Secure Online/Offline Attribute-Based Encryption for IoT Users in Cloud Computing

Xiang Li[1], Hui Tian[1(✉)], and Jianting Ning[2]

[1] College of Computer Science and Technology, National Huaqiao University, Xiamen, People's Republic of China
xlics@stu.hqu.edu.cn, cshtian@126.com

[2] School of Computing, National University of Singapore, Singapore, Singapore
jtning88@gmail.com

Abstract. To ensure the security of mass data sharing in the Internet of Things, the cloud computing platform is supposed to provide data-storage services. The ciphertext-policy attribute-based encryption (CP-ABE) schemes has attracted wide-scale attention since users can access the cloud platform in a fine-grained manner. However, there are still some problems in the existing CP-ABE schemes when directly applied in the Internet of Things environment. The problem of simultaneously achieves large computational cost in the encryption and decryption. Moreover, the privacy of access control policy actually still remains unresolved. To fill the gap of the existing schemes, this paper proposes a suitable data sharing scheme for IoT devices which can't always be online. We use the online/offline CP-ABE technology with privacy, while hiding the access control structure and reducing the computational cost of the devices when they are online. The asymptotic complexity comparison also shows that our scheme achieves high computation efficiency.

Keywords: Internet of Things · Cloud computing · Online/offline encryption · Privacy protection · Hidden access structure

1 Introduction

Nowadays, the Internet of Things (IoT) has attracted the attention of researchers in academia and industry. With the development of Internet of Things technology continuously, it is widely used in some areas, such as aviation, rail transit, safe city, industrial manufacturing, logistics management, medical and health, and smart home, etc. However, the computing and storage resources of IoT devices are often limited, which greatly limit the application of the Internet of Things in various fields. Cloud computing provides an on-demand service that provides users with useful and convenient network access. Therefore, cloud computing services can solve the problems, which include technically limited of IoT devices, and satisfy the exchange and sharing requirements of large data volume that the Internet of Things requires. However, cloud computing service providers are not completely trusted. When the data owner stores the data on the cloud server, it loses absolute control over the data. Cloud service providers (CSP) may privately share data to unauthorized users when they are tempted by

R. Steinfeld and T. H. Yuen (Eds.): ProvSec 2019, LNCS 11821, pp. 347–354, 2019.
https://doi.org/10.1007/978-3-030-31919-9_22

interests. Cloud service providers may also receive internal and external attacks, resulting in authorization exceptions and data leakage for their users and roles. In the IoT environment, sensor devices are characterized by massiveness, device differentiation and security and privacy protection difficulty. Thus, Internet of Things users have higher requirements for data security and privacy protection.

Attributes-based encryption (ABE) can well meet the needs of data confidentiality and fine-grained access control in the Internet of Things. We divide ABE into two categories: KP-ABE [1] and CP-ABE [2]. In the CP-ABE scheme, the access policy is related to the ciphertext, while the key is connected to the attribute. KP-ABE scheme is the opposite. For reducing equipment burden, some selectively efficient ABE schemes [3–6] were proposed, such as outsource data to third parties which can save local storage and computing resources. At the same time, some efficient online/offline encryption solutions [7–10] have proposed.

In the above solution, the data provider needs to be online in real-time while the ciphertext is related to the access control policy, which resulting in increasing the encrypting computational overhead. In addition, during the decryption phase, the cloud service provider needs to send the access control policy to data users along with the ciphertext, while the access policy may contain some sensitive information. If the access control policy for this data is compromised, it may be illegal. Therefore, how to reduce the encryption computing overhead while realizing the hiding of access policies has become one of the urgent problems in the cloud computing environment.

In this paper, we proposed a secure online/offline attribute-based encryption for IoT users in cloud computing. Our scheme mainly uses the online/offline ABE technology to solve the problem of large computing cost in ABE that the most expensive encrypt operations have been executed in the offline phase. What's more, in order to protect the security of access control structure. When the user uploads and downloads the ciphertext, the access control structure will be hiding.

