Identity-based encryption with wildcards

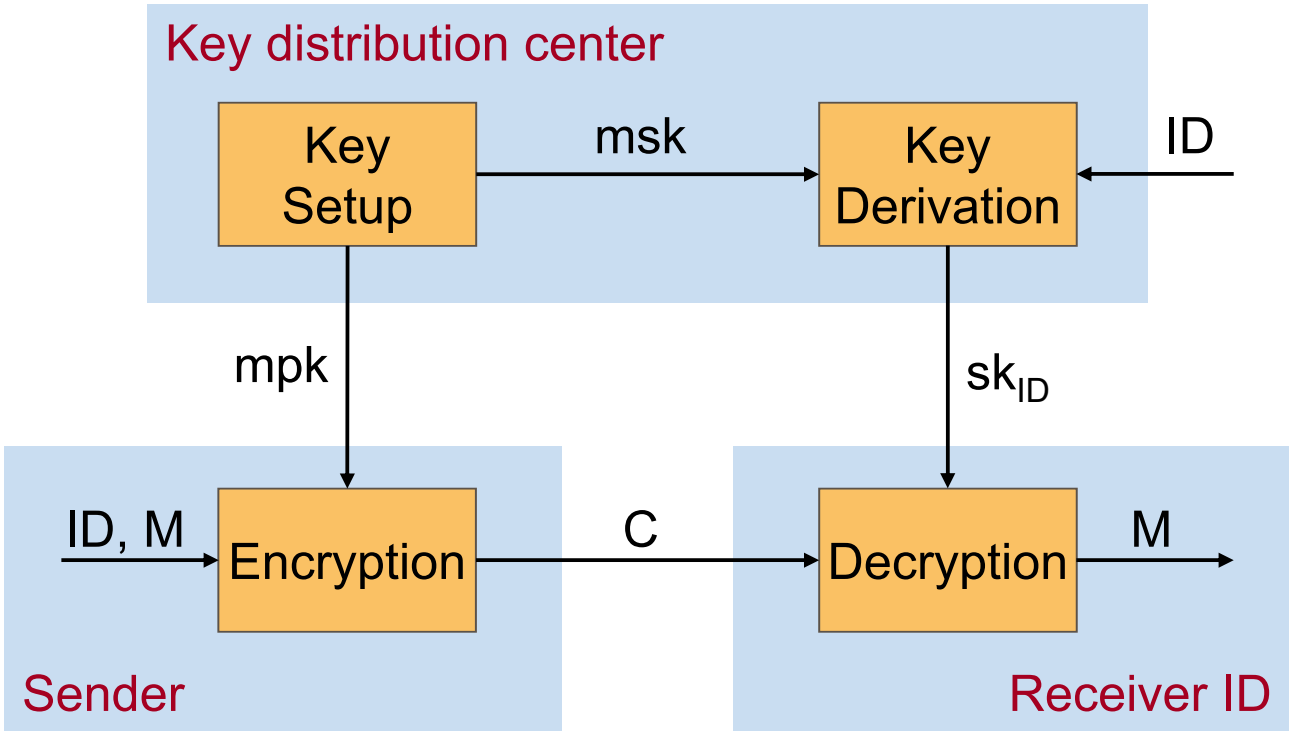
Michel Abdalla

ENS & CNRS

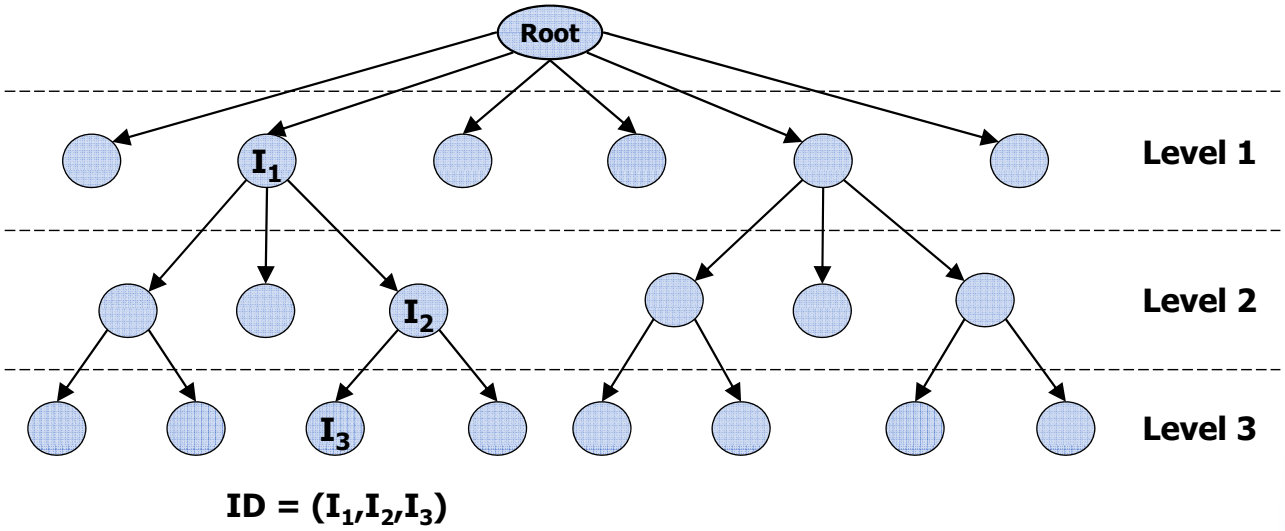
MPRI - Course 2-12-1

Identity-based encryption

