9-2020

Attribute-based encryption for cloud computing access control: A survey

Yinghui ZHANG

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Shengmin XU
Singapore Management University, smxu@smu.edu.sg

Jianfei SUN

Qi LI

See next page for additional authors

Published in ACM Computing Surveys, September 2020, 53 (4), Article number 3398036
DOI: 10.1145/3398036

Attribute-based Encryption for Cloud Computing Access Control: A Survey

YINGHUI ZHANG, School of Cyberspace Security, Xi'an University of Posts & Telecommunications, China and Singapore Management University, Singapore
ROBERT H. DENG, SHENGMIN XU, and JIANFEI SUN, Singapore Management University, Singapore
QI LI, Nanjing University of Posts & Telecommunications, China
DONG ZHENG, National Engineering Laboratory for Wireless Security, Xi'an University of Posts & Telecommunications, China and Westone Cryptologic Research Center, China

Attribute-based encryption (ABE) for cloud computing access control is reviewed in this article. A taxonomy and comprehensive assessment criteria of ABE are first proposed. In the taxonomy, ABE schemes are assorted into key-policy ABE (KP-ABE) schemes, ciphertext-policy ABE (CP-ABE) schemes, anti-quantum ABE schemes, and generic constructions. In accordance with cryptographically functional features, CP-ABE is further divided into nine subcategories with regard to basic functionality, revocation, accountability, policy hiding, policy updating, multi-authority, hierarchy, offline computation, and outsourced computation. In addition, a systematical methodology for discussing and comparing existing ABE schemes is proposed. For KP-ABE and each type of CP-ABE, the corresponding access control scenario is presented and explained by concrete examples. Specifically, the syntax of ABE is given followed by the adversarial model and security goals. ABE schemes are discussed according to the design strategies and special features and are compared in the light of the proposed assessment criteria with respect to security and performance. Compared to related state-of-the-art survey papers, this article not only provides a broader 12 categories of ABE schemes, but also makes a more comprehensive and holistic comparison. Finally, a number of open research challenges in ABE are pointed out.

This work is supported by the National Key R&D Program of China under Grant 2017YFB0802000, the National Research Foundation Singapore, under the National Satellite of Excellence in Mobile System Security and Cloud Security (NRF2018NCR-NSOE004-0001), Huawei International Pte Ltd Research Grant, the AXA Research Fund, the Shaanxi Special Support Program Youth Top-notch Talent Program, the National Natural Science Foundation of China under Grants 61772418, 61671377, and 61802303, the Key Research and Development Program of Shaanxi under Grant 2019KW-053, the Innovation Capability Support Program of Shaanxi under Grant 2020KJXX-052, the Basic Research Program of Qinghai Province under Grant 2020-ZJ-701, the Sichuan Science and Technology Program under Grant 2017GZDZX0002, and New Star Team of Xi'an University of Posts and Telecommunications under Grant 2016-02.
Authors' addresses: Y. Zhang (corresponding author), School of Cyberspace Security, Xi'an University of Posts & Telecommunications, China, 710121, Singapore Management University, Singapore, 178902; email: yhzhaang@163.com; R. H. Deng, S. Xu, and J. Sun, Singapore Management University, Singapore, 178902; emails: {robertdeng, smxu}@smu.edu.sg, sjf215.uestc@gmail.com; Q. Li, Nanjing University of Posts & Telecommunications, China, 210023; email: liqics@njupt.edu.cn; D. Zheng, National Engineering Laboratory for Wireless Security, Xi'an University of Posts & Telecommunications, China, 710121, Westone Cryptologic Research Center, China, 100070; email: zhengdong@xupt.edu.cn.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Security and privacy** → **Cryptography**; **Access control**; **Database and storage security**;

Additional Key Words and Phrases: Attribute-based encryption, cloud computing, access control, survey

ACM Reference format:
Yinghui Zhang, Robert H. Deng, Shengmin Xu, Jianfei Sun, Qi Li, and Dong Zheng. 2020. Attribute-based Encryption for Cloud Computing Access Control: A Survey. *ACM Comput. Surv.* 53, 4, Article 83 (August 2020), 41 pages.
https://doi.org/10.1145/3398036

1 INTRODUCTION

As an attractive computing paradigm, cloud computing enables the on-demand provision of diverse resources exemplified by computation power and storage over the Internet, which liberates enterprises from the maintenance of IT infrastructures and the management of data centers to concentrate on their core businesses. In addition, cloud computing enables individuals to enjoy a variety of powerful resources in a pay-as-you-consume fashion. Usually, cloud computing comes into three service models in practice, including the public, the private, and the hybrid clouds. An increasing number of organizations and individuals are choosing the public cloud because of greater flexibility, operation cost savings, and better customer support. It is forecast by Gartner [20] that the public cloud service market worldwide will grow from $182.4B in the year 2018 to $331.2B in the year 2022. Nevertheless, security concerns and privacy issues have become the dominating hindrance on the road to the wider adoption of cloud computing. Indeed, the trusting domain of third-party cloud storage servers is usually different from that of users, and hence users are reluctant to subscribe to cloud service to outsource their important data. According to the top five cloud computing predictions for 2020 by Techfunnel [86], security comes in first as the cloud's biggest challenge.

In cloud computing, as depicted in Figure 1, to realize that only authorized users have the right to access the data, either the symmetric encryption technology or the traditional public-key encryption technology can be used. However, when a new data user (DU) enters into the symmetric encryption-based access control system, the data owner (DO) has to share with the new DU a secret key that acts as a shared key and is applied to encrypt the DO's data again. Similarly, in the traditional public-key encryption-based access control, the DO is required to encrypt his data again via the new DU's public key PK_3, which is different from the public keys PK_1 and PK_2 of the original DUs. Obviously, these two access control mechanisms lack flexibility and scalability, because either shared secret keys or public keys are required for the DO to outsource his data to the cloud. Flexibility and scalability refer to the expressiveness of access control policies and the impact of newly joined data users on the access control system, respectively. Fortunately, the attribute-based encryption (ABE) technology plays a key role in realizing access control systems with fine granularity and scalability. As shown in the ABE-enabled access control mechanism in Figure 1, the flexible attributes are embedded into the ciphertext and the DO does not need to know the identities of specific DUs before encryption. When a new DU joins the system, DOs are not affected and do not have to do anything. Therefore, both flexibility and scalability are enabled in the ABE-enabled access control system.

Sahai and Waters [81] introduced the ABE notion for the first time. As a promising cryptographic primitive, ABE has successfully attracted considerable research efforts, and it comes in two categories. The first is ciphertext-policy ABE, which is often abbreviated as CP-ABE; and the second category is key-policy ABE, which is often abbreviated as KP-ABE.

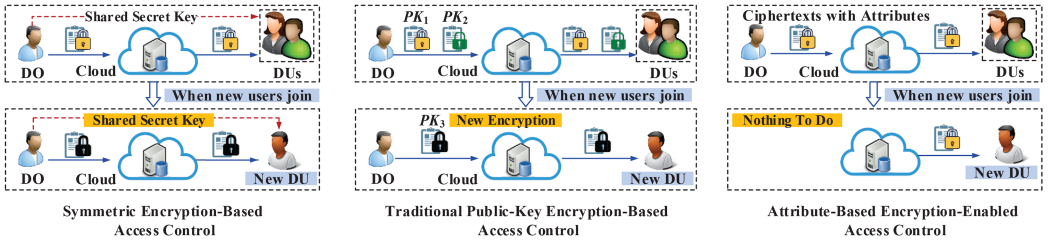


Fig. 1. Different access control mechanisms in cloud computing.

In CP-ABE, a user's attribute secret key is associated with an attribute list, and a ciphertext specifies an access policy that is defined over an attribute universe of the system. A ciphertext can be decrypted by a user if and only if the user's attribute list matches the ciphertext's access policy. Let us consider an application scenario of a teaching information system in a university. The attribute universe of the system is {IDENTITY, DEPARTMENT, YEAR} where IDENTITY={*Teacher*, *Student*}, DEPARTMENT={*CS: Computer Science*, *CE: Communication Engineering*, etc.}, and YEAR={*Year 1*, *Year 2*, *Year 3*, *Year 4*}. In this system, suppose there are some confidential records of students' grades for the computer system course encrypted under the access policy ((IDENTITY: *Student* AND DEPARTMENT: *CS*) OR IDENTITY: *Teacher*). Then, the plaintext can only be recovered by students in the *CS* department or by teachers. For example, the user Alice with the attribute list {IDENTITY: *Student*, DEPARTMENT: *CE*, YEAR: *Year 1*} cannot decrypt the ciphertext, while the user Tom with the attribute list {IDENTITY: *Teacher*} is capable of decrypting the ciphertext.

In KP-ABE, an access policy, which is defined over the system's attribute universe, is encoded into a user's attribute secret key and a ciphertext is created with respect to an attribute list. A ciphertext can be decrypted by a user if and only if the corresponding attribute list matches the access policy associated with the user's attribute secret key. For instance, if the access policy ((IDENTITY: *Student* AND DEPARTMENT: *CS*) OR IDENTITY: *Teacher*) is encoded into the user Bob's attribute secret key, then Bob fails to decrypt a ciphertext, which is computed based on the attribute list {IDENTITY: *Student*, DEPARTMENT: *CE*, YEAR: *Year 3*}. Whereas, Bob would be able to decrypt a ciphertext with respect to {IDENTITY: *Teacher*}.

Because of the high expressiveness and scalability of ABE-enabled access control on outsourced cloud data, many researchers have put their efforts into ABE. However, CP-ABE has gained much more attention than KP-ABE. The reason lies in that the access policy determination in CP-ABE is put on the data owner's hand. According to the requirements of access control in different application scenarios, CP-ABE schemes are further divided into many categories. In fact, the design of existing ABE schemes has become increasingly sophisticated to achieve various functional features. Furthermore, researchers usually assert the advantages of their ABE schemes while ignoring the drawbacks. Due to the lack of clear and comprehensive assessment criteria of ABE, existing ABE schemes cannot be evaluated and compared in terms of security and performance in a fair manner.

Although there have been surveys on ABE studies [2, 32, 84], the survey in Reference [2] focuses on revocation in CP-ABE, which is one of the categories reviewed in this article, and the ABE surveys in References [32, 84] are restricted to narrow scopes and lack comprehensive assessment criteria.

To help the readers systemically perceive the core design of distinct ABE schemes and facilitate ABE's applications to cloud computing access control, this article presents a solid and comprehensive survey of ABE in the setting of cloud computing access control. Our contributions are summarized as follows:

- First, a clear taxonomy and comprehensive assessment criteria of ABE are proposed. According to the taxonomy, ABE is categorized into KP-ABE, CP-ABE, anti-quantum ABE, and generic ABE constructions, in which CP-ABE has received the most attention of researchers.
- Second, in accordance with cryptographically functional features, CP-ABE schemes are further classified into basic CP-ABE constructions, revocable CP-ABE constructions, accountable CP-ABE constructions, policy-hiding CP-ABE constructions, CP-ABE constructions with policy updating, multi-authority CP-ABE constructions, hierarchical CP-ABE constructions, online/offline CP-ABE constructions, and outsourced CP-ABE constructions.
- Third, a systematical methodology for discussing and comparing existing ABE schemes is proposed. To be specific, for KP-ABE and each type of CP-ABE, the ABE-based access control application scenario is first presented and explained by typical examples. Then, the syntax of the specific type of ABE is given, which is followed by the adversarial model and security goals. Furthermore, the state-of-the-art for this type of ABE is reviewed by analyzing the design strategies and special features in detail. Finally, a comprehensive comparison is made in terms of the proposed assessment criteria with respect to security and performance.
- Last, a number of open research challenges are highlighted in the light of our observations on the state-of-the-art of ABE.

This survey article aims to help the non-specialists comprehend ABE systematically and benefit the researchers to keep up with the state-of-the-art technologies of ABE. Compared to the existing ABE surveys, our article not only provides a broader 12 categories of ABE schemes, but also makes a more comprehensive comparison based on the design strategies and special features. In addition, we point out the ABE solutions for each proposed cloud computing access control scenario and identify open challenges for future ABE research. This survey is expected to help practitioners to determine suitable ABE schemes and inspire researchers to explore customized ABE designs for particular cloud computing access control scenarios.

The rest of this article is organized as follows: Section 2 presents the taxonomy and assessment criteria of ABE. Following the taxonomy, different types of CP-ABE solutions are sequentially reviewed in the subsequent nine sections (Section 3 to Section 11). Concluding remarks are made in Section 12. Due to space limitation, more details of hierarchical CP-ABE, online/offline CP-ABE, and outsourced CP-ABE are given in the Supplemental Material A, the Supplemental Material B, and the Supplemental Material C, respectively. Additionally, we review KP-ABE in the Supplemental Material D. Finally, both anti-quantum ABE and generic ABE constructions are reviewed in the Supplemental Material E.

2 TAXONOMY AND ASSESSMENT CRITERIA OF ABE

2.1 Taxonomy of ABE

The proposed taxonomy of ABE is illustrated in Figure 2. During the rest of the article, we will mainly review basic CP-ABE, enhanced CP-ABE, and KP-ABE. Different from basic CP-ABE, enhanced CP-ABE further realizes other cryptographically functional features. These features can be relevant to attribute secret keys, access policies, attribute authorities, and computation efficiency. According to the features, enhanced CP-ABE schemes are divided into the following eight categories:

- *Revocable CP-ABE.* The functionality of revocation is realized in revocable CP-ABE. According to the graininess, revocation mechanisms fall into user revocation and attribute revocation. However, in the light of the effect to non-revoked users, revocation mechanisms are divided into indirect revocation and direct revocation.

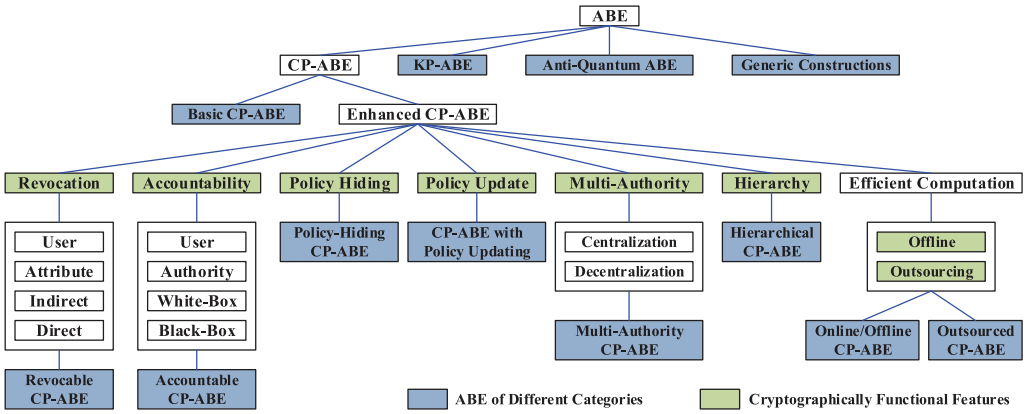


Fig. 2. Taxonomy of ABE.

- *Accountable CP-ABE.* The functionality of accountability is realized in accountable CP-ABE. Both the user traceability and the attribute authority accountability are involved in accountable CP-ABE. According to the given conditions, accountability mechanisms fall into white-box accountability and black-box accountability.
- *Policy-hiding CP-ABE.* Access policy privacy protection is further enabled in policy-hiding CP-ABE schemes.
- *CP-ABE with Policy Updating.* In basic CP-ABE, it is impossible to change a ciphertext's access policy. Considering access control in emergencies, CP-ABE with policy updating can be adopted to update the access policy in an involved ciphertext.
- *Multi-authority CP-ABE.* With this type of CP-ABE construction, distributed access privilege can be realized. According to whether a central authority exists or not, multi-authority CP-ABE schemes are divided into centralized multi-authority CP-ABE constructions and decentralized multi-authority CP-ABE constructions.
- *Hierarchical CP-ABE.* As for hierarchical CP-ABE constructions, the delegation of access privilege is organized in a hierarchical manner.
- *Online/offline CP-ABE.* To alleviate the computation burden of data owners and the attribute authority, online/offline CP-ABE is designed, which can realize offline encryption or offline key generation.
- *Outsourced CP-ABE.* To support data users (respectively, data owers and the authority) with constrained computation resources, outsourced CP-ABE is proposed to outsource laborious computation in decryption (respectively, encryption and key generation) to third-party servers.

2.2 Assessment Criteria of ABE

For a systematic comparison of existing ABE schemes, we present the assessment criteria of ABE with respect to security and performance. It is noted that the assessment criteria are proposed for fairly evaluating the properties claimed in different ABE schemes.

2.2.1 Security Assessment Criteria. As we know, different types of ABE schemes have distinct security goals, which will be explained in the corresponding section later. The fundamental security goals of ABE include data confidentiality and collusion resistance. Furthermore, a particular type of ABE scheme usually has its own special security properties. For instance, backward secrecy and forward secrecy are usually considered in revocable ABE, and accountability should be

enabled in accountable ABE. In addition, for different types of ABE schemes, data confidentiality and collusion resistance are usually realized under diverse conditions, which are reflected in the *Type of Adversaries*, the *Security Model,* and *Complexity Assumptions* involved in the security analysis. A multitude of ABE schemes may realize the same security goals under diverse conditions such as different security models. Whereas, researchers usually assert the security advantages of their ABE schemes while ignoring the drawbacks such as a weak security model.

For the fairness of security comparisons, this survey article proposes the security assessment criteria of ABE, including *Types of Adversaries*, *Security Model,* and *Complexity Assumption*, whereas giving security properties in the subsection named *Adversarial Model and Security Goals* of the concrete type of ABE.

- *Types of Adversaries.* Selective adversaries and adaptive adversaries are two typical types of adversaries considered in ABE. The adaptive adversary does not need to specify its target access policy or attribute list in advance and hence is stronger than the selective adversary. If an ABE scheme is proven secure against selective (respectively, adaptive) adversaries, it is said to achieve selective (respectively, full) security. Because the target access policy or attribute list is not specified in advance, full security proofs are technically more challenging than selective security proofs, because challengers cannot set the parameters in a targeted manner in the process of simulation.
- *Security Model.* For one thing, if generic groups are involved in an ABE scheme's security proofs, it is considered that the scheme has security in generic group models. For another, according to whether random oracles are used in the security analysis, the security models are categorized into the standard model (STM) and the random oracle model (ROM). In the concrete security analysis of an ABE scheme, non-generic groups are better than generic groups, and standard models are preferable to random oracle models. In fact, the security proofs in non-generic groups and STMs are technically more challenging, because the challengers have limited ability in the process of simulation compared with the case of generic groups and ROMs. It is noted that the generic group model and the ROM are not comparable, which is explained as follows:
  - —First, the generic group model means that no property other than equality of group elements can be directly tested by adversaries. In other words, the adversaries cannot perform most of the group operations themselves and have to rely on queries to obtain the operation results. Please refer to References [7, 83] for more details.
  - —Second, the ROM means that random oracles are involved in the model. A random oracle is a black box that responds to each query by giving a random value chosen uniformly from its output domain. If a query is repeated, it returns the same value as before. Please refer to Reference [4] for more details.
  - —Last, there are indeed ABE schemes [5, 88, 100] that are proven secure with the generic group model and the ROM involved simultaneously.
- *Complexity Assumption.* An ABE scheme's security is usually reduced to the adopted complexity assumptions. It is more desirable to prove the security under the recognized assumptions, of which the form is concise and the complexity is proved. The security proofs under complexity assumptions of concise forms are technically challenging, because fewer parameters are provided by the assumption instance and used by the challengers.

2.2.2 Performance Assessment Criteria. In the following, we present the performance assessment criteria. Note that the typical communication and storage costs can be adequately evaluated by the criteria.

