Identity-based encryption

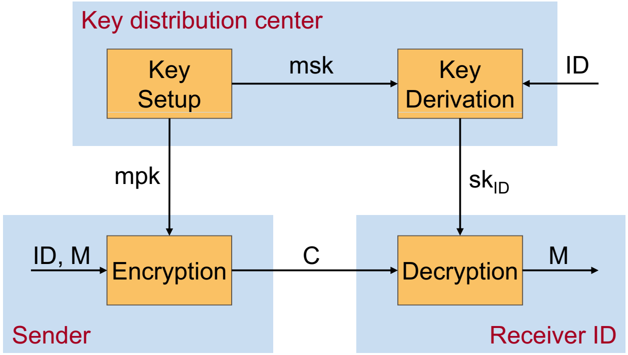
Michel Abdalla

ENS & CNRS

MPRI - Course 2-12-1

Identity-based encryption (IBE)

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



IBE properties

- Generalization of public-key encryption
  - User public key can be an arbitrary string (e.g., email address)
  - One system-wide public key for all users
- Encryption can be performed using system-wide public key and users identity
- Users need to contact a key generation center to obtain their secret keys

Outline

Identity-based encryption (IBE)

An IBE scheme is defined by four algorithms:

- Setup(1^k):
  Outputs a master public key mpk and a master secret key msk.
- KeyDer(msk, id):
  Uses the master secret key msk to compute a secret key sk_{id} for the user with identity id.
- Enc(mpk, id, m):
  Generates a ciphertext C for identity id and message m using master public key mpk.
- Dec(C, sk_{id}):
  Allows the user in possession of sk_{id} to decrypt the ciphertext C to get back a message m.

IBE security notions

In the following slides, we consider two different types of attacks (*adaptive-identity* vs. *selective-identity*) and two security goals (*indistinguishability* and *anonymity*) notions of security for IBE 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_1, 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-ID-CPA: Indistinguishability under chosen-plaintext attacks

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

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

proc Initialize(k)
$(mpk, msk) \stackrel{R}{\leftarrow} \mathsf{Setup}(1^k)$
Return mpk

proc KeyDer(id)
$sk_{id} \stackrel{R}{\leftarrow} \mathsf{KeyDer}(msk, id)$
Return sk_{id}

proc LR(id^*, m_0^*, m_1^*)
$C^* \stackrel{R}{\leftarrow} \mathsf{Enc}(mpk, id^*, m_\beta^*)$
Return C^*

proc Finalize(β')
Return β'

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

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

IND-ID-CPA: An alternative definition

- Let IBE $= (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ be an identity-based encryption scheme.
- Let $\mathcal{A}$ be an adversary against the IND-ID-CPA security of IBE.

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

<u>**proc Initialize**(k)</u>

$\beta \stackrel{R}{\leftarrow} \{0,1\}$
$(mpk, msk) \stackrel{R}{\leftarrow} \mathsf{Setup}(1^k)$
Return mpk

<u>**proc KeyDer**(id)</u>

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

<u>**proc LR**(id^*, m_0^*, m_1^*)</u>

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

<u>**proc Finalize**(β')</u>

Return $(\beta' = \beta)$

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

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

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

- Let IBE $= (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ be an identity-based encryption scheme.
- Let $\mathcal{A}$ be an adversary against the IND-sID-CPA security of IBE.

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

proc Initialize(k, id^*)	**proc LR**(m_0^*, m_1^*)
$(mpk, msk) \xleftarrow{R} \mathsf{Setup}(1^k)$	$C^* \xleftarrow{R} \mathsf{Enc}(mpk, id^*, m_\beta^*)$
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 IND-sID-CPA security of IBE is defined as

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

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

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

proc Initialize(k, id^*)

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

$(mpk, msk) \stackrel{R}{\leftarrow} \text{Setup}(1^k)$

Return mpk

proc KeyDer(id)

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

Return sk_{id}

proc LR(m_0^*, m_1^*)

$C^* \stackrel{R}{\leftarrow} \text{Enc}(mpk, id^*, m_\beta^*)$

Return C^*

proc Finalize(β')

Return $(\beta' = \beta)$

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

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

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

- Let IBE $=$ (Setup, KeyDer, Enc, Dec) be an identity-based encryption scheme.
- Let $\mathcal{A}$ be an adversary against the ANO-ID-CPA security of IBE.

