Advances in the Theory of Nonlinear Analysis and its Applications 7 (2023) No. 3, 30-44.
https://doi.org/10.17762/atnaa.v7.i3.248
Available online at www.https://atnaea.org/
Research Article

Advances in the Theory of Nonlinear Analysis and its Applications

ISSN: 2587-2648 Peer-Reviewed Scientific Journal

# Distributed attribute-based encryption with small ciphertext

Hiromu Komazawa[a], Tomofumi Matsuzawa[a]

[a] *Department of Information Sciences, Tokyo University of Science, Japan*

**Abstract**

Currently, cloud services are widely used to manage data. Hence protecting their security is extremely important. This requires meticulous control of access rights for viewers, especially for confidential corporate data in the cloud. Attribute-based encryption (ABE) is a next-generation cryptographic technique that can efficiently control access rights by associating decryption conditions based on the attributes of a given viewer with a ciphertext to allow only viewers who satisfy the decryption conditions. However, ABE requires a key generator, which is an organization that generates users public and private keys with sufficient authority to decrypt the ciphertext. To solve this security problem, distributed ABE (DABE) methods without decryptable institutions have been presented.DABE also involves a problem that the more ORs in the decryption condition, the larger the ciphertext size becomes, which is a major challenge for practical use. In this study, we propose a DABE method with reduced ciphertext size and processing load without the need for an authority that can decrypt ciphertext. To this end, based on DABE, we adopt a multi-value attribute algorithm with attributes as a set with restrictions on decryption conditions. The results of a security evaluation are presented to show that the security strength of the proposed method is equivalent to that of the existing DABE scheme.Furthermore, the increase in ciphertext size was greatly reduced, which was a major issue for the existing DABE.We also confirmed that the proposed method performed better in terms of decryption time under certain conditions. This research contributes to the development of ABE by realizing a highly practical implementation with both efficient processing and strong capabilities.

*Keywords:* Cloud Computing, Cloud Security, Attribute-based Encryption
*2010 MSC:* 94A60, 68P25

*Email addresses:* hkomazaw118@gmail.com (Hiromu Komazawa), t-matsu@is.noda.tus.ac.jp (Tomofumi Matsuzawa)

*Received July 30, 2023; Accepted: August 25, 2023; Online: October 17, 2023.*

## 1. Introduction

At present, file sharing and data management are widely implemented using cloud services. Thus, protecting the security of information is extremely important, especially when managing confidential corporate data or privacy-sensitive information such as medical records on the cloud. One existing method is to encrypt data using public key cryptography [1][2] to enable administrators to control the access rights of each user viewer. However, this approach requires knowing the number of viewers in advance, and controlling access rights may become complicated depending on the number of viewers. ABE [3][4] is a next-generation cryptographic technique that replaces these conventional methods. In addition to sharing statistical data used by medical agencies [5] and video distribution services [6], ABE has also been commonly applied to blockchain techniques [7][8] in recent years. The adoption of efficient mechanisms to control access rights using ABE is expected to centralize the management of multiple viewers at the cryptographic level without identifying individuals.

The ciphertext used in ABE has the same security strength as existing public key cryptosystems, but requires a key generator hereinafter referred to as an Attribute Certificate Authority (AA) to generate a private key for each visitor. The AA is an organization similar to a certification authority (CA) for communication via SSL encryption communication, but it can decrypt the ciphertext and has a relatively broad authority. DABE methods with distributed AAs have been developed since the 2010s to prevent the centralization of authority and resources [9][10].

In addition, ABE methods, including DABE involve the problem that the more complex the decryption conditions, the larger the ciphertext becomes. In recent years, some studies on techniques to reduce size of ciphertext and the associated processing load [11][12][13] have attracted attention as a topic of active research.

In this study, we propose an ABE method that solves the problems of AA in terms of processing load and ciphertext size for improved practicality.

## 2. Preliminaries

### *2.1. Abbreviations*

The following abbreviations are used in this manuscript.

$Z$ : Group of integers

$G$ : Group of rational point on an elliptic curve

$G_T$ : Group of pairing outputs

$W$ : Decryption condition

$L$ : User's condition

$M$ : Message mapped to $G_T$

$CT$ : Ciphertext

$PK$ : Public key

$SK$ : Secret key

### *2.2. Related Works*

#### *2.2.1. ABE with small processing load*

An ABE method with some additional restrictions on the decryption conditions were proposed to solve the problem of increasing ciphertext size and processing load due to increasingly complex decryption conditions [11].

ABE has been shown to increase the number of ciphertexts with the OR of the decryption conditions, and the ciphertexts corresponding to the split conditions are combined and retained as shown in Equation 1.

e.g., $W = 1$ *and* (2 *or* 3) *and* (4 *or* 5).

$$CT = \langle CT_{1,2,4}, CT_{1,2,5}, CT_{1,3,4}, CT_{1,3,5} \rangle. \tag{1}$$

As a conventional method to reduce processing load, a technique was provided in [13] in which each attribute was represented as a subset of an attribute set containing arbitrary attribute values while restricting the decryption conditions to AND only (hereinafter referred to as multi-value by attribute decryption). Each user was assumed to have one value among each attribute. In this related research, the processing load was further reduced applying multi-value attributes to some attribute sets. Based on this restriction, the size of the decryption condition was reduced by splitting the decryption condition into parameters corresponding to the attribute values without splitting the decryption condition, shown in Figure 1.