- *System Public Key Size.* This parameter is relevant to the storage overhead. The system public key size of many ABE solutions is linearly proportional to some parameters exemplified by the attribute number in the access control system. It would be better if the system public key is kept constant-size in ABE.
- *Ciphertext Size.* The ciphertext size is related to the communication and storage costs in ABE-enabled access control systems. In many ABE solutions, the size of ciphertexts usually linearly expands with the associated access policy complexity. It is desirable if an ABE scheme has a ciphertext of constant size.
- *Attribute Secret Key Size.* The attribute secret key size is related to the storage cost in ABE-enabled access control systems. In many ABE solutions, the size of attribute secret keys is linearly proportional to the number of involved attributes. It is desirable if an ABE scheme has a constant-size attribute secret key.
- *Computation Cost.* Both the encryption cost and the decryption cost should be taken into account, because users may be resource-limited in cloud computing. In addition, different kinds of CP-ABE schemes have their own exclusive computation cost. For revocable CP-ABE, the server-side computation cost may exist. The accountability computation cost should be considered in accountable CP-ABE. In policy-hiding CP-ABE, the decryption computation overhead comprises the matching process cost and the decryption process cost. In CP-ABE with policy updating, the update server-side computation cost is involved. For online/offline CP-ABE solutions, both the key generation cost and the encryption cost take the forms of the cost in the offline process and the cost in the online process. Furthermore, the user-side decryption cost should also be taken into consideration. In outsourced CP-ABE, besides the attribute authority's key generation cost and the data user's decryption cost, the computation cost outsourced to the key generation service provider should also be considered besides the cost of the decryption service provider. It is worth noting that the bilinear pairing (pair) is the most expensive cryptographic operation. Besides bilinear pairings, the exponentiation (exp) and the point multiplication (pm) usually should be considered regarding the computation cost. Compared with pair, exp, and pm, the basic arithmetic operations such as the multiplication and addition are usually ignored in the analysis on computation cost [27, 74, 76]. Therefore, as for the computation cost, we give a detailed analysis in terms of the expensive cryptographic operations including pair, exp, and pm.
- *Expressiveness.* The expressiveness of ABE is reflected in the access policy, which represents the graininess of the access control system. In existing ABE schemes, there are many kinds of access policies, such as the linear secret sharing scheme (LSSS) policy, tree policies, and threshold policies. The LSSS-based schemes are more efficient compared with the schemes based on the other policies while maintaining equivalent expressiveness.
- *Group.* The groups involved in ABE are divided into prime-order groups and composite-order groups according to the group order. It is noted that the prime-order ABE construction is more desirable than the composite-order ABE construction from the viewpoint of efficiencies. However, if full security is required, the design of prime-order ABE is technically more challenging than that of composite-order ABE, because the methodology for full security proofs usually relies on composite-order groups.
- *Attribute Universe.* The attribute universe in ABE is categorized into large universe, semi-large universe, and small universe. In large universe ABE, the system public key size is not affected by the attribute universe size, and there is no limitation on the number of attributes with regard to a ciphertext. The semi-large universe means that the system public key size is not affected by the attribute universe size, while the number of attributes applied to describe a ciphertext is upper bounded. The small universe means that the system public key size is

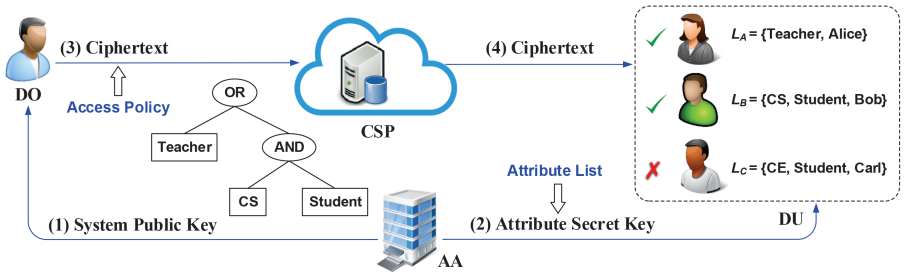


Fig. 3. The application scenario of basic CP-ABE.

related to the total attribute number in the system. Specifically, large universe ABE is most desirable, because it improves the flexibility of the corresponding access control system in the system initialization.

3 BASIC CP-ABE

3.1 Application Scenario: Basic Fine-grained Access Control

As depicted in Figure 3, fine-grained access control mechanisms can be achieved with in virtue of basic CP-ABE. The access control system involves four parties, including the cloud service provider (CSP), the attribute authority (AA), the DO, and the DU. As a fully trusted entity, the AA publishes the system public keys (procedure (1)) and issues attribute secret keys to each DU based on his/her corresponding attribute list (procedure (2)). The DO chooses an access policy $\mathbb{A}$ himself and integrates $\mathbb{A}$ into the ciphertext (procedure (3)). Then, the DU has the means for decrypting the ciphertext to retrieve the original data based on his/her attribute secret keys (procedure (4)). A successful decryption relies on the matching between the embedded attribute list and the attached access policy. For example, as shown in Figure 3, we denote the school of computer science by CS and the school of communication engineering by CE. If $\mathbb{A}$=(Teacher OR (CS AND Student)), then the ciphertext can be successfully decrypted by DUs with attribute lists L_A = {Teacher, Alice} and L_B = {CS, Student, Bob}, respectively, while it cannot be decrypted by the DU with attribute list L_C = {CE, Student, Carl}.

3.2 Syntax of Basic CP-ABE

In a basic CP-ABE scheme, there are four algorithms acting as main ingredients:

- $\text{Setup}(1^\lambda) \rightarrow (PK, MK)$: This algorithm is called the system setup algorithm and it will be run by the AA at the beginning. After inputting a security parameter λ, the algorithm returns system public keys PK and master keys MK.
- $\text{KeyGen}(PK, MK, L) \rightarrow SK_L$: It is called the attribute key generation algorithm, which will be performed by the AA. The AA takes PK, MK, and L as inputs, where L is an attribute list, and returns SK_L as the attribute secret key corresponding to L.
- $\text{Encrypt}(PK, M, \mathbb{A}) \rightarrow CT_{\mathbb{A}}$: This algorithm is called the encryption algorithm, which will be performed by the DO. The DO first chooses an access policy $\mathbb{A}$ for the target message M, and then takes as inputs PK, M, and $\mathbb{A}$. The algorithm outputs a ciphertext $CT_{\mathbb{A}}$ of M associated with $\mathbb{A}$, which will be stored on the CSP.
- $\text{Decrypt}(PK, CT_{\mathbb{A}}, SK_L) \rightarrow M$ or $\perp$: This algorithm is called the decryption algorithm, which will be run by the DU. After inputting PK, a ciphertext $CT_{\mathbb{A}}$ of M with $\mathbb{A}$, and an attribute secret key SK_L corresponding to L, the algorithm returns M if L matches $\mathbb{A}$ (denoted by $L \models \mathbb{A}$), and otherwise outputs the error symbol $\perp$ to indicate a failure of decryption.

3.3 Adversarial Model and Security Goals

In basic CP-ABE enabled access control systems, the AA is a fully trusted entity. As an honest-but-curious party, the CSP honestly executes the procedures of the system but tries to obtain secret information from ciphertexts on the cloud as much as possible.

- *Data Confidentiality.* In secure access control systems, if a DU is an unauthorized user, it should be blocked from decrypting the ciphertext, because his/her attributes fail to match the access policy. Additionally, the CSP should not be allowed to decrypt ciphertexts without authorization. For example, suppose $\mathbb{A}$=((CE AND Teacher) OR (CS AND Student)), then the ciphertext can be decrypted by the DU with L_A = {CE, Teacher, Alice} but not by the DU with L_B = {CE, Student, Bob}.
- *Collusion Resistance.* The access control system should ensure data confidentiality against collusion attacks from unauthorized DUs and the CSP. Specifically, even if many malicious DUs and the CSP collude to decrypt the ciphertext by integrating attribute secret keys, they cannot succeed if none of them can individually succeed in decryption. For example, suppose $\mathbb{A}$=((CE AND Teacher) OR (CS AND Student)), then the ciphertext can be decrypted neither by the DU with L_A = {CS, Teacher, Alice} nor by the DU with L_B = {CE, Student, Bob}. In particular, even if the DUs with L_A and L_B combine their attribute secret keys together, the ciphertext under $\mathbb{A}$ still cannot be successfully decrypted by them.

3.4 Research Status of Basic CP-ABE

Sahai and Waters [81] applied the secret-sharing technique to design two fuzzy identity-based encryption schemes. The schemes support threshold access policies and are proved selectively secure. As a small universe scheme, the security proof of the first one relies on the decisional modified bilinear Diffie-Hellman (DMBDH) assumption. The other one is a semi-large universe solution involving the decisional bilinear Diffie-Hellman (DBDH) assumption.

For the first time, Bethencourt et al. [5] developed a CP-ABE solution that allows tree-based access policies. In this scheme, a novel attribute secret key randomization technique is adopted to realize collusion-resistance, in which a two-level random masking methodology acts as the key ingredient. However, the scheme only obtains security in the sense of generic group models.

Cheung and Newport [12] designed a basic CP-ABE solution in which AND gate policies with two attributes, i.e., $\mathbf{AND}^{*}_{+,-}$, are allowed. Under the DBDH assumption, the chosen plaintext attack (CPA) security of the basic solution is given in standard models. Furthermore, they combine the basic scheme and one-time signatures to obtain an enhanced scheme, which can resist the chosen-ciphertext attack (CCA). They also adopt the idea of hierarchical attributes to raise the performance of the basic scheme. However, the limitation of the AND gate access policy is the lack of expressiveness.

To tackle the issues of expressiveness and security proof together, Goyal et al. [21] designed a tree-based CP-ABE solution. The scheme suffers from severe efficiency drawbacks due to the copies of each attribute for every position in the access tree. For reducing the computation burden during the processes of encryption and decryption, Liang et al. [51] designed a new bounded CP-ABE solution of which security proofs are given in standard models under the DBDH assumption. However, the computation efficiency still needs to be improved and the attribute universe is small. Lewko et al. [37] developed a CP-ABE construction that achieves full security under three new static assumptions. The composite group is involved in the scheme and the efficiency needs to be improved.

In the above schemes, the access policy complexity has negative influence in the performance because of the linear increasing of the ciphertext size. Herranz et al. [24] designed a CP-ABE solution with ciphertexts of constant length. The scheme's CPA security is obtained in standard

models under a static assumption. According to the computation cost, the threshold-based CP-ABE solution due to Susilo et al. [85] is more efficient than Reference [24] because of the constant length of ciphertexts. However, the expressiveness of these schemes remains to be improved to some extent. Waters [93] designed a CP-ABE solution in which LSSS can be utilized to describe access policies. The security analysis of this scheme involves the q-type assumption, where q is related to the access policy complexity. Lewko and Waters [39] presented a new methodology for proving full security based on the technique for selective security. They further designed a CP-ABE scheme with full security based on Reference [93], in which the group order is composite and involves three primes.

The attributes in the above schemes need to be enumerated in the system setup phase. To remove this restriction, Rouselakis and Waters [78] designed a CP-ABE solution with a large universe and LSSS access policies. The scheme follows the partitioning methodology and is proven secure against selective adversaries under q-type assumptions. To further improve the performance of CP-ABE, a CP-ABE solution is developed by Zhang et al. [115], in which the ciphertext size and the computation overhead are constant. Based on Reference [115], a data sharing mechanism with fine granularity is enabled in mobile clouds [116], which supports the access policy $\mathbf{AND}_m^*$ by allowing wildcards and multiple values of attributes. Malluhi et al. [65] developed an LSSS-based CP-ABE construction with efficient decryption, and the scheme is proven secure against selective adversaries under a q-type assumption. In ROM, Agrawal and Chase [1] designed a CP-ABE solution and gave its full security under the decisional linear (DLIN) assumption. Xue et al. [97] designed a tree-based CP-ABE mechanism that supports attribute comparison and the scheme's security proofs involve the DBDH assumption. Okamoto and Takashima [72] proposed a functional encryption solution with full security in which CP-ABE acts as a special instance. Even if the scheme is designed in prime order groups, it suffers from the drawbacks of complex design and low performance, because dual pairing vector spaces are involved.

3.5 Comparison of Basic CP-ABE Schemes

A comparison is made in Table 1 to analyze basic CP-ABE constructions with regard to the size of the system public key $|PK|$, the size of the ciphertext $|CT|$, the size of the attribute secret key $|SK|$, the computation efficiencies of encryption and decryption, the policy, the type of group, the attribute universe, the security model and the underlying complexity assumption. Table 1 demonstrates that the scheme of Reference [115] is most efficient with regard to the computation overhead and hence is proper for mobile users, while the access policy is less expressive. The policies of these schemes can be AND-gate policies [12, 115], threshold policies [24, 81, 85], tree polices [5, 21, 51, 97], and LSSS policies [1, 37, 39, 65, 78, 93]. In expressive LSSS-based schemes, the scheme of Reference [65] is computationally more efficient than the others. However, the scheme involves a non-static assumption in the security analysis. Full security is obtained only by the schemes of References [1, 37, 39] in which the schemes of References [37, 39] are proven secure in standard models. Although proven secure in the ROM, the scheme of Reference [1] simultaneously supports the LSSS access policy, large universe, static assumption, and the number of pairings involved in decryption is constant. Therefore, if full security, expressiveness, and decryption efficiency are required simultaneously, the scheme of Reference [1] is a preferred choice.

4 REVOCABLE CP-ABE

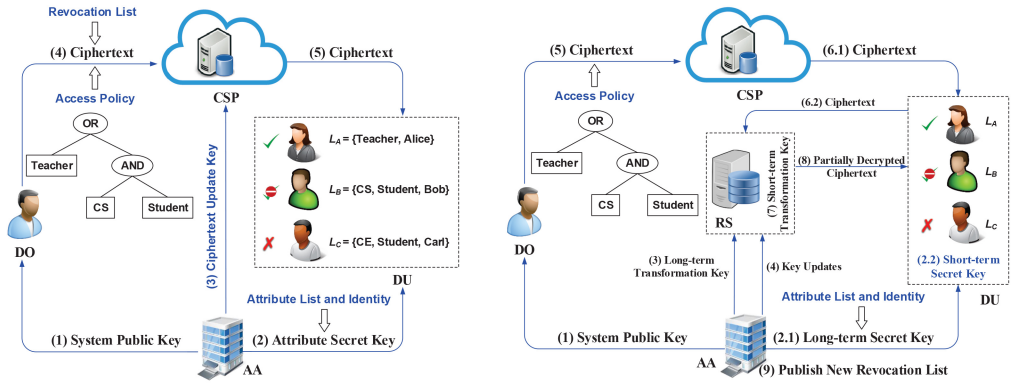
4.1 Application Scenario: Revocable Fine-grained Access Control

Revocable CP-ABE is utilized to enable revocable access control with fine granularity, which is demonstrated in Figure 4. Similar to the case of basic CP-ABE, the revocable access control system involves the AA, the CSP, the DO, and the DU. It is noted that revocation mechanisms in

Table 1. The Comparison of Basic CP-ABE Constructions

Schemes	Parameter Size			Computation Cost		Policy	Group	Universe	Security	Assumption
	$\|PK\|$	$\|CT\|$	$\|SK\|$	Encryption	Decryption					
SW05-1 [81]	$n\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$\ell\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$\ell\|\mathbb{G}\|$	$(\ell+1)$**exp**	I**pair**+I**exp**	Threshold	Prime	Small	S-STM	DMBDH
SW05-2 [81]	$(n+3)\|\mathbb{G}\|$	$(\ell+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\ell\|\mathbb{G}\|$	$(\ell+2)$**exp**	$2I$**pair**+I**exp**	Threshold	Prime	semi-Large	S-STM	DBDH
BSW [5]	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2I+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+1)\|\mathbb{G}\|$	$(2I+2)$**exp**	$(2I+1)$**pair**+I**exp**	Tree	Prime	Large	G-ROM	-
CN [12]	$(3n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n+1)\|\mathbb{G}\|$	$(n+2)$**exp**	$(n+1)$**pair**	$\mathbf{AND}^*_{+,-}$	Prime	Small	S-STM	DBDH
GJPS [21]	$(n\mu+m\nu)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$I(\mu+\nu)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n\mu+m\nu)\|\mathbb{G}\|$	$(I(\mu+\nu)+1)$**exp**	$2I$**pair**+I**exp**	Tree	Prime	Small	S-STM	DBDH
LCLX [51]	$(n\mu'+m\nu')\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$I(\mu'+\nu')\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell\mu'+m\nu')\|\mathbb{G}\|$	$(I(\mu'+\nu')+1)$**exp**	$2I$**pair**+I**exp**	Tree	Prime	Small	S-STM	DBDH
LOS+ [37]	$(n+2)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(2n_1+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$3\ell\|\mathbb{G}_{1,3}\|$	$(3n_1+2)$**exp**	$(2I+1)$**pair**+I**exp**	LSSS	Composite	Small	F-STM	New
HLR [24]	$(2n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|+(n-1)\|\mathbb{Z}_p^*\|$	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+\ell)\|\mathbb{G}\|$	$(n+\ell+1)$**exp**	3**pair**+ℓ^2**exp**	Threshold	Prime	Small	S-STM	aMSE-DDH
SYGH [85]	$(2m_1+3)\|\mathbb{G}\|+2\|\mathbb{Z}_p^*\|$	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(m_1+\ell)\|\mathbb{G}\|$	1**pair**+$(\ell+3)$**exp**	2**pair**+$(I^2+\ell)$**exp**	Threshold	Prime	semi-Large	S-STM	aMSE-DDH
Waters [93]	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+2)\|\mathbb{G}\|$	$(3I+2)$**exp**	$(2I+1)$**pair**+I**exp**	LSSS	Prime	Small	S-STM	q-type
LW12 [39]	$(n+3)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(2n_1+2)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$4\ell\|\mathbb{G}_{1,3}\|$	$(3I+3)$**exp**	$(2I+2)$**pair**+I**exp**	LSSS	Composite	Small	F-STM	q-type
RW [78]	$5\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+2)\|\mathbb{G}\|$	$(5I+2)$**exp**	$(3I+1)$**pair**+I**exp**	LSSS	Prime	Large	S-STM	q-type
ZZC [115]	$(N+1)\|\mathbb{G}\|+N\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$n\|\mathbb{G}\|$	3**exp**	2**pair**	$\mathbf{AND}^*_m$	Prime	Small	S-ROM	q-type
MST [65]	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+2)\|\mathbb{G}\|$	$(2I+2)$**exp**	2**pair**+$2I$**exp**	LSSS	Prime	Small	S-STM	q-type
AC [1]	$3\|\mathbb{G}_2\|+2\|\mathbb{G}_T\|$	$3n_1\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|+\|\mathbb{G}_T\|$	$(3\ell+1)\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|$	$(6n_1+3)$**exp**	6**pair**+$6I$**exp**	LSSS	Prime	Large	F-ROM	DLIN
XHX [97]	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2I+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+1)\|\mathbb{G}\|$	$(2I+2)$**exp**	$(2I+1)$**pair**+I**exp**	Tree	Prime	Large	S-ROM	DBDH

† $|\mathbb{G}|$: the bit length of an element in the group $\mathbb{G}$; n: the size of the attribute universe; n_1: the access policy matrix's row number; N: the system has N attribute values; m: the number of dummy attributes; m_1: the upper bound of attribute number in encryption; I: the access policy's complexity; μ (respectively, ν): the number of copies of each attribute (respectively, dummy attribute); μ' and ν': two numbers defined in advance; ℓ: the number of the user's attributes; semi-Large: the number of attributes used in encryption has an upper bound.



(a) The Case of Direct Revocation (b) The Case of Indirect Revocation

Fig. 4. The application scenario of revocable CP-ABE.

CP-ABE include direct revocation and indirect revocation. The difference lies in whether the non-revoked DU's secret key should be updated when a revocation event[1] occurs. In direct revocation, non-revoked DUs will not be affected by revocation events, and an entity revocation server (RS) is additionally added that is untrustworthy and used to enable the revocation mechanism. In revocable CP-ABE, the DU can successfully retrieve the original data by decrypting the ciphertext through his/her secret key if the access policy is met by the attribute list and the DU is (respectively, required attributes are) not revoked in the setting of user (respectively, attribute) revocation. Because concrete revocation mechanisms usually have different technical details, for ease of understanding, we focus on typical revocation solutions.

