Cryptanalysis and Improvement to Two Key-Policy Attribute-Based Encryption Schemes for Weighted Threshold Gates

Yi-Fan Tseng(✉) and Pin-Hao Chen

Department of Computer Science, National Chengchi University, Taipei, Taiwan
yftseng@cs.nccu.edu.tw

Abstract. Attribute-based encryption is one of the most suitable access control mechanism for modern data sharing models. To provide better performance, lots of attribute-based encryption schemes are constructed over without pairings. However, these schemes are either with no security proofs or broken. In this manuscript, we give the cryptanalysis of two key-policy attribute-based encryption schemes for weighted threshold gates. We propose two attack methods, the first one is able to generate valid private keys without the master secret keys, and the second one is able to recover the master secret key when an attacker gathers enough number of private keys. Moreover, an improved schemes is given in this manuscript. We also present a security analysis to show that our improved scheme fix the security flaws with only one pairing added.

Keywords: attribute-based encryption · weighted threshold gates · cryptanalysis · access control · collusion attack

1 Introduction

In the era of cloud computing, multi-user scenarios have become increasingly common, and traditional one-to-one encryption mechanisms, such as RSA [13] and ElGamal encryption [3], are no longer suitable for applications nowadays. As a result, many cryptographers are turning to attribute-based encryption (ABE) [4,20] as a solution.

ABE is a type of encryption that enables access control based on attributes, rather than specific identities. This makes it ideal for multi-user scenarios where different users may have varying access rights based on their attributes. For example, in a financial setting, employees with different roles may require different levels of access to sensitive data.

While ABE has many advantages over traditional encryption methods, one of the main challenges is reducing the computational complexity involved in the encryption and decryption process. In response to this challenge, many pairing-free ABE schemes [1,2,8,9,11,12,14–16,19,21], i.e. schemes built over elliptic curves, have been proposed to simplify the process. Unfortunately, these schemes

J. C. Hung et al. (Eds.): FC 2023, LNEE 1134, pp. 137–146, 2024.
https://doi.org/10.1007/978-981-99-9342-0_15

have all been shown to be insecure. In 2017, Herranz [6] broke the schemes of [11,12]. In 2020, Tseng and Huang demonstrated a collusion attack to [2,19], and Herranz [7] further give cryptanalysis to [2,8,9,15,16,21]. Later in 2021, Tseng [17] give a attack method to [15] so that in [15] a ciphertext can be decrypted by an unauthorized user.

In this manuscript, we further show the cryptanalysis to two pairing-free ABE schemes, [5,10]. Both these two schemes are in key-policy setting, i.e., an access structure is associated with the private key, and an attribute set is related to the ciphertext. The access structures supported by both the two schemes are weighted threshold gates $(\mathbb{A}^{\mathsf{WT}}_{k,n}, \mathsf{S}_K)$, which can be satisfied by a set of weighted attributes if the summation of the weight is greater then a pre-defined threshold value k . Unfortunately, we found that both [5,10] are insecure. In this manuscript, we propose two attack methods, which can be applied to these two ABE schemes, due to the structural similarity between [5,10]. Our first attack allows a malicious user with a private key for $(\mathbb{A}^{\mathsf{WT}}_{1,n}, \mathsf{S}_K)$ to compute a private key for $(\mathbb{A}^{\mathsf{WT}}_{k,n}, \mathsf{S}_K)$ without the knowledge of the master secret key. Furthermore, our second attack method allows an attacker colluding with several users to recover the master secret key. Moreover, an improved scheme to fix the security flaws is also given in this manuscript.

1.1 Organization

The rest of the manuscript is organized as follows. In Sect. 2, we introduce the preliminaries for our work, including notations, complexity assumption, definition for ABE, etc. In Sect. 3, we briefly review on the scheme of [5], and show our proposed two attacks to [5]. An improved scheme is demonstrated in Sect. 4. For [10], we only give the high-level description for the scheme and the cryptanalysis, in order to avoid the unnecessary duplication. Finally, we conclude our work in Sect. 6.

