EPiC Series in Computing

Volume 106, 2025, Pages 155–164

Proceedings of the 20th Asia Joint Conference on Information Security

Registered Key-Policy Attribute-Based Encryption

Shih-Pei Kao[1], Yi-Fan Tseng[1]*, Tien-Lin Tsai[1], Yi-Jiin Lu[1], Jheng-Jia Huang[2], Guan-Yu Chen[2], and Wei-Hsueh Wang[2]

[1] National Chengchi University, Taipei, Taiwan
[2] National Taiwan University of Science and Technology, Taipei, Taiwan

Abstract

Attributed-based encryption (ABE) enables fine-grained access control over encrypted data. However, ABE requires a single trusted authority to issue decryption keys, which makes ABE have a key-escrow problem. If an adversary breaks through the system, then the adversary can decrypt all ciphertext encrypted through the system. In this work, we generalize the notion of registration-based encryption (RBE) to key-policy attributed-based encryption (KP-ABE). Through the introduction of RBE, users can autonomously generate their own keys, thereby effectively resolving the key-escrow problem of KP-ABE.

1 Introduction

Traditional public key encryption (PKE) [6,18] and identity-based encryption (IBE) [1,20] only provide coarse-grained access control, which means that a ciphertext can only be decrypted by a specific user. However, in practical applications, there may be a group of recipients who need access to the ciphertext, or the recipient may have an unknown identity. In such scenarios, PKE and IBE with coarse-grained access control are not suitable. To address this issue, Sahai and Waters et al. [13, 19] introduced the concept of Attribute-Based Encryption (ABE).

ABE extends traditional public-key encryption by enabling fine-grained access control for encrypted data through the implementation of authorization policies. Its capabilities allow ABE to be applied across diverse domains such as cloud computing [7,17,23] , Internet of Things (IoT) [4,21], and personal health records (PHR) [15, 24]. ABE can generally be divided into two categories: ciphertext-policy attribute-based encryption (CP-ABE) and key-policy attribute-based encryption (KP-ABE).

In CP-ABE, ciphertexts are associated with policies, and secret keys are associated with sets of attributes. A ciphertext ct_P embedded with a policy P can be decrypted by a secret key sk_x embedded with an attribute set x only if the attributes in the key satisfy the embedded policy in the ciphertext (i.e., $P(x) = 1$). Conversely, in a KP-ABE scenario, secret keys are associated with policies, and ciphertexts are associated with sets of attributes. A secret key sk_P embedded with a policy P can decrypt a ciphertext ct embedded with an attribute set x only if the attribute in the ciphertext matches the embedded policy in the key (i.e., $P(x) = 1$). CP-ABE and KP-ABE have been extensively researched in cloud computing, PHR, and IoT. Their fundamental distinction lies in their primary objectives: CP-ABE

*Corresponding Author: yftseng@g.nccu.edu.tw

focuses on associating access policies directly with ciphertexts, ensuring that only users meeting specific attribute-based or role-based policies can decrypt and access data. In contrast, KP-ABE emphasizes embedding complex access policies within cryptographic keys. This approach enables precise control over data access based on the data's attributes themselves, thereby enhancing security and facilitating fine-grained access control in scenarios where stringent regulation of data access based on its attributes is critical.

While ABE permits fine-grained access control over encrypted data, it alters the trust model in contrast to standard public-key encryption. In standard public-key encryption, users can independently generate their public and private keys, thus preventing the establishment of a central point of failure. However, in ABE, a single trusted central authority is necessary to issue and maintain decryption keys for users. If an adversary compromises this central authority, they can decrypt all encrypted ciphertext within the system, resulting in a key-escrow problem in KP-ABE. Several methods have been proposed to address the key-escrow problem, such as multi-authority attribute-based encryption (MA-ABE) [2], distributed attribute-based encryption (DABE) [16], registration-based encryption (RBE) [10], etc.

To address the problem of relying on a single trusted authority in ABE schemes, Chase [2] proposed the concept of MA-ABE in 2007. MA-ABE allows multiple different authorities to operate simultaneously, with each authority responsible for managing the key distribution of different attributes. In MA-ABE, the sender can specify a set of attributes and the required number of attributes for each authority, ensuring that only users with a sufficient number of attribute keys can decrypt the message. This allows the sender to flexibly set decryption conditions, ensuring that only specific qualified recipients can decrypt the message. The MA-ABE scheme overcomes the limitation of a single trusted authority, providing a more flexible and secure method of attribute-based encryption with multiple authorities.

Similarly, to address the issue of relying on a single trusted authority in ABE schemes. Müller et al. [16] proposed the concept of DABE in 2008. In a DABE scheme, there are three distinct entities: a master entity, attribute authorities, and users. The master is responsible for distributing secret user keys, while attribute authorities verify if users possess specific attributes and generate corresponding secret attribute keys. Each attribute authority has complete control over its attributes and generates public attribute keys for each attribute. Eligible users receive personalized secret attribute keys through a trusted channel for decrypting ciphertexts. This design not only addresses the key escrow problem in ABE but also provides a more suitable solution for scenarios requiring multiple authoritative entities.

