A Note on Adaptive Security in Hierarchical Identity-Based Encryption

Rishab Goyal
UW-Madison

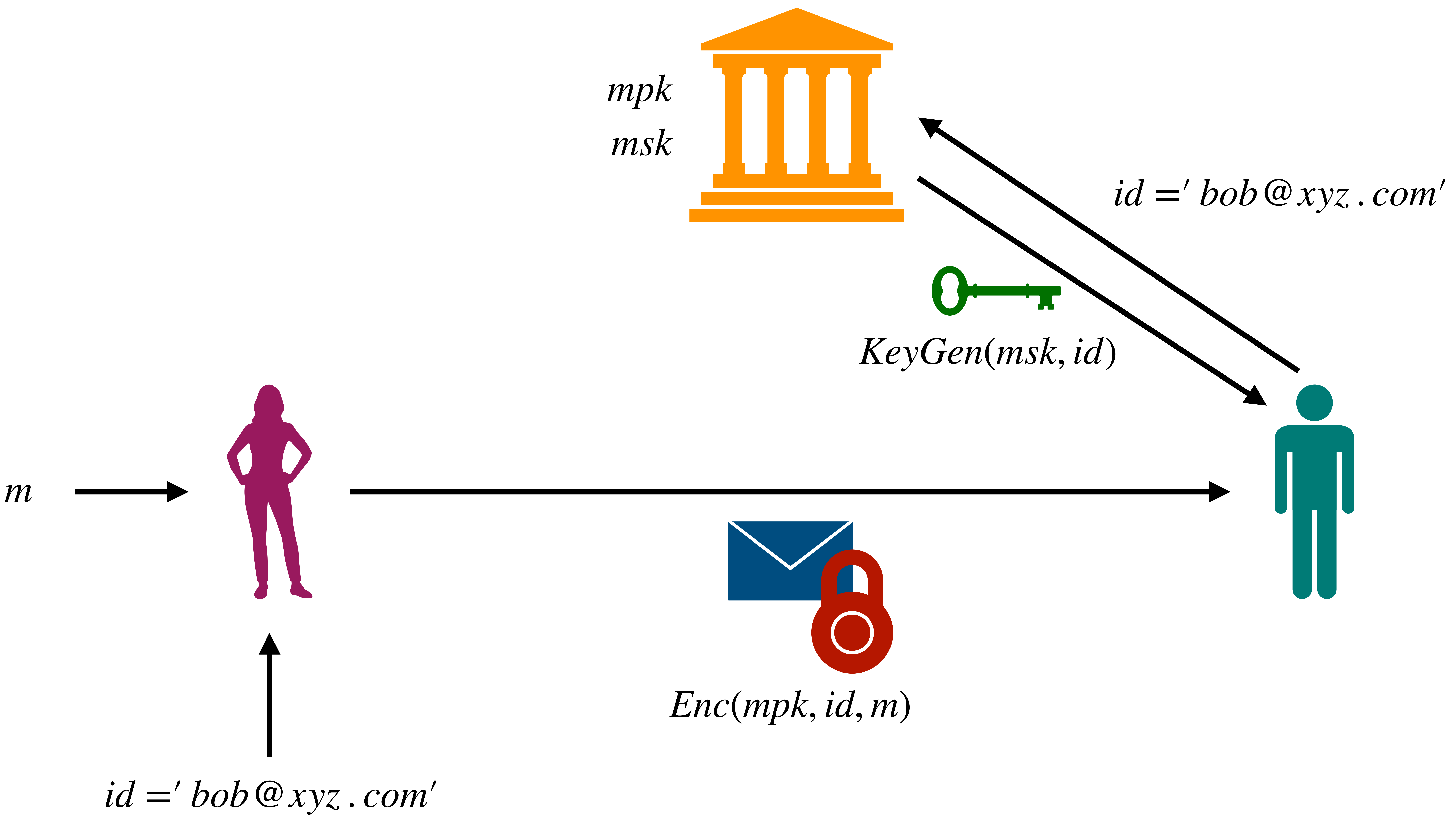
Venkata Koppula
IIT Delhi

Mahesh Sreekumar Rajasree
CISPA Helmholtz

CRYPTO 2025

Identity Based Encryption

[Shamir'85]



Hierarchical Identity Based Encryption

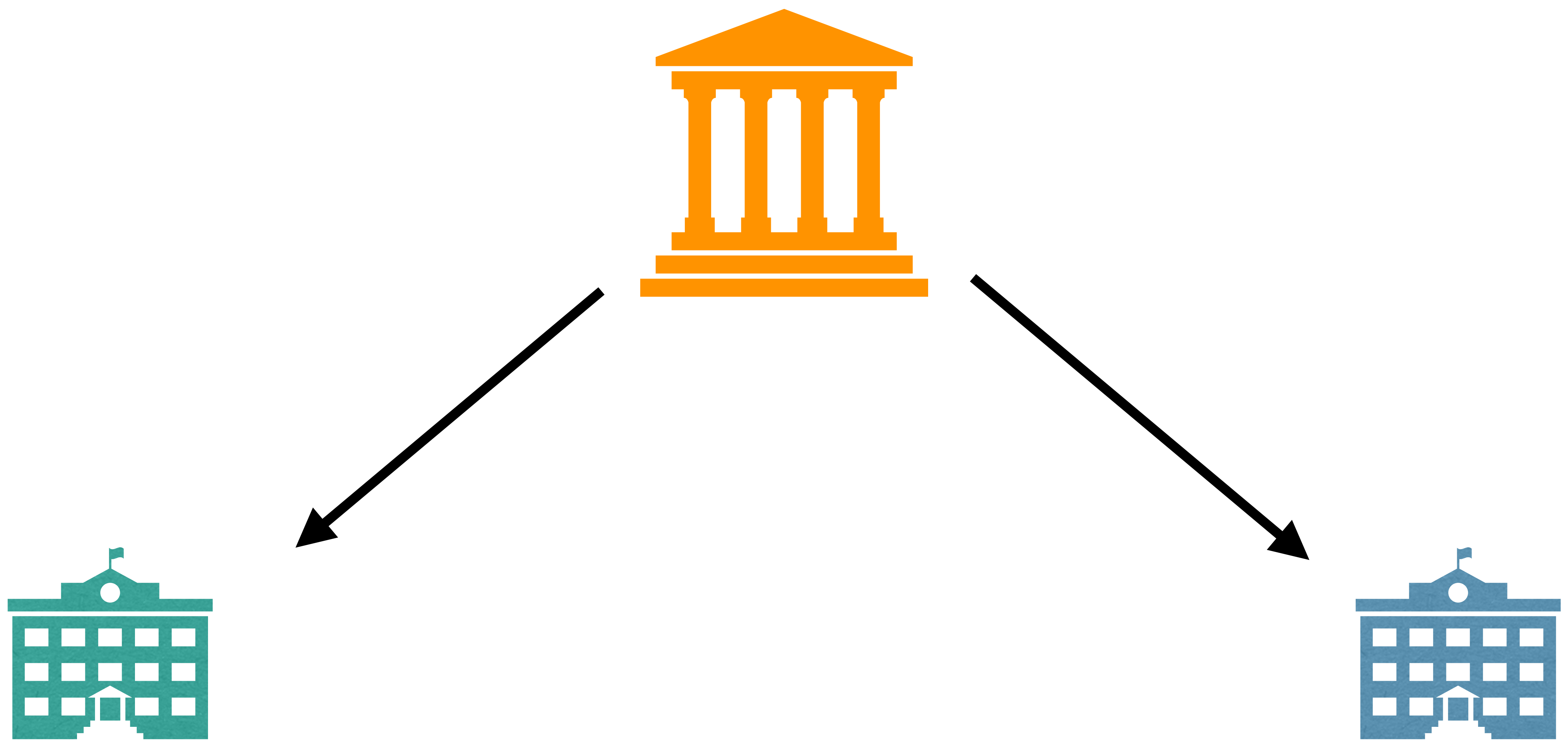
[Horwitz-Lynn'02,Gentry-Silverberg'02]

Hierarchical Identity Based Encryption

[Horwitz-Lynn’02,Gentry-Silverberg’02]

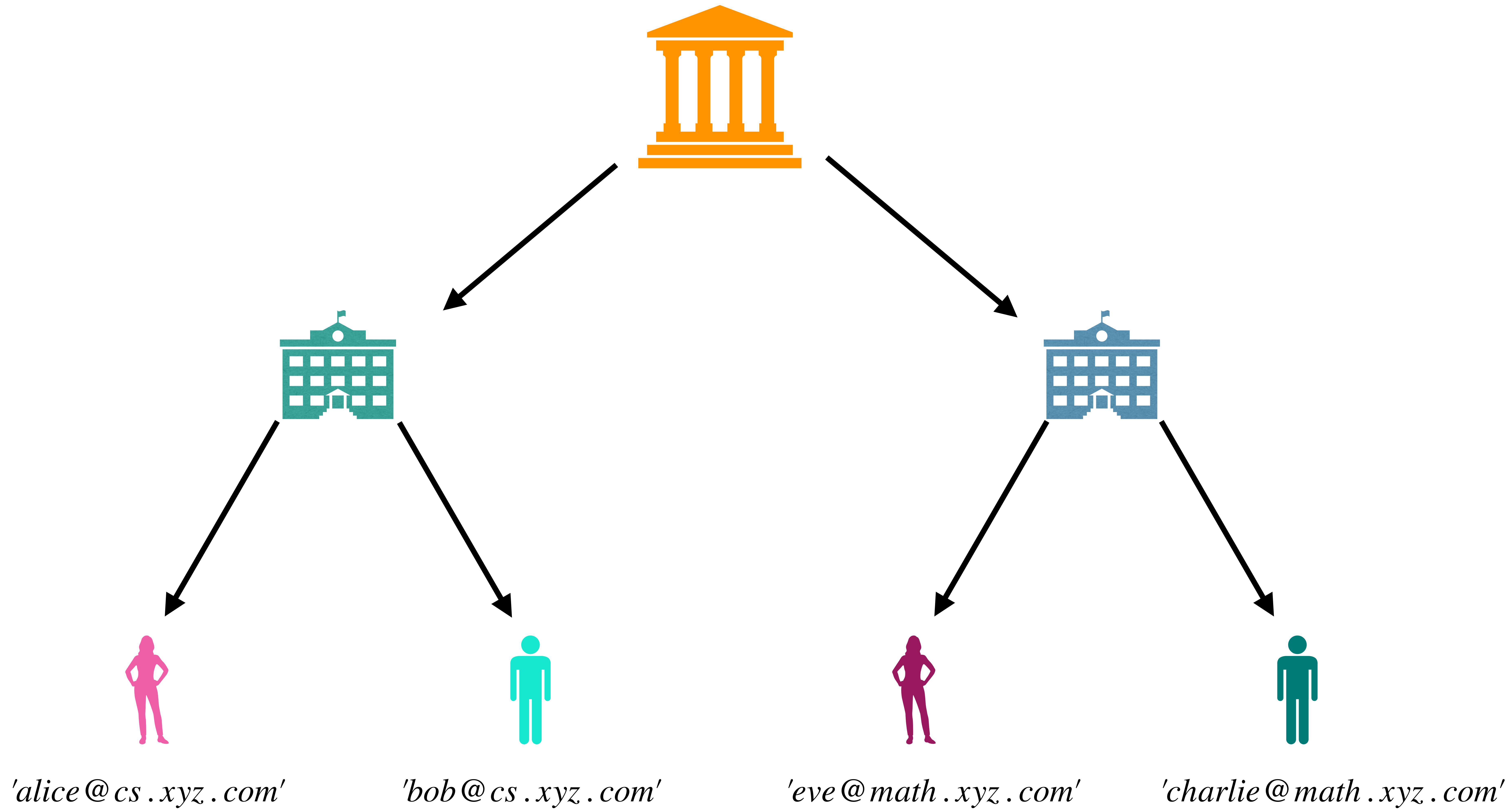
Hierarchical Identity Based Encryption

[Horwitz-Lynn’02,Gentry-Silverberg’02]



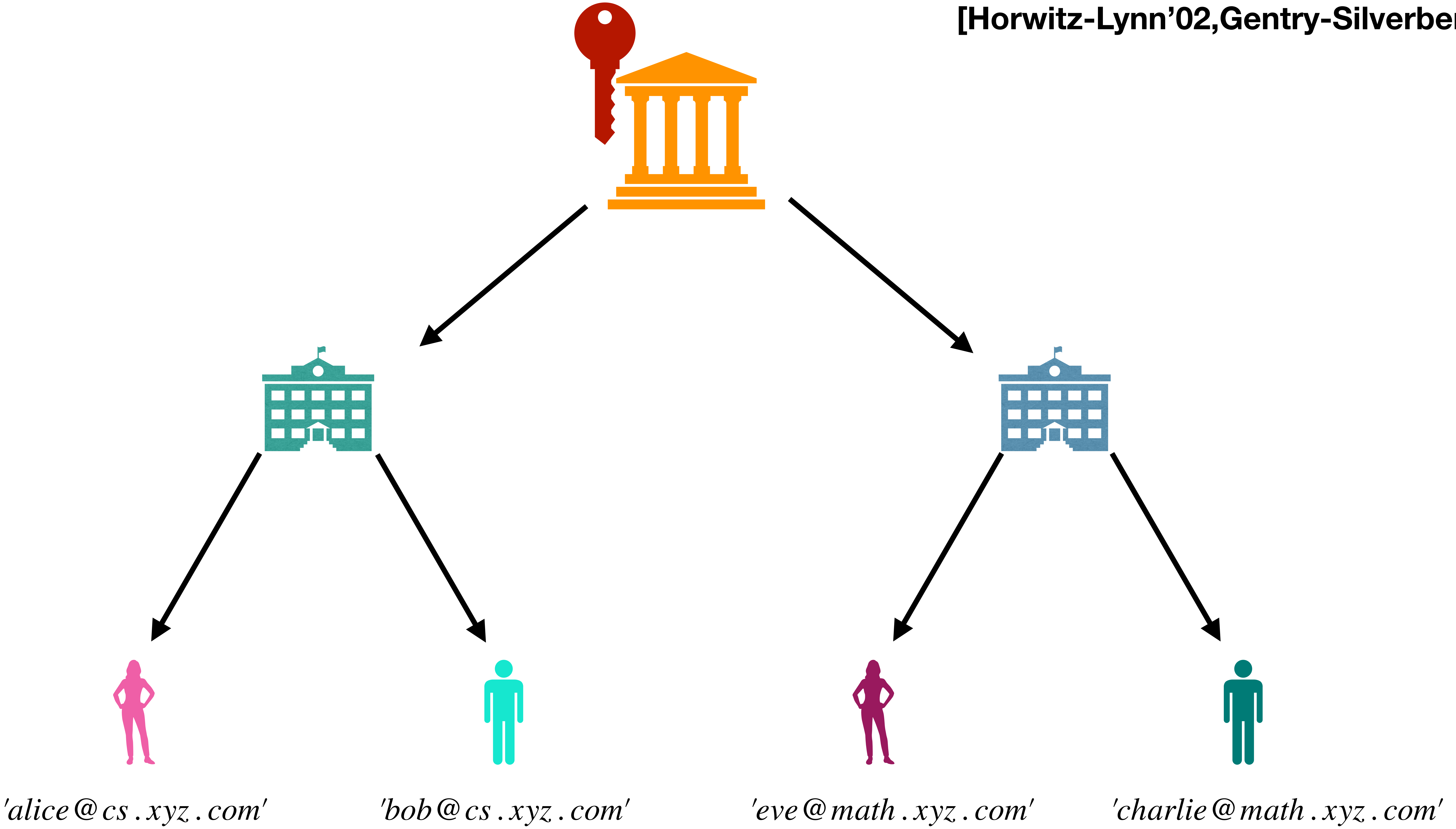
Hierarchical Identity Based Encryption

[Horwitz-Lynn'02,Gentry-Silverberg'02]



