Hierarchical identity-based encryption

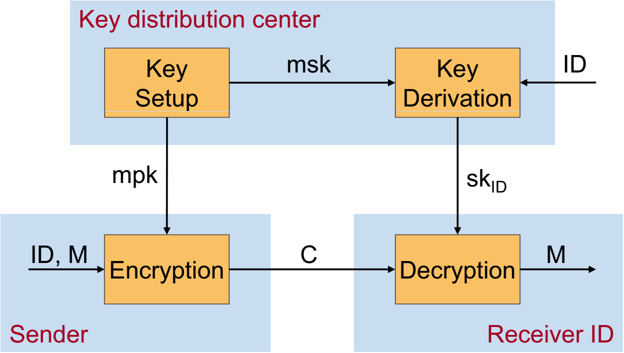
Michel Abdalla

ENS & CNRS

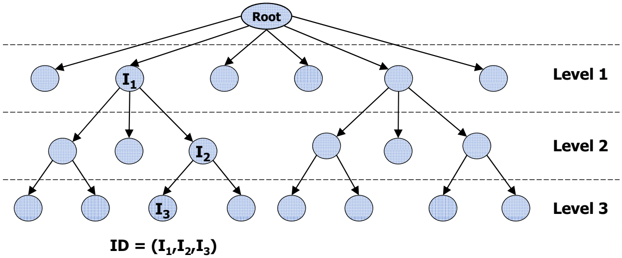
MPRI - Course 2-12-1

Identity-based encryption

Goal: Allow senders to encrypt messages based on the receiver's identity.

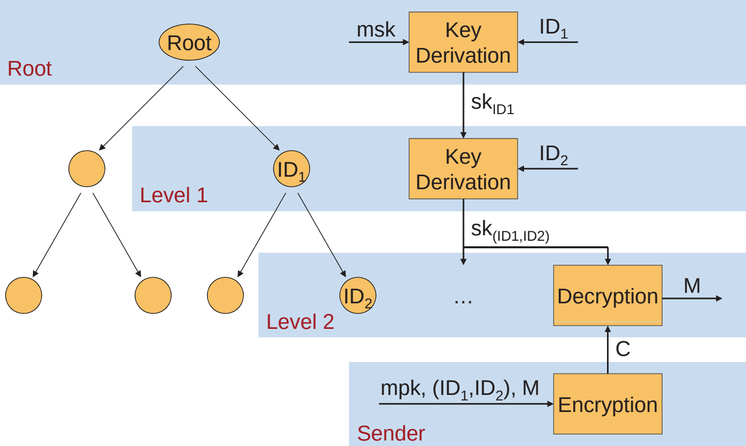


Hierarchical identity-based encryption (HIBE)



- Identities are vectors of the form $(id_1, \ldots, id_L)$, where L is the HIBE depth.
- **Hierarchical key derivation**
  Users with (id_1, id_2) can derive keys for any user whose identity is of the form $(id_1, id_1, *, \ldots, *)$

HIBE key derivation



Outline

Hierarchical Identity-based encryption (HIBE)

– Identity at level $1 \leq \ell \leq L$ is a vector $id = (id_1, \ldots, id_\ell) \in \mathrm{ID}^\ell$.
– Root identity is represented by ε.

An HIBE scheme is defined by four algorithms:

- Setup$(1^k, L)$:
  Outputs a master public key *mpk* for a HIBE of depth L along with master secret key *msk*.
- KeyDer$(sk_{(id_1,\ldots,id_\ell)}, id_{\ell+1})$:
  Uses the secret key *sk* for identity $id = (id_1, \ldots, id_\ell)$ to compute a secret key sk_{id} for the user with identity *id*.
- Enc(mpk, id, m):
  Generates a ciphertext C for identity $id = (id_1, \ldots, id_\ell)$ and message m using master public key *mpk*.
- Dec(C, sk_{id}):
  Allows the user in possession of sk_{id} for identity $id = (id_1, \ldots, id_\ell)$ to decrypt the ciphertext C and get back a message m.

HIBE security notions

Just as in the IBE case, we can consider different attacks (*adaptive-identity* vs. *selective-identity*) and goals (*indistinguishability* and *anonymity*) for HIBE schemes.

- **Indistinguishability**
  The adversary's goal is to distinguish $\mathsf{Enc}(mpk, id^*, m_0^*)$ from $\mathsf{Enc}(mpk, id^*, m_1^*)$ for values id^*, m_0^*,m_1^* of its choice.
- **Anonymity**
  The adversary's goal is to distinguish $\mathsf{Enc}(mpk, id_0^*, m^*)$ from $\mathsf{Enc}(mpk, id_1^*, m^*)$ for values id_0^*, id_1^*, m^* of its choice.
- **Adaptive-identity chosen-plaintext attacks**
  In this model, the adversary is allowed to choose the challenge identity values at the time that it asks the challenge query.
- **Selective-identity chosen-plaintext attacks**
  In this model, the adversary has to choose the challenge identity values before seeing the public key.

