribution and update of security equipment is separated from secret key. Therefore, the threat to secret key due to the loss of the security equipment can be effectively reduced.
- A SS security scheme is proposed. As is known to all, SS is a semi-trusted entity. However, in our scheme, SS cannot decrypt any ciphertext. At the same time, users can also complete the correctness verification of decrypted message.
- More formal and complete security and performance analysis are proposed. According to the designed security model, descriptive language and rigorous mathematical proof are used in security analysis. The attacker's success is reduced to the resolution of difficult problems. The security of the scheme is proved probabilistically. In performance analysis, similar schemes are compared with ours from different phases, thus achieving a more fine-grained comparison.

1.2. Related Works

In recent years, emerging technologies are booming and are reflected in many fields. In particular, the application and development of space and ocean. In 2018, Kim and Ben-Othman [15] proposed a surveillance model for multi domain IoT environment, which is supported by reinforced barriers with collision-avoidance using heterogeneous smart unmanned aerial vehicles. In the field of oceans, in order to solve some problems in data collection of underwater acoustic sensor networks (UASNS), Han et al. [16] proposed a stratification-based data collection scheme for three-dimensional UASNs. In addition, to achieve accurate and energy efficient trust evaluation in UASNs, an attack-resistant trust model based on multidimensional trust metrics is also proposed by Han et al. [17].

In addition to the research directions mentioned above, research on SS also attracts the attention of academic groups at home and abroad. Among them, storage is a major research hotspot in SS. Various sub fields of storage, for instance, data encryption [18], location detection [19], privacy protection, and data sharing [20,21] are the subjects of intense discussion as well.

In 2005, Sahai and Waters [22] first put forward fuzzy identity-based encryption (IBE) and further discussed in [23], which are the original work of ABE. After that, two variants of ABE were proposed. They are key-policy ABE (KP-ABE) [24] and ciphertext-policy ABE (CP-ABE) [25,26] respectively. The difference between them is that a given policy is associated with a key or a ciphertext. While the CP-ABE is the opposite of KP-ABE, and CP-ABE has a more important practical value. Later, lots of CP-ABE schemes with specific features are introduced by researchers. In 2008, Boldyreva et al. [27] proposed an identity-based encryption scheme that supports efficient revocation operation. Now, the scheme has been applied in KP-ABE. In 2010, Yu et al. [28] provided a CP-ABE scheme with attribute revocation. However, the length of private key and ciphertext are positively related to the number of required attributes. In addition, all attributes must be involved in key generation, encryption and decryption. Therefore, the scheme greatly increases overhead of computing and communication. Of course, in addition to revocability, there are many ABE scheme with other features. For example, schemes [24–26,29–32] that require a fully trusted authority. Nevertheless, by leveraging above schemes can only achieve access control but not two-factor data protection, let alone revocation of factor in public data occasion.

In 2017, Shivanna et al. [33] presented a double encryption privacy protection scheme. However, double encryption has a lot of defects. Loss of security equipment causes ciphertext to never be decrypted. Furthermore, this scheme makes the encryption process more complicated. In 2002, Dodis et al. [34] first designed a key-insulated public key scheme to solve the problem of private key exposure. In 2003, Dodis et al. [35] applied key insulated technology to digital signatures. However, in [34], there is a great correlation between the master key and the public key. What is more, in [35], frequent update of private key may lead to compromise of master secret key. For enhancing security of the master key, in 2006, Hanaoka et al. [36] introduced the parallel key insulated public key encryption scheme. But the security of [36] is proved under random oracle model. In 2007, Quisquater et al. [37] proposed a parallel key-insulated public key encryption scheme in the standard model. In 2016, Wang et al. [38] redesigned attribute-based data sharing mechanism to solve the key escrow problem. Simultaneously, weighted attributes in access policies are constructed by [38] to improve the expression of scheme. However, the two-factor feature is not supported by [38], let alone equipment revocation. In most of the above schemes, update of user's private key requires the participation of security equipment. This is obviously not suitable for the protection of public data. Because we hope that the user's private key is updated occasionally, and the security equipment is separated from the private key.

In contrast to the above schemes, we design a two-factor-based public data protection scheme in smart ocean management by utilizing an efficient ABE and public key encryption. Importantly, compared with [14], the proposed scheme can achieve protection of public resources, not just personal data.

1.3. Organization

The rest of this paper is made up of the following sections. Section 2 introduces some preliminary knowledge in cryptographic so that make it easier for readers to understand our works. The system model, notations, security model and system components are illustrated in Section 3. Section 4 describes our scheme in detail. Sections 5 and 6 show the security and performance analyses, respectively. Finally, Section 7 summarizes this paper.

2. Preliminaries

Before introducing our scheme in detail, it is inevitably to introduce some preliminaries in this section, including the bilinear maps, computational assumption, and attribute-based encryption.

2.1. Bilinear Maps

First, we define an algorithm, then input a security parameter k. Then, the algorithm outputs some parameters related to bilinear map, that is $(q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, e)$, where $\mathbb{G}_1$,$\mathbb{G}_2$ are two multiplicative cyclic groups with prime order $q \in \Theta(2^k)$ and g_1,g_2 are generator of $\mathbb{G}_1$. A bilinear maps $e : \mathbb{G}_1^2 \rightarrow \mathbb{G}_2$ is efficient map when it satisfies the following three properties:

Bilinearity. For all g_1,$g_2 \in \mathbb{G}_1$ and $a, b \in_R Z_q^*$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$;

Non-degeneracy. $\exists g_1$,$g_2 \in \mathbb{G}_1$, there is $e(g_1, g_2) \neq 1_{\mathbb{G}_2}$. Where $1_{\mathbb{G}_2}$ represents the unit element of $\mathbb{G}_2$;

Computability. $\forall g_1, g_2 \in \mathbb{G}_1$, there is at least an efficient algorithm to compute $e(g_1.g_2)$.

2.2. Computational Assumption

q-weak Decision Bilinear Diffie-Hellman Inversion (q-wDBDHI) Assumption. For an algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ decides the q-wDBDHI is ξ. For the following equation:

$$\begin{aligned} &|Pr[\mathcal{A}(g, g^a, \ldots, g^{a^q}, g^b, e(g, g)^{b/a}) = 1] \\ &- Pr[\mathcal{A}(g, g^a, \ldots, g^{a^q}, g^b, e(g, g)^z) = 1]| = \xi \end{aligned} \tag{1}$$

there $a, b, z \in_R Z_q^*$. q-wDBDHI assumption holds when ξ is negligible for any polynomial time algorithm.

2.3. Ciphertext-Policy Attribute-Based Encryption

CP-ABE is a cryptography technology for realizing one to many secure communication, where the DO shares message to specific users by constructing an access policy and embedding the policy into ciphertext. The most primitive CP-ABE consists of following four algorithms.

$Setup(1^k)$. This algorithm takes a security parameter k as input. It outputs a public parameter PK and a master key MK.

$KeyGen(PK, MK, S)$. This algorithm takes public parameter PK, the master key MK and an attribute set S as input. It outputs a private key SK related to the attribute S.

$Encrypt(PK, M, \mathbb{A})$. This algorithm takes the public parameter PK, a message M and an access policy $\mathbb{A}$ as input. It outputs ciphertext CT such that only the user whose attribute set satisfies the access policy can decrypt.

$Decrypt(PK, CT, SK)$. This algorithm takes the public parameter PK, a ciphertext CT and a private key SK as input. If and only if the attribute set S of the user satisfies the access policy $\mathbb{A}$, the algorithm can decrypt the message M successfully.

3. Problem Statement

3.1. The System Model

As described in Figure 1, the whole model contains four entities: Central Authority (CA), Data Owners (DOs), Users and Shipboard Server (SS).

