Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064486

ARTICLE

Several Attacks on Attribute-Based Encryption Schemes

Phi Thuong Le[1], Huy Quoc Le[2] and Viet Cuong Trinh[1,*]

[1]Faculty of Information and Communication Technology, Hong Duc University, Thanh Hoa, 40130, Vietnam
[2]School of Information Technology, Tan Tao University, Tan Tao University Avenue, Duc Hoa, 82700, Vietnam

*Corresponding Author: Viet Cuong Trinh. Email: trinhvietcuong@hdu.edu.vn

Received: 17 February 2025; Accepted: 11 April 2025; Published: 19 May 2025

ABSTRACT: Attribute-based encryption (ABE) is a cryptographic framework that provides flexible access control by allowing encryption based on user attributes. ABE is widely applied in cloud storage, file sharing, e-Health, and digital rights management. ABE schemes rely on hard cryptographic assumptions such as pairings and others (pairing-free) to ensure their security against external and internal attacks. Internal attacks are carried out by authorized users who misuse their access to compromise security with potentially malicious intent. One common internal attack is the attribute collusion attack, in which users with different attribute keys collaborate to decrypt data they could not individually access. This paper focuses on the ciphertext-policy ABE (CP-ABE), a type of ABE where ciphertexts are produced with access policies. Our first work is to carry out the attribute collusion attack against several existing pairing-free CP-ABE schemes. As a main contribution, we introduce a novel attack, termed the *anonymous key-leakage attack*, concerning the context in which users could anonymously publish their secret keys associated with certain attributes on public platforms without the risk of detection. This kind of internal attack has not been defined or investigated in the literature. We then show that several prominent pairing-based CP-ABE schemes are vulnerable to this attack. We believe that this work will contribute to helping the community evaluate suitable CP-ABE schemes for secure deployment in real-life applications.

KEYWORDS: Attribute-based encryption; ciphertext-policy; attribute collusion attack; anonymous key-leakage attack

1 Introduction

Encryption was utilized about two thousand years ago, with early examples including the scytale transposition and Caesar cipher. Encryption methods fall into two categories. The first is symmetric encryption, which uses the same key for encryption and decryption. The second is public-key encryption, where the public key differs from the private key. Both symmetric and public key encryption are one-to-one encryption techniques. However, modern applications often require one-to-many encryption. For example, in encrypted file systems like cloud storage, a file might need to be securely shared with multiple users. In encrypted email systems, an email might need to be sent securely to several recipients. A private message might need to be sent to a group on social networks. Likewise, in pay-TV systems, a broadcaster might need to transmit a channel to a group of subscribers.

To address the challenge of flexible one-to-many encryption, Sahai and Waters [1] introduced the concept of attribute-based encryption (ABE), where user attributes govern encryption and decryption. Based on this, Bethencourt et al. [2] defined a variant of ciphertext-policy attribute-based encryption (CP-ABE) and developed a CP-ABE instance using bilinear pairing. In CP-ABE, encryption assigns an access policy to the ciphertext, and decryption is restricted to users whose attributes match this policy. This method allows for fine-grained access control. CP-ABE is ideal for large-scale data sharing across numerous devices, such as in cloud storage systems, mobile pay-TV, and IoT applications. With CP-ABE, the data owner sets an access policy, and the user's secret key is derived from their attributes, enabling secure data sharing while preserving access control.

Currently, the most advanced technique for constructing CP-ABE schemes is based on bilinear pairing [3]. However, bilinear pairing is a complex operation with high computational costs, leading to long encryption times and high energy consumption. As a result, CP-ABE schemes using bilinear pairing are not suitable for lightweight devices that typically have limited computational resources and energy storage, such as mobile pay-TV, satellite transmission, and IoT applications.

To address this issue, CP-ABE schemes based on elliptic curve cryptography (ECC) were introduced [4–7]. Compared to bilinear pairing, scalar multiplication in ECC is much more efficient, making it suitable for implementation in lightweight devices. The first ECC-based CP-ABE scheme was proposed by Yao et al. [4]. However, it did not support fine-grained access policies, limiting its applicability in complex access control scenarios. Later, Ding et al. [5] introduced an ECC-based CP-ABE scheme that supports fine-grained access policies using linear secret sharing. Subsequently, Wang et al. [6] and Sowjanya and Dasgupta [7] proposed additional ECC-based CP-ABE schemes with support for fine-grained access control.

It is known that ABE (hence CP-ABE schemes)-even those limited to AND-gate policies, not just one supporting fine-grained access policy-is a generalization of identity-based encryption (IBE) (see [8]). This relation implies that the implementation of ABE (particularly, CP-ABE) encompasses the challenges of IBE. We also know that constructing a pairing-free IBE poses significant challenges in possibility, efficiency, complexity, and security, as evidenced in the literature, e.g., references [9,10] (note that theoretically several pairing-free IBE schemes have been proposed [11–13]). Therefore, pairing-free CP-ABE systems, including ECC-based ones, substantially amplifies these difficulties; the reader can also refer to [14] for a detailed discussion on this. This is also the reason, why several significant CP-ABE schemes based on bilinear pairing have also been developed, including those by [15–19], and [20–24] among others. Despite their higher computational requirements, these schemes remain important in the field.

Security in ABE. Attacks in ABE can be divided into external and internal. External attacks come from unauthorized users trying to access the system. Most ABE schemes handle these well through access control structures, assuming legitimate users do not publicly leak information. In contrast, internal attacks involve legitimate but dishonest users (malicious insiders) who may leak their secret information to outsiders. These are more difficult to prevent, as the main deterrent is the ability to trace leaks back to their source.

One of the internal attacks is the *attribute collusion attack*. In the attack, multiple users with distinct attributes, none of whom individually satisfies the access policy, collaborate by combining their attributes to bypass the policy and successfully decrypt a ciphertext. For instance, in a hospital's record system using ABE, different staff members (e.g., doctors and nurses) may have individual permissions. Without resistance to attribute collusion, a group of nurses could pool their attributes to gain access to patient records intended for doctors, violating privacy policies. Therefore, the *collusion resistance* is a required and desired property for any ABE system. In particular, the standard security model for CP-ABE is defined through the game against the attribute collusion attack.

We know that the most severe case of internal attacks is when a single user or a small group of users can generate an anonymous secret key (any polynomial-time algorithm cannot trace that) and publicly release or sell this key. To the best of our knowledge, there has been no ABE work in the literature that discusses, defines, and deals with such attacks. Our paper is the first to define this attack and call it the *anonymous key-leakage* (AKL) attack. We also stress that the attack is defined outside the security model (against the attribute collusion attack) of the ABE scheme. Therefore, an ABE scheme that is insecure against this AKL attack does not mean the scheme is insecure in its security model. However, when choosing CP-ABE schemes to deploy in practice, this new attack is worth considering.

Contribution. This work's contribution includes the following:

- We present the attribute collusion attack on several ECC-based CP-ABE schemes.
- We are the first to define the *anonymous key-leakage* (AKL) attack.
- We theoretically conduct the AKL attack on several important pairing-based CP-ABE schemes in the literature.

Related Work and Comparison. In 2020, Javier Herranz proposed the attribute collusion attack on several ECC-based CP-ABE schemes [14]. He also argued about the difficulty in constructing pairing-free CP-ABE schemes. Our work proposes the attribute collusion attack on several new other ECC-based CP-ABE schemes. Our and his proposed attack are practical but in different ECC-based CP-ABE schemes. Moreover, in this work, we additionally propose a new kind of practical attack (different from the attribute collusion attack) on several important pairing-based CP-ABE schemes (published in several flagship conferences such as ACM CCS, Eurocrypt, PKC, Asia CCS). Although these pairing-based CP-ABE schemes are still secure in their defined security model, we should consider this new attack when choosing them to deploy in practice.