IND-HID-CPA: Indistinguishability under chosen-plaintext attacks

- Let $\mathsf{HIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ be a hierarchical identity-based encryption scheme of depth L.
- Let $\mathcal{A}$ be an adversary against the IND-HID-CPA security of HIBE.

Game $\mathbf{Exp}^{\text{ind-cpa-}\beta}_{\mathsf{HIBE},L,\mathcal{A}}(k)$

proc Initialize(k, L)

$(mpk, msk) \xleftarrow{R} \mathsf{Setup}(1^k, L)$
Return mpk

proc KeyDer(id)

$sk_{id} \xleftarrow{R} \mathsf{KeyDer}(msk, id)$
Return sk_{id}

proc LR(id^*, m_0^*, m_1^*)

$C^* \xleftarrow{R} \mathsf{Enc}(mpk, id^*, m_\beta^*)$
Return C^*

proc Finalize(β')

Return β'

The advantage of $\mathcal{A}$ against the IND-HID-CPA security of HIBE is defined as

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{ind-cpa-1}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-cpa-0}}_{\mathsf{HIBE},L,\mathcal{A}}(k)) = 1\right]$$

IND-HID-CPA: An alternative definition

- Let $\mathsf{HIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ be a hierarchical identity-based encryption scheme of depth L.
- Let $\mathcal{A}$ be an adversary against the IND-HID-CPA security of HIBE.

Game $\mathbf{Exp}^{\text{ind-cpa}}_{\mathsf{HIBE},L,\mathcal{A}}(k)$

proc Initialize(k, L)

$\beta \xleftarrow{R} \{0,1\}$

$(mpk, msk) \xleftarrow{R} \mathsf{Setup}(1^k, L)$

Return mpk

proc KeyDer(id)

$sk_{id} \xleftarrow{R} \mathsf{KeyDer}(msk, id)$

Return sk_{id}

proc LR(id^*, m_0^*, m_1^*)

$C^* \xleftarrow{R} \mathsf{Enc}(mpk, id^*, m_\beta^*)$

Return C^*

proc Finalize(β')

Return $(\beta' = \beta)$

The advantage of $\mathcal{A}$ against the IND-HID-CPA security of HIBE is defined as

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = 2 \cdot \Pr\left[\mathbf{Exp}^{\text{ind-cpa}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = \mathsf{true}\right] - 1$$

IND-sHID-CPA: Indistinguishability under *selective-identity* chosen-plaintext attacks

- Let HIBE = (Setup, KeyDer, Enc, Dec) be a hierarchical identity-based encryption scheme of depth L.
- Let $\mathcal{A}$ be an adversary against the IND-sHID-CPA security of HIBE.

Game $\mathbf{Exp}^{\text{s-ind-cpa-}\beta}_{\mathsf{HIBE},L,\mathcal{A}}(k)$

proc Initialize(k, L, id^*)	**proc LR**(m_0^*, m_1^*)
$(mpk, msk) \stackrel{R}{\leftarrow} \mathsf{Setup}(1^k, L)$	$C^* \stackrel{R}{\leftarrow} \mathsf{Enc}(mpk, id^*, m_\beta^*)$
Return mpk	Return C^*
proc KeyDer(id)	**proc Finalize**(β')
$sk_{id} \stackrel{R}{\leftarrow} \mathsf{KeyDer}(msk, id)$	Return β'
Return sk_{id}	

The advantage of $\mathcal{A}$ against the IND-sHID-CPA security of HIBE is defined as

$$\mathbf{Adv}^{\text{s-ind-cpa}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{s-ind-cpa-1}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{s-ind-cpa-0}}_{\mathsf{HIBE},L,\mathcal{A}}(k)) = 1\right]$$

IND-sHID-CPA: An alternative definition

- Let $\mathsf{HIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ be a hierarchical identity-based encryption scheme of depth L.
- Let $\mathcal{A}$ be an adversary against the IND-sHID-CPA security of HIBE.

Game $\mathbf{Exp}_{\mathsf{HIBE}}^{\mathrm{s\text{-}ind\text{-}cpa}[L]}$

proc Initialize(k, L, id^*)

$\beta \xleftarrow{R} \{0,1\}$

$(mpk, msk) \xleftarrow{R} \mathsf{Setup}(1^k)$

Return mpk

proc KeyDer(id)

$sk_{id} \xleftarrow{R} \mathsf{KeyDer}(msk, id)$

Return sk_{id}

proc LR(m_0^*, m_1^*)

$C^* \xleftarrow{R} \mathsf{Enc}(mpk, id^*, m_\beta^*)$

Return C^*

proc Finalize(β')

Return $(\beta' = \beta)$

The advantage of $\mathcal{A}$ against the IND-sHID-CPA security of HIBE is defined as

$$\mathbf{Adv}_{\mathsf{HIBE},L,\mathcal{A}}^{\mathrm{s\text{-}ind\text{-}cpa}}(k) = 2 \cdot \Pr\left[\mathbf{Exp}_{\mathsf{HIBE}}^{\mathrm{s\text{-}ind\text{-}cpa}[L]} = \mathsf{true}\right] - 1$$

ANO-HID-CPA: Anonymity under chosen-plaintext attacks

- Let HIBE = (Setup, KeyDer, Enc, Dec) be a hierarchical identity-based encryption scheme of depth L.
- Let $\mathcal{A}$ be an adversary against the ANO-HID-CPA security of HIBE.

