I. Security Proof

The security of the proposed FS-DABPE construction is guaranteed by the following theorem.

Theorem 1. *If the q-DPBDHE3 assumption holds, then the proposed FS-DABPE construction is statically secure against any PPT adversary with a challenge matrix of size less than $q \times q$ and $l, \ell \leq q$. More formally, we have that*

$$\mathsf{Adv}_{\mathcal{C}}^{q\text{-DPBDHE3}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{\text{FS-DABPE}}(\lambda).$$

Proof. Throughout the proof, we will demonstrate that if there exists a PPT adversary $\mathcal{A}$ that can break the static security of the proposed FS-DABPE scheme with a non-negligible advantage, then we can construct a PPT algorithm $\mathcal{C}$ that can also break the q-DPBDHE3 assumption with a non-negligible advantage. More precisely, the algorithm $\mathcal{C}$ is constructed via simulating the security game played interactively with $\mathcal{A}$. The details of the simulation are specified as follows.

Initialization phase. The algorithm $\mathcal{C}$ initially receives an instance of the q-DPBDHE3 problem $(\boldsymbol{f}, R)$, where

$$\boldsymbol{f} = \Big(\mathtt{BG}, g_1^s, g_2^s, \{g_1^{a^i}\}_{i\in[2q], i\neq q+1}, \{g_k^{b_j a^i}\}_{(k,i,j)\in[2,2q,q], i\neq q+1}, \\ \{g_k^{s/b_i}\}_{(k,i)\in[2,q]}, \{g_k^{s a^i b_j / b_{j'}}\}_{(k,i,j,j')\in[2,q+1,q,q], j\neq j'}\Big),$$

$\mathtt{BG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g_1, g_2)$ and $R \in \mathbb{G}_T$. Additionally, it also obtains the challenge access structure $(\boldsymbol{A}^*_{n\times m}, \psi^*)$, tag t^*, time interval τ^* and messages M_0, M_1 from the adversary $\mathcal{A}$.

Then, the algorithm $\mathcal{C}$ directly assigns $h = g_1$. Furthermore, it chooses random exponents $\delta_0, \ldots, \delta_l, \gamma_0, \ldots, \gamma_\ell \in \mathbb{Z}_p$, and lets $v_i = g_1^{\delta_i} \cdot g_1^{a^{q-i+1}}$ for $i \in [l]$ as well as $h_j = g_1^{\gamma_j} \cdot g_1^{a^{q-j+1}}$ for each $j \in [\ell]$. Particularly, it sets $h_0 = g_1^{\gamma_0} \cdot \prod_{j=1, j\neq t^*}^{\ell} h_j^{-1}$ and $v_0 = g_1^{\delta_0} \cdot \prod_{i=1}^{l} v_i^{-b_{\tau^*}[i]}$.

Finally, the algorithm $\mathcal{C}$ returns the global public parameter $\mathtt{GP} = \big\{\mathtt{BG}, h, \boldsymbol{h}, \boldsymbol{v}, \mathcal{T}, \mathcal{I}, \mathcal{U}, \mathcal{S}, \phi(\cdot)\big\}$ to the adversary $\mathcal{A}$. The two hash functions $H(\cdot)$ and $F(\cdot)$ are simulated as random oracles by the algorithm $\mathcal{C}$ with two lists $\mathcal{L}_H$ and $\mathcal{L}_F$.

Query generation phase. After seeing $\mathtt{GP}$, the adversary $\mathcal{A}$ produces a set $\mathcal{S}_c \subseteq \mathcal{S}$ of corrupted attribute authorities and another set $\mathcal{S}_h \subseteq \mathcal{S}$ of attribute authorities requiring for the corresponding public keys. In addition, it also generates a set of secret key queries $\mathcal{Q}_{sk} = \{(\mathtt{gid}_i, \tau_i, S_{\mathtt{gid}_i})\}_{i=1}^k$. Then, the adversary $\mathcal{A}$ chooses two messages $M_0, M_1 \in \mathbb{G}_T$, and sends them along with $\{\mathcal{S}_c, \mathcal{S}_h, \mathcal{Q}_{sk}\}$ to the algorithm $\mathcal{C}$.

Response phase. In this phase, as a response, the algorithm $\mathcal{C}$ needs to produce public keys of attribute authorities in $\mathcal{S}_h$ and secret keys for tuples in $\mathcal{Q}_{sk}$. To this end, $\mathcal{C}$ first constructs a matrix $\boldsymbol{A}'_{n\times m}$ as follows: Let I' be the set of row indexes mapped to attributes belonging to those corrupted authorities, i.e., $I' = \{i \in [n] | \phi(\psi^*(i)) \in \mathcal{S}_c\}$. Let $n' = n - |I'|$. Define $A'_{i,j} = 0$ for each pair $(i,j) \in I' \times [n']$, and $A'_{i,j} = A^*_{i,j}$ for other cases. According to Lemma 1 presented in [1], it is known that if the algorithm $\mathcal{C}$ replaces the challenge matrix $\boldsymbol{A}^*$ with the matrix $\boldsymbol{A}'$, then the subsequent simulation is information-theoretically indistinguishable from the real one in the view of the adversary $\mathcal{A}$. Below we demonstrate how the algorithm $\mathcal{C}$ proceeds the simulation.

(1) Public keys for attribute authorities in $\mathcal{S}_h$.

For each attribute authority $\kappa \in \mathcal{S}_h$, the algorithm $\mathcal{C}$ needs to generate the public key $\mathtt{PK}_\kappa = \big\{e(h, g_2)^{\alpha_\kappa}, g_2^{\beta_\kappa}\big\}$. To this end, $\mathcal{C}$ considers the following two cases.