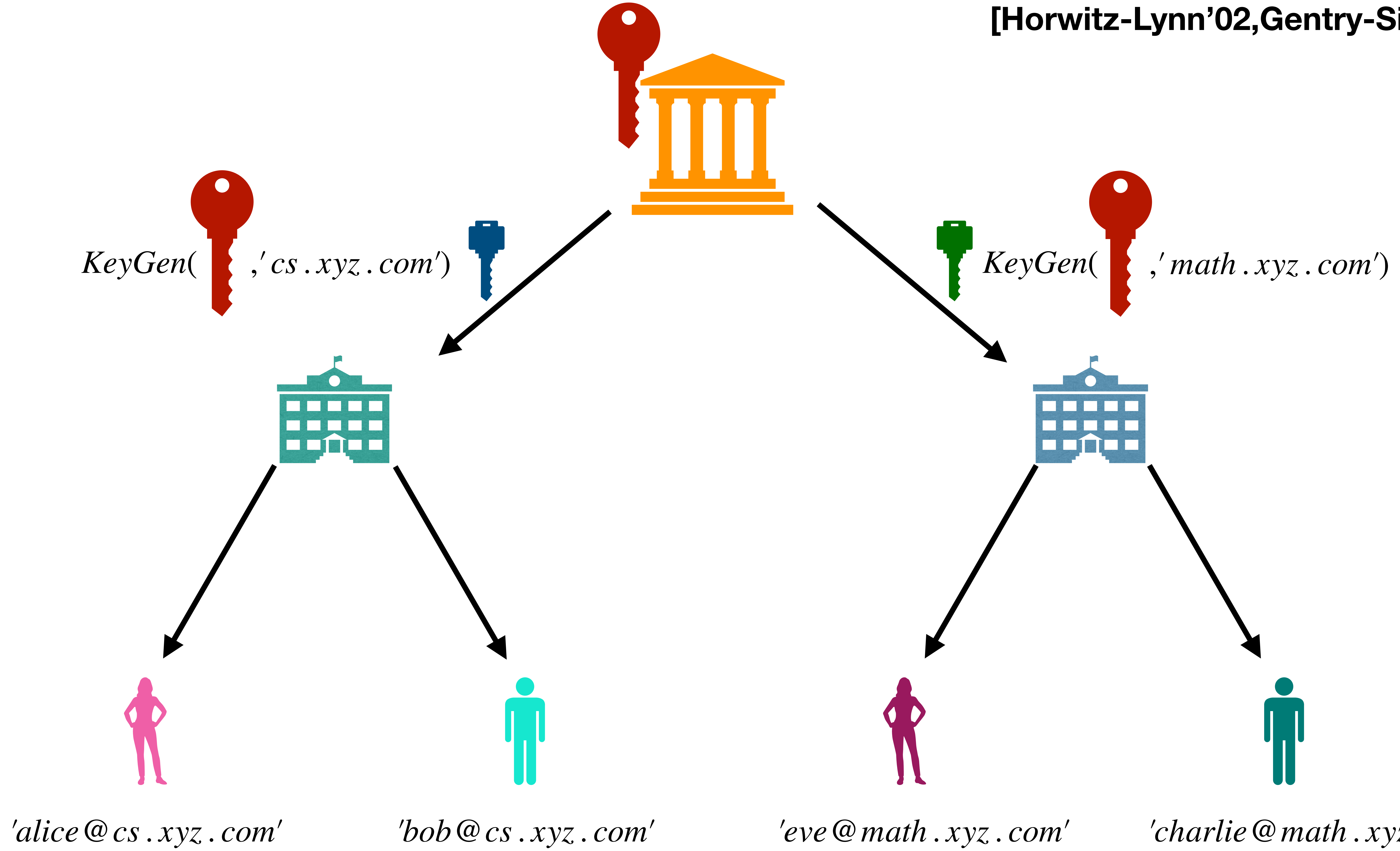
Hierarchical Identity Based Encryption

[Horwitz-Lynn'02,Gentry-Silverberg'02]



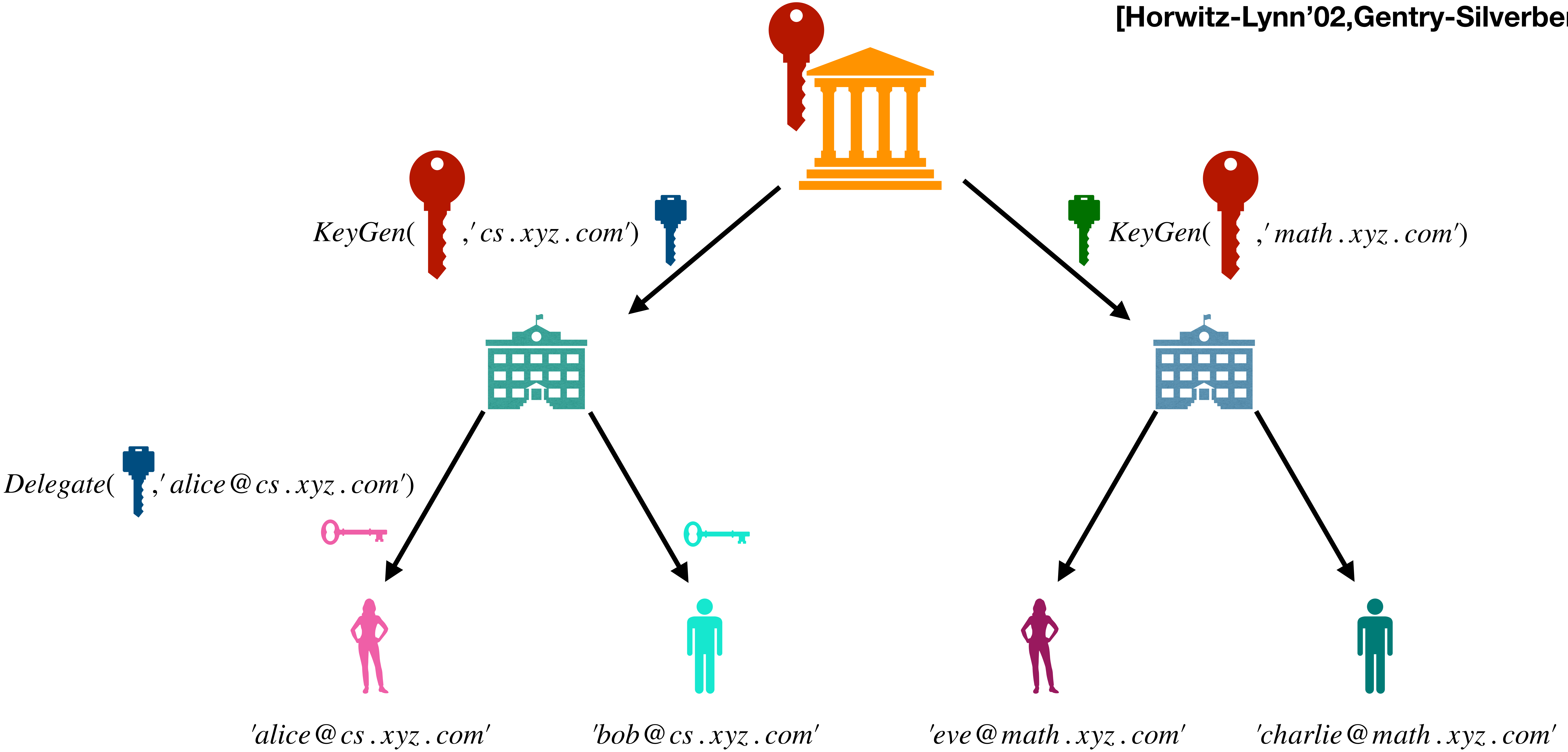
Hierarchical Identity Based Encryption

[Horwitz-Lynn'02,Gentry-Silverberg'02]



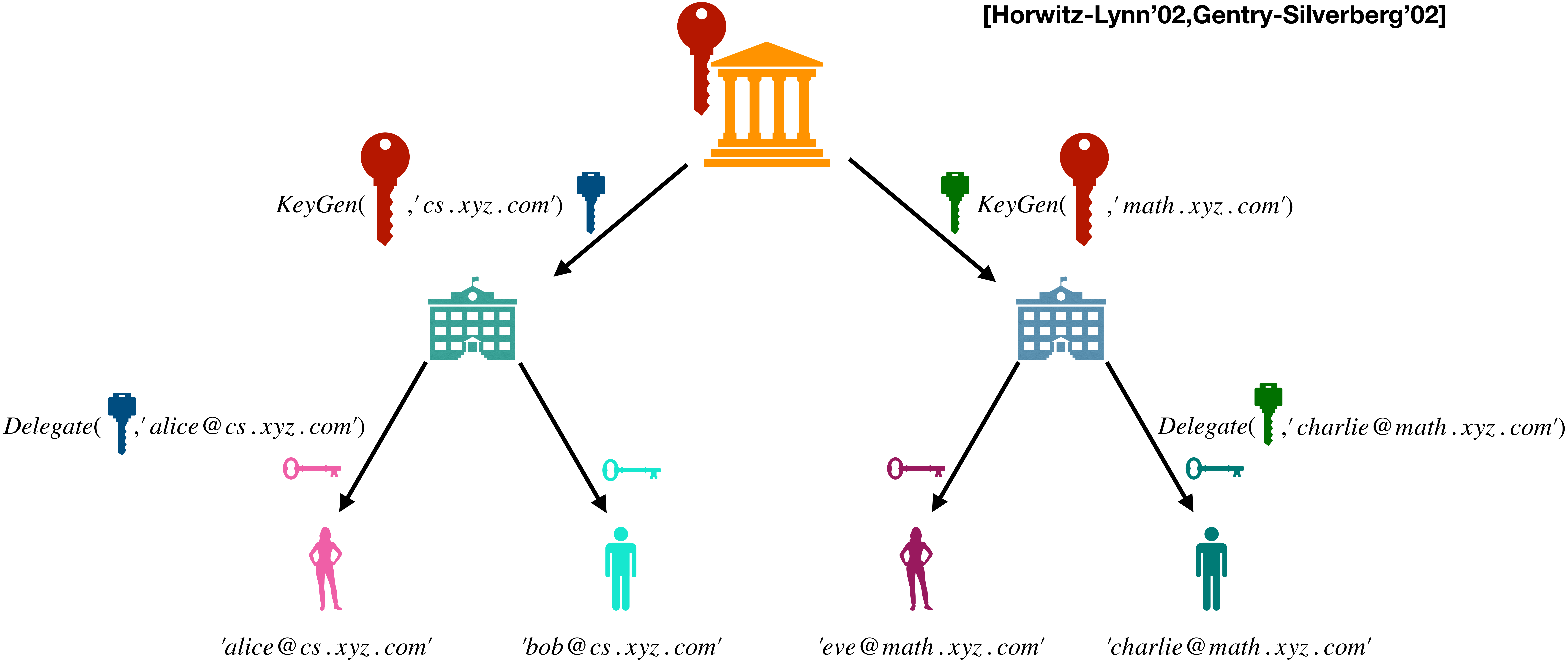
Hierarchical Identity Based Encryption

[Horwitz-Lynn’02,Gentry-Silverberg’02]



Hierarchical Identity Based Encryption

[Horwitz-Lynn'02,Gentry-Silverberg'02]



Syntax

Syntax

$(mpk, msk) \leftarrow Setup(1^\lambda)$

Syntax

$(mpk, msk) \leftarrow Setup(1^\lambda)$ $sk_{id} \leftarrow KeyGen(msk, id)$

Syntax

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$sk_{id} \leftarrow KeyGen(msk, id)$

$ct \leftarrow Enc(mpk, id, m)$

Syntax

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$sk_{id} \leftarrow KeyGen(msk, id)$

$ct \leftarrow Enc(mpk, id, m)$

$m \leftarrow Dec(sk_{id}, ct)$

Syntax

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$sk_{id} \leftarrow KeyGen(msk, id)$

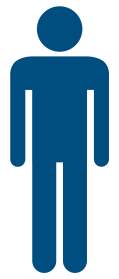
$sk_{id\|id'} \leftarrow Delegate(sk_{id}, id')$

$ct \leftarrow Enc(mpk, id, m)$

$m \leftarrow Dec(sk_{id}, ct)$

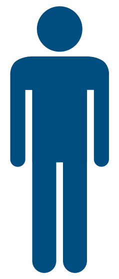
Security

Security



$(mpk, msk) \leftarrow Setup(1^{\lambda})$

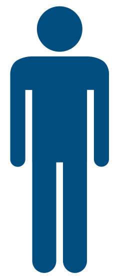
Security



$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad\quad mpk \quad\quad}$

Security

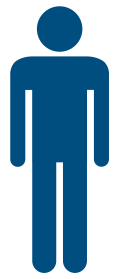


$(mpk, msk) \leftarrow Setup(1^{\lambda})$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad id \quad}$

Security



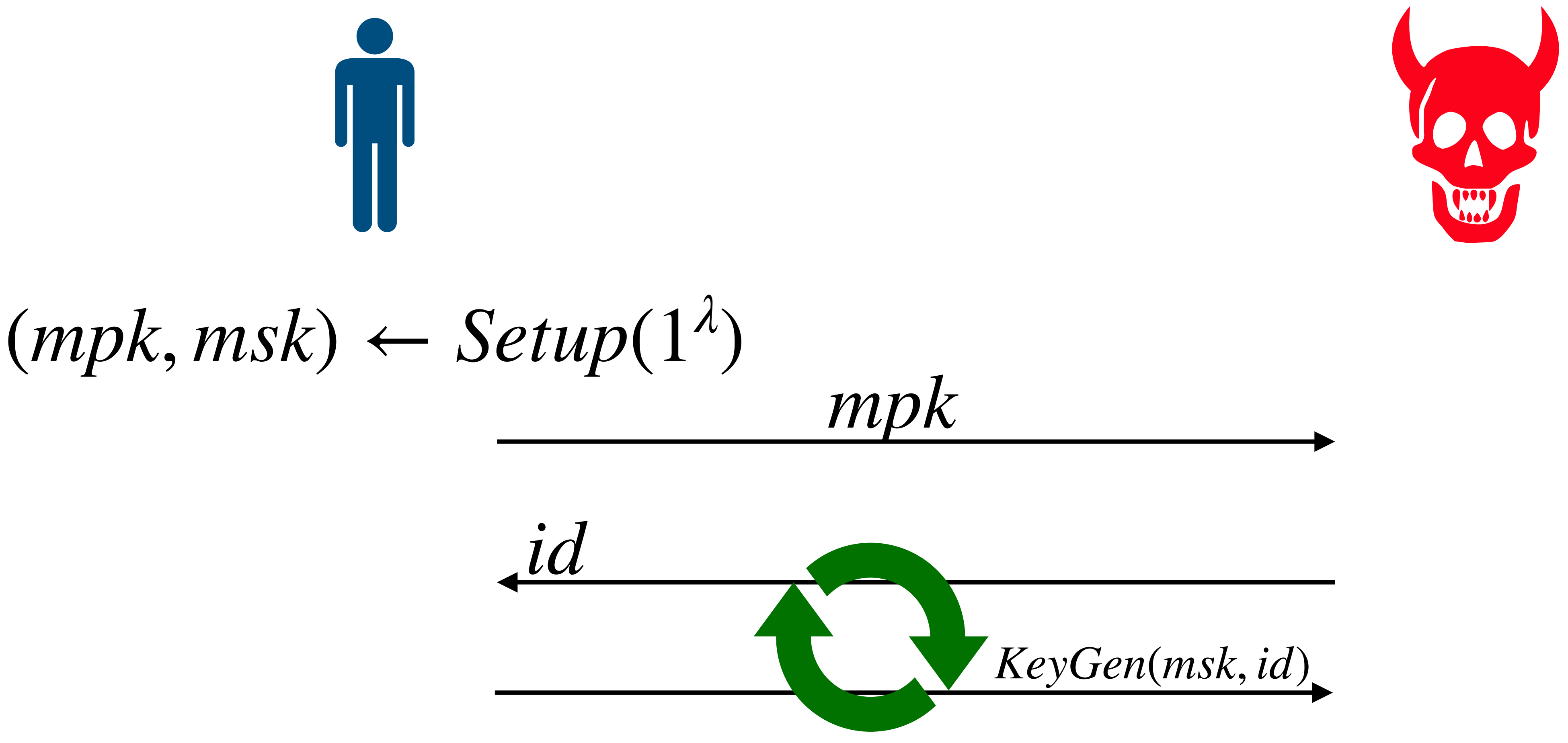
$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

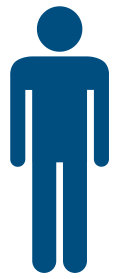
$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

Security



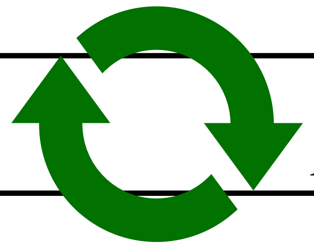
Security



$(mpk, msk) \leftarrow Setup(1^{\lambda})$

$\xrightarrow{\quad mpk \quad}$

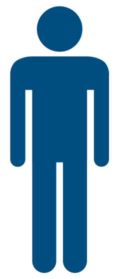
$\xleftarrow{\quad id \quad}$



$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$id \npreceq id^*$

Security



$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

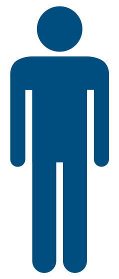
$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$id \not\preceq id^*$

Security



$(mpk, msk) \leftarrow Setup(1^{\lambda})$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

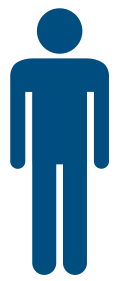
$b \leftarrow \{0,1\}$

$id \not\preceq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

Security



$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$id \npreceq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

Security

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$id \not\leq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

Security

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$id \not\leq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

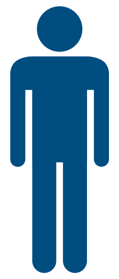
$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

Adversary wins if $b' = b$

Security



- Weak definition - No delegation!