2 Preliminaries

In this section, we give the notation used in this manuscript, and the definition of key-policy attribute-based encryption for weighted threshold gate (KP-ABE-WT).

2.1 Notations

The notations used in this manuscript are listed as follows.

- For a set S, by "$x \xleftarrow{\$} S$" we mean uniformly randomly choose an element x from S.
- For an algorithm A, we denote by "$y \leftarrow A$" that y is the output obtained by running A.
- By PPT we mean "probabilistic polynomial-time".

- By $[n, m]$ for some integers $n \leq m$, we mean $\{n, n+1, \ldots, m\}$.
- A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is said negligible in n, if for every $k \in \mathbb{N}$, there is $n_0 \in \mathbb{N}$ such that for every $n \geq n_0, |f(n)| < \frac{1}{n^k}$.

2.2 Bilinear Maps and Complexity Assumption

Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative groups with prime order p. Let g be a generator of $\mathbb{G}$. A bilinear map e, aka pairing, is defined as $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where the following properties are satisfied.

1. For all $a, b \in \mathbb{Z}_p, e(g^a, g^b) = e(g, g)^{ab}$.
2. There is an efficient algorithm to compute $e(u, v)$ for all $u, v \in \mathbb{G}$.
3. $e(g, g)$ is not the identity of $\mathbb{G}_T$.

We also give a complexity assumption which the security of our improved scheme bases on.

Definition 1 (Discrete-Log Assumption). *The discrete-log assumption says that, no PPT algorithm is able to compute* $\log_g h$ *from a given* $h \in \mathbb{G}$.

Definition 2 (M-DDH$_{\mathbb{G}_T}$ Assumption [18]**).** *Let* $a, b \xleftarrow{\$} \mathbb{Z}_p$. *Let* $e(g, g) = \mathfrak{g}$ *The M-DDH*$_{\mathbb{G}_T}$ *assumption states that, there is no PPT algorithm, given* $(g, \mathfrak{g}, g^a, \mathfrak{g}^a, \mathfrak{g}^b)$, *tells the difference between* $\mathfrak{g}^{ab}$ *and an element* $Z \xleftarrow{\$} \mathbb{G}_T$.

2.3 Access Structure

In both [5, 10], the authors propose a KP-ABE scheme for weighted threshold gates, which is defined as follows. Let S be a set of attributes. A weighted threshold gate is defined by

$$(\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}) = \{(k, n), \{w_x \mid x \in \mathsf{S}\}\},$$

where

- $w_x \geq 1$ is the weight of the attribute x;
- n is the total weight of the attributes in an attribute set S;
- $k \in [1, n]$ is the threshold.

A threshold gate is a special case of weighted threshold gate when $w_x = 1$ for all $x \in \mathsf{S}$. For a set $\mathsf{S}' \subseteq \mathsf{S}$, we say that S' satisfies $(\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S})$ if

$$\sum_{x \in \mathsf{S}'} w_x \geq k.$$

2.4 Lagrange Polynomial Interpolation

Lagrange polynomial interpolation is an algorithm to compute a polynomial f of $k-1$ degree given k points. More precisely, given k points $(x_1, y_1), \ldots, (x_k, y_k)$, the polynomial f passing the k points can be computed by

$$f(x) = \sum_{i=1}^{k} y_i \Delta_i(x),$$

where $\Delta_i(x) = \prod_{j \in [1,k] \setminus \{i\}} \frac{x - x_j}{x_i - x_j}$.

2.5 Key-Policy Attribute-Based Encryption for Weighted Threshold Gates

A KP-ABE scheme for weighted threshold gates consists of the following four algorithms $\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{KeyGen}, \mathsf{Decrypt}$.

$\mathsf{Setup}(1^\lambda)$. Taking as input the security parameter, the algorithm outputs the system parameter params and the master secret key msk. Note that params will be a implicitly input for the following algorithms.