- Central Authority: A CA is considered to be a entity that possesses unlimited computing and storage capacity. Meanwhile, a CA is also a trusted party, and its tasks are to generate system parameters, manage users (i.e., enrolling users: distribuing the secret key to every user) and distribute security equipment (sensor cards). Furthermore, the update of the security equipment is also responsible for CA. In the process of updating, CA redistributes a security equipment to the user and informs SS to update the ciphertext. Figure 2 shows the process of update.
- Data Owners: DOs are owners of message stored in SS. All the message is encrypted by using ABE. Finally, DOs upload the generated ciphertext to SS.
- Users: In smart ocean management, sailors, helmsman, managers and other crew are users. They can download the encrypted public data in SS. If the users want to get the message, they firstly do decrypt by using their security equipment and obtain the resulting primary ciphertext, then users with specific attributes can decrypt primary ciphertext by using their secret keys.
- Shipboard Server (SS): It is not a credible entity in smart ocean management. Concretely, SS is honest-but-curious, which can honestly implement the assigned tasks and return corresponding results. However, it will also do its best to collect sensitive information. Generally, SS is regarded as a party with unlimited computing power and storage space. In this paper, the DOs upload the encrypted message (primary ciphertext) to SS, then SS uses the public information obtained from CA to encrypt primary ciphertext, resulting in secondary ciphertex. In addition, SS is responsible for updating ciphertext when users' security equipment is redistributed.

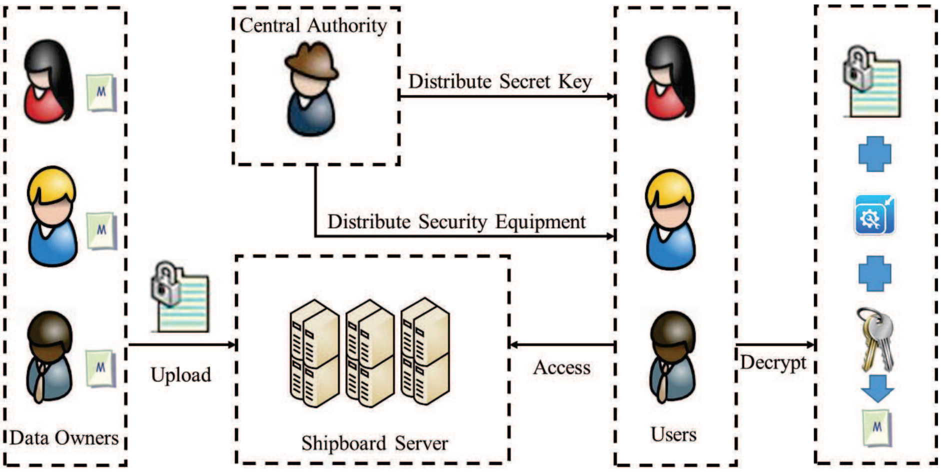


Figure 1. The system model.

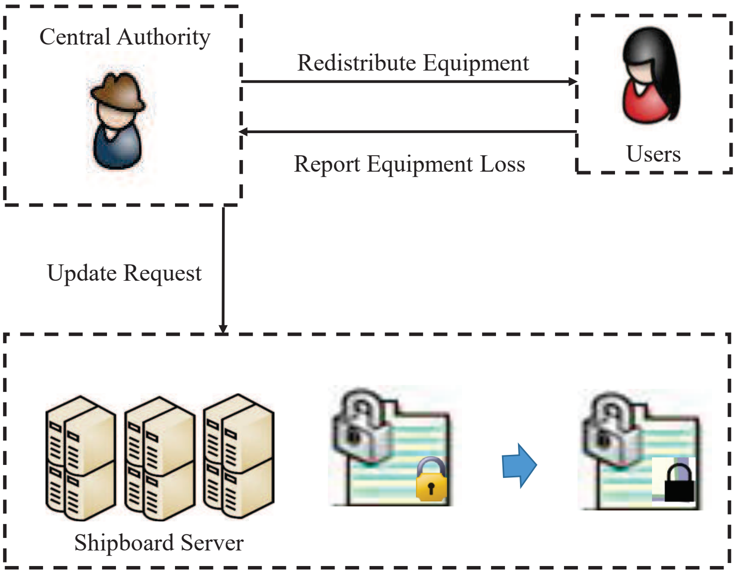


Figure 2. The process of ciphertext update and security equipment redistribution.

3.2. Notations

As is shown in Table 1, some primary notations used in our scheme are listed.

Table 1. Notations.

Notation	Interpretation
$\mathbb{G}_1, \mathbb{G}_2$	cyclic multiplicative groups
p	prime order of $\mathbb{G}_1, \mathbb{G}_2$
g, g_2, h	generators of $\mathbb{G}_1$
Z_q^*	set of nonnegative integers less than q
H_1, H_2, H_3, H_4	one-way hash function
m	a message
$\oplus$	exclusive OR
ID_i	the identity of user i
A	the attribute set (including access policy)
V	attributes value
P	the attribute that users possess
C_1	the primary ciphertext
C_2	the secondary ciphertext
C_2^*	updated ciphertext
epk_i	the public information of security equipment
esk_i	the secret information of security equipment
sk_P	the secret key of users

3.3. Security Model

In this paper, from the perspective of two factors, we mainly consider the following two threat models:

- Type-1: Decrypt without security equipment. In this case, the adversary has the right secret key, however, it has no security equipment, or security equipment and secret keys do not match.
- Type-2: Decrypt without secret key. This attack model is opposite to the previous model. In this situation, the adversary has a security equipment but no secret key, then it tries to decrypt the ciphertext.

3.4. System Components

The two-factor-based public data protection scheme in smart ocean management consists of six algorithms. The six algorithms are described separately as follows.

Setup: $(1^k) \longrightarrow (param, msk)$. The algorithm is run by CA. A security parameter k is taken as input. The algorithm outputs public parameters $param$ and master key msk.

Keygen and Security Equipment Distribution: $(param, msk, P) \longrightarrow (sk_P, epk_i, esk_i)$. The algorithm is run by CA. On inputting the public parameters $param$, the master secret key msk and the attribute P that users possess, the algorithm outputs secret key sk_P, public information epk_i, and secret information esk_i of security equipment.

Primary Encryption: $(param, A, m) \longrightarrow (C_1)$. The algorithm is run by DOs. The input includes the public parameters $param$, the message m and attribute set A. The output is the primary ciphertext C_1.

Secondary Encryption: $(param, epk_i, C_1) \longrightarrow (C_2)$. The algorithm is run by SS. The public parameters, public information epk_i of security equipment and primary ciphertext C_1 are taken as input. The algorithm outputs secondary ciphertext C_2.

Security Equipment Redistribution and Ciphertext Update: $(param, epk_i) \longrightarrow (C_2^*)$. The algorithm is run by CA and SS. On inputting the public parameters $param$ and epk_i, the algorithm outputs the ciphertext C_2^*.

Message Decryption: $(esk_i, sk_P, C_2$ or $C_2^*) \longrightarrow (m)$. The algorithm is run by users. The input includes secret information esk_i of security equipment, secret key sk_P and secondary ciphertext C_2 or C_2^*. The output is message m.

4. Our Scheme

4.1. Setup

All public parameters and master key will be generated in the setup phase. These public parameters will be shared among all parties (including DOs, Users, SS, and CA). However, the master key can only be kept by CA. The specific process of setup is as follows.

- We define $\mathbb{G}_1$ and $\mathbb{G}_2$ as cyclic multiplicative groups of prime order p, and $e : \mathbb{G}_1^2 \rightarrow \mathbb{G}_2$ is the bilinear map.
- Choose $g, g_2, h \in \mathbb{G}_1$, $\alpha, \beta \in_R Z_q^*$. Here k is a security parameter. Four collision resistant hash functions are chosen as follows : $H_1 : \mathbb{G}_1 \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow Z_q^*$, $H_3 : \mathbb{G}_2 \rightarrow \{0,1\}^*$ and $H_4 : \{0,1\}^* \rightarrow \mathbb{G}_1$. Meanwhile, setting $g_1 = g^\alpha$.
- There are n attributes in our scheme. The attribute set can be denoted as $A = \{A_1, A_2, \ldots, A_i, \ldots, A_n\}_{1 \leq i \leq n}$. Each attribute A_i has multiple attribute values $V = \{v_1, v_2, \ldots, v_i, \ldots, v_m\}_{1 \leq i \leq m}$. Each attribute value can be used as a user's ID, but the pre N bits of the attribute value is used as a common attribute of the user, here N is a threshold. Give a simple example, 201801234 is a crew number of a crew, that is, the ID of the crew. 2018 is the crew's year of admission, 01 is a department number, 234 is the sort number of the crew. When N is equal to 4, crew enrolled in 2018 can be identified. When N equals 6, the department can be identified. The public parameters $param$ is set to be $(k, q, g, g_1, g_2, h, e(g,g), H_1, H_2, H_3, H_4)$.