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$id \not\leq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

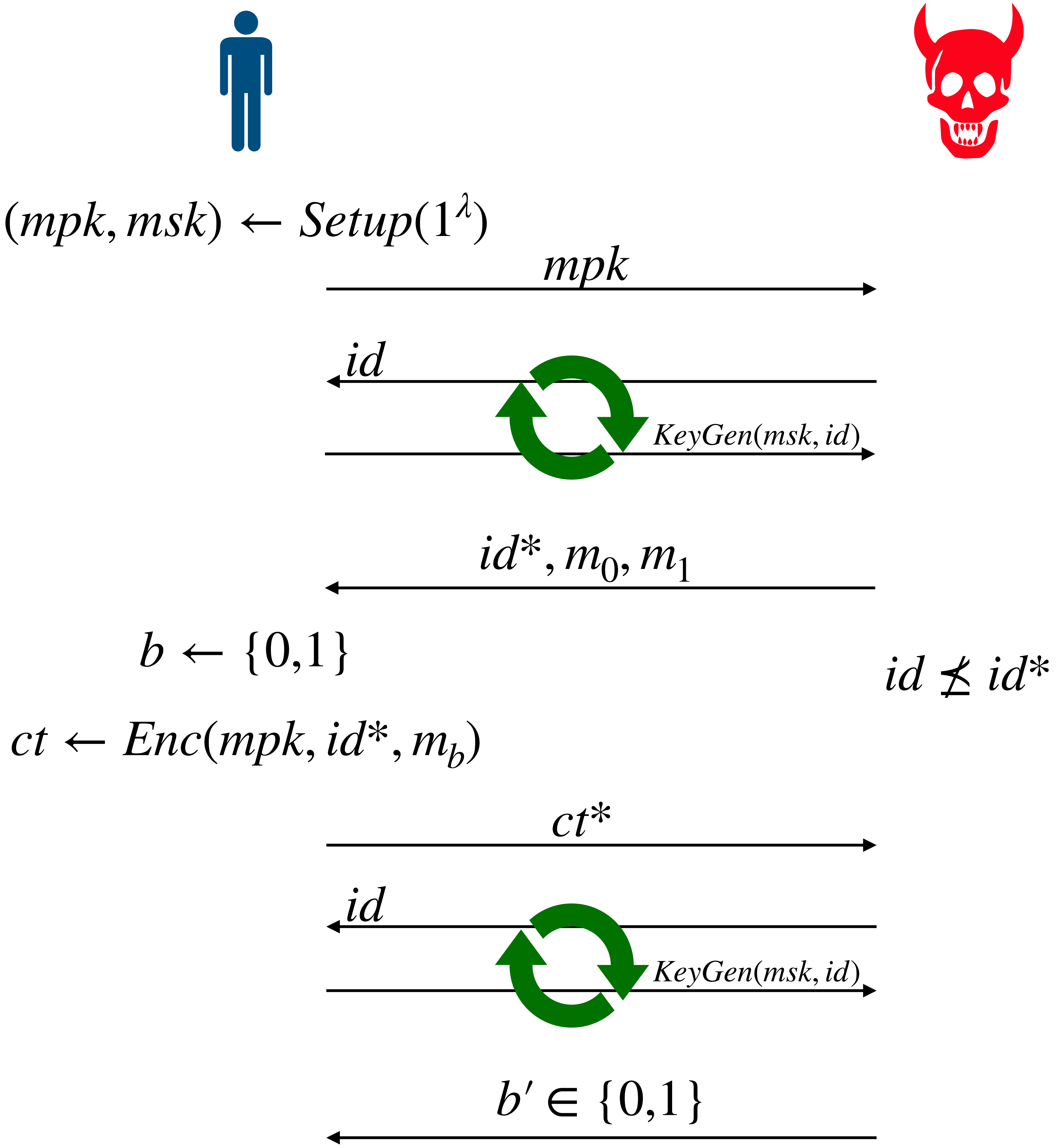
$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

Adversary wins if $b' = b$

Security



Adversary wins if $b' = b$

- Weak definition - No delegation!
- Achieved from adaptive IBE

Security

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$id \npreceq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

$\xleftarrow{\quad id \quad}$

$\xrightarrow{\quad KeyGen(msk, id) \quad}$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

Adversary wins if $b' = b$

- Weak definition - No delegation!
- Achieved from adaptive IBE
  - SETUP : $\{mpk_i, msk_i\} \leftarrow IBE.Setup(1^\lambda)$

Security

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{mpk}$

$\xleftarrow{id}$

$\xrightarrow{KeyGen(msk, id)}$

$\xleftarrow{id^*, m_0, m_1}$

$b \leftarrow \{0,1\}$

$id \npreceq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{ct^*}$

$\xleftarrow{id}$

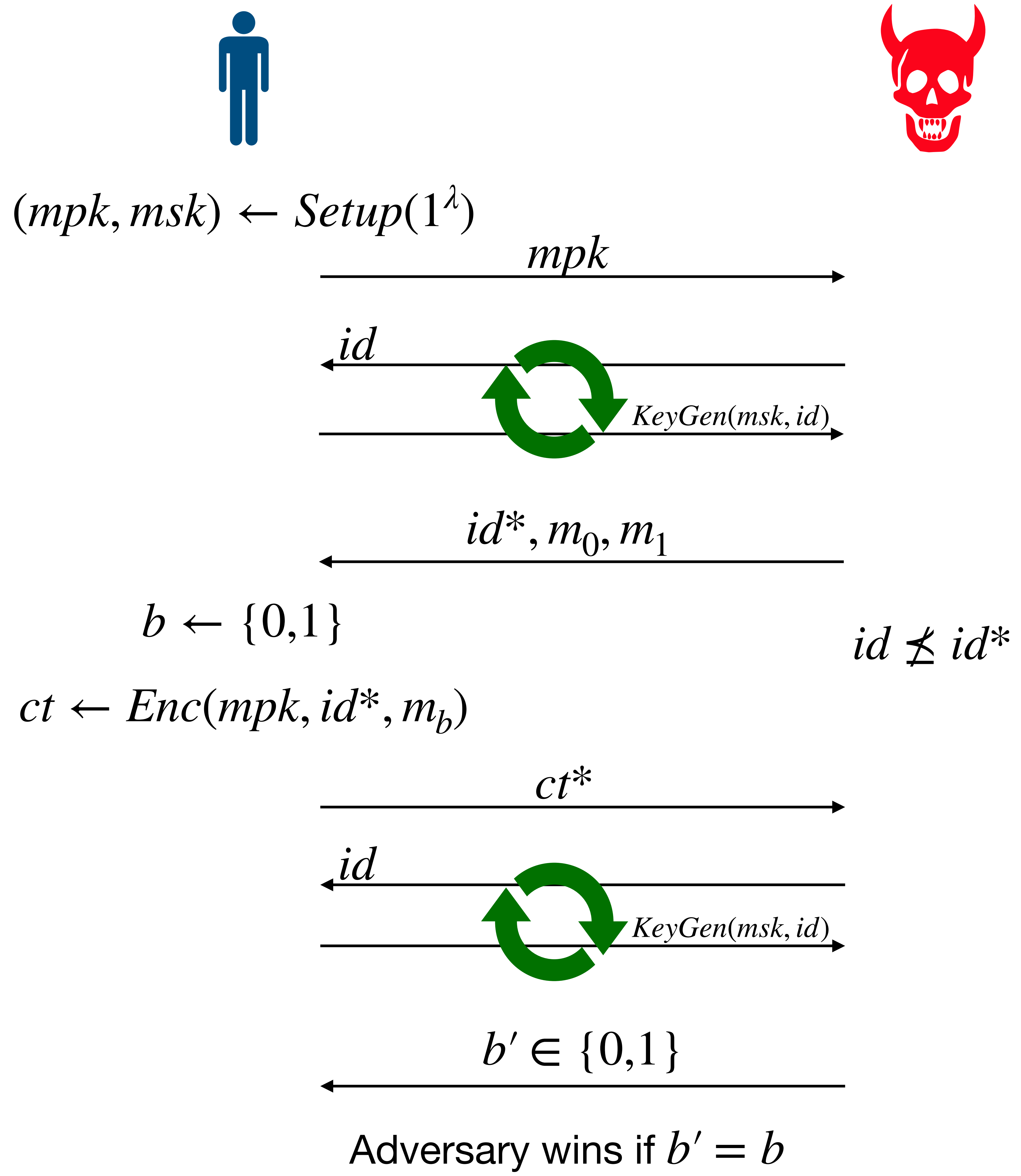
$\xrightarrow{KeyGen(msk, id)}$

$\xleftarrow{b' \in \{0,1\}}$

Adversary wins if $b' = b$

- Weak definition - No delegation!
- Achieved from adaptive IBE
  - SETUP : $\{mpk_i, msk_i\} \leftarrow IBE.Setup(1^\lambda)$
  - KEYGEN : $sk_{id} \leftarrow IBE.KeyGen(msk_{|id|}, id)$

Security



- Weak definition - No delegation!
- Achieved from adaptive IBE
    - SETUP : $\{mpk_i, msk_i\} \leftarrow IBE.Setup(1^\lambda)$
    - KEYGEN : $sk_{id} \leftarrow IBE.KeyGen(msk_{|id|}, id)$
    - ENC : $\{ct_i \leftarrow IBE.Enc(mpk_i, id[1;i], m)\}_{i \in [|id|]}$

Security

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{mpk}$

$\xleftarrow{id}$

$\xrightarrow{KeyGen(msk, id)}$

$\xleftarrow{id^*, m_0, m_1}$

$b \leftarrow \{0,1\}$

$id \npreceq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{ct^*}$

$\xleftarrow{id}$

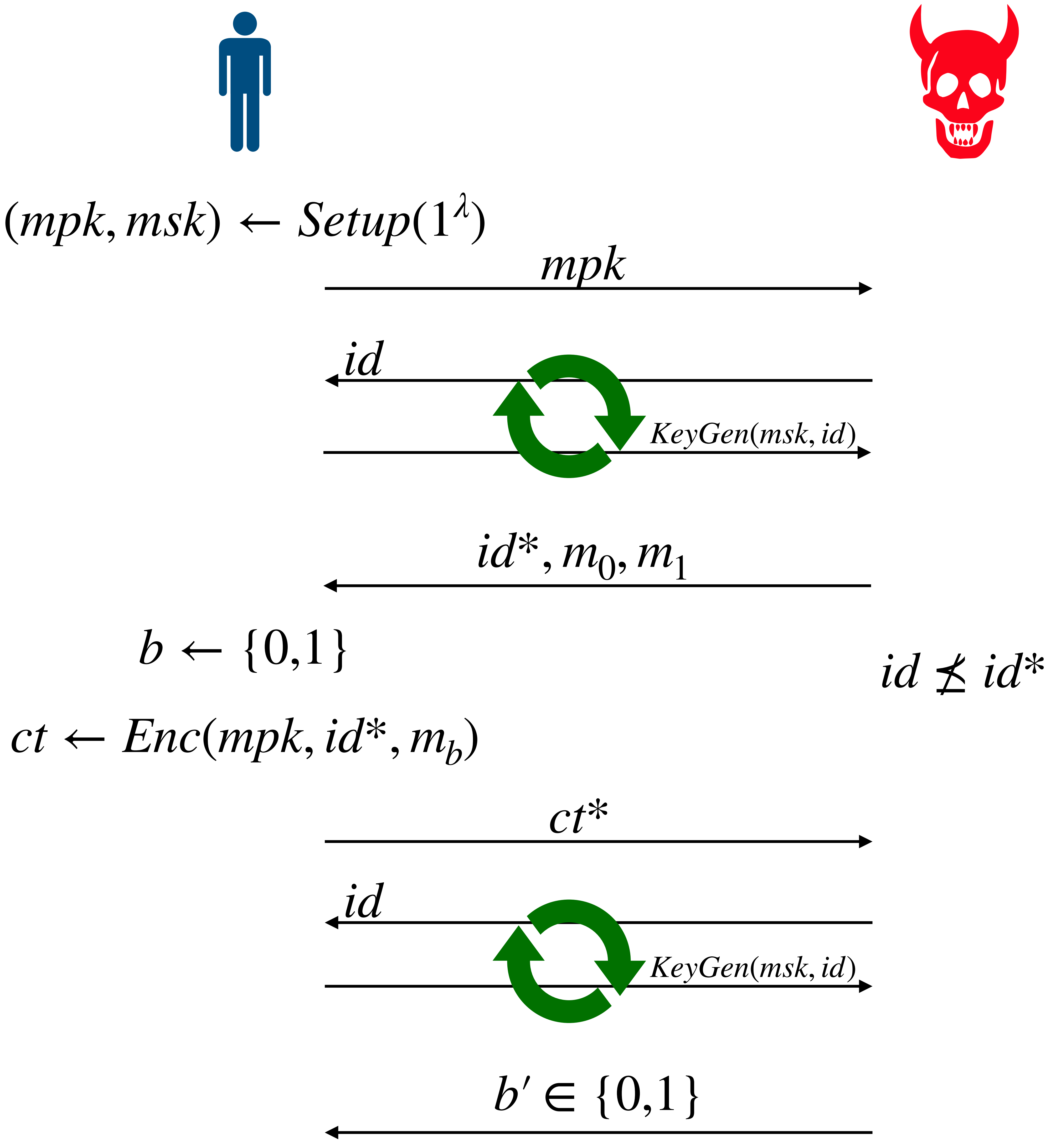
$\xrightarrow{KeyGen(msk, id)}$

$\xleftarrow{b' \in \{0,1\}}$

Adversary wins if $b' = b$

- Weak definition - No delegation!
- Achieved from adaptive IBE
  - SETUP : $\{mpk_i, msk_i\} \leftarrow IBE.Setup(1^\lambda)$
  - KEYGEN : $sk_{id} \leftarrow IBE.KeyGen(msk_{|id|}, id)$
  - ENC : $\{ct_i \leftarrow IBE.Enc(mpk_i, id[1;i], m)\}_{i \in [|id|]}$
  - DELEGATE : Return sk_{id}

Security

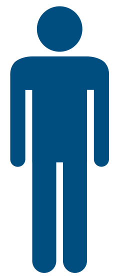


Adversary wins if $b' = b$

- Weak definition - No delegation!
- Achieved from adaptive IBE
  - SETUP : $\{mpk_i, msk_i\} \leftarrow IBE\,.\,Setup(1^\lambda)$
  - KEYGEN : $sk_{id} \leftarrow IBE\,.\,KeyGen(msk_{|id|}, id)$
  - ENC : $\{ct_i \leftarrow IBE\,.\,Enc(mpk_i, id[1;i], m)\}_{i \in [|id|]}$
  - DELEGATE : Return sk_{id}
- Immediately broken if adversary is allowed to obtain delegated keys!