In 2019, Garg et al. [10] introduced the concept of RBE as a solution to the key-escrow problem in IBE. In an RBE scheme, the central authority is substituted with a "key curator". The key curator is not responsible for issuing secret decryption keys; instead, it aggregates registered users' public keys into a short mpk (master public key). In the RBE scheme, users generate their public and secret keys, and submit their public keys and related information to the key curator for registration. The key curator updates the mpk when performing registrations. When sending messages, the sender encrypts the plaintext using the mpk. Registered users can decrypt messages using their secret key along with a helper decryption key that receives regular updates. Since the mpk changes when new users join, everyone needs to periodically refresh their helper decryption keys. Importantly, in the RBE system, the key curator does not possess any secret information; all computations are public.

All three encryption schemes address the key-escrow problem. However, unlike the former two, RBE eliminates the need for a central authority, allowing users to generate their own keys without incurring additional costs to maintain a central or attribute authority.

1.1 Related Work

In 2018, to address the key-escrow problem in IBE, Garg et al. [10] introduced the notion of RBE into IBE. This allowed users to generate their own public and secret keys, thereby eliminating the need for

trust in the private-key generator (PKG). They used Merkle trees to accumulate the public keys, but this non-black-box method was prohibitively expensive. Therefore, in 2022, Glaeser et al. [12] proposed a construction for registered IBE using vector commitments, a form of black-box cryptography, to substitute for Merkle trees.

After the concept of RBE was introduced, an increasing number of studies on RBE have been published. In 2022, Hohenberger et al. [14] applied RBE to CP-ABE. They introduced a registered ABE scheme capable of accommodating general policies and an unlimited number of users, utilizing iO and statistically binding hash functions.

In 2023, Francati et al. [8] extended the concept of RBE to the domain of functional encryption (FE). They introduced an efficient scheme for registered FE, specifically for the case of inner-product predicates with attribute hiding.

Later in the same year, Zhu et al. [25] proposed a generic pairing-based registered ABE using predicate encoding from the k-Lin assumption. The predicate encodings they designed for include CP-ABE, zero inner-product, and IBE.

1.2 Our Contribution

We found that while many registered CP-ABE schemes have been proposed, there has been relatively little research on registered KP-ABE. To address this, we propose a registered key-policy attribute-based encryption (registered KP-ABE) scheme to solve the key-escrow problem associated with KP-ABE.

In summary, our work makes the following contributions:

- We follow the generic registered ABE scheme proposed by Zhu et al. [25], design the predicate encoding of KP-ABE, and use their construction to construct the registered KP-ABE scheme.
- We propose a KP-ABE with constant-size ciphertext.

2 Preliminaries

2.1 Notations

The notation conventions used in this work are as follows: We use lower-case boldface to denote row vectors over scalars (e.g., $\mathbf{x}$) and upper-case boldface to denote vectors of group elements, as well as matrices (e.g., $\mathbf{M}$). The symbol "$\|$" indicates vector or matrix concatenation (e.g., $(\mathbf{A}\|\mathbf{B})$). Additionally, we define $\mathbf{e}_1 = (1, 0, \ldots, 0)$, with the appropriate dimension implied by the context. For a finite set S, the expression $x \leftarrow S$ indicates that x is sampled from S, and for an algorithm $\mathcal{A}$, the expression $y \leftarrow \mathcal{A}$ indicates that y is the output produced by executing $\mathcal{A}$.

2.2 Registered KP-ABE

A Registered Key-Policy Attribute-Based Encryption (Registered KP-ABE) for predicate $P : X \times Y \rightarrow \{0, 1\}$ consists of six algorithms: $Setup$, $KeyGen$, Reg, $Encrypt$, $Update$, and $Decrypt$.

- $Setup(1^\lambda) \rightarrow crs$
  Given the security parameter 1^λ as input, the setup algorithm outputs a common reference string crs.
- $KeyGen(crs, aux) \rightarrow (pk, sk)$
  Given the common reference string crs and a (possibly empty) state aux as input, the key-generation algorithm outputs a public key pk and a secret key sk.

- $Reg(crs, aux, pk, P) \to (mpk, aux')$
  Given the common reference string crs, a state aux, a public key pk, and an access structure P associated with pk as input, the registration algorithm outputs a master public key mpk and an updated state aux'.
- $Encrypt(mpk, m, S) \to ct$
  Given the master public key mpk, a message m, and a set of attributes S associated with m as input, the encryption algorithm outputs a ciphertext ct.
- $Update(crs, aux, pk) \to hsk$
  Given the common reference string crs, a state aux, and a public key pk as input, the update algorithm outputs a helper decryption key hsk.
- $Decrypt(sk, hsk, ct) \to m/\perp/GetUpd$
  Given a secret key sk, a helper decryption key hsk, and a ciphertext ct as input, the decryption algorithm outputs a message m, a special symbol $\perp$ to indicate decryption failure, or a special flag $GetUpd$ indicating that an updated helper decryption key is needed for decryption.