Very recently, Siyal et al. [25] have introduced the so-called *adaptive attribute-based honey encryption* (AABHE) scheme to protect cloud data by generating believable decoy ciphertexts ("honeywords") to deceive attackers. While our study focuses on CP-ABE vulnerabilities, particularly insider attacks such as attribute collusion and anonymous key-leakage attacks, AABHE offers a potential mitigation strategy by making leaked keys less exploitable. However, as honey encryption primarily defends against side-channel and brute-force attacks rather than insider threats, its effectiveness against collusion-based key leakage remains uncertain.

Traitor tracing [26] is a technique that allows the tracing of the traitor who used her secret key to produce a pirate box (decryption box). This technique is based on the property that each user's secret key will have a unique property associated with this user. In contrast, the AKL attack is based on the property that the secret key used to produce the pirate box can be created by many users and has no unique property. Therefore, the traitor tracing technique cannot be applied to defend against the AKL attack.

Organization. The paper is organized as follows: In Section 1, we give an introduction to the research problem. Section 2 presents some related background knowledge. Next, we describe the attribute collusion attack in several ECC-based CP-ABE schemes in Section 3. Section 4 defines the anonymous key-leakage attack and then presents the attack on several pairing-based CP-ABE schemes. Finally, we conclude this work in Section 5.

2 Preliminaries

2.1 Pairings

Given a security parameter λ, let $p = p(\lambda)$ be a prime number. Define $\mathbb{G}, \widetilde{\mathbb{G}}, \mathbb{G}_T$ as cyclic groups of order p with generators g and $\widetilde{g}$ for $\mathbb{G}$ and $\widetilde{\mathbb{G}}$, respectively. The generator g_T of $\mathbb{G}_T$ is computed as $e(g, \widetilde{g})$, where $e : \mathbb{G} \times \widetilde{\mathbb{G}} \to \mathbb{G}_T$ is a non-degenerate bilinear map (also known as a *pairing*). For all $x, y \in \mathbb{Z}_p$, the following properties hold: (i) $e(g^x, \widetilde{g}^y) = e(g, \widetilde{g})^{xy}$; (ii) If $g \neq 1_{\mathbb{G}}$ and $\widetilde{g} \neq 1_{\widetilde{\mathbb{G}}}$, then $e(g, \widetilde{g}) \neq 1_{\mathbb{G}_T}$; (iii) It is efficient to compute $e(g, \widetilde{g})$. The tuple $D = (p, \mathbb{G}, \widetilde{\mathbb{G}}, \mathbb{G}_T, e)$ is referred to as a *bilinear map group system*.

2.2 *Attribute-Based Encryption*

This section outlines the syntax and security model of a CP-ABE scheme.

2.2.1 *Syntax*

A CP-ABE scheme comprises four probabilistic algorithms:

- $(\mathsf{MSK}, \mathsf{param}) \leftarrow \textbf{Setup}(1^\lambda, \mathcal{B})$: On input a security parameter λ and an attribute universe $\mathcal{B}$, the algorithm outputs a master secret key MSK and public parameters param for the system.
- $d_u \leftarrow \textbf{Extract}(u, \mathcal{B}(u), \mathsf{MSK}, \mathsf{param})$: Given a user u, the set of attributes $\mathcal{B}(u)$ associated with u, the public parameters param, and the master secret key MSK, this algorithm produces the user's private key d_u.
- $ct \leftarrow \textbf{Encrypt}(\mathcal{M}, \mathbb{A}, \mathsf{param})$: This algorithm takes a message $\mathcal{M}$, an access policy $\mathbb{A}$ over the attribute universe $\mathcal{B}$, and the public parameters param as inputs. It generates a ciphertext ct.
- $\mathcal{M}/\perp \leftarrow \textbf{Decrypt}(ct, d_u, \mathsf{param})$: This algorithm takes a ciphertext ct, the private key d_u of user u, and the public parameters param. It returns the message $\mathcal{M}$ if the attributes $\mathcal{B}(u)$ of user u meet the criteria set by the access policy $\mathbb{A}$. If not, it outputs $\perp$.

2.2.2 *Security Model*

In ABE and particularly CP-ABE schemes, multiple users with different attributes (each user not meeting the access policy alone) may collaborate to decrypt a ciphertext they otherwise could not access individually. Users can combine their attributes to bypass the intended access control when collusion is possible. This is called the *attribute collusion attack*. We illustrate the attack in Fig. 1.

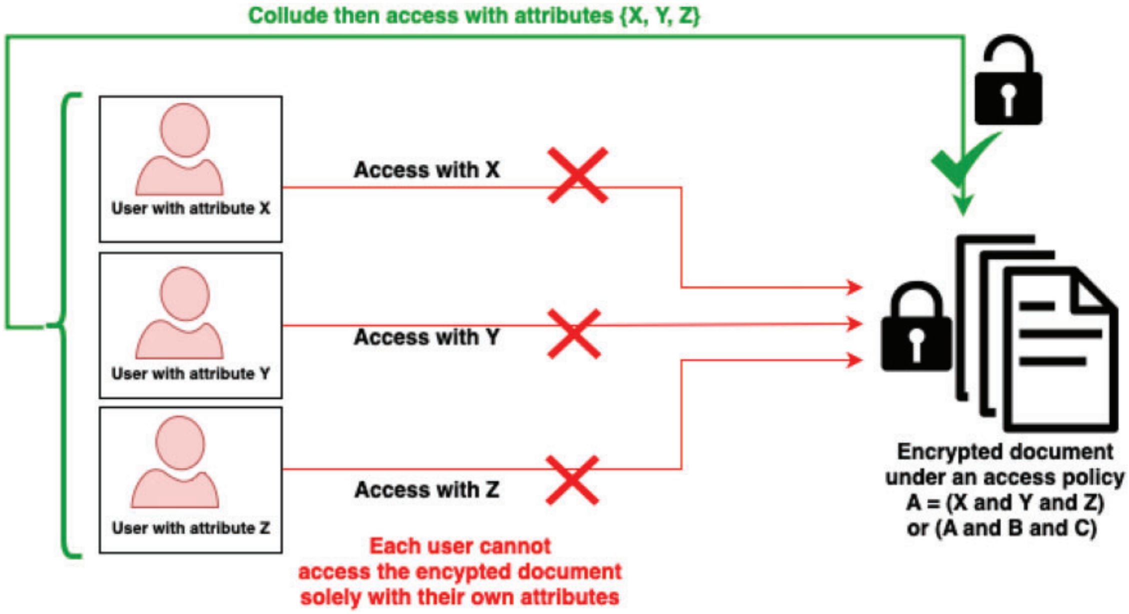
Collude then access with attributes {X, Y, Z}
User with attribute X
User with attribute Y
User with attribute Z
Access with X
Access with Y
Access with Z
Each user cannot access the encypted document solely with their own attributes
Encrypted document under an access policy A = (X and Y and Z) or (A and B and C)

Figure 1: The figure illustrates an attribute collusion attack in ABE, where multiple users combine their attributes to bypass access restrictions. Each user possesses only one attribute (X, Y, or Z), which is insufficient to decrypt the document individually. However, by colluding and pooling their attributes, they satisfy the access policy (X, Y, and Z) and successfully decrypt the document

The security model for a CP-ABE scheme is defined through the following probabilistic game involving an attacker $\mathcal{A}$ and a challenger $\mathcal{C}$, which models an attribute collusion attack:

```
ATTRIBUTE COLLUSION SECURITY GAME:
1. $(\mathsf{param}, \mathsf{MSK}) \leftarrow \textbf{Setup}(1^{\lambda}, \mathcal{B})$,        /* Challenger keeps MSK secret.*/
$\Lambda_C := \emptyset$.                                                                      /* Corruption list.*/
2. $(\Lambda_C, \{(\mathcal{B}(u), d_u)\}_u) \leftarrow \mathcal{A}^{\textbf{Extract}(\cdot,\cdot,\mathsf{MSK},\mathsf{param})}$. /* $d_u \leftarrow \textbf{Extract}(u, \mathcal{B}(u), \mathsf{MSK}, \mathsf{param})$, */
                                                                                               /* $\Lambda_C := \Lambda_C \cup \{u\}_u$. */
3. $(\mathbb{A}^*, \mathcal{M}_0^*, \mathcal{M}_1^*) \leftarrow \mathcal{A}$,                   /* where $|\mathcal{M}_0^*| = |\mathcal{M}_1^*|$.
4. $b \leftarrow_{\$} \{0,1\}$, $ct^* \leftarrow \textbf{Encrypt}(\mathcal{M}_b^*, \mathbb{A}^*, \mathsf{param})$.
5. $(\Lambda_C, \{(\mathcal{B}(u), d_u)\}_u) \leftarrow \mathcal{A}^{\textbf{Extract}(\cdot,\cdot,\mathsf{MSK},\mathsf{param})}$.
6. $b' \leftarrow \mathcal{A}$.
```

The adversary wins the game if $b' = b$ and if the attribute set $\bigcup_{u \in \Lambda_C} \mathcal{B}(u)$ does not satisfy the target policy $\mathbb{A}^*$. The adversary's advantage is defined as $\mathbf{Adv}_{\mathcal{A}} := \Pr[\mathcal{A} \text{ wins the game}]$.