Game $\mathbf{Exp}^{\text{ano-cpa-}\beta}_{\mathsf{HIBE},L,\mathcal{A}}(k)$

proc Initialize(k, L)	**proc LR**(id_0^*, id_1^*, m^*)
$(mpk, msk) \xleftarrow{R} \mathsf{Setup}(1^k)$	$C^* \xleftarrow{R} \mathsf{Enc}(mpk, id_\beta^*, m^*)$
Return mpk	Return C^*
proc KeyDer(id)	**proc Finalize**(β')
$sk_{id} \xleftarrow{R} \mathsf{KeyDer}(msk, id)$	Return β'
Return sk_{id}	

The advantage of $\mathcal{A}$ against the ANO-HID-CPA security of HIBE is defined as

$$\mathbf{Adv}^{\text{ano-cpa}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{ano-cpa-1}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ano-cpa-0}}_{\mathsf{HIBE},L,\mathcal{A}}(k)) = 1\right]$$

ANO-sHID-CPA: Anonymity under *selective-identity* chosen-plaintext attacks

- Let HIBE $=$ (Setup, KeyDer, Enc, Dec) be a hierarchical identity-based encryption scheme of depth L.
- Let $\mathcal{A}$ be an adversary against the ANO-sHID-CPA security of HIBE.

Game $\mathbf{Exp}^{\text{s-ano-cpa-}\beta}_{\mathsf{HIBE},L,\mathcal{A}}(k)$

proc Initialize(k, L, id_0^*, id_1^*)	**proc LR**(m^*)
$(mpk, msk) \xleftarrow{R} \mathsf{Setup}(1^k, L)$	$C^* \xleftarrow{R} \mathsf{Enc}(mpk, id_\beta^*, m^*)$
Return mpk	Return C^*
proc KeyDer(id)	**proc Finalize**(β')
$sk_{id} \xleftarrow{R} \mathsf{KeyDer}(msk, id)$	Return β'
Return sk_{id}	

The advantage of $\mathcal{A}$ against the ANO-sHID-CPA security of HIBE is defined as

$$\mathbf{Adv}^{\text{s-ano-cpa}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{s-ano-cpa-1}}_{\mathsf{HIBE},L,\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{s-ano-cpa-0}}_{\mathsf{HIBE},L,\mathcal{A}}(k)) = 1\right]$$

Outline

Boneh-Boyen HIBE scheme (BB)

Setup$(1^k, L)$:
$(\mathbb{G}, \mathbb{G}_T, p, \hat{e}) \xleftarrow{R} \mathcal{G}(1^k)$
$g \xleftarrow{R} \mathbb{G}$
$a \xleftarrow{R} \mathbb{Z}_p$; $A \leftarrow g^a$
$b \xleftarrow{R} \mathbb{Z}_p$; $B \leftarrow g^b$
for $i = 1, \ldots, L$; $b = 0, 1$ do
$h_{i,b} \xleftarrow{R} \mathbb{Z}_p$; $H_{i,b} \leftarrow g^{h_{i,b}}$
$mpk \leftarrow (g, A, B, H_{1,0}, \ldots, H_{L,1}, \mathbb{G}, \mathbb{G}_T, p, \hat{e})$
$msk \leftarrow g^{ab}$
return (mpk, msk)

Enc(mpk, id, m):
parse id as $(id_1, \ldots, id_\ell)$
$t \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow g^t$
for $i = 1, \ldots, \ell$ do
$C_{2,i} \leftarrow \left(H_{i,0}^{id_i} H_{i,1}\right)^t$
$K \leftarrow \hat{e}(A, B)^t$
$C_3 \leftarrow m \cdot K$
return $(C_1, (C_{2,1}, \ldots, C_{2,\ell}), C_3)$

KeyDer$(sk_{(id_1,\ldots,id_\ell)}, id_{\ell+1})$:
parse $sk_{(id_1,\ldots,id_\ell)}$ as $(sk_0, \ldots, sk_\ell)$
$r_{\ell+1} \xleftarrow{R} \mathbb{Z}_p$
$sk'_0 \leftarrow sk_0 \cdot \left(H_{i,0}^{id_{\ell+1}} H_{i,1}\right)^{r_{\ell+1}}$
$sk'_{\ell+1} \leftarrow g^{r_{\ell+1}}$
return $(sk'_0, sk_1, \ldots, sk_\ell, sk'_{\ell+1})$

Dec(sk, C):
parse $sk_{(id_1,\ldots,id_\ell)}$ as $(sk_0, \ldots, sk_\ell)$
parse C as $(C_1, C_{2,1}, \ldots, C_{2,\ell}, C_3)$
$K' \leftarrow \hat{e}(sk_0, C_1) / \prod_{i=1}^{\ell} \hat{e}(sk_i, C_{2,i})$
$m' \leftarrow C_3 / K'$
return m'