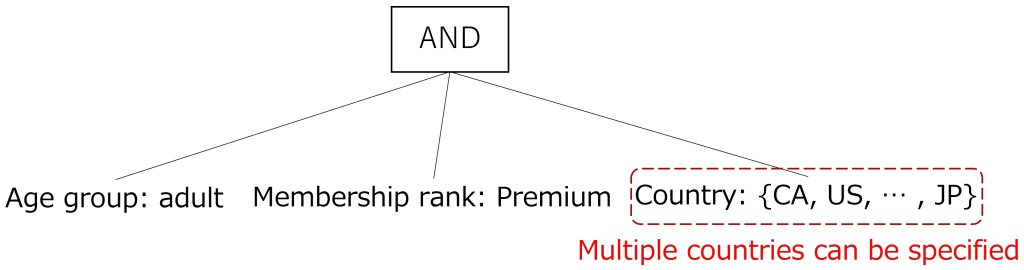


Figure 1: Restrictions on decryption conditions

Before discussing the configuration method, we explain the definition of the elliptic curves used in ABE here. In general, an elliptic curve on a finite field $F_q$ can be expressed as follows.
$a_i \in F_q$, $x, y \in F_q \times F_q$,

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \tag{2}$$

However, it can be expressed as follows when $q = p^k$(prime number $p \neq 2, 3$).
$a, b \in F_q$, $x, y \in F_q \times F_q$,

$$y^2 = x^3 + ax + b. \tag{3}$$

Moreover, a point on an elliptic curve has the following additive properties.

Addition of points on elliptic curves:

The addition $P_1 + P_2$ of two points $P_1 + P_2$ on an elliptic curve is shown in Figure 2. The intersections of elliptic curves always intersect at three points unless they are tangent, and the remaining intersection through the two points $P_1 + P_2$ is $P_3$. Finally, the $x$-axis symmetry point of $P_3$ is $P_1 + P_2$.

Doubling a point on elliptic curves:

The doubling $2P_1$ of a point $P_1$ on an elliptic curve is illustrated in Figure 3. Let $P_2$ be the intersection of tangent lines at point $P_1$. Let $2P_1$ be the point of $x$-axis symmetry of $P_2$.

For any point $P$, the point $O$ for which $P + O = O + P = P$ is referred to as the infinity point. Using the additive property, if $P, 2P, 3P...$ and so forth in sequence, we eventually reach the infinity point and return

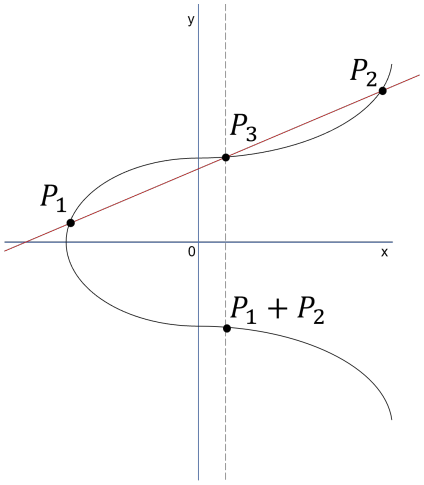


Figure 2: Addition of points on elliptic curves

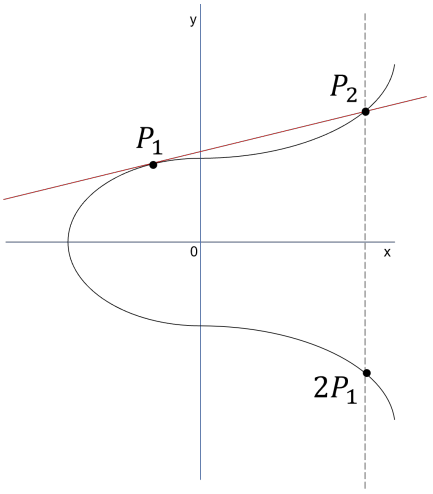


Figure 3: Doubling a point on elliptic curves

to $P$ again, and thus the point group on the elliptic curve with $P$ as the primitive is the cyclic group of the additive method. Also, finding $n$ satisfying $Q = nP(n \in Z)$ (elliptic discrete logarithm problem) is difficult, and elliptic curves with large group rank are thus widely used in cryptography.

The configuration method is as follows. ABE uses pairing maps in which pairings have the relation below. $a, b \in Z$

$$e(aP, bQ) = e(P, Q)^{ab}. \tag{4}$$

Based on this relationship, the specific configuration method is shown below. The agency designated to perform each operation is indicated after a "/". (The formulas in the notation correspond to calculations on elliptic curves based on the proposed method.)

*Setup / AA*

Among the attribute set $L$, determine the multi-value attribute set $\hat{L}$ and an other attribute set $\check{L}$ to which the multi-value attribute method is not applied (hereinafter referred to as single-value attribute). Then, generate the secret key of the AA and publish the following values as public keys, including the point on the elliptic curve corresponding to each attribute value.
$p$: Prime number, $P \in G$ $(|G| = p)$, $\{Q_k \in G\}(k \in \check{L})$, $\{R_{i,j} \in G\}(i \in \hat{L}, j \in \hat{L}_i)$,
00 $\omega \in Z_p$: Random number.

$$SK = \omega, \tag{5}$$

$$PK = \langle P, e(P, P)^{\omega}, \{Q_k\}, \{R_{i,j}\}\rangle. \tag{6}$$

*KeyGeneration / AA*

Generate a secret key for user $U$ using $PK$ and $SK$.
$L_U = [\check{L}_U, \hat{L}_U]$: User's attribute set,
$\{x_k \in Z_p\}(k \in \check{L}_U)$: Random numbers correspond to the values of single-value attribute,
$\{s_i \in Z_p\}(i \in \hat{L}_U)$: Random numbers correspond to the values of multi-value attribute,
$s = \sum_{i \in \hat{L}_U} s_i$, $u, v \in Z_p$: Random numbers.

$$SK_U = \langle SK_U^{'}, SK_U^{''}, SK_U^{'''}\rangle, \tag{7}$$

$$SK_U^{'} = (\omega - s + k)P + \sum_{i \in \hat{L}_U} (s_i P + x_i R_{i,\hat{L}_{U,i}}) - vP + \sum_{k \in \check{L}_U} uQ_k, \tag{8}$$

$$SK_U^{''} = uP, \tag{9}$$

$$SK_U^{'''} = \{x_i P\}(i \in \hat{L}_U) = \{SK_{U,i}^{'''}\}. \tag{10}$$

$SK_U^{'''}$ is enumerated in attribute order.

*Encryption / User*

Generate ciphertext for decryption conditions using $PK$.