2.3 Predicate Encodings

We fist review the concept of predicate encoding as described in [22], and adopt the formulation proposed in [25]. A predicate $P : X \times Y \to \{0,1\}$ is encoded using an (n, n_c, n_k)-predicate encoding. For all $x \in X, y \in Y$, exist $\mathbf{a}_y \in \mathbb{Z}_p^{1\times n_k}, \mathbf{K}_y \in \mathbb{Z}_p^{n\times n_k}, \mathbf{C}_x \in \mathbb{Z}_p^{n\times n_c}, \mathbf{d}_{x,y} \in \mathbb{Z}_p^{n_k\times n_c}$. Then we can compute $\mathbf{M}_{x,y} = \begin{pmatrix} \mathbf{a}_y & \mathbf{0}_{n_c} \\ \mathbf{K}_y & \mathbf{C}_x \end{pmatrix} \in \mathbb{Z}_p^{(1+n)\times(n_k+n_c)}$. That is to say, given P, we can compute n, n_c, and n_k; given x, we can calculate $\mathbf{C}_x$; given y, we can calculate $\mathbf{K}_y$ and $\mathbf{a}_y$; if both x and y are given, we can calculate $\mathbf{d}_{x,y}$. Additionally, the predicate encoding needs to satisfy two properties:

- α-reconstruction: For all $x \in X$ and $y \in Y$, if $P(x,y) = 1$ (meaning the following equation holds true), then α can be recovered: $\mathbf{M}_{x,y}\mathbf{d}_{x,y}^\intercal = \mathbf{e}_1^\intercal$.
- α-privacy: For all $x \in X$ and $y \in Y$, if $P(x,y) = 0$ (meaning the following equation holds true), then α can be hidden: For all $\alpha \in \mathbb{Z}_p^n$, $\{x, y, (\alpha\|\mathbf{w})\mathbf{M}_{x,y}\} \approx \{x, y, \alpha, (0\|\mathbf{w})\mathbf{M}_{x,y}\}, \mathbf{w} \leftarrow \mathbb{Z}_p^n$.

3 Our Registered KP-ABE

In this section, we propose a predicate encoding for KP-ABE with short ciphertexts. By applying Zhu et al.'s [25] generic construction, we obtain a registered KP-ABE with short ciphertexts.

3.1 Our Predicate Encoding for KP-ABE

We employ predicate encoding based on monotone boolean span programs [3]. A monotone boolean span program denoted by V, is defined by $\mathbf{Y} \in \mathbb{Z}_p^{m\times\ell}$ where

$P(\mathbf{x}, V) = 1 \leftrightarrow V(\mathbf{x}) = 1 \leftrightarrow \mathbf{x} \in \{0,1\}^{1\times m}$ satisfies $\mathbf{Y} \leftrightarrow \exists \boldsymbol{\omega} \in \mathbb{Z}_p^{1\times m}$ such that $\mathbf{e}_1 = \boldsymbol{\omega}\cdot\mathsf{diag}(\mathbf{x})\mathbf{Y}$.

We employ notation $\mathsf{diag}(\mathbf{x}) := \begin{pmatrix} x_1 & & \\ & \ddots & \\ & & x_m \end{pmatrix} \in \mathbb{Z}_p^{m\times m}$ for $\mathbf{x} = (x_1, \ldots, x_m)$ and note that $\mathsf{diag}(\mathbf{x}) = \mathsf{diag}(\mathbf{x})^\intercal$. Let $n = m + \ell + 1$, $n_k = m + 1$, $n_c = m + 1$, and define $\mathbf{a}_\mathbf{Y}, \mathbf{K}_\mathbf{Y}, \mathbf{C}_\mathbf{x}, \mathbf{d}_{\mathbf{x},\mathbf{Y}}$ as

follows.

$$\mathbf{a}_{\mathbf{Y}} = [\underbrace{0 \dots 0}_{m} \ 1], \qquad \mathbf{K}_{\mathbf{Y}} = \begin{bmatrix} \underbrace{0 \dots 0}_{m} & 1 \\ \mathbf{I}_m & \mathbf{0}_m \\ \mathbf{Y}^\intercal & \mathbf{0}_\ell \end{bmatrix},$$

$$\mathbf{C}_{\mathbf{x}} = \begin{bmatrix} 1 & \underbrace{0 \dots 0}_{m} \\ \mathbf{0}_m & \mathsf{diag}(\mathbf{x}) \\ \mathbf{e}_1^\intercal & \mathbf{0}_{\ell \times m} \end{bmatrix}, \quad \mathbf{d}_{\mathbf{x},\mathbf{Y}} = (\boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x}) \parallel 1 \parallel -1 \parallel -\boldsymbol{\omega}).$$

Next, we will prove that our predicate encoding satisfies two properties: α-reconstruction for correctness and α-privacy for security. α-reconstruction ensures that α can be recovered when $P(\mathbf{x}, \mathbf{Y}) = 1$, while α-privacy ensures that α is hidden when $P(\mathbf{x}, \mathbf{Y}) = 0$.