4.2. Keygen and Security Equipment Distribution

Firstly, CA will distribute security equipment for every user according to the their ID_i. Secondly, CA is responsible for generating the secret keys for users which have specific attributes. Users can use their own security equipment and secret keys to decrypt ciphertext. The specific process is as follows.

- The CA chooses $z_i \in_R Z_q^*$, and sets the public information of the security equipment as $epk_i = g^{z_i}$, and its corresponding secret information as $esk_i = z_i$. Finally, CA distributes a security equipment to a user ID_i and shares (epk_i, ID_i) with the SS.

– CA computes

$$\tau_i = H_4(s)^{-H_2(\beta||i)}, v_i = H_4(s)^{-H_2(\alpha||i)} \tag{2}$$

The secret key is $sk_P = (s, \tau_i, v_i)$, where s is a mapping of the user's attributes to strings. In addition, the set of attributes of each user is mapped to a unique string. P is a set of user-owned attributes.

4.3. Primary Encryption

DOs encrypt message based on attributes and send the encrypted messages to SS. Know public parameters $param$, message $m \in \{0,1\}^*$ and attributes set A. The process of primary encryption is as follows.

– Compute $c_1 = m \cdot \alpha_A^k$, $c_2 = g^k$, $c_3 = \beta_A^k$, $c_4 = A$, $M = H_4(m)$, and define $\alpha_A = \prod \alpha_i$, $\beta_A = \prod \beta_i$. Send the primary ciphertext $C_1 = \{c_1, c_2, c_3, c_4\}$ to SS and broadcast M to all users.

4.4. Secondary Encryption

After receiving the primary ciphertext from DOs, SS will encrypt it second times, resulting in secondary ciphertext. Knowing public parameters $param$, a primary ciphertext for the user and the information epk_i. The SS encrypts $C_1 = \{c_1, c_2, c_3, c_4\}$ to secondary ciphertext as follows

– Choose $\mu_1, \mu_2 \in_R \{0,1\}^*$, set $r = H_2(\mu_1, \mu_2)$. Compute $c_5 = c_1 \oplus (\mu_1||\mu_2)$, $c_6 = (\mu_1||\mu_2) \oplus H_3(e(g,g)^r)$, $c_7 = (epk_i)^{r \cdot H_1(epk_i)}$, $c_8 = h^r$, $c_9 = H_4(c_5, c_6, c_7, c_8)^r$. At this point, secondary ciphertext is $C_2 = (c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9)$.

4.5. Security Equipment Redistribution and Ciphertext Update Phase

Once the user's security equipment is stolen or lost, user needs to report to CA, then CA redistributes a security equipment to the user. Here, the work done by CA is similar to the previous security equipment distribution process, so it is omitted.

At the same time, CA also sends information to inform SS to update ciphertext. The information is as follows

$$\begin{aligned} rk_1 &= epk_i^{H_1(epk_i) \cdot (z_i \cdot H_1(epk_i))^{-1}} \cdot h^{\epsilon} \\ rk_2 &= epk_i^{H_1(epk_i) \cdot \epsilon} \end{aligned} \tag{3}$$

where $\epsilon \in_R Z_q^*$. After receiving rk_1, rk_2, the SS updates the ciphertext C_2 as follows.

– Check

$$\begin{aligned} e(c_7, h) &= e(epk_i^{H_1(epk_i)}, c_8) \\ e(c_8, H_4(c_5, c_6, c_7, c_8)) &= e(h, c_9) \end{aligned} \tag{4}$$

– If the above equations are not set up, the scheme stops. Otherwise it continues to execute.

Compute

$$\begin{aligned} C_{10} &= \frac{e(c_7, rk_1)}{e(c_8, rk_2)} \\ &= \frac{e((epk_i)^{r \cdot H_1(epk_i)}, epk_i^{H_1(epk_i) \cdot (z_i \cdot H_1(epk_i))^{-1}} \cdot h^{\epsilon})}{e(h^r, epk_i^{H_1(epk_i) \cdot \epsilon})} \\ &= \frac{\begin{array}{l} e((epk_i)^{r \cdot H_1(epk_i)}, h^{\epsilon}) \\ \quad \cdot e((epk_i)^{r \cdot H_1(epk_i)}, epk_i^{H_1(epk_i) \cdot (z_i \cdot H_1(epk_i))^{-1}}) \end{array}}{e(h^r, epk_i^{H_1(epk_i) \cdot \epsilon})} \\ &= e((epk_i)^{r \cdot H_1(epk_i)}, epk_i^{H_1(epk_i) \cdot (z_i \cdot H_1(epk_i))^{-1}}) \\ &= e(epk_i, epk_i)^{\frac{r \cdot H_1^2(epk_i)}{z_i \cdot H_1(epk_i))}} \\ &= e(g^{z_i}, g^{z_i})^{\frac{r \cdot H_1(epk_i)}{z_i}} \\ &= e(g, g)^{z_i \cdot r \cdot H_1(epk_i)} \end{aligned} \tag{5}$$

Finally, SS updates the ciphertext to $C_2^* = (c_2, c_3, c_4, c_5, c_6, c_{10})$.

4.6. Message Decryption

When users need to decrypt ciphertext, they can use security equipment and secret keys to decrypt. The two types of messages decryption are as follows:

– Security equipment and ciphertext are not updated.

Known $c_5 = c_1 \oplus (\mu_1 || \mu_2)$, so $c_1 = c_5 \oplus (\mu_1 || \mu_2)$. It is also known $c_6 = (\mu_1 || \mu_2) \oplus H_3(e(g, g)^r)$. As a result, the following formula can be obtained

$$c_1 = c_5 \oplus c_6 \oplus H_3(e(g, g)^r) \tag{6}$$

Because c_5 and c_6 are known, so users first use security equipment to compute $e(g, g)^r$. The process is as follows

$$\begin{aligned} e(g, g)^r &= e(g, g^r) \\ &= e(g, epk_i^{\frac{r}{z_i}}) \\ &= e(g, epk_i^{r \cdot H_1(epk_i) \cdot \frac{1}{z_i \cdot H_1(epk_i)}}) \\ &= e(g, c_7^{\frac{1}{z_i \cdot H_1(epk_i)}}) \end{aligned} \tag{7}$$

By decryption of the user's security devices, c_1 can be obtained. Next, CA checks whether the user's P can satisfy A or not. If it is true, the CA computes $\tau_A = \prod \tau_i$, $v_A = \prod v_i$. The message can be decrypted as the following equation

$$m = \frac{c_1}{e(\tau_A \cdot v_A, g^k) \cdot e(H_4(s), \beta_A^k)} \tag{8}$$

Finally, users verify the correctness of the message by checking whether $M = H_4(m)$. If $M = H_4(m)$ the computation result of the decryption, otherwise the message is wrong.

– Security equipment and ciphertext had been updated. In this case, $e(g, g)^r$ can be calculated by the following formula

$$e(g, g)^r = c_{10}^{\frac{1}{z_i \cdot H_1(epk_i)}} \tag{9}$$

The rest of the decryption process is similar to the above, so it is omitted.