- $\kappa \notin \{\phi(\psi^*(i)) | i \in [n]\}$, i.e., the challenge matrix $\boldsymbol{A}^*$ does not involve any attribute belonging to the attribute authority κ. In this case, $\mathcal{C}$ selects two random integers $\alpha_\kappa, \beta_\kappa \in \mathbb{Z}_p$, and directly assigns $\mathtt{PK}_\kappa = \big\{e(h, g_2)^{\alpha_\kappa} = e(g_1, g_2)^{\alpha_k}, g_2^{\beta_\kappa}\big\}$.
- $\kappa \in \{\phi(\psi^*(i)) | i \in [n]\}$. This case implies that several attributes involved in the challenge matrix $\boldsymbol{A}^*$ are in the attribute domain of the attribute authority κ. Let the corresponding index subset be $I_\kappa = \{\phi(\psi^*(i)) = \kappa | i \in [n]\}$. Then, $\mathcal{C}$ selects two random exponent $\alpha'_\kappa, \beta'_\kappa \in \mathbb{Z}_p$, and implicitly assigns

$$\alpha_\kappa = \alpha'_\kappa + \textstyle\sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1},\ \beta_\kappa = \beta'_\kappa + \sum_{i\in I_\kappa} \sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}.$$

Furthermore, it computes the public key as

$$e(h, g_2)^{\alpha_\kappa} = e(g_1, g_2)^{\alpha'_\kappa} \cdot \textstyle\prod_{i\in I_\kappa} \cdot e(g_1^a, g_2^{b_i a^q})^{A'_{i,1}}, \\ g_2^{\beta_\kappa} = g_2^{\beta'_\kappa} \cdot \textstyle\prod_{i\in I_\kappa} \prod_{j=2}^{n'} (g_2^{b_i a^{q-j+2}})^{A'_{i,j}}.$$

Note that above items are all computable for the algorithm $\mathcal{C}$ with the knowledge of the instance $(\boldsymbol{f}, R)$ of the q-DPBDHE3 problem. Additionally, they also have proper distributions due to the randomness of α'_κ and β'_κ.

(2) Secret keys for tuples in $\mathcal{Q}_{sk} = \{(\mathtt{gid}_i, \tau_i, S_{\mathtt{gid}_i})\}_{i=1}^k$.

For each query tuple $(\mathtt{gid}_i, \tau_i, S_{\mathtt{gid}_i}) \in \mathcal{Q}_{sk}$, the algorithm $\mathcal{C}$ is required to generate a secret key $\mathtt{SK}_{\mathtt{gid}_i, \tau_i, \mathcal{P}_j}$. We denote by $S_{\mathtt{gid}_i} = \cup_{\kappa\in\mathcal{S}_c\cup\mathcal{S}_h} S^\kappa_{\mathtt{gid}_i}$. Since the master secret keys of those corrupted authorities in $\mathcal{S}_c$ are available to the adversary $\mathcal{A}$, we thus just demonstrate how $\mathcal{C}$ produces the secret key component $\mathtt{SK}^\kappa_{\mathtt{gid}_i, \tau_i, \mathcal{P}_j} = \big\{\boldsymbol{sk}_{\mathcal{P}_j}, \{\boldsymbol{sk}_{\eta,u}\}_{\eta\in\mathcal{N}_{\tau_i}, u\in S^\kappa_{\mathtt{gid}_i}}\big\}$ for each attribute set $S^\kappa_{\mathtt{gid}_i}$, where $\kappa \in \mathcal{S}_h$.

During the process of generating secret keys, the algorithm $\mathcal{C}$ has to simulate the two hash functions $F(\cdot): \mathcal{GID} \to \mathbb{G}_1$ and $H(\cdot): \mathcal{U} \to \mathbb{G}_1$ as random oracles. We first illustrate how to do that.

$F(\cdot): \mathcal{GID} \to \mathbb{G}_1$. Given a hash query on global identifier $\mathtt{gid}$, if there exists a record $(\mathtt{gid}, F(\mathtt{gid})) \in \mathcal{L}_F$, then $\mathcal{C}$ directly returns $F(\mathtt{gid})$. Otherwise, if no such a record in $\mathcal{L}_F$ and $\mathtt{gid} \notin \{\mathtt{gid}_i\}_{i=1}^k$, then $\mathcal{C}$ randomly chooses a random integer $\theta \in \mathbb{Z}_p$, and lets $F(\mathtt{gid}) = g_1^\theta$. Furthermore, it returns $F(\mathtt{gid})$ and adds $(\mathtt{gid}, F(\mathtt{gid}))$ into $\mathcal{L}_F$. On the other hand, if $\mathtt{gid} \in \{\mathtt{gid}_i\}_{i=1}^k$, the algorithm $\mathcal{C}$ considers the following two cases.

- $\mathcal{U}_c \cup S_{\mathtt{gid}}$ *satisfies* the challenge matrix $\boldsymbol{A}^*$, where $\mathcal{U}_c$ is the set of attributes belonging to corrupted authorities. In this case, the algorithm $\mathcal{C}$ chooses a random exponent $\theta \in \mathbb{Z}_p$, and also assigns $F(\mathtt{gid}) = g_1^\theta$.
- $\mathcal{U}_c \cup S_{\mathtt{gid}}$ *does not* satisfy the challenge matrix $\boldsymbol{A}^*$. In this case, if the challenge matrix $\boldsymbol{A}^*$ does not involve any attribute in $S_{\mathtt{gid}}$, then the algorithm $\mathcal{C}$ picks a random integer $\theta \in \mathbb{Z}_p$, and assigns

$$F(\mathtt{gid}) = g_1^\theta \cdot g_1^a \cdots g_1^{a^{n'-1}} = g_1^\theta \cdot \textstyle\prod_{j=2}^{n'} g_1^{a^{j-1}}).$$

Otherwise, due to the property of LSSS, there exists a vector $\boldsymbol{d} \in \mathbb{Z}_p^{m\times 1}$ such that $d_1 = -1$ and $\boldsymbol{A}'_i \cdot \boldsymbol{d} = 0$ for all indexes $i \in I_{\mathtt{gid}} = \{i | i \in [n], \psi^*(i) \in \mathcal{U}_c \cup S_{\mathtt{gid}}\}$. In addition, since the rows mapped to attributes belonging to corrupted authorities spans the subspace of dimension $|I'|$, the vector $\boldsymbol{d}$ is therefore

