Prantl *et al. J Surveill Secur Saf* 2023;4:129-54
DOI: 10.20517/jsss.2023.30

Journal of Surveillance, Security and Safety

Survey

Open Access

Check for updates

Towards a cryptography encyclopedia: a survey on attribute-based encryption

Thomas Prantl[1], Timo Zeck[1], Lukas Horn[1], Lukas Iffländer[1], André Bauer[2], Alexandra Dmitrienko[1], Christian Krupitzer[3], Samuel Kounev[1]

[1]Computer Science Department, Julius-Maximilians-Universität Würzburg, Würzburg 97074, Germany.
[2]Computer Science Department, University of Chicago, Chicago, State 60637 Illinois, USA.
[3]Food Informatics Department, University of Hohenheim, Stuttgart 70599 , Baden-Wurttemberg, Germany.

Correspondence to: Thomas Prantl, Computer Science Department, Julius-Maximilians-Universität Würzburg, Würzburg 97074, Germany. E-mail: thomas.prantl@uni-wuerzburg.de

How to cite this article: Prantl T, Zeck T, Horn L, Iffländer L, Bauer A, Dmitrienko A, Krupitzer C, Kounev S. Towards a cryptography encyclopedia: a survey on attribute-based encryption. *J Surveill Secur Saf* 2023;4:129-54. http://dx.doi.org/10.20517/jsss.2023.30

Received: 8 Sep 2023 **First Decision:** 24 Oct 2023 **Revised:** 26 Oct 2023 **Accepted:** 20 Nov 2023 **Published:** 13 Dec 2023

Academic Editor: Moti Yung **Copy Editor:** Dan Zhang **Production Editor:** Dan Zhang

Abstract

Access control, a key feature of digital business models, such as streaming, relies on the implementation of encryption schemes. The diverse use of encryption schemes has led to the development of schemes with a variety of properties. This variety and a lack of comprehensive overview make it difficult for developers to select an appropriate scheme. To bridge this gap, we envision a cryptography encyclopedia. In this survey, we create a sub-encyclopedia for attribute-based encryption (ABE) schemes. More specifically, we provide an overview of relevant features and performance metrics and a taxonomy for ABE schemes. We also perform a performance and feature evaluation of 42 ABE schemes and apply our proposed topology to these approaches.

Keywords: Attribute-based encryption, features, performance, survey

1. INTRODUCTION

With the emergence of rapid network technologies, the speed of the Internet and the level of connectivity has been significantly increased[1]. Applications capable of handling multiple units are becoming increasingly important as electronic communication and information services have evolved in recent years[2]. For example,

new online platforms, such as Steam or Battlenet, offering video games in digital forms are on the rise and have millions of customers ([3] and [4]). Similar platforms also exist in the film and music industry, for example, where Netflix or Spotify make films and music digitally accessible to their millions of users ([5] and [6]). The basic requirement for these digital business models is that the providers can always control who has paid for which content and is, therefore, allowed to access it. In practice, such access control can be implemented with the help of encryption schemes. In addition to the ever-increasing number of encryption use cases with their different requirements, the range of encryption schemes and the features they offer have also increased over the years. For example, in addition to encrypting the communication between two parties (1-to-1 communication), modern encryption schemes are also capable of encrypting the communication among more than two parties. In scenarios with multiple parties, the schemes differ in the number of parties able to encrypt messages. There are typically two options: either only one party is able to encrypt a message for several other parties (1-to-n communication) or each party is able to encrypt messages for all other parties (n-to-n communication). For each of these three categories of encryption schemes, there are, in turn, numerous realizations in the literature, which differ in terms of their features and requirements. For example, in the domain of n-to-n encryption schemes, there are schemes that require the presence of a trusted third party (e.g., [7] and [8]), while other schemes can operate independently (e.g., [9] and [10]). This multitude of encryption schemes with a wide variety of properties poses a challenge for developers. It is hard to select the optimal scheme for a specific use case that fulfills the necessary properties while maintaining optimal efficiency. There are currently no guidelines available for comparing different schemes against each other. Furthermore, there lacks an overview detailing the relevant features that can be used as distinguishing factors or evaluating the performance of these various schemes. For developers, such guidelines and respective overviews are essential in order to be able to select an encryption scheme for a given use case. For this reason, we envision a cryptography encyclopedia that provides an overview of existing schemes, including their theoretical features and performance. In this survey paper, we make a move towards this vision by creating a sub-encyclopedia for attribute-based encryption (ABE) schemes belonging to the 1-to-n category.

More specifically, this survey provides the following contributions:

- A comprehensive overview of the relevant features of ABE schemes that can be used as distinguishing factors,
- A set of relevant performance metrics for ABE schemes,
- A taxonomy for ABE schemes procedures,
- An evaluation of 42 attribute-based schemes with respect to the distinguishing features, the performance metrics, and the proposed taxonomy.

The remainder of this paper is structured as follows. In Section 2, we provide background information required for understanding and comparing ABE schemes. In Section 3, we present the methodology used for our survey. In doing so, we show our proposed comparison criteria and taxonomy for ABE schemes and highlight the novelty of our contribution by comparing it with related work. In Sections 4 - 9, each ABE (sub-)category, from our taxonomy, is presented in more detail. A comparison of the different categories is presented in Section 10. The paper is concluded in Section 11.

2. BACKGROUND

In this section, we provide background information required for understanding and comparing ABE schemes. Section 2.1 defines the different complexity assumptions the security of the ABE schemes is based on. In Section 2.2, we explain the term access policy that is a main part of every ABE scheme. Building on this, we introduce ABE schemes in general in Section 2.3, followed by the two main categories of ABE schemes, KP-ABE and CP-ABE in the Sections 2.4 and 2.5. A conceptual comparison of the two main categories is presented in Section 2.6.

2.1. Complexity assumptions

Bilinear maps are believed to be the mainstream solution for key generation in ABE schemes. A bilinear map is an efficient, precise, and secure method for key generation in polynomial time, while other popular cryptographic systems, such as RSA or Diffie-Hellman, are hard to transform into an ABE system [11]. A bilinear map is defined as follows:

Definition 1 *Bilinear Map: Let G_1, G_2, and G_T be cyclic groups with the same large prime order q. According to [11], bilinear maps used specifically in identity-based encryption (IBE) and ABE systems fulfill the following three properties.*

- *Bilinearity: $(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ with g_1 and g_2 being the generators of G_1 and G_2, respectively, and $a, b \in \mathbb{Z}$. In the case of $G_1 = G_2$, the bilinear map is called symmetric.*
- *Computability: For any pair given by $G_1 \times G_2$, a bilinear map e has to be efficiently computable.*
- *Non-degeneracy: The bilinear map does not map all pairs in $G_1 \times G_2$ to the identity in G_T: $e(g_1, g_2) \neq 1$.*

The security of bilinear maps, and thus of ABE schemes in general, is based on the bilinear Diffie-Hellman complexity assumption, abbreviated as BDH, that describes the problem of computing the bilinear pair $e(g, g)^{abc}$ for any given tuple $\{(g, g^a, g^b, g^c)\}$ with $a, b, c \in \mathbb{Z}$ [11]. More specifically, the security of many ABE schemes is based on the decisional BDH (DBDH) and the decisional modified BDH assumption (MBDH) [12]. These are defined in the following:

Definition 2 *Decisional Bilinear Diffie-Hellman assumption (DBDH): Assume that a challenger chooses $a, b, c, z \in \mathbb{Z}$ at random. The DBDH assumption is that no adversary can distinguish the tuple (1) $(A = g^a, B = g^b, C = g^c, Z = e(g, g)^{abc})$ from the tuple (2) $(A = g^a, B = g^b, C = g^c, Z = e(g, g)^z)$ in polynomial time with a non-negligible advantage.*

Definition 3 *Decisional Modified Bilinear Diffie-Hellman assumption (MBDH): Suppose an adversary chooses $a, b, c, z \in \mathbb{Z}$ at random. The MBDH assumption specifies that no adversary is able to distinguish the tuple (1) $(A = g^a, B = g^b, C = g^c, Z = e(g, g)^{ab/c})$ from the tuple (2) $(A = g^a, B = g^b, C = g^c, Z = e(g, g)^z)$ in polynomial time with a non-negligible advantage.*

2.2. Access policy

Generally, access control is a procedure that restricts, denies, or allows access to resources. In the context of ABE, the term access policy can be described as a control structure defining if an entity has the permission to decrypt certain data [13]. In the case of ABE schemes that are often used in secure cloud environments, the access policy controls which users can decrypt certain data stored in the cloud by data owners (DOs) [13]. As an illustrative example, imagine the following academic situation: The performance record of students in a course CS must be only accessible by the respective professor or teaching assistants of the course. This can be expressed by the following access policy in the form of a boolean formula: ((Role: *Professor* AND Course: CS) OR (Role: *Teaching-Assistant* AND Course: CS)). The variables in this predicate, such as 'Professor' or 'CS', are called attributes; the predicate or policy itself is also called an access structure [13].

In addition to this representation as a Boolean formula, access policies are often represented as trees. The leaves of such a tree represent the used attributes. The remaining leaves represent gates, such as AND or OR. If the gate of the root node is fulfilled, the access is allowed, or the decryption of the ciphertext is possible [13]. As an illustration of the concept of the access policy by a tree, Figure 1 illustrates such an access policy tree for the university example described above.

The access policy in ABE schemes can either be encoded into the secret key of data users (DUs) or it can be specified by the ciphertext. In existing ABE schemes, different kinds of access policies are used, such as

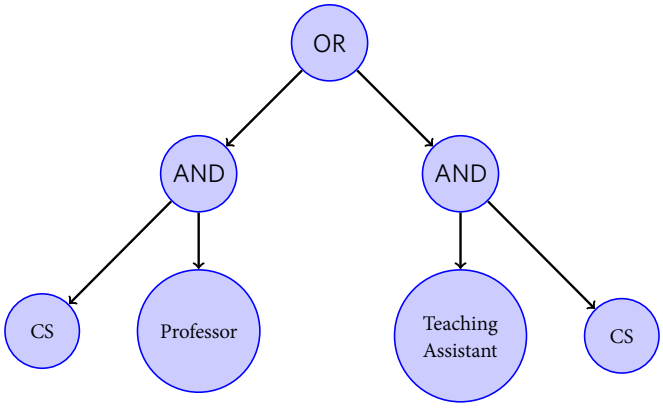


Figure 1. Example representation of an access policy as a tree, according to Priya *et al.*[13].

threshold policies, AND policies, tree policies, and the linear secret sharing scheme (LSSS) matrix. Waters[14] proposed this last methodology of access control for Ciphertext-Policy ABE (CP-ABE) only in 2011. It uses a so-called LSSS matrix over the attributes in the system, allowing for the concise expression of the previously used structures in the form of an LSSS[12]. According to Zhang *et al.*[15], schemes using LSSS access structures generally tend to be more efficient while maintaining expressiveness compared to ABE schemes based on other kinds of policies[15]. The access policy can also support either only positive attributes or both positive and negative attributes. In the first case, the access structure is then called monotonic; in the second case, it is called non-monotonic[16].