Additional comments about the BB HIBE scheme

- The secret key $sk_{(id_1,\ldots,id_\ell)} = (sk_0, \ldots, sk_\ell)$ for identity $(id_1, \ldots, id_\ell)$ has the form:
  - $sk_0 = g^{ab} \prod_{i=1}^{\ell} (H_{i,0}^{id_i} H_{i,1})^{r_i}$
  - $sk_i = g^{r_i}$ for $i = 1, \ldots, \ell$
- The secret key outputted by KeyDer can be re-randomized via

  Randomize($sk_{(id_1,\ldots,id_\ell)}$):
  parse $sk_{(id_1,\ldots,id_\ell)}$ as $(sk_0, \ldots, sk_\ell)$
  for $i = 1, \ldots, \ell$ do
  $r_i \xleftarrow{R} \mathbb{Z}_p$
  $sk'_i \leftarrow sk_i \cdot g^{r_i}$
  $sk'_0 \leftarrow sk_0 \cdot \prod_{i=1}^{\ell} (H_{i,0}^{id_i} H_{i,1})^{r_i}$
  return $(sk'_0, sk'_1, \ldots, sk'_\ell)$

Correctness of BB HIBE scheme

For a valid ciphertext, we have:

$$
\begin{aligned}
K' &= \hat{e}(sk_0, C_1) / \prod_{i=1}^{\ell} \hat{e}(sk_i, C_{2,i}) \\
&= \hat{e}(g^{ab} \prod_{i=1}^{\ell} (H_{i,0}^{id_i} H_{i,1})^{r_i}, g^t) / \prod_{i=1}^{\ell} \hat{e}(g^{r_i}, (H_{i,0}^{id_i} H_{i,1})^t) \\
&= \hat{e}(g^{ab}, g^t) \cdot \prod_{i=1}^{\ell} \hat{e}((H_{i,0}^{id_i} H_{i,1})^{r_i}, g^t) / \prod_{i=1}^{\ell} \hat{e}(g^{r_i}, (H_{i,0}^{id_i} H_{i,1})^t) \\
&= \hat{e}(g^a, g^b)^t \\
&= \hat{e}(A, B)^t \\
&= K
\end{aligned}
$$

Boneh-Boyen-Goh HIBE scheme (BBG-HIBE)

Setup:

$g_1, g_2 \xleftarrow{R} \mathbb{G}$; $\alpha \xleftarrow{R} \mathbb{Z}_p$
$h_1 \leftarrow g_1^\alpha$; $h_2 \leftarrow g_2^\alpha$
$u_i \xleftarrow{R} \mathbb{G}$ for $i = 1, \ldots, L$
$mpk \leftarrow (g_1, g_2, h_1, u_0, \ldots, u_L)$
$sk_0 \leftarrow h_2$
For $i = 1, \ldots, L+1$ do
  $sk_i \leftarrow 1$
$msk \leftarrow (sk_0, sk_1, \ldots, sk_L, sk_{L+1})$
Return (mpk, msk)

KeyDer($sk_{(id_1,\ldots,id_\ell)}, id_{\ell+1}$):

Parse $sk_{(id_1,\ldots,id_\ell)}$ as $(sk_0, sk_{\ell+1}, \ldots, sk_L, sk_{L+1})$
$r_{\ell+1} \xleftarrow{R} \mathbb{Z}_p$
$sk'_0 \leftarrow sk_0 \cdot sk_{\ell+1}^{id_{\ell+1}} \cdot \left(u_0 \prod_{i=1}^{\ell} u_i^{id_i}\right)^{r_{\ell+1}}$
For $i = \ell+2, \ldots, L$ do
  $sk'_i \leftarrow sk_i \cdot u_i^{r_{\ell+1}}$
$sk'_{L+1} \leftarrow sk_{L+1} \cdot g_1^{r_{\ell+1}}$
Return $(sk'_0, sk'_{\ell+2}, \ldots, sk'_L, sk'_{L+1})$

Enc(mpk, id, m):

Parse id as $(id_1, \ldots, id_\ell)$
$r \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow g_1^r$
$C_2 \leftarrow \left(u_0 \prod_{i=1}^{\ell} u_i^{id_i}\right)^r$
$C_3 \leftarrow m \cdot \hat{e}(h_1, g_2)^r$
Return (C_1, C_2, C_3)

Dec($sk_{(id_1,\ldots,id_\ell)}, C$):

Parse $sk_{(id_1,\ldots,id_\ell)}$ as $(sk_0, sk_{\ell+1}, \ldots, sk_{L+1})$
Parse C as (C_1, C_2, C_3)
$m' \leftarrow C_3 \cdot \frac{\hat{e}(C_2, sk_{L+1})}{\hat{e}(C_1, sk_0)}$
Return m'

Waters HIBE scheme (Wa-HIBE)

Setup:

$g_1, g_2 \xleftarrow{R} \mathbb{G}$; $\alpha \xleftarrow{R} \mathbb{Z}_p$

$h_1 \leftarrow g_1^{\alpha}$; $h_2 \leftarrow g_2^{\alpha}$

$u_{i,j} \xleftarrow{R} \mathbb{G}$ for $i = 1, \ldots, L; j = 0 \ldots n$

$mpk \leftarrow (g_1, g_2, h_1, u_{1,0}, \ldots, u_{L,n})$

$msk \leftarrow h_2$

Return (mpk, msk)

KeyDer($sk_{(id_1,\ldots,id_\ell)}, id_{\ell+1}$):

Parse $sk_{(id_1,\ldots,id_\ell)}$ as $(sk_0, \ldots, sk_\ell)$

$r_{\ell+1} \xleftarrow{R} \mathbb{Z}_p$

$sk'_0 \leftarrow sk_0 \cdot F_{\ell+1}(id_{\ell+1})^{r_{\ell+1}}$

$sk'_{\ell+1} \leftarrow g_1^{r_{\ell+1}}$

Return $(sk'_0, sk_1, \ldots, sk_\ell, sk'_{\ell+1})$

Enc(mpk, id, m):

Parse id as $(id_1, \ldots, id_\ell)$

$r \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow g_1^r$

For $i = 1, \ldots, \ell$ do

$C_{2,i} \leftarrow F_i(id_i)^r$

$C_3 \leftarrow m \cdot \hat{e}(h_1, g_2)^r$

Return $(C_1, C_{2,1}, \ldots, C_{2,\ell}, C_3)$

Dec($sk_{(id_1,\ldots,id_\ell)}, C$):

Parse $sk_{(id_1,\ldots,id_\ell)}$ as $(sk_0, \ldots, sk_\ell)$

Parse C as $(C_1, C_{2,1}, \ldots, C_{2,\ell}, C_3)$

$m' \leftarrow C_3 \cdot \frac{\prod_{i=1}^{\ell} \hat{e}(sk_i, C_{2,i})}{\hat{e}(C_1, sk_0)}$

Return m'

Outline

BDDH security of BB HIBE scheme

Theorem

Let

- BB *refer to the Boneh-Boyen HIBE scheme described above,*
- $\mathcal{G}$ *be a pairing parameter generator, and*
- $\mathcal{A}$ *be an adversary against* IND-sHID-CPA *security of* BB, *making at most a single query to the* **LR** *procedure.*

Then, there exists an adversary $\mathcal{B}$ *against the BDDH problem relative to* $\mathcal{G}$, *whose running time is that of* $\mathcal{A}$ *and such that*

$$\mathbf{Adv}^{\text{s-ind-cpa}}_{\mathsf{BB},L,\mathcal{A}}(k) \leq 2 \cdot \mathbf{Adv}^{\text{bddh}}_{\mathcal{G},k}(\mathcal{B}).$$

Security proof of BB scheme

– Proof will define a sequence of five games $(\mathsf{G}_0, \ldots, \mathsf{G}_4)$.
– For simplicity, we omit the pairing parameter generation in **Initialize**.
– We assume that id^* has length L.
– j denotes the smallest index such that $id_j \neq id_j^*$ in **LR** procedure.

- G_0: This game is the real attack game against BB.
- G_1: We change the computation of $H_{i,b}$ so that $H_{i,0}^{id_i^*} H_{i,1} = g^{\alpha_i}$ for a random α_i.
- G_2: We change the simulation of the key derivation procedure **KeyDer** so that the game answers these queries without the knowledge of the master secret key.
- G_3: We change the simulation of the **LR** procedure so that $C_{2,i}^* = C_1^{*\alpha_i}$. That is, we don't need to know t to compute it.
- G_4: We change the simulation of the **LR** procedure so that K is chosen uniformly at random.

Game G_0

Game $\mathrm{G}_0^{\mathcal{A}}$

proc Initialize(k, L, id^*)

$\beta \stackrel{R}{\leftarrow} \{0,1\}$

$g \stackrel{R}{\leftarrow} \mathbb{G}$

$a \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $A \leftarrow g^a$

$b \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $B \leftarrow g^b$

for $i = 1, \ldots, L$; $b = 0, 1$ do

$\quad h_{i,b} \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $H_{i,b} \leftarrow g^{h_{i,b}}$

$mpk \leftarrow (g, A, B, H_{1,0}, \ldots, H_{L,1})$

$msk \leftarrow g^{ab}$

Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

parse id^* as $(id_1^*, \ldots, id_\ell^*)$

$t \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $C_1 \leftarrow g^t$

for $i = 1, \ldots, \ell$ do

$\quad C_{2,i} \leftarrow (H_{i,0}^{id_i^*} H_{i,1})^t$

$K \leftarrow \hat{e}(A, B)^t$

$C_3^* \leftarrow m_\beta^* \cdot K$

Return $(C_1, (C_{2,1}, \ldots, C_{2,\ell}), C_3)$

proc KeyDer(id)

parse id as $(id_1, \ldots, id_\ell)$

for $i = 1, \ldots, \ell$ do

$\quad r_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $sk_i \leftarrow g^{r_i}$