Definition 1: A CP-ABE *scheme is considered secure against the attribute collusion attack if the advantage of any polynomial-time adversaries in the above game is negligible.*

3 Attribute Collusion Attack on Several ECC-Based CP-ABE Schemes

In this section, we show that several important existing pairing-free CP-ABE schemes [5–7] are susceptible to the attribute collusion attack, or these schemes are insecure in their defined security model. The reason why this attack exists in such schemes comes from the insecure structures of their schemes. To analyze deeper, we know that CP-ABE scheme is a generalization of IBE, and as shown in [9,10], constructing a pairing-free IBE poses significant challenges in possibility, efficiency, complexity, and security. Therefore, constructing pairing-free CP-ABE systems, including ECC-based ones, substantially amplifies these difficulties. Revising the insecure structures of their schemes is a challenging task. Note that to date, there are only a few inefficient pairing-free IBE schemes have been proposed [11–13].

3.1 Attribute Collusion Attack on Ding et al.'s Scheme

In Ding et al.'s scheme [5], each user is associated with an identity GID and receives a secret key $SK_{i,GID}$ corresponding to each possessed attribute i as $SK_{i,GID} = k_i + H(GID) \cdot n$, where k_i is a random element chosen for each attribute i, H is a secure hash function and n is the master key MSK.

For simplicity, let us consider that Bob with identity GID_{Bob} possesses attributes i and j. Alice with identity GID_{Alice} possesses attribute i. We will show that when Bob and Alice collude, they can *produce* secret key of attribute j for Alice. To this end, they first compute:

$$\begin{aligned} X &= SK_{i,GID_{Bob}} - SK_{i,GID_{Alice}} \\ &= k_i + H(GID_{Bob}) \cdot n - k_i - H(GID_{Alice}) \cdot n \\ &= H(GID_{Bob}) \cdot n - H(GID_{Alice}) \cdot n \end{aligned}$$

Next, compute the secret key of attribute j for Alice as:

$$\begin{aligned} Y &= SK_{j,GID_{Bob}} - X \\ &= k_j + H(GID_{Bob})n - (H(GID_{Bob})n - H(GID_{Alice})n) \\ &= k_j + H(GID_{Alice}) \cdot n \end{aligned}$$

Correctness. In Ding et al.'s scheme, each user associated with an identity *GID* and possessing attribute i will receive a secret key $SK_{i,GID} = k_i + H(GID) \cdot n$. Hence, it is obvious to verify that $Y = k_j + H(GID_{Alice}) \cdot n$ is the Alice's secret key corresponding to attribute j. Consider a general case, Bob possesses attributes $i, j_1, j_2, \ldots, j_k$, Alice possesses attribute i and u. A secured CP-ABE scheme requires that even if Alice and Bob collude, they cannot decrypt the access policy required possessing all attributes $i, j_1, j_2, \ldots, j_k, u$. However, as shown above, Bob can *help* Alice to *produce* secret keys corresponding to attributes $j_1, j_2, \ldots, j_k$. Alice thus can possess all attributes $i, j_1, j_2, \ldots, j_k, u$. Hence, Alice can easily decrypt the access policy requiring possessing all attributes $i, j_1, j_2, \ldots, j_k, u$, so Ding et al.'s scheme is susceptible to attribute collusion attack. We also emphasize that Bob needs to know only Alice's secret key corresponding to attribute i to help Alice to *produce* secret keys corresponding to attributes $j_1, j_2, \ldots, j_k$.

Performance. To compute the secret key for one attribute, it is easy to see that Bob only needs to compute two subtractions, one to compute X and the other to compute Y. Hence, the complexity of cryptanalysis depends on each specific case of attack. In general, to generate secret keys for t attributes, the complexity is just $2t$ subtractions, which is very practical.

3.2 Attribute Collusion Attack on Wang et al.'s Scheme

In Wang et al.'s scheme [6], the master key MSK is the set $(\alpha, \beta, \gamma, a_{att_1}, \ldots, a_{att_{|\mathcal{U}|}})$, where a_{att_i} is the secret element corresponding to attribute att_i, and $\mathcal{U}$ is the set of attributes in the system. For each user U_j possessing the attribute set $UA_j = \{att_1^j, \ldots, att_m^j\} \subseteq \mathcal{U}$, the authority chooses $u_j \overset{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes the secret key $sk_j = (sk_j^1, sk_j^2)$ for user U_j as

$$sk_j^1 = \alpha - u_j, \qquad sk_j^2 = \{(r^1_{att_i^j}, r^2_{att_i^j}) | i = 1, \ldots, m\}$$

where $(r^1_{att_i^j}, r^2_{att_i^j}), i = 1, \ldots, m$ are randomly chosen such that $a_{att_i^j} + u_j = \gamma r^1_{att_i^j} + \beta r^2_{att_i^j}$.

For simplicity, let us consider the case that Bob possesses the attribute set $UA_{Bob} = \{att_1, att_2\}$. Alice possesses the attribute set $UA_{Alice} = \{att_1\}$. We will show that when Bob and Alice collude, they can *produce* secret key of attribute att_2 for Alice. To this end, they first compute:

$$sk^1_{Bob} - sk^1_{Alice} = \alpha - u_{Bob} - (\alpha - u_{Alice}) = u_{Alice} - u_{Bob}.$$

Consider the attribute att_1, the corresponding sub secret key for Bob is $(r^{1,Bob}_{att_1}, r^{2,Bob}_{att_1})$, for Alice is $(r^{1,Alice}_{att_1}, r^{2,Alice}_{att_1})$, where

$$a_{att_1} + u_{Bob} = \gamma r^{1,Bob}_{att_1} + \beta r^{2,Bob}_{att_1},$$

and $a_{att_1} + u_{Alice} = \gamma r^{1,Alice}_{att_1} + \beta r^{2,Alice}_{att_1}$. So, it implies that

$$\begin{aligned} a_{att_1} + u_{Alice} - a_{att_1} - u_{bob} &= u_{Alice} - u_{Bob} \\ &= \gamma(r^{1,Alice}_{att_1} - r^{1,Bob}_{att_1}) + \beta(r^{2,Alice}_{att_1} - r^{2,Bob}_{att_1}). \end{aligned}$$

Consider the attribute att_2, the corresponding sub secret key for Bob is $(r^{1,Bob}_{att_2}, r^{2,Bob}_{att_2})$, where

$$a_{att_2} + u_{Bob} = \gamma r^{1,Bob}_{att_2} + \beta r^{2,Bob}_{att_2}.$$

So, they have:

$$\begin{aligned}
&a_{att_2} + u_{Bob} + u_{Alice} - u_{Bob} = a_{att_2} + u_{Alice}\\
&= \gamma r^{1,Bob}_{att_2} + \beta r^{2,Bob}_{att_2} + \gamma(r^{1,Alice}_{att_1} - r^{1,Bob}_{att_1}) + \beta(r^{2,Alice}_{att_1} - r^{2,Bob}_{att_1})\\
&= \gamma(r^{1,Bob}_{att_2} + r^{1,Alice}_{att_1} - r^{1,Bob}_{att_1}) + \beta(r^{2,Bob}_{att_2} + r^{2,Alice}_{att_1} - r^{2,Bob}_{att_1}).
\end{aligned}$$