$\mathsf{Encrypt}(\mathsf{S}, \mathsf{M})$. Taking as inputs an attribute set S and a message M, the algorithm outputs a ciphertext CT.

$\mathsf{KeyGen}(\mathsf{msk}, (\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}))$. Taking as inputs the master secret key msk and an access structure $(\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S})$ described in Sect. 2.3, the algorithm outputs a private key D.

$\mathsf{Decrypt}(\mathsf{CT}, \mathsf{D})$. Taking as inputs a ciphertext CT and a private key D, the algorithm outputs a message.

Correctness. For $\mathsf{CT} \leftarrow \mathsf{Encrypt}(\mathsf{S}_C, \mathsf{M}), \mathsf{D} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, (\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}_K))$, we have $\mathsf{M} \leftarrow \mathsf{Decrypt}(\mathsf{CT}, \mathsf{D})$ if S_C satisfies $(\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}_K)$, denoted by $\mathsf{S}_C \models (\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}_K)$.

3 Review and Cryptanalysis on Gu and Lin's KP-ABE-WT Scheme

In this section, we briefly review on the KP-ABE-WT scheme (named GL22) proposed by Gu and Lin [5] in 2022, and give the attacks to break their scheme.

3.1 Review on GL22

GL22 supports small universe, i.e., the set of all attributes in the system is polynomially large. Let $\mathcal{U}$ be the universe in GL22. We omit the description of Decrypt algorithm since our attack method does not depend on it.

$\mathsf{Setup}(1^\lambda)$. Taking as input the security parameter, the algorithm performs as follows.

1. Choose a group $\mathbb{G}$ over an elliptic curve. Let g be a generator of $\mathbb{G}$ and p be the prime order of $\mathbb{G}$.
2. Choose $t \xleftarrow{\$} \mathbb{Z}_p$ and choose $t_x \xleftarrow{\$} \mathbb{Z}_p$ for each attribute $x \in \mathcal{U}$.
3. Compute $T = g^t$ and $T_x = g^{t_x}$ for each attribute $x \in \mathcal{U}$.
4. Choose a cryptographic hash function $H : \mathbb{G} \rightarrow \mathbb{Z}_p$.
5. Output $\mathsf{params} = (p, g, T, \{T_x\}_{x\in\mathcal{U}}, H)$ and $\mathsf{msk} = (t, \{t_x\}_{x\in\mathcal{U}})$.

$\mathsf{Encrypt}(\mathsf{S}, \mathsf{M})$. Taking as inputs an attribute set S_C and a message $\mathsf{M} \in \mathbb{Z}_p$, the algorithm performs as follows.

1. Choose $s \xleftarrow{\$} \mathbb{Z}_p$.
2. Compute $C = \mathsf{M} \cdot H(T^s), C' = g^s$.
3. Compute $C_x = T_x^s$ for each $x \in \mathsf{S}_C$.
4. Output $\mathsf{CT} = (C, C'\{C_x\}_{x\in\mathsf{S}_C})$.

$\mathsf{KeyGen}(\mathsf{msk}, (\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}_K))$. Taking as inputs the master secret key msk and an access structure $(\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}_K) = \{(k,n), \{w_x \mid x \in \mathsf{S}_K\}\}$, the algorithm performs as follows.

1. For each attribute $x \in \mathsf{S}_K$ and $y \in [1, w_x]$, choose $r_{x,y} \xleftarrow{\$} \mathbb{Z}_p$. Let $R = \{r_{x,y} \mid x \in \mathsf{S}_C, y \in [1, w_x]\}$.
2. For each $r_{x,y} \in R$, compute the corresponding Lagrange basis polynomial
$$\varDelta_{r_{x,y}}(z) = \prod_{r\in R\setminus\{r_{x,y}\}} \frac{z - r}{r_{x,y} - r}.$$
3. Choose a $(k-1)$-degree polynomial q such that $q(0) = t$.
4. For each $r_{x,y} \in R$, compute $q_{x,y} = q(r_{x,y}), D_{x,y} = q_{x,y} + t_x$.
5. Output $\mathsf{D} = ((\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}_K), \{D_{x,y}, \varDelta_{r_{x,y}}(0)\})$.