In the case of direct revocation as depicted in Figure 4(a), after publishing the system public keys (procedure (1)), the attribute list and identity of the DU is integrated into the attribute secret key by the AA (procedure (2)). In particular, the CSP obtains a ciphertext update key to update previous ciphertexts (procedure (3)) and the revocation list is further attached by the DO to the ciphertext (procedure (4)) [3, 28, 109]. The DU obtains the ciphertext from the CSP and it can finally decrypt the ciphertext only if the access policy is matched by his/her attribute list and necessary attributes are not revoked (procedure (5)). For instance, as shown in Figure 4(a), $\mathbb{A}$=(Teacher OR (CS AND Student)). Even if L_B matches $\mathbb{A}$, the ciphertext cannot be decrypted by the DU with L_B = {CS, Student, Bob} when the attribute Student is revoked.

In the case of indirect revocation as depicted in Figure 4(b), after publishing the system public keys (procedure (1)), the attribute list and identity of the DU is integrated into the long-term secret key by the AA (procedure (2.1)). Based on the long-term secret key, the DU can further generate the short-term secret key (procedure (2.2)). In addition, the RS obtains the long-term transformation key (procedure (3)) and key update information (procedure (4)) from the AA [76]. The DO generates the ciphertext and uploads it to the CSP (procedure (5)). The DU obtains the ciphertext from the CSP and sends it to the RS (procedures (6.1) and (6.2)). Based on the long-term transformation key and key update information, the RS generates a short-term transformation key (procedure (7)) that is utilized to transform a ciphertext to a partially decrypted ciphertex (procedure (8)). Based on the short-term secret key, the DU can decrypt the partially decrypted ciphertext only if the access policy is matched by his/her attribute list and necessary attributes are not revoked. A new revocation list will be published by the AA once revocation events occur (procedure (9)).

[1]A revocation event means revoking a user or some attributes of a user.

4.2 Syntax of Revocable CP-ABE

A directly revocable CP-ABE scheme comprises algorithms including Setup, KeyGen, Encrypt, UKeyGen, CTUpdate, and Decrypt.[2] For algorithms Setup and KeyGen, the specifications are the same as those of basic CP-ABE, we only give other different algorithms as below.

- Encrypt$(PK, M, \mathbb{A}, \mathcal{R}) \rightarrow CT_{\mathbb{A}}$: The DO runs the encryption algorithm. After inputting PK, M, $\mathbb{A}$, which is chosen by the DO, and the attribute revocation information $\mathcal{R}$, which specifies the identities and attributes involved in revocation events, it outputs a ciphertext $CT_{\mathbb{A}}$ of M under $\mathbb{A}$, which will be stored on the CSP.
- UKeyGen$(PK, MK, \mathcal{R}^{(k)}) \rightarrow (PP^{(k)}, UK^{(k)})$: This algorithm is called the update key generation algorithm, which will be performed by the AA. After inputting PK, MK, and revocation information $\mathcal{R}^{(k)}$, which is published by the AA for the kth revocation event, this algorithm outputs the public parameter $PP^{(k)}$ and the corresponding ciphertext update key $UK^{(k)}$ associated with $\mathcal{R}^{(k)}$. Then, the AA makes $PP^{(k)}$ public and securely sends $UK^{(k)}$ to the CSP.
- CTUpdate$(PK, CT_{\mathbb{A}}, UK^{(k)}, \mathcal{R}^{(k)}) \rightarrow CT'_{\mathbb{A}}$: This algorithm is called the ciphertext update algorithm, and it will be run by the CSP. The inputs include PK, $CT_{\mathbb{A}}$, $UK^{(k)}$, and $\mathcal{R}^{(k)}$, and the output is an updated ciphertext $CT'_{\mathbb{A}}$ associated with $CT_{\mathbb{A}}$.
- Decrypt$(PK, PP, CT_{\mathbb{A}}, SK_L) \rightarrow M$ or $\perp$: The DU runs the decryption algorithm. On input PK, the public parameter PP, which is introduced by revocation events so far, $CT_{\mathbb{A}}$ of M under $\mathbb{A}$, and a secret key SK_L with regard to L, it returns M if $L \models \mathbb{A}$ and the DU's corresponding attributes are not revoked, and outputs the error symbol $\perp$ otherwise.

For indirectly revocable CP-ABE [76], there are nine algorithms: Setup, UserKG, TranKU, TranKG, DecKG, Encrypt, Transform, Decrypt, and Revoke. Because the definitions of Setup, DecKG, Encrypt, and Decrypt are similar to those of Setup, KeyGen, Encrypt, and Decrypt of basic CP-ABE, we only give other different algorithms in the following:

- UserKG$(PK, MK, ID, L, st) \rightarrow (PK_{ID}, SK_{ID})$: It is called the user key generation algorithm, which will be performed by the AA. After inputting PK, MK, the user's identity ID, the user's attribute list L, and the state st, the algorithm returns the long-term transformation key PK_{ID} and long-term secret key SK_{ID}, which are, respectively, sent to the RS via public channels and the user through a secure channel.
- TranKU$(PK, MK, t, rl, st) \rightarrow (KU_t, st')$: This algorithm is called the transformation key update algorithm, and it will be run by the AA. After inputting PK, MK, t, which is a time period, rl, which is used to denote revocation lists and a state st, the algorithm returns KU_t as a key update message and st' as a new state. Finally, the AA sends KU_t to the RS.
- TranKG$(PK, ID, PK_{ID}, KU_t) \rightarrow TK_{ID,t}$: The RS will run this algorithm, which is used to generate transformation keys. It takes as inputs PK, ID, PK_{ID}, and KU_t and outputs a time-based short-term transformation key $TK_{ID,t}$ associated with ID and the time period t.
- Transform$(PK, ID, L, TK_{ID,t}, CT_{\mathbb{A}}) \rightarrow CT'_{\mathbb{A}}$ or $\perp$: The CSP runs this algorithm, which is called the ciphertext update algorithm. The inputs include PK, ID, L, $TK_{ID,t}$, and $CT_{\mathbb{A}}$, and the output is an updated ciphertext $CT'_{\mathbb{A}}$ associated with $CT_{\mathbb{A}}$ if $L \models \mathbb{A}$, and in the time period t the identity ID has not been revoked. Otherwise, it outputs the error symbol $\perp$ to indicate the failure of the transformation.

[2] Different from References [28, 109], the algorithms UKeyGen and CTUpdate are not involved in Reference [3], because the backward secrecy is not considered.

- Revoke$(PK, t, rl, st) \rightarrow rl'$: This algorithm is called the revocation algorithm, which is performed by the AA. On inputs PK, t, rl, and st, the algorithm returns rl' as a new revocation list.

4.3 Adversarial Model and Security Goals

Revocable CP-ABE is favorable for designing revocable access control systems in which the AA is fully trustworthy. The CSP and the RS are honest-but-curious. Similar to basic CP-ABE, revocable CP-ABE should also satisfy *Data Confidentiality* and *Collusion Resistance*. In addition, because revocation is taken into account, *Backward Secrecy* and *Forward Secrecy* should be realized in revocable access control systems. Backward secrecy and forward secrecy mean that a data user should be prevented from decrypting the previous and subsequent ciphertexts, respectively, if his/her attributes required in decryption are revoked.

For example, suppose $\mathbb{A}$=(Teacher OR (CS AND Student)), then the ciphertext can be decrypted by the DU with L_A = {CS, Teacher, Alice} and the DU with L_B = {CS, Student, Bob} individually. However, if the attribute CS of L_B is revoked, then Bob cannot decrypt the previous ciphertext after ciphertext updating and all subsequent ciphertexts associated with $\mathbb{A}$. However, if the attribute CS of L_A is revoked, Alice can still successfully decrypt all the ciphertexts associated with $\mathbb{A}$.

4.4 Research Status of Revocable CP-ABE

Attrapadung and Imai [3] designed two CP-ABE schemes that realize direct revocation and the LSSS policy based on the basic CP-ABE technique and broadcast encryption. In a broadcast encryption scheme [18], the data owner as a broadcaster encrypts a message for a target set of data users. Any data user in the set can decrypt the ciphertext based on his/her secret key to get the message. Even if all data users outside of the set collude, they cannot obtain the message. The idea of direct revocation is removing the identity list of involved users from the target set of broadcast encryption. Hence, only user revocation is realized in these schemes. Furthermore, the schemes only support small universe and are proven secure under q-type assumptions.

An AND-gate policy CP-ABE solution is proposed by Yu et al. [104] and the security analysis involves the DBDH assumption. The scheme realizes indirect attribute and user revocation via the combination of proxy re-encryption and CP-ABE. In a proxy re-encryption scheme [6], a semi-trusted proxy can transform or re-encrypt a ciphertext corresponding one public key to a new ciphertext associated with another public key without having access to the underlying plaintext or secret keys. Hur and Noh [28] designed a tree-based CP-ABE solution in which indirect user and attribute revocation mechanisms are realized. The key technique in the scheme is a stateless group key distribution method based on binary trees. Under the bilinear Diffie-Hellman (BDH) assumption, the scheme is said to achieve backward secrecy and forward secrecy. However, the authors only provided informal security analysis. A similar scheme can be found in Reference [27]. Yang et al. [99] designed a CP-ABE solution with the LSSS access policy, and its security proofs are given under the q-type assumption in the ROM. The scheme enables indirect attribute and user revocation based on the ciphertext update mechanism, which is performed by a third-party server.

Motivated by Reference [80], Lee et al. [36] proposed self-updatable encryption (SUE), which is a new primitive and involves time-evolution and key-revocation mechanisms. Based on SUE, the authors further constructed a revocable CP-ABE solution, which supports the LSSS access policy and realizes direct user revocation. Phuong et al. [74] developed a new CP-ABE construction that realizes direct user revocation based on broadcast encryption. The scheme has constant size ciphertexts while only AND-gate policies are allowed. Yang et al. [100] enabled direct user revocation in tree-based CP-ABE by keeping a proxy decryption key list in the server. In the scheme, each data owner has to generate master secret keys and corresponding system public parameters. Liu et al.

[54] realized direct user revocation mechanisms for LSSS-based CP-ABE schemes by embedding a time period value and a revocation list into the ciphertext.

To realize direct attribute revocation, Zhang et al. [109] first designed an auxiliary function that can be used to specify and update the revocation related ciphertexts. Then, the authors developed a revocable CP-ABE construction based on the auxiliary function. The scheme enables direct attribute and user revocation, in which the ciphertext length is small and constant. However, it only supports the AND-gate policy and selective security. Fan et al. [17] designed a tree-based CP-ABE solution that enables dynamic membership management with arbitrary states. The scheme enables direct attribute and user revocation by adaptively updating the system public key.

Cui et al. [14] designed a CP-ABE solution in with users can be revoked in an indirect manner. In the scheme, an untrusted server can help non-revoked users to transform ciphertexts. Qin et al. [76] and Xu et al. [95] further considered the decryption key exposure issue and proposed CP-ABE schemes with indirect user revocation. Note that the schemes of References [14, 76] achieve constant-size attribute secret keys by introducing a server to help data users in decryption. The CP-ABE solution due to Li et al. [44] can prevent collusion attacks from revoked users and non-revoked users and enable indirect attribute revocation. The security proofs are given under the divisible computational Diffie-Hellman (DCDH) assumption.

4.5 Comparison of Revocable CP-ABE Schemes

In Table 2, we compare revocable CP-ABE solutions in the light of the same metrics as those used in the comparison of basic CP-ABE. In addition, we add a new metric "server-side computation cost" in Table 2. Obviously, the scheme of Reference [28] has the smallest system public key and the scheme of Reference [109] has the shortest ciphertext. The schemes of References [3, 14, 36, 54, 76, 95, 99] support LSSS policies, the schemes of References [17, 28, 44, 100] support tree policies, and the schemes of References [74, 104, 109] allow AND-gate policies. The schemes of References [14, 76, 100] are very efficient considering the user-side decryption cost, because the laborious computation tasks are outsourced to the server. However, these schemes cannot realize attribute revocation. The schemes of References [28, 44, 99, 104] realize indirect attribute revocation. Based on the state-of-the-art for revocable CP-ABE, we know only the schemes of References [17, 109] enable direct attribute revocation and direct user revocation simultaneously. Note that the scheme of Reference [109] achieves selective security and Reference [17] is fully secure in standard models. Therefore, if both full security and fine-grained direct revocation mechanisms are required, the scheme of Reference [17] can be adopted.

Table 3 further compares the special features of revocable CP-ABE constructions. To be specific, the comparison is made based on *Key Technique*, *Backward Secrecy*, *Forward Secrecy*, *Re-keying Computation Cost*, and *Re-keying Storage Cost*. We know that all the schemes of References [3, 14, 17, 28, 36, 44, 54, 74, 76, 95, 99, 100, 104, 109] realize forward secrecy, and only the schemes of References [14, 76, 100] simultaneously realize backward secrecy and forward secrecy without needing re-keying. In these schemes, the key technique is proxy-assisted decryption, which needs a proxy for decryption. Generally, it remains an open problem to design attribute directly-revocable LSSS-based CP-ABE schemes with backward and forward secrecy.

5 ACCOUNTABLE CP-ABE

5.1 Application Scenario: Accountable Fine-grained Access Control

Access control mechanisms with fine granularity and accountability can be achieved through accountable CP-ABE, as shown in Figure 5. Usually, the system involves five entities: the AA, the CSP, the DO, the DU, and the auditor. The CSP, the DO, and the DU play the same roles as those in basic

Table 2. The Comparison of Revocable CP-ABE Constructions

Schemes	Parameter Size			User-Side Computation Cost		Server-Side	Policy	Group	Universe	Security	Assumption
	$\|PK\|$	$\|CT\|$	$\|SK\|$	Encryption	Decryption	Computation Cost					
AI09a-1 [3]	$(m_1+2u+1)\|\mathbb{G}\|$	$(n_1+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+2)\|\mathbb{G}\|$	$(2I+3)$**exp**	$(2I+3)$**pair**+I**exp**	-	LSSS	Prime	semi-Large	S-STM	q-type
AI09a-2 [3]	$(m_1+7)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n_1+2r+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+4)\|\mathbb{G}\|$	$(2n_1+3r+2)$**exp**	$(2I+2r+1)$**pair**+$(I+r)$**exp**	-	LSSS	Prime	semi-Large	S-STM	q-type
YWRL [104]	$(3n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n+1)\|\mathbb{G}\|$	$(n+2)$**exp**	$(n+1)$**pair**+n**exp**	n**exp**	$\mathbf{AND}^*_{+,-}$	Prime	Small	S-STM	DBDH
HN [28]	$\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2I+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+1)\|\mathbb{G}\|+\log\ell\|\mathbb{Z}_p^*\|$	$(2I+2)$**exp**	$(2I+1)$**pair**+$(I+\log\ell)$**exp**	I**exp**	Tree	Prime	Large	-	BDH
YJR [99]	$(2n+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+2)\|\mathbb{G}\|$	$(5n_1+2)$**exp**	$(2I+1)$**pair**+I**exp**	2**exp**	LSSS	Prime	Small	S-ROM	q-type
LCL+ [36]	$(n+4d_{\max}+5)\|\mathbb{G}_1\|+3\|\mathbb{G}_T\|$	$(2d_{\max}+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\ell(d_{\max}+1)\|\mathbb{G}_3\|$	$(3n_1+3d_{\max}+5)$**exp**	$(2I+d_{\max}+3)$**pair**+$(I+2)$**exp**	4**exp**	LSSS	Composite	Small	F-STM	New
PYSC [74]	$(n+2u+3)\|\mathbb{G}\|$	$4\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n+5)\|\mathbb{G}\|$	$(2I+5)$**exp**	7**pair**+2I**exp**	-	$\mathbf{AND}^*_{+,-}$	Prime	Small	S-STM	q-type
YLL+ [100]	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2I+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$\|\mathbb{G}\|+\|\mathbb{Z}_p^*\|$	$(2I+3)$**exp**	1**exp**	$(2I+2)$**pair**+$(I+1)$**exp**	Tree	Prime	Large	G-ROM	-
LYZL [54]	$(n+r+d_t+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n_1+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+r+d_t^2+2)\|\mathbb{G}\|$	$(2n_1+r+d_t+4)$**exp**	$(2I+4)$**pair**+$(I+r+d_t)$**exp**	-	LSSS	Prime	Small	S-STM	q-type
ZCL+14 [109]	$(4n+2u+1)\|\mathbb{G}\|$	$2\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(n+1)\|\mathbb{G}\|$	$(2I+5)$**exp**	$(r+2)$**pair**	2**pair**	$\mathbf{AND}^*_{+,-}$	Prime	Small	S-STM	q-type
FHR [17]	$\|\mathbb{G}_0\|+(n+1)\|\mathbb{G}_1\|+2\|\mathbb{G}_T\|$	$2\|\mathbb{G}_0\|+2I\|\mathbb{G}_1\|+2\|\mathbb{G}_T\|$	$3\ell\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$(2I+2)$**pm**+2**exp**	$(3I+1)$**pair**+I**exp**	$(n+4\ell+7)$**pm**	Tree	Prime	Small	F-STM	DBDH
CDLQ [14]	$7\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n_1+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$\|\mathbb{Z}_p^*\|$	$(4n_1+3)$**exp**	1**exp**	$(3I+2)$**pair**+I**exp**	LSSS	Prime	Large	S-STM	q-type
QZZC [76]	$7\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n_1+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$\|\mathbb{G}\|$	$(4n_1+3)$**exp**	2**pair**	$(3I+2)$**pair**+I**exp**	LSSS	Prime	Large	S-STM	q-type
XYM [95]	$5\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+2)\|\mathbb{G}\|$	$(5n_1+2)$**exp**	$(6I+1)$**pair**+2I**exp**	$(5I+2)$**exp**	LSSS	Prime	Large	S-STM	q-type
LYH+ [44]	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2I+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n+1)\|\mathbb{G}\|$	$(2I+2)$**exp**	$(3I+2)$**pair**+I**exp**	$(3I+2)$**exp**	Tree	Prime	Large	S-ROM	DCDH

† u: the total number of users the system has; r: the number of revocation events; $d_{\max}$: the maximum length of the label strings; d_t: the maximum depth of the time tree.

Table 3. The Special Feature Comparison in Revocable CP-ABE Constructions

Schemes	Key Technique	Backward Secrecy	Forward Secrecy	Re-keying Cost	
				Computation Cost	Storage Cost
AI09a-1 [3]	Broadcast Encryption	✗	✓	-	-
AI09a-2 [3]	Broadcast Encryption	✗	✓	-	-
YWRL [104]	Proxy Re-encryption	✓	✓	r_{att}**exp**	$r_{\text{att}}\lvert\mathbb{Z}_p^*\rvert$
HN [28]	Proxy Re-encryption	✓	✓	$(2I+2)$**exp**	$\log \ell \lvert\mathbb{Z}_p^*\rvert$
YJR [99]	Proxy Re-encryption	✓	✓	$2r_{\text{att}}$**exp**	$2r_{\text{att}}\lvert\mathbb{Z}_p^*\rvert$
LCL+ [36]	Proxy Re-encryption	✓	✓	$r_{\text{user}}(2d_{\max}+3)$**exp**	$r_{\text{user}}(d_{\max}+2)\lvert\mathbb{G}_{1,3}\rvert + r_{\text{user}}\lvert\mathbb{Z}_p^*\rvert$
PYSC [74]	Broadcast Encryption	✗	✓	-	-
YLL+ [100]	Proxy-assisted Decryption	✓	✓	-	-
LYZL [54]	Broadcast Encryption	✗	✓	-	-
ZCL+14 [109]	Broadcast Encryption with Ciphertext Update	✓	✓	r**exp**	$r\lvert\mathbb{Z}_p^*\rvert$
FHR [17]	System Public Key Update	✗	✓	-	-
CDLQ [14]	Proxy-assisted Decryption	✓	✓	-	-
QZZC [76]	Proxy-assisted Decryption	✓	✓	-	-
XYM [95]	Proxy Re-encryption	✓	✓	$r_{\text{user}}(4d_t+3)$**exp**	$r_{\text{user}}(2d_t+2)\lvert\mathbb{G}\rvert$
LYH+ [44]	Proxy Re-encryption	✓	✓	$(n+2u)$**exp**	$2\log n\lvert\mathbb{G}\rvert$

† r_{att}: the number of revoked attributes; r_{user}: the number of revoked users.

