1. Introduction

Massive data grow explosively in our daily life. Data publish-subscribe service is an effective method for data publishers to selectively share data and data subscribers to receive data selectively [1]. Data subscribers subscribe to data that they are interested in from data publishers such as large companies, large research institutes, hospitals and government agencies. It’s difficult to capture and manage mass data based on traditional infrastructures and tools due to the large quantity and high speed of big data. With the increase of data volume the cloud platform is the most suitable one for data publication and subscription due to its huge storage, strong computing power and good economical utility [2].

However, the cloud server is semi-honest, it collects and speculates on private information about data and subscribers’ interests, while some malicious users peep at data that they don’t have access rights, resulting in private data disclosure. Privacy issues become the focus of data publish-subscribe system on the cloud platform.

In order to solve the problem of how to search keywords on ciphertext, the concept of searchable encryption is proposed, which enables cloud servers to match encrypted data tags with subscribers’ interests. Specifically, data publishers use searchable encryption to encrypt labels associated with their data before uploading their data to the cloud server, and subscribers subscribe data through trapdoors instead of expressing their interests in a clear form. Therefore, the cloud server can recommend appropriate data to subscribers without knowledge of potential data or interests. The first symmetric searchable encryption[3-4] scheme was proposed by song et al. [5]. In this scheme, similar stream cipher method is used to encrypt, linear scanning is used to find specific keywords, and the function of keyword...
retrieval on ciphertext is realized. According to the nature of symmetric encryption system, the data file in symmetric searchable encryption system and the keyword trap door to be searched must be encrypted with the same key, so the symmetric searchable encryption system is more suitable for personal data storage and other application scenarios. Different from symmetric searchable encryption schemes, most public key searchable encryption schemes [6-7] are constructed to encrypt the data, in the whole encryption process, the data encryptor does not need to negotiate with the data encryptor, which makes the scheme more suitable for multi-user data sharing and other fields, and its application scenario is broader than that of symmetric searchable encryption.

In order to ensure that only subscribers with access rights can access the specified data, some qualifications should be evaluated before the task matches. Attribute-based encryption [8-9] is a useful encryption data access control technology. G Wang et al., by combining the hierarchical identity encryption mechanism (HIBE) with the attribute encryption machine (cp-ABE) based on ciphertext strategy, and adopting a compromise in performance and scalability, solve the problem of safe and effective use of cloud storage services in small and medium-sized enterprises. On the basis of ABE, the property base encryption in cloud storage system is improved, and the access control in cloud environment [10-15] has been realized. ABE is an ideal password primitive for fine-granularity access control, so subscribers can use it to implement specific access requirements.

In order to solve these problems, the scheme combines three ciphergraphs attribute-based encryption, proxy re-encrypted searchable encryption and zero-knowledge proof. Based on attribute-based encryption, data publishers can encrypt data by requirements of access rights so that only data subscribers with access rights can obtain and decrypt data. In addition, in the case of zero-knowledge proof, data subscribers can prove to the cloud server that their access rights meet the data publisher’s requirements without revealing interest. Proxy-Encrypted Searchable Encryption enables cloud servers to match encrypted data tags with subscriber interests without revealing data or interest privacy.

This paper proposes a service of secure data publication and subscription on cloud platforms, which achieves data label matching and the access rights verification of data subscribers are realized to protect the privacy of users. The security of secure data publishing and subscription scheme for cloud platform is discussed, and its performance in computing and storage overhead is analyzed theoretically. The feasibility of the scheme is verified by simulation.

2. Problem of Statement

2.1. System Model

As shown in Figure 1, the model of data publish-subscribe system includes four entities: cloud server, key management server, data publisher and data subscriber.

![Figure 1 The Model of Data Publish and Subscribe System](image_url)
1) Cloud Server (CS). CS provide subscription services to customers. It receives data from data publishers, recommending data to them based on their interests, and helping data publishers choose data subscribers with access rights.

2) Key Management Server (KMS). The key management system is responsible for key generation and attribute authorization. It is responsible for the generation of the data publisher or data subscriber key, and authorizes the attribute of the data subscriber.

3) Data Publishers. Data publishers label their data with a few labels and specify requirements of access rights for data subscribers. Then, publish their data on the CS.

4) Data Subscribers. Data subscribers have a set of attributes assigned by KMS to show their subscription capabilities and post their own interests to CS.

2.2. Threat Model
In this scenario, the key management server is trusted and does not conspire with the cloud server and the data publisher and the data subscriber. The cloud server is not trusted, and he will follow the established protocol, but try to gather as much information as possible about the data and the subscriber's interests. Data subscribers in our system are untrustworthy, and they may malicious claim that their rights meet the requirements of the data publisher and that the qualified data subscriber does not actively disclose information related to the data to others.