$sk_0 \leftarrow g^{ab} \prod_{i=1}^{\ell} (H_{i,0}^{id_i} H_{i,1})^{r_i}$

Return $(sk_0, \ldots, sk_\ell)$

Game G_1

Game $G_1^{\mathcal{A}}$

proc Initialize(k, L, id^*)

$\beta \stackrel{R}{\leftarrow} \{0,1\}$

$g \stackrel{R}{\leftarrow} \mathbb{G}$

$a \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $A \leftarrow g^a$

$b \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $B \leftarrow g^b$

for $i = 1, \ldots, L$; do

$\boxed{\alpha'_i \stackrel{R}{\leftarrow} \mathbb{Z}_p \ ; \ H_{i,0} \leftarrow B^{\alpha'_i}}$

$\boxed{\alpha_i \stackrel{R}{\leftarrow} \mathbb{Z}_p \ ; \ H_{i,1} \leftarrow g^{\alpha_i} B^{-id^*_i \alpha'_i}}$

$mpk \leftarrow (g, A, B, H_{1,0}, \ldots, H_{L,1})$

$msk \leftarrow g^{ab}$

Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

parse id^* as $(id_1^*, \ldots, id_\ell^*)$

$t \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $C_1 \leftarrow g^t$

for $i = 1, \ldots, \ell$ do

$C_{2,i} \leftarrow (H_{i,0}^{id_i^*} H_{i,1})^t$

$K \leftarrow \hat{e}(A, B)^t$

$C_3^* \leftarrow m_\beta^* \cdot K$

Return $(C_1, (C_{2,1}, \ldots, C_{2,\ell}), C_3)$

proc KeyDer(id)

parse id as $(id_1, \ldots, id_\ell)$

for $i = 1, \ldots, \ell$ do

$r_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $sk_i \leftarrow g^{r_i}$

$sk_0 \leftarrow g^{ab} \prod_{i=1}^{\ell} (H_{i,0}^{id_i} H_{i,1})^{r_i}$

Return $(sk_0, \ldots, sk_\ell)$

Game G_2

Game $\mathsf{G}_2^{\mathcal{A}}$

proc Initialize(k, L, id^*)

$\beta \xleftarrow{R} \{0,1\}$
$g \xleftarrow{R} \mathbb{G}$
$a \xleftarrow{R} \mathbb{Z}_p$; $A \leftarrow g^a$
$b \xleftarrow{R} \mathbb{Z}_p$; $B \leftarrow g^b$
for $i = 1, \ldots, L$; do
  $\alpha'_i \xleftarrow{R} \mathbb{Z}_p$; $H_{i,0} \leftarrow B^{\alpha'_i}$
  $\alpha_i \xleftarrow{R} \mathbb{Z}_p$; $H_{i,1} \leftarrow g^{\alpha_i} B^{-id_i^* \alpha'_i}$
$mpk \leftarrow (g, A, B, H_{1,0}, \ldots, H_{L,1})$
$msk \leftarrow g^{ab}$
Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

parse id^* as $(id_1^*, \ldots, id_\ell^*)$
$t \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow g^t$
for $i = 1, \ldots, \ell$ do
  $C_{2,i} \leftarrow (H_{i,0}^{id_i^*} H_{i,1})^t$
$K \leftarrow \hat{e}(A, B)^t$
$C_3^* \leftarrow m_\beta^* \cdot K$
Return $(C_1, (C_{2,1}, \ldots, C_{2,\ell}), C_3)$

proc KeyDer(id)

parse id as $(id_1, \ldots, id_\ell)$
for $i = 1, \ldots, j-1, j+1, \ldots, \ell$ do
  $r_i \xleftarrow{R} \mathbb{Z}_p$; $sk_i \leftarrow g^{r_i}$

$\boxed{r_j \xleftarrow{R} \mathbb{Z}_p \ ; \ sk_i \leftarrow g^{r_j} A^{-1/(\alpha'_j(id_j - id_j^*))}}$

$\boxed{sk_0 \leftarrow A^{-\alpha_j/(\alpha'_j(id_j - id_j^*))} \prod_{i=1}^{\ell} (H_{i,0}^{id_i} H_{i,1})^{r_i}}$

Return $(sk_0, \ldots, sk_\ell)$

Game $G_3^{\mathcal{A}}$

proc Initialize(k, L, id^*)

$\beta \xleftarrow{R} \{0,1\}$
$g \xleftarrow{R} \mathbb{G}$
$a \xleftarrow{R} \mathbb{Z}_p$; $A \leftarrow g^a$
$b \xleftarrow{R} \mathbb{Z}_p$; $B \leftarrow g^b$
for $i = 1, \ldots, L$; do
  $\alpha'_i \xleftarrow{R} \mathbb{Z}_p$; $H_{i,0} \leftarrow B^{\alpha'_i}$
  $\alpha_i \xleftarrow{R} \mathbb{Z}_p$; $H_{i,1} \leftarrow g^{\alpha_i} B^{-id_i^* \alpha'_i}$
$mpk \leftarrow (g, A, B, H_{1,0}, \ldots, H_{L,1})$
$msk \leftarrow g^{ab}$
Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