3.2 Cryptanalysis on GL22

Our attack algorithms focus on collusion attacks, that is, to recover the master secret msk or generate another private key without knowing msk, given enough amount of private keys D. For simplicity, we will consider access structures for threshold gate, i.e., $w_x = 1$ for all $x \in \mathsf{S}_K$ for describing the intuition of our attack algorithms.

Attack 1. Suppose a user query a private key for the access structure $(\mathbb{A}_{1,2}^{\mathsf{WT}}, \mathsf{S}_K) = \{(1,2), \{w_x = 1 \mid x \in \mathsf{S}_K\}\}$ and $\mathsf{S}_K = \{A, B\}$ for some attributes $A, B \in \mathcal{U}$. Observe that when $k = 1$, the polynomial chosen in Step 2 of KeyGen algorithm is actually a constant polynomial $q(z) = t$, and hence[1]

$$D_A = q(r_A) + t_A = t + t_A$$
$$D_B = q(r_B) + t_B = t + t_B.$$

[1] We omit the subscript y here since all the weight are 1 and $y \in [1, 1]$.

Then the user is able to generate a private key for $(\mathbb{A}^{\mathsf{WT}}_{2,2}, \mathsf{S}_K) = \{(2,2), \{w_x = 1 \mid x \in \mathsf{S}_K\}\}$ and $\mathsf{S}_K = \{A, B\}$, given the private key $\mathsf{D}' = ((\mathbb{A}^{\mathsf{WT}}_{1,2}, \mathsf{S}_K), \{D_A, \Delta_{r_A}(0), D_B, \Delta_{r_B}(0)\})$. The details are shown as follows.

1. Choose $r_A, r_B \xleftarrow{\$} \mathbb{Z}_p$.
2. Compute $\Delta_{r_A}(z) = \frac{z - r_B}{r_A - r_B}, \Delta_{r_B}(z) = \frac{z - r_A}{r_B - r_A}$.
3. Choose $a \xleftarrow{\$} \mathbb{Z}_p$ and compute $D_A = ar_A + (t + t_A), D_B = ar_B + (t + t_A)$.
4. Output the private key for $(\mathbb{A}^{\mathsf{WT}}_{2,2}, \mathsf{S}_K) = \{(2,2), \{w_x = 1 \mid x \in \mathsf{S}_K\}\}$.

In Step 3, our attack algorithm implicitly set the polynomial $q(z) = az + t$. and no master secret is needed since $(t + t_A, t + t_B)$ has been given to the user in the private key D'. Besides, our attack algorithm can be extended into any general weighted threshold gate $(\mathbb{A}^{\mathsf{WT}}_{k,n}, \mathsf{S}_K)$, given a private key $\mathsf{D}_{\mathcal{U}}$ for $(\mathbb{A}^{\mathsf{WT}}_{1,|\mathcal{U}|}, \mathcal{U}) = \{(1, |\mathcal{U}|), \{w_x = 1 \mid x \in \mathsf{S}_K\}\}$, since

- the computation of $\Delta_{r_x}(0)$ for $x \in \mathsf{S}_K$ depends only on the choice of randomness in Step 1, which is fully controlled by the attack algorithm;
- the computation of $D_x = q(r_x) + t_x = a_{k-1}(r_x)^{k-1} + \cdots + a_1 r_A + (t + t_A)$ can be done given $\mathsf{D}_{\mathcal{U}}$.