$W = [\check{W}, \hat{W}]$: Decryption condition, $r \in Z_p$: Random number, $M \in G_T$: Message.

$$CT = \langle CT^{'}, CT^{''}, CT^{'''}, CT^{''''} = \{CT_{i,j}^{''''}\}(i \in \hat{W}_U, j \in \hat{W}_{U,i})\rangle \tag{11}$$

$$= \langle M \cdot e(P,P)^{\omega r}, rP, \sum_{k \in \check{W}} rQ_k, \{rR_{i,\hat{W}_i}\}\rangle. \tag{12}$$

The $CT^{''''}$ is enumerated in attribute order, but the publication of the corresponding attribute values depends on the specific case.

*Decryption / User*

User $U$ obtains $SK_U$ and decrypts $CT$.

$$M = \frac{CT^{'} \cdot \prod_{i \in \hat{L}_U} e(CT_{i,\hat{L}_{U,i}}^{''''}, SK_{U,i}^{'''}) \cdot e(CT^{'''}, SK_U^{''})}{e(CT^{''}, SK_U^{'})}. \tag{13}$$

It is known that with conventional ABE, the AA can decrypt the ciphertext independently. However the more complex the conditions, the greater the computational resources required by the AA.

*2.2.2. ABE with any number of possible key-issuing authorities*

To solve the problem of AA having large authority and requiring large computational resources, DABE with no decryptable institutions [10] was proposed. Based on the original DABE [9], in which AAs were distributed by attribute, the authors proposed a model in which an agency referred to as a CA manages an AA that cannot decrypt the ciphertext. This means that the resources of the AA are distributed and no agency can decrypt the ciphertext.

The configuration method is as follows. The specific configuration method is divided into components. The agency designated to perform each operation is indicated after a "/". (The formulas in the notation correspond to calculations on elliptic curves based on the proposed method.)

*Setup / CA*

Generate four points that comprise public keys on the same elliptic curve of rank $p$.

$p$: prime number, $P, Q, R, S \in G$.

$$PK = \langle P, Q, R, S\rangle. \tag{14}$$

*UserCreation / User*

Generate the user's public key using $PK$.

$U$: User, $u \in Z_p$: Random number.

$$PK_U = \langle PK_U^{'}, PK_U^{''}\rangle = \langle Q + uR, uP\rangle. \tag{15}$$

*AuthorityCreation / each AA*

Generate a secret key for each AA.

$a$: AA's attribute, $H_a() = \{0,1\}^* \to Z_p$: Random hash function.

$$SK_a = H_a(a). \tag{16}$$

*AttributeRequestPK / each AA*

Using $PK$ and $SK_a$, each AA generates an attribute public key corresponding to an attribute within the decryption condition.

$i$: Attribute within decryption condition, $a$: AA's attribute.

$$PK_i = \begin{cases} H_a(i)P & \text{if } i = a \\ \text{NULL} & \text{else} \end{cases}. \tag{17}$$

*AttributeRequestSK / each AA*

Use $PK$ and $SK_a$ to generate an attribute secret key if user $U$ had an attribute managed by AA.

$L_U$: User's attribute set, $i \in L_U$: User's attribute, $a$: AA's attribute, $x_a \in Z_p$: Random number

$$SK_i = \begin{cases} \langle SK_i^{'}, SK_i^{''}, SK_i^{'''} \rangle & \text{if } i \in L_U \text{ and } i = a \\ \text{NULL} & \text{else} \end{cases}, \tag{18}$$

$$\langle SK_i^{'}, SK_i^{''}, SK_i^{'''} \rangle = \langle H_a(i)(PK_U^{'}) + x_a S, x_a P, H_a(i)(PK_U^{''}) \rangle \tag{19}$$

$$= \langle H_a(i)(Q + uR) + x_a S, x_a P, H_a(i)(uP) \rangle. \tag{20}$$

*Encryption / User*

Using $PK$ and the attribute public key $PK_i$ corresponding to the decryption condition, generate the ciphertext corresponding to the split decryption condition, respectively.

$N$: Number of split decryption conditions, $W_j (j \in [N])$ Split decryption conditions, $M \in G_T$: Message.

$$CT = \langle CT_{W_1}, ..., CT_{W_N} \rangle. \tag{21}$$

In each $W_j$,

$r_j \in Z_p$: Random number.

$$CT_{W_j} = \langle CT_{W_j}^{'}, CT_{W_j}^{''}, CT_{W_j}^{'''}, CT_{W_j}^{''''} \rangle \tag{22}$$

$$= \langle M \cdot e(Q, \sum_{i \in W_j} PK_i)^{r_j}, r_j P, r_j R, r_j S \rangle. \tag{23}$$

*Decryption / User*

Obtain the attribute secret key $SK_i$ corresponding to the attributes held by user $U$ and decrypt the ciphertext $CT_{L_U}$ corresponding to the attribute set $L_U$ of $U$.

$L_U$: User's attribute set.

Also, let the following equations to $\alpha$ and $\beta$.

$\alpha = \sum_{i \in L_U} H_i(i),\ \beta = \sum_{i \in L_U} x_i.$

$$M = \frac{CT_{L_U}^{'} \cdot e(CT_{L_U}^{'''}, \sum_{i \in L_U} SK_i^{'''}) \cdot e(CT_{L_U}^{''''}, \sum_{i \in L_U} SK_i^{''})}{e(\sum_{i \in L_U} SK_i^{'}, CT_{L_U}^{''})} \tag{24}$$

$$= \frac{M \cdot e(Q, \alpha P)^r \cdot e(rR, \alpha u P) \cdot e(rS, \beta P)}{e(\sum_{i \in L_U} (H_i(i)(Q + uR) + x_i S), rP)} \tag{25}$$

$$= \frac{M \cdot e(Q, P)^{r\alpha} \cdot e(R, P)^{ur\alpha} \cdot e(S, P)^{r\beta}}{e(Q, P)^{r\alpha} \cdot e(R, P)^{ur\alpha} \cdot e(S, P)^{r\beta}} \tag{26}$$

$$= M. \tag{27}$$

As disscussed above in 2.2.1, the more ORs in the decryption conditions, the greater the processing load.