Correctness. Let $r^{1,Alice}_{att_2} = r^{1,Bob}_{att_2} + r^{1,Alice}_{att_1} - r^{1,Bob}_{att_1}$, $r^{2,Alice}_{att_2} = r^{2,Bob}_{att_2} + r^{2,Alice}_{att_1} - r^{2,Bob}_{att_1}$. We can easily check that $(r^{1,Alice}_{att_2}, r^{2,Alice}_{att_2})$ is Alice's secret key corresponding to attribute a_{att_2} since

$$\begin{aligned}
a_{att_2} + u_{Alice} &= \gamma(r^{1,Bob}_{att_2} + r^{1,Alice}_{att_1} - r^{1,Bob}_{att_1}) + \beta(r^{2,Bob}_{att_2} + r^{2,Alice}_{att_1} - r^{2,Bob}_{att_1})\\
&= \gamma r^{1,Alice}_{att_2} + \beta r^{2,Alice}_{att_2}.
\end{aligned}$$

as requirement of Alice's secret key corresponding to attribute att_2. In addition, to compute $(r^{1,Alice}_{att_2}, r^{2,Alice}_{att_2})$, it is easy to see that Bob only needs to know Alice's secret key corresponding to attribute att_1. Hence, as Ding et al.'s scheme [5], Wang et al.'s scheme is also susceptible to attribute collusion attack.

Performance. It is easy to see that to compute $(r^{1,Alice}_{att_2}, r^{2,Alice}_{att_2})$ (the secret key for one attribute) Bob needs to compute four additive operations and two subtractions. Hence, as Ding et al.'s scheme, the complexity of cryptanalysis depends on each specific case of attack. In general, to generate secret keys for t attributes, the complexity is just $6t$ additions/subtractions, which is very practical.

3.3 *Attribute Collusion Attack on Sowjanya et al.'s Scheme*

In Sowjanya et al.'s scheme [7], the master key MSK is the set $(\alpha, \alpha_1, \ldots, \alpha_n)$, where α_i is the secret element corresponding to attribute A_i. Each user is associated with an identity U_{ID}. If she possesses the attribute A_i, she receives a secret key D_i corresponding to attribute A_i as

$$D_i = H(U_{ID}) \cdot \alpha \cdot \alpha_i^{-1}.$$

Note that in the decryption algorithm, the data receiver needs to know $H(U_{ID})$ to decrypt. Hence, each user in the system knows her identity.

For simplicity, let us consider the case that Bob possesses the attribute set $\{A_1, A_2\}$ and Alice possesses the attribute set $\{A_1\}$, we will show that when Bob and Alice collude, they can *produce* secret key of attribute A_2 for Alice. To this end, they compute:

$$D_2^{Alice} = \frac{D_2^{Bob}}{H(U_{ID}^{Bob})} \cdot H(U_{ID}^{Alice}) = H(U_{ID}^{Alice}) \cdot \alpha \cdot \alpha_2^{-1}.$$

Note that they know U_{ID}^{Bob} and U_{ID}^{Alice}.

Correctness. We can easily verify that $D_2^{Alice} = H(U_{ID}^{Alice}) \cdot \alpha \cdot \alpha_2^{-1}$ is Alice's secret key corresponding to attribute A_2. Bob needs to know Alice's identity to compute D_2^{Alice}. Hence, Sowjanya et al.'s scheme is also susceptible to attribute collusion attacks.

Performance. To compute D_2^{Alice} (the secret key for one attribute), Bob needs to compute one multiplicative operation and one inverse operation. Hence, the complexity of cryptanalysis depends on each specific case of attack. In general, the complexity of generating secret keys for t attributes is just $2t$ multiplicative/inverse operations. Thus, it is very practical.

4 Anonymous Key-Leakage Attack on Pairing-Based CP-ABE Schemes

This section defines a novel attack called the *anonymous key-leakage* (AKL in short) attack. We then present the attack on several CP-ABE schemes, including the schemes by [15], the Riepel and Wee's CP-ABE scheme at CCS'22 [22] and some others.

4.1 Definition of the Anonymous Key-Leakage Attack

When considering encryption for a set of users, one can classify the attacks into external and internal. The external attacks come from non-legitimate users trying to access the system. Most CP-ABE schemes deal well with this kind of attack via the structure of access control and with the assumption (defined by the security model) that legitimate users leak nothing into the public database. Internal attacks are about the insiders in the system who are legitimate but dishonest users (called malicious users) who want to leak their secret information to an external pirate. It is often quite complicated to deal with internal attacks because we can only discourage legitimate users from leaking information by the possibility of tracing the source of the leakage. The most disastrous scenario is when a sole legitimate user (see Fig. 2) or multiple collusion groups of legitimate users (see Fig. 3) can produce an anonymous secret key (i.e., any polynomial time trace algorithm cannot determine who exactly generates this key) and leak this key into the public (even sell it to the black market). We name this the *anonymous key-leakage* attack.

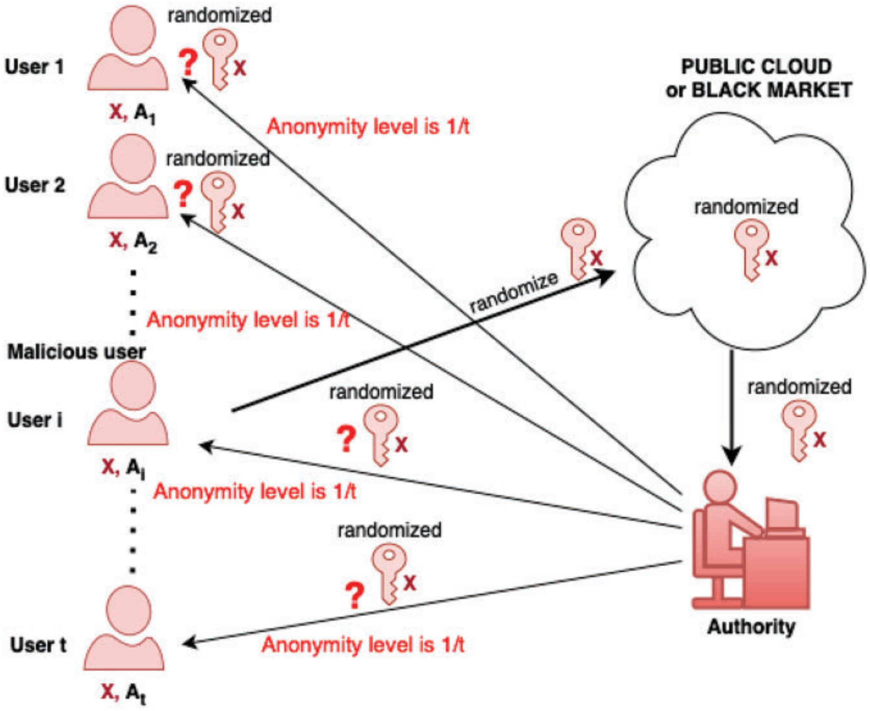

Figure 2: This figure illustrates the scenario of AKL attack with a single malicious user, where multiple users (t users) possess the same attribute set X (though each user may have additional attributes). Here, one (malicious) user could randomize a part of the secret key that corresponds to X and then publish the randomized secret key on a public cloud or a black market while retaining anonymity at a level of $1/t$, as any of the t users could perform this action