α-reconstruction. For all $\mathbf{x} \in X$ and $\mathbf{Y} \in Y$, compute $\mathbf{M}_{\mathbf{x},\mathbf{Y}}\mathbf{d}_{\mathbf{x},\mathbf{Y}}^\intercal$. If $\mathbf{M}_{\mathbf{x},\mathbf{Y}}\mathbf{d}_{\mathbf{x},\mathbf{Y}}^\intercal = \mathbf{e}_1^\intercal$, then it implies that $P(\mathbf{x}, \mathbf{Y}) = 1$, and thus α can be recovered.

$$\begin{aligned}
\mathbf{M}_{\mathbf{x},\mathbf{Y}}\mathbf{d}_{\mathbf{x},\mathbf{Y}}^\intercal &= \begin{bmatrix} \mathbf{a}_{\mathbf{Y}} & \mathbf{0}_{\mathbf{n_c}} \\ \mathbf{K}_{\mathbf{Y}} & \mathbf{C}_{\mathbf{x}} \end{bmatrix} (\boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x}) \parallel 1 \parallel -1 \parallel -\boldsymbol{\omega})^\intercal \\
&= \begin{bmatrix} \underbrace{0 \dots 0}_{m} & 1 & 0 & \underbrace{0 \dots 0}_{m} \\ \underbrace{0 \dots 0}_{m} & 1 & 1 & \underbrace{0 \dots 0}_{m} \\ \mathbf{I}_m & \mathbf{0} & \mathbf{0} & \mathsf{diag}(\mathbf{x}) \\ \mathbf{Y}^\intercal & \mathbf{0} & \mathbf{e}_1^\intercal & \mathbf{0} \end{bmatrix} \\
&\quad \cdot (\boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x}) \parallel 1 \parallel -1 \parallel -\boldsymbol{\omega})^\intercal \\
&= \begin{bmatrix} 1 \\ 1 - 1 \\ (\boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x}) - \boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x}))^\intercal \\ \mathbf{Y}^\intercal \boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x}) - \mathbf{e}_1^\intercal \end{bmatrix} \\
&= \mathbf{e}_1^\intercal \quad (\because \mathbf{Y}^\intercal \boldsymbol{\omega} \cdot \mathsf{diag}(\mathbf{x}) = \mathbf{e}_1^\intercal)
\end{aligned}$$

α-privacy. Let $\mathbf{w} = (w_0 \ \mathbf{w}' \ \mathbf{w}'')$, where $\mathbf{w}' = (w_1' \dots) \in \mathbb{Z}_p^m, \mathbf{w}'' = (w_1'' \ \dots) \in \mathbb{Z}_p^\ell$. Then we have

$$\begin{aligned}
(\alpha \| \mathbf{w}) \mathbf{M}_{x,y} &= (\alpha \| w_0 \| \mathbf{w}' \| \mathbf{w}'') \begin{bmatrix} \mathbf{0} & 1 & 0 & \mathbf{0} \\ \mathbf{0} & 1 & 1 & \mathbf{0} \\ \mathbf{I}_m & 0 & 0 & \mathsf{diag}(\mathbf{x}) \\ \mathbf{Y}^\intercal & 0 & \mathbf{e}_1^\intercal & \mathbf{0} \end{bmatrix} \\
&= (\mathbf{w}'\mathbf{I}_m + \mathbf{w}''\mathbf{Y}^\intercal \parallel \alpha + w_0 \\
&\quad \parallel w_0 + w_1'' \parallel \mathbf{w}' \mathsf{diag}(\mathbf{x})).
\end{aligned}$$

Let $\mathbf{Y}^\intercal = (\mathbf{y}_1^\intercal \| \dots \| \mathbf{y}_m^\intercal)$. We rewrite

$$\mathbf{w}'\mathbf{I}_m + \mathbf{w}''\mathbf{Y}^\intercal = (w_1' + \mathbf{w}''\mathbf{y}_1^\intercal \| \dots \| w_m' + \mathbf{w}''\mathbf{y}_m^\intercal)$$

and

$$\mathbf{w}' \mathsf{diag}(\mathbf{x}) = (w_1' x_1 \| \dots \| w_m' x_m).$$

Note that $\alpha = (\alpha + w_0) - (w_0 + w_1'') + w_1''$, in order to show the α-privacy, it is suffice to show that w_1'' is independent of the elements of $\mathbf{w}'\mathbf{I}_m + \mathbf{w}''\mathbf{Y}^\intercal$ and $\mathbf{w}' \mathsf{diag}(\mathbf{x})$. Since $P(\mathbf{x}, \mathbf{Y}) = 0$, we know that $\mathbf{e}_1 \notin \mathbf{RowSpan}(\mathbf{Y}_{\mathbf{x}})$, where $\mathbf{Y}_{\mathbf{x}} = \mathsf{diag}(\mathbf{x})\mathbf{Y}$. Assume that there are constants $\{c_i', c_i''\}_{i \in [m]}$ such that $\sum_{i=1}^m c_i'(w_i' + \mathbf{w}''\mathbf{y}_i^\intercal) + \sum_{i=1}^m c_i''(w_i' x_i) = w_1''$. Hence we have $\sum_{i=1}^m c_i' w_i' =$