## 3. Methods

We propose an ABE with reduced ciphertext size and processing load via restrictions on decryption conditions, which does not include any agency that can decrypt the ciphertext.

The configuration method is described as follows. This model is based on DABE as described in Section 2.2.2 (hereinafter referred to as "Concealed-DABE") and extends the multi-value attribute algorithm as shown in Figure 4 (Other attribute: single-value attribute). As in the model described in Section 2.2.1 (hereinafter referred to as Restricted-ABE), some restrictions are imposed on the decryption conditions.

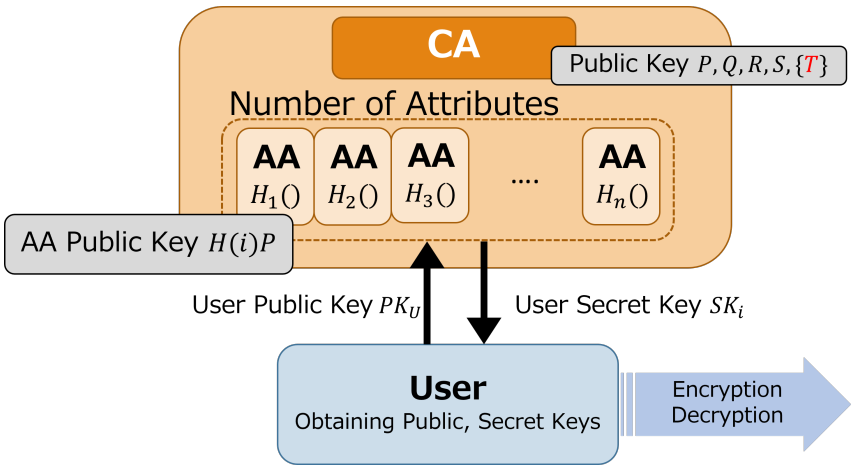


Figure 4: Configuration method

The specific configuration of the method is divided into components. The agency designated to perform each operation is indicated after a "/".

*Setup / CA*

Determine the single-value attribute set $\check{L}$ and the multi-value attribute set $\hat{L}$ among the attributes set $L$. Generate four points comprising public keys on the same elliptic curve of rank $p$ and points corresponding to the attribute values of the multi-value attribute.

$p$: prime number, $P, Q, R, S \in G$, $\{T_{i,j} \in G\}(i \in \hat{L}, j \in \hat{L}_i)$.

$$PK = \langle P, Q, R, S, \{T_{i,j}\}\rangle. \tag{28}$$

*UserCreation / User*

Use $PK$ to generate a user public key containing parameters corresponding to the attribute values possessed by user $U$.

$L_U = [\check{L}_U, \hat{L}_U]$: User's attribute set, $u \in Z_p$: Random number.

$$PK_U = \langle PK^{'}, PK^{''}, PK^{'''}\rangle, \tag{29}$$

$$PK_U^{'} = Q + uR, \tag{30}$$

$$PK_U^{''} = uP, \tag{31}$$

$$PK_U^{'''} = \{uT_{i,\hat{L}_{U,i}}\}(i \in \hat{L}_U) = \{PK_{U,i}^{'''}\}. \tag{32}$$

*AuthorityCreation / each AA*

Generate a secret key for each AA.

$a$: AA's attribute, $H_a() = \{0,1\}^* \to Z_p$: Random hash function.

$$SK_a = H_a(a). \tag{33}$$

*AttributeRequestPK / each AA*

Using $PK$ and $SK_a$, each AA generates an attribute public key corresponding to an attribute within the decryption condition.

$i$: Attribute within decryption condition, $a$: AA's attribute.

$$PK_i = \begin{cases} H_a(i)P & \text{if } i = a \\ \text{NULL} & \text{else} \end{cases}. \tag{34}$$

*AttributeRequestSK / each AA*

Use $PK$, $PK_U$, and $SK_a$ to generate an attribute secret key if user $U$ had an attribute managed by the AA; if the attribute managed by the AA is a multi-value attribute, the $PK_U$ parameters used differ (Equation 34).

$L_U = [\check{L}_U, \hat{L}_U]$: User's attribute set, $i \in L_U$: User's attributes, $a$: AA's attribute, $x_a \in Z_p$: Random number.

$$SK_i = \begin{cases} \langle SK_i^{'}, SK_i^{''}, SK_i^{'''} \rangle & \text{if } i \in L_U \text{ and } i = a \\ \text{NULL} & \text{else} \end{cases}, \tag{35}$$

$$SK_i^{'} = \begin{cases} H_a(i)(PK_U^{'}) + x_a S \\ \quad = H_a(i)(Q + uR) + x_a S & \text{if } a \in \check{L} \\ H_a(i)(PK_{U,i}^{'''}) + x_a S \\ \quad = H_a(i)(Q + uT_{i,\hat{L}_{U,i}}) + x_a S & \text{if } a \in \hat{L} \end{cases} \tag{36}$$

$$SK_i^{''} = x_a P, \tag{37}$$

$$SK_i^{'''} = H_a(i)(PK_U^{''}) = H_a(i)(uP). \tag{38}$$

*Encryption / User*

Generate ciphertext using $PK$ and the attribute public key $PK_i$ corresponding to the decryption condition.

$W = [\check{W}, \hat{W}]$: Decryption condition, $M \in G_T$: Message, $r \in Z_p$: Random number.

$$CT = \langle CT^{'}, CT^{''}, CT^{'''}, CT^{''''}, CT^{'''''} \rangle, \tag{39}$$

$$CT^{'} = M \cdot e(Q, \sum_{i \in W} PK_i)^r, \tag{40}$$

$$CT^{''} = rP, \tag{41}$$

$$CT^{'''} = rR, \tag{42}$$

$$CT^{''''} = rS, \tag{43}$$

$$CT^{'''''} = \{rT_{i,j}\}(i \in \hat{W}, j \in \hat{W}_i) = \{CT_{i,j}^{'''''}\}. \tag{44}$$

*Decryption / User*

Obtain the attribute secret key $SK_i$ corresponding to the attributes held by user $U$ and decrypt the ciphertext $CT_{L_U}$ corresponding to the attribute set $L_U$ of $U$.