2 Related Work

Currently, the attribute-based encryption (ABE) system has been widely used. Its main dependency is to use a set of attributes that describe the user's identity to represent the identity of the user. The data user's key is generated by the authorization center according to each user's attribute set, which is a set of characteristic information of the data user. Matching relationship between the user attribute set and the access structure, the decryption capability of the user is determined by realizing the control of the ciphertext. The data provider does not need to distribute the corresponding key for each data consumer. They only need to manage the attributes of the corresponding file by modifying the access control structure, which greatly increases the flexibility of access control. Considering the computational burden of the IoT device during the encryption and decryption phase, it is mainly to delegate the complex calculation by constrained IoT devices to the enough computing power nodes at present. In 2010, to address the

burden of key distribution and data management, Yu et al. [3] strengthened the attribute-based access strategy, while allowing data owners to put most of their computing tasks on the cloud server. Hur et al. proposed an attribute-based access control method [4] using CP-ABE to enforce access control policies with efficient attribute and user revocation capability. This fine-grained access control method is implemented by the ABE and the double encryption mechanism of the selective group key distribution method in each attribute group. For the ABE outsourcing decryption scheme, in the literature [5], they adopt the bilinear pairing method to realize the outsource decryption, that is, the calculate operation in the resource-constrained client is outsourced to the semi-trusted third party. However, in the above scheme, the user still needs to operate the index and multiplication operations multiple times. Green et al. [6] proposed an outsourced decryption scheme based on LSSS matrix, which allows the cloud to convert ciphertexts satisfying user attributes into ciphertext of constant size, while the cloud cannot read any part of the user's message.

Meanwhile, IoT devices include not only sensor devices with weak underlying computing capabilities, but also devices with strong computing power. These devices are sufficient to perform encryption and decryption work, but there is no guarantee that resources will be online in real time. Online/offline cryptography is an effective tool for improving encryption efficiency. The complex encryption operations are preprocessed by using high-performance devices that makes lightweight devices only need to perform a small amount of simple operations. Hohenberg [7] first proposed constructing an online/offline ABE encryption scheme in which the computational work is divided into two phases: the offline phase (preparation process) and the online phase. In 2015, Datta [8] combines searchable encryption and access control with security proof. Later, Cui [9] uses outsourced ABE technology to place most of the decryption work on the cloud server while implementing keyword search, which greatly reduces the user's computational cost. Considering resources with limited resources, Liu [10] quickly performs keyword encryption or token generation by consuming costs to the offline phase, while the mobile device is powered without consuming battery. However, the above operations do not consider the operation of the multi-authority ABE. We know that the computing power of sensor devices is limited. Before sending the sensitive message to users, we must encrypt these messages for protecting the privacy. This is a great challenge for the IoT sensor devices. Consequently, it would be much better to do a part of encrypt operation in the free time.

3 System Design

3.1 System Model and Design Goals

As shown in Fig. 1, the system architecture of our proposed scheme consists of four entities: a cloud service provider (CSP), an attribute authority (AA), data owners (DOs) and data users (DUs).

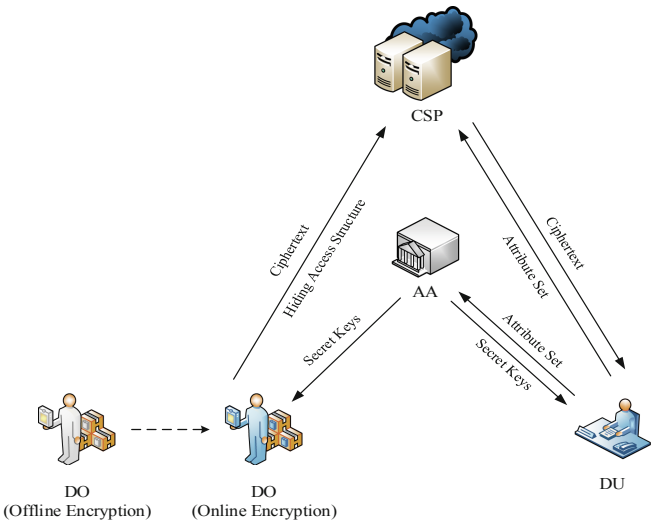

Fig. 1. System architecture of the scheme in cloud model

- CSP is responsible for storing a large amount of data generated in the Internet of Things which is composed of multiple servers. It has strong computing power, which is honest and curious.
- AA is an independent attribute authority that can generate a public key and a master secret key for DO by executing an *AuthoritySetup* algorithm. After receiving the attribute set from the user, it returns the attribute private key generating by *SecretKeyGen* algorithm.
- DO is the owner of the data. In the IoT environment, the data owner is a resource-constrained entity. It cannot guarantee that its computing resources are always online. Since most of costly computations can evaluated by running *Offline.Encrypt* algorithm, the efficiency of encryption can be greatly improved because *Online. Encrypt* algorithm only incurs little computation costs.
- DU refers to the actual user of the actual data in the Internet of Things. The entity can obtain a plaintext message through the *Decrypt* algorithm.