$$
\begin{aligned}
sk_{\eta,u,0} &= h^{\alpha_\kappa} \cdot \big(\prod_{x=0}^{j} h_x\big)^r \cdot F(\mathtt{gid}_i)^{\beta_\kappa} \cdot H(u)^{r_u} \cdot W(\eta)^{\xi_\eta} \\
&= h^{\alpha'_\kappa + \sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1}} \cdot \big(\prod_{x=0}^{j} h_x\big)^r \cdot H(u)^{r_u} \cdot (g_1^{\theta})^{\beta'_\kappa + \sum_{i\in I_\kappa}\sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}} \cdot W(\eta)^{\xi'_\eta} \cdot W(\eta)^{\sum_{x\in I_\kappa} \frac{b_x a^{j'} A'_{x,1}}{b_{\tau^*}[j'] - b_\eta[j']}} \\
&= \underline{h^{\alpha'_\kappa} \cdot g_1^{\theta\beta'_\kappa} \cdot H(u)^{r_u} \cdot W(\eta)^{\xi'_\eta} \cdot \big(\prod_{x=0}^{j} h_x\big)^r \cdot \prod_{i\in I_\kappa}\prod_{j=2}^{n'} \big(g_1^{b_i a^{q-j+2}}\big)^{\theta A'_{i,j}}} \cdot h^{\sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1}} \cdot W(\eta)^{\sum_{x\in I_\kappa} \frac{b_x a^{j'} A'_{x,1}}{b_{\tau^*}[j'] - b_\eta[j']}} \\
&\triangleq \Gamma \cdot \big(g_1^{a^{q+1}}\big)^{\sum_{i\in I_\kappa} b_i A'_{i,1}} \cdot \Big(g_1^{\delta_0} \prod_{i=1}^{l} \big(g_1^{\delta_i} \cdot g_1^{a^{q-i+1}}\big)^{-b_{\tau^*}[i]} \cdot \prod_{i=1}^{|b_\eta|} \big(g_1^{\delta_i} \cdot g_1^{a^{q-i+1}}\big)^{b_\eta[i]}\Big)^{\sum_{x\in I_\kappa} \frac{b_x a^{j'} A'_{x,1}}{b_{\tau^*}[j'] - b_\eta[j']}} \\
&= \Gamma \cdot \big(g_1^{a^{q+1}}\big)^{\sum_{i\in I_\kappa} b_i A'_{i,1}} \cdot \Big(g_1^{\delta_0 + \sum_{i=1}^{|b_\eta|} b_\eta[i]\delta_i - \sum_{i=1}^{l} b_{\tau^*}[i]\delta_i} \prod_{i=1}^{|b_\eta|} (g_1^{a^{q-i+1}})^{b_\eta[i] - b_{\tau^*}[i]} \prod_{i=|b_\eta|+1}^{l} (g_1^{a^{q-i+1}})^{-b_{\tau^*}[i]}\Big)^{\sum_{x\in I_\kappa} \frac{b_x a^{j'} A'_{x,1}}{b_{\tau^*}[j'] - b_\eta[j']}} \\
&= \Gamma \cdot \Big((g_1^{a^{j'}})^{\delta_0 + \sum_{i=1}^{|b_\eta|} b_\eta[i]\delta_i - \sum_{i=1}^{l} b_{\tau^*}[i]\delta_i} \cdot \prod_{i=1, i\neq j'}^{|b_\eta|} (g_1^{a^{q-i+j'+1}})^{b_\eta[i] - b_{\tau^*}[i]} \cdot \prod_{i=|b_\eta|+1}^{l} (g_1^{a^{q-i+j'+1}})^{-b_{\tau^*}[i]}\Big)^{\sum_{x\in I_\kappa} \frac{b_x A'_{x,1}}{b_{\tau^*}[j'] - b_\eta[j']}},
\end{aligned}
$$

$$
sk'_{\eta,u,0} = g_2^{r_u},\ sk_{\eta,u,1} = g_2^{\xi_\eta} = g_2^{\xi'_\eta} \cdot \prod_{i\in I_\kappa} \big(g_2^{b_i a^{j'}}\big)^{A'_{i,1}/(b_{\tau^*}[j'] - b_\eta[j'])},\ sk_{\mathcal{P}_j,0} = g_2^r,\ sk_{\mathcal{P}_j,x} = h_x^r\ \ (x \in [j+1, \ell]),
$$

$$
sk_{\eta,u,x} = v_x^{\xi_\eta} = v_x^{\xi'_\eta} \cdot \prod_{i\in I_\kappa} \big(g_1^{b_i a^{j'}}\big)^{\delta_i A'_{i,1}/(b_{\tau^*}[j'] - b_\eta[j'])} \cdot \prod_{i\in I_\kappa} \big(g_1^{b_i a^{q-x+j'+1}}\big)^{A'_{i,1}/(b_{\tau^*}[j'] - b_\eta[j'])}\ \ (x \in [|b_\eta|+1, l]).
$$

Fig. 1. The computation of $\mathtt{SK}^{\kappa}_{\mathtt{gid},\tau_i,\mathcal{P}_j}$ when $\tau_i > \tau^*$ and $\kappa \in \{\phi(\psi^*(i)) | i \in [n]\}$.

orthogonal to all vectors $\{\boldsymbol{c}_j\}_{j=1}^{|I'|}$, where $\boldsymbol{c}_j \in \mathbb{Z}_p^{1\times m}$ is defined as $c_{j,i} = 1$ for $i = n' + j$ and $c_{j,i} = 0$ for $i \in [n] \setminus \{n' + j\}$. This implies that $d_j = 0$ for $n' + 1 \le j \le n$. Consequently, for $i \in I_{\mathtt{gid}}$, $\boldsymbol{A}'_i \cdot \boldsymbol{d} = 0$ indicates $\sum_{j=1}^{n'} A'_{i,j} \cdot d_j = 0$. Specifically, $\mathcal{C}$ selects a random integer $\theta \in \mathbb{Z}_p$, and assigns

$$F(\mathtt{gid}) = g_1^{\theta} \cdot (g_1^{a})^{d_2} \cdots (g_1^{a^{n'-1}})^{d_{n'}} = g_1^{\theta} \cdot \prod_{j=2}^{n'} (g_1^{a^{j-1}})^{d_j}.$$

Then, $\mathcal{C}$ adds $(\mathtt{gid}, F(\mathtt{gid}))$ into $\mathcal{L}_F$and returns $F(\mathtt{gid})$.