Security

[Shi-Waters'08]



$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

Adversary wins if $b' = b$

Security

[Shi-Waters'08]

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

query $\longleftarrow$

response $\longrightarrow$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

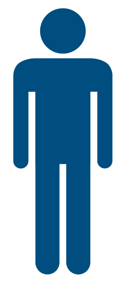
query $\longleftarrow$

response $\longrightarrow$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

Adversary wins if $b' = b$

Security



[Shi-Waters'08]

$(mpk, msk) \leftarrow Setup(1^\lambda)$

mpk →

← *query*

response →

← id^*, m_0, m_1

$b \leftarrow \{0,1\}$

$ct \leftarrow Enc(mpk, id^*, m_b)$

ct^* →

← *query*

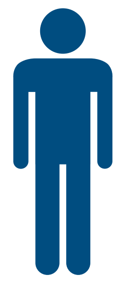
response →

← $b' \in \{0,1\}$

Adversary wins if $b' = b$

- $query = initialize(id):$
  Store $sk_{id} \leftarrow KeyGen(msk, id)$.
  No response.

Security



[Shi-Waters'08]

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad query \quad}$

$\xrightarrow{\quad response \quad}$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

$\xleftarrow{\quad query \quad}$

$\xrightarrow{\quad response \quad}$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

Adversary wins if $b' = b$

- $query = initialize(id)$:
  Store $sk_{id} \leftarrow KeyGen(msk, id)$.
  No response.

- $query = delegate(id_1, id_2)$:
  Store $sk_{id_1 \| id_2} \leftarrow Delegate(sk_{id_1}, id_2)$.
  No response.

Security

[Shi-Waters'08]

$(mpk, msk) \leftarrow Setup(1^\lambda)$

mpk →

← *query*

response →

← id^*, m_0, m_1

$b \leftarrow \{0,1\}$

$ct \leftarrow Enc(mpk, id^*, m_b)$

ct^* →

← *query*

response →

← $b' \in \{0,1\}$

Adversary wins if $b' = b$

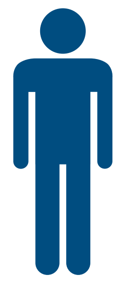
- $query = initialize(id)$:
  Store $sk_{id} \leftarrow KeyGen(msk, id)$.
  No response.

- $query = delegate(id_1, id_2)$:
  Store $sk_{id_1 \| id_2} \leftarrow Delegate(sk_{id_1}, id_2)$.
  No response.

- $query = corrupt(id)$:
  Returns sk_{id}

Security

[Shi-Waters'08]



$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$query$

$response$

$\xleftarrow{\quad id^*, m_0, m_1 \quad}$

$b \leftarrow \{0,1\}$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

$query$

$response$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

Adversary wins if $b' = b$

- $query = initialize(id)$:
  Store $sk_{id} \leftarrow KeyGen(msk, id)$.
  No response.

- $query = delegate(id_1, id_2)$:
  Store $sk_{id_1 \| id_2} \leftarrow Delegate(sk_{id_1}, id_2)$.
  No response.

