Revocable Attribute-Based Encryption Scheme with Arithmetic Span Program for Cloud-Assisted IoT

Hu Xiong$^{(\boxtimes)}$, Jinhao Chen$^{(\boxtimes)}$, Minghao Yang$^{(\boxtimes)}$, and Xin Huang$^{(\boxtimes)}$

School of Information and Software Engineering,
University of Electronic Science and Technology of China, Chengdu, China
xionghu.uestc@gmail.com, jinhaochen.cloud@gmail.com,
yangminghao1205@gmail.com, huangxin90427@gmail.com

Abstract. Efficient user revocation and description of the access policy are essential to enhance the practicality of attribute-based encryption (ABE) in real-life scenarios, such as cloud-assisted IoT. Nevertheless, existing ABE works fail to balance the two vital indicators. Motivated by this, in this paper, we present a revocable ciphertext-policy attribute-based encryption with arithmetic span programs (R-CPABE-ASP) for cloud-assisted IoT. For the first time, the presented R-CPABE-ASP achieves efficient user revocation and expressive description of access policy simultaneously. In R-CPABE-ASP, each attribute involved in access policy is merely used once to check whether a user owns access to shared data. Hence, the R-CPABE-ASP work enables efficient data encryption compared with existing revocable ABE works by reducing unnecessary cost for defining access policy. Meanwhile, as shown in the outsourced version of R-CPABE-ASP, the costly part for users to decrypt the data is outsourced to powerful cloud servers. Therefore, users in our R-CPABE-ASP can access their data in a more efficient way by merely one exponential operation. Finally, we carry out detailed theoretical analysis and experimental simulations to evaluate the performance of our work. The results fairly show that our proposed work is efficient and feasible in cloud-assisted IoT.

Keywords: Cloud-assisted IoT · Attribute-based encryption · Arithmetic span program · Revocation

1 Introduction

The Internet of Things (IoT) is a complex heterogeneous network. It connects various smart devices through communication and information technology to achieve intelligent identification, positioning, tracking, supervising and so on [1]. At present, IoT applications have been widely used in different fields such as smart cities, e-health, and intelligent transportation systems. However, with the increase of smart devices, more resources are required to manage and process the large amount of data generated by numerous smart devices in the IoT [2–6].

For example, medical systems and traffic monitoring systems generate giga-level high-definition images and videos per minute [7]. It is hard for ordinary users or smart devices in the traditional IoT to undertake the heavy burden in both storage and calculation. Fortunately, cloud-assisted IoT provides a promising solution for solving the kind of data explosion problem under the constraints of individual object capabilities. As a powerful platform, cloud computing empowers users with on-demand services [8] for storing, accessing, and processing data.

Although cloud computing brings immense benefits to IoT, it also takes unprecedented security risks due to its openness. Specifically, the data collected by the smart devices may contain the user's private information [9,10]. The curious cloud servers and the unauthorized users may make the endeavors to obtain user's personal information for financial gains. For this reason, keeping the confidentiality of user's data is vital of importance. Meanwhile, out of the needs of efficient data sharing in cloud-assisted IoT, it is desirable to design an effective mechanism that enables flexible access control. Due to the advantages in ensuring data confidentiality and realzing fine-grained access control, the primitive of attribute-based encryption (ABE) [11] was widely explored in cloud-assisted IoT. In the ABE schemes [11–14], both the ciphertext and the key are related to a set of attributes. The encrypter can formulate an encryption strategy consisting of attributes according to the sensitive content and the receiver's characteristic information. With this method, the resulting ciphertext can only be decrypted by users whose attributes meet the encryption strategy. In this way, not only the confidentiality of the sensitive data is assured but also the access control is achieved in a flexible and fine-grained way. While a series of ABE schemes followed [15,16], nonetheless, the potential user revocation [17] in cloud-assisted IoT is also challenging to the conventional ABE work.

For enhancing the practicality of ABE, the conception of revocable ABE was subsequently presented. The methods applied in the revocable ABE works for achieving user revocation can be typically divided into two categories: direct revocation and indirect revocation. In directly revocable ABE works [18,19], the data encryptors need to maintain a revocation list delivered by the trusted authority and keep it up-to-date. Obviously, the communication overhead for requesting the latest revocation list from the trusted authority is burdensome for data encryptors. To avoid the heavy overhead for maintaining the revocation list, various ABE works that supports indirection user revocation were proposed [20–23]. In an ABE work that supports indirect revocation, each non-revoked user will receive an extra key update material from the trusted authority for generating a complete decryption key. It is effectively guaranteed that the revoked users without key update materials have no access to the shared data. Despite of this, existing revocable ABE works is still insufficient to handle the complicated access policy in a large-scale cloud-assisted IoT system. In these works, the attributes involved in the access policy will be multiply used to establish a fine-grained access control of the sensitive data. It indeed causes extra overheads for embedding the access policy into a ciphertext.

Motivated by this, in this paper, we present a revocable ciphertext-policy attribute-based encryption scheme with arithmetic span program [24–27] (R-CPABE-ASP) for cloud-assisted IoT environment. The presented R-CPABE-ASP not only achieves fine-grained access control and necessary user revocation but also enables efficient access policy description. Thanks to the expressive ASP access structure and decryption outsourcing, our R-CPABE-ASP obtains high efficiency in the phases of encryption and decryption. In detail, our main contributions are listed below:

- We propose the first R-CPABE-ASP scheme that achieves user revocation and ASP access structure simultaneously. The presented R-CPABE-ASP can effectively address the potential changes in user's access right to shared data thanks to the introduction of user revocation.
- Compared with existing revocable ABE works where the attributes will be multiply required to establish an access policy, each attribute in access policy of our R-CPABE-ASP scheme is merely used once. Hence, the R-CPABE-ASP owns higher efficiency in data encryption for embedding the access policy into a ciphertext. Furthermore, an outsourced version of R-CPABE-ASP (OR-CPABE-ASP) is given, in which the overhead for data decryption is reduced to one exponential operation. Thus, even light weight users can efficiently access the data in cloud-assisted IoT environment.
- Our proposed R-CPABE-ASP is proved to be adaptively secure under the $MDDH_{k,l}^m$ assumption by using dual system encryption technology [28]. Finally, the detailed theoretical analysis and experimental simulations demonstrate that the presented R-CPABE-ASP is secure, efficient and feasible in cloud-assisted IoT.