2.3. Design Goals
1) Privacy Protection. CS is unable to obtain any information about the data content and the corresponding labels. In addition, it can’t obtain the interest of data subscribers and what attributes they have.

2) Permission verification requirements. The data publisher can access their data access, so that only qualified data subscribers of the right can prove their rights to the data publisher's needs to the cloud server.

3) Good Economical Utility. The cost of computation and storage is low.

3. Scheme Realization
This scheme includes six phases: subscription system establishment, user registration, data encryption, interest encryption, subscription matching and user revocation.

3.1. Subscription System Establishment
ABE key generation. Generates the multiplication cycle \( G_s \) and \( G_y \), \( g \) is one of a generators of \( G_s \), \( e: G_s \times G_y \rightarrow G_y \) is a bilinear pair. Select two random numbers \( \mu \) and \( \tau \) and two hash functions: \( H_s: \{0,1\} \rightarrow Z_p \) and \( H_y: G_y \rightarrow Z_p \). So:

\[
PK_{abe} = \{ p, g, e(g_s, g_y)^\mu, g^\tau, H_s, H_y \}, MK_{abe} = g^\mu 
\]

(1)

PRE-SE key generation. Generates the multiplication cycle \( G_y \) and its generator \( g_y \), which randomly generates \( y \in Z_p \) and calculates \( l = g_x^y \). Select a hash function \( H_x: G_s \rightarrow \{0,1\} \) and a pseudo-random function \( f \) with a random key \( m \). So:

\[
PK_{se} = \{ q, g, l, G_y, H_x, f \}, MK_{se} = \{d, m\} 
\]

(2)

Finally, publishes \( PK = (PK_{abe}, PK_{se}) \) and retains \( MK = (MK_{abe}, MK_{se}) \).

3.2. User Registration
Joining the subscription system. Each data publisher or subscriber registers on KMS. For data publisher or subscriber with a unique identity \( ID_s \), the KMS selects a random number \( uk_s \in Z_p \) as publisher or subscriber's PRE-SE key and sends it to the publisher or subscriber. At the same time, KMS calculates the re-encryption key \( rk = d - uk_s \) and sends \( (ID_s, rk) \) to the CS. The CS maintains a list of users \( UL \) and each user’s re-encryption key.
In addition, for the data subscriber whose attribute set is \( S \), KMS selects a random value \( v \in \mathbb{Z}_p \) and calculates the corresponding key as following:

\[
L = g_v^x, M = g_v^y, \quad \forall x \in S \quad K_{i} = H_{x}(x)^v
\]  

(3)

Then, it is sent to the data subscriber.

3.3. Data Encryption

Before uploading the data to the CS, the data publisher first encrypts the data content and data labels. Then, the CS will re-encrypt the data label ciphertext.

3.3.1. Data Content Encryption

The data content is encrypted by the data publisher using a symmetric encryption scheme. Firstly, it selects a random number \( \xi \in \mathbb{G} \) that can be converted into a symmetric key and encrypts \( M \), getting \( CT_{\mu} = SE_\mu(M) \). Then, data publisher encrypts \( \xi \) with ABE. Specifically, it defines the task requirements as LSSS structure \((D, \varphi)\), where \( D \) is the matrix of \( m \times n \), \( \varphi \) associates each row of \( D \) with a unique property. In order to share the encryption index \( m \in \mathbb{Z}_p \), the data publisher selects \( m - 1 \) random value \( Z_1, Z_2, \ldots, Z_{m-1} \in \mathbb{Z}_p \) to form a vector \( \overline{\xi} = (Z_1, Z_2, \ldots, Z_{m-1}) \) and calculates \( \lambda_i = (D \xi)_i, j \in [1,m] \) and \( \lambda_i \) can be regarded as a secret slice assigned to the attribute \( \varphi(j) \). Ciphertext is as following:

\[
CT_\xi = \{(D, \varphi), E = \xi \cdot e(g_v, g_v)^m, E' = g_{\gamma}^{m(i)}, E_0 = g_{\mu}^{\varepsilon(i)} H_\mu(\varphi(j))\} j \in [1,m]
\]  

(4)

3.3.2. Data Label Encryption

\( L_P = \{L_{P_1}, L_{P_2}, \ldots, L_{P_l}\} \) is the set of labels related with the data, the secret key of data publishers’ ID, identity is \( u_{k_i} \). For each label \( L_{P_i} \) in \( L_P \), the data publisher selects a random number \( u_i \in \mathbb{Z}_p \), and calculates \( P'_i = (P_i', P_i, P_i) \), in which \( P_i' = g_{\alpha}^{i \cdot \gamma}, \varepsilon_i = f_i(LP_i) ; P_i = (P_i')^m = g_{\beta}^{i \cdot \gamma} \cdot \alpha d ; P_i = H_i(I') \). Then, the data publisher will upload to CS.