- $query = corrupt(id)$:
  Returns sk_{id} $\qquad id \npreceq id^*$

Selective Security

$(mpk, msk) \leftarrow Setup(1^\lambda)$

$\xrightarrow{\quad mpk \quad}$

$\xleftarrow{\quad query \quad}$

$\xrightarrow{\quad response \quad}$

$\longleftarrow$

$b \leftarrow \{0,1\}$

$id \neq id^*$

$ct \leftarrow Enc(mpk, id^*, m_b)$

$\xrightarrow{\quad ct^* \quad}$

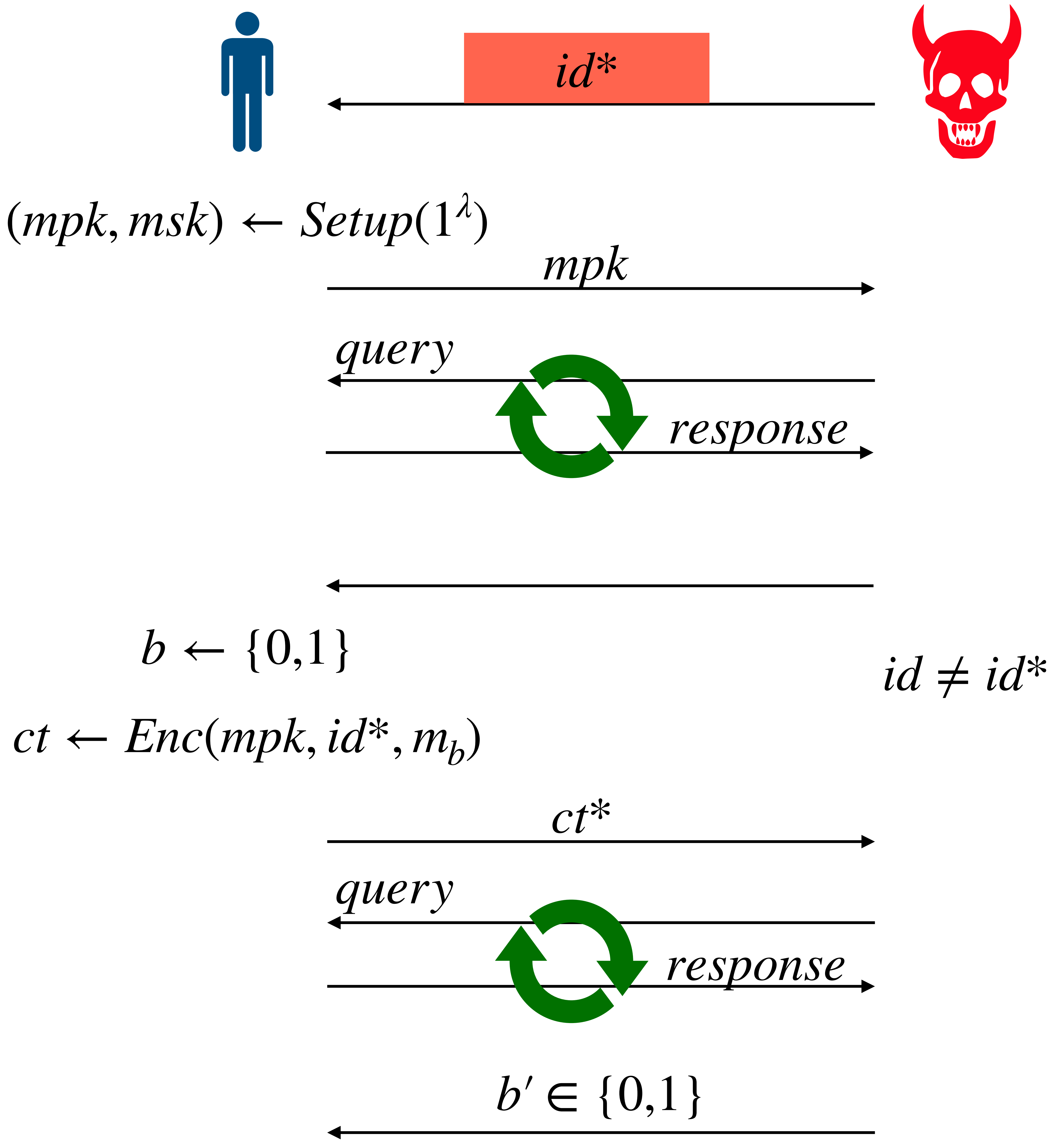
$\xleftarrow{\quad query \quad}$

$\xrightarrow{\quad response \quad}$

$\xleftarrow{\quad b' \in \{0,1\} \quad}$

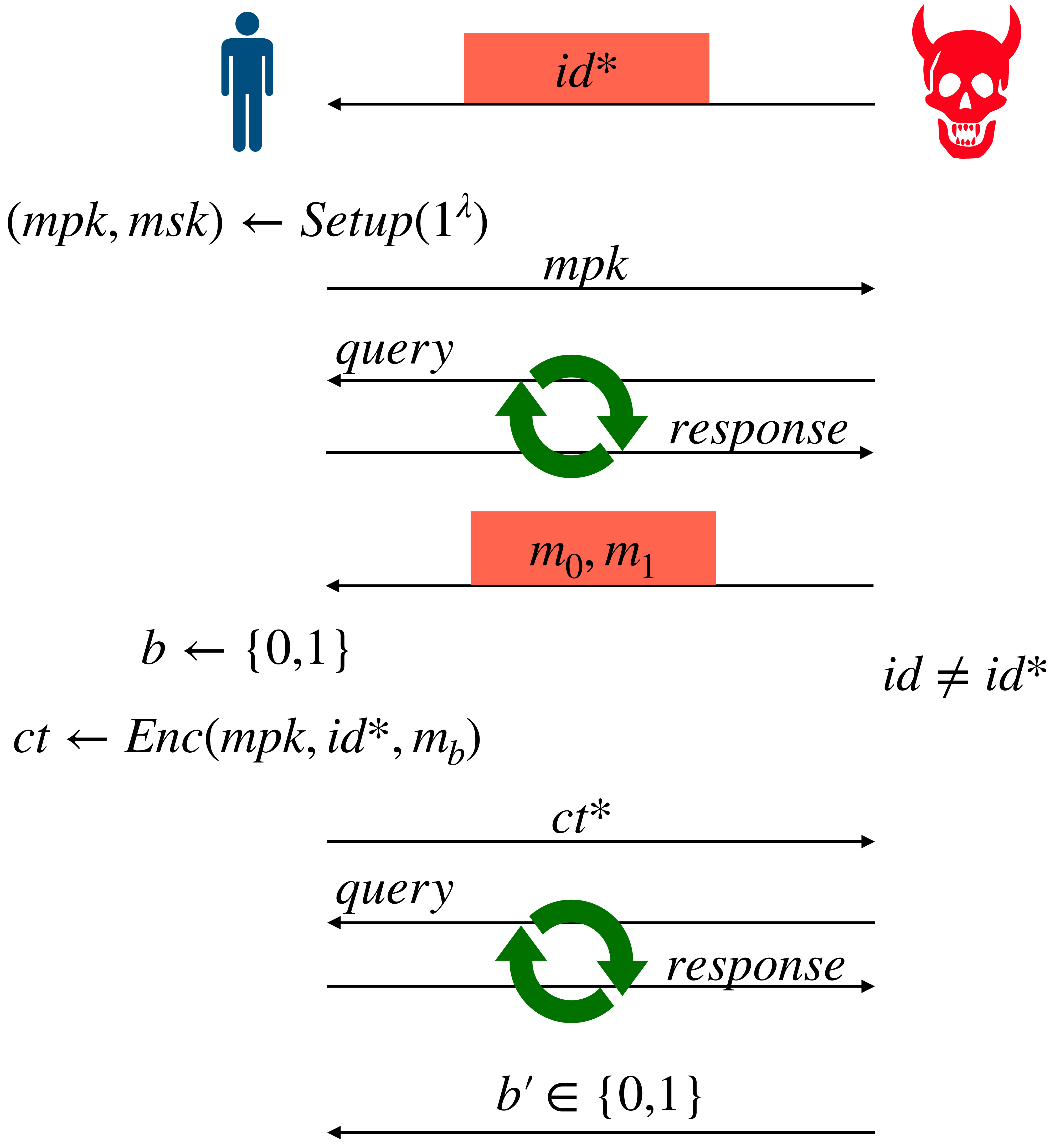
Adversary wins if $b' = b$

Selective Security



Adversary wins if $b' = b$

Selective Security

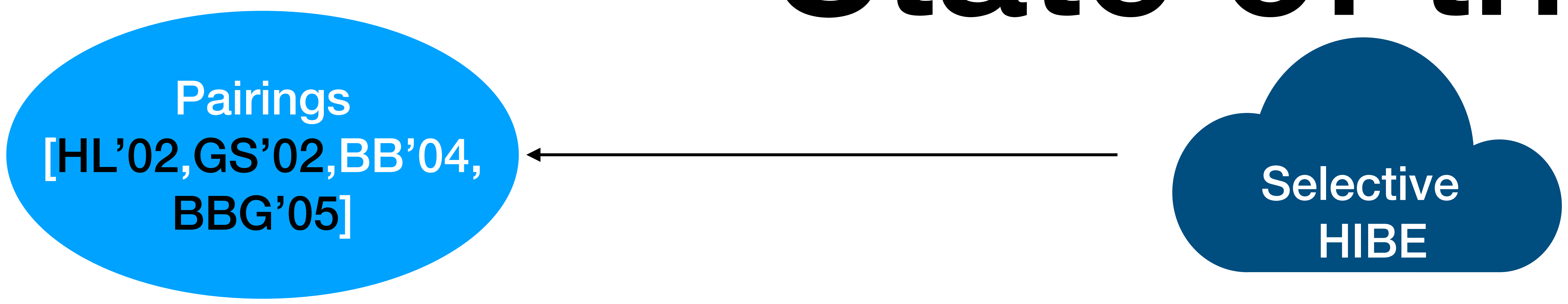


Adversary wins if $b' = b$

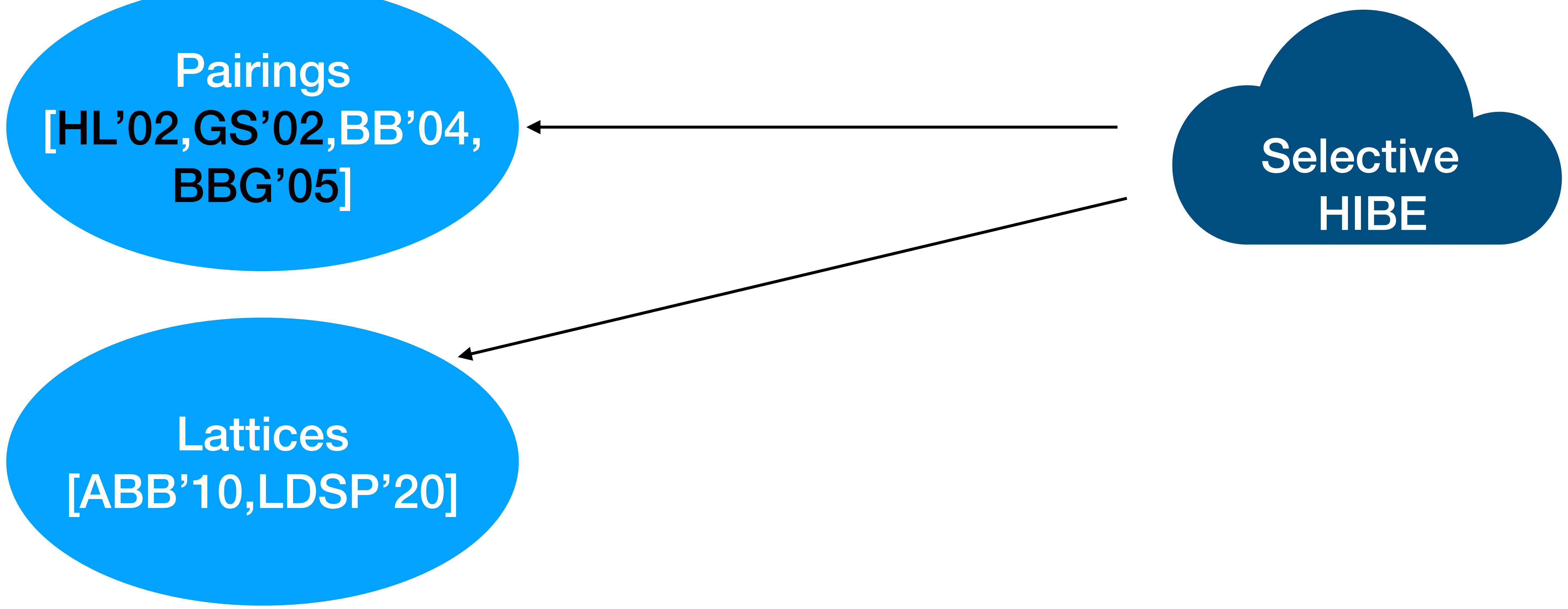
State of the Art

State of the Art

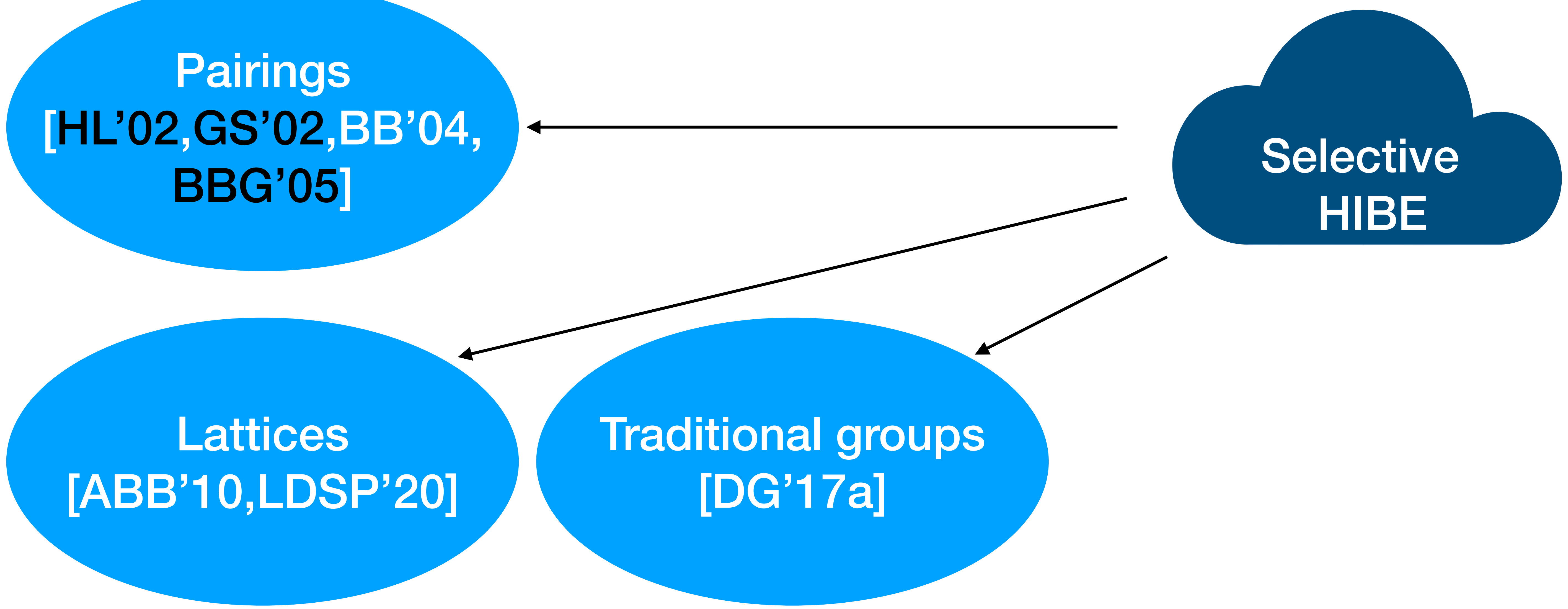
State of the Art



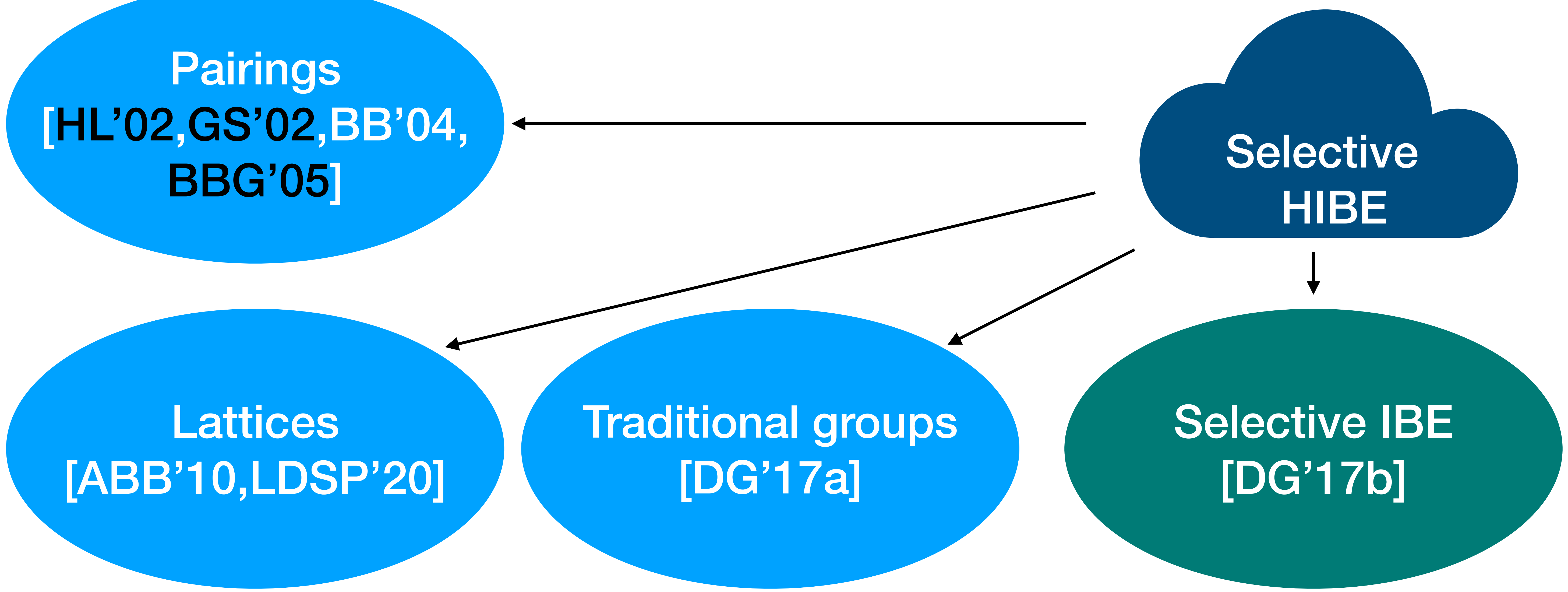
State of the Art



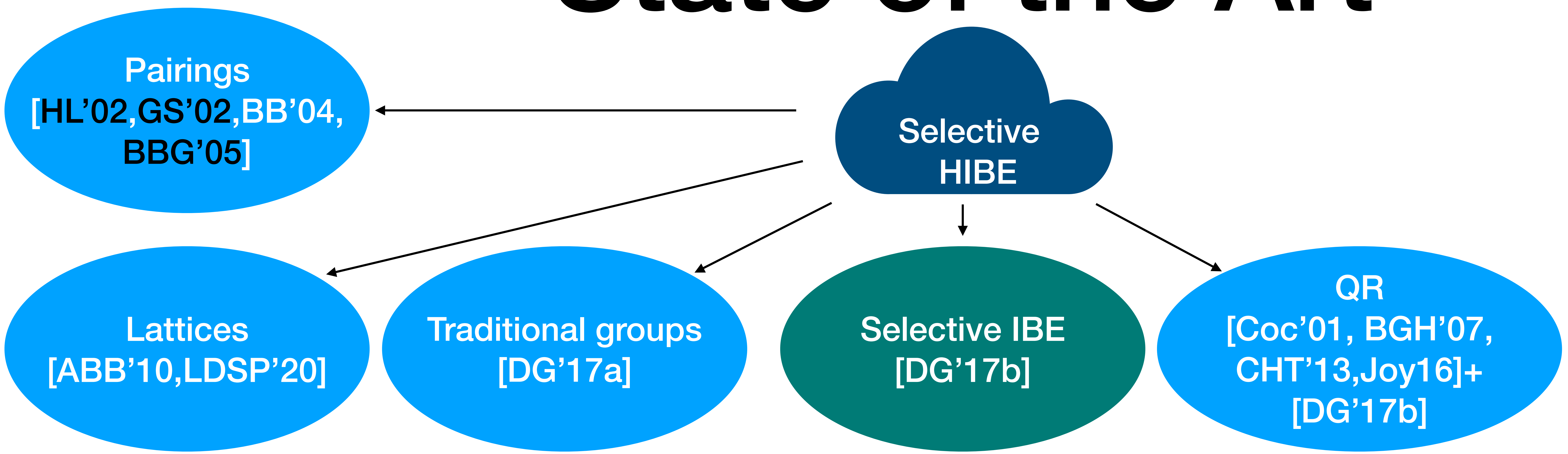
State of the Art



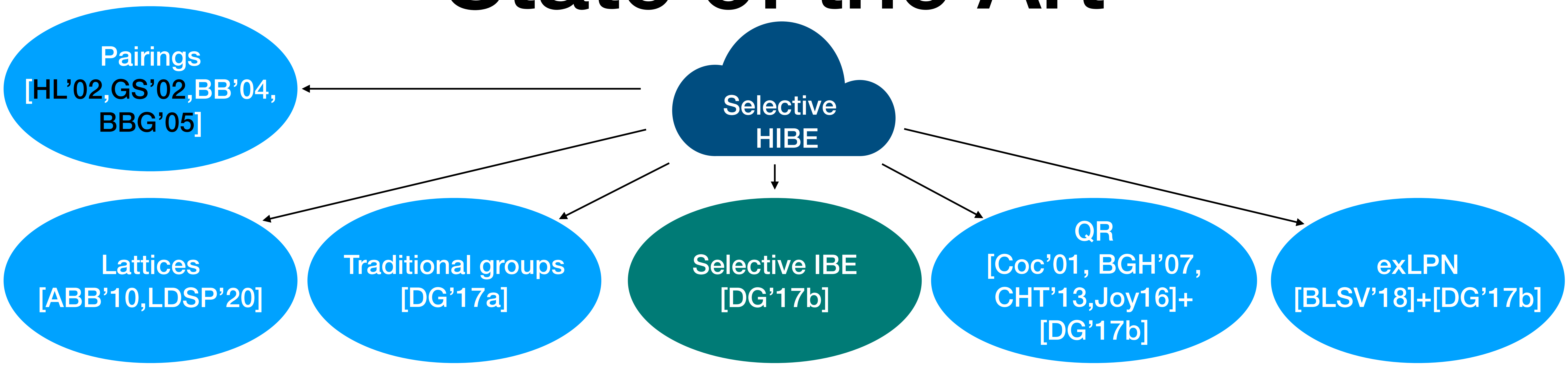
State of the Art



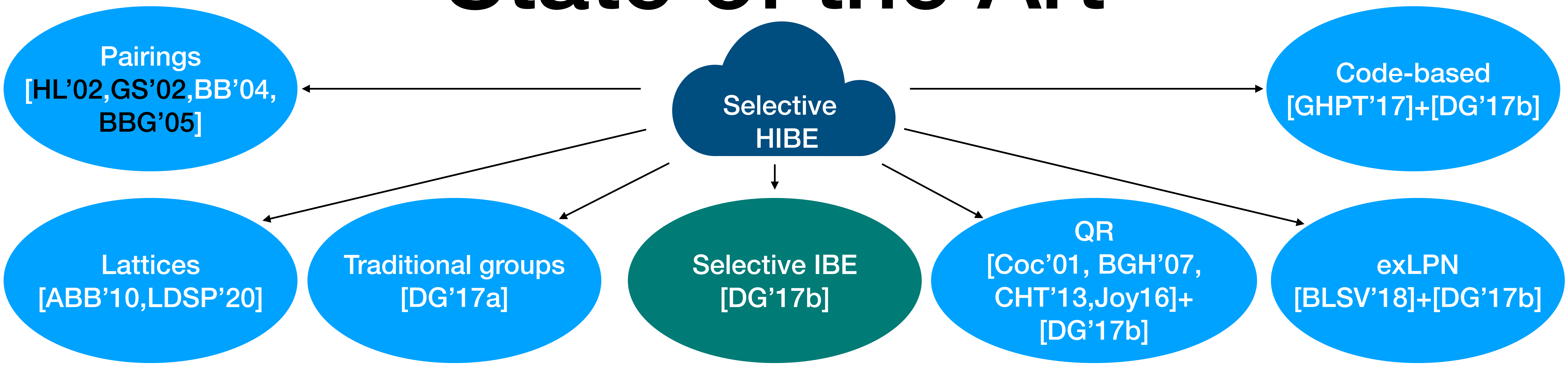
State of the Art



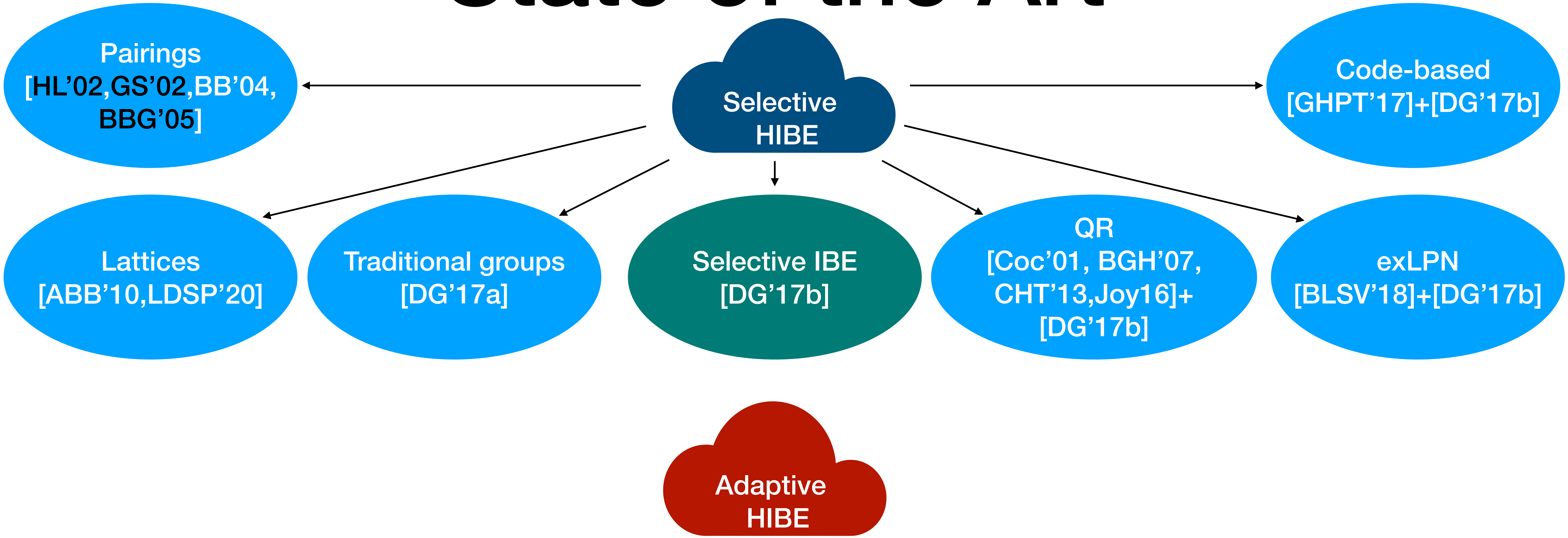