1.1 Organizations

The rest of this paper is conducted as follows: Sect. 3 gives some basic notations and structures. The concrete construction of R-CPABE-ASP is contributed in Sect. 4. The performance evaluation is carried out in Sect. 5. Finally, this paper is concluded in Sect. 6.

2 Related Work

Attribute-based encryption (ABE) has been well explored in two branches of key-policy ABE and ciphertext-policy ABE since its seminal proposal [11]. Goyal et al. [12] presented a key-policy attribute-based work, in which each user's secret key is bound to a tree-access structure. Kaaniche and Laurent [13] presented a privacy-preserving ABE scheme. In their scheme, the general tree access structure is adopted to prevent the sensitive data from being deciphered without authorization. Li et al. [14] proposed a secure cloud data storage system for cloud IoT environment by utilizing ABE with AND-gate access structure. Nevertheless, the normal access structures adopted in these works suffer from

heavy overheads to define the access policy of the sensitive data. The fatal reason is that existing access structures establish access control to sensitive data by enumerating all the attribute permutations that meet the access policy. It inevitably causes extra expenses for defining the access policy during the phase of data encryption since the same attribute is used multiple times. Subsequently, the proposal of the arithmetic span program (ASP) [24] contributes a feasible solution for describing the access policy in an efficient manner. By adopting the notion of ASP, the access policy can be defined as an arithmetic expression of the attributes involved [26]. Even for a complicated access policy, the same attribute is merely required once. For this reason, the overhead for defining the access policy is significantly reduced. Inspired by this, Chen *et al.* [25] constructed the first KP-ABE for ASP. Ma *et al.* [27] presented the first CP-ABE for ASP recently. Nevertheless, the above mentioned works can hardly handle the challenging user revocation issue caused by the change of the user's permission in real-life scenarios.

For heightening the feasibility of conventional ABE works in real-life scenarios, measures for realizing user revocation have been well studied in the context of ABE. Zhang *et al.* [18] proposed a directly revocable ABE work with constant-size ciphertext. In their work, the user revocation is realized by the revocation list embedded into the ciphertext during data encryption. However, each the encrypter needs to continuously request the latest revocation list from the trusted authority. Fairly, encrypters in their work will confront heavy communication burden for keeping the revocation list up-to-date. This issue also threatens all ABE works that support direct user revocation. For eliminating this issue, Qin *et al.* [20] proposed indirect revocable ABE work. In their work, the trusted authority periodically broadcasts the key update materials to each non-revoked user. Only the users who generate a complete decryption key with the key update materials have access to the data. In this way, the procedure of user revocation will not put any extra burden on the encrypter. Thereafter, Xu *et al.* [21] constructed a secure IoT cloud storage system that achieves fine-grained access control by LSSS access structure. Wei *et al.* [23] contributed a revocable storage system for ensuring the security of e-Health records in public cloud scenarios.

Obviously, both efficient access structure and user revocation are essential to make a practical and robust ABE work in cloud-assisted IoT. What a pity, to the best of our knowledge, the ABE work that supports ASP access structure and user revocation simultaneously has not been well discussed.

3   Preliminaries

In this section, some basic knowledge that will be used in the following part of this paper is given.

3.1   Mathematical Notations

For a prime order asymmetric bilinear pairing $(e, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h)$, $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$ are prime order groups, e is a map from $\mathbb{G} \times \mathbb{H}$ to $\mathbb{G}_T$, g and h are the generators of $\mathbb{G}$ and $\mathbb{H}$, respectively. With the basis, some operations can be defined as:

$\diamond$ Given a vector $\mathbf{A} = (g^{a_1}, g^{a_2})^\top$ and a matrix $\mathbf{B} \in \mathbb{Z}_p^{3 \times 2}$, $\mathbf{A}^{\mathbf{B}} = g^{\mathbf{B}(a_1, a_2)^\top}$, where $a_1, a_2 \in \mathbb{Z}_p$.

$\diamond$ Given a matrix $g^{\mathbf{C}_1^\top}$, where $\mathbf{C}_1 = (c_{11}, c_{21}, c_{31}) \leftarrow_R \mathbb{Z}_p^3$, $\boldsymbol{\lambda} \leftarrow_R \mathbb{Z}_p^{2 \times 1}$, the result of $g^{\boldsymbol{\lambda} \cdot \mathbf{C}_1^\top}$ can be easily obtained as:

$$
\begin{aligned}
g^{\boldsymbol{\lambda} \cdot \mathbf{C}_1^\top} &= g^{\begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix} \cdot \begin{bmatrix} c_{11} & c_{21} & c_{31} \end{bmatrix}} = g^{\begin{bmatrix} \lambda_{11} \cdot c_{11} & \lambda_{11} \cdot c_{21} & \lambda_{11} \cdot c_{31} \\ \lambda_{21} \cdot c_{11} & \lambda_{21} \cdot c_{21} & \lambda_{21} \cdot c_{31} \end{bmatrix}} \\
&= \begin{bmatrix} (g^{c_{11}})^{\lambda_{11}} & (g^{c_{21}})^{\lambda_{11}} & (g^{c_{31}})^{\lambda_{11}} \\ (g^{c_{11}})^{\lambda_{21}} & (g^{c_{21}})^{\lambda_{21}} & (g^{c_{31}})^{\lambda_{21}} \end{bmatrix},
\end{aligned}
$$

where $g^{c_{11}}, g^{c_{21}}, g^{c_{31}}$ can be gained from the matrix $g^{\mathbf{C}_1^\top}$.

$\diamond$ Given a matrix $g^{\mathbf{C}_1^\top}$ with unknown $\mathbf{C}_1 \leftarrow_R \mathbb{Z}_p^3$ and a matrix $\boldsymbol{\vartheta} \leftarrow_R \mathbb{Z}_p^{3 \times 2}$, the value of matrix $g^{\mathbf{C}_1^\top \cdot \boldsymbol{\vartheta}}$ can be easily obtained following the above-mentioned steps.