4.7. Correctness Verification

If a user's attributes set P satisfies attribute sets A in specific access structure, the user is able to decrypt the message correctly. Therefore, we have that

$$\begin{aligned}
&\frac{c_1}{e(\tau_A \cdot v_A, g^k) \cdot e(H_4(s), \beta_A^k)} \\
&= \frac{m \cdot \alpha_A^k}{e(\tau_A \cdot v_A, g^k) \cdot e(H_4(s), \beta_A^k)} \\
&= \frac{m \cdot (\prod \alpha_i)^k}{e(v_A \cdot \prod \tau_i, g^k) \cdot e(H_4(s), (\prod \beta_i)^k)} \\
&= \frac{m \cdot (\prod e(H_4(s), g)^{H_2(\alpha||i)})^k}{e(\prod (H_4(s)^{H_2(\alpha||i)}) \cdot \prod (H_4(s)^{-H_2(\beta||i)}), g^k) \cdot e(H_4(s), (\prod g^{H_2(\beta||i)})^k)} \\
&= \frac{m \cdot (e(H_4(s), g)^{\sum H_2(\alpha||i)})^k}{e((H_4(s)^{\sum H_2(\alpha||i)}) \cdot (H_4(s)^{-\sum H_2(\beta||i)}), g)^k \cdot e(H_4(s), (g^{\sum H_2(\beta||i)}))^k} \\
&= \frac{m \cdot (e(H_4(s), g)^{\sum H_2(\alpha||i)})^k}{e((H_4(s)^{\sum H_2(\alpha||i)}) \cdot (H_4(s)^{-\sum H_2(\beta||i)}), g)^k \cdot e(H_4(s)^{\sum H_2(\beta||i)}, (g))^k} \\
&= \frac{m \cdot (e(H_4(s), g)^{\sum H_2(\alpha||i)})^k}{e((H_4(s)^{\sum H_2(\alpha||i)}) \cdot (H_4(s)^{-\sum H_2(\beta||i)}) \cdot (H_4(s)^{\sum H_2(\beta||i)}), (g))^k} \\
&= \frac{m \cdot (e(H_4(s), g)^{\sum H_2(\alpha||i)})^k}{e((H_4(s)^{\sum H_2(\alpha||i)}), (g))^k} \\
&= \frac{m \cdot (e(H_4(s), g)^{\sum H_2(\alpha||i)})^k}{(e(H_4(s), g)^{\sum H_2(\alpha||i)})^k} \\
&= m
\end{aligned} \tag{10}$$

5. Security Analysis

In this section, security analysis consistent with previous security models is given.

For the Type-1 security model, here an adversary $\mathcal{A}$ can get the user's secret key sk_P, but it has no corresponding security equipment. Suppose $\mathcal{A}$ has got the secondary ciphertext $C_2 = (c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9)$ or updated ciphertext $C_2^* = (c_2, c_3, c_4, c_5, c_6, c_{10})$, which are all stored in SS, where $c_2 = g^k$, $c_3 = \beta_A^k$, $c_4 = A$, $c_5 = c_1 \oplus (\mu_1||\mu_2)$, $c_6 = (\mu_1||\mu_2) \oplus H_3(e(g,g)^r)$, $c_7 = (epk_i)^{r \cdot H_1(epk_i)}$, $c_8 = h^r$, $c_9 = H_4(c_5, c_6, c_7, c_8)^r$, $c_{10} = e(g,g)^{z_i \cdot r \cdot H_1(epk_i)}$. $\mathcal{A}$ tries to compute $H_3(e(g,g)^r)$. Of course, $e(g,g)^r$ first needs to be calculated, that is

$$\begin{aligned}
e(g,g)^r &= e(g, g^r) \\
&= e(g, epk_i^{\frac{r}{z_i}}) \\
&= e(g, epk_i^{r \cdot H_1(epk_i) \cdot \frac{1}{z_i \cdot H_1(epk_i)}}) \\
&= e(g, c_7^{\frac{1}{z_i \cdot H_1(epk_i)}})
\end{aligned} \tag{11}$$

From the above formula, it is easy to see that if $\mathcal{A}$ correctly guesses the z_i, it will be able to get $H_3(e(g,g)^r)$ successfully. Due to $z_i \in_R Z_q^*$, the probability of guessing is $\frac{1}{q}$. In addition, because of $H_3 : \mathbb{G}_2 \rightarrow \{0,1\}^*$, $\mathcal{A}$ is able to correctly guess the output of H_3 with probability $\frac{1}{2^*}$. In summary, if $\mathcal{A}$ wants to access the correct message, the probability of its guess is $\frac{1}{2^* q}$. As long as q and $*$ are big enough, the probability is ignorable.

Theorem 1. *Suppose these hash function H_1, H_2, H_3, and H_4 are all random oracles. Our scheme is secure against chosen plaintext attack under the security model if the 1-wDBDHI assumption holds.*

Proof of Theorem 1. If $\mathcal{A}$ just recovers the message with a secret key instead of a security equipment, then we can design an algorithm $\mathcal{B}$ to break 1-wDBDHI the assumption.

Setup. $\mathcal{B}$ is assigned an example of 1-wDBDHI problem. $\mathcal{B}$ sets g, $y = g^a$, $Y = g^b$, $E = e(g,g)^{\frac{b}{a}}$, chooses $\psi, \kappa \in_R Z_q^*$, and sends the public parameters $param$ to be $(k, q, g, g_1 = g^\alpha, g_2 = y, h = g^\kappa, e(g,g), H_1, H_2, H_3, H_4)$ to $\mathcal{A}$. Among them, H_1, H_2, H_3, H_4 are random oracles which is controlled by $\mathcal{B}$.

Phase 1. $\mathcal{B}$ receives the following queries from $\mathcal{A}$.

1. Security equipment queries. $\mathcal{B}$ randomly chooses $coin$, where the value of $coin$ is 0 or 1. $Pr[coin = 1] = \iota$.
   - When $coin = 0$. $\mathcal{B}$ outputs $b \in \{0,1\}$ randomly, and sends $(ID, epk_i = g^{z_i}, coin = 0, esk_i = z_i)$ to $Equipmentlist$.
   - When $coin = 1$. if $(ID, epk_i, coin = 1, z_i)$ has already existed in $Equipmentlist$. $\mathcal{B}$ sends epk_i and esk_i to $\mathcal{A}$ at the same time. Otherwise, $\mathcal{B}$ chooses $z_i \in_R Z_q^*$ and set $esk_i = z_i$, $epk_i = g^{z_i}$. Finally, $\mathcal{B}$ sends $(ID, epk_i = g^{z_i}, coin = 1, esk_i = z_i)$ to $Equipmentlist$, and returns (epk_i, esk_i) to $\mathcal{A}$.
2. Secret key queries. $\mathcal{A}$ sends a user's ID to $\mathcal{B}$, which queries the secret key of this user. $\mathcal{B}$ checks whether it has already owned ID and the corresponding sk_P. In this security model, $\mathcal{B}$ sends sk_P to $\mathcal{A}$.
3. Message decryption queries. $\mathcal{A}$ sends a ciphertext to $\mathcal{B}$. $\mathcal{B}$ decrypts the message as follows.

   For ciphertext C_2. $\mathcal{B}$ first checks whether there are tuples (m, μ_1, μ_2, r), then sets $c_7 = (epk_i)^{r \cdot H_1(epk_i)}$, $c_8 = h^r$, and $c_9 = H_4(c_5, c_6, c_7, c_8)^r$. Finally, $\mathcal{B}$ recovers $e(g,g)^r$ and computes $c_1 = c_5 \oplus c_6 \oplus H_3(e(g,g)^r)$. For ciphertext C_2^*. The process of the query is similar to the above, so it is omitted.

   Challenge. $\mathcal{A}$ outputs m_0, m_1 and (ID^*, epk_i^*). $\mathcal{B}$ gets epk_i from the list $EquipmentList$. If $coin^* = 1$, $\mathcal{B}$ aborts and outputs $a, b \in \{0,1\}$. Else, $\mathcal{B}$ proceeds.

- For original ciphertext. $\mathcal{B}$ chooses $\mu_1^*, \mu_2^* \in_R \{0,1\}^*$ and $b \in_R \{0,1\}^*$. It sets $c_1^* = m_b \cdot \alpha_A^k$, $c_2^* = g^k$, $c_3^* = \beta_A^k$, $c_4^* = A$, $c_5^* = c_1^* \oplus (\mu_1^* || \mu_2^*)$, $c_6^* = (\mu_1^* || \mu_2^*) \oplus H_3(E)$, $c_7^* = (Y)^{z_i^* \cdot H_1(epk_i^*)}$, $c_8 = Y^\kappa$, $c_9 = Y^{H_4(c_5^*, c_6^*, c_7^*, c_8^*)}$. $\mathcal{B}$ outputs $C_2^{**} = (c_2^*, c_3^*, c_4^*, c_5^*, c_6^*, c_7^*, c_8^*, c_9^*)$.
- For updated ciphertext. $\mathcal{B}$ sets $C_2^{***} = (c_2^*, c_3^*, c_4^*, c_5^*, c_6^*, c_{10}^*)$, here $c_2^* = g^k$, $c_3^* = \beta_A^k$, $c_4^* = A$, $c_5^* = c_1^* \oplus (\mu_1^* || \mu_2^*)$, $c_6^* = (\mu_1^* || \mu_2^*) \oplus H_3(E)$ and $c_{10}^* = e(g, Y^{z_i^* \cdot H_1(epk_i^*)})$.