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

proc Initialize(k)

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

proc KeyDer(id)

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

proc LR(id_0^*, id_1^*, m^*)

$C^* \stackrel{R}{\leftarrow} \mathsf{Enc}(mpk, id_\beta^*, m^*)$
Return C^*

proc Finalize(β')

Return β'

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

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

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

- Let IBE $= (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ be an identity-based encryption scheme.
- Let $\mathcal{A}$ be an adversary against the ANO-sID-CPA security of IBE.

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

proc Initialize(k)

$(mpk, msk) \xleftarrow{R} \mathsf{Setup}(1^k, id_0^*, id_1^*)$
Return mpk

proc KeyDer(id)

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

proc LR(m^*)

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

proc Finalize(β')

Return β'

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

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

Outline

Computational Diffie-Hellman (CDH)

- Let $\mathbb{G}$ be a finite cyclic group of prime order p.
- Let $\mathcal{A}$ be an adversary against the CDH problem in a group $\mathbb{G}$.

Game $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{cdh}}(\mathcal{A})$

proc Initialize($\mathbb{G}$)	**proc Finalize(Z)**
$g \stackrel{R}{\leftarrow} \mathbb{G}^*$ $x \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$; $X \leftarrow g^x$ $y \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$; $Y \leftarrow g^y$ Return $(\mathbb{G}, g, X, Y)$	Return $(Z = g^{xy})$

The advantage of $\mathcal{A}$ against the CDH problem is defined as

$$\mathbf{Adv}_{\mathbb{G}}^{\mathrm{cdh}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\mathbb{G}}^{\mathrm{cdh}}(\mathcal{A}) = \mathrm{true}\right]$$

Decisional Diffie-Hellman (DDH)

- Let $\mathbb{G}$ be a finite cyclic group of prime order p.
- Let $\mathcal{A}$ be an adversary against the CDH problem in a group $\mathbb{G}$.

Game $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh}\text{-}0}(\mathcal{A})$	**Game $\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh}\text{-}1}(\mathcal{A})$**
proc Initialize$(\mathbb{G})$	**proc Initialize**$(\mathbb{G})$
$g \xleftarrow{R} \mathbb{G}^*$	$g \xleftarrow{R} \mathbb{G}^*$
$x \xleftarrow{R} \mathbb{Z}_p^*$; $X \leftarrow g^x$	$x \xleftarrow{R} \mathbb{Z}_p^*$; $X \leftarrow g^x$
$y \xleftarrow{R} \mathbb{Z}_p^*$; $Y \leftarrow g^y$	$y \xleftarrow{R} \mathbb{Z}_p^*$; $Y \leftarrow g^y$
$z \leftarrow ab \mod p$; $Z \leftarrow g^z$	$z \xleftarrow{R} \mathbb{Z}_p^*$; $Z \leftarrow g^z$
Return $(\mathbb{G}, g, X, Y, Z)$	Return $(\mathbb{G}, g, X, Y, Z)$
proc Finalize(β')	**proc Finalize**(β')
Return $(\beta' = 1)$	Return $(\beta' = 1)$

The advantage of $\mathcal{A}$ in solving the DDH problem is defined as

$$\mathbf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh}\text{-}0}(\mathcal{A}) = \mathsf{true}\right] - \Pr\left[\mathbf{Exp}_{\mathbb{G}}^{\mathrm{ddh}\text{-}1}(\mathcal{A}) = \mathsf{true}\right]$$

Bilinear Diffie-Hellman (BDH)

- Let $\mathcal{G}$ be a *pairing parameter generator*.
- Let $\mathcal{A}$ be an adversary against the BDH problem relative to $\mathcal{G}$.

Game $\mathbf{Exp}^{\mathrm{bdh}}_{\mathcal{G},k}(\mathcal{A})$

proc Initialize(1^k)	**proc Finalize**(Z)
$(\mathbb{G}, \mathbb{G}_T, p, \hat{e}) \stackrel{R}{\leftarrow} \mathcal{G}(1^k)$	Return $(Z = \hat{e}(g,g)^{abc})$
$g \stackrel{R}{\leftarrow} \mathbb{G}^*$	
$a \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$; $A \leftarrow g^a$	
$b \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$; $B \leftarrow g^b$	
$c \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$; $C \leftarrow g^b$	
Return $(\mathbb{G}, g, A, B, C)$	

The advantage of $\mathcal{A}$ against the BDH problem relative to $\mathcal{G}$ is defined as

$$\mathbf{Adv}^{\mathrm{bdh}}_{\mathcal{G},k}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\mathrm{bdh}}_{\mathcal{G},k}(\mathcal{A}) = \mathrm{true}\right]$$

Bilinear Decisional Diffie-Hellman (BDDH)

- Let $\mathcal{G}$ be a *pairing parameter generator*.
- Let $\mathcal{A}$ be an adversary against the BDDH problem relative to $\mathcal{G}$.