3.2   Basis Structure

To simulate the composite-order groups with three primes- order subgroups. We first choose $l_1, l_2, l_3, l_w \geq 1$, and pick $\mathbf{W}_1 \leftarrow_R \mathbb{Z}_p^{l \times l_1}, \mathbf{W}_2 \leftarrow_R \mathbb{Z}_p^{l \times l_2}, \mathbf{W}_3 \leftarrow_R \mathbb{Z}_p^{l \times l_3}$, where $l = l_1 + l_2 + l_3$. $(\mathbf{W}_1^* | \mathbf{W}_2^* | \mathbf{W}_3^*)^\top$ is defined as the inverse of $(\mathbf{W}_1 | \mathbf{W}_2 | \mathbf{W}_3)$. It is clear that $\mathbf{W}_i^\top \mathbf{W}_i^* = \mathbf{I}$, and $\mathbf{W}_i^\top \mathbf{W}_j^* = \mathbf{0}$ $(i \neq j)$, where $\mathbf{I}$ is the identity matrix. And for any $\mathbf{T} \leftarrow_R \mathbb{Z}^{l \times l_w}$, there's always $\mathbf{T} = \mathbf{B}^{(1)} + \mathbf{B}^{(2)} + \mathbf{B}^{(3)}$, where $\mathbf{B}^{(1)} \leftarrow_R \mathsf{span}^{l_w}(\mathbf{W}_1^*), \mathbf{B}^{(2)} \leftarrow_R \mathsf{span}^{l_w}(\mathbf{W}_2^*), \mathbf{B}^{(3)} \leftarrow_R \mathsf{span}^{l_w}(\mathbf{W}_3^*)$.

Theorem 1. Given matrices $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_1^*, \mathbf{W}_2^*, \mathbf{W}_3^*, \mathbf{T}$ mentioned in the basis structure, the following two distributions $\{\mathbf{W}_1^\top \mathbf{T}, \mathbf{W}_3^\top \mathbf{T}, \mathbf{T}\}$ and $\{\mathbf{W}_1^\top \mathbf{T}, \mathbf{W}_3^\top \mathbf{T}, \mathbf{T} + \mathbf{P}^{(2)}\}$ are statistically identical with the probability $1 - 1/p$, where $\mathbf{P}^{(2)} \leftarrow_R \mathsf{span}^{l_w}(\mathbf{W}_2^*)$.

3.3   MDDH$_{k,l}^m$ Assumption

For any PPT adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{MDDH}_{k,l}^m}(\lambda) = |\Pr[\mathcal{A}(\mathbb{G}, g^{\mathbf{M}}, g^{\mathbf{MS}}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, g^{\mathbf{M}}, g^{\mathbf{S}'}) = 1]|$ is negligible for a security parameter λ, in which g is the generator of $\mathbb{G}$, $\mathbf{M} \leftarrow_R \mathbb{Z}_p^{l \times k}$, $\mathbf{S} \leftarrow_R \mathbb{Z}_p^{k \times m}$, $\mathbf{S}' \leftarrow_R \mathbb{Z}_p^{l \times m}$ ($m \geq 1$ and $l > k \geq 1$). According to [29], the MDDH$_{k,l}^m$ assumption is equivalent to the well-known k-Linear assumption [29]. For convenience, we denote MDDH$_{k,k+1}^1$ by MDDH$_k$ in the remainder part of this paper.

3.4   Arithmetic Span Program

The ASP access policy is formed with a vector set $\mathcal{V} = \{\mathbf{y}_i, \mathbf{z}_i\}_{i \in [n]}$ and a map $\pi : [n] \to A$, where A is an attribute set and $n = |A|$. If there is an attribute vector $\mathbf{x} = (x_{\pi(1)}, x_{\pi(2)}, \cdots, x_{\pi(n)}) \in \mathbb{Z}_p^n$ which satisfies the ASP $(\mathcal{V}, \pi)$, it is able to get $\gamma_1, \cdots, \gamma_n \in \mathbb{Z}_p$ such that $\sum\limits_{i=1}^{n} \gamma_i(\mathbf{y}_i + x_{\pi(i)}\mathbf{z}_i) = (1, 0, \cdots, 0)$.

Theorem 2. For any attribute set S that is not satisfied with $\mathcal{V} = \{(\mathbf{y}_j, \mathbf{z}_j)\}_{j \in S}$, the distributions
$\{s, \mathbf{y}_j \left(\begin{smallmatrix} l_0 s \\ \mathbf{L} \end{smallmatrix} \right) + r_j p_j, \mathbf{z}_j \left(\begin{smallmatrix} l_0 s \\ \mathbf{L} \end{smallmatrix} \right) + r_j p_j', s_j\}_{j \in S}, (\{\alpha + r l_0, r, r p_j\})_{j \in S}$ perfectly hide α, where $p_j, p_j', l_0, s, r, s_j \leftarrow_R \mathbb{Z}_p, \mathbf{L} \leftarrow_R \mathbb{Z}_p^{l'-1}$, and $r_j \neq 0$.

One-Use Restriction. Similar to the ideas of [25,30], for $\forall (\mathbf{y}_i, \mathbf{z}_i), (\mathbf{y}_j, \mathbf{z}_j) \in \mathcal{V}$, if $i \neq j$, then these two pairs of vectors correspond to different attributes.

4   Revocable CP-ABE for Arithmetic Span Programs for Cloud-Assisted IoT

4.1   System and Threat Models

The responsibility of each entity involved is described as below:

- **Key generate center** is the manager of the whole system. It is responsible for generating long-term key materials for each users and broadcast key update information to non-revoked users.
- **Smart devices** are data collectors. They can be various wearable devices or personal health monitor. Each smart devices will continuously gather the personal information and uploaded the encrypted data to the cloud server.
- **Cloud server** is a powerful third-party entity for mitigating the heavy burden of data storage and management for data users.
- **Data user** is the entity that is authorized to access the shared data. Each non-revoked data user can periodically receive the key update information from key generate center to synthesize a complete decryption key.

In the R-ABE-ASP scheme, the cloud servers that are responsible for storing the user's data may curious-but-honest. It may snoop user's personal information while honestly response for user's request. Besides, the users whose attributes mismatch the access policy or has been revoked may also make efforts to eavesdrop user's sensitive data.

4.2   Design Goals

In this paper, we plan to design a secure R-CPABE-ASP scheme that can not only achieve user revocation but also enable the efficient description of access policy. For this purpose, the R-CPABE-ASP should achieve the following goals.