Phase 2. $\mathcal{A}$ guesses a bit $b' \in \{0,1\}$. These simulated H_1, H_4 are perfect. If $\mathcal{A}$ does not either send μ_1^*, μ_2^* to H_2 or send E to H_3 before the challenge phase, the simulations of H_2 and H_3 also are perfect. $AskH_2^*$ and $AskH_3^*$ are denoted as events that (μ_1^*, μ_2^*) has been issued to H_2 and T has been issued to H_3, respectively.

We assume that as long as $\mathcal{B}$ does not abort, the responses to the security equipment queries, secret key queries, and challenge phase are perfect. We denote $Abort$ as the event that $\mathcal{B}$ aborts in the responses to the security equipment queries or in the challenge phase. Therefore, $Pr[\urcorner Abort] \geq \iota^{qse}(1-\iota)$. The maximum of ι is $\iota_{opt} = \frac{qse}{1+qse}$. Where qse is the total number of security equipment queries. Thus, the minimum of probability $Pr[\urcorner Abort]$ is $\frac{1}{\mathcal{E} \cdot (1+qse)}$, here $\mathcal{E}$ is the base of the natural logarithm.

As long as $\mathcal{A}$ releases a valid original ciphertext with the help of H_2, the simulation of ciphertext update queries is also considered to be perfect. The probability of the error in these events is $Pr[CUE] \leq \frac{q_{cu}}{q}$, here q_{cu} is the number of ciphertext update queries.

As long as $\mathcal{B}$ does not refuse the queries of some valid ciphertexts, the simulation of message decryption queries is perfect. Val, $AskH_2$, and $AskH_3$ are these events that a valid ciphertext is returned, (μ_1, μ_2) is issued to H_2 and $e(g,g)^r$ is issued to H_3, respectively. According to the above simulation, we can get $Pr[Val|\neg AskH_3] \leq \frac{qH_3}{2^l} + \frac{1}{q}$ and $Pr[Val|\neg AskH_2] \leq \frac{qH_2}{2^l} + \frac{1}{q}$, here qH_3 and qH_2 are the numbers of querying H_2 and H_3, respectively. $Pr[DRErr]$ is the probability that the event $Val|(\neg AskH_2 \vee \neg AskH_3)$ occurs. Therefore, we can get $Pr[DRErr] \leq (\frac{qH_2 + qH_3}{2^l} + \frac{2}{q}) \cdot qmd$, here qmd is the number of message decryption queries.

Based on the analysis of the above three simulations, the following probability relationships can be calculated.

$$\begin{aligned} \epsilon &= |Pr[b = b'] - \frac{1}{2}| \\ &\leq \frac{1}{2} Pr[(H_2^*|\neg H_3^*) \vee H_3^* \vee CUE \vee DRErr|\neg Abort] \\ &\leq \frac{1}{2Pr[\neg Abort]}(AskH_3^* + \frac{qH_2 + (qH_2 + qH_3)qmd}{2^l} + \frac{2qmd + q_{cu}}{q}) \end{aligned} \tag{12}$$

Therefore, we have that

$$\begin{aligned} \epsilon' &\geq \frac{1}{qH_3}(AskH_3^*) \\ &\geq \frac{1}{qH_3}(\frac{2\epsilon}{\mathcal{E}(1+qse)} - \frac{qH_2 + (qH_2 + qH_3)qmd}{2^l} - \frac{2qmd + q_{cu}}{q}) \end{aligned} \tag{13}$$

Finally, we have that

$$\begin{aligned} &|Pr[\mathcal{A}(g, g^a, \ldots, g^{a^q}, g^b, e(g,g)^{b/a}) = 1] \\ &- Pr[\mathcal{A}(g, g^a, \ldots, g^{a^q}, g^b, e(g,g)^z) = 1]| \geq \epsilon' \end{aligned} \tag{14}$$

as required, which completes the proof. □

For the Type-2 security model, according to the previous construction, although an $\mathcal{A}$ has already owned security equipment, it has no secret key. Therefore, $\mathcal{A}$ does not have enough attributes to satisfy the A, and it can't decrypt the ciphertext to get message. Because SS is a semi-trusted entity, message m is encrypted and uploaded by the DO. Meanwhile, attribute sets and access policy formulated by DO are also incorporated into ciphertext.

In the process of ABE, a message m is obfuscated with the α_A^k, here, α_A is determined by the attribute sets and access policy A, the k is security parameter which is randomly generated in every oracle. From the above analysis, the α_A^k is secure, and the message m is also secure.

In the process of decryption, $\mathcal{A}$ does not have sk_P or has the wrong secret key, it can't get the attribute set P which is granted by CA. CA is a trusted party, it won't have a collusion attack with SS. From the above analysis, as long as α_A^k is not easy to crack, the message m is secure.

6. Performance Analysis

In this section, the performance of proposed scheme is analyzed from different perspectives. Meanwhile, the comparison with [14,38] is also analyzed in terms of features, communication and computational cost. The results of the comparison reveal that our scheme is more suitable for the protection of public data resources and achieves more functions, but dosen't require a great increase of cost. In general, our scheme is more suitable to be practically deployed

First of all, some notations used in efficiency analysis are defined as follows. $|\mathbb{G}_1|$ and $|\mathbb{G}_2|$ are utilized to denote the length of an element in groups $\mathbb{G}_1$ and $\mathbb{G}_2$, l denotes the length of security parameter, ck denotes the key length of a symmetric encryption algorithm, $|Z_q^*|$ denotes the length of an element in Z_q^*. $|m|$ and $|*|$ denote the length of message m and arbitrary 01 strings, respectively. PA, EXP_1, EXP_2, and H are utilized to denote the cost of a bilinear pairing , an exponentiation in $\mathbb{G}_1$, an exponentiation in $\mathbb{G}_2$ and a one-way hash function, respectively. EM and DM are utilized to denote the cost of symmetric encryption and decryption, respectively. In our scheme, the calculation of the bilinear pairing and the exponentiation is over the supersingular elliptic curve, which is defined in preliminaries. Thus, the computational complexities of the bilinear pairing is $\mathcal{O}(m^2)$. Here, m is the extension degree of the finite field Z_q^*. n denotes the number of users.

For features comparison. We compare our scheme with [14,38] in terms of access control policy, two-factor protection, equipment revocation and key and equipment separation. The results are shown in Table 2. Compared with [14,38] respectively, it can be seen that only our scheme can achieve all the three functions at the same time. Compared with our scheme, ref. [14] cannot simultaneously share data with a group of users, limited scope of application may be a major drawback. Especially, it is worth saying that since [38] does not support the two-factor mechanism, which greatly reduces the security of this scheme. In short, refs. [14,38] only achieve partial design goal of our scheme.

Table 2. Features comparison of related schemes.

Schemes	Access Control Policy	Two Factors	Equipment Revocation	Key and Equipment Separation
Ours	ABE	Yes	Yes	Yes
[38]	ABE	No	No	Yes
[14]	IBE	Yes	Yes	No

For theoretical comparison. Communication, computational and time complexity comparison are demonstrated in Tables 3–5, respectively. In Table 4, KSED, PE, SE, SERCU, MDSC, and MDUC are initial capitalization of keygen and security equipment distribution, primary encryption, secondary encryption, security equipment redistribution and ciphertext update, message decryption (from secondary ciphertext) and message decryption (from updated ciphertext), respectively. In the rest of this paper, this abbreviation will also be used. Compared with [38] which lacks two-factor data protection, it can be seen that our scheme requires increase a little computational cost in security equipment redistribution and ciphertext update phase. This is because the redistribution of security equipment is supported by our scheme. In the process of ciphertext generation, our scheme does not require a symmetric encryption operation. Furthermore, it is worth of mentioning that cost of secondary encryption can be outsourced to SS. A similar situation also exists in Table 3, which is that our scheme needs extra communication cost in transmission of security equipment and updating ciphertext. However, the total cost of our scheme is less than [38]. Compared with [14] which lacks public data protection, the cost of our scheme is less than [14] at all phases, this is mainly due to the fact that an efficient ABE is adopted by our scheme. In addition, under the premise of ensuring safety, the generation and redistribution of security equipment of our scheme is also more streamlined than [14]. It can be seen from Table 5 that the time complexity of our scheme is not higher than [14,38]. Moreover, the computational complexity of our scheme is linearly related with the number of users, which indicates that our scheme is suitable for a real time security guard scenario.