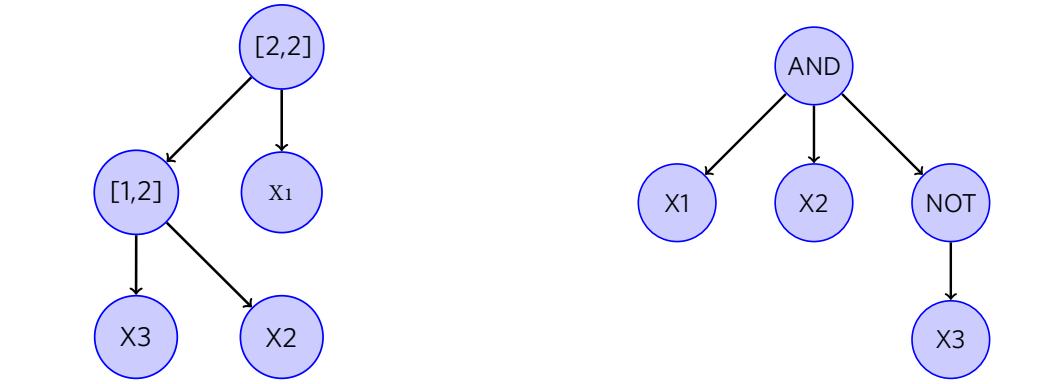
Threshold access policies are used in various ABE schemes, such as the first Key-Policy ABE (KP-ABE) scheme by Goyal *et al.*[17]. A tree can act as a representation of this access policy. In such a tree, the leaves represent attributes while the other nodes (root and intermediate) consist of threshold gates of the form $[m, n]$. Here, m is the threshold value of the number of attributes to be fulfilled for satisfying the gate. The total number of attributes following the gate is represented with n. The threshold gate $[1, n]$ represents the OR gate, and $[n, n]$ represents the AND-gate[18]. The access policy is fulfilled when the root of the tree is satisfied, allowing a user to decrypt the data.

Figure 2a shows an example of such a threshold gate tree structure with the attributes X1, X2, and X3 for the access policy (X1 AND (X2 OR X3)). Here, the gate [1,2] is an OR gate that is satisfied if one or more of the attributes X2 and X3 following are fulfilled. The gate [2,2] represents an AND-gate, accordingly.

The access structure in the KP-ABE scheme by Goyal *et al.*[17] is a so-called monotonic access structure. In such an access policy, a negative attribute cannot be represented. It can be used, for example, to exclude users to whom the owner does not want to disclose data. Ostrovsky *et al.*[19] proposed the first ABE scheme with non-monotonic access structures. This scheme supports both positive and negative attributes. Consequently, such non-monotonic access structures are able to express more complicated access policies. However, non-monotonic access structures introduce additional overhead for encryption and decryption and increase the size of ciphertext and keys when compared to schemes with monotonic policies[12]. This scheme is only reasonably applicable for a fixed number of attributes and for a limited number of users. Figure 2b shows an example of such a non-monotonic access structure (X1 AND X2 AND NOT X3) with the attributes X1, X2, and X3.

2.3. ABE schemes

In this section, we introduce the fundamental concept of ABE and its two sub-categories, namely KP-ABE and CP-ABE[16]. We also describe the different kinds of access structures used in ABE schemes.



(a) Threshold access structure[18]. **(b)** Non-monotonic access structure[18].

Figure 2. Illustration of threshold and non-monotonic access structure.

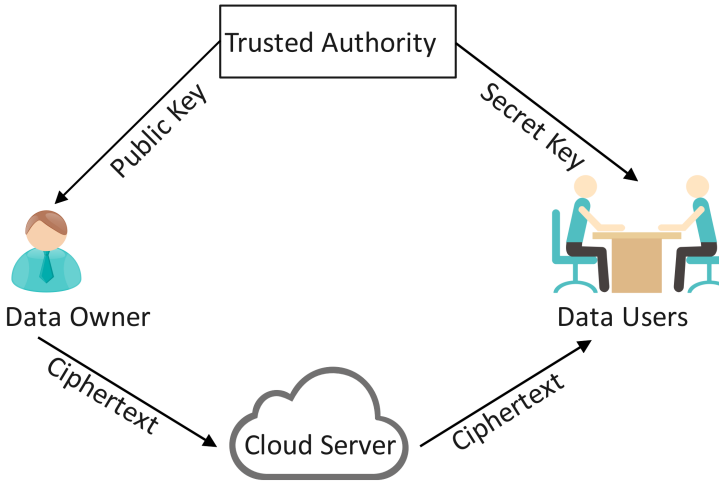


Figure 3. Architecture of the basic Attribute-Based Encryption (in the style of Kumar *et al*.[18]).

In 2005, Sahai and Waters[20] proposed the first ABE scheme, called FIBE, a simplification of IBE[13]. ABE is a one-to-many algorithm using public key cryptography for protecting data in the cloud. The special property of ABE is that the ciphertext and the secret keys of the users are based on a set of attributes used for controlling whether a specific user is allowed to decrypt the data[15]. This basic ABE scheme involves three different entities, namely an authority, a DO, and a DU[18]. Generating the keys according to attribute sets that are necessary for the DOs and users is the responsibility of the authority[16]. The DO receives a public key for encrypting the data along with the attributes before storing it in the cloud. DUs obtain a secret key, also called a private key, from the authority, and they can use this secret key to decrypt data from the cloud[18]. In the ABE scheme by Sahai and Waters, the decryption is only possible in case a threshold number of d components of the attributes in the ciphertext are equal to the attributes in the user secret key[16]. The architecture of this basic ABE scheme is presented on an abstract, high level in Figure 3, similar to the graphics in[18].

As a promising cryptographic primitive for a more secure cloud, research efforts on attribute-based encryption have significantly increased over the last few years. In FIBE, the first and basic ABE scheme, the threshold is the only access structure supported. During the setup phase, it gets fixed by the authority. Nevertheless, many practical applications have increasing needs for flexible access control policies that support operations such as 'and', 'or', 'threshold', and 'non'[12]. Therefore, more complex types of ABE schemes were proposed. Since their

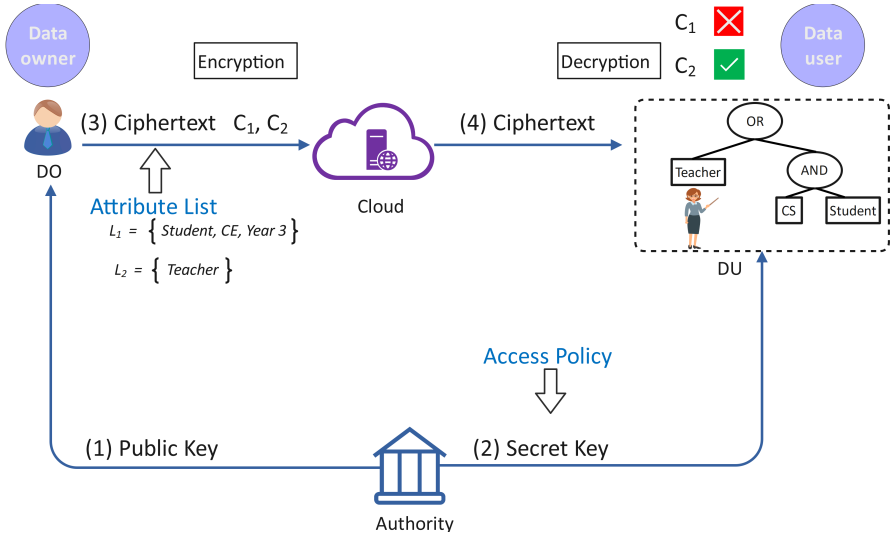


Figure 4. The general concept of KP-ABE (in the style of Zhang *et al.*[18]).

emergence in the year 2005, numerous ABE schemes have been proposed, including new variations of basic ABE focusing on different properties[15]. These schemes can generally be classified into the two categories: CP-ABE and KP-ABE[11–13,15,16,18].

2.4. KP-ABE

In 2006, Goyal *et al.*[17] proposed the first KP-ABE scheme extending basic ABE. The scheme was intended for fine-grained sharing of encrypted data and has been proven secure under the DBDH assumption. Generally, in KP-ABE schemes, the access policy is encoded into the secret keys of the user while the ciphertexts are created using a set of attributes. A ciphertext can only successfully be decrypted by a user if the attributes of the encrypted data fulfill the access policy embedded into the user's secret key[13].

For instance, let us assume an academic situation with a user possessing a secret key with the following access policy encoded into it: ((Identity: *Student* AND Department: *CS*) OR Identity: *Teacher*). In this case, the user is not permitted to decrypt a ciphertext generated based on the attribute set {Identity: *Student*, Department: *CE*, Year: *3*}. However, the user would be able to decrypt a ciphertext with the attribute set {Identity: *Teacher*}[15]. Figure 4 illustrates this concept of KP-ABE. A trusted authority publishes the public key (1) and generates the secret keys with respect to the user's access policy (2). The DO chooses an attribute list L and integrates L into the ciphertext (3). The user is able to decrypt and access the data stored in the cloud based on the access policy that is embedded into his secret key (4). The fact that the access policy is integrated into the secret keys of the users is a major drawback of KP-ABE schemes. As a consequence, the DO cannot specify who can decrypt the ciphertext. Instead, they can only influence this by selecting a set of attributes for it[12].

2.5. CP-ABE

In 2007, Bethencourt *et al.*[21] proposed the first CP-ABE scheme using a monotonic threshold gate access structure. In contrast to KP-ABE schemes, in CP-ABE, the secret keys of users are associated with a set of attributes instead of an access policy. This association is integrated into the ciphertext rather than into the user's secret key in KP-ABE[11].

Figure 5 illustrates the general concept of CP-ABE using a typical application example. It includes four entities, namely a trusted authority, the DU, the DO, and a cloud where the data is stored. The trusted authority publishes the public key (1), which the DO uses to encrypt their data, and distributes the corresponding secret keys to each user (2). Hereby, the attribute list of a DU is integrated into his secret key. After selecting an

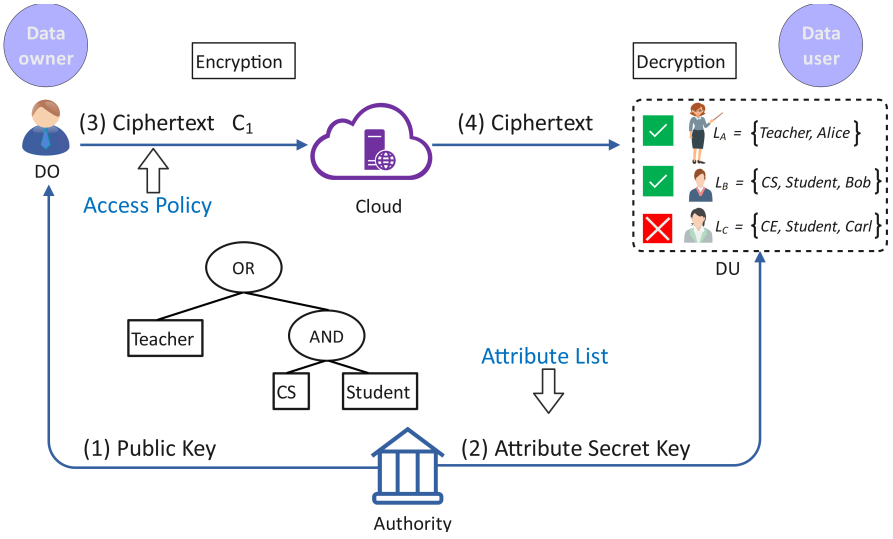


Figure 5. Concept of CP-ABE (in the style of Zhang *et al.*[15]).

access policy for his data, the DO embeds it into the ciphertext of the corresponding data encrypted with the public key (3). DUs can retrieve the encrypted data from the cloud. In order to access the original data, they need to decrypt the ciphertext using their secret keys. However, successful decryption is only possible if the attributes integrated into the secret key match the one inside the ciphertext (4).