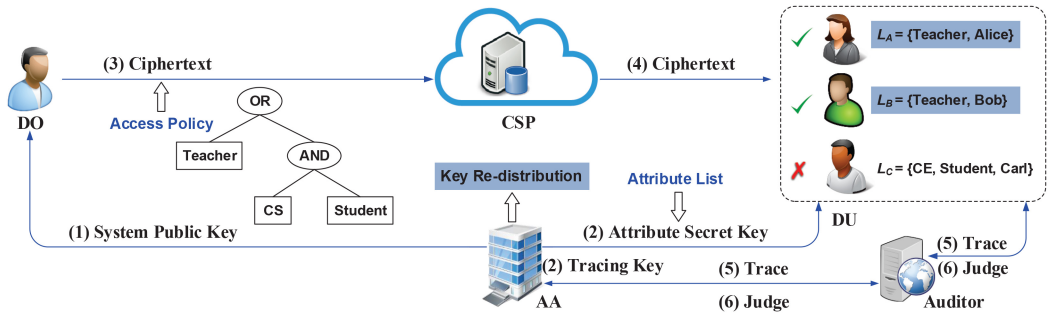


Fig. 5. The application scenario of accountable CP-ABE.

CP-ABE. Besides publishing system public keys (procedure (1)) and issuing attribute secret keys, the AA is semi-trusted and generates a tracing key in accountable CP-ABE (procedure (2)). The DO chooses an access policy $\mathbb{A}$ himself and integrates $\mathbb{A}$ into the ciphertext (procedure (3)). Then, the DU obtains the ciphertext from the CSP to retrieve the original data based on his/her attribute secret keys (procedure (4)). Once malicious behaviors exist, the corresponding entity should be traced (procedure (5)) and then is judged by the auditor (procedure (6)).

It is noted that the functionality of accountability in CP-ABE is of importance in practical applications, because two types of key abuse issues exist. For one thing, the key abuse behavior of a user should be traced to prevent framing other honest users who have the attribute set that is also held by malicious users. For another, the illegal key re-distribution behavior of the AA should also be accountable. Considering the example in Figure 5, if $\mathbb{A}$=(Teacher OR (CS AND Student)), then the ciphertext can be successfully decrypted by the DU with L_A = {Teacher, Alice} and the DU with L_B = {Teacher, Bob}, while it cannot be decrypted by the DU with L_C = {CE, Student, Carl}. Because Alice and Bob have the same attribute list, the key abuse behavior of Alice may be mistaken

as that of Bob. Another case is Bob is framed due to the key re-distribution behavior of the AA. In any case, the entity who misbehaves should be accountable by the auditor. The accountability of CP-ABE has two categories: (1) Black-box accountability: only a decryption equipment is leaked. In this case, given a decryption equipment as a black box, the entity who constructs the black box can be specified by the auditor. (2) White-box accountability: the attribute secret key is directly leaked. In this case, given an attribute secret key, the entity who leaks this key can be specified by the auditor.

5.2 Syntax of Accountable CP-ABE

In a black-box accountable CP-ABE, the definitions of Encrypt and Decrypt are the same as those of basic CP-ABE. Therefore, we only explain the other algorithms here.

- $\mathsf{Setup}(1^\lambda, u) \to (TK, PK, MK)$: The AA will run this algorithm, and it is called the system setup algorithm. After inputting λ and u, which represents the number the system's users, the algorithm returns the tracing key TK, PK, and MK.
- $\mathsf{KeyGen}(PK, MK, i, L) \to SK_{i,L}$: The AA will run this algorithm to generate attribute secret keys. After inputting PK, MK, an index $i \in \{1, 2, \ldots, u\}$, and L, the algorithm returns $SK_{i,L}$ as the secret key corresponding to i and L.
- $\mathsf{Trace}^{\mathsf{D}}(PK, TK, L) \to \mathcal{I}$: The tracing algorithm can be run by anyone. Given black-box access to a key-like decryption box D, after inputting PK, the tracing key TK, an attribute list L, the algorithm returns $\mathcal{I} \subseteq \{1, 2, \ldots, u\}$ as an index set for specifying the malicious set. Note that this definition defines public accountability. If the record of attribute secret keys is involved, this algorithm correspondingly defines private accountability.
- $\mathsf{Judge}(\cdot) \to$ guilty or innocent: This is a protocol among the AA, the identified identity $i \in \mathcal{I}$, and the auditor. The AA inputs (PK, MK, D, TK), the user inputs $(PK, SK_{i,L})$, and the auditor inputs (PK, D). The auditor will judge whether the user i is guilty.

For white-box accountable CP-ABE, the definitions of Setup, Encrypt, and Decrypt are the same as those of basic CP-ABE. Here, we only explain the other algorithms.

- $\mathsf{KeyGen}(PK, MK, ID, L) \to SK_{ID,L}$: The AA will run this algorithm to generate attribute secret keys. After inputting PK, MK, an identity ID, and L, the algorithm returns $SK_{ID,L}$ as the secret key corresponding to ID and L.
- $\mathsf{Trace}(PK, SK_{ID,L}) \to ID$ or $\top$: The tracing algorithm can be run by anyone. After inputting PK and an attribute secret key $SK_{ID,L}$, the algorithm first checks whether $SK_{ID,L}$ is well-formed or not. If $SK_{ID,L}$ is well-formed, which means it can be used in a well-formed decryption process, the algorithm outputs ID to indicate that $SK_{ID,L}$ is linked to ID. Otherwise, it outputs $\top$ to indicate that $SK_{ID,L}$ is invalid and there is no need to trace it. Note that this definition defines weak public accountability. If the master key is involved, this algorithm correspondingly defines private accountability.
- $\mathsf{Judge}(\cdot) \to$ guilty or innocent: This is an interactive protocol between a user with identity ID and the auditor. When the user is identified as a malicious user by the system based on the traced key $SK^*_{ID,L}$, the user inputs $(PK, SK_{ID,L})$, and the auditor inputs $(PK, SK^*_{ID,L})$ and judges whether the user is guilty.

5.3 Adversarial Model and Security Goals

Accountable CP-ABE solutions are applied to establish accountable access control systems in which the AA is semi-trusted. Similar to basic CP-ABE, accountable CP-ABE should satisfy *Data Confidentiality* and *Collusion Resistance* in addition to accountability. On one hand, *user*

accountability should be achieved to trace malicious users. For example, suppose $\mathbb{A}$=(Teacher OR (CS AND Student)), then the ciphertext can be decrypted by the DU with L_A = {Teacher, Alice} and the DU with L_B = {Teacher, Bob} individually. If Alice maliciously leaks her secret key, the auditor should decide Alice is guilty while Bob is not. On the other hand, *authority accountability* should be enabled to trace the malicious key re-distribution behavior of the AA. For instance, if the AA re-distributes the secret key to other users based on the attribute "Teacher," it should not be possible to frame Alice and Bob.

5.4 Research Status of Accountable CP-ABE

Li et al. [41] designed a CP-ABE solution supporting private user accountability in black-box models. The scheme has selective security in the ROM and only supports AND-gate policies.

To improve the expressiveness of access policies, Liu et al. [57] constructed an LSSS-based CP-ABE solution supporting user accountability. However, the scheme is based on the white-box model. Later, a new CP-ABE solution due to Liu et al. [58] can support the LSSS policy and realize public black-box user accountability. Although the schemes of References [57, 58] have the ability of resisting adaptive adversaries in standard models, they only support small attribute universe. To fill the above gap, Ning et al. [69] developed a CP-ABE solution to simultaneously realize large universe and white-box user accountability by optimizing the accountability of the scheme of Reference [57] even if its security analysis is based on the selective model.

The above schemes only realize accountability of malicious users, and the key redistribution behavior of the attribute authority cannot be traced. To address this issue, Ning et al. [68] designed an LSSS-based CP-ABE construction, which enables weak public accountability of both users and the attribute authority in the white-box model by a Paillier-style encryption-based commitment. The security proofs of the scheme are presented under new assumptions and the q-strong Diffie-Hellman (q-SDH) assumption. However, Zhang et al. [113] showed a security weakness of the scheme [68] by re-randomizing the attribute secret key and constructed an LSSS-based CP-ABE scheme supporting the user and authority accountability with full security in the ROM in the weak-public white-box model.

Ning et al. [66] further designed a black-box accountable LSSS-based CP-ABE solution by combining conventional CP-ABE with anonymous identity-based encryption with identity hierarchies. Full security and public user accountability are ensured in this scheme.

Liu et al. [61] described a large universe CP-ABE solution that simultaneously considered the issues of direct user revocation and black-box public user accountability. In standard models, the security analysis under q-type assumptions shows its selective security. To further realize ciphertext updating, a new CP-ABE solution is constructed by Liu et al. [59], in which the ciphertexts corresponding to the published revocation list are updated. In addition, the scheme simultaneously supports white-box user accountability and direct user revocation.

Jiang et al. [29] solved the issue of key delegation abuse in CP-ABE and proposed a white-box user-accountable CP-ABE solution of which the security analysis is given in generic group models. Nevertheless, this solution only supports AND-gate policies and hence is less expressive than other relevant schemes.

5.5 Comparison of Accountable CP-ABE Schemes

In Table 4, we make a comparison of accountable CP-ABE solutions according to the metrics used in the comparison of basic CP-ABE. In addition, we add a new metric "accountability computation cost" in Table 4. Obviously, the schemes of References [41, 69, 113] have a small and constant-size system public key. The schemes of References [57, 58, 66, 68] enable the LSSS policy based on composite-order groups and small universe. The schemes of References [41, 61, 69, 113] are large

Table 4. The Comparison of Accountable CP-ABE Constructions

Schemes	Parameter Size			User-Side Computation Cost		Accountability Computation Cost	Policy	Group	Universe	Security	Assumption
	$\|PK\|$	$\|CT\|$	$\|SK\|$	Encryption	Decryption						
LRZW [41]	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4N+8\|ID\|)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4n+4\|ID\|)\|\mathbb{G}\|$	$(4N+8\|ID\|+1)$**exp**	$(4n+4\|ID\|)$**pair**	$(4N+8\|ID\|+1)$**exp**	$\mathbf{AND}^{\star}_{m}$	Prime	Large	S-ROM	DBDH, DLIN
LCW13a [57]	$(n+3)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(2n_1+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+3)\|\mathbb{G}_{1,3}\|$	$(3n_1+3)$**exp**	$(2I+1)$**pair**$+(I+2)$**exp**	$(2\ell+5)$**pair**$+3$**exp**	LSSS	Composite	Small	F-STM	New
LCW13b [58]	$(n+3\sqrt{u}+3)\|\mathbb{G}_1\|+\sqrt{u}\|\mathbb{G}_T\|$	$(16\sqrt{u}+2n_1)\|\mathbb{G}_1\|+\sqrt{u}\|\mathbb{G}_T\|$	$(\ell+4)\|\mathbb{G}_{1,3}\|$	$(21\sqrt{u}+5n_1+1)$**exp**	$(2I+10)$**pair**$+I$**exp**	$8\lambda(u/\epsilon)^2(u+1)(21\sqrt{u}+5n_1+1)$**exp**	LSSS	Composite	Small	F-STM	q-type
NDC+15 [69]	$6\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n_1+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+3)\|\mathbb{G}\|$	$(5n_1+3)$**exp**	$(3I+1)$**pair**$+(I+2)$**exp**	Arithmetic operations	LSSS	Prime	Large	S-STM	q-type
NDCW [68]	$(n+6)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(2n_1+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+3)\|\mathbb{G}_{1,3}\|+2\|\mathbb{G}_1\|$	$(3n_1+5)$**exp**	$(2I+3)$**pair**$+(I+4)$**exp**	$(2\ell+7)$**pair**$+8$**exp**	LSSS	Composite	Small	F-STM	New, SDH
ZLZ+ [113]	$4\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(2n_1+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+3)\|\mathbb{G}_{1,3}\|+2\|\mathbb{G}_1\|$	$(3n_1+4)$**exp**	$(2I+3)$**pair**$+5$**exp**	$(2\ell+5)$**pair**$+8$**exp**	LSSS	Composite	Large	S-STM	New, SDH
NCD+16 [66]	$(n+u+5)\|\mathbb{G}_{1,4}\|+\|\mathbb{G}_4\|+\|\mathbb{G}_T\|$	$(2n_1+4)\|\mathbb{G}_{1,4}\|+\|\mathbb{G}_T\|$	$(\ell+u+6)\|\mathbb{G}_{1,3}\|$	$(3n_1+5)$**exp**	$(2I+5)$**pair**$+I$**exp**	$8\lambda(u/\epsilon)^2(u+1)(3n_1+5)$**exp**	LSSS	Composite	Small	F-STM	q-type
LW16 [61]	$(4\sqrt{u}+5)\|\mathbb{G}_1\|+\sqrt{u}\|\mathbb{G}_T\|$	$(15\sqrt{u}+3n_1)\|\mathbb{G}_1\|+\sqrt{u}\|\mathbb{G}_T\|$	$(2\ell+\sqrt{u}+2)\|\mathbb{G}_1\|$	$(20\sqrt{u}+7n_1+1)$**exp**	$(3I+9)$**pair**$+I$**exp**	$8\lambda(u/\epsilon)^2(u+1)(20\sqrt{u}+7n_1+1)$**exp**	LSSS	Composite	Large	S-STM	q-type
LDZW [59]	$(n+2u+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n_1+r+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+6)\|\mathbb{G}\|$	$(2n_1+3)$**exp**	$(2I+3)$**pair**$+(I+\log u)$**exp**	$(2\ell+2)$**pair**$+2$**exp**	LSSS	Prime	Small	S-STM	SDH
JSMG18a [29]	$2(n+\log u)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2(n+\log u)-I)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+\log u)\|\mathbb{G}\|$	$(2(n+\log u)-I+1)$**exp**	$(n+\log u+I)$**pair**	$(n+\log u)$**pair**	**AND**	Prime	Small	G-ROM	DBDH

† $|ID|$: the bit length of a user's identity; λ: the security parameter of the system; ϵ: the lower-bound of the black-box's decryption ability.

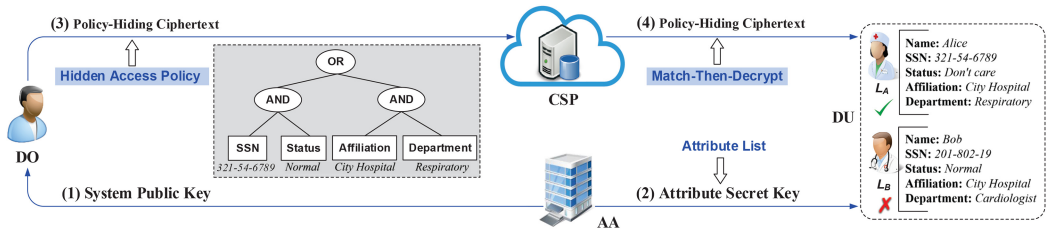


Fig. 6. The application scenario of policy-hiding CP-ABE.

universe constructions that can resist attacks from selective adversaries. Only the schemes of References [29, 41, 59, 69] are designed in prime order groups. However, the accountability of these four schemes can only be performed by the attribute authority. Based on the state-of-the-art for accountable CP-ABE, we know the black-box accountability is only realized in the schemes of References [41, 58, 61, 66], in which the scheme of Reference [41] achieves private user accountability and the schemes of References [58, 61, 66] enable public user accountability. Only the schemes of References [68, 113] simultaneously realize the accountability of both users and the authority even if it is based on the weak-public white-box model. Accordingly, if user accountability, authority accountability, and large universe are required simultaneously, the scheme of Reference [113] is preferred.

6 POLICY-HIDING CP-ABE

6.1 Application Scenario: Policy-hiding Fine-grained Access Control

Figure 6 shows fine-grained access control with attribute privacy protection based on policy-hiding CP-ABE. Usually, the system involves four entities: the AA, the CSP, the DO, and the DU, and they play the same roles as those in basic CP-ABE. It is noted that policy-hiding CP-ABE can be categorized into fully hiding CP-ABE schemes and partially hiding CP-ABE schemes. In fully hiding CP-ABE, no information about the the access policy's attribute information is revealed. In partially hiding CP-ABE, sensitive attribute values that are involved in policies are hidden while the corresponding attribute names are made public similar to the ciphertext. Currently, fully hiding CP-ABE can only be indirectly constructed based on threshold policies and attribute-hiding inner-product encryption. Particularly, in existing literatures, there are no concrete fully hiding CP-ABE solutions. Therefore, in the subsequent description, we directly use the term "policy-hiding" instead of "partially hiding."

In access control systems with fine granularity and hidden access policies, the procedures (1), (2), (3), and (4) are similar to those of basic fine-grained access control except that the access policy is hidden in the procedures (3) and (4). Specifically, to realize attribute privacy protection, the chosen policy will be embedded in the ciphertext in a privacy-aware manner such that a third party including the CSP is not able to read the access policy. This is especially true in the fields of military, commerce, and healthcare.

For instance, suppose the DO encrypts a health record under access policy $\mathbb{A}$=((Affiliation: *City Hospital* AND Department: *Respiratory*) OR (SSN: *321-54-6789* AND Status: *Normal*)), and the ciphertext is outsourced to the CSP. Note that everyone including the CSP can read the access policy and may figure out that the user with Social Security number *321-54-6789* possibly suffers a respiratory problem. In this case, the privacy of the user is leaked, which reflects the significance of concealing the access policy. As shown in Figure 6, in policy-hiding CP-ABE, the attributes that are tied to the leaf nodes in the tree policy are hidden.

6.2 Syntax of Policy-hiding CP-ABE

In a policy-hiding CP-ABE solution, there are algorithms including Setup, KeyGen, Encrypt, and Decrypt. Note that the definitions of the first two algorithms are similar to those of basic CP-ABE.

- Encrypt$(PK, M, \mathbb{A}) \rightarrow CT_{\mathbb{A}}$: It is adopted by the DO to generate ciphertexts. The DO first chooses an access policy $\mathbb{A}$, then he takes PK and M as inputs to return $CT_{\mathbb{A}}$ as the ciphertext of M while hiding $\mathbb{A}$, which will be stored on the CSP.
- Decrypt$(PK, CT_{\mathbb{A}}, SK_L) \rightarrow M$ or $\perp$: This algorithm is adopted by the DU to recover messages. After inputting PK, $CT_{\mathbb{A}}$ of M without knowing $\mathbb{A}$, and SK_L with regard to L, it returns M if $L \models \mathbb{A}$, and otherwise outputs $\perp$ to indicate an error.

Remark 1. To improve the decryption efficiency in policy-hiding CP-ABE, previous schemes [35, 110] introduce the idea of anonymous attribute matching and divide the decryption algorithm into two phases including matching phase and decryption phase.

6.3 Adversarial Model and Security Goals

Anonymous access control systems can be realized through policy-hiding CP-ABE, in which the AA is trusted and the CSP will be honest-but-curious. Similar to basic CP-ABE, policy-hiding CP-ABE should also satisfy *Data Confidentiality* and *Collusion Resistance*. In addition, the access policy should be protected in policy-hiding access control systems. For instance, when recorded health data are encrypted by the DO using a policy-hiding CP-ABE solution, the chosen access policy is hidden and tied to the corresponding ciphertext with the form (Affiliation: $*$ AND Department: $*$) OR (SSN: $*$ AND Status: $*$) while the concrete values are replaced with "*".

6.4 Research Status of Policy-hiding CP-ABE

Based on Reference [12], Nishide et al. [70] designed a policy-hiding CP-ABE solution of which the security is based on ROM under the DBDH and DLIN assumptions. However, the scheme only realizes selective security. To further realize full security against adaptive adversaries, Lai et al. [34] designed a CP-ABE solution in groups of composite order that is fully secure under new assumptions. Nevertheless, the schemes of References [34, 70] only support AND-gate policies and hence are less expressive. In addition, the schemes are small-universe constructions, and the ciphertext length is linearly related to the total number of attribute values.

To achieve expressiveness, Lai et al. [35] developed a CP-ABE construction based on Reference [37] with LSSS policies. The dual system encryption technique [92] is applied to give the security analysis. To improve the decryption efficiency in policy-hiding CP-ABE, Zhang et al. [110] designed a new policy-hiding CP-ABE solution in which an attribute-matching process is performed before full decryption. The key idea lies in that exclusive ciphertext components are utilized to test whether the attribute list satisfies the hidden policy without needing decryption. The test is computationally efficient compared with full decryption.