3.3.3. Labels Re-encryption

Firstly, the CS queries \( UL \) to get \( r_{k_i} \) of the data publisher with the identity \( ID_i \). Secondly, CS re-encrypts the data label ciphertext. For each label ciphertext \( P'_i \), it calculates \( P_i = (P_i', P_i) , P_i = (P'_i)^m \cdot P_i = P_i' = H_i(I') \). Finally, ciphertext is generated as:

\[
CT = <ID_i, CT_{u_i}, CT_{\xi}, [W_{i}]_{init}> \]  

(5)

3.4. Interest Encryption

Data subscribers need to encrypt and upload their subscription interest to the CS. Subscriber’s interest will be encrypted by the data subscriber to protect privacy and then matched by CS re-encryption.

3.4.1. Data Subscribers’ Interest Encryption

Firstly, data subscriber sets data’s interest as an LSSS structure \((C, \psi)\), in which \( C \) is the matrix of \( m \times n \), \( \psi \) associates each row of \( C \) to a unique interest keyword. As for \( i \in [1,n] \), the data subscriber selects a random number \( t_i \in \mathbb{Z}_p \) and calculates \( T_{Q_i} = (t_{Q_i}, t_{Q_i}') \). Among them, \( t_{Q_i}' = g_{\alpha_i}, t_{Q_i} = I_i \cdot g_{\alpha_i} \cdot \psi(i) \cdot u_k \), is the secret key of the data subscriber with \( ID \), identity, \( \sigma_i = f_i(\psi(i)) \). Secondly, data subscriber will upload \(<C, T_{Q_i}, t_{Q_i}, \sigma_i>\) to the CS. In the process of uploading, the keyword mapping function \( \psi \) is removed from the LSSS structure to protect interest privacy.

3.4.2. Data Subscribers’ Interest Re-Encryption

As for \( i \in [1,n] \), CS re-encrypts \( T_{Q_i} \) and then generates trapdoor component \( T_{Q_i} = t_{Q_i}^{r_i} \cdot t_{Q_i}' = I_i ^m \). Corresponding trapdoor components are the same for same keywords and different data’s interests. CS
builds a trapdoor component list \( TCL \) for the interests of data subscribers, where each keyword only needs to storage a trapdoor component. Therefore, for each \( i \in [1,m_1] \), the CS firstly asks \( TQ_i \) if it is in the \( TCL \). If not, it will add \( TQ_i \) to the list.

Finally, the data subscriber’s interest trapdoor is \( < ID_j, C, X = \{Index(i)\}_{i \in [m_1]} > \), \( Index(i) \) is the \( TCL \) index of trapdoor component \( TQ_i \) which relates with the \( i \) row of \( C \).

### 3.5. Interest Matching and Rights Verification

In order to accurately push subscriptions, CS will test whether publishers’ data labels match subscriber’s interests. In addition, it will judge if those matching subscribers meet access rights. Details as following:

#### 3.5.1. Subscription Matching

For data ciphertext \( CT_i \), CS draws the keyword ciphertext \( \{P_j\}_{j \in [0,m_2]} \). After that, for each \( P_j \), the CS compares it to \( TQ_j \) in the \( TCL \) to find the matching ciphertext. Particularly, it verify \( H_i(P_j) = (TQ_j)^{x} = P_2 \). If it matches, \( i \) will be added to the index \( Y \), \( i \) represents the index of the trapdoor component \( TQ_i \) in the \( TCL \). Then, for the trapdoor \( < ID_j, C, X = \{Index(i)\}_{i \in [m_1]} > \), CS calculates \( W = X \cap Y \) and recovers \( J = \{j | Index(j) \in W \cap j \in [1,m_1] \} \). Finally, if it exits \( \{\alpha_i\} \), satisfying \( \sum \alpha_i \odot C_j = (1,0, \cdots, 0) \) then the tag set of the data matches the interest of the subscriber.