$H(\cdot) : \mathcal{U} \to \mathbb{G}_1$. Given an attribute u, if there already exists a record $(u, H(u)) \in \mathcal{L}_H$, then the algorithm $\mathcal{C}$ directly returns $H(u)$. Otherwise, if $\phi(u) \in \mathcal{S}_c$ or $\phi(u) \notin \{\phi(\psi^*(i)) | i \in [n]\}$, then $\mathcal{C}$ randomly chooses a group element $g' \in \mathbb{G}_1$, and assigns $H(u) = g'$. On the other hand, if $\phi(u) \in \{\phi(\psi^*(i)) | i \in [n]\}$, let $I_u = \{i | i \in [n], \phi(\psi^*(i)) = \phi(u)\} \setminus \{i | i \in [n], \psi^*(i) = u\}$ be the index subset of rows that belong to the authority $\phi(u)$ but u is not in its attribute domain. Then, the algorithm $\mathcal{C}$ selects a random exponent $\zeta \in \mathbb{Z}_p$, and assigns

$$H(u) = g^{\zeta} \cdot \prod_{i\in I_u} \prod_{j=1}^{n'} (g^{b_i a^{q+1-j}})^{A'_{i,j}}.$$

Furthermore, $\mathcal{C}$ adds $(u, H(u))$ into $\mathcal{L}_H$ and returns $H(u)$.

Below we show how the algorithm $\mathcal{C}$ generates a secret key $\mathtt{SK}^{\kappa}_{\mathtt{gid}_i,\tau_i,\mathcal{P}_j}$ for each attribute subset $S^{\kappa}_{\mathtt{gid}_i}$ ($\kappa \in \mathcal{S}_h$). To this end, we also consider the following two cases.

• $\mathcal{U}_c \cup S_{\mathtt{gid}}$ *satisfies* the challenge matrix $\boldsymbol{A}^*$. In this case, due to the restriction placed on the adversary $\mathcal{A}$'s queries, we have that either $\tau_i > \tau^*$ or $t^* \in \mathcal{P}_j$. We further divide this case into the following two subcases.

1) $\tau_i > \tau^*$. In this subcase, there must exist an index $j' \in [l]$ such that $b_\eta[j'] \neq b_{\tau^*}[j']$ for all $\eta \in \mathcal{N}_{\tau_i}$. Then, $\mathcal{C}$ utilizes the masters secret key of the authority κ to generate the secret key $\mathtt{SK}^{\kappa}_{\mathtt{gid},\tau_i,\mathcal{P}_j} = \{\boldsymbol{sk}_{\mathcal{P}_j}, \{\boldsymbol{sk}_{\eta,u}\}_{\eta\in\mathcal{N}_{\tau_i}, u\in S^{\kappa}_{\mathtt{gid}}}\}$, where $\boldsymbol{sk}_{\mathcal{P}_j} = (sk_{\mathcal{P}_j,0}, sk_{\mathcal{P}_j,j+1}, \ldots, sk_{\mathcal{P}_j,\ell})$ and $\boldsymbol{sk}_{\eta,u} = (sk_{\eta,u,0}, sk'_{\eta,u,0}, sk_{\eta,u,1}, sk_{\eta,u,|b_\eta|+1}, \ldots, sk_{\eta,u,l})$. Furthermore, if $\kappa \notin \{\phi(\psi^*(i)) | i \in [n]\}$, then the master key pair $(\alpha_\kappa, \beta_\kappa)$ is available to $\mathcal{C}$. This enables $\mathcal{C}$ to compute

$$
\begin{aligned}
&sk_{\eta,u,0} = h^{\alpha_\kappa} \cdot (\prod_{x=0}^{j} h_x)^r \cdot F(\mathtt{gid}_i)^{\beta_\kappa} \cdot H(u)^{r_u} \cdot W(\eta)^{\xi_\eta}, \\
&sk'_{\eta,u,0} = g_2^{r_u},\ sk_{\eta,u,x} = v_x^{\xi_\eta}\ (x \in [|b_\eta|+1, l]), \\
&sk_{\eta,u,1} = g_2^{\xi_\eta},\ sk_{\mathcal{P}_j,0} = g_2^r,\ sk_{\mathcal{P}_j,x} = h_x^r\ (x \in [j+1, \ell]),
\end{aligned}
$$

where $r \in \mathbb{Z}_p$ is a random integer, r_u and ξ_η are random integers sampled from $\mathbb{Z}_p$ for each $u \in S^{\kappa}_{\mathtt{gid}_i}$ and $\eta \in \mathcal{N}_{\tau_i}$. On the other hand, if $\kappa \in \{\kappa \in \{\phi(\psi^*(i)) | i \in [n]\}$, then the master secret key $(\alpha_\kappa, \beta_\kappa)$ is unknown for $\mathcal{C}$. To this end, for each node $\eta \in \mathcal{N}_{\tau_i}$ and each attribute $u \in S^{\kappa}_{\mathtt{gid}_i}$, $\mathcal{C}$ picks random integers $\xi'_\eta, r_u \in \mathbb{Z}_p$, respectively, as well as another one $r \in \mathbb{Z}_p$. Moreover, $\mathcal{C}$ implicitly assigns $\xi_\eta = \xi'_\eta + \cdot \sum_{x\in I_\kappa} b_x a^{j'} A'_{x,1} / (b_{\tau^*}[j'] - b_\eta[j'])$, and computes secret key components as shown in Figure 1. Note that these components do not involve the unknown item $g_1^{a^{q+1}}$, and thus are all computable for $\mathcal{C}$.

2) $t^* \in \mathcal{P}_j$, i.e., the secret key $\mathtt{SK}^{\kappa}_{\mathtt{gid},\tau_i,\mathcal{P}_j}$ is punctured with the challenge tag t^*. Similarly, if $\kappa \notin \{\phi(\psi^*(i)) | i \in [n]\}$, then the algorithm $\mathcal{C}$ can trivially compute the secret key as in case $\tau > \tau^*$. When $\kappa \in \{\phi(\psi^*(i)) | i \in [n]\}$, for each node $\eta \in \mathcal{N}_{\tau_i}$ and each attribute $u \in S^{\kappa}_{\mathtt{gid}_i}$, the algorithm $\mathcal{C}$ chooses random integers $\xi_\eta, r_u \in \mathbb{Z}_p$, respectively. In addition, it chooses a random exponent $r' \in \mathbb{Z}_p$, and lets $r = r' - a^{t^*} \sum_{i\in I_\kappa} b_i A'_{i,1}$. Then, it computes the secret key components as illustrated in Figure 2.