Hur [26] proposed a tree-based policy-hiding CP-ABE solution for realizing secure and privacy-preserving data sharing in smart grid. The idea in essence is to obfuscate attributes in the access tree by updating the plain attributes assigned to each leaf node with a hash value. In addition, the scheme is able to outsource laborious decryption overhead to the CSP. However, the disadvantage of the scheme is the lack of formal security proofs. Zhou et al. [119] developed a policy-hiding CP-ABE construction that can compress the ciphertext length to a constant. The scheme enables AND-gate policies and obtains provable security under the q-type assumption. To hide AND-gate policies, the scheme due to Phuong et al. [73] transforms attributes and access policies to two vectors and then hides the access policy based on the idea of inner product encryption.

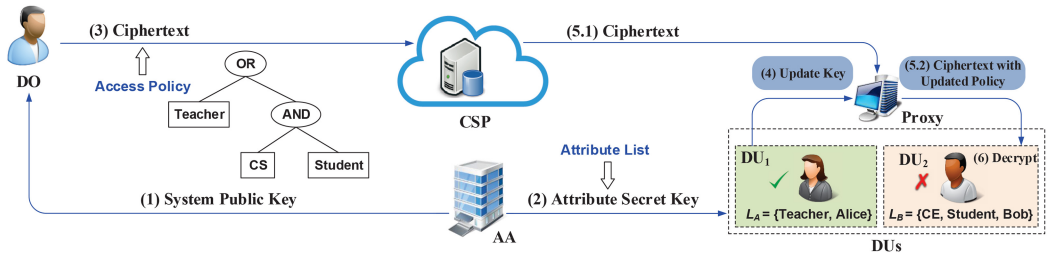


Fig. 7. The application scenario of CP-ABE with policy updating.

For better performance, Zhang et al. [111] designed a policy-hiding CP-ABE solution in which only several bilinear pairings in decryption are required. However, the expressiveness of the scheme needs to be improved. Qiu et al. [77] designed an AND-gate CP-ABE solution to hide access policies and enable keyword search. The scheme has selective security in generic group models.

To simultaneously improve expressiveness and computation efficiency, Zhang et al. [117] constructed a policy-hiding large-universe CP-ABE scheme that supports LSSS policies. In particular, the attribute matching process only needs two bilinear pairings and the scheme's full security is demonstrated in standard models. Very recently, the scheme of Reference [117] is improved in Reference [107] to achieve better decryption efficiency. Xiong et al. [94] designed a fine-grained broadcast encryption scheme supporting hidden LSSS policies. However, the user still needs to perform expensive bilinear pairing operations even if the outsourced computation technique is used.

6.5 Comparison of Policy-hiding CP-ABE Schemes

We compare policy-hiding CP-ABE schemes in Table 5 with regard to the metrics used in the comparison of basic CP-ABE schemes. A different point is that the decryption cost is divided into "matching phase" and "decryption phase" in Table 5. It easily follows that the schemes of References [26, 107, 110, 111, 117] support large universe and only the schemes of References [34, 35, 107, 117] are capable of resisting adaptive adversaries in standard models. Only the scheme of Reference [119] has a constant-size ciphertext even if its encryption algorithm involves bilinear pairing operations. As for the decryption cost, the schemes of References [26, 107, 110, 111, 117] realize efficient attribute matching, because the computation cost is not affected by the access policy's complexity. Only the schemes of References [26, 107, 111] enable constant computation overhead for decryption. As for the expressiveness of access policies, only the schemes of References [35, 107, 117] support the expressive LSSS policies. The schemes of References [70, 73, 77, 119] are small universe constructions based on prime order groups. Based on the state-of-the-art for policy-hiding CP-ABE, we know only the scheme of Reference [26] further realizes outsourced decryption, and the rest of policy-hiding CP-ABE constructions cannot support other functionalities. In general, the schemes of References [107, 117] are suitable for attribute-based and privacy-aware access control in the setting of mobile clouds considering the decryption performance and full security.

7 CP-ABE WITH POLICY UPDATING

7.1 Application Scenario: Fine-grained and Policy Updating Access Control

As shown in Figure 7, CP-ABE with policy updating, in which the technique of proxy re-encryption is usually involved, can be applied to enable policy updating access control systems with fine

Table 5. The Comparison of Policy-hiding CP-ABE Constructions

Schemes	Parameter Size			Encryption Cost	Decryption Cost		Policy	Group	Universe	Security	Assumption
	$\|PK\|$	$\|CT\|$	$\|SK\|$		Matching Phase	Decryption Phase					
NYO [70]	$(2N+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2N+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(N+1)\|\mathbb{G}\|$	$(2N+2)$**exp**	$(3n+1)$**pair**	$(3n+1)$**pair**	$\mathbf{AND}^*_m$	Prime	Small	S-STM	DBDH, DLIN
LDL11 [34]	$(N+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(N+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+1)\|\mathbb{G}\|$	$(N+2)$**exp**	$(n+1)$**pair**	$(n+1)$**pair**	$\mathbf{AND}^*_m$	Composite	Small	F-STM	New
LDL12 [35]	$(n+2)\|\mathbb{G}_1\|+2\|\mathbb{G}_4\|+\|\mathbb{G}_T\|$	$2\|\mathbb{G}_1\|+4n_1\|\mathbb{G}_{1,4}\|+2\|\mathbb{G}_T\|$	$(\ell+2)\|\mathbb{G}_{1,3}\|$	$(8n_1+4)$**exp**	$(2I+1)$**pair**+I**exp**	$(2I+1)$**pair**+I**exp**	LSSS	Composite	Small	F-STM	New
ZCL+13 [110]	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3N+4)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(n+2)\|\mathbb{G}\|$	$(3N+6)$**exp**	3**pair**	4n**pair**	$\mathbf{AND}^*_m$	Prime	Large	S-ROM	DBDH, DLIN
Hur13a [26]	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2I+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3\ell+1)\|\mathbb{G}\|$	$(2I+3)$**exp**	1**pair** + 1**exp**	1**pair** + 1**exp**	Tree	Prime	Large	-	-
ZHW [119]	$(3N+1)\|\mathbb{G}\|$	$2\|\mathbb{G}\|+\|M\|$	$(2\ell+1)\|\mathbb{G}\|$	1**pair**+ 3**exp**	$(2n+1)$**pair**	$(2n+1)$**pair**	$\mathbf{AND}^*_{+,-}$	Prime	Small	S-STM	q-type
PYS [73]	$(8n+30)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4I+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4n+3)\|\mathbb{G}\|$	$(12n+39)$**exp**	$(4n+12)$**pair**	$(4n+12)$**pair**	$\mathbf{AND}^*_{+,-}$	Prime	Small	S-STM	DBDH, DLIN
ZCL+17 [111]	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3N+3)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(3n+4)\|\mathbb{G}\|$	$(3N+5)$**exp**	2**pair**	4**pair**	$\mathbf{AND}^*_m$	Prime	Large	S-ROM	DBDH, DLIN
QLSZ [77]	$(N+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(N+n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3n+1)\|\mathbb{G}_2\|$	$(2n+3)$**exp**	$(2n+1)$**pair**+1**exp**	$(2n+1)$**pair**+1**exp**	$\mathbf{AND}^*_m$	Prime	Small	Generic	-
ZZD18 [117]	$2\|\mathbb{G}_1\|+\|\mathbb{G}_4\|+\|\mathbb{G}_{1,4}\|+\|\mathbb{G}_T\|$	$(3n_1+2)\|\mathbb{G}_{1,4}\|+2\|\mathbb{G}_T\|$	$(\ell+2)\|\mathbb{G}_{1,3}\|$	$(6I+4)$**exp**	2**pair**+2I**exp**	$(I+2)$**pair**+2I**exp**	LSSS	Composite	Large	F-STM	New
ZHMR [107]	$4\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(n_1+1)\|\mathbb{G}_{1,4}\|+\|\mathbb{G}_T\|$	$(\ell+2)\|\mathbb{G}_{1,3}\|$	$(3n_1+2)$**exp**	2**pair**+2I**exp**	2**pair**+2I**exp**	LSSS	Composite	Large	F-STM	New

† $|M|$: the length of the message M.

granularity. The access control system involves five entities: the AA, the CSP, the DO, the DU, and the proxy. The AA is trusted, and the CSP and the proxy are honest-but-curious. In policy updating access control systems, the procedures (1), (2), and (3) are similar to those of basic fine-grained access control. CP-ABE with policy updating allows the proxy to help users to recover messages from ciphertexts, even if the original access policy is not matched by the attribute list. Specifically, based on an update key (procedure (4)) from another user whose attributes satisfy the original policy, the proxy can transform the ciphertext to a new one with an updated policy (procedures (5.1) and (5.2)). Finally, the ciphertext with the updated policy can be decrypted by the user (procedure (6)). Considering the example in Figure 7, if $\mathbb{A}$=(Teacher OR (CS AND Student)), then the ciphertext can be successfully decrypted by DU_1 with attribute list L_A = {Teacher, Alice}, while it cannot be decrypted by DU_2 with attribute list L_B = {CE, Student, Bob}. If DU_1 is going on holiday, she can delegate DU_2 in advance to decrypt on her behalf. To achieve this goal, DU_1 just gives an update key to the proxy who takes charge of updating the ciphertext intended for DU_1. The update key is then used by the proxy to update the underlying policy in the original ciphertext such that DU_2 can successfully recover the message in the updated ciphertext.

7.2 Syntax of CP-ABE with Policy Updating

A CP-ABE scheme supporting policy updating comprise algorithms including Setup, KeyGen, Encrypt, the update key generation algorithm UKeyGen, the ciphertext updating algorithm Update, and Decrypt. Note that the first three algorithms have the identical definitions to those of basic CP-ABE.

- $\mathsf{UKeyGen}(PK, SK_L, \mathbb{A}') \rightarrow UK_{L\rightarrow\mathbb{A}'}$: After inputting PK, a secret key SK_L corresponding to L and $\mathbb{A}'$, the algorithm returns an update key $UK_{L\rightarrow\mathbb{A}'}$, where L represents the attribute list of DU_1 and $\mathbb{A}'$ is satisfied by the attributes of DU_2.
- $\mathsf{Update}(PK, UK_{L\rightarrow\mathbb{A}'}, CT_{\mathbb{A}}) \rightarrow CT_{\mathbb{A}'}$: After inputting PK, an update key $UK_{L\rightarrow\mathbb{A}'}$ and a ciphertext $CT_{\mathbb{A}}$, if and only if $L \models \mathbb{A}$, the algorithm generates a ciphertext $CT_{\mathbb{A}'}$ with $\mathbb{A}'$ acting as the access policy and the plaintext being the same as that of $CT_{\mathbb{A}}$. That is, the access policy is updated from $\mathbb{A}$ to $\mathbb{A}'$.
- $\mathsf{Decrypt}(PK, CT_{\mathbb{A}}, SK_L) \rightarrow M$ or $\perp$: The DU (DU_1 or DU_2) takes as inputs PK, $CT_{\mathbb{A}}$ of M with regard to $\mathbb{A}$, and SK_L in which L is integrated, and returns M if and only if $L \models \mathbb{A}$. Note that if $CT_{\mathbb{A}}$ is the original (respectively, updated) ciphertext, it is decrypted by DU_1 (respectively, DU_2).

7.3 Adversarial Model and Security Goals

CP-ABE with policy updating should satisfy *Data Confidentiality* and *Collusion Resistance*. In addition, because the honest-but-curious proxy is introduced and the updated ciphertexts exist, *Master Key Security* should be realized. To be specific, even if the proxy colludes with DU_2, it is infeasible for them to learn of secret keys of DU_1. For example, if $\mathbb{A}$=(Teacher OR (CS AND Student)) and $\mathbb{A}'$=(Teacher OR (CE AND Student)), the proxy can transform the ciphertext $CT_{\mathbb{A}}$ to $CT_{\mathbb{A}'}$ by updating the access policy from $\mathbb{A}$ to $\mathbb{A}'$ based on the update key $UK_{L\rightarrow\mathbb{A}'}$, where $L \models \mathbb{A}$. Obviously, even if DU_2 cannot decrypt $CT_{\mathbb{A}}$, he is capable of decrypting $\mathbb{A}'$. Based on *Master Key Security*, the secret key of DU_1 is protected from the collusion attack of the proxy and DU_2.

7.4 Research Status of CP-ABE with Policy Updating

Based on Reference [12], Liang et al. [50] developed a CP-ABE solution that enables policy updating through the technique of proxy re-encryption. More precisely, this construction is a ciphertext-policy attribute-based proxy re-encryption scheme, denoted by CP-ABPRE for short,

which is proven selectively secure. The CP-ABPRE construction due to Luo et al. [62] is a small-universe solution and the policy supports multiple attribute values and wildcards in the form of AND denoted by AND^*_m. The security proofs of the scheme are presented in standard models under the DBDH and the computational bilinear Diffie-Hellman (CBDH) assumptions. Liu et al. [55] introduced a new CP-ABPRE solution in which time is taken into account. The main idea lies in that both a disjunctive normal form (DNF) access policy and time are embedded into the ciphertext. In addition, each user has an attribute list and time. The security proofs are based on ROM under the complexity assumption BDH. Liang et al. [49] designed a more expressive CP-ABPRE solution by the technique of dual system encryption. The scheme adopts groups of composite order and is able to resist adaptive attacks in standard models. Yang et al. [101] designed a new CP-ABPRE solution that is a tree-based construction. The scheme enables constant-size attribute secret key based on the technique of proxy re-encryption, and its security analysis involves generic group models.

To support attribute privacy protection and policy updating at the same time, Zhang et al. [112] formalized the notion of anonymous CP-ABPRE for the first time and proposed a concrete scheme. In the scheme, a technique named as "match-then-re-encrypt" is proposed to help the proxy check whether a ciphertext should be transformed without requiring the access policy. Security proofs of the scheme are presented without needing the ROM under the assumptions DBDH, DLIN, and CBDH. Jiang et al. [30] developed a CP-ABE construction with policy updating and attribute revocation. The ciphertext for user decryption is of constant length. The scheme supports AND-gate policies and is proven selectively secure. Yeh et al. [102] designed an attribute-based health records access control system, in which the techniques of CP-ABE, proxy re-encryption, and Merkle hash trees are used. The policy updating is realized by updating the involved attributes in the ciphertext.

7.5 Comparison of CP-ABE Schemes with Policy Updating

We compare CP-ABE solutions with policy updating in Table 6 in terms of the metrics used in the comparison of basic CP-ABE schemes. In addition, we further consider the update server-side computation cost for one update in Table 6. We know that only the scheme of Reference [101] has constant-size system public key and hence supports large universe. However, generic group models are involved in the security analysis. Only the schemes of References [30, 55, 101] have constant decryption cost, which is not affected by the complexity of access policies. In particular, the ciphertext of the scheme of Reference [30] is constant-size. The scheme of Reference [49] is more expressive than other schemes even if composite-order groups are adopted in the concrete design. The schemes of References [30, 49, 50, 62, 112] have security in the standard model. In particular, based on the state-of-the-art for CP-ABE with policy updating, we know only the scheme of Reference [112] realizes privacy-aware policy updating; that is, the access policy is hidden and can be updated by the proxy. Therefore, in sensitive applications such as attribute-based medical health access control systems, the scheme of Reference [112] can be adopted. In resource-limited applications, the schemes of References [30, 55, 101] are preferred.

8 MULTI-AUTHORITY CP-ABE

8.1 Application Scenario: Multi-authority Fine-grained Access Control

Figure 8 illustrates that multi-authority CP-ABE can enable distributed and access control with fine granularity. The access control system involves five entities: the central authority (CA), the AAs, the CSP, the DO, and the DU. Usually, the CA is completely trustworthy and publishes system public keys. Each AA is responsible for an attribute set, and any user can apply for corresponding secret keys from AAs. In multi-authority fine-grained access control systems, the procedures (1) and (2) are similar to those of basic fine-grained access control. In addition, the DU obtains a

Table 6. The Comparison of CP-ABE Constructions with Policy Updating

Schemes	Parameter Size			User-Side Computation Cost		Update Server-Side Computation Cost (One Update)	Policy	Group	Universe	Security	Assumption
	$\|PK\|$	$\|CT\|$	$\|SK\|$	Encryption	Decryption						
LCLS09 [50]	$(3n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+1)\|\mathbb{G}\|$	$(n+3)$**exp**	$(n+1)$**pair**	$(n+1)$**pair**	$\mathbf{AND}^*_{+,-}$	Prime	Small	S-STM	ADBDH, CTDH
LHC [62]	$(N+2n+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4\ell+1)\|\mathbb{G}\|$	$(n+3)$**exp**	$(2n+1)$**pair**	$(2n+1)$**pair**	$\mathbf{AND}^*_m$	Prime	Small	S-STM	DBDH, CBDH
LWW [55]	$(n+3)\|\mathbb{G}\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+1)\|\mathbb{G}\|$	1**pair**+$(n+2)$**pm**	2**pair**	1**pair**+$(n+1)$**pm**	DNF	Prime	Small	S-ROM	BDH
LAL+ [49]	$(n+6)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(2n_1+4)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|+\ell_{\mathsf{sig}}$	$(\ell+3)\|\mathbb{G}_{1,3}\|$	$(3n_1+6)$**exp**+C_{sig}	$(2I+2)$**pair**+I**exp**+C_{ver}	$(3\ell+11)$**pair**+$(6\ell+5)$**exp**+C_{sig}+C_{ver}	LSSS	Composite	Small	F-STM	New, q-type
YZL+ [101]	$\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{Z}_p\|$	2**pair** + $(2n_1+3)$**exp**	1**pair**+2**exp**	$(2I+1)$**pair**+$(I+1)$**exp**	Tree	Prime	Large	G-ROM	-
ZLCL16 [112]	$(3N+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3N+3)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(4n+4)\|\mathbb{G}\|$	$(3N+5)$**exp**	$(3I+3)$**pair**	$(3n+4)$**pair**+1**exp**	$\mathbf{AND}^*_m$	Prime	Small	S-STM	DBDH, DLIN, CBDH
JSMG18b [30]	$(2n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|+\|\mathbb{Z}_p\|$	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+1)\|\mathbb{G}\|$	$(r_{\max}+3)$**exp**	2**pair**+3**exp**	1**pair**+$2r_{\max}$**exp**	$\mathbf{AND}^*_m$	Prime	Small	S-STM	q-type

† ℓ_{sig}: the bit length of a signature; C_{sig} (respectively, C_{ver}): the computation cost of signature generation (respectively, verification); $r_{\max}$: the maximum number of revocation events.

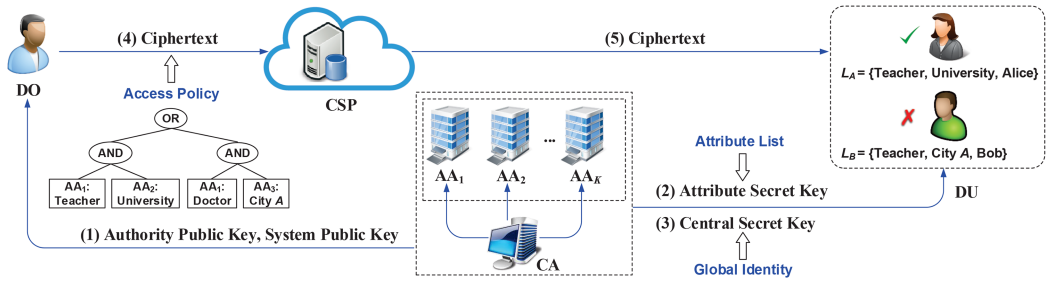


Fig. 8. The application scenario of multi-authority CP-ABE.

central secret key associated with his/her global identity from the CA (procedure (3)). The DO outsources ciphertexts of original messages to the CSP under an access policy over attributes from different AAs (procedure (4)). Finally, the DU can obtain the ciphertext from the CSP to recover the corresponding message based on secret keys (procedure (5)). For example, as shown in Figure 8, $\mathbb{A}$=((AA$_1$:Teacher AND AA$_2$:University) OR (AA$_1$:Doctor AND AA$_3$:City A)), and the ciphertext will be successfully decrypted by the DU with L_A = {Teacher, University, Alice}, while it cannot be decrypted by the DU with L_B = {Teacher, City A, Bob}. The attributes "Teacher" and "Doctor" are managed by AA$_1$, the attribute "University" is managed by AA$_2$, and the attribute "City A" is managed by AA$_3$. It is noted that the CA in some multi-authority CP-ABE schemes is removed and these schemes are called decentralized CP-ABE. Because traditional multi-authority CP-ABE has the key-escrow problem due to the CA, decentralized CP-ABE is preferable for designing access control systems with fine granularity.