State of the Art



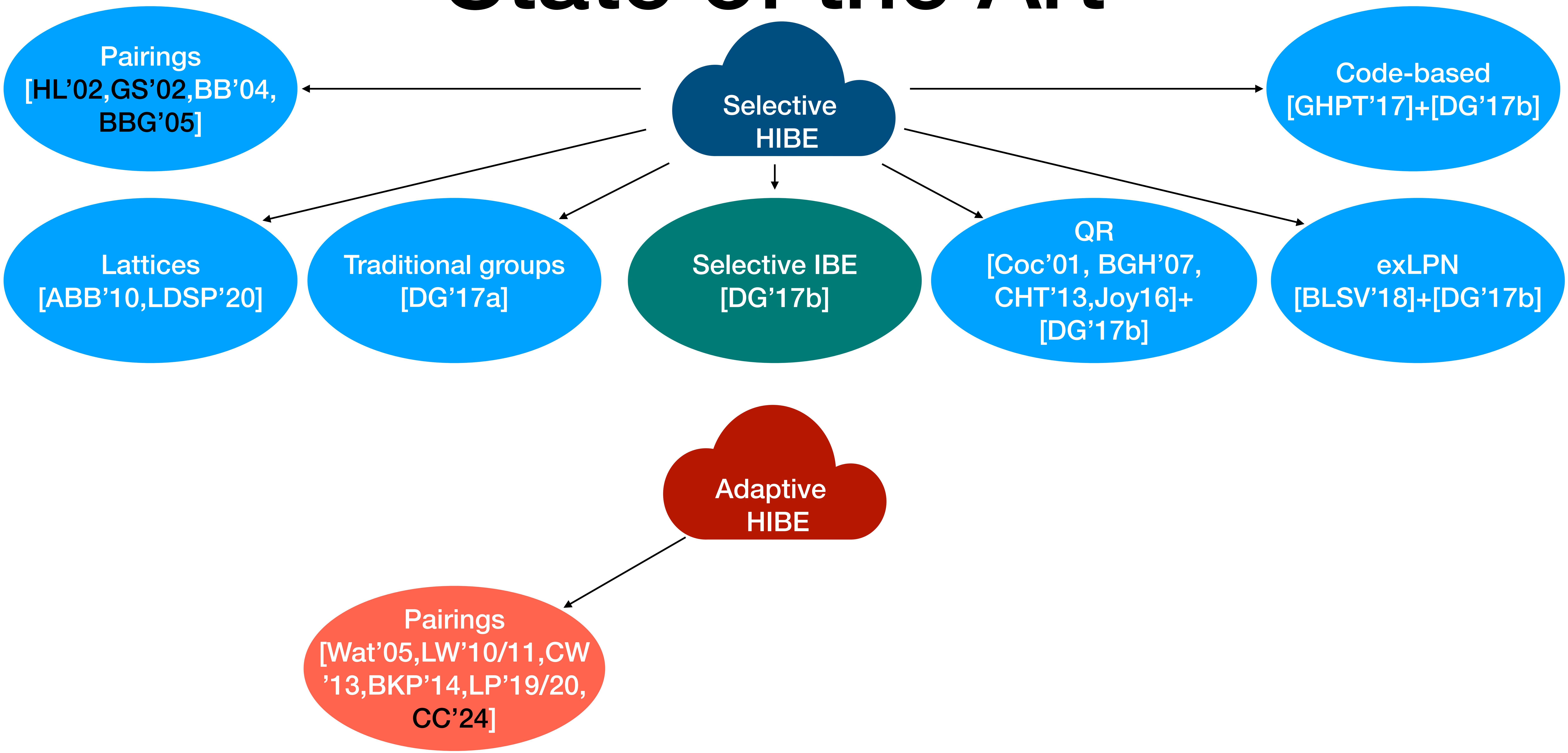
State of the Art



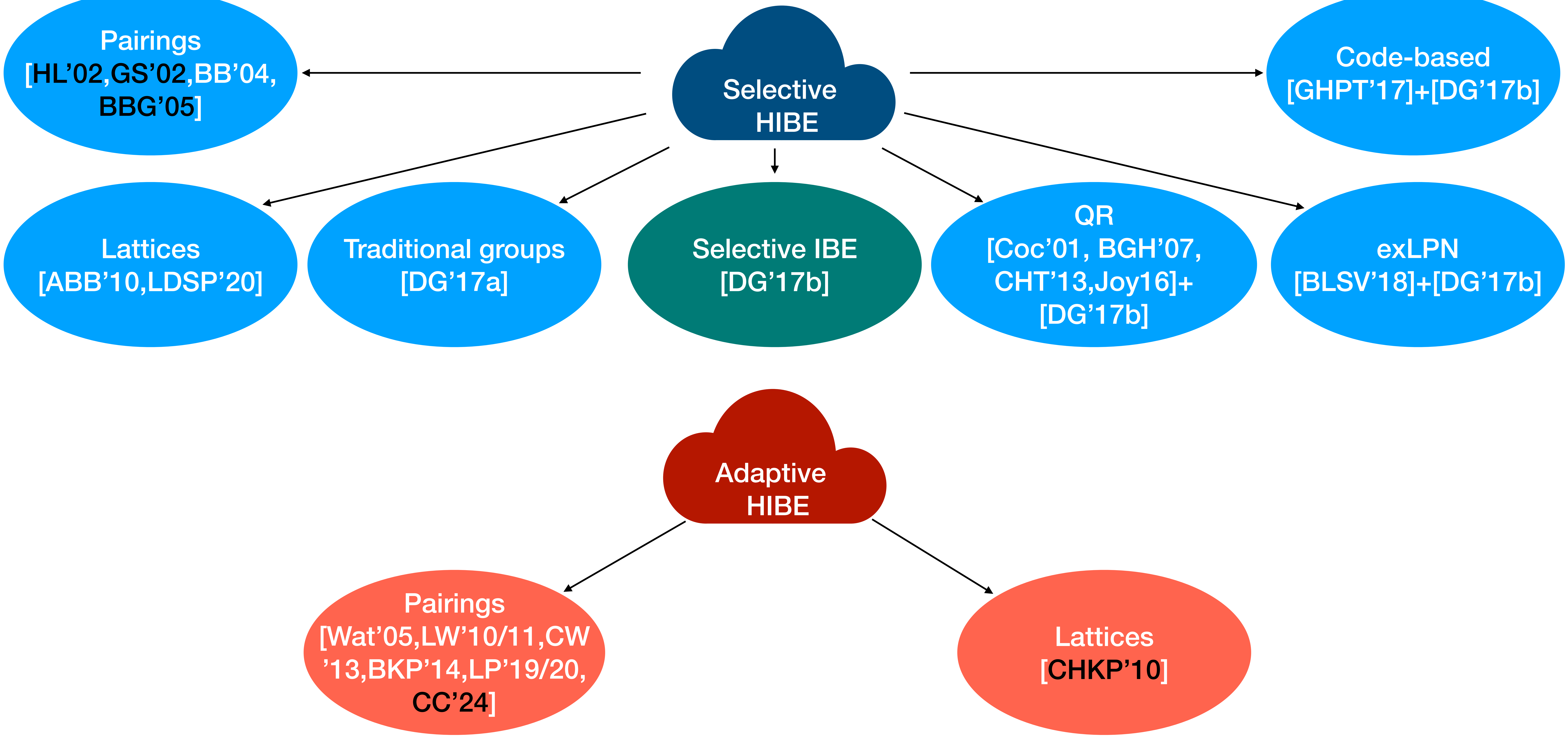
State of the Art



State of the Art



State of the Art



Challenges

Challenges

- Despite various heuristic approaches, achieving *adaptive secure* HIBE **in the standard model** without relying on strong or non-standard assumptions remains an open challenge.

Challenges

- Despite various heuristic approaches, achieving *adaptive secure* HIBE **in the standard model** without relying on strong or non-standard assumptions remains an open challenge.

- Lewko-Waters'14, Brakerski-Medina'24 demonstrate that HIBE schemes satisfying "checkable" or partition-based proof conditions (and many early constructions) cannot be adapted to full security.

Security

Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Security

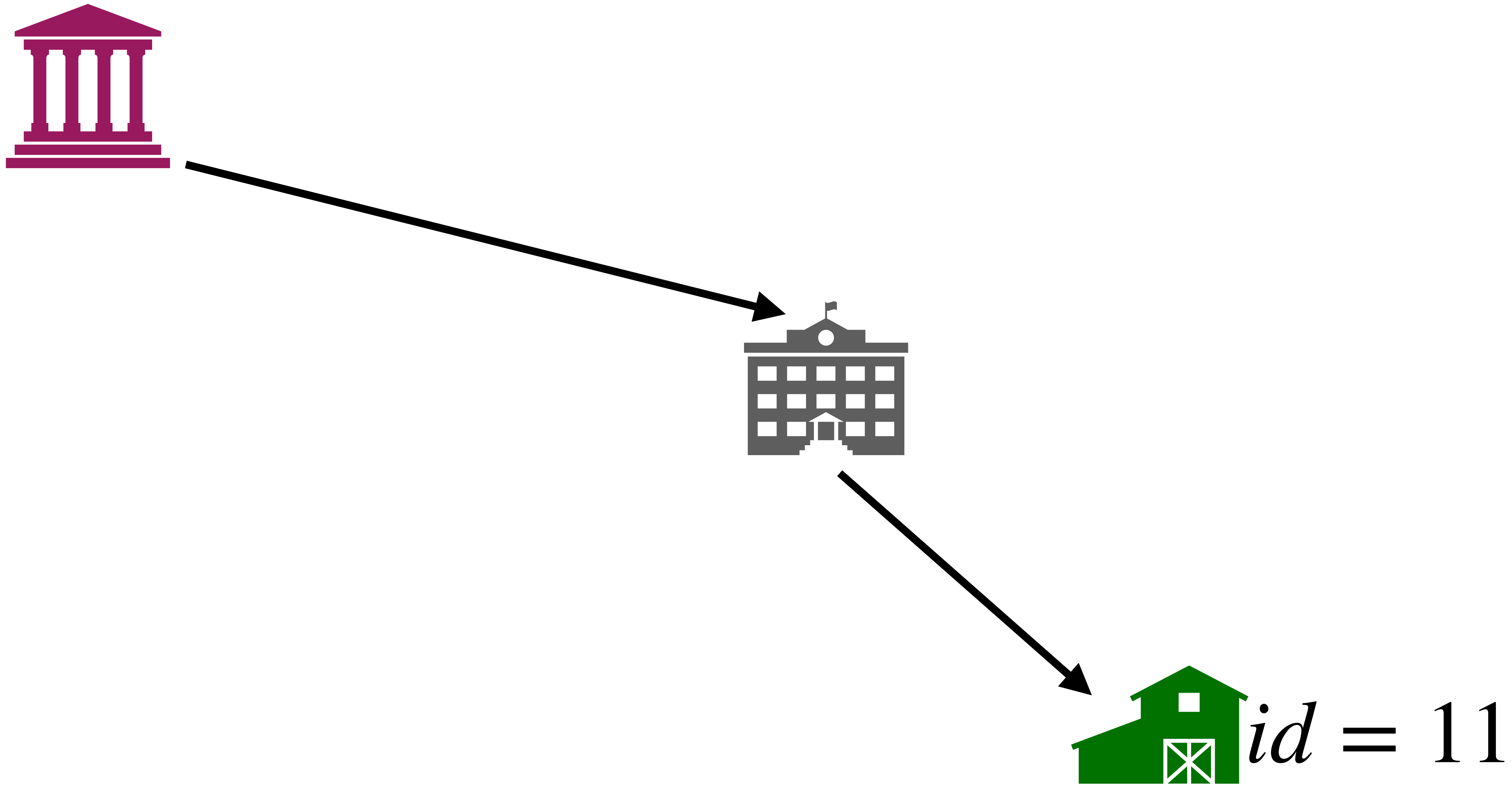
Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Security

Attacker

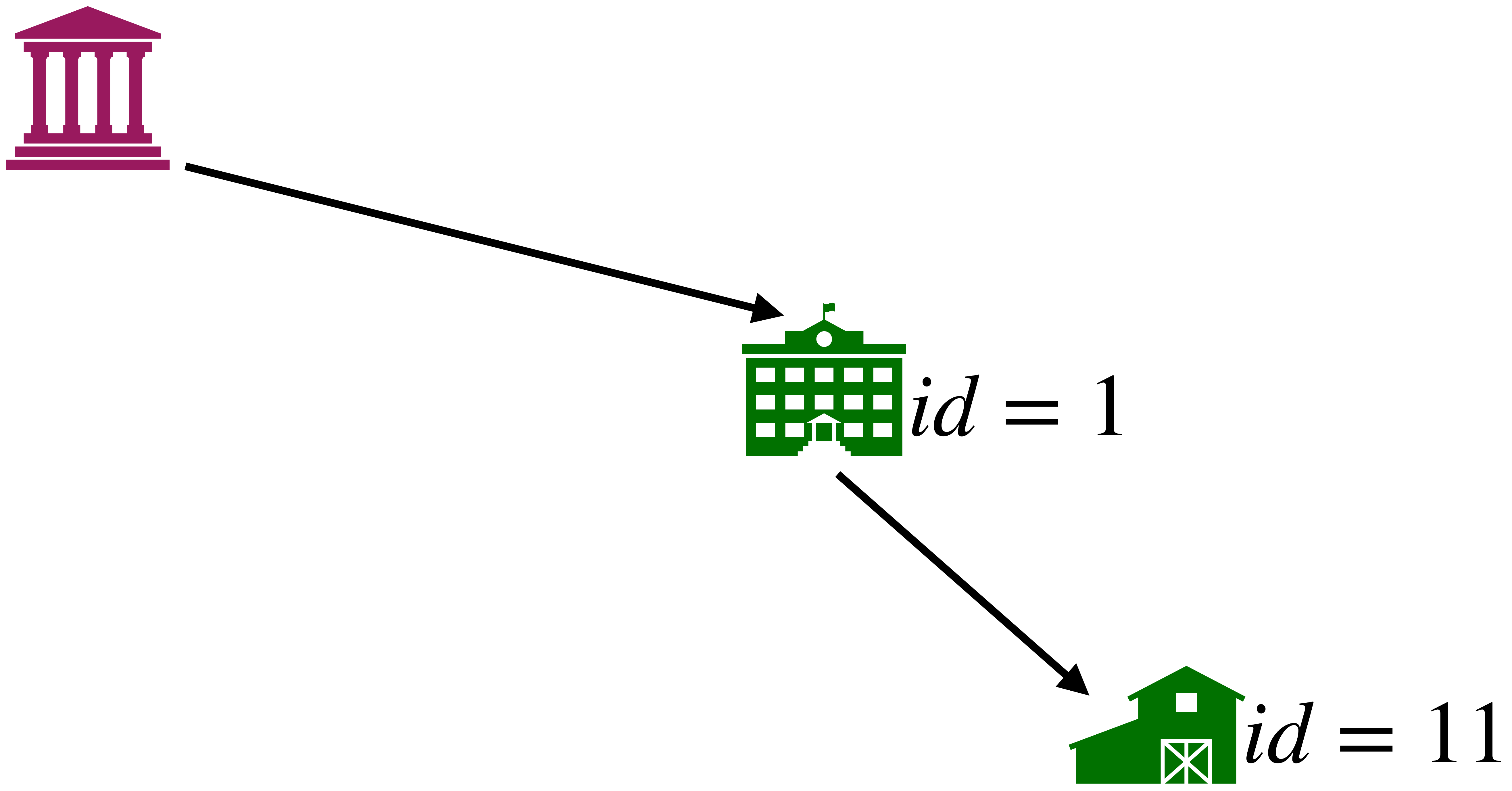
1. Obtains mpk
2. Adaptively makes queries
3. Sends $id*, m_0, m_1$
4. Obtains challenge ciphertext $ct*$
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$