• $\mathcal{U}_c \cup S_{\mathtt{gid}}$ *does not* satisfy the challenge matrix $\boldsymbol{A}^*$. In this case, recall that there exits a vector $\boldsymbol{d}$ such that $d_1 = 1$ and $\boldsymbol{A}'_i \cdot \boldsymbol{d} = 0$ for all $i \in I_{\mathtt{gid}} = \{i | \psi^*(i) \in \mathcal{U}_c \cup S_{\mathtt{gid}}\} \subseteq [n]$. To generate the secret key $\mathtt{SK}^{\kappa}_{\mathtt{gid}_i,\tau_i,\mathcal{P}_j}$, the algorithm $\mathcal{C}$ first chooses a random integer $r \in \mathbb{Z}_p$, and computes the secret key

$$
\begin{aligned}
sk_{\eta,u,0} &= h^{\alpha_\kappa} \cdot \big(\prod_{x=0}^{j} h_x\big)^{r} \cdot F(\texttt{gid}_i)^{\beta_\kappa} \cdot H(u)^{r_u} \cdot W(\eta)^{\xi_\eta} \\
&= h^{\alpha'_\kappa + \sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1}} \cdot \big(\prod_{x=0}^{j} h_x\big)^{r'} \cdot \big(\prod_{x=0}^{j} h_x\big)^{-a^{t^*}\sum_{i\in I_\kappa} b_i A'_{i,1}} \cdot H(u)^{r_u} \cdot (g_1^{\theta})^{\beta'_\kappa + \sum_{i\in I_\kappa}\sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}} \cdot W(\eta)^{\xi_\eta} \\
&= \underline{h^{\alpha'_\kappa} \cdot g_1^{\theta\beta'_\kappa} \cdot H(u)^{r_u} \cdot W(\eta)^{\xi_\eta} \cdot \big(\prod_{x=0}^{j} h_x\big)^{r'} \cdot \prod_{i\in I_\kappa}\prod_{j=2}^{n'} \big(g_1^{b_i a^{q-j+2}}\big)^{\theta A'_{i,j}}}\, h^{\sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1}} \big(\prod_{x=0}^{j} h_x\big)^{-a^{t^*}\sum_{i\in I_\kappa} b_i A'_{i,1}} \\
&\triangleq \Gamma \cdot \big(g_1^{a^{q+1}}\big)^{\sum_{i\in I_\kappa} b_i A'_{i,1}} \cdot \Big(g_1^{\gamma_0} \prod_{i=1, i\neq t^*}^{\ell} \big(g_1^{\gamma_i} \cdot g_1^{a^{q-i+1}}\big)^{-1} \cdot \prod_{x=1}^{j} \big(g_1^{\gamma_x} \cdot g_1^{a^{q-x+1}}\big)\Big)^{-a^{t^*}\sum_{i\in I_\kappa} b_i A'_{i,1}} \\
&= \Gamma \cdot \big(g_1^{a^{q+1}}\big)^{\sum_{i\in I_\kappa} b_i A'_{i,1}} \cdot \Big(\big(g_1^{a^{t^*}}\big)^{\gamma_0 + \sum_{i=j+1}^{\ell}\gamma_i} \cdot g_1^{a^{q+1}} \cdot \prod_{i=j+1}^{\ell} g_1^{a^{q-i+t^*+1}}\Big)^{-\sum_{i\in I_\kappa} b_i A'_{i,1}} \\
&= \Gamma \cdot \Big(\big(g_1^{a^{t^*}}\big)^{\gamma_0 + \sum_{i=j+1}^{\ell}\gamma_i} \cdot \prod_{i=j+1}^{\ell} g_1^{a^{q-i+t^*+1}}\Big)^{-\sum_{i\in I_\kappa} b_i A'_{i,1}}, \\
sk'_{\eta,u,0} &= g_2^{r_u},\ sk_{\eta,u,1} = g_2^{\xi_\eta},\ sk_{\eta,u,x} = v_x^{\xi_\eta}\ \ (x \in [|b_\eta|+1, l]),\ sk_{\mathcal{P}_j,0} = g_2^{r} = g_2^{r'} \cdot \prod_{i\in I_\kappa} \big(g_2^{b_i a^{t^*}}\big)^{-A'_{i,1}}, \\
sk_{\mathcal{P}_j,x} &= h_x^{r} = h_x^{r'} \cdot \prod_{i\in I_\kappa} \Big(\big(g_1^{a^{t^*}}\big)^{\gamma_x} \cdot g_1^{a^{q-x+t^*+1}}\Big)^{-A'_{i,1}}\ \ (x \in [j+1, \ell]).
\end{aligned}
$$

Fig. 2. The computation of $\texttt{SK}^{\kappa}_{\texttt{gid},\tau_i,\mathcal{P}_j}$ when $t^* \in \mathcal{P}_j$ and $\kappa \in \{\phi(\psi^*(i))|i \in [n]\}$.

