1 Security Proof

Theorem 1. *If the q-BDHE assumption holds, no polynomial time adversary can selectively break the DB-SS-IOV with a challenge matrix of size $l^* \times n^*$, where $n^* < q$.*

Proof. Suppose there is an adversary $\mathcal{A}$ can break the DB-SS-IOV scheme with a non-negligible advantage ε. $\mathcal{A}$ can query any attribute keys and proxy keys that cannot be used to decrypt the challenge ciphertext. Then we can build a simulator $\mathcal{B}$ to break the DB-SS-IOV scheme with the advantage $\varepsilon/2$.

Init. $\mathcal{A}$ selects a user revocation list U^* and a challenge access structure $W^* = (M^*, \rho^*, \mathcal{S}^*)$, where M^* is a $l^* \times n^*$ access matrix, $\mathcal{S}^* = (I_S^*, S^*)$, I_S^* is the attribute names set, $S^* = \left\{\beta^*_{\rho^*(i)}\right\}_{i\in[1,l^*]}$ is the attribute values set, ρ^* maps a row in M^* into an attribute name in I_S^*.

Setup. $\mathcal{B}$ generates the system public parameters by performing the following steps.

- Chooses $\alpha_0 \in Z_p$ at random and set $e(g,g)^{\alpha_1} = e(g,g)^{\alpha_0} e(g^d, g^{d^q})$, then $\alpha_1 = \alpha_0 + d^{q+1}$.
- Selects $a_0 \in Z_p$ and computes g^{a_0}, $\mu = g^d$, $\nu = g^{d^q}$.
- For U^*, let $I_{U^*} = \{i \in \text{path}(uid) \mid uid \in U^*\}$, randomly selects $v_i \in Z_p$ where $i = 0, 1, \ldots, 2N-2$. If $i \in I_{U^*}$, set $y_i = g^{v_i} g^{d^i}$, then $\xi_i = v_i + d^i$; otherwise, set $y_i = g^{v_i} g^{d^q}$, then $\xi_i = v_i + d^q$.

Then $\mathcal{B}$ publishes the system public parameters $GP = \left\langle G_0, G_1, e, g, \mu, \nu, e(g,g)^{\alpha_1}, g^{a_0}, \{y_i\}_{i=0}^{2N-2} \right\rangle$.

Phase 1. $\mathcal{B}$ answers the key queries from $\mathcal{A}$ with the attribute sets $(uid_1, \mathcal{S}_1), (uid_2, \mathcal{S}_2), \ldots, (uid_{Q_1}, \mathcal{S}_{Q_1})$, where $\mathcal{S}_i = (I_S, S), i \in [1, Q_1]$ and $S = \{s_i\}_{i\in I_S}$ is the attribute value set. For each $s_i \in S$, if $s_i = \beta^*_{p^*(i)}$ then set $u_i = s_i + \sum_{n=1}^{n^*} d^n M^*_{k,n}$, where $i \in \{1, 2, \ldots, l^*\}$; otherwise set $u_i = s_i$. There are four cases below, where $\mathcal{S} \models W^*$ represents that the $\mathcal{S}$ meets the access policy W^*, and the $\mathcal{S} \not\models W^*$ represents that the $\mathcal{S}$ does not meet the access policy W^*.

Case 1: If $\mathcal{S} \models W^*$ and $uid \notin U^*$, then terminate.

Case 2: If $\mathcal{S} \models W^*$ and $uid \in U^*$, then $\mathcal{B}$ performs the following steps:

- Randomly chooses $c \in Z_p$. Let $r = -\frac{d^q}{a_0+c} + \frac{d^{q-1}}{a_0+c} \cdot \frac{M_{i,1}}{M^*_{i,2}}$, then computes $K_1 = c, L_1 = L_0^{a_k} = g^{a_k r}$,

$$
\begin{aligned}
&K_0 = g^{\frac{\alpha'}{a_0+c}} \left(g^{\frac{d^q}{a_0+c}}\right)^{\frac{M^*_{i,1}}{M^*_{i,2}}} = g^{\frac{\alpha_1}{a_0+c}} \mu^r, \\
&L_0 = [(g^{d^q})^{\frac{1}{a_0+c}}]^{-1} [(g^{d^{g-1}})^{\frac{1}{a_0+c}}]^{\frac{M^*_{i:1}}{M^*_{i,2}}} = g^r, \\
&K_i = [(g^{d^q})^{\frac{s_i}{a_0+c}}]^{-1} \cdot [(g^{d^{q-1}})^{\frac{s_i M^*_{i,1}}{a_0+c M^*_{i,2}}}]^{\frac{M^*_{i,1}}{M^*_{i,2}}} \cdot g^{u_i \cdot r} v^{-(a_0+c)r} \\
&[(\prod_{k=1}^{n^*} g^{-d^{q+k} \cdot M^*_{i,k}}) \cdot (\prod_{k=1}^{n^*} g^{d^{q+k-1} \cdot M^*_{i,k}})^{\frac{M^*_{i,1}}{M^*_{i,2}}}]^{\frac{1}{a_0+c}} \cdot g^{d^{2q}} \\
&= g^{u_i \cdot r} \nu^{-(a_0+c)r}
\end{aligned} \tag{1}
$$