Security

Attacker

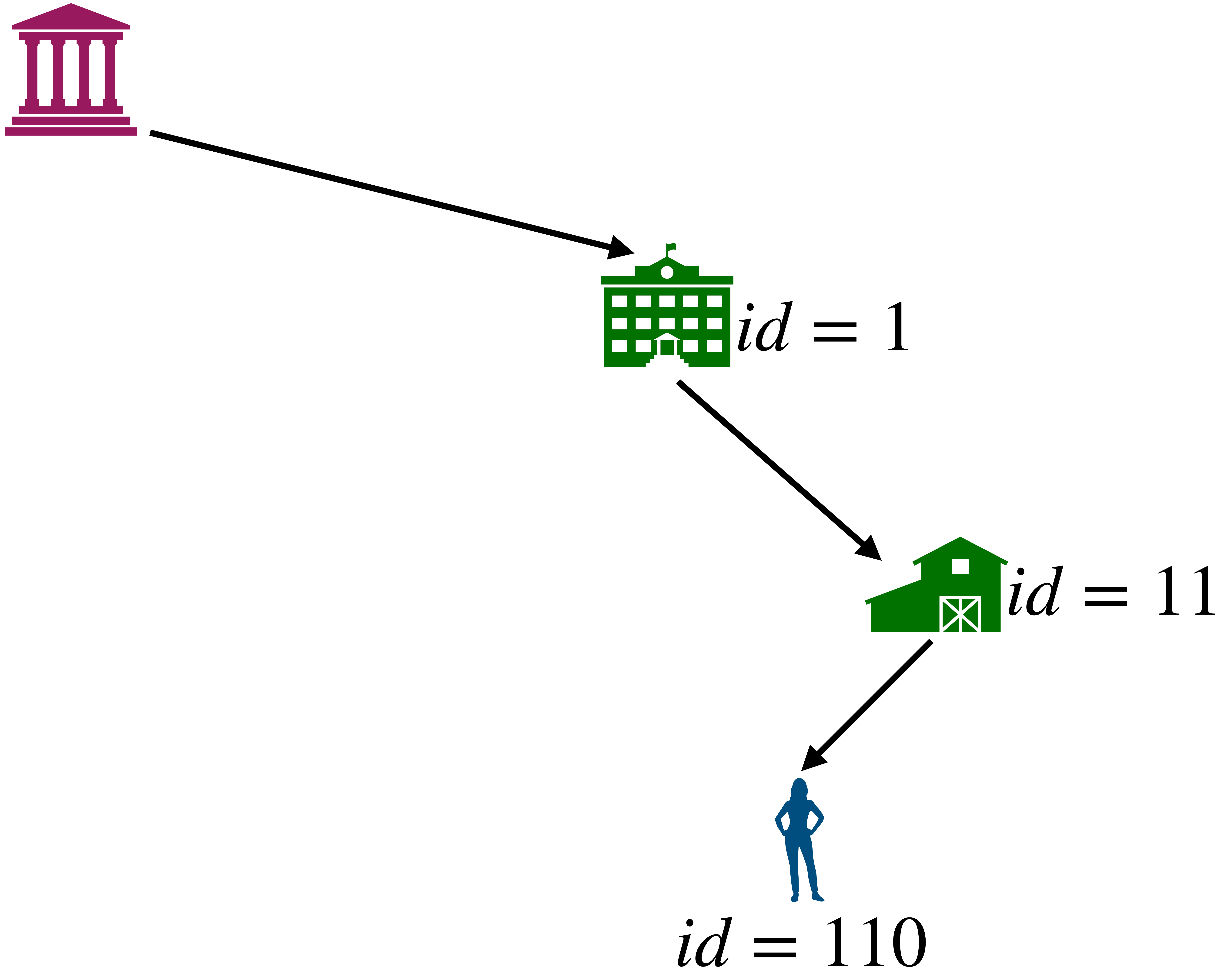
1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$



Security

Attacker

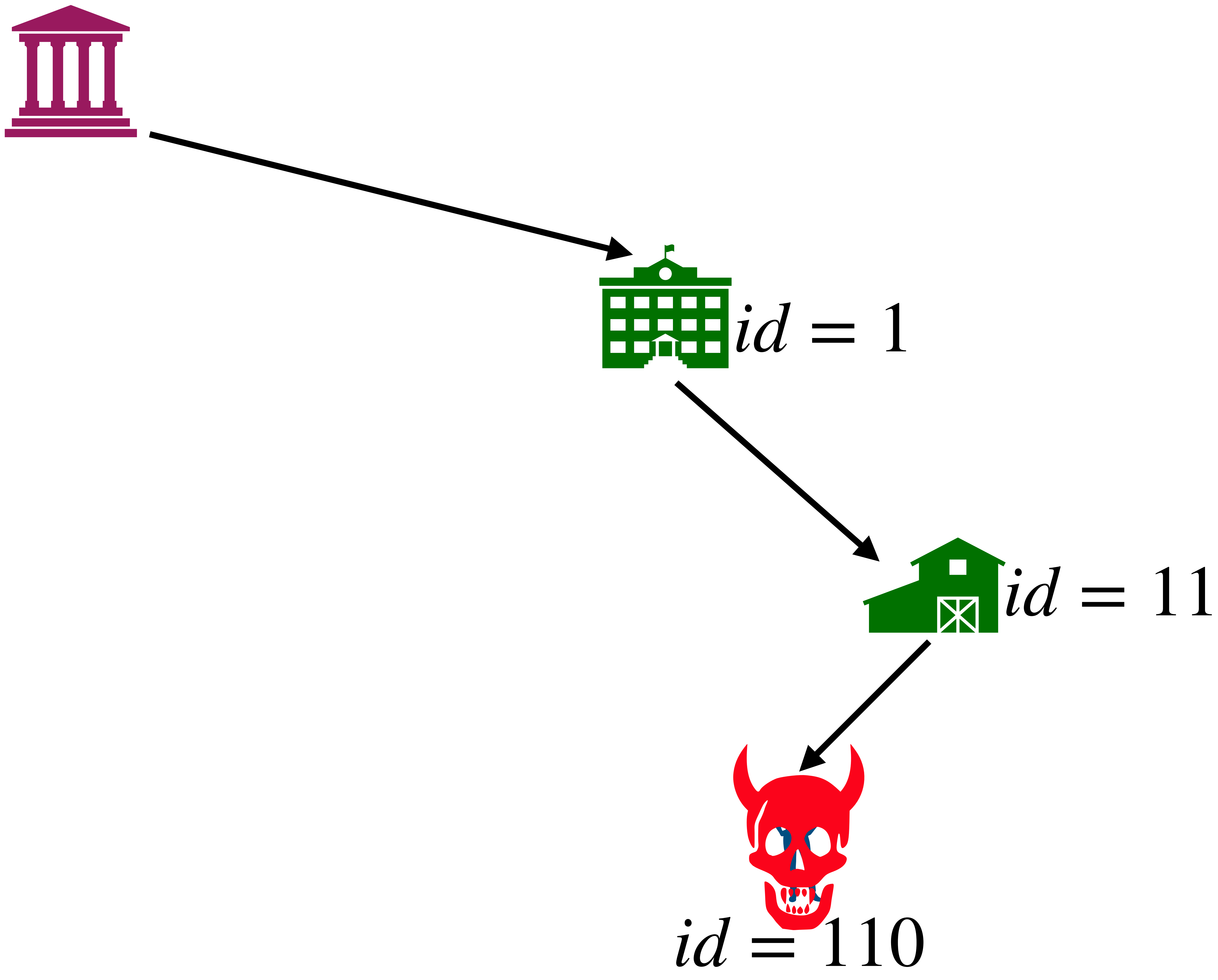
1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$



Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

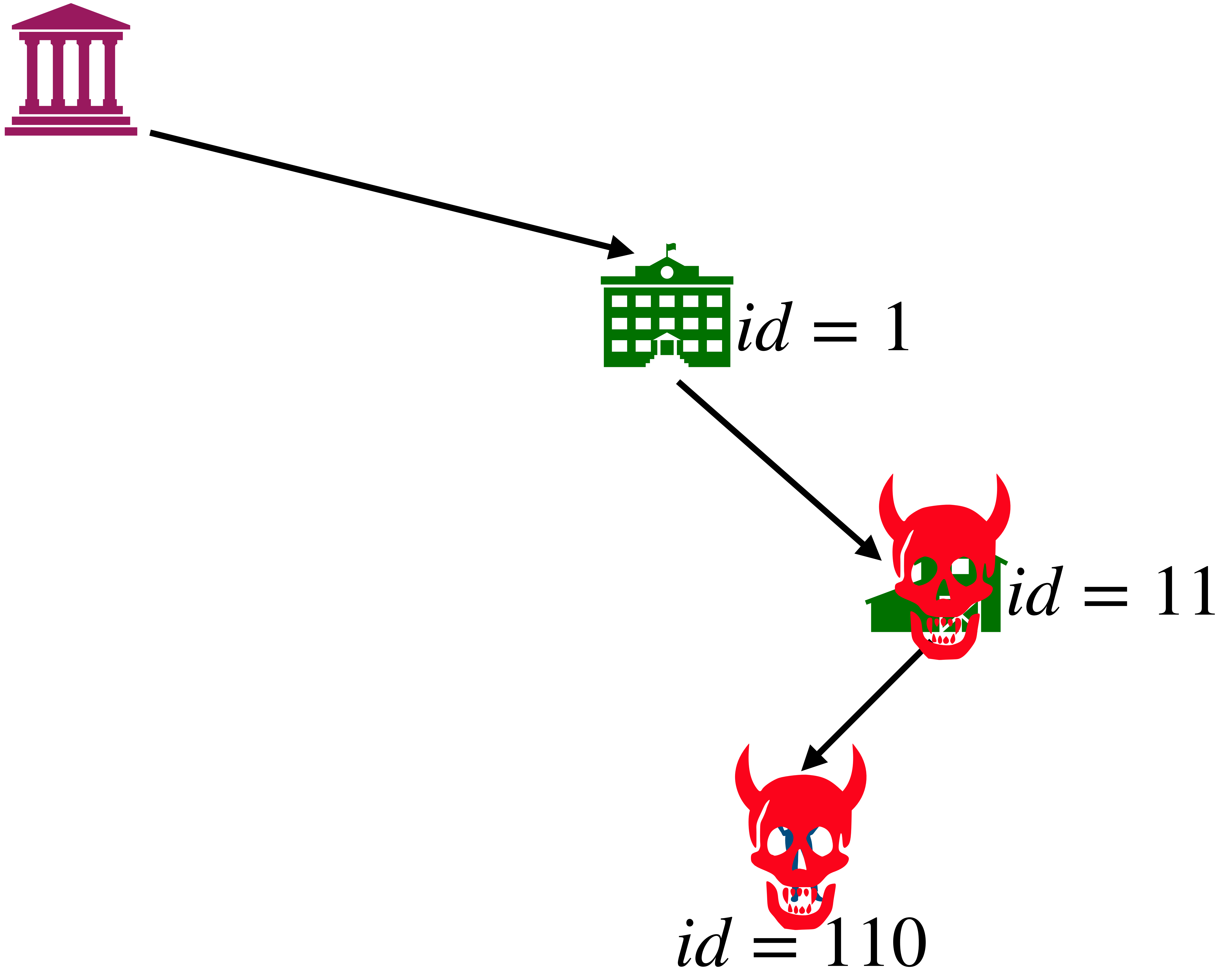
Security



Security

Attacker

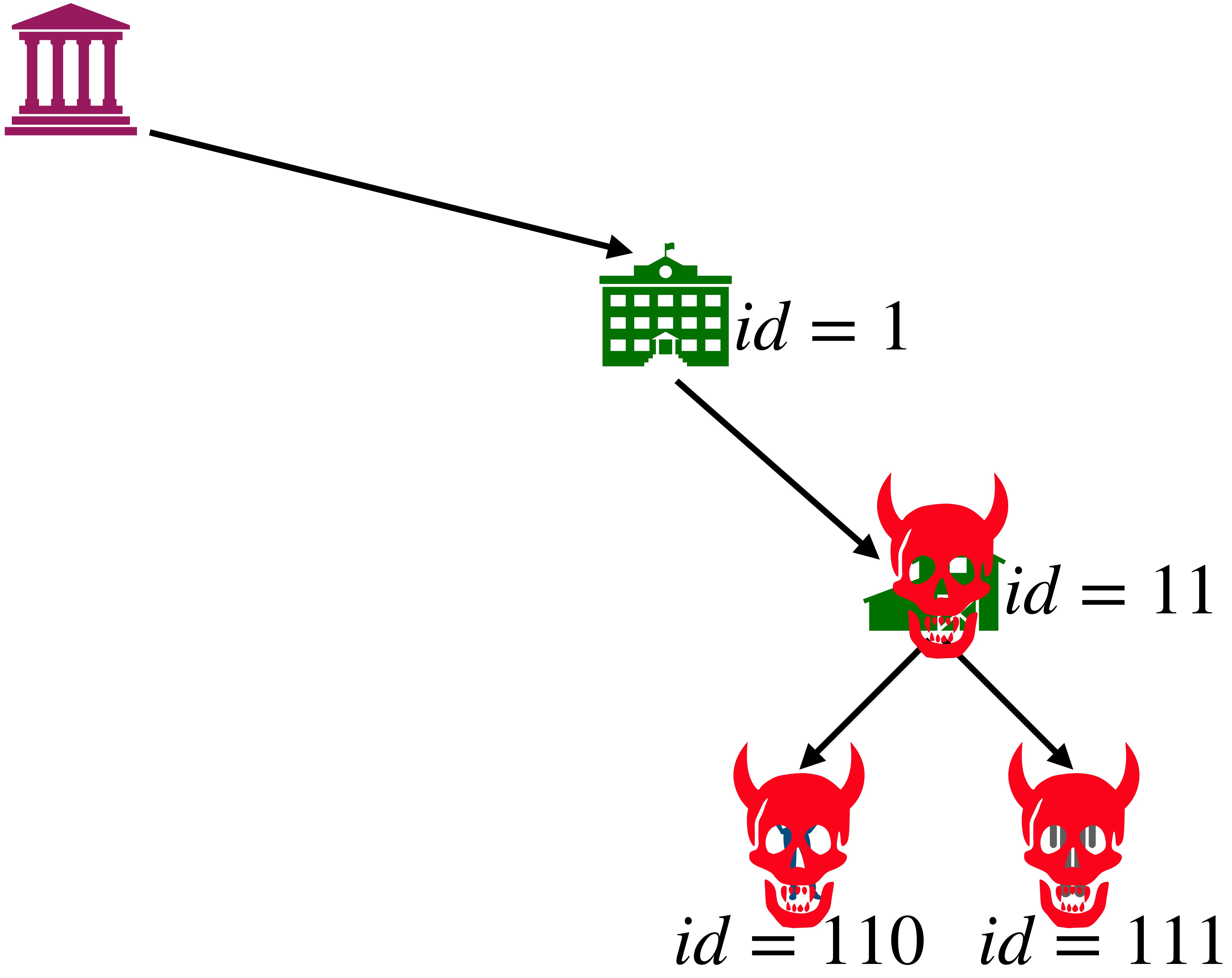
1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$