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

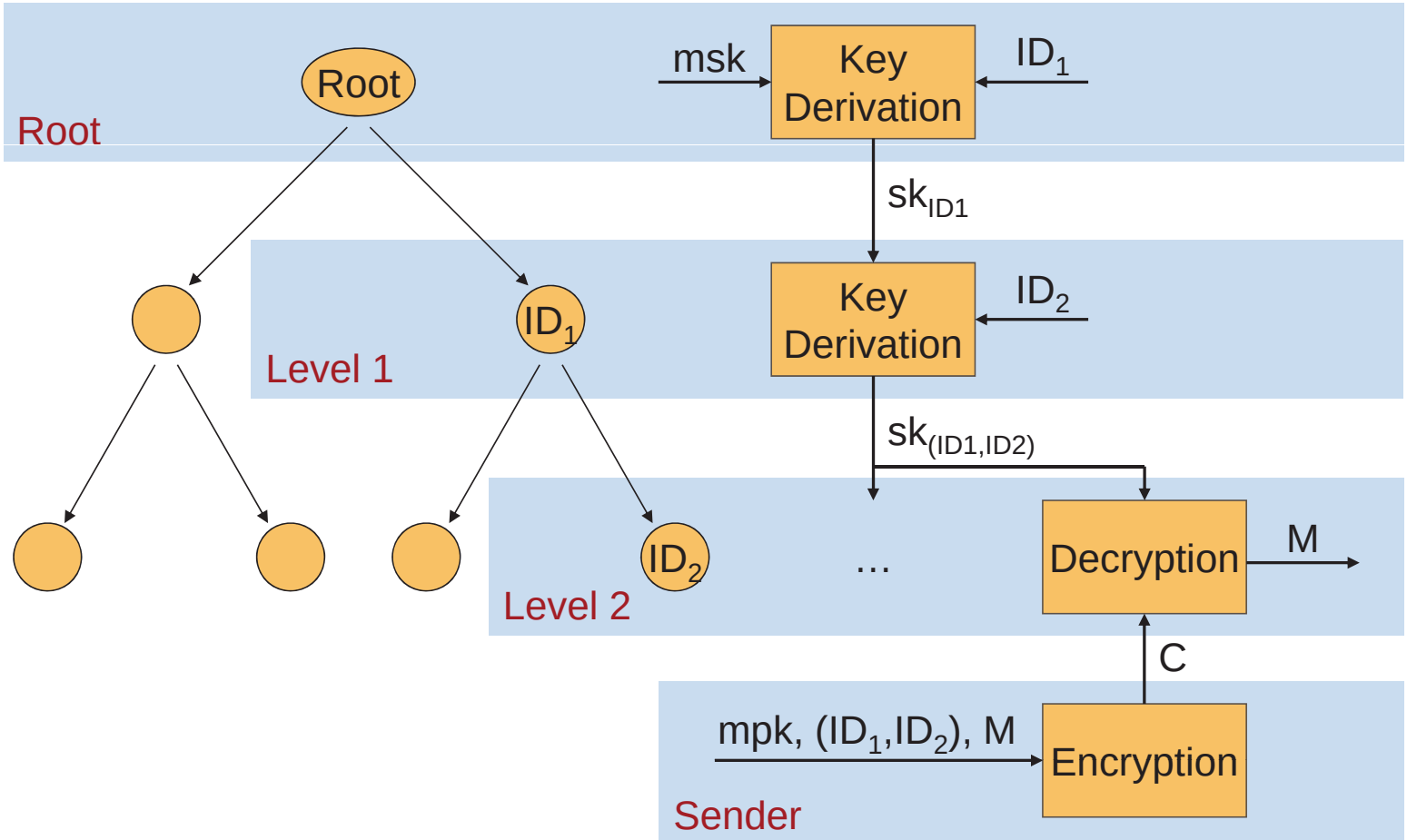


Hierarchical identity-based encryption (HIBE)



- 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 $(id_1, id_1, *, \ldots, *)$