Table 3. Communication cost comparison.

Schemes	Ours	[38]	[14]
secret key length	$2\vert\mathbb{G}_1\vert$	$3\vert Z_q^*\vert$	$2\vert\mathbb{G}_1\vert$
security equipment length	$\vert\mathbb{G}_1\vert + \vert Z_q^*\vert$	$\perp$	$2\vert\mathbb{G}_1\vert + 2\vert Z_q^*\vert$
primary ciphertext length	$(\vert m\vert + 2)\vert\mathbb{G}_1\vert$	$\vert ck\vert$	$6\vert\mathbb{G}_1\vert + 4l$
secondary ciphertext length	$(\vert m\vert + 5)\vert\mathbb{G}_1\vert + \vert * \vert$	$\vert\mathbb{G}_1\vert + \vert\mathbb{G}_2\vert$	$3\vert\mathbb{G}_1\vert+\vert\mathbb{G}_2\vert + 4l$
updated ciphertext length	$(\vert m\vert + 2)\vert\mathbb{G}_1\vert + \vert * \vert + \vert\mathbb{G}_2\vert$	$\perp$	$2\vert\mathbb{G}_1\vert$

Table 4. Computational cost comparison.

Phases	Ours	[38]	[14]
KSED	$EXP_1 + 4H$	$12EXP_1 + 2EXP_2 + H$	$4EXP_1$
PE	$3EXP_1$	EM	$2EXP_1 + EXP_2 + PA + 3H$
SE	$3EXP_1 + EXP_2 + 4H$	$4EXP_1 + EXP_2 + 2H$	$3EXP_1 + EXP_2 + PA + 3H$
SERCU	$4EXP_1 + 6PA + 5H$	$\perp$	$6EXP_1 + 6PA + 5H$
MDSC	$7EXP_1 + 2PA + 3H$	$PA + 2EXP_2$	$9EXP_1 + 2PA + 3H$
MDUC	$6EXP_1 + EXP_2 + 2PA + 3H$	DM	$8EXP_1 + EXP_2 + 2PA + 2H$

Table 5. Time complexity comparison.

Time Complexity	Ours	[38]	[14]
communication complexity	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2\sqrt{n})$	$\mathcal{O}(n^2)$
computational complexity	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm^2)$

For practical comparison. In practical efficiency test, the test environment is set to be: Intel(R) Core(TM) i5-6500 CPU @ 3.2 GHz, 8 GB RAM, GNU Multiple Precision Arithmetic (GMP) library, Pairing-Based Cryptography (PBC) library, and C language are used on a Linux system with Ubuntu 16.04 TLS. Microsoft Office Excel 2016 and Matlab 2016a are used by us as tools for drawing statistical figures. As we all know, PBC library is a free and portable C language library. Through an abstract interface, programmers can implement pair-based cryptosystem without considering the specific mathematical details or even the knowledge of elliptic curve and theory. Ubuntu 16.04 TLS is a free open source desktop operating system based on Linux, which combines Windows visualization and Linux stability. The comparison results are shown in Figure 3, as the number of users increases, the computational cost of the three schemes almost increases linearly, but the growth trend of our scheme is slow, and it is consistent with previous theoretical analysis.

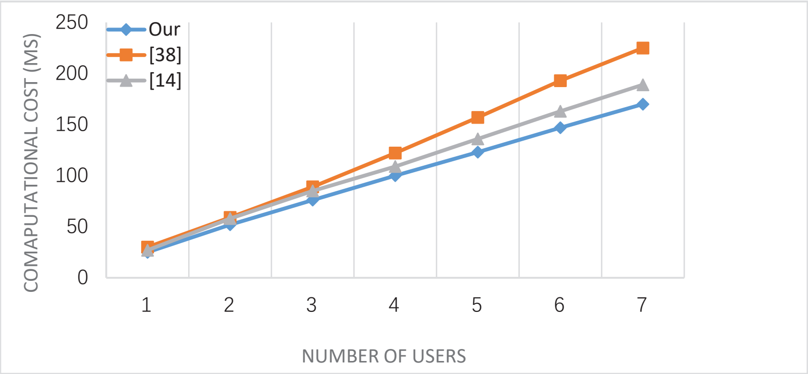
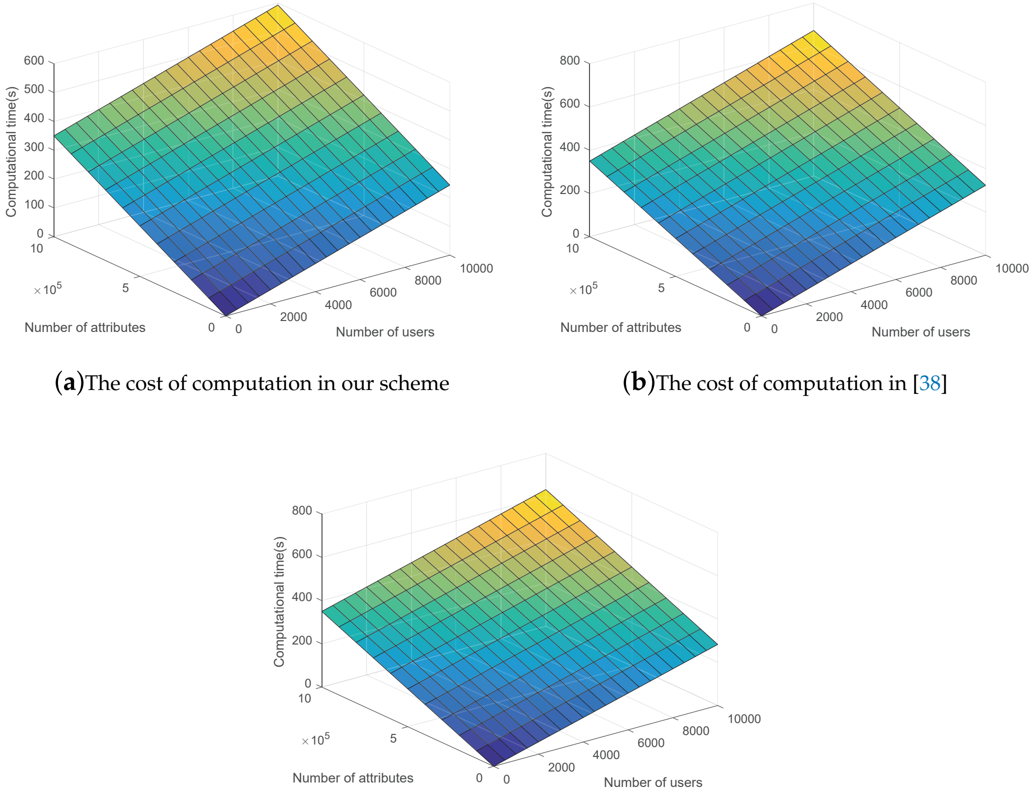


Figure 3. The computational cost comparison.

In order to make a more comprehensive comparison, the computational cost of attributes is also introduced. As shown in Figure 4, (a) is the three-dimensional experimental result of our scheme, (b) is the experimental result of [38], and (c) is the experimental result of [14]. We set the number of users varies from 0 to 10,000 and the number of attributes varies from 0 to 1,000,000. The computational cost increases when either the number of users or the number of attributes. It is easy to see that although upward trends of the three figures are similar, the trend of our scheme is more gentle. For example, when the number of users and the number of attributes are 1000 and 1,000,000 respectively, time less than 600 s is consumed in our scheme, whereas [14,38] take more than 600 s, which also indicates the high efficiency of our scheme.