In our scheme, we prescribe some security assumptions to meet the real IoT environment's needs. we assume AA is fully trusted while does not reveal user data and collude with users. The CSP is semi-trusted (honest-but-curious) entity which can honestly save user-uploaded data and perform user's tasks. But it may be curious about the data content. Meanwhile, users are not completely trusted. Malicious users may hide their identity to obtain sensitive information.

3.2 Proposed Scheme

This section is dedicated to proposing our scheme, which has six algorithms: *GlobalSetup*, *AuthoritySetup*, *SecretKeyGen*, *Offline.Encrypt*, *Online.Encrypt*, *Decrypt*.

System Initialization. Similar to the scheme [11], this phase is required to initialization the public parameter and to generation public keys and secret keys.

GlobalSetup$(1^k) \rightarrow (PP)$ This algorithm inputs a security parameter 1^k, and then outputs public parameter

$$PP = \{g, h, e, p, \mathbb{G}, \mathbb{G}_T, H\}.$$

The algorithm chooses two random generators g, h from $\mathbb{G}$. And selects two bilinear groups $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ of prime order e, p. Furthermore, we employ a strong collision-resistant hash function $H : \{0,1\} \rightarrow \mathbb{G}$.

AuthoritySetup$(PP) \rightarrow (\mathrm{PK}, \mathrm{MSK})$. Taking as input the system public parameters PP, the authority chooses α, β, γ randomly from $\mathbb{Z}_p$. Then, AA picks random generators u from $\mathbb{G}$. AA publishes the public key and the master secret key

$$\mathrm{PK} = \left\{e(g,g)^{\alpha}, h^{\beta}, g^{\gamma}, \mathrm{u}\right\}, \mathrm{MSK} = \{\mathrm{PK}, \alpha\}.$$

Secret Key Generation. In this phase, the attribute authority issues a key extract algorithm with hidden access structure, which not get any information about user's identifier and attributes to protect user's privacy.

SecretKeyGen$(PP, \mathrm{GID}_U, \mathrm{PK}, \mathrm{U}, De_{ID}, CM_{ID}) \rightarrow (SK_U)$. Firstly, data user execute commitment algorithm Commit$(PP, \mathrm{GID}_U) \rightarrow (CM_{ID}, De_{ID})$ and send (CM_{ID}, De_{ID}) to attribute authority. Then AA take public parameters PP, an attribute set $\mathrm{U} = \{A_1, A_2 \ldots\ldots, A_n\}$, the public key PK and commitment (CM_{ID}, De_{ID}) as input. Then if Decommit algorithm output the right sight, it computes $K_1 = g^{\beta}$, and for i = 1 to n, it computes $K_{i,1} = (u^{A_i} h^{\beta})^{t_i}, K_{i,2} = g^{t_i}$. Otherwise, it outputs the error messages and the *SecretKeyGen* algorithm is terminated. The algorithm outputs

$$SK_U = \left\{K_1, \left\{K_{i,1}, K_{i,2}\right\}_{i \in [1,n]}\right\}$$

which authority picks $t_1, t_2, \ldots, t_n \in \mathbb{Z}_p$.

Encryption. This phase is divided into the offline data creation and online data creation. Data owner who is resource-limited generates offline ciphertexts by running *Offiline.Encrypt* and generates the final ciphertext by running *Online.Encrypt*.

Offline.Encrypt$(PP, \mathrm{PK}) \rightarrow \left(\mathrm{CT}_{Off}\right)$. The offline encryption algorithm takes in the public parameters only. The algorithm randomly picks $\mathrm{s}, \lambda \in \mathbb{Z}_p$ and computes $C_0 = g^s$. Next it chooses random $\tau_j, x_j \in \mathbb{Z}_p$ for each $\mathrm{j} \in [1, n]$ The algorithm sets key $= e(g,g)^{\alpha s}, C_{j,1} = g^{-\tau_j}, C_{j,2} = \left(u^{x_j} h^{\beta}\right)^{\tau_j}, C_{j,3} = h^{x_j}$. The algorithm outputs

$$\mathrm{CT}_{Off} = \left\{key, C_0, \{C_{j,1}, C_{j,2}, C_{j,3}, x_j, \tau_j\}_{\mathrm{j} \in [1,n]}\right\}.$$