Game $\mathbf{Exp}^{\mathrm{bddh\text{-}0}}_{\mathcal{G},k}(\mathcal{A})$	Game $\mathbf{Exp}^{\mathrm{bddh\text{-}1}}_{\mathcal{G},k}(\mathcal{A})$
proc Initialize(1^k)	**proc Initialize**(1^k)
$(\mathbb{G}, \mathbb{G}_T, p, \hat{e}) \xleftarrow{R} \mathcal{G}(1^k)$	$(\mathbb{G}, \mathbb{G}_T, p, \hat{e}) \xleftarrow{R} \mathcal{G}(1^k)$
$g \xleftarrow{R} \mathbb{G}^*$	$g \xleftarrow{R} \mathbb{G}^*$
$a \xleftarrow{R} \mathbb{Z}_p^*$; $A \leftarrow g^a$	$a \xleftarrow{R} \mathbb{Z}_p^*$; $A \leftarrow g^a$
$b \xleftarrow{R} \mathbb{Z}_p^*$; $B \leftarrow g^b$	$b \xleftarrow{R} \mathbb{Z}_p^*$; $B \leftarrow g^b$
$c \xleftarrow{R} \mathbb{Z}_p^*$; $C \leftarrow g^b$	$c \xleftarrow{R} \mathbb{Z}_p^*$; $C \leftarrow g^b$
$z \leftarrow abc \mod p$; $Z \leftarrow \hat{e}(g,g)^z$	$z \xleftarrow{R} \mathbb{Z}_p^*$; $Z \leftarrow \hat{e}(g,g)^z$
Return $(\mathbb{G}, g, A, B, C, Z)$	Return $(\mathbb{G}, g, A, B, C, Z)$
proc Finalize(β')	**proc Finalize**(β')
Return $(\beta' = 1)$	Return $(\beta' = 1)$

The advantage of $\mathcal{A}$ in solving the BDDH problem is defined as

$$\mathbf{Adv}^{\mathrm{bddh}}_{\mathcal{G},k}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\mathrm{bddh\text{-}0}}_{\mathcal{G},k}(\mathcal{A}) = \mathsf{true}\right] - \Pr\left[\mathbf{Exp}^{\mathrm{bddh\text{-}1}}_{\mathcal{G},k}(\mathcal{A}) = \mathsf{true}\right]$$

Outline

Boneh-Boyen IBE scheme (BB1)

Setup(1^k):
$(\mathbb{G}, \mathbb{G}_{\mathrm{T}}, p, \hat{e}) \overset{R}{\leftarrow} \mathcal{G}(1^k)$
$g \overset{R}{\leftarrow} \mathbb{G}$
$a \overset{R}{\leftarrow} \mathbb{Z}_p \ ; \ A \leftarrow g^a$
$b \overset{R}{\leftarrow} \mathbb{Z}_p \ ; \ B \leftarrow g^b$
$h \overset{R}{\leftarrow} \mathbb{Z}_p \ ; \ H \leftarrow g^h$
$mpk \leftarrow (g, A, B, H, \mathbb{G}, \mathbb{G}_{\mathrm{T}}, p, \hat{e})$
$msk \leftarrow g^{ab}$
return (mpk, msk)

Enc(mpk, id, m):
$t \overset{R}{\leftarrow} \mathbb{Z}_p \ ; \ C_1 \leftarrow g^t$
$C_2 \leftarrow \left(B^{id} H\right)^t$
$K \leftarrow \hat{e}(A, B)^t$
$C_3 \leftarrow m \cdot K$
return (C_1, C_2, C_3)

KeyDer(msk, id):
$r \overset{R}{\leftarrow} \mathbb{Z}_p$
$usk_1 \leftarrow g^r$
$usk_2 \leftarrow msk \cdot \left(B^{id} H\right)^r$
return (usk_1, usk_2)

Dec(usk, C):
parse usk as (usk_1, usk_2)
parse C as (C_1, C_2, C_3)
$K' \leftarrow \hat{e}(usk_2, C_1)/\hat{e}(usk_1, C_2)$
$m' \leftarrow C_3/K'$
return m'