– Data confidentiality: The data stored on the cloud servers can only be decrypted by authorized users. It should be inaccessible to the adversary defined in the threat model.
– Efficiency: The data is supposed to be stored and shared in an efficient manner. The overhead for smart devices to encrypt the sensitive data and the cost for data users to access corresponding data should be carried out in a low-cost manner.
– Reliable access control: Considering the complex application scenarios in cloud-assisted IoT, the user revocation mechanism is also necessary except for fine-grained access control to sensitive data.

4.3   Outline of the R-CPABE-ASP

Different from the CP-ABE scheme with other access structures like LSSS, an attribute vector $\mathbf{x}$ corresponding to the attribute set S is needed in the R-CPABE-ASP scheme, which controls the various relationship between attributes and access structure. And now, we show the outline of a R-CPABE-ASP scheme that is formed with seven algorithms as below.

Setup(λ): This algorithm is executed by the key generate center to initialize the system. Taking a security parameter λ as input, the algorithm outputs the public parameter pp and the master secret key msk.

AttrKeyGen(st, $pp, msk, S, \mathbf{x}, $ID): This algorithm is executed by the key generate center to generate the attribute related key for data users. Taking the state information st, public parameter pp, the master secret key msk, an attribute set S, an attribute vector $\mathbf{x}$ and unique credential ID of a user as input, this algorithm outputs the corresponding attribute key AK_{ID}.

KeyUpdate(st, $msk, t, $REL): This algorithm is executed by the key generate center to produce the key update materials for each non-revoked user. Taking the state information st, the master secret key msk, current time period t and revocation list REL as input, this algorithm outputs the key update information KU_t.

KeyGen($pp, AK_{\mathsf{ID}}, KU_t$): This algorithm is executed by each users to produce the complete decryption key. Taking the public parameter pp, the attribute key AK_{ID} of a user and the key update information UK_t as input, this algorithm outputs the whole secret key sk_t.

Encrypt($pp, (\mathcal{V}, \pi), msg, t$): This algorithm is executed by various smart devices. Taking the public parameter pp, an arithmetic span program $(\mathcal{V}, \pi)$, a message msg and current time period t as input, the algorithm outputs the ciphertext ct_t under time period t.

Decrypt($ct_t, sk_{t'}, (\mathcal{V}, \pi)$): This algorithm is carried out by the authorized data users to access the shared data. Taking a ciphertext ct_t, the secret key $sk_{t'}$ and an arithmetic span program $(\mathcal{V}, \pi)$, the algorithm outputs the message msg or a symbol $\perp$ for a failure decryption.

Revoke(REL, ID, t): This algorithm is carried out by key generate center to revoke the compromised users. Taking the revocation list REL, the credential ID of a user and current time period t, this algorithm updates the REL by adding a tuple (ID, t).

4.4   Construction

The detailed description of the proposed puncturable ciphertext-policy attribute-based encryption scheme for arithmetic span program is presented as below.

- Setup(λ, ℓ, N): On input a security parameter λ, the maximum length of the time period ℓ and the maximum users of the system N, this algorithm sets a bilinear group generator $\mathcal{G}$ and computes $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h) \leftarrow \mathcal{G}(\lambda)$. In the tuple $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h)$, there exists a map $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$, and g, h are the generators of $\mathbb{G}$ and $\mathbb{H}$, respectively. In addition, this algorithm selects a hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$ and some parameters $\mathbf{C}_1 \leftarrow_R \mathbb{Z}_p^3, \mathbf{D} \leftarrow_R \mathbb{Z}_p^2, \mathbf{K}, \mathbf{K}_0, \mathbf{K}_1, \mathbf{K}', \mathbf{K}_0', \mathbf{K}_1', \mathbf{L}_0, \boldsymbol{\vartheta}_1, \cdots, \boldsymbol{\vartheta}_\ell \leftarrow_R \mathbb{Z}_p^{3\times2}, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \leftarrow_R \mathbb{Z}_p^3$ randomly. Then, this algorithm picks a binary tree BT with N leaf nodes that are used to store the information of users and initialize an empty list REL to record the revoked user's credential, as well as the state information $\mathsf{st} = \mathsf{BT}$. Finally, this algorithm outputs the public parameter as $pp = (g, h, H, g^{\mathbf{C}_1^\top}, g^{\mathbf{C}_1^\top \mathbf{K}}, g^{\mathbf{C}_1^\top \mathbf{K}_0}, g^{\mathbf{C}_1^\top \mathbf{K}_1}, g^{\mathbf{C}_1^\top \mathbf{K}'}, g^{\mathbf{C}_1^\top \mathbf{K}_0'}, g^{\mathbf{C}_1^\top \mathbf{K}_1'}, g^{\mathbf{C}_1^\top \mathbf{L}_0}, e(g^{\mathbf{C}_1^\top}, h^{\boldsymbol{\alpha}_1}), e(g^{\mathbf{C}_1^\top}, h^{\boldsymbol{\alpha}_2}), \boldsymbol{\vartheta}_1, \cdots, \boldsymbol{\vartheta}_\ell)$ and the master secret key as $msk = (\mathbf{C}_1, \mathbf{D}, \mathbf{K}, \mathbf{K}_0, \mathbf{K}_1, \mathbf{K}', \mathbf{K}_0', \mathbf{K}_1', \mathbf{L}_0, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$.

- AttrKeyGen($\mathsf{st}, pp, msk, S, \mathbf{x}, \mathsf{ID}$): On input the state information st, public parameter pp, the master secret key msk, an attribute set S, an attribute vector $\boldsymbol{x}$ and the unique credential ID of the user that generated by the system when the user joins for the first time, this algorithm arbitrarily assigns an unused leaf node δ of BT to store the information of this user. After that, for each node $\varrho \in \mathsf{Path}(\delta)$, this algorithm picks an random vector $\boldsymbol{\xi}_\varrho$ and stores it in this node. Then, this algorithm further calculates $IK_\varrho = h^{\boldsymbol{\xi}_\varrho}$. Subsequently, this algorithm samples $\mathbf{b}, \{\mathbf{b}_y, \mathbf{b}_y'\}_{y \in S} \leftarrow_R \mathsf{span}(\mathbf{D})$ and then computes

$$K_0 = h^{\mathbf{b}}, \{K_{0,y} = h^{\mathbf{b}_y}, K_{0,y}' = h^{\mathbf{b}_y'}\}_{y \in S},$$

$$\{K_{1,y} = h^{(\mathbf{K}+x_y\mathbf{K}')\mathbf{b}+(\mathbf{K}_0+H(y)\mathbf{K}_1)\mathbf{b}_y+x_y(\mathbf{K}_0'+H(y)\mathbf{K}_1')\mathbf{b}_y'}\}_{y \in S}, K_2 = h^{\boldsymbol{\alpha}_1 - \mathbf{L}_0\mathbf{b}}.$$

  Finally, this algorithm outputs the attribute key $AK_{\mathsf{ID}} = \{K, IK_\varrho\}_{\varrho \in \mathsf{Path}(\delta)}$ where $K = (K_0, \{K_{0,y}, K_{0,y}', K_{1,y}\}_{y \in S}, K_2)$.