Online.Encrypt$\left(PP, \mathrm{U}, CT_{Off}, \mathrm{PK}\right) \rightarrow (\mathrm{CT})$. The online encryption algorithm takes as input the public parameters PP, the data owner's attribute U, an offline ciphertext CT_{Off} and the public key PK. The owner computes $P_j = e\left(h^{\beta}, H\left(U_j\right)\right)$ for each

j $\in [1, Y]$, where U_j denotes attribute of access policy T and Y is the number of attributes in T. Next, the access policy T is converted to LSSS access control structure (M, ρ), while we use P_j to replace the attribute U_j in the access policy. The structure control matrix M is an l × n matrix and l ≤ P. It set the vector $\mathbf{y} = (s, y_2, \ldots, y_n)^T$ in which $y_2, \ldots, y_n \in \mathbb{Z}_p$ is random where T denotes the transpose of the matrix. Then it computes a vector of shares of s as $(\lambda_1, \lambda_2, \ldots, \lambda_l)^T = M\mathbf{y}$. The algorithm computes $C_{j,4} = \lambda_j - x_j, C_{j,5} = \tau_j(A_j - x_j)$. Eventually, the algorithm sets the ciphertext as

$$\mathrm{CT} = \left\{(M, \rho), C_0, C_1, \left\{\{C_{j,1}, C_{j,2}, C_{j,3}, C_{j,4}, C_{j,5}\}_{j\in[1,P]}\right\}\right.$$

Decryption. In this phase, data user downloads a ciphertext CT from CSP, and performs the following algorithm Decrypt based on secret key SK_u to recover the consequent message.

Decrypt(SK, CT) → key. It takes a secure private key $SK_U = \left\{K_1, \left\{K_{i,1}, K_{i,2}\right\}_{i\in[1,n]}\right\}$ from *SecretKeyGen* algorithm and a ciphertext $\mathrm{CT} = \left\{(M, \rho), C_0, C_1, \left\{\{C_{j,1}, C_{j,2}, C_{j,3}, C_{j,4}, C_{j,5}\}_{j\in[1,P]}\right\}\right.$ for hiding access structure (M, ρ). If SK_U does not satisfy the hiding structure, then the algorithm outputs an error message. Or else, the algorithm computes constants $\sum_{i\in I} w_i\lambda_i = s$ for making $w_i \in \mathbb{Z}_p$, I⊆{1, 2, . . ., 1} and setting λ_i is the result of the secret s share. The cloud computes

$$e(g,g)^{\alpha s} = \frac{e(K_0, C_0)}{e\left(h^{\sum_{i\in I} w_i \cdot C_{j,4}}, K_1\right) \cdot \prod_{i\in I}\left(e\left(K_{i,1}, C_{j,1}\right)\right) \cdot e\left(K_{i,2}, C_{j,2} \cdot u^{C_{j,5}}\right) \cdot e\left(K_1, C_{j,3}\right)\right)^{w_i}}$$

where j is the index of the attribute A_i in S (it depends on i).

4 Performance Evaluation

Table 1. Computation cost comparisons of online/offline attribute-based encryption schemes

Schemes	Offline encryption	Online encryption	Decryption (user side)
OOABE [7]	(3**N** + 3)**E** + **N**·**M**	(|**A**| + 1)**M**	3k**P** + 2k**E** + 3km
DCP-ABKS-CKDO [11]	2**M** + (5**M** + **G**)	**E** + **M**+2(**M** + **G**)	2k**E** + 2km
ABDS [12]	3**E**	4**P** + (2k + 2)**E** + (3k + 2)**M** + **H**	3k**P** + 2k**E** + 3km
OOABKS [9]	**G** + 4**E** + **N**·**M**	(|**A**| + 1)**M** + **G**	k**P** + k**E**
Ours	3**E** + 2**N**·**M**	**E** + **H**+(|**A**| + 1)**M**	3k**P** + k**E** + 3km

In this section, we provide estimate on the performance of the comparison results in Table 1, which compare the proposed scheme with some existing schemes in the efficient respects. The comparison results are summarized in Table 1, where **A**, **G**, **P**, **E** and **M** represent the number of attributes, the size of an element in $\mathbb{Z}_p$, a pairing operation, an exponentiation operation and a multiplication operation in bilinear

groups, respectively. And the complexity of the access structure is denoted by k. The symbol **H** is a chameleon hash operation. The symbol **N** means the size of offline ciphertext pool and it is determined by the size of the attribute universe (Table 2).