$L_U = [\check{L}_U, \hat{L}_U]$: User's attribute set.

$$M = \frac{CT^{'} \cdot \prod_{i \in \hat{L}_U} e(CT_{i,\hat{L}_{U,i}}^{'''''}, SK_i^{'''}) \cdot e(CT^{'''}, \sum_{j \in \check{L}_U} SK_j^{'''}) \cdot e(CT^{''''}, \sum_{k \in L_U} SK_k^{''})}{e(\sum_{k \in L_U} SK_k^{'}, CT^{''})}. \tag{45}$$

Let $f(M)$ be an equation that maps $L_U$ in Equation 23 to $\check{L}_U$. Also, let the following equations to $\alpha$ and $\beta$.

$\alpha = \sum_{i \in \hat{L}_U} H_i(i),\ \beta = \sum_{i \in \hat{L}_U} x_i.$

$$= \frac{CT' \cdot \prod_{i \in \hat{L}_U} e(CT''''_{i,\hat{L}_{U,i}}, SK'''_i) \cdot e(CT''', \sum_{j \in \check{L}_U} SK'''_j) \cdot e(CT'''', \sum_{k \in L_U} SK''_k)}{e(\sum_{i \in \hat{L}_U} (H_a(i)(PK'''_{U,i}) + x_a S) + \sum_{j \in \check{L}_U} (H_a(i)(PK'_U) + x_a S), rP)}, \tag{46}$$

$$= \frac{e(Q, \sum_{i \in \hat{L}_U} PK_i)^r \cdot \prod_{i \in \hat{L}_U} e(CT''''_{i,\hat{L}_{U,i}}, SK'''_i)}{e(\sum_{i \in \hat{L}_U} (H_a(i)(PK'''_{U,i}) + x_a S), rP)} \cdot f(M), \tag{47}$$

$$= \frac{e(Q, \sum_{i \in \hat{L}_U} H_i(i)P)^r \cdot \prod_{i \in \hat{L}_U} e(rT_{i,\hat{L}_{U,i}}, H_i(i)(uP))}{e(\sum_{i \in \hat{L}_U} (H_a(i)(Q + uT_{i,\hat{L}_{U,i}}) + x_a S), rP)} \cdot f(M), \tag{48}$$

$$= \frac{e(Q,P)^{r\alpha} \cdot \prod_{i \in \hat{L}_U} e(T_{i,\hat{L}_{U,i}}, P)^{ruH_i(i)}}{e(Q,P)^{r\alpha} \cdot \prod_{i \in \hat{L}_U} e(T_{i,\hat{L}_{U,i}}, P)^{ruH_i(i)}} \cdot f(M), \tag{49}$$

$$= M. \tag{50}$$

## 4. Results

In this section, we present the results of an evaluation of the security and performance of the proposed approach in terms of ciphertext size and processing load.

### *4.1. Security*

The security of the proposed method conforms to that of the base Concealed-DABE. In the present work, we mainly describe the extended functionality.

#### *4.1.1. Security by adding multi-value attributes*

Based on the parameters of the proposed method, the following parameters are added to the model of Concealed-DABE: $PK'''_U$ for the user public key, $SK'''_i$ for the attribute secret key, and $CT''''$ for the ciphertext. The parameters that could be decrypted if compromised are $U$ for the user, $H_i()$ for the AA, and $r$ for the ciphertext. Although $PK'''_U$, $SK'''_i$, and $CT''''$ all contain the above parameters, they are difficult to decrypt owing to the difficulty of the elliptic discrete logarithm problem. In addition, the above parameters are not shared among agencies, and no agency can decode them. Therefore, the proposed method has the same security strength as Concealed-DABE.

### *4.2. Comparison of the performance*

Table 1 shows the configuration of the proposed approach compared with that of related studies.

Table 1: Structure of the method

| | Restricted-ABE | Concealed-DABE | Proposed method |
|---|---|---|---|
| multi-value attribute | ○ | × | ○ |
| Decentralization of AA | × | ○ | ○ |

The parameters used in the evaluation are shown in Table 2.

Table 2: Parameter List

| Parameter | Description |
|---|---|
| $W = [\check{W}, \hat{W}]$ | Decryption condition (for methods using multi-value attribute, divide into single-value attributes and multi-value attributes) |
| $n$ | Number of attributes included in decryption condition |
| $\check{n}$ | Number of single-value attributes included in decryption condition |
| $\hat{n}$ | number of multi-value attribute in the decryption condition |
| $\hat{n}_i$ | Number of attribute values for each multi-value attribute in the decryption condition |
| $\hat{n}_{sum}$ | Total number of attribute values of multi-value attributes included in the decryption condition ($\sum_{i \in \hat{W}} \hat{n}_i$) |
| $\hat{n}_{prod}$ | Number of combinations of attribute values of multi-value attributes included in the decryption condition ($\prod_{i \in \hat{W}} \hat{n}_i$) |
| $C_G$ | Processing time for scalar multiplication on $G$ |
| $C_{G_T}$ | Processing time for power operations on $G_T$ |
| $C_{pair}$ | Processing time for pairing operations |

The proposed approach was implemented using a PBC library and measurements were performed on a PC with specifications as shown in Table 3.