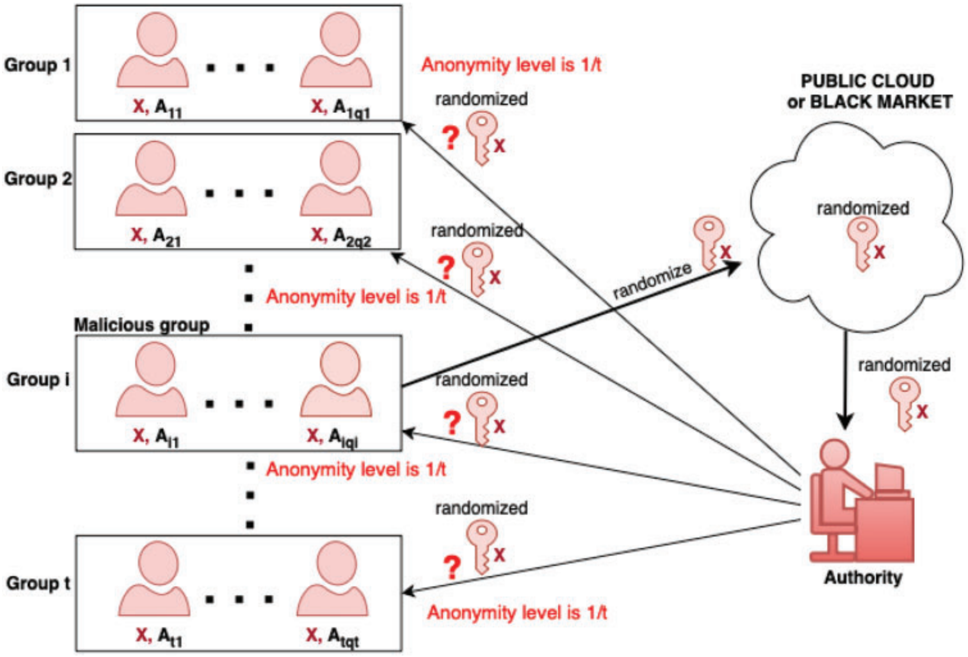

Figure 3: This figure depicts a case of AKL attack with a group of malicious users, where multiple groups (say, t) of users have a common set of attributes X. Here, one (malicious) group of users could randomize a part of the secret key that corresponds to X and then publish the randomized secret key on a public cloud or a black market while retaining anonymity at a level of $1/t$ as any of the t groups could perform this action

Definition 2: *An AKL attack is formally defined as follows.*

- *In the beginning, a single malicious user (or a group of malicious users) receives the secret keys from the authority (who runs the key generation).*
- *The malicious user (or a group of malicious users that collude to produce) produces a new randomized secret key d with the property that this key can also be produced by t different users (or by t different groups of users).*
- *The malicious user (or a group of malicious users) publishes d in the public domain and remains anonymous at level t, in the sense that any polynomial time trace algorithm cannot exactly determine who generates d with a larger correctness probability $1/t$.*

If such an AKL attack exists on a scheme, then this scheme is said to be susceptible to the AKL attack.

The difference between the AKL attack model and the traditional security model is emphasized below.

- In the traditional security model, the attacker has all the secret keys of the malicious users and tries to decrypt the ciphertext that each of these secret keys is not capable of decrypting (equivalent to trying to create a new secret key that is *stronger* than all malicious secret keys).
- In the AKL attack model, the attacker still has all secret keys of the malicious users. However, on the contrary, she tries to create a new secret key that is a part of each of these secret keys (equivalent to trying to create a new secret key that is *weaker* than each malicious secret key), so that no one can know who created this new secret key. The AKL attack thus occurs due to the key structure of CP-ABE schemes.

In the rest of this section, we describe the AKL attack on several important CP-ABE schemes. However, we emphasize that these CP-ABE schemes are still secure in their defined security model. We argue here that even though the AKL attack is defined outside of their security model, it is worth considering this new kind of attack when choosing CP-ABE schemes to deploy in practice.

4.2 AKL Attack From a Single Malicious User

4.2.1 AKL Attack on Waters' Scheme

We first recall the Waters' scheme [15] at PKC'11 as follows.

- **Setup**$(\lambda, \mathcal{B})$: Let λ be a security parameter, $N = |\mathcal{B}|$ be the maximal number of attributes in the system, and let $(p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ be a bilinear group system. First, pick a random generator $g \in \mathbb{G}$, random scalars $a, \alpha \in \mathbb{Z}_p$, and then calculates g^a, g^α. Next, generate$N$ group elements in $\mathbb{G}$ associated with N attributes in the system $h_1, \ldots, h_N$. The master secret key is MSK $= g^\alpha$. The public parameters are $\mathsf{param} = (g, g^a, e(g,g)^\alpha, h_1, \ldots, h_N)$.
- **Extract**$(i, \mathcal{B}_i, \mathsf{MSK}, \mathsf{param})$: The set of attributes of user i is $\mathcal{B}_i$. Pick randomly $r \in \mathbb{Z}_p$, and compute the secret key for user i as $d_i = (d_0, d_0', \{d_i\}_{i \in \mathcal{B}_i})$, where: $d_0 = g^\alpha \cdot g^{a \cdot r}, d_0' = g^r, \{d_i = h_i^r\}_{i \in \mathcal{B}_i}$.
- **Encrypt**$(\mathcal{M}, (M, \rho) \in (\mathbb{Z}_p^{\ell \times \ell'}, \mathcal{F}([\ell] \to [N])), \mathsf{param})$: First pick $k \overset{\$}{\leftarrow} \mathbb{Z}_p$, and compute: $C_0 = \mathcal{M} \cdot e(g,g)^{k\alpha}, C_0' = g^k$, then choose a random vector $\overrightarrow{v} = (k, y_2, \ldots, y_{\ell'}) \in \mathbb{Z}_p^{\ell'}$. For $i = 1, \ldots, \ell$, compute $\lambda_i = \overrightarrow{v} \cdot M_i$, where M_i is the i-th row of M. Computes:

  $$C_i = g^{a.\lambda_i} h_{\rho(i)}^{-k}, i = 1, \ldots, \ell.$$

  Finally, the ciphertext $CP = (C_0, C_0', \ldots, C_\ell)$ is output.
- **Decrypt**$(CP, d_i, \mathsf{param})$: First parse the ciphertext CP as $(C_0, C_0', \ldots, C_\ell)$, then find the set of attributes $\mathcal{A} \subset \mathcal{B}_i$ that satisfies the access policy (M, ρ). Let $I = \{i : \rho(i) \in \mathcal{A}\}$, find the constants $\{\beta_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \beta_i \cdot M_i = (1, 0, \ldots, 0)$, where M_i is the i-th row of the matrix M. Next, compute

  $$e(\prod_{i \in I} C_i^{-\omega_i}, d_0') \cdot e(C_0', d_0 \prod_{i \in I} d_{\rho(i)}^{-\omega_i}) = e(g,g)^{\alpha \cdot k}.$$

  Finally, return $\mathcal{M} = \frac{C_0}{e(g,g)^{k\alpha}}$.

The AKL *attack*. For simplicity, we assume that several users possess the common set of attributes $A_{i_1}, \ldots, A_{i_k}$. We will show that a malicious user u can publish a part of her secret key related to this set of attributes and remain anonymous. To this aim, she randomly chooses r' and, based on her secret key, computes a new sub-key d corresponding to new randomness $r + r'$:

$$d_0 = g^\alpha \cdot g^{a \cdot (r + r')}, d_0' = g^{r + r'}, \{d_i = h_i^{r + r'}\}_{i = i_1, \ldots, i_k}$$

and then publishes d to the public domain.

Correctness. It is straightforward to see that $d = (d_0, d_0', \{d_i\}_{i = i_1, \ldots, i_k})$ is a secret key corresponding to the attribute set $A_{i_1}, \ldots, A_{i_k}$. In addition, it is obvious that any user can produce d (assume t such users) who possesses at least the attribute set $A_{i_1}, \ldots, A_{i_k}$. Hence, any polynomial time trace algorithm cannot determine who generates d with a correctness probability larger than $1/t$.

Performance. To compute d, user needs to compute $k + 2$ exponential operations and $k + 2$ multiplicative operations. Hence, the complexity of cryptanalysis depends on each specific case of AKL attack. In the general case, the complexity of the AKL attack is $T + 2$ exponential operations and $T + 2$ multiplicative operations, where T is the number of attributes in the common set of attributes, and therefore it is very practical.