For example, assume three DUs, with the respective attribute lists $L_A = \{Teacher, Alice\}$, $L_B = \{CS, Student, Bob\}$, and $L_C = \{CE, Student, Carl\}$. For a ciphertext encrypted with the policy (Teacher OR (CS AND Student)), Alice and Bob can successfully access the original data while Carl is not able to decrypt the ciphertext.

2.6. General Comparison of KP- and CP-ABE

KP-ABE and CP-ABE schemes work in an opposite manner in terms of the integration of access policy and attributes. As a result, these schemes are generally suitable for different kinds of applications due to differences in the flexibility and complexity assumptions[12].

Both KP- and CP-ABE can support complex strategies for access structures. Therefore, they are suitable for applications that require fine-grained control of data sharing. However, in KP-ABE schemes, the access policy is integrated into the secret keys of the users. Consequently, DOs have no control over which DUs can actually access and decrypt their data. In comparison, CP-ABE is generally more applicable for most real-world cloud data sharing applications, in which the DOs have full control of their data[13]. KP-ABE schemes are rather suitable for query applications, such as pay-TV subscriptions, access to databases, or targeted broadcasts. On the other hand, CP-ABE schemes are often used in applications that require access control, for example, access to social networking user profiles or electronic medical systems[12].

KP-ABE schemes are usually proven secure under the standard DBDH complexity assumption described in Section 2.1 . Since CP-ABE schemes generally tend to be more complex than KP-ABE schemes, it is also more difficult to prove their security. In order to achieve chosen-plaintext attack (CPA) security under the standard complexity assumption in CP-ABE, the focus of research is on the design of suitable access policies. We can differentiate between three types of access structures that are often used in CP-ABE schemes, namely AND-gate, access tree, and LSSS matrix, as described in Section 2.2 .

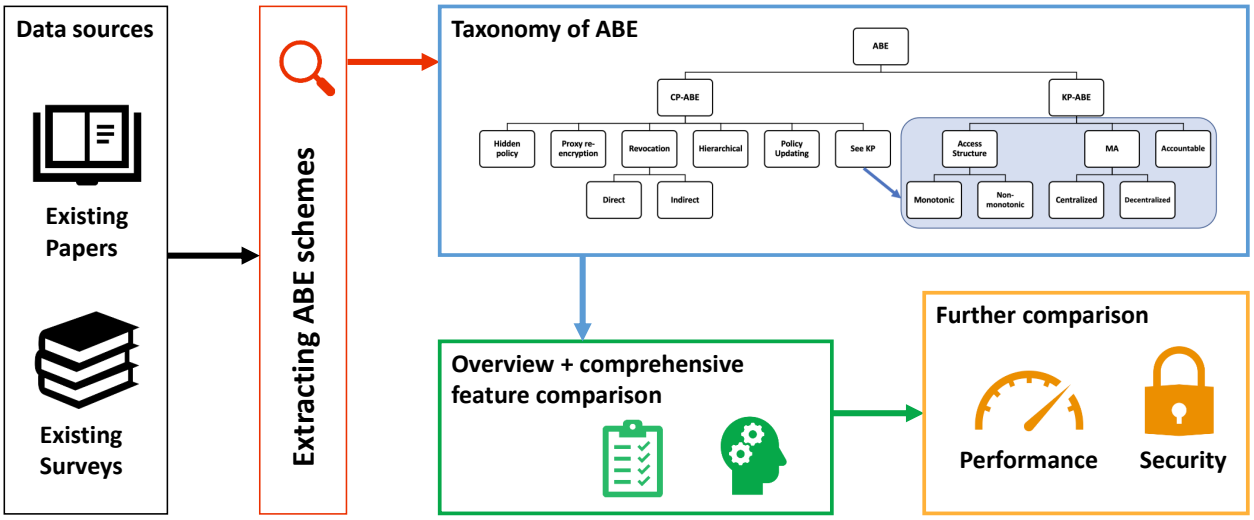


Figure 6. Survey methodology.

3. SURVEY METHODOLOGY

In Section 3.1 , we first provide an overview of the methodology used for our survey. In Sections 3.2 and 3.3, we go into more detail about the taxonomy and evaluation criteria for ABE schemes. A distinction from related work is made in Section 3.4 , which also highlights the novelty of our survey.

3.1. Methodology

The methodology used for our survey is illustrated in Figure 6. In order to gain an overview of existing publications on ABE schemes, we initially chose the existing ABE surveys from sources such as [12], [18], [16], [13], and [11]. We gradually increased this initial dataset by iteratively applying Forward Snowballing as long as we could find new publications on ABE schemes. From the total number of publications identified in this way, we then selected the schemes we included in our survey. More specifically, our inclusion criteria involved considering a scheme that (1) proposed a new ABE scheme or an extension of an existing one not previously examined by us and (2) that was proposed after 2014. The exclusion criterion was based on the obsolescence of a scheme since newer schemes often offer the same or more features. This resulted in 42 ABE schemes meeting our criteria. We then created a taxonomy for these 42 schemes and determined comparison metrics and features. We go into more detail on the taxonomy and comparison metrics in Sections 3.2 and 3.3 below, and we distinguish them from related work in Section 3.4 .

3.2. Taxonomy of ABE

In order to review ABE schemes and conduct a systematic comparison, we first propose a taxonomy of ABE schemes. Figure 7 illustrates our proposed taxonomy for ABE. ABE schemes realize different properties in terms of cryptographic functionality. These properties have an impact on the computational efficiency, security, and the formulation of access policies or attribute secret keys within a scheme. Our proposed taxonomy includes the following categories of ABE schemes according to these features. The taxonomy also allows us to identify combinations of features that are not yet fulfilled by existing systems. Thus, among the ABE schemes we have considered, there is still no KP-ABE scheme that offers the features such as Hidden Policy or Accountability. In the following, we describe the categories, omitting Access Structure since we have already presented it in Section II:

- Multi-Authority ABE: Contrary to traditional ABE, in which a single authority is able to decrypt all ciphertexts, in multi-authority (MA) ABE, there are multiple authorities operating simultaneously. The MA-

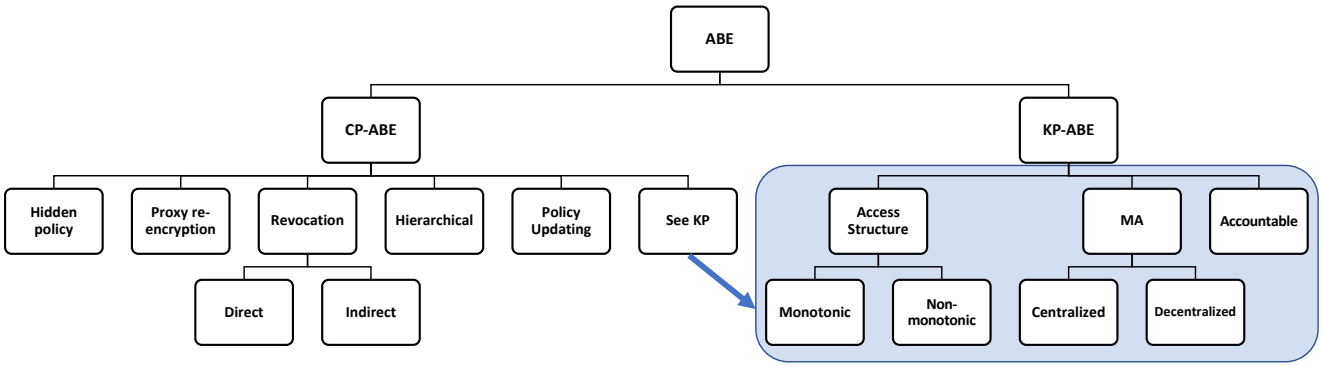


Figure 7. Proposed ABE taxonomy.

ABE category can be further divided into centralized and decentralized MA ABE, depending on whether a central authority (CA) is used or not. This property of multi-authorities can be applied to both KP- and CP-ABE[18].

- Accountable ABE: In order to improve the security of ABE and prevent the abuse of keys, schemes belonging to this category provide means to ensure accountability. Accountable ABE schemes specifically focus on two problems: the illegal sharing of keys among users and the misconduct of attribute authorities (AAs). Accountable ABEs implement both user accountability/tractability and the accountability of AAs. Accountability can be applied to both KP- and CP-ABE[12,15].
- Hidden Policy ABE: Hidden policy ABE schemes, also called policy-hiding ABE, aim to protect the privacy of access policies. This feature is only relevant to CP-ABE, where DOs include the access policy into the ciphertext sent to the cloud, as described in Section 2.5 . As a consequence, in such non-hidden-policy CP-ABE, anyone who is able to access this ciphertext can learn the access policy. The hidden policy category overcomes this problem by hiding the access structure[18]. This property of policy hiding can only be applied to CP-ABE [18].
- Proxy Re-encryption ABE: The concept of proxy re-encyption (PRE) aims to make data sharing more efficient. Proxy Re-encryption ABE schemes apply this PRE primitive to the attribute-based context. In this category, data encrypted with an access policy is re-encrypted by a proxy into a new ciphertext according to a new access policy. As a result, users fulfilling the new access policy are now able to decrypt the ciphertext without the proxy learning any useful information about the data. This feature is only relevant to CP-ABE[12].
- Policy Updating ABE: Traditional CP-ABE does not allow to dynamically change the access policy of a ciphertext. For access control applications that require such changes, for example, in case of emergencies, Policy Updating ABE provides mechanisms to update the access policy in a ciphertext[15]. KP-ABE schemes cannot update their policies, while some CP-ABE schemes are able to do this.
- Revocable ABE: Revocation mechanisms in ABE can be divided into user revocation and attribute revocation. The attribute revocation deals with dynamically editing the attributes of the users due to expiration, revocation, or addition of attributes. User revocation is used to remove or add users, for example, to deal with malicious behavior. Revocable ABE can be realized in two different ways, resulting in two subcategories, direct and indirect revocation[18]. User revocation is only possible with CP-ABE.
- Hierarchical ABE: In hierarchical ABE, the assignment of access rights is organized in a hierarchical manner. Only CP-ABE can provide this fine-grained access control; however, it is not optimal for large hierarchical structures, such as enterprises, due to the lack of a full delegation mechanism. Hierarchical ABE schemes provide scalability for such large enterprises by using a hierarchical structure for attributes[16].

3.3. Evaluation Criteria for ABE

In this section, we introduce and describe the metrics and features used in our systematic comparison of ABE schemes. We divide this comparison into the evaluation of the security features and the performance of schemes.

Security: ABE has two fundamental security features: Data confidentiality and collusion resistance. However, these features are usually achieved under different premises regarding the types of adversaries and the security models and the underlying complexity assumption involved in the security analysis. In consequence, there may be different ABE schemes fulfilling the same security goals, but they cannot be considered equally secure since the underlying conditions (e.g., security models) are different[15]. Therefore, we compare the security of ABE schemes according to their underlying complexity assumption and security model resulting from the types of adversaries.

- Security Model: The security model of an ABE scheme captures the specific types of adversaries against which the scheme is proven to be secure. Security proofs of ABE typically consider two types of adversaries, namely the selective adversary and the adaptive adversary. The selective adversary has to select its target access policy or attribute list in advance, which the adaptive adversary does not have to do. As a result, the adaptive adversary is more powerful. An ABE scheme can achieve selective or full security, depending on the type of the adversary it is proven secure against. This is referred to as the security model of the scheme. A full security proof is technically more challenging than a selective security proof. In the first case, the target access policy or attribute list is not pre-determined. Thus, an adversary in the simulation process is not able to set parameters in a targeted manner. A scheme that offers a fully secure model generally has a higher security than a scheme with selective security[13].
- Complexity Assumption: ABE schemes use and rely on complexity assumptions, so they are another important aspect of their security. The DBDH assumption is also called the standard complexity assumption, as it is used in many ABE schemes[11]. We describe the complexity assumptions in Section 2.1 .