- KeyUpdate($\mathsf{st}, msk, t, \mathsf{REL}$): For each node $\varrho \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, t)$, this algorithm retrieves the random vector $\boldsymbol{\xi}_\varrho$ stored from the node ϱ. After that, it generates the key update information as follows:

$$KU_{\varrho,0} = h^{\boldsymbol{\alpha}_2 - \boldsymbol{\xi}_\varrho} \cdot h^{\sum\limits_{k \in \varUpsilon} \boldsymbol{\vartheta}_k \cdot \boldsymbol{\tau}_{\varrho,1}}, KU_{\varrho,1} = h^{\boldsymbol{\tau}_{\varrho,1}},$$

  where $\varUpsilon = \{k | t(k) = 0\}$ and $\boldsymbol{\tau}_{\varrho,1} \leftarrow_R \mathbb{Z}_p^{2\times1}$ is randomly picked. Finally, this algorithm outputs key update information $KU_t = \{\varrho, KU_{\varrho,0}, KU_{\varrho,1}\}_{\varrho \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, t)}$.

- Keygen($pp, AK_{\mathsf{ID}, KU_t}$): For each node $\varrho \in \mathsf{Path}(\delta) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, t)$, this algorithm picks $\boldsymbol{\tau}_1 \leftarrow_R \mathbb{Z}_p^{2\times1}$ and calculates the time-related decryption key

$$K_{\mathsf{ID},t} = (IK_\varrho \cdot KU_{\varrho,0} \cdot h^{\sum\limits_{k \in \varUpsilon} \boldsymbol{\vartheta}_k \cdot \boldsymbol{\tau}_1}, KU_{\varrho,1} \cdot h^{\boldsymbol{\tau}_1}),$$

where $\Upsilon = \{k|t(k) = 0\}$. Finally, this algorithm outputs the whole decryption key $sk_t = (K_0, \{K_{0,y}, K'_{0,y}, K_{1,y}\}_{y \in S}, K_2, K_{\mathsf{ID},t})$.

– Encrypt$(pp, (\mathcal{V}, \pi), msg, t)$: Note that, $\mathcal{V} = (\mathbf{y}_j, \mathbf{z}_j)_{j \in [n]}$, and π is a map from $(\mathbf{y}_j, \mathbf{z}_j)$ to A. Let l denote the length of the vectors in $(\mathcal{V}, \pi)$. After receiving the public parameter pp, an arithmetic span program $(\mathcal{V}, \pi)$ satisfying with an attribute set A and a message $msg \in \mathbb{G}_T$, this algorithm performs below. This algorithm samples $\mathbf{L} \leftarrow_R \mathbb{Z}_p^{(l-1) \times 2}$, $s, \{s_j\}_{j \in [n]}$, $s_t \leftarrow_R \mathbb{Z}_p$ and then computes

$$C_0 = g^{s\mathbf{C}_1^\top}, C_{0,j} = g^{s_j \mathbf{C}_1^\top}, C_{1,j} = g^{\mathbf{y}_j \left(s\mathbf{C}_1^\top \mathbf{L}_0 \mathbf{L} \right) + s_j \mathbf{C}_1^\top \mathbf{K}},$$

$$C'_{1,j} = g^{\mathbf{z}_j \left(\begin{smallmatrix} s\mathbf{C}_1^\top \mathbf{L}_0 \\ \mathbf{L} \end{smallmatrix} \right) + s_j \mathbf{C}_1^\top \mathbf{K}'}, C_{2,j} = g^{s_j \mathbf{C}_1^\top (\mathbf{K}_0 + H(\pi(j)) \cdot \mathbf{K}_1)},$$

$$C'_{2,j} = g^{s_j \mathbf{C}_1^\top (\mathbf{K}'_0 + H(\pi(j)) \cdot \mathbf{K}'_1)}, C_3 = msg \cdot e(g, h)^{s\mathbf{C}_1^\top \boldsymbol{\alpha}_1} \cdot e(g, h)^{s_t \mathbf{C}_1^\top \boldsymbol{\alpha}_2},$$

$$C_4 = g^{-s_t \cdot \mathbf{C}_1^\top}, C_5 = g^{\mathbf{C}_1^\top \sum\limits_{k \in \Upsilon} \vartheta_k \cdot s_t}, \Upsilon \in \{k|t(k) = 0\}.$$

Finally, this algorithm outputs the ciphertext
$ct_t = (C_0, \{C_{0,j}, C_{1,j}, C_{2,j}, C'_{1,j}, C'_{2,j}\}_{j \in [n]}, C_3, C_4, C_5)$ under time period t.

– Decrypt$(ct_t, sk_{t'}, (\mathcal{V}, \pi))$: This algorithm aborts as $t' < t$. Or else, if the attribute vector $\mathbf{x}$ satisfies the arithmetic span program $(\mathcal{V}, \pi)$, this algorithm computes $\gamma_1, \gamma_2, \cdots \in \mathbb{Z}_p$ such that $\sum_{j \in [n]} \gamma_j(\mathbf{y}_j + x_{\pi(j)} \mathbf{z}_j) = (1, 0, \cdots, 0)$. On input a ciphertext associated and a secret key sk, this algorithm computes

$$Y_j = e(C_{1,j} \cdot C'^{x_{\pi(j)}}_{1,j}, K_0) \cdot e(C_{2,j}, K_{0,\pi(j)}) \cdot$$
$$e(C'^{x_{\pi(j)}}_{2,j}, K'_{0,\pi(j)}) \cdot e(C_{0,j}, K_{1,\pi(j)})^{-1}$$
$$A = e(C_0, K_2) \cdot \prod_{j \in [n]} Y_j^{\gamma_j}, B = e(C_4, K_{\mathsf{ID},t',1}) \cdot e(C_5, K_{\mathsf{ID},t',2}), \Upsilon = \{k|t(k) = 0\},$$

and recovers the message as $msg = \frac{C_3 \cdot B}{A}$.