4.2.2 AKL Attack on Riepel and Wee's Scheme

We first recall Riepel and Wee's CP-ABE scheme at CCS'22 [22] as follows.

- **Setup**$(\lambda, \mathcal{B})$: Let λ be a security parameter, $N = |\mathcal{B}|$ and $(p, \mathbb{G}, \widetilde{\mathbb{G}}, \mathbb{G}_T, g, \tilde{g}, e)$ be a bilinear group system, and $\mathcal{H} : [N+1] \to \mathbb{G}$ be a hash function. Pick $\alpha \in \mathbb{Z}_p$ randomly.

  The master secret key is MSK = α. The public parameters are

  $$\mathsf{param} = (p, \mathbb{G}, \widetilde{\mathbb{G}}, \mathbb{G}_T, g, \tilde{g}, e, e(g, \tilde{g})^{\alpha}, \mathcal{H}).$$

- **Extract**$(i, \mathcal{B}_i, \mathsf{MSK}, \mathsf{param})$: The set of attributes of user i is $\mathcal{B}_i$. The algorithm picks randomly a scalar $r \in \mathbb{Z}_p$, and computes the secret key for user i as

  $$d_i = (d_0, d_0', \{d_u\}_{u \in \mathcal{B}_i})$$

  where $d_0 = g^{\alpha} \cdot \mathcal{H}(N+1)^r, d_0' = \tilde{g}^r, \{d_u = \mathcal{H}(u)^r\}_{u \in \mathcal{B}_i}$.
- **Encrypt**$((M, \pi) \in (\mathbb{Z}_p^{\ell \times \ell'}, \mathcal{F}([\ell] \to \mathcal{B})), \mathsf{param})$: Let $\rho(i) = |\{z | \pi(z) = \pi(i), z \leq i\}|$, and $\tau = max_{i \in [\ell]} \rho(i)$ corresponding to maximum number of times an attribute is used in M.

  Randomly pick $k \xleftarrow{\$} \mathbb{Z}_p, \overrightarrow{v} = (k, y_2, \ldots, y_{\ell'}) \xleftarrow{\$} \mathbb{Z}_p^{\ell'}, \overrightarrow{w} = (w_1, w_2, \ldots, w_\tau) \xleftarrow{\$} \mathbb{Z}_p^{\tau}$, then computes:

  $$ct_1 = \tilde{g}^k, \qquad ct_{2,j} = \tilde{g}^{w_j}$$

  for $j \in [\tau]$. For $i \in [\ell]$, compute $\lambda_i = \overrightarrow{v} \cdot M_i$, where M_i is the i-th row of$M$. Computes:

  $$ct_{3,i} = \mathcal{H}(N+1)^{\lambda_i} \cdot \mathcal{H}(\pi(i))^{w_{\rho(i)}}, i = 1, \ldots, \ell$$

  Finally, output the ciphertext $CP = (ct_1, (ct_{2,j})_{j \in [\tau]}, (ct_{3,i})_{i \in [\ell]})$ along with a description of (M, π), and session key $K = e(g, \tilde{g})^{k \cdot \alpha}$.
- **Decrypt**$(CP, d_i, \mathsf{param})$: First parse the ciphertext CP as $(ct_1, (ct_{2,j})_{j \in [\tau]}, (ct_{3,i})_{i \in [\ell]})$, then find the attribute set $\mathcal{A} \subseteq \mathcal{B}_i$ which satisfies the access policy (M, π). Let $I = \{i : \pi(i) \in \mathcal{A}\}$, find $\{\beta_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \beta_i \cdot M_i = (1, 0, \ldots, 0)$, where M_i is the i-th row of$M$. Next, compute the session key: $K = e(d_0, ct_1) \cdot \frac{\prod_{j \in [\tau]} e(\prod_{i \in I, \rho(i) = j} (d_{\pi(i)})^{\beta_i}, ct_{2,j})}{e(\prod_{i \in I} (ct_{3,i})^{\beta_i}, d_0')}$.

The AKL *Attack*. For simplicity, we assume that there are several users $\{A_1, \ldots, A_t\}$ who possess the common set of attributes $P\mathcal{B}_i' = \{u_1, \ldots, u_k\}$. To build the AKL attack, a malicious user $\mathcal{A}_i, 1 \leq i \leq t$, asks secret key corresponding to this set of attributes to get

$$d_0 = g^{\alpha} \cdot \mathcal{H}(N+1)^r, d_0' = \tilde{g}^r, \{d_u = \mathcal{H}(u)^r\}_{u \in \mathcal{B}_i'}$$

and then randomly chooses r' to compute a new secret key d corresponding to new randomness $r + r'$:

$$d = (d_0 = g^{\alpha} \cdot \mathcal{H}(N+1)^{r+r'}, d_0' = \tilde{g}^{r+r'}, \{d_u = \mathcal{H}(u)^{r+r'}\}_{u \in \mathcal{B}_i'}).$$

Finally, she publishes d to the public domain.

Correctness. It is easy to verify that $d = (d_0, d_0', \{d_u\}_{u \in \mathcal{B}_i'})$ is a secret key corresponding to attribute set $\mathcal{B}_i'$. In addition, d can be produced by any user (assume t such users) who possesses at least the attribute set $\mathcal{B}_i'$. Hence, any polynomial time trace algorithm cannot determine who generates d with a correctness probability larger than $1/t$.

Performance. To compute d, user needs to compute $k+2$ exponential operations and $k+2$ multiplicative operations (k is the cardinality of the attribute set $\mathcal{B}_i'$). Hence, the complexity of cryptanalysis depends on each specific case of AKL attacks. In the general case, the complexity of the AKL attack is $T + 2$ exponential operations and $T + 2$ multiplicative operations, where T is the number of attributes in the common set of attributes; it thus is very practical.

To conclude, we name several other CP-ABE schemes that are also susceptible to the AKL attack from a single malicious user such as [23,24,27,28].

4.3 AKL Attack from a Group of Malicious Users

This section describes the AKL attack from a group of malicious users on several important CP-ABE schemes. Regarding CP-ABE scheme, there is an elegant generic framework to construct adaptively secure CP-ABE schemes introduced in [18–21,29]. This framework also leads to CP-ABE schemes with short ciphertext size and efficient decryption time. We will show that all these schemes are susceptible to the AKL attack from a group of malicious users.

We consider the AKL attack on Chen et al.'s CP-ABE scheme [29] (at Appendix B.2, similarity for other schemes [18–21]), for a given parameter k (see discussion in [29] and below), the secret key of user u, who possesses the attributes $(1, \dots, m)$, is of the following form:

$$K_0 = g^{\mathbf{B.r_u}}, K_1 = g^{\mathbf{A_1.r_u}}, \dots, K_m = g^{\mathbf{A_m.r_u}}, K_{\ell+1} = g^{\mathbf{k+C.r_u}},$$

where $\mathbf{A_1}, \dots, \mathbf{A_m}$, $\mathbf{B}$, $\mathbf{C} \in \mathbb{Z}_p^{k+1\times k}$ are matrices, $\mathbf{k} \in \mathbb{Z}_p^{k\times 1}$ is a vector, ℓ is the number of attributes in the system ($\ell \geq m$). Note that all $\mathbf{A_1}, \dots, \mathbf{A_m}, \mathbf{A_{m+1}}, \dots, \mathbf{A_\ell}, \mathbf{B}, \mathbf{C}, \mathbf{k}$ are in the master key (more precisely, in their scheme $\mathbf{C} = \mathbf{VB}$, and each matrix $\mathbf{A_i} = \mathbf{W_{iB}}$, $\boldsymbol{i} = \mathbf{1}, \dots, \ell$), $\mathbf{r_u} \in \mathbb{Z}_p^{k\times 1}$ is a random vector chosen for user u. We use the same group vector notation as in [29], which means that the first row of the matrix $\mathbf{A_i}$ is $\mathbf{A_{i,1}} = (a_{i,1,1}, \dots, a_{i,1,k})$, $\mathbf{r_u} = (r_{u,1}, \dots, r_{u,k})$ and the group element $K_{1,1} = g^{a_{i,1,1}.r_{u,1}+\dots+a_{i,1,k}.r_{u,k}}$. Note also that $K_1 = (K_{1,1}, \dots, K_{1,k+1})$.