- Suppose $path\,(uid) = \{i_0, \ldots, i_d\}$, where $i_0 = root$ and i_d is the leaf node value in the binary tree that is related to the user uid. Since $uid \in U^*$, then $i_d \in I_{U^*}, \xi_{i_d} = v_{i_d} + d^{i_d}$ is concluded. $\mathcal{B}$ calculates

$$
K_u = (g^{d^q})^{-1} \cdot (g^{d^{q-1}})^{\frac{M^*_{i,1}}{M^*_{i,2}}}]^{\frac{1}{(v_{id}+d^{id}) \cdot (a_0+c)}} = g^{r/\xi_{id}}. \tag{2}
$$

Case 3: If $\mathcal{S} \not\models W^*$ and $uid \in U^*$, $\mathcal{B}$ performs as follows:

- According to the definition of LSSS, randomly choose a vector $\vec{\omega} = (\omega_1, \omega_2, \ldots, \omega_{n^*})^{\mathsf{T}} \in Z_p^{n^*}$, where $\omega_1 = -1$ and $M_i^* \cdot \vec{\omega} = 0$ for $i \in [2, l^*]$.
- Select $c \in Z_p$ and set $K_1 = c$,
- Randomly chooses $h \in Z_p$ and implicitly define $r = \frac{1}{a_0+c}\left(h + \omega_1 d^q + \omega_2 d^{q-1} \cdots + \omega_{n^*} d^{q-n^*+1}\right)$,
- Calculate K_0, L_0 and L_1 as follows:

$$
\begin{aligned}
&L_0 = g^{\frac{h}{a_0+c}} \prod_{i=1}^{n^*} (g^{\omega_i d^{q+1-i}})^{\frac{1}{a_0+c}} = g^r, \\
&L_1 = g^{\frac{a_0 h}{a_0+c}} \prod_{i=1}^{n^*} (g^{\omega_i d^{q+1-i}})^{\frac{a_0}{a_0+c}} = g^{a_0 r}, \\
&K_0 = (g^{\alpha_1' + dh} \prod_{i=2}^{n^*} g^{\omega_i d^{q+2-i}})^{\frac{1}{a_0+c}} = g^{\frac{\alpha_1}{a_0+c}} \mu^r,
\end{aligned} \tag{3}
$$

- For $\forall \tau \in I_S$, if there exists i such that $\rho^*(i) = \tau$ and $s_\tau = \beta^*_{\rho^*(i)}$, then $\mathcal{B}$ computes

$$
\begin{aligned}
&K_\tau = L_0^{s_\tau} [\prod_{j=1}^{n^*} (g^{t \cdot d^j} \cdot \prod_{k=1}^{n^*} g^{\omega_k d^{q+1+j-k}})^{M^*_{i,j}}]^{\frac{1}{a_0+c}} \\
&\cdot (g^{-t \cdot d^q} \prod_{i=1}^{n^*} g^{-\omega_i d^{2q+1-i}}).
\end{aligned} \tag{4}
$$

Otherwise, the $K_\tau = L_0^{s_\tau} (g^{t \cdot d^q} \prod_{i=1}^{n^*} g^{\omega_i} d^{2q+1-i})^{-1}$.

- Suppose $path\,(uid) = \{i_0, \ldots, i_d\}$, where $i_0 = root$ and i_d is the leaf node value in the binary tree that is related to the user uid. Since $uid \in U^*$, then $i_d \in I_{U^*}, sk_{i_d} = v_{i_d} + d^{i_d}$ is obtained. $\mathcal{B}$ computes $K_u = (g^t \prod_{i=1}^{n^*} g^{\omega_i d^{q+1-i}})^{1/\left(v_{i_d} + d^{i_d}\right) \cdot (a_0+c)} = g^{r/\xi_{i_d}}$.