Correctness of BB1 IBE scheme

For a valid ciphertext, we have:

$$
\begin{aligned}
K' &= \hat{e}(usk_2, C_1)/\hat{e}(usk_1, C_2) \\
&= \hat{e}(msk \cdot (B^{id}H)^r, g^t)/\hat{e}(g^r, (B^{id}H)^t) \\
&= \hat{e}(g^{ab} \cdot (B^{id}H)^r, g^t)/\hat{e}(g^r, (B^{id}H)^t) \\
&= \hat{e}(g^{ab}, g^t) \cdot \hat{e}((B^{id}H)^r, g^t)/\hat{e}(g^r, (B^{id}H)^t) \\
&= \hat{e}(g^a, g^b)^t \cdot \hat{e}((B^{id}H), g)^{rt}/\hat{e}(g, (B^{id}H))^{rt} \\
&= \hat{e}(A, B)^t \\
&= K
\end{aligned}
$$

BDDH security of BB1 IBE scheme

Theorem

Let

- BB1 *refer to the Boneh-Boyen IBE scheme described above,*
- $\mathcal{G}$ *be a pairing parameter generator, and*
- $\mathcal{A}$ *be an adversary against* $\mathrm{IND\text{-}sID\text{-}CPA}$ *security of* BB1, *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*

$$\mathbf{Adv}^{\mathrm{s\text{-}ind\text{-}cpa}}_{\mathcal{A},\mathsf{BB1}}(k) \leq 2 \cdot \mathbf{Adv}^{\mathrm{bddh}}_{\mathcal{G},k}(\mathcal{B}).$$

Security proof of BB1 scheme

Proof will define a sequence of five games $(\mathsf{G}_0, \ldots, \mathsf{G}_4)$.
For simplicity, we assume that $mpk = (g, A, B, H)$ and omit the other values. We also omit the pairing parameter generation in procedure **Initialize**.

- G_0 This game is the real attack game against BB1.
- G_1 We change the computation of H so that $B^{id^*} H = g^{\alpha}$ for a random α.
- 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 $C_2^* = C_1^{*\alpha}$. 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 $G_0^{\mathcal{A}}$

proc Initialize(k, id^*)

$\beta \xleftarrow{R} \{0,1\}$
$g \xleftarrow{R} \mathbb{G}$
$a \xleftarrow{R} \mathbb{Z}_p$; $A \leftarrow g^a$
$b \xleftarrow{R} \mathbb{Z}_p$; $B \leftarrow g^b$
$h \xleftarrow{R} \mathbb{Z}_p$; $H \leftarrow g^h$
$mpk \leftarrow (g, A, B, H)$
$msk \leftarrow g^{ab}$
Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

$t \xleftarrow{R} \mathbb{Z}_p$; $C_1^* \leftarrow g^t$
$C_2^* \leftarrow (B^{id^*} H)^t$
$K \leftarrow \hat{e}(A, B)^t$
$C_3^* \leftarrow m_\beta^* \cdot K$
Return (C_1^*, C_2^*, C_3^*)

proc KeyDer(id)

$r \xleftarrow{R} \mathbb{Z}_p$; $usk_1 \leftarrow g^r$
$usk_2 \leftarrow g^{ab} \cdot (B^{id} H)^r$
Return (usk_1, usk_2)

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

proc Initialize(k, id^*)

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

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

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

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

$\boxed{\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p \; ; \; H \leftarrow B^{-id^*} g^{\alpha}}$

$mpk \leftarrow (g, A, B, H)$

$msk \leftarrow g^{ab}$

Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

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

$C_2^* \leftarrow (B^{id^*} H)^t$

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

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

Return (C_1^*, C_2^*, C_3^*)

proc KeyDer(id)

$r \stackrel{R}{\leftarrow} \mathbb{Z}_p \; ; \; usk_1 \leftarrow g^r$

$usk_2 \leftarrow g^{ab} \cdot (B^{id} H)^r$

Return (usk_1, usk_2)

Game G_2

Game $G_2^{\mathcal{A}}$

proc Initialize(k, id^*)

$\beta \stackrel{R}{\leftarrow} \{0,1\}$
$g \stackrel{R}{\leftarrow} \mathbb{G}$
$a \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $A \leftarrow g^a$
$b \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $B \leftarrow g^b$
$\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $H \leftarrow B^{-id^*} g^{\alpha}$
$mpk \leftarrow (g, A, B, H)$
$msk \leftarrow g^{ab}$
Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

$t \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $C_1^* \leftarrow g^t$
$C_2^* \leftarrow (B^{id^*} H)^t$
$K \leftarrow \hat{e}(A, B)^t$
$C_3^* \leftarrow m_\beta^* \cdot K$
Return (C_1^*, C_2^*, C_3^*)

proc KeyDer(id)