Attack 2. Consider a private key[2] $\mathsf{D} = ((\mathbb{A}^{\mathsf{WT}}_{k,n}, \mathsf{S}_K), \{D_x, \Delta_{r_x}(0)\})$ for an access structure $\{(k,n), \{w_x = 1 \mid x \in \mathsf{S}_K\}\}$ and $\mathsf{S}_K \subseteq \mathcal{U}$, where $D_x = q(r_x) + t_x$ for $x \in \mathsf{S}_K$. By the correctness of Lagrange polynomial interpolation, we have that, for any subset $U \subset \mathsf{S}_K$ with $|U| = k$,

$$\sum_{x \in U} q(r_x)\Delta_{r_x}(0) = q(0) = t. \tag{1}$$

Therefore, by Eq. (1), we have

$$\sum_{x \in U} D_x \Delta_{r_x}(0) = \sum_{x \in U} (q(r_x) + t_x)\Delta_{r_x}(0) = t + \sum_{x \in U} \Delta_{r_x}(0) \cdot t_x. \tag{2}$$

As $D_x, \Delta_{r_x}(0)$ for $x \in U$ is given in D, there are only $|U| + 1 = k + 1$ unknown variables in Eq. (2), i.e. $t, \{t_x\}_{x \in U}$. Therefore, given private keys $\mathsf{D}^{(1)}, \ldots, \mathsf{D}^{(k+1)}$ for $\{(k,n), \{w_x = 1 \mid x \in \mathsf{S}^{(1)}_K\}\}, \ldots, \{(k,n), \{w_x = 1 \mid x \in \mathsf{S}^{(k+1)}_K\}\}$, respectively, such that $U \subseteq \mathsf{S}^{(1)}_K \cap \cdots \cap \mathsf{S}^{(k+1)}_K$, anyone is able to recover $t, \{t_x\}_{x \in U}$ by solving a linear equation system.

We give the following simple example to illustrate **Attack 2**. Let

$$\begin{aligned}
\mathsf{D}^{(1)} &= ((\mathbb{A}^{\mathsf{WT}}_{2,3}{}^{(1)}, \mathsf{S}^{(1)}_K), \{D^{(1)}_A, \Delta_{r^{(1)}_A}(0), D^{(1)}_B, \Delta_{r^{(1)}_B}(0), D^{(1)}_C, \Delta_{r^{(1)}_C}(0)\}),\\
\mathsf{D}^{(2)} &= ((\mathbb{A}^{\mathsf{WT}}_{2,3}{}^{(2)}, \mathsf{S}^{(2)}_K), \{D^{(2)}_A, \Delta_{r^{(2)}_A}(0), D^{(2)}_B, \Delta_{r^{(2)}_B}(0), D^{(2)}_D, \Delta_{r^{(2)}_D}(0)\}),\\
\mathsf{D}^{(3)} &= ((\mathbb{A}^{\mathsf{WT}}_{2,3}{}^{(3)}, \mathsf{S}^{(3)}_K), \{D^{(3)}_A, \Delta_{r^{(3)}_A}(0), D^{(3)}_B, \Delta_{r^{(3)}_B}(0), D^{(3)}_E, \Delta_{r^{(3)}_E}(0)\}),
\end{aligned}$$

[2] We again omit the subscript y here since all the weight are 1 and $y \in [1, 1]$.

be the private keys for

$$\begin{aligned}
\mathbb{A}_{2,3}^{\mathsf{WT}^{(1)}} &= \{(2,3), \{w_x = 1 \mid x \in \mathsf{S}_K^{(1)}\}\}, \mathsf{S}_K^{(1)} = \{A, B, C\},\\
\mathbb{A}_{2,3}^{\mathsf{WT}^{(2)}} &= \{(2,3), \{w_x = 1 \mid x \in \mathsf{S}_K^{(2)}\}\}, \mathsf{S}_K^{(2)} = \{A, B, D\},\\
\mathbb{A}_{2,3}^{\mathsf{WT}^{(3)}} &= \{(2,3), \{w_x = 1 \mid x \in \mathsf{S}_K^{(3)}\}\}, \mathsf{S}_K^{(3)} = \{A, B, E\}.
\end{aligned}$$