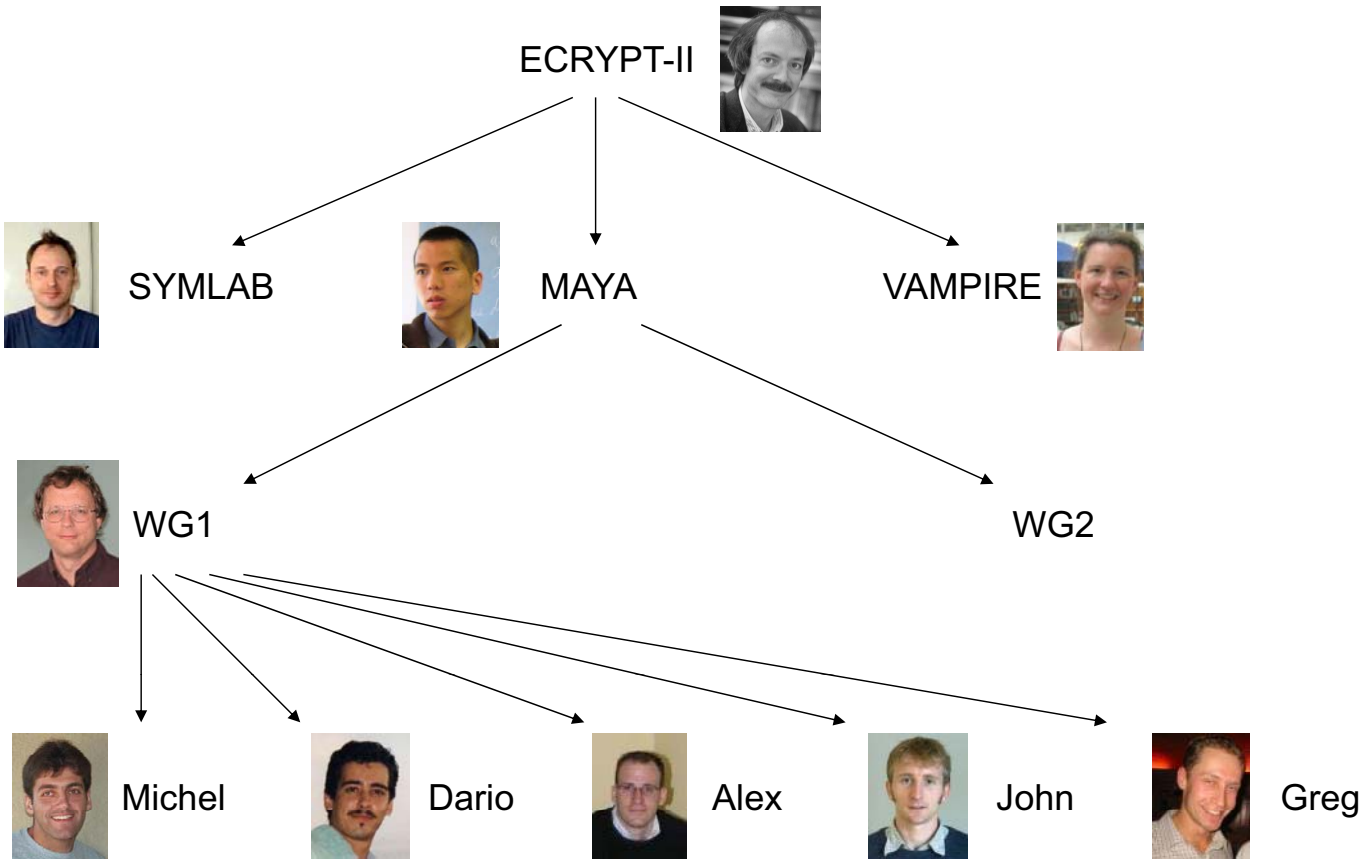
HIBE key derivation



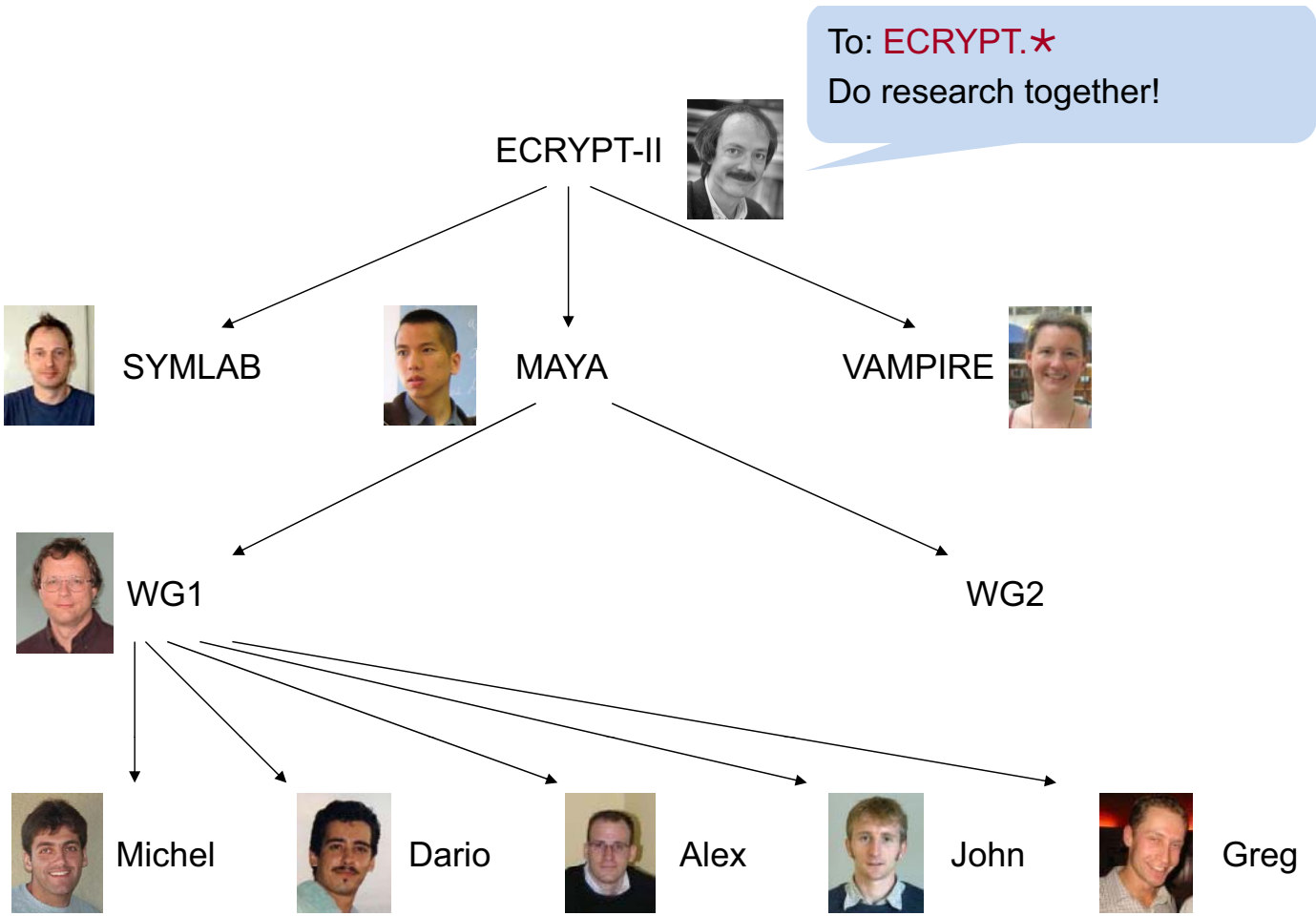
Identity-based encryption with wildcards (WIBE)

- Identities are vectors $(id_1, \ldots, id_L)$
- Hierarchical key derivation
- Encryption: receiver identity can contain "wildcards"
- Decryption by any "matching" identity
- Example
  $C = \mathrm{Enc}(mpk, (id_1, *, id_3), m)$ can be decrypted by any identity of the form (id'_1, id'_2, id'_3) where $id'_1 = id_1$ and $id'_3 = id_3$ but *by nobody else*.