8.2 Syntax of Multi-authority CP-ABE

The algorithms Setup, AAKeyGen, CAKeyGen, Encrypt, and Decrypt are the key ingredients of a multi-authority CP-ABE solution. It is noted that the definitions of Encrypt and Decrypt are the same as those of basic CP-ABE except that the attributes in access policies and secret keys are from different AAs. In the following, we present the other algorithms:

- Setup$(1^\lambda) \rightarrow (\{pk_k, sk_k\}_{1\le k\le K}, PK, MK)$: This algorithm is called the system setup algorithm, which will be run by the CA. After inputting λ, the algorithm returns (pp_k, sk_k) as a public and secret key pair of AA$_k$ with $1 \le k \le K$. It additionally outputs PK and MK where PK is publicly published and MK is secretly maintained by the CA.
- AAKeyGen$(PK, sk_k, \text{GID}, L^{(k)}) \rightarrow SK_{L^{(k)}}$: This algorithm is called the attribute secret key generation algorithm, which will be run by the AAs. After inputting PK, the AA$_k$'s secret key sk_k, a user's global identity GID, and $L^{(k)}$ as an attribute list of which the attributes are managed by AA$_k$, the algorithm outputs the secret key $SK_{L^{(k)}}$ corresponding to $L^{(k)}$. If the user GID's attribute list is $L = \{L^{(k)}\}_{k\in\mathcal{I}}$, then $SK_L = \{SK_{L^{(k)}}\}_{k\in\mathcal{I}}$ acts as attribute secret keys, where $\mathcal{I}$ represents the index set of AAs of which the attribute domains include the user's attributes.
- CAKeyGen$(PK, MK, \text{GID}) \rightarrow SK_C$: This algorithm is called the central key generation algorithm, which will be run by the CA. After inputting PK, MK, and GID, the algorithm outputs SK_C as the user GID's central secret key.

Remark 2. In decentralized CP-ABE, the public and secret key pairs of AAs are generated by themselves and the central key generation algorithm is no longer needed. The definition of the

algorithm for generating attribute secret key is the same with that of the above CA enabled multi-authority CP-ABE.

8.3 Adversarial Model and Security Goals

Similar to basic CP-ABE, multi-authority CP-ABE should also satisfy *Data Confidentiality* and *Collusion Resistance*. The difference lies in that *Collusion Resistance* further considers the attributes of DUs from different AAs. For example, suppose $\mathbb{A}$=((AA_1:Teacher AND AA_2:University) OR (AA_1:Doctor AND AA_3:City *A*)), then the ciphertext cannot be decrypted by the DU with L_B = {Teacher, City *A*, Bob} and the DU with L_C = {Doctor, City *B*, Carl} individually. Even if Bob and Carl collude by combining their secret keys, they should not succeed in decrypting the ciphertext under $\mathbb{A}$. In addition, multi-authority CP-ABE should take the AA corruption issue into account. Certainly, the upper bound of allowed corrupted AAs is usually less than the total number of AAs. It is indispensable to design multi-authority CP-ABE schemes with the upper bound as large as possible.

8.4 Research Status of Multi-authority CP-ABE

Chase [10] designed a multi-authority ABE construction on the groups of prime order. In the scheme, a CA and multiple AAs exist. The CA is in charge of issuing seeds for each AA, which further generates attribute-related keys of users. However, the CA is endowed with a super power in the system such that it is capable of decrypting any possible ciphertext. Thus, it may become the system security vulnerability.

Aiming to remove such trusted CA and prevent the colluding AAs from pooling their information on a particular user, Chase et al. [11] first designed an anonymous key distribution mechanism and then gave a multi-authority CP-ABE solution, where no CA exists and each pair of AAs shares secret pseudorandom functions (PRF) seeds in the initialization phase. The scheme is secure if at most $n_a - 2$ AAs are corrupted, where n_a denotes the number of involved AAs. Lin et al. [52] considered the problem of incorrect key distribution in the multi-authority setting and introduced several threshold multi-authority ABE schemes without a trusted CA. Their basic CP-ABE scheme is secure against at most $\lfloor \frac{n_a}{2} - 1 \rfloor$ AAs' corruption.

Different from the selective security in References [10, 11, 52], for the first time, Lewko and Waters [38] designed a fully secure decentralized CP-ABE solution on the groups of composite order in the ROM without needing cooperations among multiple AAs. The employed access policy can be described by any LSSS matrix. To achieve adaptive security in standard models, Liu et al. [56] developed a multi-authority CP-ABE solution on composite-order groups with multiple CAs and AAs. These different authorities are responsible for generating keys for users that are related to identities and attributes, respectively.

To alleviate the computation overhead caused by multiple CAs, a multi-authority CP-ABE solution with a CA and AAs in cloud storage is developed by Li et al. [48]. In contrast to that of Reference [10], the CA in this scheme is employed to issue identity-related keys and cannot decrypt any ciphertext. Similar to that in Reference [56], the scheme's security proofs are given in standard models by the technique of dual system encryption. Li et al. [47] showed how to deploy the multi-authority ABE [11] to realize secure health record access control. This scheme also supports indirect user and attribute revocation and no CA exists. Considering the user's privacy concerns of both identity and attributes, Han et al. [23] gave a privacy-preserving solution by combining a key extract protocol with decentralized CP-ABE, where no CA exists and there is no requirement of interactions among AAs. Besides, any AA is allowed to dynamically join or leave without re-initializing the access control system. Jung et al. [31] simultaneously addressed the issues of data confidentiality and user identity privacy and proposed a semi-anonymous privilege control scheme

and further extended it to support full anonymity by leveraging the oblivious transfer technique. This scheme does not need CAs and is proved to be secure if at least two AAs are not corrupted.

Rouselakis and Waters [79] designed an LSSS-based large-universe decentralized CP-ABE solution with selective security in the ROM. Cui and Deng [13] investigated the attribute revocation issue, which is challenging and important in decentralized CP-ABE. The AAs periodically update the attribute-related key components, and the revoked users cannot obtain the new keys. To prevent collusion attacks, a user's attribute secret key components are tied up with a time attribute and his/her identifiers. Although the key generation task is assigned to multiple AAs, the drawback of single-point failure still exists, since each AA governs a partial and disjoint attribute universe of the system. To tackle this technical challenge, Xue et al. [98] presented a robust and auditable multi-authority CP-ABE solution, where the total system attribute universe is managed by multiple AAs, each of which can independently generate secret keys of any possible attribute set for users. The malicious behaviors of AAs can also be detected by a CA. The decentralized CP-ABE solution due to Zhang et al. [106] is a large-universe construction that adopts prime order groups while enabling traceability in the sensing of white box. Any malicious user who deliberately leaks his/her partial/modified keys would be precisely identified. Yu et al. [103] designed a decentralized CP-ABE solution that can revoke malicious users in a direct way. In the revocation phase, both the related public parameters and ciphertext components are updated, and the keys of remaining users are not required to update. A verifiable revocation mechanism is designed to detect if the target ciphertext has been correctly updated by the cloud server.

8.5 Comparison of Multi-authority CP-ABE Schemes

Table 7 compares the numeric results and characteristics among multi-authority CP-ABE constructions. The column named *AA Corruption* shows the upper bound of allowed corrupted AAs of the corresponding scheme. It is obvious that only one exponent operation is required in Reference [48] by employing an efficient outsourced approach. The schemes of References [10, 11, 23, 31, 47, 52, 98, 103] adopt prime order groups and are proven secure in the standard model. The large attribute universe is only supported by the schemes of References [79, 106]. Only the schemes of References [13, 38, 48, 56] achieve full security, in which the security proofs of the schemes of References [48, 56] are based on standard models. Based on the state-of-the-art for multi-authority CP-ABE, we know the schemes of References [11, 13, 23, 31, 38, 47, 52, 79, 103, 106] are decentralized, among which the multiple AAs in References [11, 31, 47] need to cooperate to initialize the system. The architecture of schemes of References [10, 48, 98] requires a CA and the scheme of Reference [56] needs multiple CAs. Specifically, the CA in References [10, 98] can decrypt any possible ciphertext and it may be the system security bottleneck. Every AA in Reference [98] is responsible for governing all of the attributes in the system, while each AA in the other schemes can only manage a disjoint subset of the system attribute universe. The schemes of References [13, 23, 38, 48, 56, 79, 103, 106] are secure if at least one of the AAs is not corrupted by the adversary. Due to the fact that the AAs need to cooperate with each other to set up the system, the schemes of References [11, 31, 47] are secure against at most $n_a - 2$ AAs' corruption. Generally, if both full security and decentralization are required, the schemes of References [13, 38] are preferred.

9 HIERARCHICAL CP-ABE

In this section, we present the application scenario of hierarchical CP-ABE: hierarchical fine-grained access control. More details including the syntax, the adversarial model and security goals, the research status, and comparisons of hierarchical CP-ABE are given in the Supplemental Material A.

Table 7. The Comparison of Multi-authority CP-ABE Constructions

Schemes	Parameter Size			Computation Cost		Policy	Group	Universe	Security	AA Corruption	Assumption
	$\|PK\|$	$\|CT\|$	$\|SK\|$	Encryption	Decryption						
Chase07 [10]	$(n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+1)\|\mathbb{G}\|$	$(\ell+2)$**exp**	$(I+1)$**pair**+I**exp**	Threshold	Prime	Small	S-STM	n_a	DBDH
CC09 [11]	$(n+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+1)\|\mathbb{G}\|$	$(\ell+2)$**exp**	$(I+1)$**pair**+I**exp**	Threshold	Prime	Small	S-STM	n_a-2	DBDH
LCLS08 [52]	$(n\mu+m\nu+3)\|\mathbb{G}\|$	$I(\mu+\nu)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\ell\|\mathbb{G}\|$	$(I(\mu+\nu)+1)$**exp** +**pair**	I**pair**+$I+t$**exp**	Tree	Prime	Small	S-STM	$\lfloor\frac{n_a}{2}-1\rfloor$	DBDH
LW11 [38]	$(n+1)\|\mathbb{G}_1\|+n\|\mathbb{G}_T\|$	$2n_1\|\mathbb{G}_1\|+(n_1+1)\|\mathbb{G}_T\|$	$\ell\|\mathbb{G}\|$	$(5I+1)$**exp**+**pair**	$2I$**pair**+I**exp**	LSSS	Composite	Small	F-ROM	n_a-1	New
LCH+ [56]	$(n+2)\|\mathbb{G}_1\|+n_c\|\mathbb{G}_T\|$	$(2n_1+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(2n_c+n_a n_c+\ell)\|\mathbb{G}\|$	$(3I+n_c+1)$**exp**	$(2I+1)$**pair**+I**exp**	LSSS	Composite	Small	F-STM	n_a-1	New
LML+ [48]	$(n+n_a+1)\|\mathbb{G}_1\|+n_a\|\mathbb{G}_T\|$	$(2n_1+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(2+n_a+\ell)\|\mathbb{G}\|$	$(3I+2)$**exp**	1**exp**	LSSS	Composite	Small	F-STM	n_a-1	New
LYZ+12 [47]	$(n+n_a+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+n_a+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\ell+1)\|\mathbb{G}\|$	$(\ell+n_a+2)$**exp**	$(I+n_a+1)$**pair**+$I+n_a$**exp**	Threshold	Prime	Small	S-STM	n_a-2	DBDH
HSM+14 [23]	$(2n+4n_a+3)\|\mathbb{G}\|+n_a\|\mathbb{G}_T\|$	$(3n_a+2n_1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(6n_a+\ell)\|\mathbb{G}\|$	$(3I+4n_a)$**exp**	$(4n_a+2I)$**pair**+(n_a+I)**exp**	LSSS	Prime	Small	S-STM	n_a-1	q-type
JLW+15 [31]	$\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n_t+2n_t n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+1)\|\mathbb{G}\|$	$(2n_t I+n_t+2)$**exp**	$(2n_t I+1)$**pair**+$(n_t I+n_t)$**exp**	LSSS	Prime	Small	S-STM	n_a-2	DBDH
RW15 [79]	$(n_a+1)\|\mathbb{G}\|+n_a\|\mathbb{G}_T\|$	$4n_1\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\ell\|\mathbb{G}\|$	$(6n_1+1)$**exp**	$3I$**pair**+I**exp**	LSSS	Prime	Large	S-ROM	n_a-1	q-type
CD16 [13]	$(n+1)\|\mathbb{G}_1\|+n\|\mathbb{G}_T\|$	$2n_1\|\mathbb{G}_1\|+(n_1+1)\|\mathbb{G}_T\|$	$\ell\|\mathbb{G}\|$	$(5I+1)$**exp**+**pair**	$2I$**pair**+I**exp**	LSSS	Composite	Small	F-ROM	n_a-1	New
XXH+ [98]	$(n+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\ell+2)\|\mathbb{G}\|$	$(3I+2)$**exp**	$(2I+1)$**pair**+I**exp**	LSSS	Prime	Small	S-STM	n_a	q-type
ZLML [106]	$(3n_a+1)\|\mathbb{G}\|+n_a\|\mathbb{G}_T\|$	$5n_1\|\mathbb{G}\|+(n_1+1)\|\mathbb{G}_T\|$	$4\ell\|\mathbb{G}\|$	$(8I+1)$**exp**+**pair**	$3I$**pair**+$4I$**exp**	LSSS	Prime	Large	S-ROM	n_a-1	q-type
YWN+ [103]	$(2n+n_a+2)\|\mathbb{G}\|+n_a\|\mathbb{G}_T\|$	$(4n_1+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2n_a+4\ell)\|\mathbb{G}\|$	$(5I+2)$**exp**	$(5I+n_a)$**pair**+$2I$**exp**	LSSS	Prime	Small	S-STM	n_a-1	q-type

† n_c: the number of CAs; n_a: the number of involved AAs; n_t: the number of involved trees.

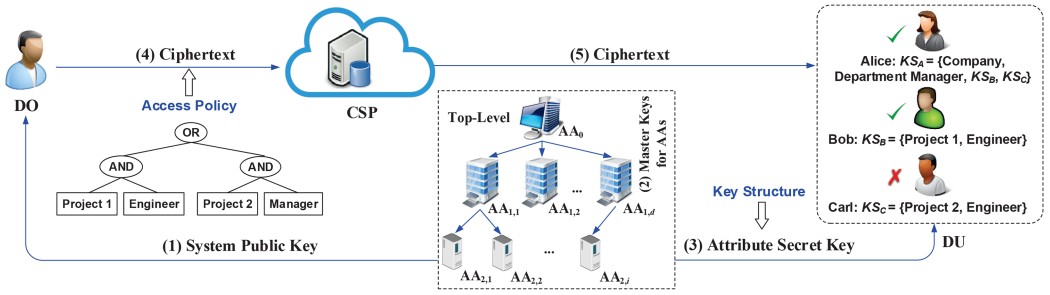


Fig. 9. The application scenario of hierarchical CP-ABE.

Application Scenario: Hierarchical CP-ABE can be adopted in realizing hierarchical access control with fine granularity as shown in Figure 9. The access control system involves four types of entities: the AAs, the CSP, the DO, and the DU. Compared with basic CP-ABE, the difference lies in that there are many AAs and they are in a hierarchical organization. An AA may act as a DU in hierarchical CP-ABE. Each higher-level AA delegates its own lower-level AAs by issuing them master keys or secret keys associated with key structures. Note that a key structure in hierarchical CP-ABE is a set associated with attributes. However, the key structure is different from the attribute list in that each element in the key structure can still be an attribute set [88, 89]. Each lower-level AA trusts its own high-level AA. Similar to the AA in basic CP-ABE, the top-level AA is completely trustworthy in hierarchical CP-ABE. In hierarchical attribute-based access control systems, the top-level AA publishes the system public keys (procedure (1)) and issues lower-level master keys for lower-level AAs based on corresponding key structures (procedure (2)). The DU can obtain attribute secret keys associated with his/her key structures from the corresponding AA (procedure (3)). The DO outsources ciphertexts of messages to the CSP under a specified policy (procedure (4)). If a given ciphertext can be decrypted by a user (procedure (5)), then other higher-level users including AAs can also succeed in decrypting the ciphertext. For example, as shown in Figure 9, $\mathbb{A}$=((Project 1 AND Engineer) OR (Project 2 AND Manager)), and the ciphertext can be successfully decrypted by the DU Bob with the key structure KS_B = {Project 1, Engineer}, while it cannot be decrypted by the DU Carl with the key structure KS_C = {Project 2, Engineer}. In addition, hierarchical CP-ABE allows the DU Alice with the key structure KS_A = {Company, Department Manager, KS_B, KS_C} to decrypt the ciphertext, which makes sense, because the company's department manager Alice indeed has the privilege to decrypt the ciphertext in practice. In other words, the company's department manager delegates the engineers of the project 1 to decrypt the ciphertext.

10 ONLINE/OFFLINE CP-ABE

In this section, we present the application scenario of online/offline CP-ABE: fine-grained access control with offline computation. The syntax, the adversarial model and security goals, the research status, and comparisons of online/offline CP-ABE are given in the Supplemental Material B.

Application Scenario: Figure 10 shows that CP-ABE with offline computation can be utilized to design fine-grained access control mechanisms for resource-constrained data owners in the setting of mobile cloud computing. Similar to the scenario of basic CP-ABE, the system involves the AA, the CSP, the DO, and the DU, in which the AA is completely trustworthy. In fine-grained access control systems with offline computation, the AA publishes the system public keys (procedure (1)) and generates offline keys named immediate attribute secret keys (procedure (2)) in the offline

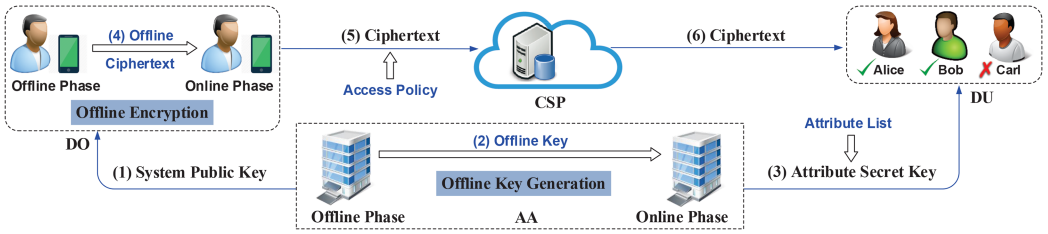


Fig. 10. The application scenario of online/offline CP-ABE.

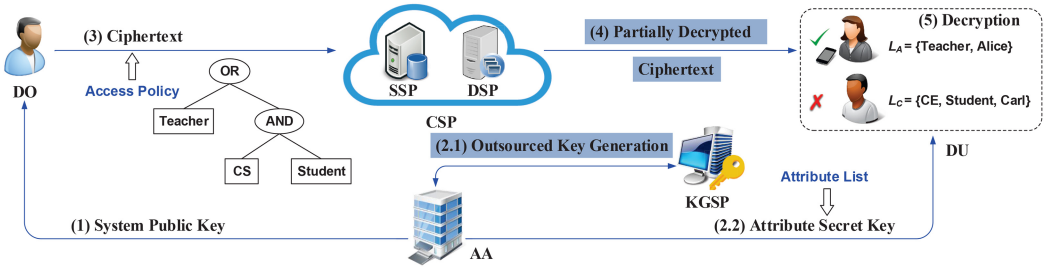


Fig. 11. The application scenario of outsourced CP-ABE.

phase. In the online phase of key generation, the AA issues attribute secret keys to each DU based on his/her corresponding attribute list (procedure (3)). The computational task for encryption can be finished by the DO even if the access policy and message are not determined (procedure (4)). Hence, once the message is given, the DO can efficiently generate the corresponding ciphertext in the online phase (procedure (5)). Finally, the DU can decrypt the ciphertext to retrieve the original message based on his/her attribute secret keys (procedure (6)). To improve the efficiency of key distribution, the AA does the vast majority of the work for generating attribute secret keys offline, before knowing the DU's attribute list. Therefore, online/offline CP-ABE can be utilized to enable large-scale access control with resource-constrained data owners.