$-\sum_{i=1}^{m} c_i''(w_i' x_i)$ and $\sum_{i=1}^{m} c_i' \mathbf{y_i} = \mathbf{e}_1$. Let $\mathbf{c}' = (c_1' \| \ldots \| c_m')$, $\mathbf{c}'' = (c_1'' \| \ldots \| c_m'')$. The equations shown above can be rewritten as

$$\begin{aligned}\mathbf{c}'(\mathbf{w}')^\intercal &= -\mathbf{c}''(\mathbf{w}'\mathsf{diag}(\mathbf{x}))^\intercal \\ &= -\mathbf{c}''\mathsf{diag}(\mathbf{x})^\intercal \mathbf{w}'^\intercal \\ &= -\mathbf{c}''\mathsf{diag}(\mathbf{x})\mathbf{w}'^\intercal \ (\because \mathsf{diag}(\mathbf{x}) = \mathsf{diag}(\mathbf{x})^\intercal),\end{aligned}$$

which implies $\mathbf{c}' = -\mathbf{c}''\mathsf{diag}(\mathbf{x})$, and $\mathbf{e}_1 = \sum_{i=1}^{m} c_i' \mathbf{y_i} = \mathbf{c}'\mathbf{Y} = (-\mathbf{c}'')\mathsf{diag}(\mathbf{x})\mathbf{Y}$, which contracts to the fact that $\mathbf{e}_1 \notin \mathbf{RowSpan}(\mathbf{Y_x})$. This prove the α-privacy, since the independence between w_1'' and of the elements of $\mathbf{w}'\mathbf{I}_m + \mathbf{w}''\mathbf{Y}^\intercal$ and $\mathbf{w}'\mathsf{diag}(\mathbf{x})$ implies that α is complete hidden from the adversary's view.

3.2 Our Slotted Registered KP-ABE

In the following subsection, we incorporate the predicate encoding we designed for KP-ABE into the generic registered-ABE scheme proposed by Zhu et al. [25]. This slotted registered KP-ABE scheme is based on the SXDH (1-Lin) assumption and is designed for read-once boolean span programs. The scheme consists of six algorithms: $Setup$, $KeyGen$, Ver, Agg, $Encrypt$, and $Decrypt$.

$Setup(1^\lambda, P, 1^L)$.

1. Generate $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$.
2. Sample $\mathbf{a} \leftarrow \mathbb{Z}_p^{1\times 3}$, $\mathbf{b}^\intercal \leftarrow \mathbb{Z}_p^2$, $\mathbf{k} \leftarrow \mathbb{Z}_p^{1\times 3}$.
3. For all $i \in [L]$, sample $\mathbf{V}_i \leftarrow \mathbb{Z}_p^{3\times 2}$, $\mathbf{W}_i \leftarrow \mathbb{Z}_p^{3\times 2(m+l+1)}$, $\mathbf{R}_i \leftarrow \mathbb{Z}_p^{4\times 3}$, $\mathbf{r}_i \leftarrow \mathbb{Z}_p$.
4. For all $i \in [L]$, let $\mathbf{A}_i = \left(\begin{smallmatrix}\mathbf{a}\\ \mathbf{R}_i\end{smallmatrix}\right) \in \mathbb{Z}_p^{5\times 3}$ and sample
$$\mathbf{a}'_i \leftarrow \mathbb{Z}_p^{1\times 2}, \mathbf{b}'^\intercal_i \leftarrow \mathbb{Z}_p^2, \mathbf{K}'_i \leftarrow \mathbb{Z}_p^{5\times 2}, \mathbf{K}'_{i,0}, \mathbf{K}'_{i,1} \leftarrow \mathbb{Z}_p^{2\times 2}.$$
5. For all $i \in [L]$, compute
$$\begin{aligned}&\mathbf{P}_i = \mathbf{A}_i^\intercal \mathbf{K}'_i, \quad \mathbf{p_{i,0}} = \mathbf{a}'_i \mathbf{K}'_{i,0}, \quad \mathbf{p_{i,1}} = \mathbf{a}'_i \mathbf{K}'_{i,1}, \\ &\mathbf{c}'^\intercal_i = \mathbf{K}'_i \mathbf{b}'^\intercal_i, \quad \mathbf{c}'^\intercal_{i,0} = \mathbf{K}'_{i,0} \mathbf{b}'^\intercal_i, \quad \mathbf{c}'^\intercal_{i,1} = \mathbf{K}'_{i,0} \mathbf{b}'^\intercal_i.\end{aligned}$$
6. For all $i \in [L]$, set $crs_i = ([\mathbf{a}'_i, \mathbf{P}_i, \mathbf{p_{i,0}}, \mathbf{p_{i,1}}]_1, [\mathbf{b}'^\intercal_i, \mathbf{c}'^\intercal_i, \mathbf{c}'^\intercal_{i,0}, \mathbf{c}'^\intercal_{i,1}]_2)$ and $td_i = \mathbf{K}'_i$.
7. Output
$$crs = \begin{pmatrix} [\mathbf{a}]_1, [\mathbf{a}\mathbf{k}^\intercal]_T, \{crs_i, [\mathbf{R}_i, \mathbf{a}\mathbf{V}_i, \mathbf{a}\mathbf{W}_i]_1\}_{i\in[L]}, \\ \{[\mathbf{b}^\intercal \mathbf{r}_j, \mathbf{V}_j \mathbf{b}^\intercal \mathbf{r}_j + \mathbf{k}^\intercal]_2\}_{j\in[L]}, \\ [\mathbf{V}_i \mathbf{b}^\intercal \mathbf{r}_j, \mathbf{W}_i(\mathbf{I}_{m+l+1} \otimes \mathbf{b}^\intercal \mathbf{r}_j)]_2\}_{j\in[L], i\in[L]\setminus\{j\}} \end{pmatrix}.$$

$KeyGen(crs, i)$.