Performance: In order to evaluate the efficiency of ABE schemes, we mainly refer to their storage, communication, and computation costs. We present the features reflecting these costs in the following:

- Public Key Size: The size of the public keys in ABE schemes is the first feature that influences its storage costs. In many cases, the size of the public keys increases linearly with the number of attributes in the system. However, public keys that are constant in size are preferable in an ideal scenario[12].
- Secret Key Size: The size of the corresponding secret keys in ABE schemes is the next feature relevant to the storage costs. Similar to the size of the public keys, the size of secret keys in many ABE schemes is often linearly proportional to the number of attributes in the system. Ideally, an ABE scheme should have secret keys of constant size[16].
- Ciphertext Size: The encrypted data stored in the cloud by DOs and retrieved by DUs, together with the communication costs, influence the size of a ciphertext in an ABE scheme. It is also linearly proportional to the complexity of the corresponding access policy[15].
- Encryption and Decryption Costs: The costs for encryption and decryption in ABE schemes are two major components of their computation costs. In IoT cloud scenarios, there may be small devices involved with limited resources, thus requiring low communication costs. The most expensive cryptographic operation regarding encryption and decryption in ABE is the bilinear pairing. In addition to this operation, we consider the two operations, point multiplication and exponentiation, in the evaluation of the computation costs. In comparison to these three operations, basic arithmetic operations, such as addition or multiplication, generally have less impact on the computation costs and are, therefore, typically ignored when evaluating computation costs in ABE[15].
- Access Policy: The access policy of an ABE scheme represents the expressiveness and granularity of the corresponding access control system. As described in Section 2.2 , existing ABE schemes can have different

Table 1. Overview of the different features/categories of ABE considered in existing surveys compared to the detailed feature comparison in our survey

	[22]	[12]	[18]	[16]	[13]	[11]	We
KP and CP	✓	✓	✓	✓	✓	✓	✓
Multi-authority	×	✓	✓	×	×	×	✓
Accountable	×	✓	×	×	×	×	✓
Hidden-policy	×	×	✓	×	×	×	✓
Proxy re-encryption	✓	✓	✓	×	×	×	✓
Policy update	✓	×	×	×	×	✓	
Revocable	✓	✓	✓	×	×	×	✓
Hierarchical	×	×	✓	✓	×	×	✓
Non-monotonic	✓	×	×	✓	×	✓	✓
Public key size	✓	✓	✓	×	×	×	✓
Secret key size	✓	✓	✓	✓	×	✓	✓
Ciphertext size	✓	✓	✓	✓	×	✓	✓
Computation costs	×	✓	✓	✓	general	✓	✓
Access policy	✓	✓	×	✓	×	×	✓
Security model	×	×	✓	×	✓	×	✓
Complexity assumption	×	✓	✓	×	✓	✓	✓
Comprehensive comparison	×	×	×	×	×	×	✓

types of access policies. Popular policies are the tree policy, the threshold policy, or the LSSS policy. When comparing LSSS-based schemes with those employing other access policies, the former typically achieves better efficiency. Additionally, while the ABE schemes based on the LSSS policy can generally achieve enhanced efficiency compared to other access policies, they also maintain an equivalent level of expressiveness as these alternative schemes [18].

- Scalability: The scalabilty of ABE schemes is an important aspect related to their basic structure and functionality. In order to achieve scalability, schemes should allow dynamic expansion of the system. Generally, scalable ABE schemes need to include some form of hierarchy in the structure of authorities in order to efficiently deal with large and hierarchical structures [11].

3.4. Differentiation from Related Work

Several surveys [11–13,16,18,22] have been published on ABE. They cover both KP-ABE and CP-ABE. We propose a refined taxonomy breaking down CP-ABE and KP-ABE into further categories, namely MA, accountable, hidden policy, proxy re-encryption, policy updating, revocable, hierarchical, and (non-)monotonic ABE. Pang *et al.* [12] only mentioned the categories MA, accountable, proxy re-encrypion, and revocable. Kumar et al. [18] investigated only MA, hidden-policy, revocable, and hierarchical ABE. The survey by Lee *et al.* [16] just contained the categories, hierarchical and non-monotonic. Priya *et al.* [13] mentioned neither of these categories, besides basic KP- and CP-ABE. Qiao *et al.* [11] only investigated the non-monotonic category. We further apply our taxonomy for a detailed feature comparison of ABE schemes. We are not aware of any existing survey providing such a detailed comparison of ABE schemes.

In addition to the feature comparison, we investigated current ABE schemes from all these categories and systematically compared them according to the discussed evaluation criteria. The criteria consider both the efficiency and security of the ABE schemes. For efficiency, we examine the storage costs represented by the size of the public and the secret keys, the communication costs in terms of the size of the ciphertext, and the computation costs represented by the costs for encryption and decryption. Moreover, we included the access policy of an ABE scheme as an additional criterion. Regarding security, we focused on the two fundamental properties of ABE schemes, namely the security model and the adopted complexity assumption. Most other surveys mentioned storage, communication, and computation costs. However, [11,13,16] did not examine the size of public keys in ABE schemes. Additionally, [13] lacked the features of secret key size and ciphertext

size and only generally investigated computation costs. Besides[12,16], no other survey mentioned the access policy of ABE schemes. Only[13,18] mentioned the security model as a property of ABE schemes. The adopted complexity assumption of an ABE scheme is mentioned in all surveys except for Lee et al[16]. However, again, we have a detailed comparison of the efficiency and security of a great number of ABE schemes that other existing surveys do not include.

Most surveys examined only a handful of schemes up to a maximum of ten. Kumar *et al.*[18] compared a larger number of schemes; however, they focused on older schemes and did not consider many important recent schemes. Additionally, they did not investigate every scheme they mentioned. In fact, in their comparison, they left out schemes for which they did not specify their achievement of features or performance regarding the costs or security. Altogether, we surveyed and compared a total of 42 different ABE schemes, resulting in a much broader coverage compared to the existing surveys. Moreover, similar to Kumar *et al.*[18], all existing surveys focus on older schemes proposed in 2014 or earlier. In contrast, our survey includes many recent schemes published in 2015 or later.

The survey by Rasori *et al.* [22] aimed to identify a selection of ABE schemes that are suitable for use in the IoT domain. For this purpose, schemes are first analyzed theoretically, and then, the performance of nine methods on typical IoT hardware is determined. However, no complete overview of the theoretical performance of the considered schemes is provided. Regarding features, however, the expressiveness of their access policies was given for all schemes.

In summary, we provide a systematization of existing knowledge on ABE in addition to a detailed comparison of existing ABE schemes. Table 1 provides an overview of the features and evaluation criteria used in our survey compared to the related work.

4. MULTI-AUTHORITY ABE

In contrast to traditional single-authority ABE, the MA ABE assumes multiple authorities operating simultaneously. The ability of the authority in single-authority ABE to decrypt all ciphertexts can be problematic in many scenarios. In order to overcome this security issue, MA ABE schemes have been proposed. Such schemes allow multiple authorities to exist in an ABE scheme, which, in turn, manage different attributes. Thus, access policies can be generated, which require the attributes managed by more than one authority, making it impossible for a single authority to decrypt all ciphertexts. The MA-ABE category can be further divided into centralized ([23–26]) and decentralized ([27–30]) MA ABE, depending on whether a CA is used or not. This property of MAs can be applied to both KP- and CP-ABE[12].

4.1. Centralized

Chase *et al.*[23] proposed the first MA ABE scheme. This KP-ABE scheme used one CA and multiple AAs. Identity-related keys are issued to users by the CA, which was also responsible for issuing seeds for each AA. The AAs managed attributes and issued attribute-related keys. A user's keys from different AAs were linked by a global identifier (GID) of the corresponding user. In this scheme, the CA still needed to be fully trusted since it was able to decrypt all ciphertexts, becoming a vulnerability in the system.

Li *et al.*[24] designed a MA CP-ABE scheme that also employed one CA and multiple AAs. In this scheme, the CA was also in charge of issuing the identity-related keys. However, contrary to[23], it was not able to decrypt any ciphertext. The security proofs of the scheme were based on the technique of dual system encryption and were given in the standard models. This scheme supported outsourcing of decryption and indirect user and attribute revocation.

Xue *et al.*[26] proposed an auditable and robust MA CP-ABE scheme addressing the issue of failures in AAs. Here, every AA was able to independently generate secret keys for any possible attribute set for users. In the case of a maliciously behaving AA, the CA was able to detect such behavior.

Yu *et al.*[25] presented a MA CP-ABE with the ability to revoke malicious users directly. Corresponding public parameters and ciphertext components were both updated via the revocation process. Keys of the remaining users remained untouched. To detect if a specific ciphertext was correctly updated by the cloud server, the revocation mechanism also offered verification.

4.2. Decentralized

Chase *et al.*[27] aimed to remove the need for a trusted CA, as in[23], and proposed a decentralized MA CP-ABE scheme without one. They used a secret pseudorandom function (PRF) and each pair of AAs shared the seeds during the initialization phase. If at most $(n - 2)$ AAs were corrupted, where n denotes the number of total AAs, the scheme was considered secure. Moreover, the scheme protected users from colluding with AAs that pool their information on a particular user.

Lewko and Waters[28] proposed a decentralized MA scheme offering full security, contrary to[23,27], which only provided selective security. However, with a large attribute universe, this could become inefficient. Nevertheless, in the random oracle model (ROM), it was the first adaptively (fully) secure MA CP-ABE scheme that was proven secure. Since no collaboration among AAs during the setup and key generation phase was required, it improved previous MA CP-ABE schemes. Any LSSS matrix could be used to describe the used access policy.

Han *et al.*[29] designed a decentralized KP-ABE scheme with privacy protection. The secret key distribution was carried out by AAs, among which cooperation was not required. The privacy preserving key extraction protocol was used under the standard complexity assumption. However, user collusion, meaning two users pooling their decryption keys to generate decryption keys, was not prevented by this scheme.

In[30], Zhang *et al.* presented a decentralized MA CP-ABE scheme supporting a large attribute universe. It was based on prime order groups and enabled tractability. This means that malicious users that leak their keys (even partially or modified) could be identified by the system.

4.3. Further Comparison

The schemes[23,25–27] were based on prime order groups and were proven secure under the standard model. Reference[30] was the only scheme that supports a large attribute universe. Scheme[24] was the only one that achieved full (adaptive) security; its security proof was based on the standard model. Moreover, it employed an efficient outsourced approach for decryption; thus, only one exponent operation was required. The schemes in[27–30] were decentralized, among which [27] required cooperation between AAs for initialization. In contrast, the schemes[23,24,26] were centralized and required a CA. The CAs in [23,26] were able to decrypt any ciphertext, thus possibly becoming a security vulnerability of the system. Each AA in[26] was responsible for managing all attributes in the system, whereas the AAs in the other schemes were only responsible for a subset of the total attribute universe. These subsets were disjunct among the AAs. The schemes[24,25,28,30] remained secure as long as one AA was not corrupted by an adversary. If both decentralized architecture and full security are required, we recommend the scheme[28].

5. ACCOUNTABLE ABE

Accountable ABE schemes have been introduced to improve the security of ABE and prevent the abuse of keys. Such schemes focus on solving two problems: the illegal sharing of keys among users and the misconduct of

AAs. They include both user accountability/tractability and the accountability of AAs[12]. Use accountability enables the tracking of unauthorized sharing of keys between users, while authority accountability prevents illegal key (re-)distribution by authorities. In the following, we review accountable ABE schemes from both the KP and the CP categories.