Observer that if a single user wants to produce an AKL attack as in the case of Riepel and Wee's CP-ABE scheme above, she needs to randomize the vector $\mathbf{r_u}$ by first choosing a random element $t \in \mathbb{Z}_p$ and then trying to derive a new decryption key K' associated to a new random vector $\mathbf{r'_u} = t.\mathbf{r_u}$. However, there are two problems:

1. First, she cannot randomize $K_{\ell+1} = g^{\mathbf{k+C.r_u}}$. Note that $\mathbf{k}$ is in the master key.
2. Second, assume that she chooses a random element $t \in \mathbb{Z}_p$ and randomizes $K_i^t = g^{\mathbf{d_i}}$ for some $\mathbf{d_i}$, it becomes feasible for the Tracing algorithm to trace back K_i. Since, given $g^{\mathbf{d_i}}$, there is only one K_i satisfying $K_i^t = g^{\mathbf{d_i}}$ for some $t \in \mathbb{Z}_p$.

The above problems lead to the fact that the AKL attack from a single malicious user cannot straightforwardly apply in these schemes. However, we show that the AKL attack from a group of malicious users can be applied in these schemes.

More precisely, assume that q users $u_1, \dots, u_q$ who possess the common set of attributes $(1, \dots, m)$ secretly collude. As far as $q \geq 2$, an AKL attack from a group of malicious users can be realized. Indeed, each user u_i, $i = 1, \dots, q$, chooses a random element $t_i \in \mathbb{Z}_p$ such that $\sum_{i\in[q]} t_i = 1$. They then compute the derived secret key $K' = (K'_0, K'_1, \dots, K'_m, K'_{\ell+1}) = (g^{\mathbf{d_0}}, g^{\mathbf{d_1}}, \dots, g^{\mathbf{d_m}}, g^{\mathbf{d}_{\ell+1}})$ as follows:

$$K'_0 = \prod_{i\in[q]} K_{u_i,0}^{t_i}, \quad K'_1 = \prod_{i\in[q]} K_{u_i,1}^{t_i}, \quad \dots, \quad K'_m = \prod_{i\in[q]} K_{u_i,m}^{t_i}, \quad K'_{\ell+1} = \prod_{i\in[q]} K_{u_i,\ell+1}^{t_i}.$$

Finally, they publish K' to the public domain.

Correctness. It is easy to verify that the derived secret key K' is in right form with a random vector $\mathbf{r}' = \sum_{i\in[q]} t_i \cdot \mathbf{r}_{\mathbf{u_i}}$ where $\mathbf{r}_{\mathbf{u_i}}$ is a random vector associated to secret key of user u_i. We see that $K'_{\ell+1} = g^{\mathbf{k}+\mathbf{C}.\mathbf{r}'}$ is in right form since $\sum_{i\in[q]} t_i = 1$. That means the first problem above is solved.

Given K' associated with a random vector $\mathbf{r}'$, it is easy to see that any combination of q users where q is bigger than the length of the vector $\mathbf{r}'$ (denoted k) can lead to $\mathbf{r}'$, as one easily solves equations k with variables q (where k is a system parameter giving the size of the vectors and matrices and q is the number of colluding users). Therefore, if there are more than k users ($q \geq k$) sharing the same m attributes, the tracking algorithm cannot exactly identify who publishes K' in the public domain. That means the second problem above is solved. More precisely, assume that $q \geq k \times t$ then we have at least t different groups of users who can produce K', note that we only need k malicious users in q users to mount the AKL attack. In the context of the AKL attack, we can say that the scheme is susceptible to the AKL attack since any polynomial time trace algorithm cannot determine who generates K' with a correctness probability larger than $1/t$.

To fight against the aforementioned attack, the parameter k (fixed at the setup phase) needs to be large enough to ensure that no more than k users share the same m attributes. In these schemes, the key and ciphertext sizes are linear in $k^2 \cdot \ell$ and $k \cdot \ell$, respectively. Note that ℓ is the length of a one-use linear secret-sharing matrix. Therefore, their scheme is completely impractical if k is large. In fact, when compared to other schemes, the competitive case of these schemes is when $k = 1$.

Performance. Consider $K'_0 = \prod_{i\in[q]} K^{t_i}_{u_i,0}$, each $K_{u_i,0}$ is a vector of $k+1$ elements. Therefore, to compute K'_0 we need $(k+1)q$ exponential operations and $(k+1)q$ multiplicative operations. Similarity for other terms $K'_1, \ldots, K'_m, K'_{\ell+1}$. Hence, to compute $K' = (K'_0, K'_1, \ldots, K'_m, K'_{\ell+1})$ we need $(m+2)(k+1)q$ exponential operations and $(m+2)(k+1)q$ multiplicative operations, where m is the number of attributes in the common set of attributes, q is the number of colluded users (note that $q \geq k$), and k is the system parameter. As discussed above, compared to other schemes, the competitive case of these schemes is when $k = 1$, which means it is efficient to build an AKL attack from a group of malicious users.

4.4 Discussion on Potential Countermeasures to the AKL Attack and Performance Evaluation

We observe that AKL attacks directly apply on CP-ABE schemes whose secret keys have the following two properties:

- First, the secret key consists of sub-key components, each of which corresponds to an attribute.
- Second, these sub-key components can be re-randomized by themselves or combined to be re-randomized.

We thus cannot apply directly AKL attacks on CP-ABE schemes, such as [30–33], whose secret keys do not have the aforementioned two properties. Regarding CP-ABE schemes that are susceptible to AKL attacks, to modify these schemes so that AKL attacks cannot be applied directly to them, we need at least to change the secret key structure in these schemes to ensure that whose secret keys do not have the aforementioned two properties. This will lead to changing the entire structure of each scheme (including encryption and decryption algorithms) and the corresponding security proof of each scheme. This is a large and challenging workload. We consider this as an open problem for future research.

Regarding the success rate of AKL attack, it is straightforward to see that the AKL attack from a single malicious is considered successful if the number of users (denotes t) sharing the same common set of attributes is large enough (at least $t \geq 2$) and there are at least one malicious user in such t users. On the other hand, the AKL attack from a group of malicious users is considered successful if the number of users (in this case denotes q) sharing the same common set of attributes is large enough (at least $q \geq 2k$) and there are at least k colluded malicious users in such q users. Note that if $k = 1$, the AKL attack from a group of malicious users is exactly the AKL attack from a single malicious user.

We exploit the AKL attack on schemes [15,22,29] and evaluate its effectiveness based on attack success rate, computational complexity and practicality, as shown in Table 1. The attack succeeds in [15] and [22] when there is at least one malicious user and the number of users sharing the same common set of attributes is at least 2 ($t \geq 2$), whereas in [29], it requires at least k colluded malicious users and at least ($q \geq 2k$) users sharing the same common set of attributes. The computational complexity of the AKL attack on [15] and [22] is $(T+2)E + (T+2)M$ which is quite efficient. In contrast, attacking [29] is significantly more complex, with a cost of $2q(k+1)(T+2)E + 2q(k+1)(T+2)M$. In terms of practicality, references [15] and [22] allow both single AKL and group AKL, whereas reference [29] mitigates single-user attacks but remains vulnerable to group-based one.