#### 3.5.2. Rights Verification

Before assigning data to match subscribers, CS will verify if the subscriber’s attribute set meets the data publisher’s specified access rights requirements. CS randomly generates \( \theta \in Z \), calculates \( F = g^\theta \), and sends \( < CT_i, F > \) to matching subscribers. After receiving it, the subscriber first checks if its attributes meet the task requirements \( (\theta, \varphi) \) in \( CT_i \). If so, there will be a set of constants \( \{\delta_i\} \) which makes \( \sum \delta_i D_i = (1,0, \cdots, 0) \). \( I \) is defined as \( I = \{i | \varphi(i) \in S \cap i \in [1,m_2] \} \). The subscribers recover \( \xi \) by calculating and decrypting \( CT_i \) as following:

\[
\prod_{i \in I}(e(M,E_i)e(L_{\varphi(i)},E_i))^{\delta_i} = \xi
\]

Then, the subscriber calculates \( T = F^{H_i(\xi)} \) and sends it to the CS. After verifying \( T = (E)^{\xi} \), CS selects subscribers that meet the requirements and sends them to qualified subscribers.

#### 3.6. Users Revocation

In order to revoke the misbehaving data publishers and data subscribers in the subscription system, KMS sends the user \( ID_i \) to the CS and demands it to cut out re-encryption key \( rk_i \) in \( UL \).

### 4. Security Analysis

#### 4.1. Data Privacy

When uploading the data, the data publisher uses the symmetric key to encrypt the data, so it is necessary to export the symmetric key \( \xi \) in order to decrypt the data. In the discrete logarithm problem, it is difficult for cloud servers to export symmetric keys from \( E' = g^\xi \) and \( T = F^{H_i(\xi)} \), so the privacy of encrypted data is protected. For data label privacy, this scheme is based on elgamal encryption, elgamal encryption in clear text attack is secure, so data label privacy is protected.

#### 4.2. Data Subscriber Privacy

Although the same keyword has the same trapdoor component in different subscriber interests because the discrete logarithm \( DL \) problem is difficult for \( g \), it is impossible to damage the keyword privacy.
about interest by ciphertext attack. For subscriber attribute privacy, CS can only know if the subscriber's attributes meet the publisher's rights requirements, but it can’t get the subscriber's specific attributes. Therefore, this scheme can protect subscriber privacy.

4.3. Access authentication security requirements

To prove to CS that they have permission to view data, subscribers should provide $T = g_a^{m(I)}$ to CS. Based on the Computational Diffie-Hellman hypothesis, subscribers cannot calculate $g_a^b$ only from $g_a^{m(I)}$ and $g_a^{m(I)}$. Therefore, only those data subscribers who satisfy the access structure can restore $\xi$ and calculate $g_a^{m(I)}$ successfully verified by CS.

5. Performance Evaluation

In this paper, the performance of this scheme is evaluated from the costs of computation and a large number of experiments are carried out to verify the practicability of the scheme.

5.1. The Analysis of Theory

In Table 1 and Table 2, the calculated amount of data publisher, data subscriber, and cloud server in encryption, re-encryption, permission verification, and subscription matching is displayed. In addition, in addition to that long calculation time of the pairing operation, the calculation is more complex, and the other operation is relatively simple, and the operation speed is fast.

In the following tables, $A$ : exponent arithmetic, $P$ : pairing operations, $S$ : multiply operation $H$ : hash operations. $m, m'$ : attributes’ numbers used for ABE encryption and decryption; $m, m'$ : labels’ number in the set and subscribers’ interest;

Table 1. User-generated Computational Cost

| User     | Data Publisher | Data subscriber |
|----------|----------------|-----------------|
| Data Encryption | $2m_a + 3A + (m_a + 1)S$ | $m_k(3A + H)$ |
| Label Encryption | $m_k(3A + H)$ | $m(I)(A + H)$ |
| Interest Encryption | $(m'_I + 1)A + 2m'_S + (2m'_I + 1)P$ | $m_k(A + S)$ |
| Rights Verification | $m_k(A + S)$ | $m(I)(A + S)$ |

Table 2. Calculation the Cost of CS

| User | CS |
|------|----|
| Label Re-encryption | $m_k(A + S)$ |
| Interest Re-encryption | $m(I)(A + S)$ |
| Subscription Matching | $\leq m_kN(S + H)$ |

5.2. Simulation Realization

The simulation is implemented on a laptop with 16GB RAM, and Ubuntu16.04 operating system. This scheme sets the number of attributes and keywords involved in the simulation to 10 to 100, and the total number of elements in the TCL is 100. The test runs 100 times, and the run time of the data publisher, data subscriber, and CS is shown in figure 2.