5.1. Accountable ABE Schemes

The first accountable CP-ABE scheme was proposed by Li *et al.*[31] in 2009. It embedded additional user-specific information into the user's secret key, thereby enabling private user accountability. Thus, the system could detect and prevent illegal keys from sharing among users. The scheme was selectively secure in the ROM. However, it lacked expressiveness by only supporting AND-gate access policies.

Wang *et al.*[32] presented a KP-ABE scheme with an accountable authority. The main idea of this scheme was a key-splitting trick. In this trick, the secret value of the keys was divided into two parts. These parts were then utilized to generate two partial keys. One of them corresponded to the access policy, and the other to the identity. However, the traceability algorithm required the entire well-formed decryption key.

Aiming to realize accountability both for users and AAs, Ning *et al.*[33] described an accountable CP-ABE scheme. This scheme used the efficient LSSS matrix as an access policy. In the white-box model, this enabled weak public accountability both for users and the AA. The authors presented the security proofs of the scheme under the q-string Diffie-Hellman (Q-SDH) assumption and some further assumptions.

In[34], Zhang *et al.* presented a security weakness in the scheme of Ning *et al.*[33], which resulted from re-randomization of the attribute secret key. Subsequently, they proposed a CP-ABE scheme with both user and AA accountability. The scheme also used LSSS for an access policy and was proven fully secure in the ROM.

Furthermore, Ning *et al.*[35] proposed an accountable LSSS-based CP-ABE scheme. It was a combination of conventional CP-ABE, anonymous IBE, and identity hierarchies. Moreover, full security and public user accountability were provided in this scheme.

Liu *et al.*[36] designed another accountable CP-ABE scheme. The scheme provides public user accountability and user revocation. For this, a publicly available revocation list exists that is used to update the ciphertexts accordingly.

5.2. Further Comparison

The schemes [31,34] provided the advantage of small and constant-size public keys. In[33,35], an efficient LSSS policy was used. They had a small attribute universe and were based on composite-order groups. Reference[32] was based on prime order groups with a small attribute universe and tree access policies. The two schemes[31,34] provided a large attribute universe and were resistant against selective adversaries. Besides[32], only the schemes[31,36] were constructed in prime order groups. Only the AA was able to perform the accountability of these schemes. Only two of the accountable ABE schemes simultaneously achieved the accountability of both users and AAs. We recommend the scheme proposed by Zhang *et al.*[34] for applications that require authority accountability, user accountability, and a large universe.

6. POLICY-HIDING ABE

Hidden policy ABE schemes, also called policy-hiding ABE, aim to protect the privacy of access policies. Only CP-ABE schemes can achieve this feature. This is because, as described in Section 2.5 , in CP, the DOs include the access policy into the ciphertext sent to the cloud. As a consequence, CP-ABE, without the hidden-policy feature risks, that anyone who is able to access the ciphertext can also learn the access policy. ABE schemes

with the hidden policy feature try to overcome this problem by hiding the access structure[18]. By hiding the access policy, it is no longer possible, for example, for attackers to recognize and suppress target group-specific ciphertexts or to recognize when a new ciphertext has been generated for which target group.

6.1. Hidden-Policy Schemes

Nishide *et al.*[37] proposed the first hidden policy schemes for CP-ABE. The security of their scheme was based on the DBDH and DLIN assumptions in the ROM. However, it was only proven secure for selective adversaries. The authors used the AND-gate on multi-valued attributes with wildcard access structures. These wildcards introduced "don't care" values for attributes in access structures. Hidden policy in this scheme was achieved by the inner product predicate encryption technique. Lai *et al.*[38] presented a new hidden-policy CP-ABE scheme in order to achieve full security against adaptive adversaries. This approach provided full security under new complexity assumptions and is based on composite-order groups as well. Both schemes[37,38] lacked expressiveness concerning their access structure since they only used policies with AND-gates. Moreover, the length of the ciphertext depended linearly on the number of attribute values, and they only supported a small attribute universe.

Next, Phuong *et al.*[39] proposed a hidden-policy CP-ABE scheme aiming to reduce the size of ciphertexts, which was constant in their scheme. For access policy, AND-gates were used with positive, negative, and wildcard attributes. This scheme had two vectors and transformed the attributes and access policies into them in order to hide the AND-gate policies. Furthermore, it used inner product encryption to hide the access policies. The scheme was only proven secure against selective adversaries.

With the goal of improving the computational efficiency and expressiveness, Zhang *et al.*[40] proposed a new policy-hiding CP-ABE scheme. In this approach, the authors used the efficient and expressive LSSS policies to design a policy-hiding ABE scheme with a large attribute universe. The attribute matching process required only two bilinear pairings and, thus, was efficient. Moreover, the scheme provided full security in the standard model.

In[41], Zhang *et al.* further improved their approach from[40] and proposed an extension of their scheme that, among other things, improved the efficiency by reducing decryption costs.

6.2. Further Comparison

The schemes[38,40,41] provided full security as they were proven secure against adaptive adversaries, while the schemes[37,39] provided only selective security. The schemes[37,39] only supported a small attribute universe, while the two schemes[40,41] supported a large attribute universe. Furthermore, both schemes had low decryption costs. They achieve efficient attribute matching since their computation costs were unaffected by the complexity of the access policy. Only the scheme[41] provided constant computation costs in the decryption process. In terms of expressiveness of access policies, the schemes[37–39] only used AND-gate access structures. The references[37,39] also used wildcard attributes, while[39] additionally supported non-monotonic access structures with negative attributes. Only the schemes of[40,41] supported the expressive LSSS policies. For applications requiring privacy-aware access control, for example, in mobile clouds, the scheme[41] can be recommended, considering its efficiency and full security features.

7. PROXY RE-ENCRYPTION AND POLICY UPDATING ABE

Traditional CP-ABE does not allow to dynamically change the access policy of a ciphertext. However, some access control applications require such a change. For example, in the case of emergencies, policy updating ABE schemes provide mechanisms to update the access policy in a ciphertext[15]. The policy updating feature in ABE is usually achieved by the technique of proxy re-encryption ABE. The concept of PRE aims to make

Table 2. Feature comparison of the ABE schemes

Schemes	Category	MA	Acc	HP	PRE	Revocable User	Revocable Attribute	Hierarchical	PU	Non-Mono
[17,62]	KP									
[19]	KP									y
[23]	KP	Central								
[24]	CP	Central				Indirect	Indirect			
[25]	CP	Central				Direct				
[26]	CP	Central	Authority							
[29]	KP	Decentral								
[27,28]	CP	Decentral								
[30]	CP	Decentral	User							
[30]	CP	Decentral	User							
[32]	KP		Authority							
[36]	CP		User			Direct				
[31,35]	CP		User							
[33,34]	CP		Both							
[37,38,40,41]	CP			y						
[39]	CP			y						y
[42,43]	CP				y				y	y
[44]	CP				y				y	
[45]	CP				y	Direct			y	
[46]	CP			y	y				y	
[47,48]	CP					Indirect	Indirect			
[49]	CP					Indirect				
[50]	KP					Indirect			AU	
[52,53]	CP					Direct	Direct			
[51]	CP					Direct				
[54]	KP					Direct				y
[55]	KP					Direct			AU	
[58–61]	CP							y		
[57]	CP					Direct		y		
[56]	CP					Indirect	Indirect	y		

MA: Multi-authority, Acc: Accountable, HP: Hidden-policy, PRE: Proxy re-encryption, PU: Policy updating, Non-Mono: Non-monotonic, AU: Attribute update (for KP)

Table 3. Notation

Notation	Description	Notation	Description
n	Size of attribute universe	G	Length of element in group g
l	Number of user's (CP) / ciphertext (KP) attributes	I	Complexity of the access policy
pair	Costs for a bilinear pairing	exp	Cost for exponentiation
pm	Cost for point multiplication	m_1	The upper bound of attribute number in encryption
$l_{p/n}$	Number of positive/negative attributes of a policy	n_1	The row number of the access policy matrix
n_a	The number of AAs involved	u	Number of users
ID	Length of identity of a user	N	Number of attribute values in the system
l_s	Length of a signature	$C_{s/v}$	Computation for generating signature/verification
t	Lifetime of the system	$n_{r/c}$	Number of rows/columns
$d_{t/v}$	Max depth of attribute trees/vectors	th	Threshold value
$hash$	Length of output of hash-function	h_ID	Number of hierarchies of the identity of the user
n_f	Number of files in 1 encryption	n_ch	Number of child nodes of a transport node

data sharing more efficient. The basic technique of PRE allows semitrusted parties, the so-called proxies, to transform the ciphertexts. Therefore, one ciphertext encrypted under the public key of one party can be transformed into a ciphertext intended for another party. This "new" ciphertext is based on the same plaintext. Proxy re-encryption ABE schemes apply this PRE primitive to the attribute-based context. In this category, data encrypted with an access policy is re-encrypted by a proxy into a new ciphertext according to a new access policy. As a result, users who comply with the new access policy can decrypt the encrypted text without the

proxy learning any useful information about the data. This feature is only relevant to CP-ABE[12].

7.1. PRE and PU Schemes

Liang *et al.*[42] proposed the first proxy re-encryption CP-ABE scheme. If the DO is offline in this scheme, he delegated his capacities to a proxy in the data access control. In this CP-ABPRE scheme, a proxy was allowed to transform the access policies of ciphertexts. Similar to the technique explained above, he transformed one with one specified access policy into one with another access policy. The policy is only represented as AND-gates on positive and negative attributes. The scheme used proxy re-encryption to enable policy updating. CP-ABPRE was proven secure only against selective adversaries and had high computational costs as it demanded a high number of pairings.

Luo *et al.*[43] extended the previous scheme and introduced the re-encryption control. Here, the DO had the power to decide if the ciphertext could be re-encrypted or not. The scheme used AND-gates with multi-value and wildcard attributes in its access policy. Moreover, it also supported non-monotonic access structures. The scheme was proven selectively secure under the standard complexity assumption.

Liang *et al.*[44] used the technique of dual system encryption and proposed a more efficient and expressive CP-ABPRE scheme. The scheme was proven secure against adaptive adversaries in the standard model under composite-order groups. Therefore, it achieved full security.

In 2016, Yang *et al.*[45] suggested another CP-ABPRE scheme. This approach used tree-based access control and also implemented policy updating through proxy re-encryption. The scheme had attribute secret keys of constant size. Its security analysis used generic group models. The scheme also supported direct user revocation in addition to policy updating and PRE.

Zhang *et al.*[46] designed a so-called anonymous CP-ABPRE scheme. In this approach, the technique called "match-then-re-encrypt" was proposed. This technique helped the proxy to decide whether a ciphertext should be transformed without requiring the access policy. The scheme was proven selectively secure under the assumptions of DBDH, DLIN, and CBDH.

7.2. Further Comparison

The scheme[44] used LSSS for access policies. Therefore, even if composite-order groups were implemented in the concrete design, it was more expressive than the other schemes. The schemes[42–44,46] were secure under the standard assumption. Only the scheme[45] had constant-size public keys. Therefore, it supported a large attribute universe. Moreover, the scheme[45] was the only approach with constant decryption costs that were independent of the complexity of the access structures. For security-sensitive applications, we recommend the scheme[46]. In the case of applications that have constrained resources, we recommend an efficient scheme[45].