parse id^* as $(id_1^*, \ldots, id_\ell^*)$
$t \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow g^t$
for $i = 1, \ldots, \ell$ do
  $\boxed{C_{2,i} \leftarrow C_1^{*\,\alpha_i}}$
$K \leftarrow \hat{e}(A, B)^t$
$C_3^* \leftarrow m_\beta^* \cdot K$
Return $(C_1, (C_{2,1}, \ldots, C_{2,\ell}), C_3)$

proc KeyDer(id)

parse id as $(id_1, \ldots, id_\ell)$
for $i = 1, \ldots, j-1, j+1, \ldots, \ell$ do
  $r_i \xleftarrow{R} \mathbb{Z}_p$; $sk_i \leftarrow g^{r_i}$
$r_j \xleftarrow{R} \mathbb{Z}_p$; $sk_i \leftarrow g^{r_j} A^{-1/(\alpha'_j(id_j - id_j^*))}$
$sk_0 \leftarrow A^{-\alpha_j/(\alpha'_j(id_j - id_j^*))} \prod_{i=1}^{\ell} (H_{i,0}^{id_i} H_{i,1})^{r_i}$
Return $(sk_0, \ldots, sk_\ell)$

Game G_4

Game $G_4^{\mathcal{A}}$

proc Initialize(k, L, id^*)

$\beta \xleftarrow{R} \{0,1\}$
$g \xleftarrow{R} \mathbb{G}$
$a \xleftarrow{R} \mathbb{Z}_p$; $A \leftarrow g^a$
$b \xleftarrow{R} \mathbb{Z}_p$; $B \leftarrow g^b$
for $i = 1, \ldots, L$; do
  $\alpha'_i \xleftarrow{R} \mathbb{Z}_p$; $H_{i,0} \leftarrow B^{\alpha'_i}$
  $\alpha_i \xleftarrow{R} \mathbb{Z}_p$; $H_{i,1} \leftarrow g^{\alpha_i} B^{-id^*_i \alpha'_i}$
$mpk \leftarrow (g, A, B, H_{1,0}, \ldots, H_{L,1})$
$msk \leftarrow g^{ab}$
Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

parse id^* as $(id_1^*, \ldots, id_\ell^*)$
$t \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow g^t$
for $i = 1, \ldots, \ell$ do
  $C_{2,i} \leftarrow C_1^{*\,\alpha_i}$
$\boxed{K \xleftarrow{R} \mathbb{G}}$
$C_3^* \leftarrow m_\beta^* \cdot K$
Return $(C_1, (C_{2,1}, \ldots, C_{2,\ell}), C_3)$

proc KeyDer(id)

parse id as $(id_1, \ldots, id_\ell)$
for $i = 1, \ldots, j-1, j+1, \ldots, \ell$ do
  $r_i \xleftarrow{R} \mathbb{Z}_p$; $sk_i \leftarrow g^{r_i}$
$r_j \xleftarrow{R} \mathbb{Z}_p$; $sk_i \leftarrow g^{r_j} A^{-1/(\alpha'_j(id_j - id_j^*))}$
$sk_0 \leftarrow A^{-\alpha_j/(\alpha'_j(id_j - id_j^*))} \prod_{i=1}^{\ell} (H_{i,0}^{id_i} H_{i,1})^{r_i}$
Return $(sk_0, \ldots, sk_\ell)$

Probability analysis

Claim 1 $\mathbf{Adv}^{\text{s-ind-cpa}}_{\mathsf{BB},L,\mathcal{A}}(k) = 2 \cdot \Pr\left[\, \mathsf{G}_0^{\mathcal{A}} = \mathsf{true} \,\right] - 1$

Claim 2 $\Pr\left[\, \mathsf{G}_1^{\mathcal{A}} = \mathsf{true} \,\right] = \Pr\left[\, \mathsf{G}_0^{\mathcal{A}} = \mathsf{true} \,\right]$

Claim 3 $\Pr\left[\, \mathsf{G}_2^{\mathcal{A}} = \mathsf{true} \,\right] = \Pr\left[\, \mathsf{G}_1^{\mathcal{A}} = \mathsf{true} \,\right]$

Claim 4 $\Pr\left[\, \mathsf{G}_3^{\mathcal{A}} = \mathsf{true} \,\right] = \Pr\left[\, \mathsf{G}_2^{\mathcal{A}} = \mathsf{true} \,\right]$

Claim 5 $\left|\Pr\left[\, \mathsf{G}_4^{\mathcal{A}} = \mathsf{true} \,\right] - \Pr\left[\, \mathsf{G}_3^{\mathcal{A}} = \mathsf{true} \,\right]\right| \le \mathbf{Adv}^{\mathrm{bddh}}_{\mathcal{G},k}(\mathcal{B})$

Claim 6 $\Pr\left[\, \mathsf{G}_4^{\mathcal{A}} = \mathsf{true} \,\right] = 1/2$

It's straightforward to verify that the security theorem follows from the claims above.