Table 3: PC specification

| | |
|---|---|
| OS | "Ubuntu 20.04 64-bit" |
| CPU | "AMD Ryzen 5800X@4.8GHz" |
| Memory | "32GB" |
| Compiler | "g++ (9.4.0)" |

The decryption conditions used for the actual measurement were as follows. Decryption condition 1 with a small number of $\hat{n}_{prod}$ combinations of conditions and decryption condition 2 with a large number of $\hat{n}_{prod}$ were specified. Specific examples of each decryption condition are also shown in Figure 5 and Figure 6.
[Decryption condition 1]:
The specific values of the parameters used for decryption condition 1 were as follows.

$$n = 4,\ \check{n} = 1,\ \hat{n} = 3,\ \hat{n}_1, \hat{n}_2, \hat{n}_3 = 2,\ \hat{n}_{sum} = 6,\ \hat{n}_{prod} = 8. \tag{51}$$

An example of a decryption condition that satisfies the condition above is given as follows.

$$W = 0\ \textit{and}\ (2\ \textit{or}\ 3)\ \textit{and}\ (4\ \textit{or}\ 5)\ \textit{and}\ (6\ \textit{or}\ 7). \tag{52}$$

Single and multi-value attribute sets were used as follows.

$$= [\check{W}, \hat{W}] = \left[\{0\}, \left\{ \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 6 \\ 7 \end{bmatrix} \right\}\right]. \tag{53}$$

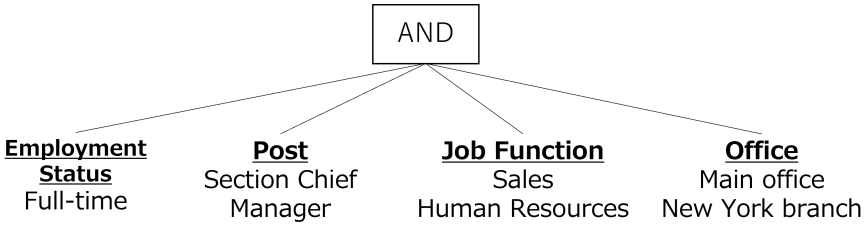


Figure 5: An concrete example of decryption condition 1

[Decryption condition 2]:
The specific values of the parameters used for decryption condition 1 were as follows.

$$n = 7,\ \check{n} = 2,\ \hat{n} = 5,\ \hat{n}_1, \hat{n}_2, \hat{n}_3, \hat{n}_4, \hat{n}_5 = 5,\ \hat{n}_{sum} = 25,\ \hat{n}_{prod} = 3125. \tag{54}$$

An example of a decryption condition that satisfies the condition above is given as follows.

$$\begin{gathered} W = 0 \textit{ and } 1 \textit{ and } (2 \textit{ or } 3 \textit{ or } 4 \textit{ or } 5 \textit{ or } 6) \textit{ and } (7 \textit{ or } 8 \textit{ or } 9 \textit{ or } 10 \textit{or } 11) \\ \textit{and } (12 \textit{ or } 13 \textit{ or } 14 \textit{ or } 15 \textit{ or } 16) \textit{ and } (17 \textit{ or } 18 \textit{ or } 19 \textit{ or } 20 \textit{ or } 21) \\ \textit{and } (22 \textit{ or } 23 \textit{ or } 24 \textit{ or } 25 \textit{ or } 26), \end{gathered} \tag{55}$$

Single and multi-value attribute sets were used as follows.

$$= [\check{W}, \hat{W}] = \left[ \{0,\ 1\}, \left\{ \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, \begin{bmatrix} 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{bmatrix}, \begin{bmatrix} 12 \\ 13 \\ 14 \\ 15 \\ 16 \end{bmatrix}, \begin{bmatrix} 17 \\ 18 \\ 19 \\ 20 \\ 21 \end{bmatrix}, \begin{bmatrix} 22 \\ 23 \\ 24 \\ 25 \\ 26 \end{bmatrix} \right\} \right]. \tag{56}$$

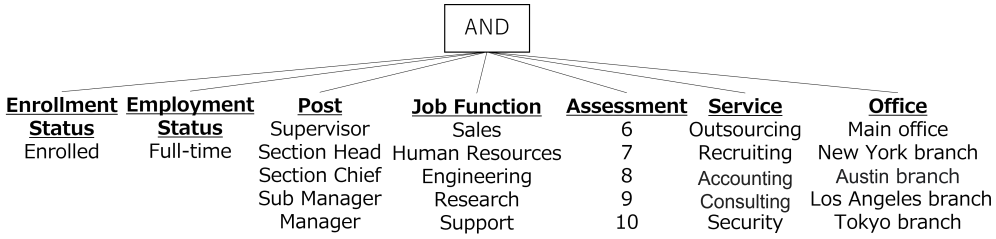


Figure 6: An concrete example of decryption condition 2

*4.2.1. Comparison of ciphertext size*

Here, we compare the sizes of the ciphertexts used in related methods with those of the proposed approach. First, we compare the generalized ciphertext size, as shown in Table 4 with the measured ciphertext size shown in Tables 5 and 6.

Table 4: Generalized ciphertext size

| | Restricted-ABE | Concealed-DABE | Proposed method |
|---|---|---|---|
| Number of $G$ | $\hat{n}_{sum} + 2$ | $\hat{n}_{prod} \cdot 3$ | $\hat{n}_{sum} + 3$ |
| Number of $G_T$ | 1 | $\hat{n}_{prod}$ | 1 |

Table 5: Ciphertext size in the experiment (decryption condition 1)

| | Restricted-ABE | Concealed-DABE | Proposed method |
|---|---|---|---|
| Ciphertext size (bytes) | 648 | 2584 | 713 |
| Number of $G$ | 8 | 24 | 9 |
| Number of $G_T$ | 1 | 8 | 1 |

Table 6: Ciphertext size in the experiment (decryption condition 2)

| | Restricted-ABE | Concealed-DABE | Proposed method |
|---|---|---|---|
| Ciphertext size (bytes) | 1883 | 1009375 | 1948 |
| Number of $G$ | 27 | 9375 | 28 |
| Number of $G_T$ | 1 | 3125 | 1 |

Comparing the ciphertext size of the proposed method and Restricted-ABE, we can see that of the proposed method had one larger $G$ value. In addition, the difference was relatively small because the $G$ value can be compressed at output time in terms of implementation.In contrast, the ciphertext size of the proposed method and Concealed-DABE was smaller than that of the proposed method except when $\hat{n}_{prod} = 1$ (indicating that multi-value attributes were not used or each multi-value attribute had one attribute value), and the size difference increased with $\hat{n}_{prod}$. Therefore, the ciphertext size of the proposed method was approximately the same as that of Restricted-ABE and was smaller than that of Concealed-DABE.