8. REVOCABLE ABE

We classify revocation mechanisms in ABE into two classes, namely, user revocation and attribute revocation. The attribute revocation dynamically deals with editing the attributes of the users, including expiration, revocation, or addition of attributes. User revocation is used to remove or add users, for example, to deal with malicious behavior. Revocable ABE can be realized in two different ways, corresponding to the two subcategories: direct and indirect revocation[18]. Direct revocation works by taking into account a corresponding revocation list when creating a ciphertext, while indirect revocation involves the authorities updating keys accordingly, making the keys of removed members useless.

Table 4. Comparison of the ABE schemes according to evaluation criteria storage, communication, and computation costs

Scheme	Storage		Communication	Computation	
	Public key	Secret key	Ciphertext	Encryption	Decryption
Goyal *et al.* [17]	$nG + G_T$	$I * G$	$lG + G_T$	$(l+1)$exp	$(I+1)$exp
Ostrovsky *et al.* [19]	$(2+2m_1)G + G_T$	$2l_pG + 3l_nG$	$(2l+1)G + G_T$	$(2l+2)$exp+pair	$(5I+1)$pair+Iexp
Lai *et al.* [62]	$(n+2)G + G_T$	$(2n_1 + n_1^2))G$	$(l+1)G + G_T$	$l+2$ exp	2pair+$(I+2)$exp
Chase *et al.* [23]	$(n+1)G + G_T$	$(l+1)G$	$(l+1)G + G_T$	$(l+2)$exp	$(I+1)$pair+Iexp
Li *et al.* [24]	$(n+n_a+1)G + n_aG_T$	$(2+n_a+l)G$	$(2n_1+1)G + G_T$	$(3I+2)$exp	1 exp
Yu *et al.* [25]	$(2n+na+2)G + n_aG_T$	$(2n_a+4l)G$	$(4n_1+1)G + G_T$	$(5I+2)$exp	$(5I+n_a)$pair+2Iexp
Xue *et al.* [26]	$(n+2)G + G_T$	$(2n_1+1)G + G_T$	$(2l+2)G$	$(3I+2)$exp	$(2I+1)$pair+I exp
Han *et al.* [29]	$(2n_a+n)G$	$2n_1G$	$(l+1)G + G_T$	$(l+n_a+2)$exp	$(I+na+1)$pair+I exp
Chase *et al.* [27]	$(n+1)G + G_T$	$(l+1)G$	$(l+1)G + G_T$	$(l+2)$exp	$(I+1)$pair+Iexp
Lewko *et al.* [28]	$(n+1)G + nG_T$	lG	$2n_1G + (n_1+1)G_T$	$(5I+1)$exp+pair	2Ipair+Iexp
Zhang *et al.* [30]	$(3n_a+1)G + n_aG_T$	$4lG$	$5n_1G + (n_1+1)G_T$	$(8I+1)$exp+pair	3I pair+4I exp
Wang *et al.* [32]	$(n+4)G + 2G_T$	$3IG + IZ_p$	$(l+3)G + G_T$	$(l+3)$exp	$(I+3)$pair+2I exp
Liu *et al.* [36]	$(n+2u+2)G + G_T$	$(l+6)G$	$(n_1+r+2)G + G_T$	$(2n_1+3)$exp	$(2I+3)$pair+$(I+log\ u)$exp
Ning *et al.* [35]	$(n+u+5)G_{1,4} + G_4 + G_T$	$(l+u+6)G_{1,3}$	$(2n_1+4)G_{1,4} + G_T$	$(3n1+5)$exp	$(2I+5)$pair+I exp
Li *et al.* [31]	$2G + G_T$	$(4n+4ID)G$	$(4N+8ID)G + G_T$	$(4N+8ID+1)$exp	$(4n+4ID)$pair
Zhang *et al.* [34]	$4G + G_T$	$(l+3)G_{1,3} + 2G$	$(2n_1+3)G + G_T$	$(3n_1+4)$exp	$(2I+3)$pair+5exp
Ning *et al.* [33]	$(n+6)G + G_T$	$(l+3)G_{1,3} + 2G$	$(2n_1+4)G + G_T$	$(3n_1+5)$exp	$(2I+3)$pair+$(I+4)$exp
Nishide *et al.* [37]	$(2N+1)G + G_T$	$(N+1)G$	$(2N+1)G + G_T$	$(2N+2)exp$	$2*(3n+1)$pair
Lai *et al.* [38]	$(N+2)G + G_T$	$(n+1)G$	$(N+1)G + G_T$	$N+2$)exp	$2*(n+1)$pair
Phuong *et al.* [39]	$(8n+30)G + G_T$	$(4n+3)G$	$(4I+2)G + G_T$	$(12n+39)$exp	$2*(4n+12)$pair
Zhang *et al.* [41]	$4G + G_T$	$(l+2)G$	$(n_1+1)\|G_{1,4} + G_T$	$(3n_1+2)$exp	$2*$(2pair+2Iexp)
Zhang *et al.* [40]	$2G_1 + G_4 + G_{1,4} + G_T$	$(l+2)G$	$(3n_1+2)G1,4 + 2G_T$	$(6I+4)$exp	$(I+2)$pair+2Iexp
Liang *et al.* [42]	$(3n+2)G + G_T$	$(2l+1)G$	$(n+2)G + G_T$	$(n+3)$exp	$(n+1)$pair
Luo *et al.* [43]	$(N+2n+4)G + G_T$	$(4l+1)G$	$(n+2)G + G_T$	$(n+3)$exp	$(2n+1)$pair
Liang *et al.* [44]	$(n+6)G + G_T$	$(l+3)G$	$(2n_1+4)G1 + G_T + l_s$	$(3n_1+6)$exp+C_s	$(2I+2)$pair4 + I exp+C_v
Yang *et al.* [45]	$G + G_T$	$3Z_p$	$(2n_1+1)G + G_T$	2pair+$(2n1+3)$exp	1 pair+2 exp
Zhang *et al.* [46]	$(3N+4)G + G_T$	$(4n+4)G$	$(3N+3)G + 2G_T$	$(3N+5)$exp	$(3I+3)$pair
Hur *et al.* [47]	$G + G_T$	$(2l+1)G + log\ lZ_p$	$(2I+1)G + G_T$	$(2I+2)$exp	$(2I+1)$pair+$(I+log\ l)$exp
Yang *et al.* [48]	$(2n+4)G + G_T$	$(2l+2)G$	$(3n_1+1)G + G_T$	$(5n_1+2)$exp	$(2I+1)$pair+I exp
Cui *et al.* [49]	$7G + G_T$	Z_p	$(3n_1+2)G + G_T$	$(4n_1+3)$exp	1 exp
Xu *et al.* [50]	$(n+log_2t)G + G_T$	$2n_1log_2uG$	$(l+2)G + G_T$	$(l+3)$exp	$(I+2)$pair+$(I+1)$exp
Yang *et al.* [51]	$2G + G_T$	$(n+1)G$	$2G + 2G_T$	$(2I+5)$exp	$(r+2)$pair
Fan *et al.* [52]	$G_0 + (n+1)G_1 + 2G_T$	$3lG_0 + G_1$	$2G_0 + 2IG_1 + 2G_T$	$(2I+2)$pm+2 exp	$(3I+1)$ pair+I exp
Zhang *et al.* [53]	$(4n+2u+1)G$	$(n+1)\|G$	$2G + 2G_T$	$(2I+5)$exp	$(r+2)$ pair
Lewko *et al.* [54]	$4G + G_T$	$8G$	$(3l+1)G + G_T$	$(4l+2)$exp	$(2l_p+3l_n+1)$pair+2l_n exp
Shi *et al.* [55]	$(2m_1+7)G + G_T$	$(2I+2)G$	$(l+1)G + (log\ r+1)G_T$	$log\ r$ pair+$(l+2log\ r+2)$ exp	3I pair+I exp
Deng *et al.* [58]	$(n_r+n_c+3)G + G_T$	$(d_vl+2)G$	$(3n_1+2)G$	$((l+4)n1+2)$ exp	$(3n_vec+1)$ pair+$n_vec$exp
Li *et al.* [61]	$(n+d_v+2)G_1 + G_3 + G_T$	$(d_v+1)lG + G_{1,3}$	$(3n_1+1)G + G_T$	$(4n_1+d_v+2)$exp	$(3I+1)$ pair+I exp
Teng *et al.* [60]	$(2n+5)G + G_T$	$n(n+2)G$	$3G + G_T$	6 exp	6 pair+$(2th+2)$ exp
Wan *et al.* [57]	$5G + G_T$	$(2N+n+1)G$	$(3I+2)G + G_T$	$(3I+3)$ exp	2I pair+I exp
Wang *et al.* [56]	$2G$	$(l+2)G$	$(N+n)G + hash$	$(2N+n)$ pm	(h_ID+1) pair
Wang *et al.* [59]	$2G + G_T$	$(2l+1)G$	$(n_f+(n_ch+2)I)G + n_fG_T$	$(2n_f+(n_ch+2)I)$exp	$(2I+n_f)$ pair+I exp

8.1. Indirect Revocation

Hur and Noh [47] proposed a tree-based CP-ABE solution with fine-grained indirect user and attribute revocation mechanisms. The key technique of this scheme for efficient revocation is a stateless group key distribution method based on binary trees. This made the revocation efficient. Both backward secrecy and forward secrecy were achieved under the BDH assumption. However, the authors provided only an informal security analysis, and the scheme was susceptible to collusion attacks.

Yang *et al.* [48] also used the LSSS access policy to present a similar CP-ABE scheme. It was proven secure under the q-type assumption in the ROM model. Updates were performed by a third-party server, and both indirect attribute and user revocation were enabled by this scheme based on this update mechanism.

Cui *et al.* [49] proposed a CP-ABE scheme that did not support AA revocation; thus, only users could be revoked

Table 5. Comparison of the ABE schemes according to evaluation criteria access policy, security model and complexity assumption

Scheme	Access policy	Security model	Complexity assumption
Goyal*et al.*[17]	Tree	Selective	DBDH
Ostrovsky *et al.*[19]	LSSS	Selective	DBDH
Lai *et al.*[62]	LSSS	full	DBDH
Chase *et al.*[23]	Threshold	Selective	DBDH
Li *et al.*[24]	LSSS	Full	New
Yu *et al.*[25]	LSSS	Selective	q-type
Xue *et al.*[26]	LSSS	Selective	q-type
Han *et al.*[29]	Threshold	Selective	q-type
Chase *et al.*[27]	Threshold	Selective	DBDH
Lewko *et al.*[28]	LSSS	Full	New
Zhang *et al.*[30]	LSSS	Selective	q-type
Wang *et al.*[32]	Tree	Selective	mDBDH
Liu *et al.*[36]	LSSS	Selective	SDH
Ning *et al.*[35]	LSSS	Full	q-type
Li *et al.*[31]	AND	Selective	DBDH,D-lin
Zhang *et al.*[34]	LSSS	Selective	SDH, new
Ning *et al.*[33]	LSSS	Full	SDH, new
Nishide *et al.*[37]	And	Selective	DBDH,D-lin
Lai *et al.*[38]	And	Full	New
Phuong *et al.*[39]	And	Selective	DBDH,D-lin
Zhang *et al.*[41]	LSSS	Full	New
Zhang *et al.*[40]	LSSS	Full	New
Liang *et al.*[42]	And	Selective	ADBDH,CTDH
Luo *et al.*[43]	And	Selective	DBDH,CBDH
Liang *et al.*[44]	LSSS	Full	q-type,new
Yang *et al.*[45]	Tree	Selective	DBDH
Zhang *et al.*[46]	And	Selective	DBDH,D-lin,CBDH
Hur *et al.*[47]	Tree	Selective	BDH
Yang *et al.*[48]	LSSS	Selective	q-type
Cui *et al.*[49]	LSSS	Selective	q-type
Xu *et al.*[50]	LSSS	Selective	DBDH
Yang *et al.*[51]	And	Selective	q-type
Fan *et al.*[52]	Tree	Full	DBDH
Zhang *et al.*[53]	And	Selective	q-type
Lewko *et al.*[54]	Non-Mono	Selective	q-type
Shi *et al.*[55]	LSSS	Selective	multilin. DDH
Deng *et al.*[58]	LSSS	Full	New
Li *et al.*[61]	LSSS	Full	New
Teng *et al.*[60]	Threshold	Selective	q-type
Wan *et al.*[57]	Tree	Selective	BDH
Wang *et al.*[56]	DNF	Selective	BDH
Wang *et al.*[59]	Tree	Selective	DBDH

in an indirect manner. Non-revoked users were able to exploit an untrusted server to transform ciphertexts.