In this example, $U = \{A, B\} \subseteq \mathsf{S}_K^{(1)} \cap \mathsf{S}_K^{(2)} \cap \mathsf{S}_K^{(3)}$. By Eq. (2) we have

$$\begin{cases}
D_A^{(1)} \cdot \varDelta_{r_A^{(1)}}(0) + D_B^{(1)} \cdot \varDelta_{r_B^{(1)}}(0) = t + \varDelta_{r_A^{(1)}}(0) \cdot t_A + \varDelta_{r_B^{(1)}}(0) \cdot t_B,\\
D_A^{(2)} \cdot \varDelta_{r_A^{(2)}}(0) + D_B^{(2)} \cdot \varDelta_{r_B^{(2)}}(0) = t + \varDelta_{r_A^{(2)}}(0) \cdot t_A + \varDelta_{r_B^{(2)}}(0) \cdot t_B,\\
D_A^{(3)} \cdot \varDelta_{r_A^{(3)}}(0) + D_B^{(3)} \cdot \varDelta_{r_B^{(3)}}(0) = t + \varDelta_{r_A^{(3)}}(0) \cdot t_A + \varDelta_{r_B^{(3)}}(0) \cdot t_B.
\end{cases}$$

Thus (t, t_A, t_B) can be easily recovered by solving the linear equation systems shown above.

4 An Improved Scheme

The main reason causing the security flaws shown in Sect. 3.2 is that, the information of the master secret key has been directly exposed in a private key. Equation (2) shows the linear relation between D and msk. A straightforward way to fix the problem is to raise D to the power of g. However, this method would make the number of pairings be $\mathcal{O}(\mathsf{S}_K)$ in $\mathsf{Decrypt}$ algorithm.

To reduce the number of pairings as possible, we move the most of the computations of $\mathsf{GL22}$ to the group $\mathbb{G}_T$, and randomize the components $D_{x,y}$ in D with a new randomness β. We give our improved version below. Let $\mathfrak{g} = e(g, g)$.

Setup is the same as $\mathsf{GL22}$, except that $T = \mathfrak{g}^t$ and $T_x = \mathfrak{g}^{t_x}$ for $x \in \mathcal{U}$.

$\mathsf{Encrypt}$ is the same as $\mathsf{GL22}$, except that $C' = \mathfrak{g}^s$ and an additional component $C'' = g^s$ is added.

KeyGen is the same as $\mathsf{GL22}$, except that

1. a random number β is chosen from $\mathbb{Z}_p$;
2. $D_{x,y}$ is computed as $q_{x,y} + t_x + \beta$;
3. an additional component $E = g^\beta$ is added.

$\mathsf{Decrypt}(\mathsf{CT}, \mathsf{D})$. Taking as inputs a ciphertext $\mathsf{CT} = (C, C', C''\{C_x\}_{x \in \mathsf{S}_C})$ and a private key $\mathsf{D} = ((\mathbb{A}_{k,n}^{\mathsf{WT}}, \mathsf{S}_K^{(1)}), \{D_{x,y}, \varDelta_{r_{x,y}}(0)\}, E)$, the algorithm performs as follows.

1. Compute $F = e(C'', E) = e(g^s, g^\beta) = \mathfrak{g}^{s\beta}$.
2. For $x \in \mathsf{S}_K$ and $y \in [1, w_x]$, compute

$$F_{x,y} = \frac{(C')^{D_{x,y}}}{C_x \cdot F} = \frac{\mathfrak{g}^{s(q_{x,y} + t_x + \beta)}}{\mathfrak{g}^{st_x} \cdot \mathfrak{g}^{s\beta}} = \mathfrak{g}^{sq_{x,y}}.$$

3. Compute
$$T^s = \mathfrak{g}^{st} = \prod_{x \in \mathsf{S}_K} F_{x,y}^{\Delta_{r_x,y}(0)}.$$
4. Recover $\mathsf{M} = C/H(T^s)$.