$\tilde{r} \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $usk_1 \leftarrow g^{\tilde{r}} A^{-1/(id-id^*)}$
$usk_2 \leftarrow A^{-\alpha/(id-id^*)} \cdot (B^{id} H)^{\tilde{r}}$
Return (usk_1, usk_2)

Game G_3

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

proc Initialize(k, id^*)

$\beta \stackrel{R}{\leftarrow} \{0,1\}$
$g \stackrel{R}{\leftarrow} \mathbb{G}$
$a \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $A \leftarrow g^a$
$b \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $B \leftarrow g^b$
$\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $H \leftarrow B^{-id^*} g^{\alpha}$
$mpk \leftarrow (g, A, B, H)$
$msk \leftarrow g^{ab}$
Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

$t \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $C_1^* \leftarrow g^t$
$\boxed{C_2^* \leftarrow {C_1^*}^{\alpha}}$
$K \leftarrow \hat{e}(A, B)^t$
$C_3^* \leftarrow m_\beta^* \cdot K$
Return (C_1^*, C_2^*, C_3^*)

proc KeyDer(id)

$\tilde{r} \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $usk_1 \leftarrow g^{\tilde{r}} A^{-1/(id-id^*)}$
$usk_2 \leftarrow A^{-\alpha/(id-id^*)} \cdot (B^{id} H)^{\tilde{r}}$
Return (usk_1, usk_2)

Game G_4

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

proc Initialize(k, id^*)

$\beta \xleftarrow{R} \{0,1\}$
$g \xleftarrow{R} \mathbb{G}$
$a \xleftarrow{R} \mathbb{Z}_p$; $A \leftarrow g^a$
$b \xleftarrow{R} \mathbb{Z}_p$; $B \leftarrow g^b$
$\alpha \xleftarrow{R} \mathbb{Z}_p$; $H \leftarrow B^{-id^*} g^{\alpha}$
$mpk \leftarrow (g, A, B, H)$
$msk \leftarrow g^{ab}$
Return mpk

proc Finalize(β')

Return $(\beta' = \beta)$

proc LR(m_0^*, m_1^*)

$t \xleftarrow{R} \mathbb{Z}_p$; $C_1^* \leftarrow g^t$
$C_2^* \leftarrow {C_1^*}^{\alpha}$
$\boxed{K \xleftarrow{R} \mathbb{G}}$
$C_3^* \leftarrow m_\beta^* \cdot K$
Return (C_1^*, C_2^*, C_3^*)

proc KeyDer(id)

$\tilde{r} \xleftarrow{R} \mathbb{Z}_p$; $usk_1 \leftarrow g^{\tilde{r}} A^{-1/(id-id^*)}$
$usk_2 \leftarrow A^{-\alpha/(id-id^*)} \cdot (B^{id} H)^{\tilde{r}}$
Return (usk_1, usk_2)

Probability analysis

Claim 1 $\mathbf{Adv}^{\text{s-ind-cpa}}_{\mathcal{A},\mathsf{BB1}}(k) = 2 \cdot \Pr\left[\, \mathsf{G}_0^{\mathcal{A}} = \mathsf{true} \,\right] - 1$

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

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 is still uniformly distributed in $\mathbb{G}$.
- Claim 4 follows from the fact that C_2^* is still being correctly computed.

$$\begin{aligned} C_2^* &= (B^{id^*} H)^t \\ &= (B^{id^*} B^{-id^*} g^{\alpha})^t \\ &= g^{\alpha t} \\ &= C_1^{*\alpha} \end{aligned}$$

- 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 (usk_1, usk_2) is still a valid random secret key for user id, where r is being implicitly set to $\tilde{r} - a/(id - id^*)$.

$$
\begin{aligned}
usk_1 &= g^r = g^{\tilde{r}-a/(id-id^*)} \\
&= g^{\tilde{r}} g^{-a/(id-id^*)} \\
&= g^{\tilde{r}} A^{-1/(id-id^*)} \\
usk_2 &= g^{ab}(B^{id}H)^r \\
&= g^{ab}(B^{id}H)^{-a/(id-id^*)}(B^{id}H)^{\tilde{r}} \\
&= g^{ab}(B^{id}B^{-id^*}g^{\alpha})^{-a/(id-id^*)}(B^{id}H)^{\tilde{r}} \\
&= g^{ab}(g^{b(id-id^*)}g^{\alpha})^{-a/(id-id^*)}(B^{id}H)^{\tilde{r}} \\
&= g^{ab}g^{-ab}g^{-a\alpha/(id-id^*)}(B^{id}H)^{\tilde{r}} \\
&= A^{-\alpha/(id-id^*)}(B^{id}H)^{\tilde{r}}
\end{aligned}
$$