Figure 2(a) shows the run time of the subscriber’s interest encryption and rights verification. The subscriber's interest encryption time is almost the same as the publisher’s label encryption time. The two kinds of time linearly increase as the amount of data and the number of label increase. Figure 2(b) shows the running time of data encryption and label encryption, both of which linearly increase as the amount of data and the number of label increase. The subscriber can complete the capability verification operation in a short time because this scheme applies the LSSS structure generation method, which mainly includes ABE keyword re-encryption, interest re-encryption and subscription matching. For 100 keywords, the time of keyword re-encryption and interest re-encryption are both less than 20 ms, which indicates that CS re-encryption operation is efficient.
Meanwhile, CS needs to test each label of data with each component in TCL, the calculation time is longer, and the subscription matching time increases as the number of keywords increases. However, for an expensive subscription matching process, each task only needs to be executed for one time.

6. Conclusion

This paper presents a cloud platform-based security data distribution and subscription scheme, which enables the cloud server to efficiently realize the matching of the subscriber's interest and the data tag. In the scenario of this article, the subscriber can specify flexible interests to receive the data they want, while the publisher is able to access their data access rights verification requirements with only the subscribers with access rights to get the data. Cloud servers assign data to subscribers according to their interests and access rights, so that only subscribers with access rights are entitled to data of interest to them. In addition, in this scheme, the privacy of data publishers and data subscribers can be effectively protected. At the same time, this scheme also supports users to cancel, can cancel the improper behavior of data publishers and data subscribers in the subscription system, and improve the security of the system.

The next step is to study the application of attribute signature in permission verification of data publishing and subscription system. In this scheme, in order to verify that the matching subscriber has access to the data, CS first needs to send ciphertext and send it to the subscriber, which can be regarded as an interactive zero-knowledge proof. Attribute-based signature (ABS) [16-17] enables signatories whose attributes satisfy predicates to approve messages without disclosing their attribute privacy, so subscribers can use it to prove to CS that their attributes meet access requirements without interacting with CS.
References

[1] Yang, K., Zhang, K., Jia, X., Hasan, M. A., & Shen, X. (2016) Privacy-preserving attribute keyword based data publish-subscribe service on cloud platforms. Information Sciences, 387:116-131.

[2] Hayes, B. (2008) Cloud computing. Communications of the Acm, 51(7), 9-11.

[3] Wang, S., Zhao, D., Zhang, Y. (2017) Searchable attribute-based encryption scheme with attribute revocation in cloud storage. PLoS ONE, 12(8):1-20.

[4] Deng, Z., Li, K., Li, K., Zhou, J. (2017) A multi-user searchable encryption scheme with keyword authorization in a cloud storage. Future Generation Computer Systems, 72, 208-218.

[5] Song, D. X., Wagner, D., Perrig, A. (2000) Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE Symposium on Security and Privacy. California. pp. 44-55.

[6] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G. (2004) Public key encryption with keyword search. In: International conference on the theory and applications of cryptographic techniques. Heidelberg. pp. 506-522.

[7] Boneh, D., Waters, B. (2007) Conjunctive, subset, and range queries on encrypted data. In: Theory of Cryptography Conference. Heidelberg. pp. 535-554

[8] Shu, J., Jia, X. (2016) Secure task recommendation in crowdsourcing. In: 2016 IEEE Global Communications Conference (GLOBECOM). Washington. pp. 1-6.

[9] Gorbunov, S., Vaikuntanathan, V., Wee, H. (2015). Attribute-based encryption for circuits. Journal of the ACM (JACM), 62(6), 45.

[10] Waters, B. (2011, March). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: International Workshop on Public Key Cryptography. Heidelberg. pp. 53-70.

[11] Yang, K., Liu, Z., Jia, X., Shen, X. S. (2016). Time-domain attribute-based access control for cloud-based video content sharing: A cryptographic approach. IEEE Transactions on Multimedia, 18(5), 940-950.

[12] Yang, K., Jia, X. (2014). Expressive, efficient, and revocable data access control for multi-authority cloud storage. IEEE transactions on parallel and distributed systems, 25(7), 1735-1744.

[13] Do, J. M., Song, Y. J., Park, N. (2011) Attribute based proxy re-encryption for data confidentiality in cloud computing environments. In: 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering. Washington. pp. 248-251.

[14] Lee, C. C., Chung, P. S., Hwang, M. S. (2013) A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments. IJ Network Security, 15(4), 231-240.

[15] Cha, B., Seo, J., Kim, J. (2012) Design of attribute-based access control in cloud computing environment. In: Proceedings of the International Conference on IT Convergence and Security 2011. Dordrecht. pp. 41-50.

[16] Ge, A. J., Ma, C. G., Zhang, Z. F. (2012) Attribute-based signature scheme with constant size signature in the standard model. IET Information Security, 6(2), 47-54.

[17] Ding, S., Zhao, Y., Liu, Y. (2014) Efficient traceable attribute-based signature. In: 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. Beijing. pp. 582-589.