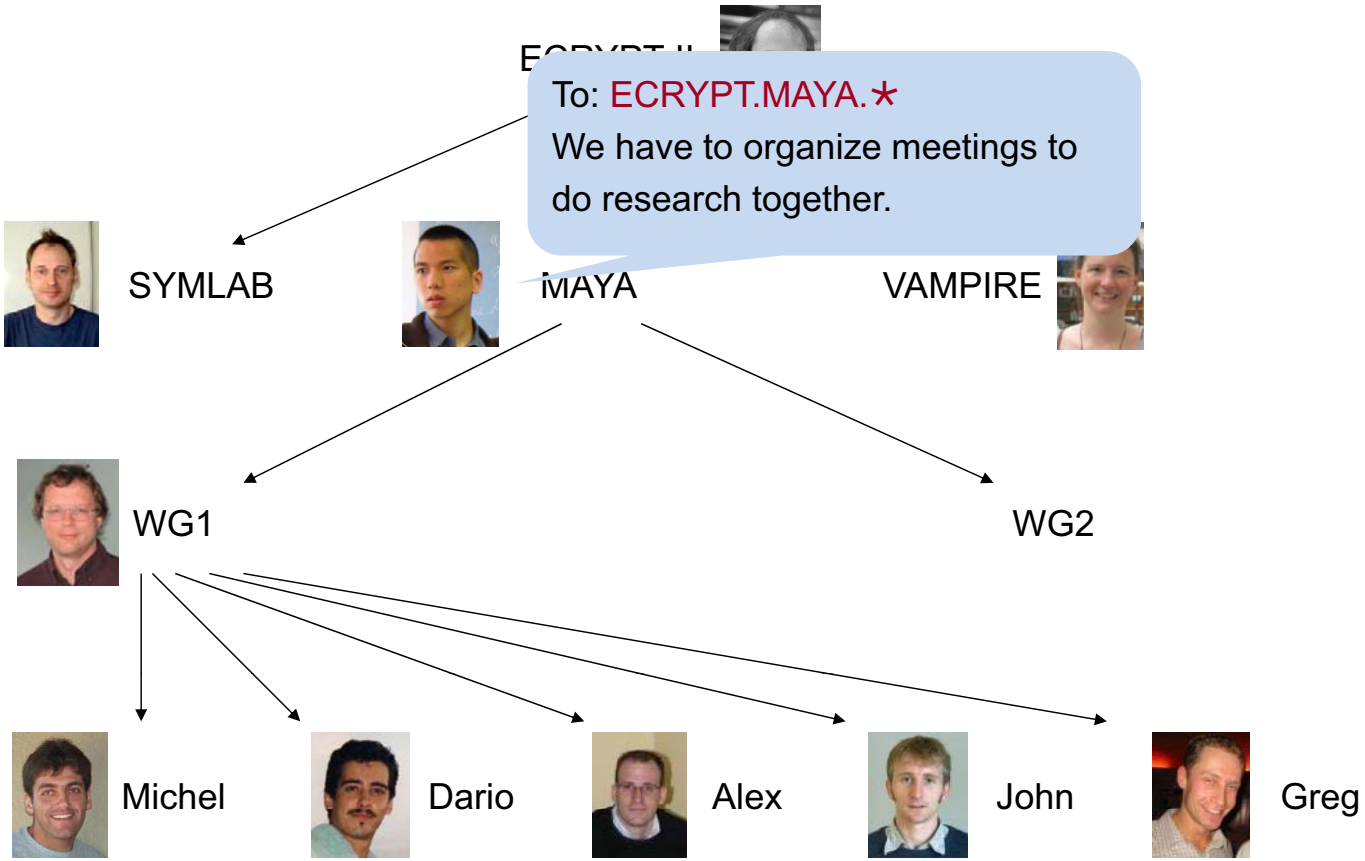
WIBE example 1



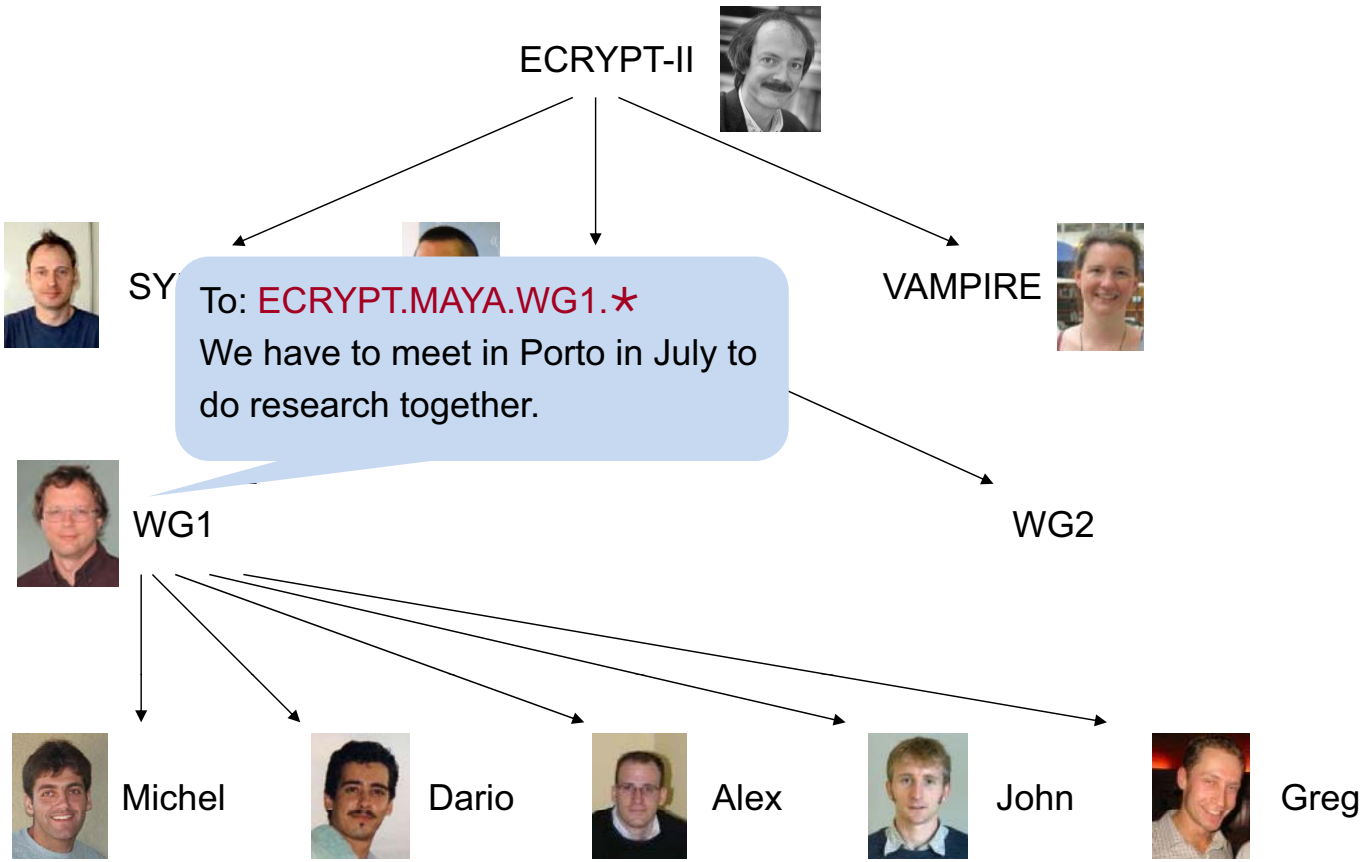
WIBE example 1



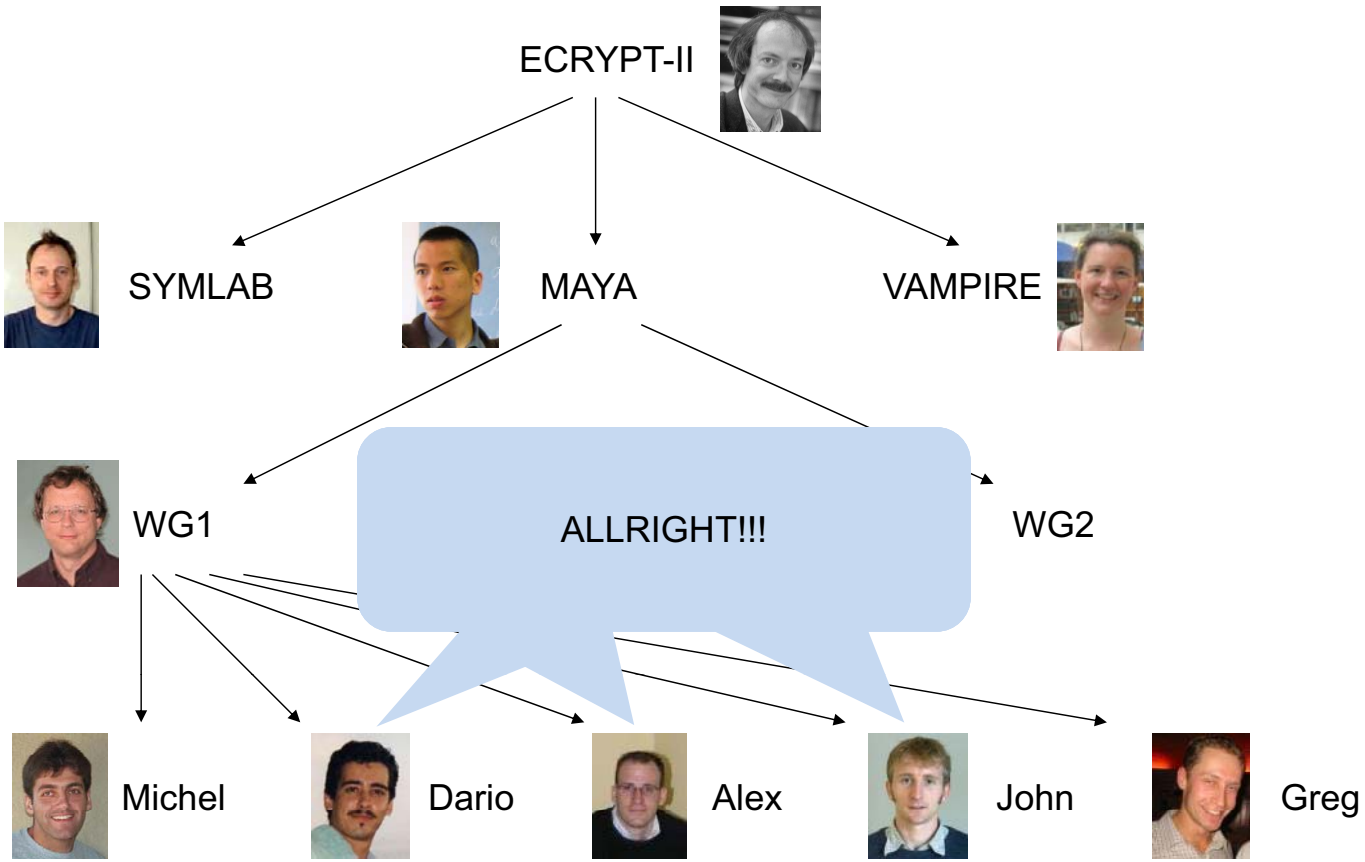
WIBE example 1



WIBE example 1



WIBE example 1



WIBE example 2

Structured email addresses `name@dept.univ.edu`
Send identity-based encrypted email to

- individual users: `JohnSmith@cs.univ.edu`
- computer science department: `*@cs.univ.edu`
- entire university: `*@*.univ.edu`
- all computer science departments: `*@cs.*.edu`
- all sysadmins: `sysadmin@*.univ.edu`
- spammers' dream: `*@*.*.*`

Outline

WIBE: Basic definitions

- A pattern at level $1 \leq \ell \leq L$ is a vector $P = (P_1, \ldots, P_\ell) \in (\{0,1\}^* \cup \{*\})^\ell$, where $*$ is a special wildcard symbol.
- An identity $id = (id_1, \ldots, id_{\ell'})$ *matches* P, denoted $id \in_* P$, if and only if $\ell' \leq \ell$ and for all $i = 1, \ldots, \ell'$ we have that $id_i = P_i$ or $P_i = *$.
- Root identity is represented by ε.

WIBE Syntax

An WIBE scheme is defined by four algorithms:

- Setup$(1^k, L)$:
  Outputs a master public key mpk for a WIBE 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, P, m):
  Generates a ciphertext C for pattern $P = (P_1, \ldots, P_\ell)$ and message m using master public key mpk.
- Dec(C_P, 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, if id *matches* P.