– Revoke$(\mathsf{REL}, \mathsf{ID}, t)$: After receiving a tuple (ID, t), this algorithm updates REL by inserting this tuple to revocation list and outputs updated list REL.

Limited by the space, the detailed security proof of R-CPABE-ASP is given in the full version of our paper [31], and omitted here.

4.5   Outsourced Version of R-CPABE-ASP Scheme

In this section, a outsourced R-CPABE-ASP (OR-CPABE-ASP) scheme is proposed. The OR-CPABE-ASP scheme consists the following 8 algorithms:

The Setup, AttrKeyGen, Keygen, Encrypt and Revoke algorithms are the same as R-CPABE-ASP scheme. The only difference is the outsourced R-CPABE-ASP scheme has following three algorithms:

Keygen.rand(sk_t): The algorithm samples $\tau \in \mathbb{Z}_p$ and computes $\bar{K}_0 = K_0^\tau, \{\bar{K}_{0,y} = K_{0,y}^\tau, \bar{K}_{0,y}' = K_{0,y}'^\tau, \bar{K}_{1,y} = K_{1,y}^\tau\}_{y \in S}, \bar{K}_2 = K_2^\tau, \bar{K}_{\mathsf{ID},t} = K_{\mathsf{ID},t}^\tau$. Finally, it sets $\bar{sk}_t = \{\bar{K}_0, \{\bar{K}_{0,y}, \bar{K}_{0,y}', \bar{K}_{1,y}\}_{y \in S}, \bar{K}_2, \bar{K}_{\mathsf{ID},t}\}$ as the conversion key and outputs $(\bar{sk}_t, \tau)$ as the retrieval key.

Decrypt.out($ct_t, \bar{sk}_{t'}, (\mathcal{V}, \pi)$): This algorithm aborts as $t' < t$. Or else, if the attribute vector $\mathbf{x}$ satisfies the arithmetic span program $(\mathcal{V}, \pi)$, this algorithm computes $\gamma_1, \gamma_2, \cdots \in \mathbb{Z}_p$ such that $\sum_{j \in [n]} \gamma_j (\mathbf{y}_j + x_{\pi(j)} \mathbf{z}_j) = (1, 0, \cdots, 0)$. On input a ciphertext associated and a secret key sk, this algorithm computes

$$A = e(C_0, \bar{K}_2) \cdot \prod_{j \in [n]} \left[\left(e(C_{1,j} \cdot C_{1,j}'^{x_{\pi(j)}}, \bar{K}_0) \cdot e(C_{2,j}, \bar{K}_{0,\pi(j)}) \right) \right]^{\gamma_j}$$
$$\cdot \left[e(C_{2,j}'^{x_{\pi(j)}}, \bar{K}_{0,\pi(j)}') \cdot e(C_{0,j}, \bar{K}_{1,\pi(j)})^{-1}) \right]^{\gamma_j},$$
$$B = e(C_4, \bar{K}_{\mathsf{ID},t',1}) \cdot e(C_5, \bar{K}_{\mathsf{ID},t',2}), \Upsilon = \{k | \tilde{t}(k) = 0\}, \bar{C}_3 = C_3, \bar{C}' = \frac{B}{A}$$

and outputs $\bar{ct} = (\bar{C}_3, \bar{C}')$

Decrypt.user($\bar{ct}, \tau$): The algorithm recover the message as $msg = (\bar{C}_3/\bar{C})'^{-\tau}$.

4.6 Rationales Discuss

– Data confidentiality: according to the detailed security analysis displayed in the full version of our paper [31], our R-CPABE-ASP is adaptively secure against an adversary who has no corresponding attribution set or was revoked in the prior time period. Thereby, the presented R-CPABE-ASP scheme meets the requirement of data confidentiality.
– Efficiency: because of the introduction of ASP access structure, the access policy involved in the encryption phase can be defined in a more efficient way than the works adopting normal access structure. Hence, encryption efficiency is significantly improved by reducing the redundant description of the access policy. Meanwhile, as shown in the outsourced version of the R-CPABE-ASP, the decryption cost for data users is merely one exponential operation. By considering this, both the encryption efficiency and decryption efficiency are effectively assured in R-CPABE-ASP.
– Reliable access control: the compromised user can be efficiently revoked in the R-CPABE-ASP since the adoption of the user revocation mechanism. Hence, the reliability of data access is assured in R-CPABE-ASP.

5 Performance Evaluation

After the concrete construction of our OR-CPABE-ASP, an outsourced version of the OR-CPABE-ASP is given to mitigate the heavy decryption burden for users in our scheme. In this part, we mainly consider the OR-CPABE-ASP version for meeting the requirement of efficiency in cloud-assisted IoT scenarios. For making

Table 1. Properties and efficiency comparison among the works

Scheme	QZZC [20]	XYML [21]	CDLQ [22]	OR-CPABE-ASP
Revocation model	Indirect	Indirect	Indirect	Indirect
Security	Selective	Selective	Selective	Adaptive
Decryption key size	$(2|S|+5)|\mathbb{G}|$	$(2|S|+3)|\mathbb{G}|$	$(2|S|+3)|\mathbb{G}|$	$(10+7|S|)|\mathbb{H}|$
Ciphertext size	$|\mathbb{G}|+(3|A|+2)|\mathbb{G}|$	$|\mathbb{G}_T|+(3|A|+|V|+2)|\mathbb{G}|$	$|\mathbb{G}_T|+(3|A|+2)|\mathbb{G}|$	$9(1+|A|)|\mathbb{G}|$
Update key size	$2|\mathbb{G}|$	$2|\mathbb{G}|$	$2|\mathbb{G}|$	$5|\mathbb{G}|$
Encryption cost	$E_T+(4|A|+2)E_G$	$E_T+(5|A|+|V|+2)E_G$	$E_T+(4|A|+2)E_G$	$2E_T+(11|A|+8)E_G$
Decryption cost	$2P$	$(3|A|+2)P+|A|E_T$	E_T	E_T
Key update cost	$2E_G$	$(|S|+4)E_G$	$2E_G$	$5E_H$

an overall performance evaluation in terms of our OR-CPABE-ASP and the state-of-the-art works[1], both theoretical analysis and experimental simulations are carried out.