11 OUTSOURCED CP-ABE

In this section, we present the application scenario of outsourced CP-ABE: fine-grained access control with outsourced computation. More details including the syntax, the adversarial model and security goals, the research status, and comparisons of outsourced CP-ABE are given in the Supplemental Material C.

Application Scenario: Figure 11 shows that CP-ABE with outsourced computation can be used to design access control mechanisms for resource-constrained users in mobile clouds. For access control systems from outsourced CP-ABE construction, the involved entities include the AA, the CSP, the DO, the DU, and the key generation service provider (KGSP) where the CSP consists of the decryption service provider (DSP) and the storage service provider (SSP). Similar to basic CP-ABE, only the AA is completely trustworthy. In fine-grained access control systems with outsourced computation, the AA first publishes the system public keys (procedure (1)). Then, the AA outsources the computation in key generation to the KGSP (procedure (2.1)) to generate attribute secret keys for the DU (procedure (2.2)). The DO outsources ciphertexts of messages to the SSP of the CSP under a specified policy (procedure (3)). The DU is capable of outsourcing the main decryption task to the DSP. Based on the partially decrypted ciphertext (procedure (4)) that is generated by the DSP, the original message will be finally recovered by the DU with small computation cost

Table 8. Comprehensive Feature Comparison of CP-ABE Constructions

Schemes	Revocation		Accountability		PH	PU	MA	Hierarchy	Computation	
	User	Attribute	User	Authority					Offline	Outsourcing
SW05-1, SW05-2 [81], CN [12], BSW [5], GJPS [21] LCLX [51], LOS+ [37], HLR [24], SYGH [85], Waters [93] LW12 [39], RW [78], ZZC [115], MST [65], AC [1], XHX [97]	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
AI09a-1, AI09a-2 [3], LCL+ [36], PYSC [74], LYZL [54]	Direct	✗	✗	✗	✗	✗	✗	✗	✗	✗
YLL+ [100]	Direct	✗	✗	✗	✗	✗	✗	✗	✗	Dec
YWRL [104], HN [28], YJR [99]	Indirect	Indirect	✗	✗	✗	✗	✗	✗	✗	✗
ZCL+14 [109], FHR [17]	Direct	Direct	✗	✗	✗	✗	✗	✗	✗	✗
CDLQ [14], QZZC [76]	Indirect	✗	✗	✗	✗	✗	✗	✗	✗	Dec
XYM [95]	Indirect	✗	✗	✗	✗	✗	✗	✗	✗	✗
LYH+ [44]	Indirect	Indirect	✗	✗	✗	✗	✗	✗	✗	✗
LRZW [41]	✗	✗	Pri, BB	✗	✓	✗	✗	✗	✗	✗
LCW13a [57]	✗	✗	WP, WB	✗	✗	✗	✗	✗	✗	✗
LCW13b [58], NCD+16 [66]	✗	✗	Pub, BB	✗	✗	✗	✗	✗	✗	✗
NDC+15 [69], JSMG18a [29]	✗	✗	Pri, WB	✗	✗	✗	✗	✗	✗	✗
NDCW [68], ZLZ+ [113]	✗	✗	WP, WB	WP, WB	✗	✗	✗	✗	✗	✗
LW16 [61]	Direct	✗	Pub, BB	✗	✗	✗	✗	✗	✗	✗
LDZW [59]	Direct	✗	Pri, WB	✗	✗	✗	✗	✗	✗	✗
NYO [70], LDL11 [34], LDL12 [35], ZCL+13 [110], ZHW [119] PYS [73], ZCL+17 [111], QLSZ [77], ZZD18 [117], ZHMR [107]	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
Hur13a [26]	✗	✗	✗	✗	✓	✗	✗	✗	✗	Dec
LCLS09 [50], LHC [62], LAL+ [49], JSMG18b [30]	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
LWW [55]	Direct	✗	✗	✗	✗	✓	✗	✓	✗	✗
YZL+ [101]	Direct	✗	✗	✗	✗	✓	✗	✗	✗	Dec
ZLCL16 [112]	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
Chase07 [10], LCH+ [56]	✗	✗	✗	✗	✗	✗	with CA	✗	✗	✗
CC09 [11], RW15 [79], JLW+15 [31]	✗	✗	✗	✗	✗	✗	no CA	✗	✗	✗
LCLS08 [52], LW11 [38], HSM+14 [23]	✗	✗	✗	✗	✗	✗	no CA	✗	✗	✗
LML+ [48]	Indirect	Indirect	✗	✗	✗	✗	with CA	✗	✗	Dec
CD16 [13]	Indirect	Indirect	✗	✗	✗	✗	no CA	✗	✗	✗
LYZ+12 [47]	Indirect	Indirect	✗	✗	✗	✓	no CA	✗	✗	✗
XXH+ [98]	✗	✗	✗	Pri, WB	✗	✗	with CA	✗	✗	✗
ZLML [106]	✗	✗	Pri, WB	✗	✗	✗	no CA	✗	✗	✗
YWN+ [103]	✗	✓	✗	✗	✗	✗	with CA	✗	✗	✗
LWWR11 [42], DWQ+14 [16], WZL+16 [90] TYX+17 [87], LYZ19 [45]	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
WLD12 [88]	Direct	✗	✗	✗	✗	✗	✗	✓	✗	✗
WLWG11 [89]	✗	Indirect	✗	✗	✗	✗	✗	✓	✗	✗
LZCX18 [46]	✗	✗	✗	✗	✗	✗	✗	✗	Enc	✗
HW14 [25]	✗	✗	✗	✗	✗	✗	✗	✗	Enc (or KeyGen)	✗
LJW+[60]	✗	✗	✗	✗	✗	✗	✗	✗	KeyGen, Enc	v-Dec
ZZL+[118]	✗	✗	✗	✗	✗	✗	with CA	✗	KeyGen, Enc	✗
DR17 [15]	Indirect	✗	✗	✗	✗	✗	no CA	✗	Enc	Dec
MZY+[64]	✗	✗	✗	✗	✗	✗	✗	✗	KeyGen, Enc	Dec
ZWZ [114]	✗	✗	✗	✗	✓	✗	✗	✗	KeyGen, Enc	✗
GHW [22]	✗	✗	✗	✗	✗	✗	✗	✗	✗	Dec
LDGW [33], QDLM [75], LZMW [53], LWZ+ [43] FNWL [19], NCD+17 [67]	✗	✗	✗	✗	✗	✗	✗	✗	✗	v-Dec
LHLC [40]	✗	✗	✗	✗	✗	✗	✗	✗	✗	KeyGen, v-Dec
WHZ+ [91]	✗	✗	✗	✗	✗	✗	✗	✗	✗	Enc, Dec
ZML [108]	✗	✗	✗	✗	✗	✗	✗	✗	✗	KeyGen, Enc, Dec
MZW+ [63]	✗	✗	✗	✗	✗	✗	✗	✗	✗	Enc, v-Dec
OSZ [71]	✗	✗	✗	✗	✗	✗	✗	✗	✗	Enc

† PH: Policy Hiding; PU: Policy Update; MA: Multi-Authority; Pri: Private; Pub: Public; WP: Weak Public; BB: Black-Box; WB: White-Box; v-Dec: verifiable outsourced decryption.
‡ The gray cells show the features of the main CP-ABE schemes of given categories.

based on his/her attribute secret keys (procedure (5)). Because the AA can outsource the majority of key generation tasks to the KGSP to improve the system efficiency, outsourced CP-ABE enables large-scale access control systems with resource-limited users.

Each type of CP-ABE has heretofore been reviewed in Sections 3–11. As we know, a concrete CP-ABE scheme can fall into different categories simultaneously. To clearly show the cryptographically functional features of each CP-ABE scheme, Table 8 makes a comprehensive feature comparison of CP-ABE schemes following the features shown in Figure 2. These features are not present in the basic CP-ABE schemes, and they are realized in enhanced CP-ABE shown in gray cells in Table 8. Because each CP-ABE scheme has been reviewed in the corresponding section, Table 8 only shows the concrete feature comparison. Obviously, all the features of a given CP-ABE scheme can be easily specified, which is exactly the purpose of presenting Table 8.

12 CONCLUSION AND FUTURE DIRECTIONS

Cloud computing is today the dominant computing paradigm that makes it possible for anyone to access dynamic resources in a flexible manner. As a significant one-to-many cryptographic technique, ABE successfully tackles the security and privacy issues in outsourced data and hence is suitable for cloud computing access control. This survey provided a comprehensive overview of

the state-of-the-art in ABE. First, we proposed a taxonomy, in which ABE is classified into CP-ABE, KP-ABE, anti-quantum ABE, and generic ABE. As a major part of this survey, CP-ABE further falls into nine subcategories including basic CP-ABE schemes, revocable CP-ABE schemes, accountable CP-ABE schemes, policy-hiding CP-ABE schemes, CP-ABE schemes with policy updating, multi-authority CP-ABE schemes, hierarchical CP-ABE schemes, online/offline CP-ABE schemes, and outsourced CP-ABE schemes. In addition, we proposed a comprehensive and holistic assessment criteria for ABE. Then, each type of CP-ABE and KP-ABE were methodically analyzed based on the access control application scenario, the specification, the adversarial model, the security goal, the design strategies, and features. To illustrate the advantages and drawbacks of ABE schemes, we made detailed comparisons based on the proposed assessment criteria with respect to security and performance. Finally, for the sake of large-scale deployments of cloud platforms in the future, we pointed out that much exploration is needed for mature ABE technologies.

There are a lot of challenging and interesting problems in the ABE research.

- *Fully-secure and expressive ABE solutions on prime order groups in standard models:* As mentioned previously, fully secure ABE is more advantageous than selectively secure ABE, because it does not require adversaries to specify their target access policies or attribute lists until receiving the system public keys [37]. Existing fully secure ABE solutions are usually designed on composite-order groups, and dual system encryption [92] and complex assumptions are involved in the security proofs [37, 39, 117]. ABE schemes on composite-order groups are less efficient than those on prime order groups because of larger parameters. In addition, dual system encryption is complicated in that semi-functional keys, and ciphertexts need to be constructed in the proof of security [92]. A few existing fully secure ABE schemes in prime order groups suffer from the drawbacks of low efficiency, less expressive access policies, or using random oracle models [1, 37]. Therefore, it is necessary to design fully secure and expressive ABE schemes on prime order groups in the standard model.
- *Efficient ABE schemes without pairings:* Typical cryptographic operations include the bilinear pairing, the exponentiation, the point multiplication and arithmetic operations in groups, and so on. The computational complexity of the bilinear pairing is larger than that of other operations [82]. However, bilinear pairings are required in most of the existing ABE schemes, such as revocable ABE [76], accountable ABE [29], policy-hiding ABE [117], ABE with policy updating [112], and multi-authority ABE [103]. Compared to the symmetric and traditional public-key encryption, ABE suffers from the disadvantage of high computational cost because of the frequent pairing operations. For the practicality of ABE in resource-constrained scenarios exemplified by sensor-based sensitive data collection, it is meaningful to design efficient ABE schemes without pairings.
- *Expressive ABE schemes with as many functionalities as possible:* As mentioned earlier, functionalities such as revocation, accountability, attribute privacy protection, policy updating, decentralization, and key hierarchy are very important for practical deployments of ABE-based access control systems. As for revocation, the scheme of Reference [109] realizes direct attribute and user revocation, whereas it only supports AND-gate policies. It remains an open problem to design attribute directly revocable LSSS-based ABE schemes with backward and forward secrecy. Previous accountable ABE schemes suffer from several drawbacks such as the AND-gate policy [29, 41] and selective security [59]. Particularly, decentralization is not considered in accountable ABE. Existing policy-hiding ABE schemes involve either the AND-gate policy or composite-order groups [70, 117]. ABE schemes with policy updating cannot support attribute revocation [49, 112]. The issues of revocation and

accountability have not been well addressed in decentralized ABE [38] and hierarchical ABE [45]. In general, it would be interesting to construct expressive ABE schemes with as many functionalities as possible.

- *Efficient anti-quantum ABE:* With the technical advancements of quantum computation, it has been widely accepted that many public-key encryption schemes including ABE need security enhancements to resist possible quantum attacks. Although lattice-based algorithms can resist quantum attacks, there are only a few lattice-based ABE constructions [9, 105] that are selectively secure. In addition, lattice-based schemes lack practicability, because they have only been considered secure for inefficiently large parameters. Accordingly, more attention should be paid to anti-quantum ABE of better performance in the long term.
- *New methodologies for the security proof of ABE:* As two important proof techniques in public-key cryptography, partitioning [8] and dual system encryption [92] have been widely adopted by researchers. In a partitioning-based security proof of ABE, the target access policy or attribute list is partitioned into two parts. These two parts are, respectively, used to create attribute private keys and the challenge ciphertext [21, 96]. Although the partitioning technique has been proved useful in selective security, it is inadequate for proving full security of ABE [92]. The fundamental problem lies in that the access policy of ABE is complex, which makes the partitioning technique unusable. To overcome this obstacle, the proof technique of dual system encryption is introduced [92]. In the dual system encryption technique, ciphertexts and keys take on two forms: normal or semi-functional. Semi-functional ciphertexts and keys are only used in security proofs. A semi-functional ciphertext cannot be decrypted by a semi-functional key. In the security proof of ABE, the dual system encryption technique usually relies on composite-order groups to prove full security and hence has a performance limitation [37, 117]. Generally, to tackle the above challenges, it is interesting to explore new methodologies for security proof of ABE. The achievement in proof methods will promote more secure and efficient designs of ABE.

ACKNOWLEDGMENTS

The authors would like to thank the editors and referees for their invaluable suggestions.

REFERENCES

[1] Shashank Agrawal and Melissa Chase. 2017. FAME: Fast attribute-based message encryption. In *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS'17)*. ACM, 665–682.

[2] Ruqayah R. Al-Dahhan, Qi Shi, Gyu Myoung Lee, and Kashif Kifayat. 2019. Survey on revocation in ciphertext-policy attribute-based encryption. *Sensors* 19, 7 (2019), 1–22.

[3] Nuttapong Attrapadung and Hideki Imai. 2009. Conjunctive broadcast and attribute-based encryption. *Pairing-Based Cryptog.–Pairing'09*, Vol. 5671. Springer, 248–265.

[4] Mihir Bellare and Phillip Rogaway. 1993. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS'93)*. ACM, 62–73.

[5] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-policy attribute-based encryption. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'07)*. IEEE, 321–334.

[6] Matt Blaze, Gerrit Bleumer, and Martin Strauss. 1998. Divertible protocols and atomic proxy cryptography. In *Proceedings of the Advances in Cryptology Conference (EUROCRYPT'98)*, Vol. 1403. Springer, 127–144.

[7] Dan Boneh and Xavier Boyen. 2004. Short signatures without random oracles. In *Proceedings of the Advances in Cryptology Conference (EUROCRYPT'04)*, Vol. 3027. Springer, 56–73.

[8] Dan Boneh and Matt Franklin. 2001. Identity-based encryption from the Weil pairing. In *Proceedings of the Advances in Cryptology Conference (CRYPTO'01)*, Vol. 2139. Springer, 213–229.

[9] Xavier Boyen and Qinyi Li. 2016. Turing machines with shortcuts: Efficient attribute-based encryption for bounded functions. In *Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS'16)*, Vol. 9696. Springer, 267–284.

[10] Melissa Chase. 2007. Multi-authority attribute based encryption. In *Proceedings of the Theory of Cryptography Conference (TCC'07)*, Vol. 4392. Springer, 515–534.

[11] Melissa Chase and Sherman S. M. Chow. 2009. Improving privacy and security in multi-authority attribute-based encryption. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*. ACM, 121–130.

[12] Ling Cheung and Calvin Newport. 2007. Provably secure ciphertext policy ABE. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*. ACM, 456–465.

[13] Hui Cui and Robert H. Deng. 2016. Revocable and decentralized attribute-based encryption. *Comput. J.* 59, 8 (2016), 1220–1235.

[14] Hui Cui, Robert H. Deng, Yingjiu Li, and Baodong Qin. 2016. Server-aided revocable attribute-based encryption. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'16)*, Vol. 9879. Springer, 570–587.

[15] Sourya Joyee De and Sushmita Ruj. 2017. Efficient decentralized attribute based access control for mobile clouds. *IEEE Trans. Cloud Comput.* 8, 1 (2020), 124–137.

[16] Hua Deng, Qianhong Wu, Bo Qin, Josep Domingo-Ferrer, Lei Zhang, Jianwei Liu, and Wenchang Shi. 2014. Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts. *Inf. Sci.* 275 (2014), 370–384.

[17] Chun-I Fan, Vincent Shi-Ming Huang, and He-Ming Ruan. 2013. Arbitrary-state attribute-based encryption with dynamic membership. *IEEE Trans. Comput.* 63, 8 (2013), 1951–1961.

[18] Amos Fiat and Moni Naor. 1994. Broadcast encryption. In *Proceedings of the Advances in Cryptology Conference (CRYPTO'93)*, Vol. 773. Springer, 480–491.

[19] Xingbing Fu, Xuyun Nie, Ting Wu, and Fagen Li. 2018. Large universe attribute based access control with efficient decryption in cloud storage system. *J. Syst. Softw.* 135 (2018), 157–164.

[20] Gartner. 2019. Gartner forecasts worldwide public cloud revenue to grow 17.5 percent in 2019. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2019-04-02-gartner-forecasts-worldwide-public-cloud-revenue-to-g.

[21] Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. 2008. Bounded ciphertext policy attribute based encryption. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP'08)*, Vol. 5126. Springer, 579–591.

[22] Matthew Green, Susan Hohenberger, and Brent Waters. 2011. Outsourcing the decryption of abe ciphertexts. In *Proceedings of the 20th USENIX Conference on Security (USENIX'11)*. USENIX Association, 1–11.

[23] Jinguang Han, Willy Susilo, Yi Mu, Jianying Zhou, and Man Ho Au. 2014. PPDCP-ABE: Privacy-preserving decentralized ciphertext-policy attribute-based encryption. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'14)*, Vol. 8713. Springer, 73–90.

[24] Javier Herranz, Fabien Laguillaumie, and Carla Ràfols. 2010. Constant size ciphertexts in threshold attribute-based encryption. In *Proceedings of the International Workshop on Public Key Cryptography (PKC'10)*, Vol. 6056. Springer, 19–34.

[25] Susan Hohenberger and Brent Waters. 2014. Online/offline attribute-based encryption. In *Proceedings of the International Workshop on Public Key Cryptography (PKC'14)*, Vol. 8383. Springer, 293–310.

[26] Junbeom Hur. 2013. Attribute-based secure data sharing with hidden policies in smart grid. *IEEE Trans. Parallel Distrib. Syst.* 24, 11 (2013), 2171–2180.

[27] Junbeom Hur. 2013. Improving security and efficiency in attribute-based data sharing. *IEEE Trans. Knowl. Data Eng.* 25, 10 (2013), 2271–2282.

[28] Junbeom Hur and Dong Kun Noh. 2011. Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Trans. Parallel Distrib. Syst.* 22, 7 (2011), 1214–1221.

[29] Yinhao Jiang, Willy Susilo, Yi Mu, and Fuchun Guo. 2018. Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing. *Fut. Gen. Comput. Syst.* 78 (2018), 720–729.

[30] Yinhao Jiang, Willy Susilo, Yi Mu, and Fuchun Guo. 2018. Ciphertext-policy attribute-based encryption supporting access policy update and its extension with preserved attributes. *Int. J. Inf. Sec.* 17, 5 (2018), 533–548.