Correctness. The correctness nearly follows that of GL22, except the difference due the newly-added randomness β. Thus, we cancel the term $\mathfrak{g}^{s\beta}$ in Step 2 of Decrypt algorithm, with the cost of only 1 pairing.

Security Analysis. To see why the attacks shown in Sect. 3.2 do not work in our improved scheme, note that there is a newly-added randomness β is added in KeyGen algorithm. β will be sampled each time KeyGen algorithm is perform. Besides, the information of β is hidden in E, which is impossible to be retrieved due to the discrete-log assumption. Therefore, Eq. (2) shown in Sect. 3.2 will become

$$\sum_{x \in U} D_x \Delta_{r_x}(0) = t + \sum_{x \in U} \Delta_{r_x}(0) \cdot t_x + \beta \cdot \left(\sum_{x \in U} \Delta_{r_x}(0) \right). \qquad (3)$$

Thanks to the existence of β, the number of unknown variable now increases with the number of private keys obtained by the attacker, which makes the attacker impossible to recover msk by solving a linear equation system. Furthermore, according to the M-DDH$_{\mathbb{G}_T}$ assumption, even with the knowledge of $(g, \mathfrak{g}, C'' = g^s, C' = \mathfrak{g}^s, T = \mathfrak{g}^t)$, no PPT algorithm distinguishes $T^s = \mathfrak{g}^{st}$ from an uniformly random element in $\mathbb{G}_T$. This fact implies that the information of M is hidden from the attacker's view, and thus guarantees the security of our improved scheme.

5 Cryptanalysis on Lin *et al.*'s KP-ABE-WT Scheme

In this section, we show the insecurity of the KP-ABE-WT scheme (named LHXS17) proposed by Lin *et al.* [10] in 2017. Due to the conceptual similarity of GL22 and LHXS17, we only give the high-level description for LHXS17 to avoid the unnecessary duplication, and show the intuition for the corresponding cryptanalysis.

LHXS17 is almost identical to GL22, except that, in GL22 the Langrange coefficients $\Delta_{r_x,y}(0)$ is included as a part of the private key D, while in LHXS17 $\Delta_{r_x,y}(0)$ is computed in Decrypt algorithm. By this operation, GL22 has lower computation cost in Decrypt algorithm than LHXS17, with the cost of doubling the private key size. Besides, since $\Delta_{r_x,y}(0)$ needs to be computed by user, in LHXS17 the randomness r_x used in KeyGen algorithm is set to be some public indices instead of fresh random numbers, which allows anyone to compute $\Delta_{r_x,y}(0)$ for any user. Therefore, our attack methods shown in Sect. 3.2 work well for LHXS17.

6 Conclusion

With the raise of cloud computing, ABE has become one of the most suitable cryptographic primitives for multi-user scenario. In order to reduce the computation cost, lots of ABE schemes are designed without using pairings. However, all of these schemes are either flawed or lacking of security proofs. In this manuscript, we find out the security issues of [5,10] by giving two attack methods. Our attack methods are generate private keys without msk, and even recover msk. Moreover, an improved scheme have been given to fix the security problem of [5,10]. Our improved scheme requires only one pairing, which may be an optimal result when constructing ABE in pairing groups. In the future, we will prove the security of the improved scheme, and attempt to further improve the efficiency and the expressiveness of the proposed scheme.

Acknowledgment. This work was partially supported by the National Science and Technology Council of Taiwan, under grants 111-2221-E-004-005-, 111-2218-E-004-001-MBK.