$$
\begin{aligned}
sk_{\eta,u,0} &= h^{\alpha_\kappa} \cdot \big(\prod_{x=0}^{j} h_x\big)^{r} \cdot F(\texttt{gid})^{\beta_\kappa} \cdot H(u)^{r_u} \cdot W(\eta)^{\xi_\eta} \\
&= h^{\alpha'_\kappa + \sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1}} \cdot \big(\prod_{x=0}^{j} h_x\big)^{r} \cdot F(\texttt{gid})^{\beta'_\kappa + \sum_{i\in I_\kappa}\sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}} \cdot H(u)^{r'_u} \cdot H(u)^{-\sum_{i\in[n']} a^{i}} \cdot W(\eta)^{\xi_\eta} \\
&= \underline{h^{\alpha'_\kappa} \cdot F(\texttt{gid})^{\beta'_\kappa} \cdot H(u)^{r'_u} \cdot W(\eta)^{\xi_\eta} \cdot \big(\prod_{x=0}^{j} h_x\big)^{r}} \cdot h^{\sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1}} \cdot F(\texttt{gid})^{\sum_{i\in I_\kappa}\sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}} \cdot H(u)^{-\sum_{k\in[n']} a^{k}} \\
&\triangleq \Gamma \cdot \big(g_1^{a^{q+1}}\big)^{\sum_{i\in I_\kappa} b_i A'_{i,1}} \cdot (g_1)^{\sum_{i\in I_\kappa}\sum_{j'=2}^{n'}(\sum_{j=2}^{n'} b_i a^{q-j'+j+1} A'_{i,j'} + \theta b_i a^{q-j'+2} A'_{i,j'})} \cdot (g_1)^{-\zeta\sum_{k=1}^{n'} a^{k} - \sum_{i\in I_\kappa}\sum_{j=1}^{n'}\sum_{k=1}^{n'} b_i a^{q-j+k+1} A'_{i,j}} \\
&= \Gamma \cdot (g_1)^{\theta\sum_{i\in I_\kappa}\sum_{j'=2}^{n'} b_i a^{q-j'+2} A'_{i,j'}} \cdot (g_1)^{-\zeta\sum_{k=1}^{n'} a^{k}} \cdot (g_1)^{-\sum_{i\in I_\kappa}\sum_{k=2}^{n'} b_i a^{q+k} A'_{i,1}} \cdot (g_1)^{-\sum_{i\in I_\kappa}\sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}} \\
&= \Gamma \cdot \prod_{i\in I_\kappa}\prod_{j'=2}^{n'} \big(g_1^{b_i a^{q-j'+2}}\big)^{\theta A'_{i,j'}} \cdot \big(\prod_{k=1}^{n'} g_1^{a^{k}}\big)^{-\zeta} \cdot \prod_{i\in I_\kappa}\prod_{k=2}^{n'} \big(g_1^{b_i a^{q+k}}\big)^{A'_{i,1}} \cdot \prod_{i\in I_\kappa}\prod_{j=2}^{n'} \big(g_1^{b_i a^{q-j+2}}\big)^{A'_{i,j}}, \\
sk'_{\eta,u,0} &= g_2^{r_u} = g_2^{r'_u} \cdot \prod_{k=1}^{n'} \big(g^{a^{k}}\big)^{-1},\ \ sk_{\eta,u,1} = g_2^{\xi_\eta},\ \ sk_{\eta,u,x} = v_x^{\xi_\eta}\ \ (x \in [|b_\eta|+1, l]).
\end{aligned}
$$

Fig. 3. The computation of $\boldsymbol{sk}_{\eta,u}$ when $\kappa \in \{\phi(\psi^*(i))|i \in [n]\}$ and $S^{\kappa}_{\texttt{gid}} \cap \{\psi^*(i)|i \in [n]\} = \emptyset$.

component $\boldsymbol{sk}_{\mathcal{P}_j}$ as follows:

$$sk_{\mathcal{P}_j,0} = g_2^{r},\ sk_{\mathcal{P}_j,x} = h_x^{r}\ (x \in [j+1, \ell]).$$

Then, for each attribute $u \in S^{\kappa}_{\texttt{gid}_i}$, depending on its belonging $\phi(u) = \kappa$, $\mathcal{C}$ calculates the secret key component $\boldsymbol{sk}_{\eta,u}$ with different approaches that are specified as follows.

1) $\kappa \notin \{\phi(\psi^*(i))|i \in [n]\}$. In this case, the master secret key $(\alpha_\kappa, \beta_\kappa)$ of the authority κ is known to the algorithm $\mathcal{C}$. Thus, it can trivially produce $\boldsymbol{sk}_{\eta,u}$ as in previous case.
2) $\kappa \in \{\phi(\psi^*(i))|i \in [n]\}$ and $S^{\kappa}_{\texttt{gid}} \cap \{\psi^*(i)|i \in [n]\} = \emptyset$. In this case, the challenge matrix $\boldsymbol{A}^*$ involves the authority κ but does not involve any attribute in $S^{\kappa}_{\texttt{gid}}$. At this point, also recall that we defined $F(\texttt{gid}) = g_1^{\theta} \cdot \prod_{j=2}^{n'} g_1^{a^{j-1}}$ and $H(u) = g^{\zeta} \cdot \prod_{i\in I_u}\prod_{j=1}^{n'} (g^{b_i a^{q+1-j}})^{A'_{i,j}}$. Particularly, since $\{i|i \in [n], \psi^*(i) = u\} = \emptyset$ in this case, and therefore $I_u = \{i|i \in [n], \phi(\psi^*(i)) = \kappa\} = I_\kappa$. Then, $\mathcal{C}$ chooses random integers $r'_u, \xi_\eta \in \mathbb{Z}_p$, and lets $r_u = r'_u - \sum_{k\in[n']} a^k$. Furthermore, it computes the secrete key components as shown in Figure 3.
3) $\kappa \in \{\phi(\psi^*(i))|i \in [n]\}$ and $S^{\kappa}_{\texttt{gid}} \cap \{\psi^*(i)|i \in [n]\} \neq \emptyset$. In this case, several attributes belonging to $S^{\kappa}_{\texttt{gid}}$ appear in $\boldsymbol{A}^*$, and we defined that $F(\texttt{gid}) = g_1^{\theta} \cdot \prod_{j=2}^{n'} (g_1^{a^{j-1}})^{d_j}$ and $H(u) = g^{\zeta} \cdot \prod_{i\in I_u}\prod_{j=1}^{n'} (g^{b_i a^{q+1-j}})^{A'_{i,j}}$. Here note that I_u is a proper subset of I_κ (i.e., $I_u \subset I_\kappa$). Then, the algorithm $\mathcal{C}$ selects random integers $\xi_\eta, r'_u \in \mathbb{Z}_p$, and implicitly assigns $r_u = r'_u - \sum_{k=1}^{n'} a^k d_k$. Moreover, it computes the secret key component $\boldsymbol{sk}_{\eta,u}$ as illustrated in Figure 4. Note that during the computation of $sk_{\eta,u,0}$, the unknown term $g_1^{a^{q+1}}$ is eliminated since $\sum_{j=1}^{n'} d_j A'_{i,j} = \boldsymbol{A}'_i \cdot \boldsymbol{d} = 0$ holds for all $i \in I_\kappa \setminus I_u = \{i|i \in [n], \psi^*(i) = u\}$, and other terms are all known for the algorithm $\mathcal{C}$.

(3) Challenge ciphertext.