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**, $\mathcal{B}$ chooses α at random, sets $H = B^{-id^*} g^\alpha$ and returns $mpk = (g, A, B, H)$ as the public key.
- When simulating procedure **LR**, $\mathcal{B}$ sets $C_1^* = C$, $C_2^* = C_1^{*\alpha}$, 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, $\Pr\left[\, \mathsf{G}_3^{\mathcal{A}} = \mathsf{true} \,\right] = \Pr\left[\, \mathbf{Exp}^{\mathrm{bddh\text{-}0}}_{\mathcal{G},k}(\mathcal{B}) = \mathsf{true} \,\right]$.
- 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, $\Pr\left[\, \mathsf{G}_4^{\mathcal{A}} = \mathsf{true} \,\right] = \Pr\left[\, \mathbf{Exp}^{\mathrm{bddh\text{-}1}}_{\mathcal{G},k}(\mathcal{B}) = \mathsf{true} \,\right]$.
- The claim follows.

Boneh-Franklin BasicIdent IBE scheme

- Let $\mathcal{G}$ be a *pairing parameter generator*.
- Let $H : \{0,1\}^* \to \mathbb{G}^*$ be a random oracle.

Setup(1^k):
$(\mathbb{G}, \mathbb{G}_T, p, \hat{e}) \stackrel{R}{\leftarrow} \mathcal{G}(1^k)$
$g \stackrel{R}{\leftarrow} \mathbb{G}$; $s \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$; $S \leftarrow g^s$
$msk \leftarrow s$
$mpk \leftarrow ((\mathbb{G}, \mathbb{G}_T, p, \hat{e}), S, H)$
return (mpk, msk)

KeyDer(msk, id):
$Q_{id} \leftarrow H(id)$
$usk \leftarrow Q_{id}^s$
return (usk)

Enc(mpk, id, m):
$r \stackrel{R}{\leftarrow} \mathbb{Z}_p$; $C_1 \leftarrow g^r$
$Q_{id} \leftarrow H(id)$; $K \leftarrow (\hat{e}(S, Q_{id}))^r$
$C_2 \leftarrow m \cdot K$
return (C_1, C_2)

Dec(usk, C):
parse C as (C_1, C_2)
$K \leftarrow \hat{e}(C_1, usk)$
$m' \leftarrow C_2 / K$
return m'

BDDH security of Boneh-Franklin BasicIdent IBE scheme

Theorem

Let

- BF *refer to the Boneh-Franklin* BasicIdent *IBE scheme in the previous slide,*
- $\mathcal{G}$ *be a pairing parameter generator, and*
- $\mathcal{A}$ *be an adversary against* IND-ID-CPA *security of* BF*, making at most* q_H *queries to the random oracle* H *and 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*

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

Security proof of BF scheme

Proof will define a sequence of five games ($G_0, \ldots, G_4$).
For simplicity, we assume that $mpk = S$ and omit the other values. We also omit the pairing parameter generation in procedure **Initialize**.

- G_0 This game is the real attack game against BF.
- G_1 We guess the hash query involved in the challenge query and abort if the guess is incorrect, returning a random output for the game.
- G_2 We change the simulation of the random oracle procedure **H** so that the game knows the discrete log of $H(id)$ for any identity other than the challenge.
- G_3 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_4 In this game, we change the simulation of the **LR** procedure so that K is chosen uniformly at random.

Game G_0

Game $G_0^{\mathcal{A}}$

proc Initialize(k)

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

$\Lambda_H \leftarrow \varepsilon$; $ctr \leftarrow 0$

$s \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$; $S \leftarrow g^s$

$msk \leftarrow s$

$mpk \leftarrow S$

Return mpk

proc KeyDer(id)

if $(ctr, id, Y, y) \notin \Lambda_H$, $\mathbf{H}(id)$

$usk \leftarrow H(id)^s$

Return usk

proc H(id)

if $(ctr, id, Y, y) \in \Lambda_H$, return Y

$ctr \leftarrow ctr + 1$; $Y \stackrel{R}{\leftarrow} \mathbb{G}$

$\Lambda_H \leftarrow \Lambda_H \cup \{(ctr, id, Y, y)\}$

Return (C_1^*, C_2^*)