1. Sample $\mathbf{U}_i \leftarrow \mathbb{Z}_p^{3\times 2}$.
2. Let $\mathbf{M}_i = \left(\begin{smallmatrix}\mathbf{t}_i\\ \mathbf{Q}_i\end{smallmatrix}\right) = \left(\begin{smallmatrix}\mathbf{a}\mathbf{U}_i\\ \mathbf{R}_i\mathbf{U}_i\end{smallmatrix}\right) \in \mathbb{Z}_p^{5\times 2}$.
3. Sample $\mathbf{s}_i^\intercal \leftarrow \mathbb{Z}_p^2$.
4. Compute $\pi_i = [\underbrace{\mathbf{U}_i^\intercal \mathbf{P}_i + \mathbf{s}_i^\intercal(\mathbf{p}_{i,0} + \mathbf{p}_{i,1})}_{\pi_{i,0}}, \underbrace{\mathbf{s}_i^\intercal \mathbf{a}'_i}_{\pi_{\mathbf{i,1}}}]_1$.

5. Extract $[\mathbf{R}_i]_1$ and $\{[\mathbf{b}^\intercal \mathbf{r}_j]_2\}_{j\in[L]\setminus\{i\}}$ from crs.
6. Output $pk_i = ([\underbrace{\mathbf{a}\mathbf{U}_i}_{\mathbf{t}_i}, \underbrace{\mathbf{R}_i\mathbf{U}_i}_{\mathbf{Q}_i}]_1, \{[\underbrace{\mathbf{U}_i\mathbf{b}^\intercal\mathbf{r}_j}_{\mathbf{h}_{i,j}^\intercal}]_2\}_{j\in[L]\setminus\{i\}}, \pi_i)$ and $sk_i = \mathbf{U}_i$.

$Ver(crs, i, pk_i)$.

1. Compute $pk_i = ([\mathbf{t}_i, \mathbf{Q}_i]_1, \{[\mathbf{h}_{i,j}^\intercal]_2\}_{j\in[L]\setminus\{i\}}, \pi_i)$.
2. Extract $[\mathbf{b'}_i^\intercal, \mathbf{c'}_i^\intercal, \mathbf{c'}_{i,0}^\intercal, \mathbf{c'}_{i,1}^\intercal]_2$ from crs_i in crs.
3. Compute $\mathbf{M}_i$ and π_i.
4. Check $e([\pi_{i,0}]_1, [\mathbf{b'}_i^\intercal]_2) \stackrel{?}{=} e([\mathbf{M}_i^\intercal]_1, [\mathbf{c'}_i^\intercal]_2) \cdot e([\pi_{i,1}]_1, [\mathbf{c'}_{i,0}^\intercal + \mathbf{c'}_{i,1}^\intercal]_2)$.
5. For each $j \in [L]\setminus\{i\}$, check $e([\mathbf{a}]_1, [\mathbf{h}_{i,j}^\intercal]_2) \stackrel{?}{=} e([\mathbf{t}_i]_1, [\mathbf{b}^\intercal\mathbf{r}_j]_2)$.
6. If all the above checks pass, output 1; otherwise, output 0.

$Agg(crs, (pk_i, \mathbf{Y}_i)_{i\in[L]})$.

1. For all $i \in [L]$, compute $pk_i = ([\mathbf{t}_i, \mathbf{Q}_i]_1, \{[\mathbf{h}_{i,j}^\intercal]_2\}_{j\in[L]\setminus\{i\}}, \pi_i)$.
2. Compute mpk.

$$\begin{aligned} mpk \quad = \Bigg(& [\mathbf{a}]_1, \Bigg[\textstyle\sum_{i\in[L]} ((\mathbf{a}\mathbf{V}_i + \mathbf{t}_i)((\mathbf{0}_{1\times m} \| 1) \otimes \mathbf{I}_2) \\ & + \mathbf{a}\mathbf{W}_i \left(\begin{pmatrix} \mathbf{0}_{1\times m} & 1 \\ \mathbf{I}_m & \mathbf{0}_m \\ \mathbf{Y}^\intercal & \mathbf{0}_\ell \end{pmatrix} \otimes \mathbf{I}_2 \right) \Bigg]_1, \\ & \left[\textstyle\sum_{i\in[L]} \mathbf{a}\mathbf{W}_i \right]_1, [\mathbf{a}\mathbf{k}^\intercal]_T \Bigg). \end{aligned}$$