For the given challenge access structure $(\boldsymbol{A}^*_{n\times m}, \psi^*)$[1], tag t^*, time interval τ^* as well as messages M_0 and M_1, the algorithm $\mathcal{C}$ chooses a random bit $b \in \{0,1\}$, and produces the challenge ciphertext $\texttt{CT}^* = \big\{C_0^*, \{C^*_{k,1}, C^*_{k,2}, C^*_{k,3}, C^*_{k,4}, C^*_{k,5}, C^*_{k,6}\}_{k\in[n]}\big\}$ as follows.

First, it computes $C_0^* = M_b \cdot R$, which also implicitly assigns $C_0^* = M_b \cdot e(h, g_2)^{sa^{q+1}}$. Moreover, $\mathcal{C}$ lets $\boldsymbol{u} = (sa^{q+1}, 0, \ldots, 0)$ and $\boldsymbol{w} = (0, sa^{q}, \ldots, sa^{q-n'+2}, 0, \ldots, 0)$.

[1] In fact, we use the modified matrix $\boldsymbol{A}'$ to produce the challenge ciphertext.

$$
\begin{aligned}
sk_{\eta,u,0} &= h^{\alpha_\kappa} \cdot \big(\prod_{x=0}^{j} h_x\big)^r \cdot F(\mathtt{gid})^{\beta_\kappa} \cdot H(u)^{r_u} \cdot W(\eta)^{\xi_\eta} \\
&= h^{\alpha'_\kappa + \sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1}} \cdot \big(\prod_{x=0}^{j} h_x\big)^r \cdot F(\mathtt{gid})^{\beta'_\kappa + \sum_{i\in I_\kappa}\sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}} \cdot H(u)^{r'_u} \cdot H(u)^{-\sum_{k=1}^{n'} a^k d_k} \cdot W(\eta)^{\xi_\eta} \\
&= \underline{h^{\alpha'_\kappa} \cdot F(\mathtt{gid})^{\beta'_\kappa} \cdot H(u)^{r'_u} \cdot W(\eta)^{\xi_\eta} \cdot \big(\prod_{x=0}^{j} h_x\big)^r} \cdot h^{\sum_{i\in I_\kappa} b_i a^{q+1} A'_{i,1}} \cdot F(\mathtt{gid})^{\sum_{i\in I_\kappa}\sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}} \cdot H(u)^{-\sum_{k=1}^{n'} a^k d_k} \\
&\triangleq \Gamma \cdot \big(g_1^{a^{q+1}}\big)^{\sum_{i\in I_\kappa} b_i A'_{i,1}} \cdot (g_1)^{\sum_{i\in I_\kappa}\sum_{j'=2}^{n'}(\sum_{j=2}^{n'} b_i d_j a^{q-j'+j+1} A'_{i,j'} + \theta b_i a^{q-j'+2} A'_{i,j'})} \cdot (g_1)^{-\sum_{k=1}^{n'} d_k a^k(\zeta + \sum_{i\in I_u}\sum_{j=1}^{n'} b_i d_k a^{q-j+1} A'_{i,j})} \\
&= \Gamma \cdot \big(g_1^{a^{q+1}}\big)^{\sum_{i\in I_\kappa\setminus I_u} b_i \sum_{j=1}^{n'} d_j A'_{i,j}} \cdot (g_1)^{\theta \sum_{i\in I_\kappa}\sum_{j'=2}^{n'} b_i a^{q-j'+2} A'_{i,j'}} \cdot (g_1)^{\sum_{i\in I_\kappa\setminus I_u}\sum_{j'=2}^{n'}\sum_{j=2,j\neq j'}^{n'} b_i d_j a^{q-j'+j+1} A'_{i,j'}} \\
&\quad \cdot (g_1)^{-\zeta \sum_{k=1}^{n'} d_k a^k} \cdot (g_1)^{-\sum_{i\in I_u}\sum_{k=2}^{n'} b_i d_k a^{q+k} A'_{i,1}} \cdot (g_1)^{-\sum_{i\in I_u}\sum_{j=2}^{n'} b_i d_1 a^{q-j+2} A'_{i,j}} \\
&= \Gamma \cdot \prod_{i\in I_\kappa}\prod_{j'=2}^{n'} \big(g_1^{b_i a^{q-j'+2}}\big)^{\theta A'_{i,j'}} \cdot \big(\prod_{k=1}^{n'} g_1^{a^k}\big)^{-\zeta d_k} \cdot \prod_{i\in I_\kappa\setminus I_u}\prod_{j'=2}^{n'}\prod_{j=2,j\neq j'}^{n'} \big(g_1^{b_i a^{q-j'+j+1}}\big)^{d_j A'_{i,j'}} \\
&\quad \cdot \prod_{i\in I_u}\prod_{k=2}^{n'} \big(g_1^{b_i a^{q+k}}\big)^{-d_k A'_{i,1}} \cdot \prod_{i\in I_u}\prod_{j=2}^{n'} \big(g_1^{b_i a^{q-j+2}}\big)^{-A'_{i,j}}, \\
sk'_{\eta,u,0} &= g_2^{r_u} = g_2^{r'_u} \cdot \prod_{k=1}^{n'} \big(g^{a^k}\big)^{-d_k}, \quad sk_{\eta,u,1} = g_2^{\xi_\eta}, \quad sk_{\eta,u,x} = v_x^{\xi_\eta} \quad (x \in [|b_\eta|+1, l]).
\end{aligned}
$$

Fig. 4. The computation of $\boldsymbol{sk}_{\eta,u}$ when $\kappa \in \{\phi(\psi^*(i))|i \in [n]\}$ and $S^\kappa_{\mathtt{gid}} \cap \{\psi^*(i)|i \in [n]\} \neq \emptyset$.

Second, for each index $k \in [n]$, depending on the belonging of the attribute $\psi^*(k)$, C calculates corresponding ciphertext components with the following different approaches.