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

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

$K \leftarrow \hat{e}(S, H(id^*))^r$

$C_2^* \leftarrow m_\beta^* \cdot K$

Return (C_1^*, C_2^*)

proc Finalize(β')

Return $(\beta' = \beta)$

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

proc Initialize(k)

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

$\boxed{i^* \stackrel{R}{\leftarrow} \{1,\ldots,q_H\}}$

$\Lambda_H \leftarrow \varepsilon$; $ctr \leftarrow 0$

$s \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$; $S \leftarrow g^s$

$msk \leftarrow s$

$mpk \leftarrow S$

Return mpk

proc KeyDer(id)

if $(ctr, id, Y, y) \notin \Lambda_H$, $\mathbf{H}(id)$

$usk \leftarrow H(id)^s$

Return usk

proc H(id)

if $(ctr, id, Y, y) \in \Lambda_H$, return Y

$ctr \leftarrow ctr + 1$; $Y \stackrel{R}{\leftarrow} \mathbb{G}$

$\boxed{\text{if } i^* = ctr \text{ and } id \neq id^*, \text{ abort}}$

$\Lambda_H \leftarrow \Lambda_H \cup \{(ctr, id, Y, y)\}$

Return (C_1^*, C_2^*)

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

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

$K \leftarrow \hat{e}(S, H(id^*))^r$

$C_2^* \leftarrow m_\beta^* \cdot K$

Return (C_1^*, C_2^*)

Game $G_2^{\mathcal{A}}$

proc Initialize(k)

$\beta \xleftarrow{R} \{0,1\}$
$i^* \xleftarrow{R} \{1,\ldots,q_H\}$
$\Lambda_H \leftarrow \varepsilon$; $ctr \leftarrow 0$
$s \xleftarrow{R} \mathbb{Z}_p^*$; $S \leftarrow g^s$
$msk \leftarrow s$
$mpk \leftarrow S$
Return mpk

proc KeyDer(id)

if $(ctr, id, Y, y) \notin \Lambda_H$, $\mathbf{H}(id)$
$usk \leftarrow H(id)^s$
Return usk

proc H(id)

if $(ctr, id, Y, y) \in \Lambda_H$, return Y
$ctr \leftarrow ctr + 1$; $\boxed{y \xleftarrow{R} \mathbb{Z}_p \ ; \ Y \leftarrow g^y}$
if $i^* = ctr$ and $id \neq id^*$, abort
$\Lambda_H \leftarrow \Lambda_H \cup \{(ctr, id, Y, y)\}$
Return (C_1^*, C_2^*)

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

$r \xleftarrow{R} \mathbb{Z}_p$; $C_1^* \leftarrow g^r$
$K \leftarrow \hat{e}(S, H(id^*))^r$
$C_2^* \leftarrow m_\beta^* \cdot K$
Return (C_1^*, C_2^*)

Game G_3

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

proc Initialize(k)

$\beta \xleftarrow{R} \{0,1\}$
$i^* \xleftarrow{R} \{1,\ldots,q_H\}$
$\Lambda_H \leftarrow \varepsilon\ ;\ ctr \leftarrow 0$
$s \xleftarrow{R} \mathbb{Z}_p^*\ ;\ S \leftarrow g^s$
$msk \leftarrow s$
$mpk \leftarrow S$
Return mpk

proc KeyDer(id)

if $(ctr, id, Y, y) \notin \Lambda_H$, $\mathbf{H}(id)$
read $(ctr, id, Y, y) \in \Lambda_H$
$usk \leftarrow S^y$
Return usk

proc H(id)

if $(ctr, id, Y, y) \in \Lambda_H$, return Y
$ctr \leftarrow ctr + 1\ ;\ y \xleftarrow{R} \mathbb{Z}_p\ ;\ Y \leftarrow g^y$
if $i^* = ctr$ and $id \neq id^*$, abort
$\Lambda_H \leftarrow \Lambda_H \cup \{(ctr, id, Y, y)\}$
Return (C_1^*, C_2^*)

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

$r \xleftarrow{R} \mathbb{Z}_p\ ;\ C_1^* \leftarrow g^r$
$K \leftarrow \hat{e}(S, H(id^*))^r$
$C_2^* \leftarrow m_\beta^* \cdot K$
Return (C_1^*, C_2^*)

Game G_4

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

proc Initialize(k)