3. For all $j \in [L]$, compute $hsk_j = [\mathbf{k}_0^\intercal, \mathbf{k}_1^\intercal, \mathbf{K_2}, \mathbf{K_3}]_2$

$$\begin{aligned} \mathbf{k}_0^\intercal &= \mathbf{b}^\intercal\mathbf{r}_j \\ \mathbf{k}_1^\intercal &= \mathbf{V}_j\mathbf{b}^\intercal\mathbf{r}_j + \mathbf{k}^\intercal \\ \mathbf{K_2} &= \textstyle\sum_{i\in[L]\setminus\{j\}} \Bigg((\mathbf{V}_i\mathbf{b}^\intercal\mathbf{r}_j + \mathbf{h}_{i,j}^\intercal)(\mathbf{0}_{1\times m} \| 1) \\ &\quad + \mathbf{W}_i(\mathbf{I}_{m+\ell+1} \otimes \mathbf{b}^\intercal\mathbf{r}_j) \begin{pmatrix} \mathbf{0}_{1\times m} & 1 \\ \mathbf{I}_m & \mathbf{0}_m \\ \mathbf{Y}^\intercal & \mathbf{0}_\ell \end{pmatrix} \Bigg) \\ \mathbf{K_3} &= \textstyle\sum_{i\in[L]\setminus\{j\}} \mathbf{W}_i(\mathbf{I}_{m+\ell+1} \otimes \mathbf{b}^\intercal\mathbf{r}_j) \end{aligned}$$

4. Output $(mpk, (hsk_j)_{j\in[L]})$.

$Encrypt(mpk, \mathbf{x}, m)$.

1. Sample $s \leftarrow \mathbb{Z}_p$.

2. Compute $ct_{\mathbf{x}} = ([\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2]_1, C)$.

$$\begin{aligned}
\mathbf{c}_0 &= s\mathbf{a} \\
\mathbf{c}_1 &= \textstyle\sum_{i\in[L]}(s\mathbf{a}\mathbf{V}_i + s\mathbf{t}_i)((\mathbf{0}_{1\times m}\|1)\otimes\mathbf{I}_2) \\
&\quad + s\mathbf{a}\mathbf{W}_i\left(\begin{pmatrix}\mathbf{0}_{1\times m} & 1\\ \mathbf{I}_m & \mathbf{0}_m\\ \mathbf{Y}^\intercal & \mathbf{0}_\ell\end{pmatrix}\otimes\mathbf{I}_2\right) \\
\mathbf{c}_2 &= \textstyle\sum_{i\in[L]} s\mathbf{a}\mathbf{W}_i\left(\begin{pmatrix}1 & \mathbf{0}_{1\times m}\\ \mathbf{0}_m & \mathsf{diag}(\mathbf{x})\\ \mathbf{e}_1^\intercal & \mathbf{0}_\ell\end{pmatrix}\otimes\mathbf{I}_2\right) \\
C &= [s\mathbf{a}\mathbf{k}^\intercal]_T\cdot m
\end{aligned}$$

3. Output $ct_{\mathbf{x}}$.

$Decrypt(sk_{i^*}, hsk_{i^*}, ct_{\mathbf{x}})$.

1. Compute $[\mathbf{z_1}, \mathbf{z_2}, z_3, z_4]_T$

$$[\mathbf{z}_1]_T = e([\mathbf{c}_1\|\mathbf{c}_2]_1, [\mathbf{I}_{2m+2}\otimes\mathbf{k}_0^\intercal]_2)$$

$$[\mathbf{z}_2]_T = e([\mathbf{c}_0,]_1, [\mathbf{K}_2\|\mathbf{K}_3\mathbf{C}_x]_2)$$

$$[z_3]_T = e([\mathbf{c}_0\mathbf{U}_{i^*}]_1, [\mathbf{k}_0^\intercal]_2)$$

$$[z_4]_T = e([\mathbf{c}_0]_1, [\mathbf{k}_1^\intercal]_2)$$

2. Compute $\mathbf{d}_{\mathbf{x},\mathbf{y}_{i^*}} = (\boldsymbol{\omega}\cdot\mathsf{diag}(\mathbf{x})\|1\| - 1\|-\boldsymbol{\omega})$.
3. Output $m' = [(\mathbf{z}_1 - \mathbf{z}_2)\mathbf{d}^\intercal_{\mathbf{x},\mathbf{y}_{i^*}} - z_3 - z_4]_T\cdot C$

4 Comparison

In this Section, we compare our proposed work with other encryption methods that have incorporated RBE. In TABLE 1 below, it is evident that many works have introduced RBE into IBE and CP-ABE, and other encryption methods. However, research on integrating RBE into KP-ABE is relatively scarce. Our work distinguishes itself by incorporating RBE into KP-ABE.