Table 1: A comparative analysis of the AKL attack applied to some schemes based on the attack's success rate, computational complexity, and practicality. Here, E represents *the exponential operation*, and M denotes *the multiplicative operation*. The variable T signifies the number of attributes within the common attribute set. Finally, MU means *colluded malicious user*

Schemes	Success rate	Complexity	Single AKL	Group AKL
[15]	$t \geq 2, 1MU$	$(T+2)E + (T+2)M$	Yes	Yes
[22]	$t \geq 2, 1MU$	$(T+2)E + (T+2)M$	Yes	Yes
[29]	$q \geq 2k, kMU$	$2q(k+1)(T+2)E + 2q(k+1)(T+2)M$	No	Yes

5 Conclusion

This work considers the attribute collusion attack on several ECC-based CP-ABE schemes. Then, we define a novel type of insider attack, called the anonymous key-leakage attack or AKL. This attack can be applied to many important existing pairing-based CP-ABE schemes. We argue that even though these pairing-based CP-ABE schemes are still secure in their security model, we should consider the AKL attack when choosing these schemes to deploy in practice.

Acknowledgement: The authors would like to thank the reviewers for their insightful comments.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Viet Cuong Trinh; methodology, Viet Cuong Trinh; investigation, Phi Thuong Le; resources, Viet Cuong Trinh; writing—original draft preparation, Phi Thuong Le; writing—review and editing, Huy Quoc Le, Viet Cuong Trinh; visualization, Phi Thuong Le, Huy Quoc Le; supervision, Viet Cuong Trinh; project administration, Viet Cuong Trinh. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Abbreviations

The following abbreviations are used in this manuscript:

AABHE	Adaptive attribute-based honey encryption

ABE	Attribute-based encryption
AKL	Anonymous key-leakage
CP-ABE	Ciphertext-policy ABE
ECC	Elliptic curve cryptography
IBE	Identity-based encryption

References

1. Sahai A, Waters B. Fuzzy identity-based encryption. In: Advances in cryptology–EUROCRYPT 2005. Berlin/Heidelberg: Springer; 2005. Vol. 3494.
2. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP '07); 2007; Berkeley, CA, USA.
3. Galbraith SD, Paterson KG, Smart NP. Pairings for cryptographers. Discrete Appl Math. 2008;156(16):3113–21. doi:10.1016/j.dam.2007.12.010.
4. Yao X, Chen Z, Tian Y. A lightweight attribute-based encryption scheme for the Internet of Things. Future Gener Comput Syst. 2015;49(1):104–12. doi:10.1016/j.future.2014.10.010.
5. Ding S, Li C, Li H. A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT. IEEE Access. 2018;6:27336–45. doi:10.1109/ACCESS.2018.2836350.
6. Wang Y, Chen B, Li L, Ma Q, Li H, He D. Efficient and secure ciphertext-policy attribute-based encryption without pairing for cloud-assisted smart grid. IEEE Access. 2020;8:40704–13. doi:10.1109/ACCESS.2020.2976746.
7. Sowjanya K, Dasgupta M. A ciphertext-policy attribute based encryption scheme for wireless body area networks based on ECC. J Inf Secur Appl. 2020;54(C):102559. doi:10.1016/j.jisa.2020.102559.
8. Herranz J. Attribute-based encryption implies identity-based encryption. IET Inf Secur. 2017;11(6):332–337. doi:10.1049/iet-ifs.2016.0490.
9. Boneh D, Papakonstantinou PA, Rackoff C, Vahlis Y, Waters B. On the impossibility of basing identity based encryption on trapdoor permutations. In: 2008 49th Annual IEEE Symposium on Foundations of Computer Science; 2008; Philadelphia, PA, USA.
10. Papakonstantinou P, Rackoff C, Vahlis Y. How powerful are the DDH hard groups? Cryptology eprint Archive. 2012;2012/653:1–31.
11. Gaborit P, Hauteville A, Phan DH, Tillich JP. Identity-based encryption from codes with rank metric. In: Advances in cryptology–CRYPTO 2017. Cham, Switzerland: Springer; 2017. Vol. 10403 .
12. Döttling N, Garg S. Identity-based encryption from the diffie-hellman assumption. J ACM. 2021;68(3):14.
13. Izabachène M, Prabel L, Roux-Langlois A. Identity-based encryption from lattices using approximate trapdoors. In: Information security and privacy. Cham, Switzerland: Springer; 2023. Vol. 13915 .
14. Herranz J. Attacking pairing-free attribute-based encryption schemes. IEEE Access. 2020;8:222226–32. doi:10.1109/ACCESS.2020.3044143.
15. Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Public key cryptography–PKC 2011. Berlin/Heidelberg: Springer; 2011. Vol. 6571.
16. Rouselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption. In: Proceedings of the ACM Conference on Computer and Communications Security; 2013; New York, NY, USA.
17. Hohenberger S, Waters B. Attribute-based encryption with fast decryption. In: Public-key cryptography–PKC 2013. Berlin/Heidelberg: Springer; 2013. Vol. 7778.
18. Attrapadung N. Dual system encryption via doubly selective security: framework, fully secure functional encryption for regular languages, and more. In: Advances in cryptology–EUROCRYPT 2014. Berlin/Heidelberg: Springer; 2014. Vol. 8441.
19. Wee H. Dual system encryption via predicate encodings. In: Theory of cryptography. Berlin/Heidelberg: Springer; 2014. Vol. 8349.
20. Agrawal S, Chase M. FAME: fast attribute-based message encryption. In: Proceedings of the ACM Conference on Computer and Communications Security; 2017; New York, NY, USA.

21. Agrawal S, Chase M. Simplifying design and analysis of complex predicate encryption schemes. In: Advances in cryptology–EUROCRYPT 2017; Berlin/Heidelberg: Springer; 2017. Vol. 10210.
22. Riepel D, Wee H. FABEO: fast attribute-based encryption with optimal security. In: Proceedings of the ACM Conference on Computer and Communications Security; 2022; New York, NY, USA.
23. Shen H, Zhou J, Wu G, Zhang M. Multi-keywords searchable attribute-based encryption with verification and attribute revocation over cloud data. IEEE Access. 2023;11:139715–27. doi:10.1109/ACCESS.2023.3334733.
24. Venema M, Alpár G. GLUE: generalizing unbounded attribute-based encryption for flexible efficiency trade-offs. In: Public-key cryptography–PKC 2023. Cham, Switzerland: Springer; 2023. Vol. 13940 .
25. Siyal R, Jun L, Elaffendi M, Asim M, Iftikhar S, Alnashwan R, et al. Adaptive attribute-based honey encryption: a novel solution for cloud data security. Comput Mater Contin. 2025;82(2):2637–64. doi:10.32604/cmc.2025.058717.
26. Luo F, Al-Kuwari S. Generic construction of black-box traceable attribute-based encryption. IEEE Trans Cloud Comput. 2023;11(1):942–55. doi:10.1109/TCC.2021.3121684.
27. Rouselakis Y, Waters B. Efficient statically-secure large-universe multi-authority attribute-based encryption. In: Financial cryptography and data security. Berlin/Heidelberg: Springer; 2015. Vol. 8975.
28. Malluhi QM, Shikfa A, Trinh VC. A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption. In: ASIA CCS, 2017—Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security; 2017; New York, NY, USA.
29. Chen J, Gay R, Wee H. Improved dual system ABE in prime-order groups via predicate encodings. In: Advances in cryptology–EUROCRYPT 2015. Berlin/Heidelberg: Springer; 2015. Vol. 9057.
30. Canard S, Phan DH, Trinh VC. Attribute-based broadcast encryption scheme for lightweight devices. IET Inf Secur. 2018;12(1):52–59. doi:10.1049/iet-ifs.2017.0157.
31. Attrapadung N, Hanaoka G, Yamada S. Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs. In: Advances in cryptology–ASIACRYPT 2015. Berlin/Heidelberg: Springer; 2015. Vol. 9452.
32. Attrapadung N. Dual system encryption framework in prime-order groups via computational pair encodings. In: Advances in cryptology–ASIACRYPT 2016. Berlin/Heidelberg: Springer; 2016. Vol. 10032.
33. Le HQ, Le PT, Trinh ST, Susilo W, Trinh VC. Levelled attribute-based encryption for hierarchical access control. Comput Stand Interfaces. 2025;93(2):103957. doi:10.1016/j.csi.2024.103957.