(**a**)The cost of computation in our scheme

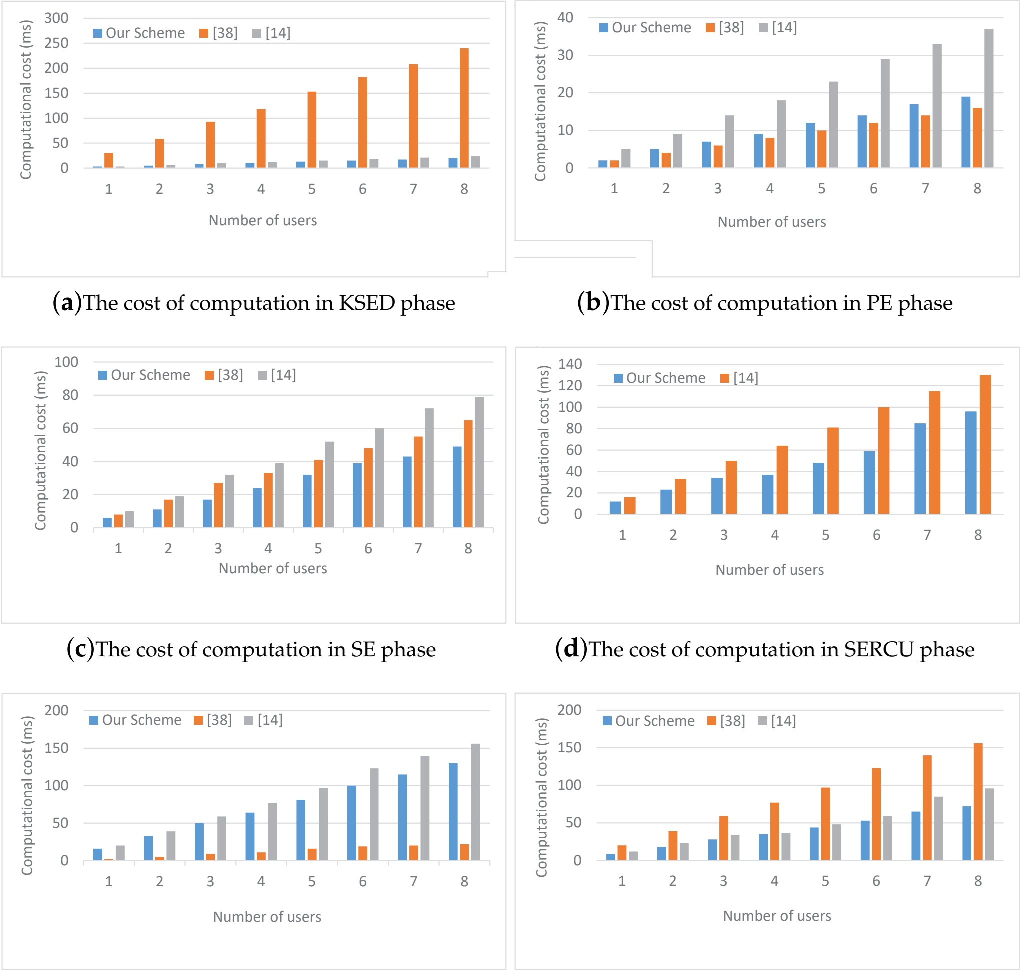
(**b**)The cost of computation in [38]

(**c**)The cost of computation in [14]

Figure 4. The computational cost comparision.

Figure 5 compares computational cost of our scheme with [14,38] from different phases. Here, the number of attributes per user is set to 2. (a) shows the computational time for generating key and distributing security equipment. Apparently, the computational cost of [14] is a little more than our scheme, however, the computational cost of [38] is much greater than our scheme and grows faster. The main reason for this situation is that there are more bilinear pairing operations and hash functions in [14,38] at this phase. (c), (d) and (f) demonstrate the computational time of secondary encryption, security equipment redistribution, and ciphertext update and message decryption (from updated ciphertext) respectively. The analysis of them is similar to (a) and is not covered here. (b), and (e) show the computational cost of primary encryption and message decryption (from secondary ciphertext) respectively. In the above two phases, the computational cost of [38] is less than our scheme. The main reason is that symmetric encryption and decryption algorithms are applied in [38] at these two phases. It is well known that symmetric encryption algorithms are generally more efficient than public key encryption algorithms. However, the total computational cost of each phase of [38] is still higher than our scheme.

In order to evaluate the performance more intuitively, the simulation experiment of our scheme, refs. [14,38] are deployed on a mobile device. The experiment is implemented on Nexus 5X Android virtual machine with Four-core CPU, 2 GB running memory, and 32 G body memory. Moreover, the codes are written in Android Studio with Java programming language to obtain the experimental results. Figure 6 is one of experimental results of three different schemes obtained from the mobile device. It is easy to see from Figure 6 that 1607.48 ms, 2221.67 ms, and 1977.32 ms are cost of our schme, [14,38] respectively when the number of attributes is 50. The simulation result for each phase is also consistent with Figure 5. Obviously, our scheme is more suitable for deployment in a real-world application.



(**a**)The cost of computation in KSED phase

(**b**)The cost of computation in PE phase

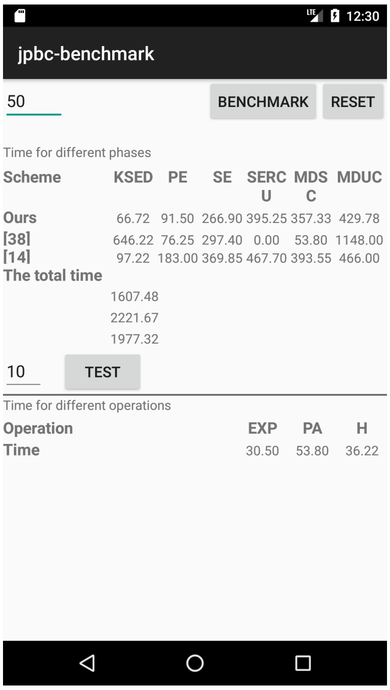
(**c**)The cost of computation in SE phase

(**d**)The cost of computation in SERCU phase

(**e**)The cost of computation in MDSC phase

(**f**)The cost of computation in MDUC phase

Figure 5. The computational cost comparision in different phases.



50

Time for different phases

Scheme	KSED	PE	SE	SERCU	MDSC	MDUC
Ours	66.72	91.50	266.90	395.25	357.33	429.78
[38]	646.22	76.25	297.40	0.00	53.80	1148.00
[14]	97.22	183.00	369.85	467.70	393.55	466.00
The total time						
	1607.48					
	2221.67					
	1977.32					

10

Time for different operations

Operation	EXP	PA	H
Time	30.50	53.80	36.22

Figure 6. The experimental results of different phases on mobile device.

In short, through the above various performance analysis, it is not difficult to find that our scheme can achieve two-factor protection for public data and redistribution of security equipment with less overhead.

7. Conclusions

Various data resources are important objects of ocean management, which may involve navigation, mining, shipping, and even national security, secure and efficient data protection schemes are especially needed. At present, the research on two-factor data security protection scheme and smart ocean management is flourishing. Given the shortcomings of existing schemes, we propose a two-factor-based public data protection scheme in smart ocean management. In our scheme, DOs are allowed to encrypt the message with some attributes (including access policy), users who satisfy certain attributes can combine their own secret keys and security equipment to decrypt ciphertext. In addition, the revocation of security equipment is also an important advantage of our scheme, which solves the problem of equipment loss well and brings many conveniences to smart ocean management. The analysis of security and performance shows that our scheme is more efficient than similar schemes on the premise of ensuring security.

Although sufficient contributions have been included in our works, there are still several challenges that we will leave as a future work. Firstly, our scheme supports one-time equipment revocation that may be not sufficient enough in practice. Therefore, multiple revocation for equipment will be one of the next works to be completed. Secondly, the access policy in our scheme is only an abstract framework and does not involve specific access control structures, such as tree or matrix. Therefore, how to design a specific access control structure to better adapt to smart ocean management is one of the further works.

Author Contributions: Methodology, J.S.; Investigation, Software and Writing—original draft, X.J.; Writing—review & editing, Y.C.; Data curation and Resources, D.L.; Data curation and Formal analysis, T.Z.