Xu *et al.*[50] proposed an LSSS-based KP-ABE scheme supporting indirect user revocation. The revocation mechanism was realized with the help of a subset-cover framework. This scheme achieved forward secrecy only. Moreover, the scheme also provided attribute update mechanisms. Nevertheless, this scheme only achieved selective security, and the decryption costs are still high.

8.2. Direct Revocation

Yang *et al.*[51] proposed a CP-ABE approach that used trees for direct user revocation. To achieve that, the server kept a proxy decryption key list that was used for that. Every DO had to generate master and secret keys, together with the suitable corresponding system public parameters.

Fan *et al.*[52] presented another tree-based CP-ABE scheme with revocation. The scheme allowed dynamic membership management with arbitrary states. Moreover, this approach provided both direct attribute and direct user revocation. The system's public key was adaptively updated to achieve that.

In [53], Zhang *et al.* described a CP-ABE scheme for direct attribute and direct user revocation. The scheme was based on an auxiliary function presented by the authors. It was used to specify and update the ciphertexts for revocation. The length of ciphertexts in this scheme was both constant and small. However, it also had some downsides because it only supported the AND-gate policy and only provided selective security.

A revocable KP-ABE scheme was proposed by Lewko et al [54]. This scheme had short parameters and secret keys. In order to achieve better decryption efficiency, the costs depending on "parameter and secret key size" were transferred to the side of the ciphertext. In the scheme, direct user revocation was realized based on a new technique called "two equation". However, the proposed solution only provided selective security, and q-type assumptions were involved in the proofs. Additionally, the scheme did not support attribute revocation.

Shi *et al.* [55] presented another KP-ABE scheme with direct user revocation. The scheme provided verifiable ciphertext delegation. This enabled an untrusted third party to update ciphertexts without any delegated keys. Both backward secrecy and forward secrecy were achieved in the scheme. However, the scheme was based on multilinear pairings. This is impractical since the candidates of multilinear pairings are limited and the security is questionable. In addition to revocation, the scheme also provided attribute update mechanisms.

8.3. Further Comparison

In terms of access policies, the schemes [48,49] supported expressive LSSS-based access structures. The schemes [47,51,52] used tree-based access structures, while [53] only provided access structures with AND-gates. Both user and attribute revocation were realized in [47,48,52,53], while the remaining schemes [49–51,54,55] only provided user revocation. However, the schemes [49,51] were very efficient in terms of user-side decryption costs since they outsourced extensive calculations to a server. The schemes [47,48] realized indirect attribute revocation in addition to indirect user revocation, while [52,53] realized both revocations directly. Only selective security was achieved in [53]. However, [52] was proven fully secure against adaptive adversaries in the standard models. Thus, the scheme [52] can be recommended in case full security and direct and fine-grained revocation are required. All schemes realize forward secrecy. However, only [49,51] additionally provide backward secrecy (without re-keying).

9. HIERARCHICAL ABE

In hierarchical ABE, the assignment of access rights is organized in a hierarchical manner. CP-ABE provides fine-grained access control. However, it is not optimal for large hierarchical structures such as enterprises. This is because it lacks a full delegation mechanism. Hierarchical ABE schemes can provide scalability for such large enterprises by using a hierarchical structure for attributes [16].

9.1. Hierarchical Schemes

Wang *et al.* [56] suggested a hierarchical CP-ABE scheme aimed at enhancing the flexibility of delegation for any user to join the access control system. The scheme combined hierarchical identity-based encryption and CP-ABE. Additionally, it supported indirect user and attribute revocation. In the ROM model, it had selective security under the BDH assumption. However, this approach suffered from performance drawbacks since the decryption overhead, due to pairings, was linearly proportional to the complexity of the access policy.

Wan *et al.* [57] described a tree-based hierarchical CP-ABE scheme that supported direct user revocation. Furthermore, since the security analysis relied on that of the basic CP-ABE scheme by Bethencourt *et al.* [21], the scheme had formal security in the ROM with generic groups.

In [58], Deng *et al.* presented a hierarchical CP-ABE scheme in which the attributes were organized as a matrix. The scheme supports LSSS-based access policies and, hence, was more expressive than the other approaches

mentioned before. Even though it was already fully secure in standard models, composite-order groups were used in the scheme design.

Wang *et al.*[59] proposed a hierarchical CP-ABE scheme that supported a large attribute universe and tree policies. However, it was only secure against selective adversaries.

Teng *et al.*[60] invented a hierarchical CP-ABE scheme with the goal of improving communication efficiency. It had ciphertexts of constant size and used a threshold-based access policy. The security analysis in standard models showed resistance to selective adversaries under q-type assumptions.

Li *et al.*[61] also designed a hierarchical CP-ABE scheme that provided resistance against side channel attacks. The scheme proved resilient to both master key leakage and secret key leakage under composite-order groups. The complexity of LSSS policies resulted in a larger computation overhead.

9.2. Further Comparison

Only the scheme[60] had ciphertexts of constant size. Additionally, the costs for encryption in this scheme were also constant and small. The schemes[56,57,59] all had public keys of constant size. Thus, they supported a large attribute universe. The two schemes[58,61] used an LSSS-based access policy. Therefore, they were the most expressive schemes. Moreover, both of these schemes were secure against adaptive adversaries under standard models. In addition to the hierarchical feature, only the scheme[57] provided direct user revocation, and only the scheme[56] realized indirect user and attribute revocation. As a result, the scheme[60] is especially appropriate for applications with limited resources due to its efficiency in the decryption process and the low communication costs. In case additional revocation of both user and attributes is required, the scheme[56] can be recommended.

10. EVALUATION SUMMARY

We now summarize our detailed analysis and evaluation of the ABE schemes. Table 2 provides a systematic feature comparison according to our taxonomy described in Section 3.2 . Each row of the table refers to one or more ABE schemes that possess the same properties in terms of the features represented in the column names. For improved clarity and readability, we leave a cell in the table blank in case a scheme does not have the feature of the corresponding column. Additionally, Table 3 provides an explanation of the notation used in Table 4. Subsequently, Table 4 and Table 5 systematically compare the ABE schemes according to the evaluation criteria described in Section 3.3 . Table 4 focuses on the storage, communication, and computation costs, while Table 5 presents and compares the scheme according to their access policy, security model, and complexity assumption. With the help of theses tables, we provide a comprehensive overview and comparison of the ABE schemes from different categories covering both KP- and CP-ABE methods.

11. CONCLUSIONS

Cloud computing is becoming a more and more omnipresent computing paradigm used in a wide range of application scenarios. A large amount of today's digital services relies on some forms of cloud services providing dynamic access to resources anywhere and anytime. Attribute-based encryption (ABE) successfully achieves secure and private access control in such cloud computing scenarios. In this paper, we provided a systematic survey and comparison of current ABE schemes. We first proposed a taxonomy, dividing CP-ABE and KP-ABE into further categories, namely multi-authority, accountable, hidden policy, proxy re-encryption, policy updating, revocable, hierarchical, and (non-)monotonic ABE. Then, we investigated current ABE schemes from all these categories and systematically compared them according to the evaluation criteria described in Section 3.3 . In this comparison, our goal was to focus on both the efficiency and security of ABE schemes.

We surveyed both established classical schemes and newer schemes published in 2015 or later. Altogether, we surveyed and compared a total of 42 different ABE schemes. In summary, we provided a comprehensive systematization of existing knowledge on ABE in addition to a detailed evaluation and comparison of existing ABE schemes.

In future research on ABE, important challenges and issues have to be solved. First of all, efficient ABE schemes without bilinear pairing operations are a crucial and desirable goal. The typical cryptographic operations of ABE mainly include the bilinear pairing, the exponentiation, the point multiplication, and arithmetic operations in groups. However, bilinear pairings have high computational costs that are especially larger than those of other operations[15]. However, as one can see in Table 4, current ABE schemes often require bilinear pairings, which is an issue. In comparison to other one-to-many or many-to-many cryptographic techniques, such as symmetric encryption or traditional public-key encryption, ABE schemes have the disadvantage of high computation costs due to frequent bilinear pairing operations. In order to achieve more practicality in ABE, especially for resource-limited scenarios with mobile devices, the design of efficient ABE schemes without these pairing operations is an important research challenge.

Another important goal is the design of expressive ABE schemes that achieve as many features of ABE as possible. Such features, for example, the revocation of users or attributes, the accountability of users or authorities, the update of policies, the protection of the privacy of attributes or policy, or decentralization and hierarchical scalability, are crucial for deploying ABE schemes in modern applications. For example, as we can see in the results of our survey, there is currently no ABE scheme that provides direct user and attribute revocation while having expressive LSSS-based access policy and simultaneously achieving both forward and backward secrecy without re-keying. Moreover, many accountable schemes have the drawbacks of only supporting AND-gate policy or selective security. Furthermore, among the schemes we considered, there exists no hidden-policy or policy updating schemes that additionally provide accountability or the features of hierarchical or multi-authority ABE.

As an outlook on the future development of computers, technology advances towards the creation of quantum computers with immense computational power. In this context, many cryptographic techniques, including public-key encryption such as ABE, require security advancements in order to resist possible attacks from quantum-based adversaries.

DECLARATIONS

Authors' contributions

The primary contributions were made by Thomas Prantl and Timo Zeck, while the other authors supported the paper in different roles, facilitating its publication.

Availability of data and materials

Not applicable.

Financial support and sponsorship

This research has been funded by the Federal Ministry of Education and Research of Germany in the framework KMU-innovativ - Verbundprojekt: Secure Internet of Things Management Platform - SIMPL (project number 16KIS0852)[63].