Table 2. Function compare between our scheme and other scheme

Schemes	Access structure	Hidden policy	Protect GID privacy	Online/Offline encryption
OOABE [7]	LSSS	No	No	Yes
DCP-ABKS-CKDO [11]	LSSS	No	No	Yes
ABDS [12]	LSSS	No	No	Yes
OOABKS [9]	LSSS	No	No	Yes
CSCD [13]	$(AND/OR)_m$	Yes	Yes	No
HCPABE [14]	AND	Yes	No	No
Ours	LSSS	Yes	Yes	Yes

We compare the proposed scheme with the state-of-the-art schemes with regard to the generation cost of the offline encryption cost, the online encryption cost and the decryption cost. In the online phase, our scheme reduces nearly half of cost compared with ABDS [12] while it less than other schemes. Because in our scheme, we only complete the encryption of using the access control policy in this phase. Our scheme incurs more computation costs than ABDS [12] in the offline phase, but the total workload of the user can be significantly reduced, which is suitable for the resource-limited users. Thus, the proposed scheme is efficient with respect to the computation costs on the user side and achieves security goals. Consider the function of our proposed scheme and several related schemes, we can observe that our scheme is superior to other schemes. All the online/offline schemes are allowed LSSS ciphertext policies.

5 Conclusion

In this paper, aiming at tackling the computation efficiency and weak data security issues, we proposed a secure online/offline attribute-based encryption for IoT users in cloud computing. Different from existing CP-ABE schemes, our scheme realizes efficient data encryption and privacy protection while heavy encryption computations are performed during the offline phase making the whole encryption phase faster and more efficient than existing schemes. For protect the access control, we hide the access structure in online phase and protect the data user key in secret key generation phase. Theoretical analysis indicate that the proposed data sharing scheme is extremely suitable for IoT users who have enough computing power but not real-time online. The security of our scheme is proven secure in the proposed selective chosen attribute set. The performance analysis show that our solution can be used to control access for shared data in an internet of things environment.

Acknowledgements. This research is supported by the National Natural Science Foundation of China under Grant Nos. U1536115 and U1405254, the Natural Science Foundation of Fujian Province of China under Grant No. 2018J01093, and the Subsidized Project for Postgraduates' Innovative Fund in Scientific Research of Huaqiao University No. 18013083012.

References

1. 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 - CCS 2006, Alexandria, Virginia, USA, pp. 89–98. ACM Press (2006)
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP 2007), pp. 321–334 (2007)
3. Yu, S., Wang, C., Ren, K., Lou, W.: Achieving secure, scalable, and fine-grained data access control in cloud computing. In: 2010 Proceedings IEEE INFOCOM, pp. 1–9 (2010)
4. Hur, J., Noh, D.K.: Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Trans. Parallel Distrib. Syst. **22**, 1214–1221 (2011)
5. Tsang, P.P., Chow, S.S.M., Smith, S.W.: Batch pairing delegation. In: Miyaji, A., Kikuchi, H., Rannenberg, K. (eds.) IWSEC 2007. LNCS, vol. 4752, pp. 74–90. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75651-4_6
6. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of ABE ciphertexts, 16
7. Hohenberger, S., Waters, B.: Online/offline attribute-based encryption. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 293–310. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_17
8. Datta, P., Dutta, R., Mukhopadhyay, S.: Fully secure online/offline predicate and attribute-based encryption. In: Lopez, J., Wu, Y. (eds.) ISPEC 2015. LNCS, vol. 9065, pp. 331–345. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17533-1_23
9. Cui, J., Zhou, H., Xu, Y., Zhong, H.: OOABKS: online/offline attribute-based encryption for keyword search in mobile cloud. Inf. Sci. **489**, 63–77 (2019)
10. Liu, Z., Jiang, Z.L., Wang, X., Huang, X., Yiu, S.M., Sadakane, K.: Offline/online attribute-based encryption with verifiable outsourced decryption. Concurr. Comput. Pract. Exper. **29**, e3915 (2017)
11. Xu, Q., Tan, C., Zhu, W., Xiao, Y., Fan, Z., Cheng, F.: Decentralized attribute-based conjunctive keyword search scheme with online/offline encryption and outsource decryption for cloud computing. Future Gener. Comput. Syst. **97**, 306–326 (2019)
12. Li, J., Zhang, Y., Chen, X., Xiang, Y.: Secure attribute-based data sharing for resource-limited users in cloud computing. Comput. Secur. **72**, 1–12 (2018)
13. Zhang, Y., Li, J., Yan, H.: Constant size ciphertext distributed CP-ABE scheme with privacy protection and fully hiding access structure. IEEE Access **7**, 47982–47990 (2019)
14. Phuong, T.V.X., Yang, G., Susilo, W.: Hidden ciphertext policy attribute-based encryption under standard assumptions. IEEE Trans. Inf. Forensics Secur. **11**, 35–45 (2016)