Case 4: If $\mathcal{S} \not\models W^*$ and $uid \notin U^*$, the calculation process of L_0, L_1, K_0 and K_τ is the same as case 3. Since $uid \notin U^*$, then $i_d \notin I_{U^*}$, $sk_{i_d} = v_{i_d} + d^q$. Next, $\mathcal{B}$ calculates $K_u = (g^t \prod_{i=1}^{n^*} \cdot g^{\omega_i d^{g+1-i}})^{\frac{1}{(v_{i_d}+dq) \cdot (a_0+c)}} = g^{r/\xi_{i_d}}$.

Challenge. $\mathcal{A}$ sends two equal-length keys k_0, k_1 to $\mathcal{B}$. $\mathcal{B}$ performs the following processes:

- Tosses a fair coin $b \in \{0, 1\}$ and performs the $Online.Encrypt$ algorithm. Computes $C_2 = k_b \cdot e\,(g,g)^{\alpha_1 s}$.
- $\mathcal{B}$ performs the $Offline.Encrypt$ algorithm and computes $C_0 = g^s$, and $C_1 = g^{a_0 s}$.
- $\mathcal{B}$ randomly chooses $r_2, \ldots, r_n \in Z_p^*$ and sets $\vec{v} = (s, sd + r_2, sd^2 + r_3, \ldots, sd^{n^*-1} + r_{n^*})^{\mathsf{T}} \in Z^*_{p^*}$, then calculate

$$
\begin{aligned}
&C_{i,1} = \prod_{j=2}^{n^*} (g^{dr_j})^{M^*_{i,j}} \prod_{j=1}^{n^*} (g^{sd^j})^{M^*_{i,j}} g^{-a_0 d^{q+i}} (g^{sd^{j-1}})^{M^*_{i,j}}, \\
&C_{i,2} = (g^{t_{\rho^*(i)}})^{-a_0 d^i} \prod_{j=2}^{n^*} (g^{d^j M^*_{i,j}})^{-a_0 d^i} \cdot g^{r_j M^*_{i,j} + sd^{j-1} M^*_{i,j}} \\
&C_{i,3} = g^{-a_0 d^i}
\end{aligned} \tag{5}
$$

• For $\forall j \in \operatorname{cover}(U^*)$, since $\xi_j = v_j + d^q$ and $y_j = g^{v_j+d^q}$, then $\mathcal{B}$ sets $T_j = (g^s)^{v_j+d^q} = y_j^s$.

Finally, $\mathcal{B}$ sends the ciphertext $CT =< C_0, C_1, C_2, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[1,l^*]}, \{T_j\}_{j\in\operatorname{cover}(U^*)} >$ to $\mathcal{A}$.

Phase 2. This stage is the same as stage 1.

Guess. $\mathcal{A}$ eventually output a guess b' of b.

• If $b = b', \mathcal{B}$ outputs a guess $\mu' = 0$ of μ. If $\mu = 0$ then $Z = e(g,g)^{\alpha^{q+1}s}$. $\mathcal{A}$ can obtain a valid ciphertext. Suppose the advantage of $\mathcal{A}$ is $\varepsilon = \Pr[b = b' \mid \mu = 0] - \frac{1}{2}$, then $\Pr[b = b' \mid \mu = 0] = \Pr[\mu = \mu' \mid \mu = 0]$. Thus, the advantage of $\mathcal{B}$ in winning the game is $\Pr[\mu = \mu' \mid \mu = 0] = \varepsilon + \frac{1}{2}$.

• If $b \neq b'$, $\mathcal{B}$ outputs a guess $\mu' = 1$ of μ. If $\mu = 1$ then Z is a randomly chosen number from G_1. In this case, $\mathcal{A}$ cannot get any information about b. In such case, The advantage of $\mathcal{A}$ is $\Pr[b \neq b' \mid \mu = 1] = \frac{1}{2}$, then we can obtain $\Pr[b \neq b' \mid \mu = 1] = \Pr[\mu = \mu' \mid \mu = 1]$. Therefore, the advantage of B in winning the game is $\Pr[\mu = \mu' \mid \mu = 1] = \frac{1}{2}$.

Finally, the advantage of $\mathcal{B}$ in solving q-BDHE hardness assumption is

$$\begin{aligned}\Pr[\mu = \mu'] =& \Pr[\mu = \mu' \mid \mu = 0] \cdot \Pr[\mu = 0] \\ &+ \Pr[\mu = \mu' \mid \mu = 1] \cdot \Pr[\mu = 1] - \frac{1}{2} \\ =&(\varepsilon + \frac{1}{2}) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{1}{2}\varepsilon\end{aligned}$$

□