$\beta \xleftarrow{R} \{0,1\}$
$i^* \xleftarrow{R} \{1,\ldots,q_H\}$
$\Lambda_H \leftarrow \varepsilon$; $ctr \leftarrow 0$
$s \xleftarrow{R} \mathbb{Z}_p^*$; $S \leftarrow g^s$
$msk \leftarrow s$
$mpk \leftarrow S$
Return mpk

proc KeyDer(id)

if $(ctr, id, Y, y) \notin \Lambda_H$, $\mathbf{H}(id)$
read $(ctr, id, Y, y) \in \Lambda_H$
$usk \leftarrow S^y$
Return usk

proc H(id)

if $(ctr, id, Y, y) \in \Lambda_H$, return Y
$ctr \leftarrow ctr + 1$; $y \xleftarrow{R} \mathbb{Z}_p$; $Y \leftarrow g^y$
if $i^* = ctr$ and $id \neq id^*$, abort
$\Lambda_H \leftarrow \Lambda_H \cup \{(ctr, id, Y, y)\}$
Return (C_1^*, C_2^*)

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

$r \xleftarrow{R} \mathbb{Z}_p$; $C_1^* \leftarrow g^r$
$\boxed{K \xleftarrow{R} \mathbb{G}_T}$
$C_2^* \leftarrow m_\beta^* \cdot K$
Return (C_1^*, C_2^*)

Probability analysis

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

Claim 2 $\Pr\left[\, \mathsf{G}_1^{\mathcal{A}} = \mathsf{true} \,\right] = (1 - 1/q_H) \cdot 1/2 + 1/q_H \cdot \Pr\left[\, \mathsf{G}_0^{\mathcal{A}} = \mathsf{true} \,\right]$

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

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

- Claim 1 follows the security definition.
- Claims 3 and 4 are true because the changes made to the games do not affect their outcome.
- Claims 2 follows from the fact that the output of the game is chosen uniformly at random when aborting.

$$
\begin{aligned}
&\Pr\left[\,\mathsf{G}_1^{\mathcal{A}} = \text{true}\,\right] \\
&= \Pr\left[\,\mathsf{G}_1^{\mathcal{A}} = \text{true} \wedge \mathrm{abort}\,\right] + \Pr\left[\,\mathsf{G}_1^{\mathcal{A}} = \text{true} \wedge \overline{\mathrm{abort}}\,\right] \\
&= \Pr\left[\,\mathsf{G}_1^{\mathcal{A}} = \text{true} | \mathrm{abort}\,\right] \Pr\left[\,\mathrm{abort}\,\right] + \\
&\quad \Pr\left[\,\mathsf{G}_1^{\mathcal{A}} = \text{true} | \overline{\mathrm{abort}}\,\right] \Pr\left[\,\overline{\mathrm{abort}}\,\right] \\
&= 1/2 \cdot (1 - 1/q_H) + \Pr\left[\,\mathsf{G}_1^{\mathcal{A}} = \text{true} | \overline{\mathrm{abort}}\,\right] \cdot 1/q_H \\
&= 1/2 \cdot (1 - 1/q_H) + \Pr\left[\,\mathsf{G}_0^{\mathcal{A}} = \text{true}\,\right] \cdot 1/q_H
\end{aligned}
$$

Proof of claims (cont.)

- 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 $\mathcal{B}$.
  - $\mathcal{B}$ sets $mpk = A$, $C_1^* = B$, $H(id^*) = C$, and $K = Z$. Everything else in the simulation is performed 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, $\Pr\left[\mathsf{G}_3^{\mathcal{A}} = \mathsf{true}\right] = \Pr\left[\mathbf{Exp}^{\mathrm{bddh}\text{-}0}_{\mathcal{G},k}(\mathcal{B}) = \mathsf{true}\right]$.
  - 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, $\Pr\left[\mathsf{G}_4^{\mathcal{A}} = \mathsf{true}\right] = \Pr\left[\mathbf{Exp}^{\mathrm{bddh}\text{-}1}_{\mathcal{G},k}(\mathcal{B}) = \mathsf{true}\right]$.
  - The claim follows.
- Claim 6 follows from the fact that $\mathcal{A}$ has no information about β in G_4 and that the output of the game is chosen uniformly at random when aborting.

References I

Dan Boneh and Xavier Boyen.
Efficient selective-ID secure identity based encryption without random oracles.
EUROCRYPT 2004, LNCS 3027, pages 223–238, May 2004. Springer.

Dan Boneh and Matthew K. Franklin.
Identity based encryption from the Weil pairing.
SIAM Journal on Computing, 32(3):586–615, 2003.
Extended Abstract in CRYPTO 2001.

Adi Shamir.
Identity-based cryptosystems and signature schemes.
CRYPTO 1984, LNCS 196, pages 47–53, August 1984. Springer.