*4.2.2. Comparison of processing times*

Next, we compare the processing times for encryption and decryption of related techniques with those of the proposed method. Processing times were evaluated based on the number of scalar multiplications of $G$, power operations on $G_T$, and pairing operations (processing times for other operations are within the error range). In decryption, the conditions corresponding to each ciphertext or the attribute values corresponding to the ciphertext parameters may or may not be disclosed (depending on the application design). If not disclosed, all ciphertexts that match the decryptor's conditions must be searched. Assuming this case, the decryption time required to decrypt all ciphertexts is also shown in Table 7.

Table 7: Generalized processing time

| | Restricted-ABE | Concealed-DABE | Proposed method |
|---|---|---|---|
| Encryption | $(\hat{n}_{sum}+2)C_G + C_{G_T}$ | $\hat{n}_{prod}(3C_G + C_{G_T} + C_{pair})$ | $(\hat{n}_{sum}+3)C_G + C_{G_T} + C_{pair}$ |
| Decryption | $(\hat{n}+2)C_{pair}$ | $3C_{pair}$ | $(\hat{n}+3)C_{pair}$ |
| Decryption (All Pattern) | $(\hat{n}_{prod}\cdot\hat{n}+2)C_{pair}$ | $\hat{n}_{prod}\cdot 3C_{pair}$ | $(\hat{n}_{prod}\cdot\hat{n}+3)C_{pair}$ |

Comparing the proposed method with Restricted-ABE, the proposed method took $C_G + C_{pair}$ time for encryption and $C_{pair}$ time more processing time for decryption.

We also compared the proposed method with Concealed-DABE. In encryption, the proposed method does not require $\hat{n}_{prod}$ street multipliers, so the processing time was small for most conditions. However, when $\hat{n}_{sum} > \hat{n}_{prod}$ (e.g., each multi-value attribute has one attribute value), Concealed-DABE was faster. When decrypting a single ciphertext, the proposed method took longer for the same number of multi-value attributes. Given that parallel processing is possible for all decryptions, the processing time per thread is shown in Table 8.

Table 8: Processing time of each thread in decryption parallelization

| | Restricted-ABE | Concealed-DABE | Proposed method |
|---|---|---|---|
| 1 | $2C_{pair}$ | $3C_{pair}$ | $3C_{pair}$ |
| 2 | $\hat{n}C_{pair}$ | $3C_{pair}$ | $\hat{n}C_{pair}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $\hat{n}_{prod}$ | $\hat{n}C_{pair}$ | $3C_{pair}$ | $\hat{n}C_{pair}$ |
| $\hat{n}_{prod}+1$ | $\hat{n}C_{pair}$ | | $\hat{n}C_{pair}$ |

The proposed method was slower than related techniques when the number of multi-value attribute was three or more. The encryption and decryption processing times of the proposed method involve a trade-off relationship depending on the number of multi-value attributes.

The average of five actual measurements is shown in Tables 9 and 10.

Table 9: Processing time (decryption condition 1)

| | Restricted-ABE | Concealed-DABE | Proposed method |
|---|---|---|---|
| Encryption (msec) | 7.196 | 25.819 | 8.645 |
| Decryption (msec) | 3.509 | 2.219 | 3.520 |
| Decryption-Parallel (msec) | 3.318 | 2.849 | 3.432 |

Table 10: Processing time (decryption condition 2)

| | Restricted-ABE | Concealed-DABE | Proposed method |
|---|---|---|---|
| Encryption (msec) | 23.947 | 9884.575 | 25.028 |
| Decryption (msec) | 4.913 | 2.350 | 5.642 |
| Decryption-Parallel (msec) | 1104.460 | 707.590 | 1105.317 |

In the actual measurement, the difference in computation speed was caused by the state of the terminal, but it may be observed that there was no significant difference in performance between the proposed method and Restricted-ABE. In addition, the parallel processing of decryption depended considerably on the number of CPU cores used, and the processing time became slower when $\hat{n}_{prod}$ exceeded the number of cores. Therefore, the processing time of the proposed method was equivalent to that of Restricted-ABE, and it exhibited a processing performance advantage over Concealed-DABE in encryption, as well as an inferior or equivalent performance in decryption.

## 5. Discussion

The proposed method has the same security strength as the base Concealed-DABE and the same performance as Restricted-ABE. In other words, the ciphertext size and encryption time are reduced by eliminating a large authority. Therefore, the proposed approach is superior to Restricted-ABE in terms of security and to Concealed-DABE in terms of the ciphertext size and encryption processing performance. In addition, if the decryption conditions are not disclosed and there are less than three multi-value attributes and many attribute values of multi-value attributes in the decryption conditions, the results confirmed that our method was superior to Concealed-DABE in terms of ciphertext size, encryption time, and decryption time. We will consider the feasibility of practical use of this method in future research.

As we reduced ciphertext size while maintaining high security strength, the proposed method enables a reduction of communication capacity and storage. Moreover, the more multi-value attribute used, the more advantageous they are for encryption. However, because the decryption load increases, the maximum number of multi-value attributes is approximately 5, considering the use of various edge devices. This corresponds

to decryption condition 2, and the number of attributes suffices for the use case for corporate workers, as shown in Figure 6. Because AA increases as the number of attributes increases, attribute prerequisites must be set according to the environment of servers and edge devices.

Some future issues of note are described below.

One issue is that we need to consider resistance to collusive attack. This describes a problem in which users who do not satisfy the decryption conditions can decrypt each other's content by sharing their own attribute secret keys. As a countermeasure, we also conducted an experimental evaluation in which the decryption conditions were not disclosed and the assigned attributes were unknown. To solve this problem fundamentally, parameters corresponding to the decryptor conditions would need to be included in the ciphertext, in addition to multiplying the attribute secret keys of the attributes included in the decryption conditions. AA and the cloud that stores the ciphertext can also work together to restrict user behavior and prevent collusion attacks [14].