5.1   Theoretical Analysis

The detailed comparisons in terms of essential properties, communication and computation efficiency are summarized in Table 1. In this table, $|S|$ represents the number of attributes held by the user; $|A|$ is the size of attributes involved in the access policy; $|V|$ is the size of bits with the value of 0 in a time period string; $|\mathbb{G}|$, $|\mathbb{H}|$ and $|\mathbb{G}_T|$ denote the size of one element in group $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{G}_T$; $E_{\mathbb{G}}$, $E_{\mathbb{H}}$ and $E_{\mathbb{T}}$ present one exponential operations in groups $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{G}_T$, separately.

From the results of Table 1, all the works of QZZC [20], XYML [21], CDLQ [22] and ours adopt the indirect model to achieve user revocation. According to the results displayed in Table 1, it is fair to make a summary that both the works of QZZC [20] and CDLQ [22] outperform our OR-CPABE-ASP work in decryption key size, ciphertext size, update key size, data encryption efficiency and key update efficiency. Nonetheless, the works of QZZC [20] and CDLQ [22] only achieves selective security, which is too strict to handle the real-life security threats. Furthermore, our OR-CPABE-ASP scheme is more efficient in terms of data decryption than that in work of QZZC [20]. In QZZC [20], users need to carry out two heavy pairing operations to access the data stored on the cloud servers. While the cost for users to access the data is merely one exponential operation in our work. On the other hand, it is straightforward to see that the OR-CPABE-ASP scheme performs well in terms of security, ciphertext size, encryption efficiency and key update efficiency compared with XYML [21]. In XYML [21], the overhead for data encryption is affected by the value of current time period. The time period in their work is re-encoded as a binary string. During the phase of data encryption, the bits with a value of 0 in the time period string will induce extra exponential operations. In the worst case that all

---

[1] Denote the works of Qin *et al.* [20], Xu *et al.* [21] and Cui *et al.* [22] by QZZC, XYML and CDLQ, respectively.
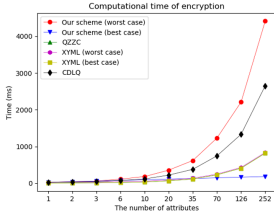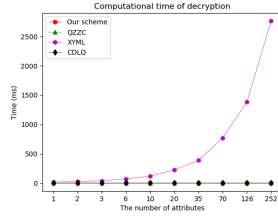
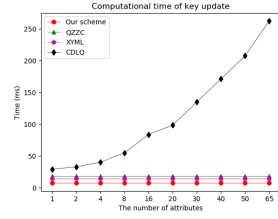**Fig. 1.** Encryption cost   **Fig. 2.** Decryption cost   **Fig. 3.** Key update cost

bits of the time period string are with the value of 0, the resulting overheads for data encryption and ciphertext delivery will come to an unacceptably high level. By considering this, only our proposed OR-CPABE-ASP work is secure, efficient and feasible in the complex real-life cloud-assisted IoT scenarios.

5.2 Experimental Simulations

To discuss the feasibility and efficiency of the OR-CPABE-ASP work in a more comprehensive manner, the simulations for testing the computation efficiency are conducted in this section. We implement all the works with the PBC library on a Win 10 operation system installed with an Core i7-7700 @3.60 GHz processor and 8 G RAM. The results of simulations are shown in Fig. 1, Fig. 2 and Fig. 3.

The data encryption efficiency comparison among the works is shown in Fig. 1. In particular, we simulate XYML [21] and our OR-CPABE-ASP work under the extreme conditions. In XYML [21], data encryption cost is related to the length of bits with value of 0 in the current time period string. In this way, as all the bits are with value of 1, their work enjoy the best efficiency in terms of data encryption. On the contrary, if all the bits are with value of 0, the cost for data encryption is at ist maximum. Since the more expressive ASP access structure is introduced in our proposed work, hence there are also two cases in the data encryption phase of OR-CPABE-ASP work. The reason is that our OR-CPABE-ASP work is a ciphertext-policy variant of ABE. The data encryption cost will be affected by the expressiveness of the access structure. For instance, the access structure specifies that the user who has the 5 of 10 attributes is able to access the data. In traditional access structure, such as LSSS, all the possible attribute combinations are traversed to test whether a user owns the corresponding attributes (the worst case of our OR-CPABE-ASP work). However, in ASP access structure, the access policy can be easily described as the arithmetic expression results of these attributes. In other words, the involved attributes are merely required once such that the cost for defining the access policy is significantly reduced (the best cast of our work). Obviously, the OR-CPABE-ASP work is more efficient in data encryption than the state-of-the-art works [20–22] according to Fig. 1. It is easy to see that our OR-CPABE-ASP work is almost unaffected by the number of attributes involved in access policy.

Compared with the existing works [20–22] where the encryption overhead increases dramatically with the growth of the attribute's number, the OR-CPABE-ASP scheme is more applicable to the cloud-assisted IoT scenarios where the access policy for ciphertext may be complicated.

The cost for data decryption is shown in Fig. 2. From the details displayed in Fig. 2, the cost for data decryption in XYML [21] is affected by the number of attributes involved. In a sharp contrary, QZZC [20], CDLQ [22] and the OR-CPABE-ASP work keep a low and constant overhead in data decryption. A similar trend is also demonstrated during the phase of key update as shown in Fig. 3. It is worth noting that the OR-CPABE-ASP scheme owns the best efficiency in terms of the key update for each non-revoked user.

As described above, our proposed OR-CPABE-ASP work outperforms the state-of-the-art works in both the efficiency of encryption and decryption, which makes the possibility of fast data storage and convenient data access in our work. Taking into this account, our work is expected to achieve efficient data storage and sharing in cloud-assisted IoT environments compared with the state-of-the-art works utilizing similar underlying primitive.