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Renugadevi N, Swaminathan G, Kumar AS. Key management schemes for secure group communication in wireless networks - a survey. In: 2014 International Conference on Contemporary Computing and Informatics (IC3I); 2014. pp. 446–50. DOI
2. Boyd C, Mathuria A, Stebila D. Protocols for authentication and key establishment. 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg; 2020. DOI
3. Steam. STEAM- & SPIELSTATISTIKEN;. Available from: https://store.steampowered.com/stats/?l=german. [Last accessed on 29 Nov 2023].
4. Activision Blizzard I. Activision Blizzard Announces Third-Quarter 2019 Financial Results;. Available from: https://investor.activision.com/static-files/594047f5-10b9-4fbf-b43b-45c8552cbd79. [Last accessed on 29 NOV 2021].
5. Department SR. Anzahl der zahlenden Streaming-Abonnenten von Netflix weltweit vom 3. Quartal 2011 bis zum 1. Quartal 2021. Available from: https://de.statista.com/statistik/daten/studie/196642/umfrage/abonnenten-von-netflix-quartalszahlen/. [Last accessed on 29 Nov 2023].
6. Brandt M. 155 Millionen Premium-Kunden. Available from: https://de.statista.com/infografik/13769/monatlich-aktive-nutzer-und-zahlende-abonnenten-von-spotify-weltweit/. [Last accessed on 29 Nov 2023].
7. Prantl T, Ten P, Iffländer L, et al. Evaluating the Performance of a State-of-the-Art Group-oriented Encryption Scheme for Dynamic Groups in an IoT Scenario. In: Proceedings of the 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS); 2020. pp. 1–8. DOI
8. Nishat K, Purushothama BR. Group-oriented encryption for dynamic groups with constant rekeying cost. *Security Comm Networks* 2016;9:4120–37. DOI
9. Rodeh O, Birman K, Dolev D. Optimized group rekey for group communication systems. Available from: https://www.researchgate.net/publication/2325687_Optimized_Group_Rekey_for_Group_Communication_Systems/. [Last accessed on 29 Nov 2023].
10. Waldvogel M, Caronni G, Sun D, Weiler N, Plattner B. The versaKey framework: versatile group key management. *IEEE J Select Areas Commun* 1999;17:1614-31. DOI
11. Qiao Z, Liang S, Davis S, Jiang H. Survey of attribute based encryption. In: 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD); 2014. pp. 1–6. DOI
12. Bordogna G, Pang L, Yang J, Jiang Z. A survey of research progress and development tendency of attribute-based encryption. *ScientificWorldJournal*2014;2014:193426. DOI
13. Priya A, Tiwari R. A survey: attribute based encryption for secure cloud. *IJOSTHE* 2018;5:12. DOI
14. Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano D, Fazio N, Gennaro R, Nicolosi A, editors. Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography (PKC). Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. pp. 53–70. DOI
15. Zhang Y, Deng RH, Xu S, Sun J, Li Q, Zheng D. Attribute-based encryption for cloud computing access control: a survey. *ACM Comput Surv* 2021;53:1–41. DOI
16. Lee CC, Chung PS, Hwang MS. A survey on attribute-based encryption schemes of access control in cloud environments. *Int J Netw Secur* 2013;15:231–40. DOI
17. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS). New York, NY, USA: Association for Computing Machinery; 2006. pp. 89–98. DOI
18. Premkamal PK, Pasupleti SK, Alphonse PJA. Attribute based encryption in cloud computing: a survey, gap analysis, and future directions. *J Netw Comput Appl* 2018;108:37 – 52. DOI
19. Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS). New York, NY, USA: Association for Computing Machinery; 2007. pp. 195–203. DOI
20. Sahai A, Waters B. Fuzzy identity-based encryption. In: Cramer R, editor. Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Berlin, Heidelberg: Springer Berlin Heidelberg; 2005. pp. 457–73. DOI
21. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In:2007 IEEE Symposium on Security and Privacy (SP '07), Berkele, CA, USA, 2007, pp. 321-34, DOI

22. Rasori M, Manna ML, Perazzo P, Dini G. A survey on attribute-based encryption schemes suitable for the internet of things. *IEEE Internet Things J* 2022;9:8269–90. DOI
23. Chase M. Multi-authority attribute based encryption. In: Vadhan SP, editor. Proceedings of the 4th Theory of Cryptography Conference (TCC). Berlin, Heidelberg: Springer Berlin Heidelberg; 2007. pp. 515–34. DOI
24. Li Q, Ma J, Li R, Liu X, Xiong J, Chen D Secure, efficient and revocable multi-authority access control system in cloud storage. *Computers Security* 2016;59:45–59. DOI
25. Yu P, Wen Q, Ni W, et al. Decentralized, revocable and verifiable attribute-based encryption in hybrid cloud system. *Wireless Pers Commun* 2019;106:719–38. DOI
26. Xue K, Xue Y, Hong J, et al. RAAC: robust and auditable access control with multiple attribute authorities for public cloud storage *IEEE Trans Inform Forensic Secur* 2017;12:953–67. DOI
27. Chase M, Chow SSM. Improving privacy and security in multi-authority attribute-based encryption. In: Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS). ACM. New York, NY, USA: Association for Computing Machinery; 2009. pp. 121–30. DOI
28. Lewko A, Waters B. Decentralizing attribute-based encryption. In: Paterson KG, editor. Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. pp. 568–88. DOI
29. Han J, Susilo W, Mu Y, Yan J. Privacy-preserving decentralized key-policy attribute-based encryption. *IEEE Trans Parallel Distrib Syst* 2012;23:2150–62. DOI
30. Zhang K, Li H, Ma J, Liu X. Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability. *Sci China Inf Sci* 2018;61. DOI
31. Li J, Ren K, Zhu B, Wan Z. Privacy-aware attribute-based encryption with user accountability. In: Samarati P, Yung M, Martinelli F, Ardagna CA, editors. Proceedings of the 12th International Information Security Conference (ISC). Berlin, Heidelberg: Springer Berlin Heidelberg; 2009. pp. 347–62. DOI
32. Wang Y, Chen K, Long Y, Liu Z. Accountable authority key policy attribute-based encryption. *Sci China Inf Sci* 2012;55:1631–38. DOI
33. Ning J, Dong X, Cao Z, Wei L. Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud. In: Pernul G, Y A Ryan P, Weippl E, editors. Proceedings of the 20th European Symposium on Research in Computer Security (ESORICS). Cham: Springer International Publishing; 2015. pp. 270–89. DOI
34. Zhang Y, Li J, Zheng D, Chen X, Li H. Towards privacy protection and malicious behavior traceability in smart health. *Pers Ubiquit Comput* 2017;21:815–30. DOI
35. Ning J, Cao Z, Dong X, Gong J, Chen J. Traceable CP-ABE with short cipher- texts: How to catch people selling decryption devices on eBay efficiently. In: Askoxylakis I, Ioannidis S, Katsikas S, Meadows C, editors. Proceedings of the 21st European Symposium on Research in Computer Security (ESORICS). Cham: Springer International Publishing; 2016. pp. 551–289. DOI
36. Liu Z, Duan S, Zhou P, Wang B. Traceable-then-revocable ciphertext-policy attribute-based encryption scheme. *Future Gener Comp Sy* 2017;93:903–13. DOI
37. Nishide T, Yoneyama K, Ohta K. ABE with partially hidden encryptor-specified access structure. In: Bellovin SM, Gennaro R, Keromytis A, Yung M, editors. Proceedings of the 6th International Conference on Applied Cryptography and Network Security (ACNS). Berlin, Heidelberg: Springer Berlin Heidelberg; 2008. pp. 111–29. DOI
38. Lai J, Deng RH, Li Y. Fully secure cipertext-policy hiding CP-ABE. In: Bao F, Weng J, editors. Proceedings of the 7th International Conference on Information Security Practice and Experience (ISPEC). Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. pp. 24–39. DOI
39. Phuong TVX, Yang G, Susilo W. Hidden ciphertext policy attribute-based encryption under standard assumptions. *IEEE Trans Inform Forensic Secur* 2016;11:35–45. DOI
40. Zhang Y, Zheng D, Deng RH. Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet Things J* 2018;5:2130–45. DOI
41. Zhang L, Hu G, Mu Y, Rezaeibagha F. Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system. *IEEE Access* 2019;7:33202–13. DOI
42. Liang X, Cao Z, Lin H, Shao J. Attribute based proxy re-encryption with delegating capabilities. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS). ACM. New York, NY, USA: Association for Computing Machinery; 2009. pp. 276–86. DOI
43. Luo S, Hu J, Chen Z. Ciphertext policy attribute-based proxy re-encryption. In: Soriano M, Qing S, López J, editors. Information and Communications Security. Berlin: Springer Berlin Heidelberg; 2010. pp. 401-15. DOI
44. Liang K, Au MH, Liu JK, et al. A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. *Future Gener Comp Sy* 2015;52:95–108. DOI
45. Yang Y, Zhu H, Lu H, Weng J, Zhang Y, Choo KR. Cloud based data sharing with fine-grained proxy re-encryption. *Pervasive Mob Comput* 2016;28:122–34. DOI
46. Zhang Y, Li J, Chen X, Li H. Anonymous attribute-based proxy re-encryption for access control in cloud computing. *Security Comm Networks* 2016;9:2397–411. DOI
47. Hur J, Noh DK. Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Trans Parallel Distrib Syst* 2011;22:1214–21. DOI
48. Yang K, Jia X, Ren K. 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. ACM. New York, NY, USA: Association for Computing Machinery; 2013. pp. 523–28. DOI

49. Cui H, Deng RH, Li Y, Qin B. Server-aided revocable attribute-based encryption. In: Askoxylakis I, Ioannidis S, Katsikas S, Meadows C, editors. Proceedings of the European Symposium on Research in Computer Security (ESORICS). Cham: Springer International Publishing; 2016. pp. 570–87. DOI
50. Xu S, Yang G, Mu Y, , Deng RH. Secure fine-grained access control and data sharing for dynamic groups in the cloud. *IEEE Trans Inform Forensic Secur* 2018;13:2101–13. DOI
51. Yang Y, Liu JK, Liang K, Cho KKR, Zhou J. Extended proxy-assisted approach: Achieving revocable fine-grained encryption of cloud data. In: Pernul G, Y A Ryan P, Weippl E, editors. Proceedings of the European Symposium on Research in Computer Security (ESORICS). Cham: Springer International Publishing; 2015. pp. 146–66. DOI
52. Fan CI, Huang VSM, Ruan HM. Arbitrary-state attribute-based encryption with dynamic membership. *IEEE Trans Comput* 2013;63:1951–61. DOI
53. Zhang Y, Chen X, Li J, Li H, Li F. Attribute-based data sharing with flexible and direct revocation in cloud computing. *KSII TIIS* 2014;8:4028–49. DOI
54. Lewko A, Sahai A, Waters B. Revocation systems with very small private keys. In: 2010 IEEE Symposium on Security and Privacy (SP). IEEE. Los Alamitos, CA, USA: IEEE Computer Society; 2010. pp. 273–85. DOI
55. Shi Y, Zheng Q, Liu J, Han Z. Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation. *Inf Sci* 2015;295:221–31. DOI
56. Wang G, Liu Q, Wu J, Guo M. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *Comput Secur* 2011;30:320–31. DOI
57. Wan Z, Liu J, , Deng RH. HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE Trans Inform Forensic Secur* 2012;7:743–54. DOI
58. Deng H, Wu Q, Qin B, et al. Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts. *Inf Sci* 2014;275:370–84. DOI
59. Wang S, Zhou J, Liu JK, Yu J, Chen J, Xie W An efficient file hierarchy attribute-based encryption scheme in cloud computing. *IEEE Trans Inform Forensic Secur* 2016;11:1265–77. DOI
60. Teng W, Yang G, Xiang Y, Zhang T, Wang D. Attribute-based access control with constant-size ciphertext in cloud computing. *IEEE Trans Cloud Comput* 2017;5:617–27. DOI
61. Li J, Yu Q, Zhang Y. Hierarchical attribute based encryption with continuous leakage-resilience. *Inf Sci* 2019;2:113–34. DOI
62. Lai J, Deng RH, Li Y, Weng J. Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security. New York, NY, USA: Association for Computing Machinery; 2014. pp. 239–48. DOI
63. Prantl T, Ben Yahya AE, Dmitrienko A, et al. SIMPL: Secure IoT Management Platform. In: Proceedings of the 1st ITG Workshop on IT Security (ITSec 2020), April 2-3 2020. Tübingen: Universität Tübingen; 2020. . DOI