- $\psi^*(k) \in \mathcal{U}_c$, that is, the attribute $\psi^*(k)$ belongs to some corrupted authority. In this case, due to the construction of the matrix $\boldsymbol{A}'$, we have that $\lambda_k = \boldsymbol{A}'_k \boldsymbol{u}^{\mathrm{T}} = 0$ and $\omega_k = \boldsymbol{A}'_k \boldsymbol{w}^{\mathrm{T}} = 0$. Then, C selects a random integer $z_k \in \mathbb{Z}_p$, and uses the public key $\mathtt{PK}_{\rho(k)} = \{e(h,g_2)^{\alpha_{\rho(k)}}, g_2^{\beta_{\rho(k)}}\}$[2] to compute:

$$
\begin{aligned}
&C^*_{k,1} = e(h,g_2)^{\lambda_k} \cdot e(h,g_2)^{\alpha_{\rho(k)} z_k} = \big(e(h,g_2)^{\alpha_{\rho(k)}}\big)^{z_k}, \\
&C^*_{k,2} = g_2^{-z_k}, C^*_{k,3} = g_2^{\beta_{\rho(k)} z_k} \cdot g_2^{\omega_k} = \big(g_2^{\beta_{\rho(k)}}\big)^{z_k}, C^*_{k,4} = W(\tau^*)^{z_k}, \\
&C^*_{k,5} = \big(\prod_{j=0,j\neq t^*}^{\ell} h_j\big)^{z_k}, C^*_{k,6} = H(\psi^*(k))^{z_k}.
\end{aligned}
$$

- $\psi^*(k) \notin \mathcal{U}_c$, i.e., $\psi^*(k)$ does not belong to any corrupted authority. In this case, it holds that $\lambda_k = \boldsymbol{A}'_k \boldsymbol{u}^{\mathrm{T}} = s a^{q+1} \cdot A'_{k,1}$ and $\omega_k = \boldsymbol{A}'_k \boldsymbol{w}^{\mathrm{T}} = \sum_{j=2}^{n'} s a^{q-j+2} A'_{k,j}$. Then, the algorithm C implicitly assigns $z_k = -s/b_k$, and computes:

$$
\begin{aligned}
C^*_{k,1} &= e(h,g_2)^{\lambda_k} \cdot e(h,g_2)^{\alpha_{\rho(k)} z_k} \\
&= e(g_1,g_2)^{s a^{q+1} \cdot A'_{k,1}} \cdot e(g_1,g_2)^{-\left(\alpha'_{\rho(k)} + \sum_{i\in I_{\rho(k)}} b_i a^{q+1} A'_{i,1}\right) s/b_k} \\
&= e(g_1^{s/b_k}, g_2)^{-\alpha'_{\rho(k)}} \cdot \prod_{i\in I_{\rho(k)}\setminus\{k\}} e\big(g_1^{s b_i a^{q+1}/b_k}, g_2\big)^{A'_{i,1}}, \\
C^*_{k,2} &= g_2^{-z_k} = g_2^{s/b_k}, \\
C^*_{k,3} &= g_2^{\beta_{\rho(k)} z_k} \cdot g_2^{\omega_k} \\
&= g_2^{-\left(\beta'_{\rho(k)} + \sum_{i\in I_{\rho(k)}}\sum_{j=2}^{n'} b_i a^{q-j+2} A'_{i,j}\right) s/b_k} \cdot g_2^{\sum_{j=2}^{n'} s a^{q-j+2} A'_{k,j}} \\
&= \big(g_2^{s/b_k}\big)^{-\beta'_{\rho(k)}} \cdot \prod_{i\in I_{\rho(k)\setminus\{k\}}}\prod_{j=2}^{n'} \big(g_2^{s b_i a^{q-j+2}/b_k}\big)^{-A'_{i,j}}, \\
C^*_{k,4} &= W(\tau^*)^{z_k} = \big(v_0 \cdot \prod_{i=1}^{l} (v_i)^{b_{\tau^*}[i]}\big)^{-s/b_k} = \big(g_1^{s/b_k}\big)^{-\delta_0}, \\
C^*_{k,5} &= \big(\prod_{j=0,j\neq t^*}^{\ell} h_j\big)^{z_k} = \big(h_0 \prod_{j=1,j\neq t^*}^{\ell} h_j\big)^{-s/b_k} = \big(g_1^{s/b_k}\big)^{-\gamma_0}, \\
C^*_{k,6} &= H(\psi^*(k))^{z_k} = \big(g_1^{\zeta} \cdot \prod_{i\in I_{\psi^*(k)}}\prod_{j=1}^{n'} (g_1^{b_i a^{q+1-j}})^{A'_{i,j}}\big)^{-s/b_k} \\
&= \big(g_1^{s/b_k}\big)^{-\zeta} \cdot \prod_{i\in I_{\psi^*(k)}}\prod_{j=1}^{n'} \big(g_1^{s b_i a^{q+1-j}/b_k}\big)^{-A'_{i,j}}.
\end{aligned}
$$

In particular, according to the definition of $I_{\psi^*(k)}$, we have that $k \notin I_{\psi^*(k)}$. Therefore, $C^*_{k,6}$ is computable for C.

[2]Recall that $\rho(k) = \phi(\psi^*(k))$.

Guess phase. In this phase, the adversary $\mathcal{A}$ outputs a guess bit b'. If $b' = b$, then the algorithm C outputs 1 that indicates $R = e(g_1,g_2)^{s a^{q+1}}$. Moreover, when $R = e(g_1,g_2)^{s a^{q+1}}$, from the adversary $\mathcal{A}$'s view, the algorithm C simulates the security game perfectly. Thus, we have that

$$
\begin{aligned}
\Pr[C(\boldsymbol{f}, e(g_1,g_2)^{s a^{q+1}}) = 1] &= \Pr\left[b' = b | e(g_1,g_2)^{s a^{q+1}}\right] \\
&= \mathsf{Adv}^{\text{FS-DABPE}}_{\mathcal{A}}(\lambda) + 1/2.
\end{aligned}
$$

On the other hand, if R is a random group element sampled from $\mathbb{G}_T$, then the challenge ciphertext $\mathtt{CT}^*$ dose not reveal any information about the challenge message M_b, which also implies that the adversary $\mathcal{A}$ essentially made a random guess about b. Consequently, we have that

$$
\Pr[C(\boldsymbol{f}, R) = 1] = \Pr[b' = b | R] = 1/2.
$$

Then, we can conclude that

$$
\begin{aligned}
&\mathsf{Adv}^{q\text{-DPBDHE3}}_{C}(\lambda) \\
&= |\Pr[C(\boldsymbol{f}, e(g_1,g_2)^{s a^{q+1}}) = 1] - \Pr[C(\boldsymbol{f}, R) = 1]| \\
&= \mathsf{Adv}^{\text{FS-DABPE}}_{\mathcal{A}}(\lambda).
\end{aligned}
$$

This completes the proof. □

References

[1] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 315–332.