Another issue is that the proposed method is less flexible in terms of decryption conditions than Concealed-DABE because the format of the attribute set and the attributes possessed by the user are predetermined. For practical purposes, changing attributes already set for an application is relatively rare. In view of this, if the assumption is that each user has one value for each attribute, the flexibility is relatively similar to that of Concealed-DABE. If the user has more than one value for each attribute, the following scheme can be applied.

e.g., $W$: Decryption condition, $U$: User, $L_U$: User's condition.

$$PK = \left\langle P, Q, R, S, \begin{bmatrix} T_2 \\ T_3 \end{bmatrix}, \begin{bmatrix} T_4 \\ T_5 \\ T_6 \end{bmatrix} \right\rangle, \tag{57}$$

$$W = [\check{W}, \hat{W}] = \left[ \{0\}, \left\{ \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \text{ and } 6 \end{bmatrix} \right\} \right], \tag{58}$$

$$L_U = [\check{L}_U, \hat{L}_U] = \left[ \{0\}, \left\{ 2, \begin{bmatrix} 5 \\ 6 \end{bmatrix} \right\} \right]. \tag{59}$$

For the above conditions of 5 and 6, the public keys $T_5$ and $T_6$ can be replaced by $T_5 + T_6$. However, expressing the logical product of attribute values between different attributes is difficult and requires some ingenuity.

## 6. Conclusions

In this study, we have proposed a DABE method with reduced processing load and ciphertext size that does not implement any authority that can decrypt the ciphertext. Based on DABE without a decryption-capable agency, the proposed method avoid the need for any agency with strong authority, which is a challenge in ABE. Furthermore, the processing load and ciphertext size of the proposed DABE scheme are reduced by applying a multi-value attribute algorithm that considers some attributes as a set of attribute values (a logical OR representation of attribute values). The results of an experimental evaluation of the performance of the proposed method showed that the ciphertext size and encryption time of the base DABE without multi-value attributes increased with a pattern of conditions divided by logical ORs in the decryption conditions, whereas the proposed method suppressed the increase with the number of attribute values of the multi-value attributes. The proposed method was also effective in terms of decryption time. We confirmed that the proposed method also showed an advantage in terms of decryption time under certain conditions. However, as a precondition, the user's attribute conditions must be formatted and the attributes to be used as multi-value attributes must be determined in advance. Although some improvements such as countermeasures against collusion attacks are necessary, the results demonstrated that the proposed ABE is more practical than existing methods owing to its high processing power and robust security capabilities.

## References


[1] Joseph K. Liu, X.Huang, R.Lu, J.Li, Fine-Grained Two-Factor Access Control for Web-Based Cloud Computing Services, IEEE Transactions on Information Forensics and Security Vol. 11 (2016), No. 3, 484-497.

[2] Qijun G., Liu P., Lee W.-C., Chu C., KTR: An efficient key management scheme for secure data access control in wireless broadcast services, IEEE Transactions on Dependable and Secure Computing Vol. 6 (2009), No. 3, 188-201.

[3] V.Goyal, O.Pandey, A.Sahai, B.Waters, Attribute-based encryption for fine-grained access control of encrypted data, Proc. of the 13th ACM conference on Computer and communications securityCCS'06 (2006), 89-98.

[4] J.Bethencourt, A.Sahai, B.Waters, Ciphertext-Policy Attribute-Based Encryption, Proc. of the 2007 IEEE Symposium on Security and Privacy (2007), 321-334.

[5] C.Guo, R.Zhuang, Y.Jie, Y.Ren, T.Wu, K.Choo, Fine-grained Database Field Search Using Attribute-Based Encryption for E-Healthcare Clouds, J Med Syst Vol. 40 (2016), No. 235, 1-8.

[6] John P.Papanis, Stavros I.Papapanagiotou, Aziz S.Mousas, Georgios V.Lioudakis, Dimitra I.Kaklamani, Iakovos S.Venieris, On the use of Attribute-Based Encryption for multimedia content protection over Information-Centric Networks, Special Issue: Information-Centric Networking for Multimedia, Social and Peer-to-Peer Communications (2014), 422-435.

[7] Z.Guo, G.Wang, Y.Li, J.Ni, R.Du, M.Wang, Accountable Attribute-Based Data Sharing Scheme Based on Blockchain for Vehicular Ad Hoc Network, Proc. of the 2022 IEEE Internet of Things Journal (2022), 1-1.

[8] A.Ghorbel, M.Ghorbel, M.Jmaiel, Accountable privacy preserving attribute-based access control for cloud services enforced using blockchain, IJIS Security 21 (2022), 489-508.

[9] S.Müller, S.Katzenbeisser, C.Eckert, Distributed Attribute-Based Encryption, Proc. of ICISC'08 LNCS 5461 (2009), 20-36.

[10] G. Ohtake, Y. Doi, Attribute-based encryption with arbitrary number of authoritiesAttribute-based encryption with arbitrary number of authorities, IEICE technical report (2010), 153-158.

[11] K. Ogawa, G. Ohtake, G. Hanaoka, S. Yamada, K.Kasamatsu, T. Yamakawa, H. Imai, Partially Wildcarded Ciphertext-Policy Attribute-BasedEncryption and Its Performance Evaluation, IEICE Trans. Fundamentals Vol. E100-A (2017), No. 9, 1846-1856.

[12] A.Saidi, O.Nouali, A.Amira, SHARE-ABE: an efficient and secure data sharing framework based on ciphertext-policy attribute-based encryption and Fog computing Cluster Computing 25 (2022), 167-185.

[13] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters, Fully Secure Functional Encryption: Attribute-based Encryption and (Hierarchical) Inner Product Encryption, Proc. of Eurocrypt'10 LNCS 6110 (2010), 62-91.

[14] S.Zhao, R.Jiang, B.Bhargava, RL-ABE: A Revocable Lattice Attribute Based Encryption Scheme Based on R-LWE Problem in Cloud Storage, IEEE Transactions on Services Computing Vol. 15 (2022), No. 2, 1026-1035.