Funding: This work is supported by the National Science Foundation of China under Grant No. 61672295, No. U1836115 and No. 61672290, Natural Science Foundation of Jiangsu Province of China under Grant No. BK20181408, Guangxi Key Laboratory of Cryptography and Information Security under Grant No. GCIS201715, the MIST (MInistry of Science & ICT), Korea, under the National Program for Excellence in SW supervised by the IITP (Institute for Information & communications Technology Promotion) (2017-0-00137), the State Key Laboratory of Information Security under Grant No. 2017-MS-10, the CICAEET fund, and the PAPD fund.

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

References

1. Kim, H.; Mokdad, L.; Ben-Othman, J. Designing UAV Surveillance Frameworks for Smart City and Extensive Ocean with Differential Perspectives. *IEEE Commun. Mag.* **2018**, *56*, 98–104. [CrossRef]
2. Rio, J.D.; Toma, D.M.; Martinez, E.; Oreilly, T.C.; Delory, E.; Pearlman, J.S.; Waldmann, C.; Jirka, S. A Sensor Web Architecture for Integrating Smart Oceanographic Sensors into the Semantic Sensor Web. *IEEE J. Ocean. Eng.* **2017**, *43*, 830–842.
3. Hayes, B. Cloud computing. *Commun. ACM* **2008**, *51*, 9–11. [CrossRef]
4. Zhang, Y.Q.; Wang, X.F.; Liu, X.F.; Liu, L. Survey on Cloud Computing Security. *J. Softw.* **2016**, *8271*, 302–311.
5. 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**. [CrossRef]
6. Yu, J.; Ren, K.; Wang, C.; Varadharajan, V. Enabling Cloud Storage Auditing with Key-Exposure Resistance. *IEEE Trans. Inf. Forensics Secur.* **2017**, *10*, 1167–1179.
7. Chen, H.C.H.; Hu, Y.; Lee, P.P.C.; Tang, Y. NCCloud: A Network-Coding-Based Storage System in a Cloud-of-Clouds. *IEEE Trans. Comput.* **2014**, *63*, 31–44. [CrossRef]
8. Wang, C.; Chow, S.S.M.; Wang, Q.; Ren, K.; Lou, W. Privacy-Preserving Public Auditing for Secure Cloud Storage. *IEEE Trans. Comput.* **2013**, *62*, 362–375. [CrossRef]
9. Shen, J.; Wang, C.; Wang, A.; Ji, S.; Zhang, Y. A Searchable and Verifiable Data Protection Scheme for Scholarly Big Data. *IEEE Trans. Emerg. Top. Comput.* **2018**. [CrossRef]

10. Chen, R.; Mu, Y.; Yang, G.; Guo, F.; Wang, X. Dual-Server Public-Key Encryption With Keyword Search for Secure Cloud Storage. *IEEE Trans. Inf. Forensics Secur.* **2017**, *11*, 789–798. [CrossRef]
11. Xu, P.; Wu, Q.; Wang, W.; Susilo, W.; Domingo-Ferrer, J.; Jin, H. Generating Searchable Public-Key Ciphertexts with Hidden Structures for Fast Keyword Search. *IEEE Trans. Inf. Forensics Secur.* **2017**, *10*, 1993–2006.
12. Ma, M.; He, D.; Kumar, N.; Choo, K.K.R.; Chen, J. Certificateless searchable public key encryption scheme for industrial internet of things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 759–767. [CrossRef]
13. Xu, G.; Qiu, S.; Ahmad, H.; Xu, G.; Guo, Y.; Zhang, M.; Xu, H. A Multi-Server Two-Factor Authentication Scheme with Un-Traceability Using Elliptic Curve Cryptography. *Sensors* **2018**. [CrossRef] [PubMed]
14. Liu, J.K.; Liang, K.; Susilo, W.; Liu, J.; Xiang, Y. Two-Factor Data Security Protection Mechanism for Cloud Storage System. *IEEE Trans. Comput.* **2016**, *65*, 1992–2004. [CrossRef]
15. Kim, H.; Ben-Othman, J. A Collision-free Surveillance System using Smart UAVs in Multi Domain IoT. *IEEE Commun. Lett.* **2018**, *22*, 2587–2590. [CrossRef]
16. Han, G.; Shen, S.; Song, H.; Yang, T.; Zhang, W. A stratification-based data collection scheme in underwater acoustic sensor networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10671–10682. [CrossRef]
17. Han, G.; Jiang, J.; Shu, L.; Guizani, M. An Attack-Resistant Trust Model based on Multidimensional trust Metrics in Underwater Acoustic Sensor Network. *IEEE Trans. Mob. Comput.* **2015**, *14*, 2447–2459. [CrossRef]
18. Fu, Z.; Ren, K.; Shu, J.; Sun, X.; Huang, F. Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 2546–2559. [CrossRef]
19. Lim, H.; Tuladhar, K.; Kim, H. Detecting Location Spoofing using ADAS Sensors in VANETS. In Proceedings of the IEEE CCNC, Las Vegas, NV, USA, 11–14 January 2019.
20. Shen, J.; Wang, A.; Wang, C.; Li, J.; Zhang, Y. Content-centric Group User Authentication for Secure Social Networks. *IEEE Trans. Emerg. Top. Comput.* **2017**. [CrossRef]
21. Shen, J.; Zhou, T.; He, D.; Zhang, Y.; Sun, X.; Xiang, Y. Block design-based key agreement for group data sharing in cloud computing. *IEEE Trans. Dependable Secure Comput.* **2017**. [CrossRef]
22. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In Proceedings of the International Conference on Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; pp. 457–473.
23. Pirretti, M.; Traynor, P.; Mcdaniel, P.; Waters, B. Secure attribute-based systems. *J. Comput. Secur.* **2010**, *18* , 799–837. [CrossRef]
24. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 89–98.
25. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-Policy Attribute-Based Encryption. In Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
26. Ling, C.; Newport, C. Provably secure ciphertext policy ABE. In Proceedings of the ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 29 October–2 November 2007; pp. 456–465.
27. Boldyreva, A.; Goyal, V.; Kumar, V. Identity-based encryption with efficient revocation. In Proceedings of the ACM Conference on Computer and Communications Security, Leuven, Belgium, 23–25 July 2008; pp. 417–426.
28. Yu, S.; Wang, C.; Ren, K.; Lou, W. Attribute based data sharing with attribute revocation. In Proceedings of the 5th International Symposium on ACM Symposium on Information, Computer and Communications Security, Beijing, China, 13–16 April 2010; pp. 261–270.
29. Liu, X.; Ma, J.; Xiong, J.; Liu, G. Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data. *Int. J. Netw. Secur.* **2014**, *16*, 437–443.
30. Waters, B. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. *Lect. Notes Comput. Sci.* **2011**, *2008*, 321–334.
31. Xie, X.; Ma, H.; Li, J.; Chen, X. An Efficient Ciphertext-Policy Attribute-Based Access Control towards Revocation in Cloud Computing. *J. Univ. Comput.* **2013**, *19*, 2349–2367.
32. Shen, J.; Liu, D.; Bhuiyan, M.Z.A.; Shen, J.; Sun, X.; Castiglione, A. Secure Verifiable Database Supporting Efficient Dynamic Operations in Cloud Computing. *IEEE Trans. Emerg. Top. Comput.* **2017**. [CrossRef]
33. Shivanna, K.; Deva, S.P.; Santoshkumar, M. Privacy Preservation in Cloud Computing with Double Encryption Method. In *Computer Communication, Networking and Internet Security*; Springer: Singapore, 2017; pp. 125–133.

34. Dodis, Y.; Katz, J.; Xu, S.; Yung, M. Key-Insulated Public Key Cryptosystems. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology, Amsterdam, The Netherlands, 28 April–2 May 2002; pp. 65–82.
35. Dodis, Y.; Katz, J.; Xu, S. Strong Key-Insulated Signature Schemes. In Proceedings of the International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography, Miami, FL, USA, 6–8 January 2003; pp. 130–144.
36. Hanaoka, G.; Hanaoka, Y.; Imai, H. Parallel key-insulated public key encryption. In *International Workshop on Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 105–122.
37. Quisquater, J.J.; Yung, M. Parallel key-insulated public key encryption without random oracles. In Proceedings of the International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, 16–20 April 2007; pp. 298–314.
38. Wang, S.; Liang, K.; Liu, J.K.; Chen, J.; Yu, J.; Xie, W. Attribute-based data sharing scheme revisited in cloud computing. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1661–1673. [CrossRef]