Proof of Claims 1, 2, 4, and 6

- Claim 1 follows the security definition.
- Claim 2 follows from the fact that $H_{i,b}$ is still uniformly distributed in $\mathbb{G}$.
- Claim 4 follows from the fact that C_2^* is still being correctly computed.

$$\begin{aligned} C_{2,i}^* &= (H_{i,0}^{id_i^*} H_{i,1})^t \\ &= ((B^{\alpha_i'})^{id_i^*} g^{\alpha_i} B^{-id_i^* \alpha_i'})^t \\ &= g^{\alpha_i t} \\ &= C_1^{*\alpha_i} \end{aligned}$$

- Claim 6 follows from the fact that $\mathcal{A}$ has no information about β in G_4.

Proof of Claim 3

Claim 3 follows from the fact that $(sk_0, \ldots, sk_\ell)$ is still a valid random secret key for user $id = (id_1, \ldots, id_\ell)$, where $\tilde{r}_j = r_j - a/(\alpha'_j(id_j - id^*_j))$ is the randomness being used to generate sk_j.

$$
\begin{aligned}
sk_j &= g^{\tilde{r}_j} = g^{r_j - a/(\alpha'_j(id_j - id^*_j))} \\
&= g^{r_j} g^{-a/(\alpha'_j(id_j - id^*_j))} \\
&= g^{r_j} A^{-1/(\alpha'_j(id_j - id^*_j))} \\
sk_0 &= g^{ab}(H_{j,0}^{id_j} H_{j,1})^{\tilde{r}_j} \textstyle\prod_{i=1}^{j-1}(H_{i,0}^{id_i} H_{i,1})^{r_i} \prod_{i=j+1}^{\ell}(H_{i,0}^{id_i} H_{i,1})^{r_i} \\
&= g^{ab}((B^{\alpha'_j})^{id_j} g^{\alpha_j} B^{-id^*_j \alpha'_j})^{-a/(\alpha'_j(id_j - id^*_j))} \textstyle\prod_{i=1}^{\ell}(H_{i,0}^{id_i} H_{i,1})^{r_i} \\
&= g^{ab}((g^{b\alpha'_j})^{id_j} g^{\alpha_j} g^{-bid^*_j \alpha'_j})^{-a/(\alpha'_j(id_j - id^*_j))} \textstyle\prod_{i=1}^{\ell}(H_{i,0}^{id_i} H_{i,1})^{r_i} \\
&= g^{ab}(g^{b\alpha'_j(id_j - bid^*_j)} g^{\alpha_j})^{-a/(\alpha'_j(id_j - id^*_j))} \textstyle\prod_{i=1}^{\ell}(H_{i,0}^{id_i} H_{i,1})^{r_i} \\
&= g^{ab} g^{-ab}(g^{\alpha_j})^{-a/(\alpha'_j(id_j - id^*_j))} \textstyle\prod_{i=1}^{\ell}(H_{i,0}^{id_i} H_{i,1})^{r_i} \\
&= A^{-\alpha_j/(\alpha'_j(id_j - id^*_j))} \textstyle\prod_{i=1}^{\ell}(H_{i,0}^{id_i} H_{i,1})^{r_i}
\end{aligned}
$$

Proof of Claim 5

In order to prove Claim 5, we need to build an adversary $\mathcal{B}$ against the BDDH problem.

- Let $(\mathbb{G}, g, A, B, C, Z)$ be the input of $\mathcal{B}$.
- To simulate procedure **Initialize**, $\mathcal{B}$ sets $H_{i,0} = B^{\alpha'_i}$ and $H_{i,1} = g^{\alpha_i} B^{-id^*_i \alpha'_i}$ for random α_i, α'_i and returns $mpk = (g, A, B, H_{1,0}, \ldots, H_{L,1})$ as the public key.
- When simulating procedure **LR**, $\mathcal{B}$ sets $C^*_1 = C$, $C^*_{2,i} = {C^*_1}^{\alpha_i}$, and $K = Z$.
- $\mathcal{B}$ simulates procedures **KeyDer** and **Finalize** exactly as in G_3.
- When $\mathcal{B}$ is being executed in Game $\mathbf{Exp}^{\mathrm{bddh\text{-}0}}_{\mathcal{G},k}(\mathcal{B})$, $\mathcal{B}$ simulates G_3 to $\mathcal{A}$. That is, $\Pr\left[\,\mathrm{G}_3^{\mathcal{A}} = \mathrm{true}\,\right] = \Pr\left[\,\mathbf{Exp}^{\mathrm{bddh\text{-}0}}_{\mathcal{G},k}(\mathcal{B}) = \mathrm{true}\,\right]$.
- When $\mathcal{B}$ is being executed in Game $\mathbf{Exp}^{\mathrm{bddh\text{-}1}}_{\mathcal{G},k}(\mathcal{B})$, $\mathcal{B}$ simulates G_4 to $\mathcal{A}$. That is, $\Pr\left[\,\mathrm{G}_4^{\mathcal{A}} = \mathrm{true}\,\right] = \Pr\left[\,\mathbf{Exp}^{\mathrm{bddh\text{-}1}}_{\mathcal{G},k}(\mathcal{B}) = \mathrm{true}\,\right]$.
- The claim follows.