6   Conclusion

In this paper, we construct the first revocable CP-ABE for arithmetic span programs (R-CPABE-ASP) for cloud-assisted IoT. Compared with existing revocable ABE works, the proposed R-CPABE-ASP scheme has the ability to describe the complicated access policy in an easier manner by adopting arithmetic span programs access structure. Hence, the data encryption efficiency in the R-CPABE-ASP work can be significantly enhanced. Meanwhile, we also show how to mitigate the heavy data decryption burden for users by giving an outsourced version of the R-CPABE-ASP scheme. Finally, the detailed theoretical analysis and experimental simulations demonstrate that our R-CPABE-ASP is secure and efficient in cloud-assisted IoT.

References

1. Farhan, M., et al.: IoT-based students interaction framework using attention-scoring assessment in elearning. Future Gener. Comput. Syst. **79**, 909–919 (2018)
2. Yu, S.: Big privacy: challenges and opportunities of privacy study in the age of big data. IEEE Access **4**, 2751–2763 (2016)
3. Yu, S., Liu, M., Dou, W., Liu, X., Zhou, S.: Networking for big data: a survey. IEEE Commun. Surv. Tutorials **19**(1), 531–549 (2016)

4. Xiong, H., et al.: Heterogeneous signcryption with equality test for IIoT environment. IEEE Internet Things J. (2020). https://doi.org/10.1109/JIOT.2020.3008955

5. Xiong, H., Bao, Y., Nie, X., Asoor, Y.I.: Server-aided attribute-based signature supporting expressive access structures for industrial Internet of Things. IEEE Trans. Ind. Inform. **16**(2), 1013–1023 (2019)

6. Xiong, H., Wu, Y., Jin, C., Kumari, S.: Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT. IEEE Internet of Things J. (2020). https://doi.org/10.1109/JIOT.2020.2999510

7. Belguith, S., Kaaniche, N., Russello, G.: PU-ABE: lightweight attribute-based encryption supporting access policy update for cloud assisted IoT. In: 11th IEEE International Conference on Cloud Computing, CLOUD 2018, San Francisco, CA, USA, 2–7 July 2018, pp. 924–927. IEEE Computer Society (2018)

8. Lee, C., Chung, P., Hwang, M.: A survey on attribute-based encryption schemes of access control in cloud environments. I. J. Netw. Secur. **15**(4), 231–240 (2013)

9. Fu, A., Yu, S., Zhang, Y., Wang, H., Huang, C.: NPP: a new privacy-aware public auditing scheme for cloud data sharing with group users. IEEE Trans. Big Data (2017). https://doi.org/10.1109/TBDATA.2017.2701347

10. Mei, Q., Xiong, H., Chen, J., Yang, M., Kumari, S., Khan, M.K.: Efficient certificateless aggregate signature with conditional privacy preservation in IoV. IEEE Syst. J. (2020). https://doi.org/10.1109/JSYST.2020.2966526

11. Sahai, A., Waters, B.: Fuzzy identity based encryption. IACR Cryptology ePrint Archive, vol. 2004, p. 86 (2004)

12. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006), pp. 89–98 (2006)

13. Kaaniche, N., Laurent, M.: Privacy-preserving multi-user encrypted access control scheme for cloud-assisted IoT applications. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 590–597. IEEE (2018)

14. Li, J., Zhang, Y., Ning, J., Huang, X., Poh, G.S., Wang, D.: Attribute based encryption with privacy protection and accountability for cloudIoT. IEEE Trans. Cloud Comput. (2020). https://doi.org/10.1109/TCC.2020.2975184

15. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. IACR Cryptology ePrint Archive vol. 2006, p. 309 (2006)

16. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F., (eds.) Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, 28–31 October 2007, pp. 195–203. ACM (2007)

17. Xiong, H., Choo, K.-K.R., Vasilakos, A.V.: Revocable identity-based access control for big data with verifiable outsourced computing. IEEE Trans. Big Data (2017). https://doi.org/10.1109/TBDATA.2017.2697448

18. Zhang, Y., Chen, X., Li, J., Li, H., Li, F.: FDR-ABE: attribute-based encryption with flexible and direct revocation. In: 2013 5th International Conference on Intelligent Networking and Collaborative Systems, pp. 38–45. IEEE (2013)

19. Wu, A., Zheng, D., Zhang, Y., Yang, M.: Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing. Sensors **18**(7), 2158 (2018)

20. Qin, B., Zhao, Q., Zheng, D., Cui, H.: (dual) server-aided revocable attribute-based encryption with decryption key exposure resistance. Inf. Sci. **490**, 74–92 (2019)

21. Xu, S., Yang, G., Mu, Y., Liu, X.: A secure IoT cloud storage system with fine-grained access control and decryption key exposure resistance. Future Gen. Comput. Syst. **97**, 284–294 (2019)

22. Cui, H., Deng, R.H., Li, Y., Qin, B.: Server-aided revocable attribute-based encryption. In: Askoxylakis, I., Ioannidis, S., Katsikas, S., Meadows, C. (eds.) ESORICS 2016. LNCS, vol. 9879, pp. 570–587. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45741-3_29

23. Wei, J., Chen, X., Huang, X., Hu, X., Susilo, W.: RS-HABE: revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud. IEEE Trans. Dependable Secur. Comput. (2019). https://doi.org/10.1109/TDSC.2019.2947920

24. Attrapadung, N., Hanaoka, G., Yamada, S.: Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 575–601. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_24

25. Chen, J., Gong, J., Kowalczyk, L., Wee, H.: Unbounded ABE via bilinear entropy expansion, revisited. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 503–534. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_19

26. Ishai, Y., Wee, H.: Partial garbling schemes and their applications. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8572, pp. 650–662. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43948-7_54

27. Ma, C., Gao, H., Wei, D.: A CP-ABE scheme supporting arithmetic span programs. Secur. Commun. Netw. **2020**, 1–16 (2020)

28. Attrapadung, N.: Dual system encryption via doubly selective security: framework, fully secure functional encryption for regular languages, and more. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_31

29. Escala, A., Herold, G., Kiltz, E., Rafols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. J. Crypto. **30**(1), 242–288 (2017)

30. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_20

31. Xiong, H., Chen, J., Yang, M., Huang, X.: Revocable attribute-based encryption scheme with arithmetic span program for cloud-assisted IoT. IACR Cryptology ePrint Archive, 2020:553 **2020**, 553 (2020)