[31] Taeho Jung, Xiang-Yang Li, Zhiguo Wan, and Meng Wan. 2015. Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. *IEEE Trans. Inf. Forens. Sec.* 10, 1 (2015), 190–199.

[32] Praveen Kumar, Kumar Syam, and P. J. A. Alphonse. 2018. Attribute based encryption in cloud computing: A survey, gap analysis, and future directions. *J. Netw. Comput. Applic.* 108 (2018), 37–52.

[33] Junzuo Lai, R. H. Deng, Chaowen Guan, and Jian Weng. 2013. Attribute-based encryption with verifiable outsourced decryption. *IEEE Trans. Inf. Forens. Sec.* 8, 8 (2013), 1343–1354.

[34] Junzuo Lai, Robert H. Deng, and Yingjiu Li. 2011. Fully secure cipertext-policy hiding CP-ABE. In *Proceedings of the 7th International Conference on Information Security Practice and Experience (ISPEC'11)*, Vol. 6672. Springer, 24–39.

[35] Junzuo Lai, Robert H. Deng, and Yingjiu Li. 2012. Expressive CP-ABE with partially hidden access structures. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'12)*. ACM, 18–19.

[36] Kwangsu Lee, Seung Geol Choi, Dong Hoon Lee, Jong Hwan Park, and Moti Yung. 2013. Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency. In *Proceedings of the Advances in Cryptology Conference (ASIACRYPT'13)*, Vol. 8269. Springer, 235–254.

[37] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. 2010. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Proceedings of the Advances in Cryptology Conference (EUROCRYPT'10)*, Vol. 6110. Springer, 62–91.

[38] Allison Lewko and Brent Waters. 2011. Decentralizing attribute-based encryption. In *Proceedings of the Advances in Cryptology Conference (EUROCRYPT'11)*, Vol. 6632. Springer, 568–588.

[39] Allison Lewko and Brent Waters. 2012. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *Proceedings of the Advances in Cryptology Conference (CRYPTO'12)*, Vol. 7417. Springer, 180–198.

[40] Jin Li, Xinyi Huang, Jingwei Li, Xiaofeng Chen, and Yang Xiang. 2014. Securely outsourcing attribute-based encryption with checkability. *IEEE Trans. Parallel Distrib. Syst.* 25, 8 (2014), 2201–2210.

[41] Jin Li, Kui Ren, Bo Zhu, and Zhiguo Wan. 2009. Privacy-aware attribute-based encryption with user accountability. In *Proceedings of the International Information Security Conference (ISC'09)*, Vol. 5735. Springer, 347–362.

[42] Jin Li, Qian Wang, Cong Wang, and Kui Ren. 2011. Enhancing attribute-based encryption with attribute hierarchy. *Mobile Netw. Applic.* 16, 5 (2011), 553–561.

[43] Jiguo Li, Yao Wang, Yichen Zhang, and Jinguang Han. 2017. Full verifiability for outsourced decryption in attribute based encryption. *IEEE Trans. Serv. Comput.* (2017). Retrieved from http://dx.doi.org/10.1109/TSC.2017.2710190.

[44] Jiguo Li, Wei Yao, Jinguang Han, Yichen Zhang, and Jian Shen. 2017. User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage. *IEEE Syst. J.* 12, 2 (2017), 1767–1777.

[45] Jiguo Li, Qihong Yu, and Yichen Zhang. 2019. Hierarchical attribute based encryption with continuous leakage-resilience. *Inf. Sci.* 484 (2019), 113–134.

[46] Jin Li, Yinghui Zhang, Xiaofeng Chen, and Yang Xiang. 2018. Secure attribute-based data sharing for resource-limited users in cloud computing. *Comput. Sec.* 72 (2018), 1–12.

[47] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou. 2012. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE Trans. Parallel Distrib. Syst.* 24, 1 (2012), 131–143.

[48] Qi Li, Jianfeng Ma, Rui Li, Ximeng Liu, Jinbo Xiong, and Danwei Chen. 2016. Secure, efficient and revocable multi-authority access control system in cloud storage. *Comput. Sec.* 59 (2016), 45–59.

[49] Kaitai Liang, Man Ho Au, Joseph K. Liu, Willy Susilo, Duncan S. Wong, Guomin Yang, Yong Yu, and Anjia Yang. 2015. A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. *Fut. Gen. Comput. Syst.* 52 (2015), 95–108.

[50] Xiaohui Liang, Zhenfu Cao, Huang Lin, and Jun Shao. 2009. Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS'09)*. ACM, 276–286.

[51] Xiaohui Liang, Zhenfu Cao, Huang Lin, and Dongsheng Xing. 2009. Provably secure and efficient bounded ciphertext policy attribute based encryption. In *Proceedings of the 4th ACM Symposium on Information, Computer and Communications Security (ASIACCS'09)*. ACM, 343–352.

[52] Huang Lin, Zhenfu Cao, Xiaohui Liang, and Jun Shao. 2008. Secure threshold multi authority attribute based encryption without a central authority. In *Proceedings of the International Conference on Cryptology in India (INDOCRYPT'08)*, Vol. 5365. Springer, 426–436.

[53] Suqing Lin, Rui Zhang, Hui Ma, and Mingsheng Wang. 2015. Revisiting attribute-based encryption with verifiable outsourced decryption. *IEEE Trans. Inf. Forens. Sec.* 10, 10 (2015), 2119–2130.

[54] Joseph K. Liu, Tsz Hon Yuen, Peng Zhang, and Kaitai Liang. 2018. Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list. In *Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS'18)*, Vol. 10892. Springer, 516–534.

[55] Qin Liu, Guojun Wang, and Jie Wu. 2014. Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. *Inf. Sci.* 258 (2014), 355–370.

[56] Zhen Liu, Zhenfu Cao, Qiong Huang, Duncan S. Wong, and Tsz Hon Yuen. 2011. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'11)*, Vol. 6879. Springer, 278–297.

[57] Zhen Liu, Zhenfu Cao, and Duncan S. Wong. 2012. White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures. *IEEE Trans. Inf. Forens. Sec.* 8, 1 (2012), 76–88.

[58] Zhen Liu, Zhenfu Cao, and Duncan S. Wong. 2013. Blackbox traceable CP-ABE: How to catch people leaking their keys by selling decryption devices on eBay. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS'13)*. ACM, 475–486.

[59] Zhenhua Liu, Shuhong Duan, Peilin Zhou, and Baocang Wang. 2017. Traceable-then-revocable ciphertext-policy attribute-based encryption scheme. *Fut. Gen. Comput. Syst.* 93 (2017), 903–913.

[60] Zechao Liu, Zoe L. Jiang, Xuan Wang, Xinyi Huang, Siu-Ming Yiu, and Kunihiko Sadakane. 2017. Offline/online attribute-based encryption with verifiable outsourced decryption. *Concurr. Comput. Pract. Exper.* 29, 7 (2017), 1–17.

[61] Zhen Liu and Duncan S. Wong. 2016. Practical attribute-based encryption: Traitor tracing, revocation, and large universe. *Comput. J.* 59, 7 (2016), 983–1004.

[62] Song Luo, Jianbin Hu, and Zhong Chen. 2010. Ciphertext policy attribute-based proxy re-encryption. In *Proceedings of the International Conference on Information and Communications Security (ICICS'10)*, Vol. 6476. Springer, 401–415.

[63] Hui Ma, Rui Zhang, Zhiguo Wan, Yao Lu, and Suqing Lin. 2015. Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing. *IEEE Trans. Depend. Sec. Comput.* 14, 6 (2015), 679–692.

[64] Hui Ma, Rui Zhang, Guomin Yang, Zishuai Song, Shuzhou Sun, and Yuting Xiao. 2018. Concessive online/offline attribute based encryption with cryptographic reverse firewalls—Secure and efficient fine-grained access control on corrupted machines. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'18)*, Vol. 11099. Springer, 507–526.

[65] Qutaibah M. Malluhi, Abdullatif Shikfa, and Viet Cuong Trinh. 2017. A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption. In *Proceedings of the 12th ACM on Asia Conference on Computer and Communications Security (ASIACCS'17)*. ACM, 230–240.

[66] Jianting Ning, Zhenfu Cao, Xiaolei Dong, Junqing Gong, and Jie Chen. 2016. Traceable CP-ABE with short ciphertexts: How to catch people selling decryption devices on eBay efficiently. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'16)*, Vol. 9879. Springer, 551–569.

[67] Jianting Ning, Zhenfu Cao, Xiaolei Dong, Kaitai Liang, Hui Ma, and Lifei Wei. 2017. Auditable σ-time outsourced attribute-based encryption for access control in cloud computing. *IEEE Trans. Inf. Forens. Sec.* 13, 1 (2017), 94–105.

[68] Jianting Ning, Xiaolei Dong, Zhenfu Cao, and Lifei Wei. 2015. Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'15)*, Vol. 9327. Springer, 270–289.

[69] Jianting Ning, Xiaolei Dong, Zhenfu Cao, Lifei Wei, and Xiaodong Lin. 2015. White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. *IEEE Trans. Inf. Forens. Sec.* 10, 6 (2015), 1274–1288.

[70] Takashi Nishide, Kazuki Yoneyama, and Kazuo Ohta. 2008. ABE with partially hidden encryptor-specified sccess structure. In *Proceedings of the Conference on Applied Cryptography and Network Security (ACNS'08)*, Vol. 5037. Springer, 111–129.

[71] Go Ohtake, Reihaneh Safavi-Naini, and Liang Feng Zhang. 2019. Outsourcing scheme of ABE encryption secure against malicious adversary. *Comput. Sec.* 86 (2019), 437–452.

[72] Tatsuaki Okamoto and Katsuyuki Takashima. 2010. Fully secure functional encryption with general relations from the decisional linear assumption. In *Proceedings of the Advances in Cryptology Conference (CRYPTO'10)*, Vol. 6223. Springer, 191–208.

[73] Tran Viet Xuan Phuong, Guomin Yang, and Willy Susilo. 2016. Hidden ciphertext policy attribute-based encryption under standard assumptions. *IEEE Trans. Inf. Forens. Sec.* 11, 1 (2016), 35–45.

[74] Tran Viet Xuan Phuong, Guomin Yang, Willy Susilo, and Xiaofeng Chen. 2015. Attribute based broadcast encryption with short ciphertext and decryption key. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'15)*, Vol. 9327. Springer, 252–269.

[75] Baodong Qin, Robert H. Deng, Shengli Liu, and Siqi Ma. 2015. Attribute-based encryption with efficient verifiable outsourced decryption. *IEEE Trans. Inf. Forens. Sec.* 10, 7 (2015), 1384–1393.

[76] Baodong Qin, Qinglan Zhao, Dong Zheng, and Hui Cui. 2019. (Dual) Server-aided revocable attribute-based encryption with decryption key exposure resistance. *Inf. Sci.* 490 (2019), 74–92.

[77] Shuo Qiu, Jiqiang Liu, Yanfeng Shi, and Rui Zhang. 2017. Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack. *Sci. China Inf. Sci.* 60, 5 (2017), 1–12.

[78] Yannis Rouselakis and Brent Waters. 2013. Practical constructions and new proof methods for large universe attribute-based encryption. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS'13)*. ACM, 463–474.

[79] Yannis Rouselakis and Brent Waters. 2015. Efficient statically-secure large-universe multi-authority attribute-based encryption. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC'15)*, Vol. 8975. Springer, 315–332.

[80] Amit Sahai, Hakan Seyalioglu, and Brent Waters. 2012. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *Proceedings of the Advances in Cryptology Conference (CRYPTO'12)*, Vol. 7417. Springer, 199–217.

[81] Amit Sahai and Brent Waters. 2005. Fuzzy identity-based encryption. In *Proceedings of the Advances in Cryptology Conference (EUROCRYPT'05)*, Vol. 3494. Springer, 457–473.

[82] Michael Scott. 2005. Computing the Tate pairing. In *Proceedings of the Cryptographers' Track at the RSA Conference (CT-RSA'05)*, Vol. 3376. Springer, 293–304.
[83] Victor Shoup. 1997. Lower bounds for discrete logarithms and related problems. In *Proceedings of the Advances in Cryptology Conference (EUROCRYPT'97)*, Vol. 1233. Springer, 256–266.
[84] Mehdi Sookhak, F. Richard Yu, Muhammad Khurram Khan, Yang Xiang, and Rajkumar Buyya. 2017. Attribute-based data access control in mobile cloud computing: Taxonomy and open issues. *Fut. Gen. Comput. Syst.* 72 (2017), 273–287.
[85] Willy Susilo, Guomin Yang, Fuchun Guo, and Qiong Huang. 2018. Constant-size ciphertexts in threshold attribute-based encryption without dummy attributes. *Inf. Sci.* 429 (2018), 349–360.
[86] TechFunnel. 2018. Top 5 cloud computing predictions for 2020. Retrieved from https://www.techfunnel.com/information-technology/top-5-cloud-computing-predictions-for-2020.
[87] Wei Teng, Geng Yang, Yang Xiang, Ting Zhang, and Dongyang Wang. 2017. Attribute-based access control with constant-size ciphertext in cloud computing. *IEEE Trans. Cloud Comput.* 5, 4 (2017), 617–627.
[88] Zhiguo Wan, Jun'e Liu, and Robert H. Deng. 2012. HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE Trans. Inf. Forens. Sec.* 7, 2 (2012), 743–754.
[89] Guojun Wang, Qin Liu, Jie Wu, and Minyi Guo. 2011. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *Comput. Sec.* 30, 5 (2011), 320–331.
[90] Shulan Wang, Junwei Zhou, Joseph K. Liu, Jianping Yu, Jianyong Chen, and Weixin Xie. 2016. An efficient file hierarchy attribute-based encryption scheme in cloud computing. *IEEE Trans. Inf. Forens. Sec.* 11, 6 (2016), 1265–1277.
[91] Zhijie Wang, Dijiang Huang, Yan Zhu, Bing Li, and Chun-Jen Chung. 2015. Efficient attribute-based comparable data access control. *IEEE Trans. Comput.* 64, 12 (2015), 3430–3443.
[92] Brent Waters. 2009. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *Proceedings of the Advances in Cryptology Conference (CRYPTO'09)*, Vol. 5677. Springer, 619–636.
[93] Brent Waters. 2011. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proceedings of the Public-Key Cryptography Conference (PKC'11)*, Vol. 6571. Springer, 53–70.
[94] Hu Xiong, Yanan Zhao, Li Peng, Hao Zhang, and Kuo-Hui Yeh. 2019. Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing. *Fut. Gen. Comput. Syst.* 97 (2019), 453–461.
[95] Shengmin Xu, Guomin Yang, and Yi Mu. 2019. Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation. *Inf. Sci.* 479 (2019), 116–134.
[96] Shengmin Xu, Guomin Yang, Yi Mu, and Robert H. Deng. 2018. Secure fine-grained access control and data sharing for dynamic groups in the cloud. *IEEE Trans. Inf. Forens. Sec.* 13, 8 (2018), 2101–2113.
[97] Kaiping Xue, Jianan Hong, Yingjie Xue, David S. L. Wei, Nenghai Yu, and Peilin Hong. 2017. CABE: A new comparable attribute-based encryption construction with 0-encoding and 1-encoding. *IEEE Trans. Comput.* 66, 9 (2017), 1491–1503.
[98] Kaiping Xue, Yingjie Xue, Jianan Hong, Wei Li, Hao Yue, David S. L. Wei, and Peilin Hong. 2017. RAAC: Robust and auditable access control with multiple attribute authorities for public cloud storage. *IEEE Trans. Inf. Forens. Sec.* 12, 4 (2017), 953–967.
[99] Kan Yang, Xiaohua Jia, and Kui Ren. 2013. Attribute-based fine-grained access control with efficient revocation in cloud storage systems. In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS'13)*. ACM, 523–528.
[100] Yanjiang Yang, Joseph K. Liu, Kaitai Liang, Kim-Kwang Raymond Choo, and Jianying Zhou. 2015. Extended proxy-assisted approach: Achieving revocable fine-grained encryption of cloud data. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'15)*, Vol. 9327. Springer, 146–166.
[101] Yanjiang Yang, Haiyan Zhu, Haibing Lu, Jian Weng, Youcheng Zhang, and Kim-Kwang Raymond Choo. 2016. Cloud based data sharing with fine-grained proxy re-encryption. *Pervas. Mobile Comput.* 28 (2016), 122–134.
[102] Lo-Yao Yeh, Pei-Yu Chiang, Yi-Lang Tsai, and Jiun-Long Huang. 2018. Cloud-based fine-grained health information access control framework for lightweight IoT devices with dynamic auditing and attribute revocation. *IEEE Trans. Cloud Comput.* 6, 2 (2018), 532–544.
[103] Ping Yu, Qiaoyan Wen, Wei Ni, Wenmin Li, Caijun Sun, Hua Zhang, and Zhengping Jin. 2019. Decentralized, revocable and verifiable attribute-based encryption in hybrid cloud system. *Wirel. Person. Commun.* 106 (2019), 719–738.
[104] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. 2010. Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*. ACM, 261–270.
[105] Jiang Zhang, Zhenfeng Zhang, and Aijun Ge. 2012. Ciphertext policy attribute-based encryption from lattices. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'12)*. ACM, 16–17.

[106] Kai Zhang, Hui Li, Jianfeng Ma, and Ximeng Liu. 2018. Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability. *Sci. China Inf. Sci.* 61, 3 (2018), 032102.
[107] Leyou Zhang, Gongcheng Hu, Yi Mu, and Fatemeh Rezaeibagha. 2019. Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system. *IEEE Access* 7 (2019), 33202–33213.
[108] Rui Zhang, Hui Ma, and Yao Lu. 2017. Fine-grained access control system based on fully outsourced attribute-based encryption. *J. Syst. Softw.* 125 (2017), 344–353.
[109] Yinghui Zhang, Xiaofeng Chen, Jin Li, Hui Li, and Fenghua Li. 2014. Attribute-based data sharing with flexible and direct revocation in cloud computing. *KSII Trans. Internet Inf. Syst.* 8, 11 (2014), 4028–4049.
[110] Yinghui Zhang, Xiaofeng Chen, Jin Li, Duncan S. Wong, and Hui Li. 2013. Anonymous attribute-based encryption supporting efficient decryption test. In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS'13)*. ACM, 511–516.
[111] Yinghui Zhang, Xiaofeng Chen, Jin Li, Duncan S. Wong, Hui Li, and Ilsun You. 2017. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. *Inf. Sci.* 379 (2017), 42–61.
[112] Yinghui Zhang, Jin Li, Xiaofeng Chen, and Hui Li. 2016. Anonymous attribute-based proxy re-encryption for access control in cloud computing. *Sec. Commun. Netw.* 9, 14 (2016), 2397–2411.
[113] Yinghui Zhang, Jin Li, Dong Zheng, Xiaofeng Chen, and Hui Li. 2017. Towards privacy protection and malicious behavior traceability in smart health. *Person. Ubiq. Comput.* 21, 5 (2017), 815–830.
[114] Yinghui Zhang, Axin Wu, and Dong Zheng. 2018. Efficient and privacy-aware attribute-based data sharing in mobile cloud computing. *J. Amb. Intell. Human. Comput.* 9, 4 (2018), 1039–1048.
[115] Yinghui Zhang, Dong Zheng, Xiaofeng Chen, Jin Li, and Hui Li. 2014. Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts. In *Proceedings of the Provable Security Conference (ProvSec'14)*, Vol. 8782. Springer, 259–273.
[116] Yinghui Zhang, Dong Zheng, Xiaofeng Chen, Jin Li, and Hui Li. 2016. Efficient attribute-based data sharing in mobile clouds. *Pervas. Mobile Comput.* 28 (2016), 135–149.
[117] Yinghui Zhang, Dong Zheng, and Robert H. Deng. 2018. Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet Things J.* 5, 3 (2018), 2130–2145.
[118] Yinghui Zhang, Dong Zheng, Qi Li, Jin Li, and Hui Li. 2016. Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing. *Sec. Commun. Netw.* 9, 16 (2016), 3688–3702.
[119] Zhibin Zhou, Dijiang Huang, and Zhijie Wang Lou. 2015. Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption. *IEEE Trans. Comput.* 64, 1 (2015), 126–138.

Received September 2019; revised April 2020; accepted May 2020