Reference	Encryption
[10]	IBE
[12]	IBE
[11]	IBE
[14]	CP-ABE
[25]	CP-ABE, IBE, Zero Inner-Product
[9]	CP-ABE, Broadcast Encryption
[8]	Functional Encryption (CP-ABE, IPE)
[5]	Functional Encryption
Ours	KP-ABE

Table 1: Papers Introducing RBE

5 Conclusion

In this work, we follow the generic registered-ABE scheme proposed by Zhu et al. [25] and design a predicate encoding for KP-ABE. Based on their approach, we construct a registered KP-ABE. This registered KP-ABE addresses the key-escrow problem in KP-ABE by enabling users to generate their own keys. However, it still has limitations. Since our registered KP-ABE relies on predicate encoding, if the KP-ABE being converted does not meet our predicate encoding requirements, it cannot be transformed into a registered KP-ABE using our proposed method. Future work could focus on enhancing the flexibility of predicate encoding to accommodate a broader range of KP-ABE schemes, thereby improving the versatility and reliability of registered KP-ABE.

Acknowledgments

This work was partially supported by the National Science and Technology Council of Taiwan, under grants NSTC 112-2221-E-011 -094 -MY2, NSTC 112-2221-E-011 -092 -MY2, NSTC 113-2634-F-011 -002 -MBK, and NSTC 113-2221-E-004-012 -. This research is grateful for the support of "The National Defense Science and Technology Academic Collaborative Research Project in 2025."

References

[1] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

[2] Melissa Chase. Multi-authority attribute based encryption. In *Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007. Proceedings 4*, pages 515–534. Springer, 2007.

[3] Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system abe in prime-order groups via predicate encodings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 595–624. Springer, 2015.

[4] Sangjukta Das and Suyel Namasudra. Multiauthority cp-abe-based access control model for iot-enabled healthcare infrastructure. *IEEE Transactions on Industrial Informatics*, 19(1):821–829, 2022.

[5] Pratish Datta and Tapas Pal. Registration-based functional encryption. *IACR Cryptol. ePrint Arch.*, 2023:457, 2023.

[6] Whitfield Diffie and Martin E Hellman. New directions in cryptography. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pages 365–390. 2022.

[7] TP Ezhilarasi, N Sudheer Kumar, TP Latchoumi, and N Balayesu. A secure data sharing using idss cp-abe in cloud storage. In *Advances in Industrial Automation and Smart Manufacturing: Select Proceedings of ICAIASM 2019*, pages 1073–1085. Springer, 2021.

[8] Danilo Francati, Daniele Friolo, Monosij Maitra, Giulio Malavolta, Ahmadreza Rahimi, and Daniele Venturi. Registered (inner-product) functional encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 98–133. Springer, 2023.

[9] Cody Freitag, Brent Waters, and David J Wu. How to use (plain) witness encryption: Registered abe, flexible broadcast, and more. In *Annual International Cryptology Conference*, pages 498–531. Springer, 2023.

[10] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. Registration-based encryption: removing private-key generator from ibe. In *Theory of Cryptography: 16th International Conference, TCC 2018, Panaji, India, November 11–14, 2018, Proceedings, Part I 16*, pages 689–718. Springer, 2018.

[11] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, Ahmadreza Rahimi, and Sruthi Sekar. Registration-based encryption from standard assumptions. In *IACR international workshop on public key cryptography*, pages 63–93. Springer, 2019.

[12] Noemi Glaeser, Dimitris Kolonelos, Giulio Malavolta, and Ahmadreza Rahimi. Efficient registration-based encryption. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1065–1079, 2023.

[13] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.

[14] Susan Hohenberger, George Lu, Brent Waters, and David J Wu. Registered attribute-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 511–542. Springer, 2023.

[15] Hanshu Hong, Di Chen, and Zhixin Sun. A practical application of cp-abe for mobile phr system: a study on the user accountability. *SpringerPlus*, 5:1–8, 2016.

[16] Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. Distributed attribute-based encryption. In *Information Security and Cryptology–ICISC 2008: 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers 11*, pages 20–36. Springer, 2009.

[17] Anup R Nimje, VT Gaikwad, and HN Datir. Attribute-based encryption techniques in cloud computing security: an overview. *Int. J. Comput. Trends Technol*, 4(3):419–422, 2013.

[18] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[19] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24*, pages 457–473. Springer, 2005.

[20] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology: Proceedings of CRYPTO 84 4*, pages 47–53. Springer, 1985.

[21] Kranthi Kumar Singamaneni, Anil Kumar Budati, and Thulasi Bikku. An efficient q-kpabe framework to enhance cloud-based iot security and privacy. *Wireless Personal Communications*, pages 1–29, 2024.

[22] Hoeteck Wee. Dual system encryption via predicate encodings. In *Theory of Cryptography Conference*, pages 616–637. Springer, 2014.

[23] Yang Zhao, Mao Ren, Songquan Jiang, Guobin Zhu, and Hu Xiong. An efficient and revocable storage cp-abe scheme in the cloud computing. *Computing*, 101:1041–1065, 2019.

[24] Yao Zhen. *Privacy-preserving personal health record system using attribute-based encryption*. PhD thesis, Worcester Polytechnic Institute, 2011.

[25] Ziqi Zhu, Kai Zhang, Junqing Gong, and Haifeng Qian. Registered abe via predicate encodings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 66–97. Springer, 2023.