References

1. Cheng, R., Wu, K., Su, Y., Li, W., Cui, W., Tong, J.: An efficient ECC-based CP-ABE scheme for power IoT. Processes **9**(7) (2021). https://doi.org/10.3390/pr9071176. https://www.mdpi.com/2227-9717/9/7/1176
2. Ding, S., Li, C., Li, H.: A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT. IEEE Access **6**, 27336–27345 (2018). https://doi.org/10.1109/ACCESS.2018.2836350
3. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) Advances in Cryptology, pp. 10–18. Springer, Berlin Heidelberg, Berlin, Heidelberg (1985)
4. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 89–98 (2006). https://doi.org/10.1145/1180405.1180418
5. Gu, Z., Lin, G.: A pairing-free key policy weighted attributed-based encryption. Available at SSRN 4173677 (2022)
6. Herranz, J.: Attribute-based encryption implies identity-based encryption. IET Inf. Secur. **11**(6), 332–337 (2017)
7. Herranz, J.: Attacking pairing-free attribute-based encryption schemes. IEEE Access 8, 222,226–222,232 (2020). https://doi.org/10.1109/ACCESS.2020.3044143
8. Karati, A., Amin, R., Biswas, G.P.: Provably secure threshold-based ABE scheme without bilinear map. Arab. J. Sci. Eng. **41**, 3201–3213 (2016)
9. Khandla, D., Shahy, H., Bz, M.K., Pais, A.R., Raj, N.: Expressive CP-ABE scheme satisfying constant-size keys and ciphertexts. Cryptology ePrint Archive, Report 2019/1257 (2019).https://ia.cr/2019/1257
10. Lin, G., Hong, H., Xia, Y., Sun, Z.: An expressive, lightweight and secure construction of key policy attribute-based cloud data sharing access control. J. Phys.: Conf. Series **910**(1), 012,010 (2017)

11. Odelu, V., Das, A.K.: Design of a new cp-abe with constant-size secret keys for lightweight devices using elliptic curve cryptography. Security Commun. Netw. **9**(17), 4048–4059 (2016). https://doi.org/10.1002/sec.1587, https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1587
12. Odelu, V., Das, A.K., Khurram Khan, M., Choo, K.R., Jo, M.: Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts. IEEE Access **5**, 3273–3283 (2017)
13. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
14. Sowjanya, K., Dasgupta, M., Ray, S.: A lightweight key management scheme for key-escrow-free ecc-based CP-ABE for IoT healthcare systems. J. Syst. Architect. **117**, 102,108 (2021). https://doi.org/10.1016/j.sysarc.2021.102108, https://www.sciencedirect.com/science/article/pii/S1383762121000849
15. Sowjanya, K., Dasgupta, M., Ray, S., Obaidat, M.S.: An efficient elliptic curve cryptography-based without pairing KPABE for internet of things. IEEE Syst. J. **14**(2), 2154–2163 (2020). https://doi.org/10.1109/JSYST.2019.2944240
16. Tan, S.Y., Yeow, K.W., Hwang, S.O.: Enhancement of a lightweight attribute-based encryption scheme for the internet of things. IEEE Internet Things J. **6**(4), 6384–6395 (2019). https://doi.org/10.1109/JIOT.2019.2900631
17. Tseng, Y.-F.: Cryptanaylsis to Sowjanya et al.'s ABEs from ECC. In: Tsihrintzis, G.A., Wang, S.-J., Lin, I.-C. (eds.) 2021 International Conference on Security and Information Technologies with AI, Internet Computing and Big-data Applications, pp. 287–294. Springer International Publishing, Cham (2023). https://doi.org/10.1007/978-3-031-05491-4_29
18. Tseng, Y.F., Liu, Z.Y., Tso, R.: Practical inner product encryption with constant private key. Appl. Sci. **10**(23) (2020)
19. 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 **8**, 40704–40713 (2020). https://doi.org/10.1109/ACCESS.2020.2976746
20. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) Public Key Cryptography - PKC 2011, pp. 53–70. Springer, Berlin Heidelberg, Berlin, Heidelberg (2011)
21. Yao, X., Chen, Z., Tian, Y.: A lightweight attribute-based encryption scheme for the internet of things. Future Gen. Comput. Syst. **49**, 104–112 (2015). https://doi.org/10.1016/j.future.2014.10.010, https://www.sciencedirect.com/science/article/pii/S0167739X14002039