WIBE security notions

Here, we will consider two different attacks (*adaptive-identity* vs. *selective-identity*) and one goal (*indistinguishability*) for WIBE schemes.

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

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

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

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

proc Initialize(k, L)

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

Return mpk

proc KeyDer(id)

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

Return sk_{id}

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

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

Return C^*

proc Finalize(β')

Return β'

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

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

IND-WID-CPA: An alternative definition

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

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

proc Initialize(k, L)

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

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

Return mpk

proc KeyDer(id)

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

Return sk_{id}

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

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

Return C^*

proc Finalize(β')

Return $(\beta' = \beta)$

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

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

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

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

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

proc Initialize(k, L, P^*)

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

Return mpk

proc KeyDer(id)

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

Return sk_{id}

proc LR(m_0^*, m_1^*)

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

Return C^*

proc Finalize(β')

Return β'

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

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

IND-sWID-CPA: An alternative definition

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

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

proc Initialize(k, L, P^*)

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

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

Return mpk

proc KeyDer(id)

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

Return sk_{id}

proc LR(m_0^*, m_1^*)

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

Return C^*

proc Finalize(β')

Return $(\beta' = \beta)$

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

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

Outline

Boneh-Boyen WIBE scheme (BB-WIBE)

– $W(P) = \{1 \leq i \leq \ell \ : \ P_i = *\}$ denotes the set of wildcard positions in P.

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 $i = 0 \ldots L$; $b = 0, 1$ 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)

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

Enc(mpk, P, m):
parse P as $(P_1, \ldots, P_\ell)$
$t \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow g^t$
for $i = 1, \ldots, \ell$ do
if $i \notin W(P)$ then $C_{2,i} \leftarrow \left(H_{i,0}^{P_i} H_{i,1}\right)^t$
if $i \in W(P)$ then $C_{2,i} \leftarrow (H_{i,0}^t, H_{i,1}^t)$
$K \leftarrow \hat{e}(A, B)^t$
$C_3 \leftarrow m \cdot K$
return $(P, C_1, (C_{2,1}, \ldots, C_{2,\ell}), C_3)$

Dec$(sk_{(id_1,\ldots,id_\ell)}, C)$:
parse $sk_{(id_1,\ldots,id_\ell)}$ as $(sk_0, \ldots, sk_\ell)$
parse C as $(P, C_1, C_{2,1}, \ldots, C_{2,\ell}, C_3)$
for $i = 1, \ldots, \ell$ do
if $i \notin W(P)$ then $C'_{2,i} \leftarrow C_{2,i}$
if $i \in W(P)$ then
parse $C_{2,i}$ as $(v_{i,1}, v_{i,2})$
$C'_{2,i} \leftarrow v_{i,1}^{id_i} \cdot v_{i,2}$
$K' \leftarrow \hat{e}(sk_0, C_1) / \prod_{i=1}^{\ell} \hat{e}(sk_i, C'_{2,i})$
$m' \leftarrow C_3 / K'$
return m'

Additional comments about the BB-WIBE WIBE 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-WIBE WIBE scheme

– Let $id = (id_1, \ldots, id_\ell)$ and $P = (P_1, \ldots, P_{\ell'})$. If $id \in_* P$, then $\ell' \leq \ell$ and for all $i = 1, \ldots, \ell'$ we have that $id_i = P_i$ or $P_i = *$.
– Let $W(P) = \{1 \leq i \leq \ell' \ : \ P_i = *\}$ denotes the set of wildcard positions in P and let $W(P)_{\leq \ell}$ denote the subset of indices that are smaller or equal to ℓ.

For a valid ciphertext, we have:

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

Security of BB-WIBE WIBE scheme

Theorem

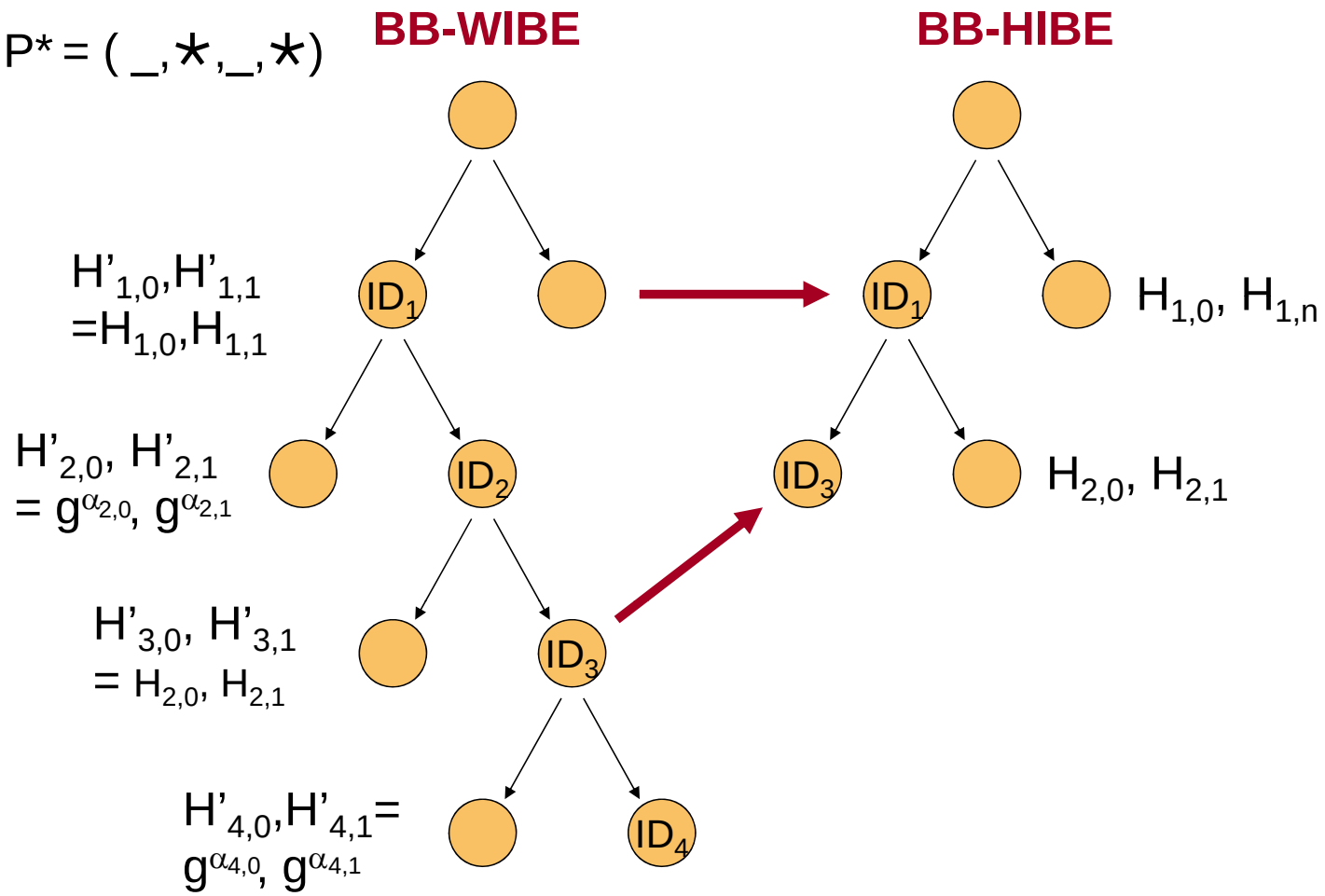
Let

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

Then, there exists an adversary $\mathcal{B}$ *against the* IND-sHID-CPA *security of* BB-HIBE, *whose running time is that of* $\mathcal{A}$ *and such that*

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

Proof idea


P* = (_,*,_,*)
BB-WIBE
BB-HIBE
H'1,0,H'1,1
=H1,0,H1,1
ID1
ID1
H1,0, H1,n
H'2,0, H'2,1
= gα2,0, gα2,1
ID2
ID3
H2,0, H2,1
H'3,0, H'3,1
= H2,0, H2,1
ID3
H'4,0,H'4,1=
gα4,0, gα4,1
ID4

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

– $W(P) = \{1 \leq i \leq \ell \ : \ P_i = *\}$ denotes the set of wildcard positions in P.

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, P, m):
Parse P as $(P_1, \ldots, P_\ell)$
$r \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow g_1^r$
$C_2 \leftarrow \left(u_0 \prod_{i=1, i \notin W(P)}^{\ell} u_i^{P_i}\right)^r$
$C_3 \leftarrow m \cdot \hat{e}(h_1, g_2)^r$
$C_4 \leftarrow (u_i^r)_{i \in W(P)}$
Return (P, C_1, C_2, C_3, C_4)

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 (P, C_1, C_2, C_3, C_4)
Parse C_4 as $(v_i)_{i \in W(P)}$
$C'_2 \leftarrow C_2 \prod_{i=1, i \in W(P)}^{\ell} v_i^{id_i}$
$m' \leftarrow C_3 \cdot \frac{\hat{e}(C'_2, sk_{L+1})}{\hat{e}(C_1, sk_0)}$
Return m'

Waters WIBE scheme (Wa-WIBE)

– $W(P) = \{1 \leq i \leq \ell \ : \ P_i = *\}$ denotes the set of wildcard positions in P.

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, P, m):

Parse P as $(P_1, \ldots, P_\ell)$

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

For $i = 1 \ldots \ell$ do

If $i \notin W(P)$ then $C_{2,i} \leftarrow F_i(id_i)^r$

If $i \in W(P)$ then $C_{2,i} \leftarrow (u_{i,0}^r, \ldots, u_{i,n}^r)$

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

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

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

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

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

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

If $i \notin W(P)$ then $C'_{2,i} \leftarrow C_{2,i}$

If $i \in W(P)$ then

Parse $C_{2,i}$ as $(v_0, \ldots, v_n)$

$C'_{2,i} \leftarrow v_0 \prod_{i \in [id_i]} v_i$

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

Return m'