Security

Attacker

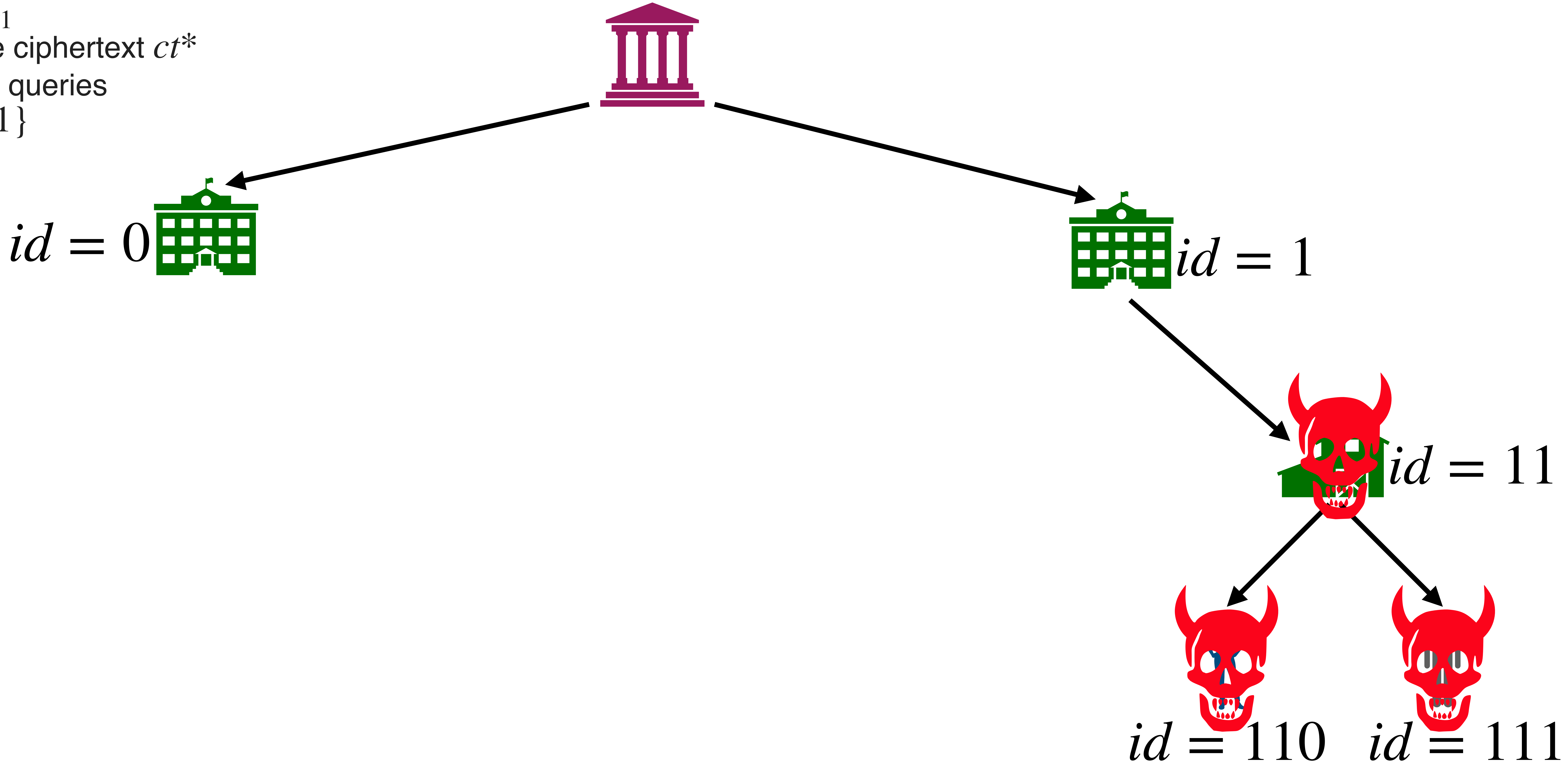
1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$



Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

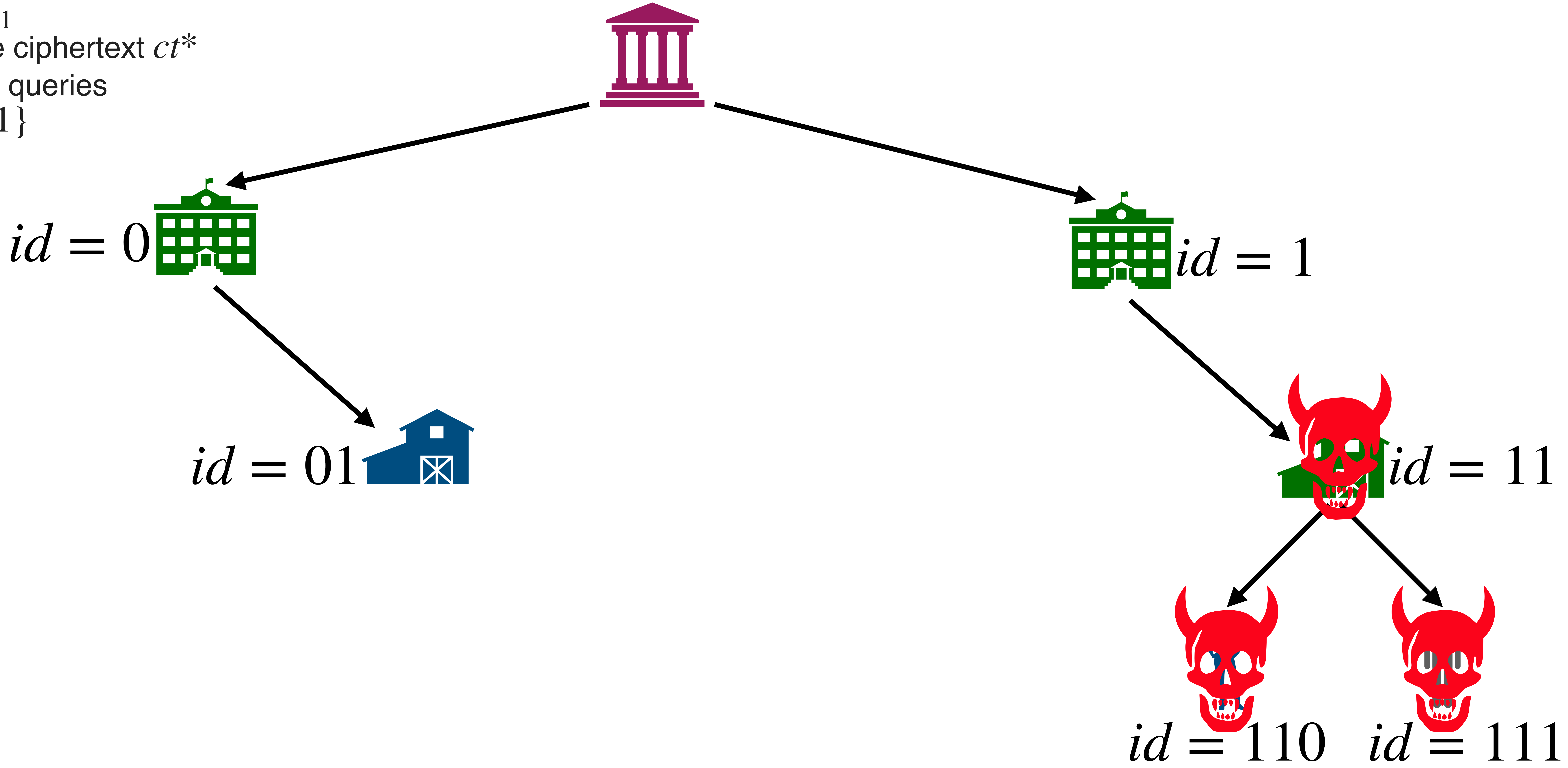
Security



Security

Attacker

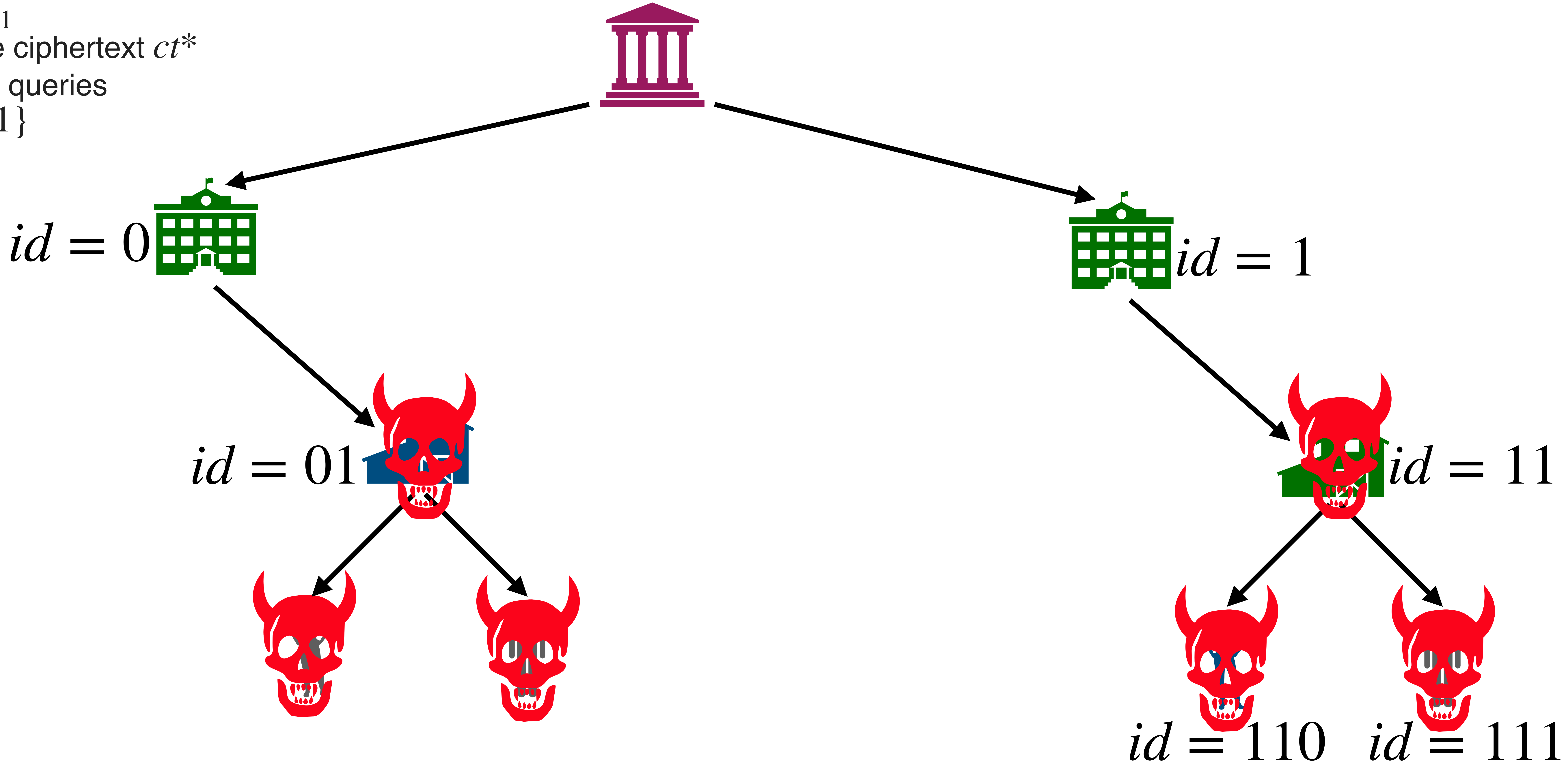
1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$



Security

Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$



Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Security

Security

Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Security

Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Security

Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Security

Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Security

Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Security

Attacker

1. Obtains mpk
2. Adaptively makes queries
3. Sends id^*, m_0, m_1
4. Obtains challenge ciphertext ct^*
5. Adaptively makes queries
6. Outputs $b' \in \{0,1\}$

Our Work

Our Work

Theorem 1: Assuming a ***selectively*** *secure IBE* system, there exists a ***fully*** *secure HIBE* system.

Our Work

Theorem 1: Assuming a ***selectively*** *secure IBE* system, there exists a ***fully*** *secure HIBE* system.

Our Work

Theorem 1: Assuming a ***selectively*** *secure IBE* system, there exists a ***fully*** *secure HIBE* system.

Theorem 2: Assuming hardness of CDH or LWE, there exists a ***fully*** *secure* ***anonymous*** *HIBE* system.

DG17 Construction

DG17 Construction

Building blocks

DG17 Construction

Building blocks

- Delegatable PRF (S, E, D)

DG17 Construction

Building blocks

- Delegatable PRF (S, E, D)
- Garbling scheme $(Garb, Eval)$

DG17 Construction

Building blocks

- Delegatable PRF (S, E, D)
- Garbling scheme $(Garb, Eval)$
- IBE (Gen, KG, Enc, Dec)

DG17 Construction

Building blocks

- Delegatable PRF (S, E, D)
- Garbling scheme $(Garb, Eval)$
- IBE OTSE (Gen, KG, Enc, Dec)

Delegatable PRF

Delegatable PRF

$Gen(1^{\lambda}) \rightarrow$ key k

Delegatable PRF

$Gen(1^\lambda) \to$ key k

$Eval(k, x) \to y_x$

Delegatable PRF

$Gen(1^\lambda) \rightarrow$ key k

$Eval(k, x) \rightarrow y_x$

$Delegate(k, x) \rightarrow$ delegated key k_x

Delegatable PRF

$Gen(1^\lambda) \rightarrow$ key k

$Eval(k, x) \rightarrow y_x$

$Delegate(k, x) \rightarrow$ delegated key k_x

Correctness: For all x, x'

$Eval(k, x \| x') = Eval(k_x, x')$

$Delegate(k, x \| x') = Delegate(k_x, x')$

Delegatable PRF

$Gen(1^\lambda) \rightarrow$ key k

$Eval(k, x) \rightarrow y_x$

$Delegate(k, x) \rightarrow$ delegated key k_x

Correctness: For all x, x'

$$Eval(k, x \| x') = Eval(k_x, x')$$

$$\boxed{Delegate(k, x \| x') = Delegate(k_x, x')}$$

Delegatable PRF

$Gen(1^\lambda) \rightarrow$ key k

$Eval(k, x) \rightarrow y_x$

$Delegate(k, x) \rightarrow$ delegated key k_x

Correctness: For all x, x'

$Eval(k, x\|x') = Eval(k_x, x')$

$\boxed{Delegate(k, x\|x') = Delegate(k_x, x')}$

Security

Delegatable PRF

$Gen(1^\lambda) \rightarrow$ key k

$Eval(k, x) \rightarrow y_x$

$Delegate(k, x) \rightarrow$ delegated key k_x

Correctness: For all x, x'

$Eval(k, x \| x') = Eval(k_x, x')$

$\boxed{Delegate(k, x \| x') = Delegate(k_x, x')}$

Security

$(x^*,$

$(x^*,$

Delegatable PRF

$Gen(1^\lambda) \to$ key k

$Eval(k, x) \to y_x$

$Delegate(k, x) \to$ delegated key k_x

Correctness: For all x, x'

$Eval(k, x \| x') = Eval(k_x, x')$

$\boxed{Delegate(k, x \| x') = Delegate(k_x, x')}$

Security

$(x^*, \{Eval(k, x')\}_{x' \leq x^*},$ $\qquad (x^*,$

Delegatable PRF

$Gen(1^\lambda) \to$ key k

$Eval(k, x) \to y_x$

$Delegate(k, x) \to$ delegated key k_x

Correctness: For all x, x'

$$Eval(k, x \| x') = Eval(k_x, x')$$

$$\boxed{Delegate(k, x \| x') = Delegate(k_x, x')}$$

Security

$(x^*, \{Eval(k, x')\}_{x' \preceq x^*},$

$(x^*, \{y_i^*\}_{i \in [|x^*|]},$

Delegatable PRF

$Gen(1^\lambda) \to$ key k

$Eval(k, x) \to y_x$

$Delegate(k, x) \to$ delegated key k_x

Correctness: For all x, x'

$Eval(k, x \| x') = Eval(k_x, x')$

$\boxed{Delegate(k, x \| x') = Delegate(k_x, x')}$

Security

$(x^*, \{Eval(k, x')\}_{x' \preceq x^*}, \{x, Eval(k, x)\},$

$(x^*, \{y_i^*\}_{i \in [|x^*|]}, \{x, Eval(k, x)\},$

Delegatable PRF

$Gen(1^\lambda) \to$ key k

$Eval(k, x) \to y_x$

$Delegate(k, x) \to$ delegated key k_x

Correctness: For all x, x'

$Eval(k, x \| x') = Eval(k_x, x')$

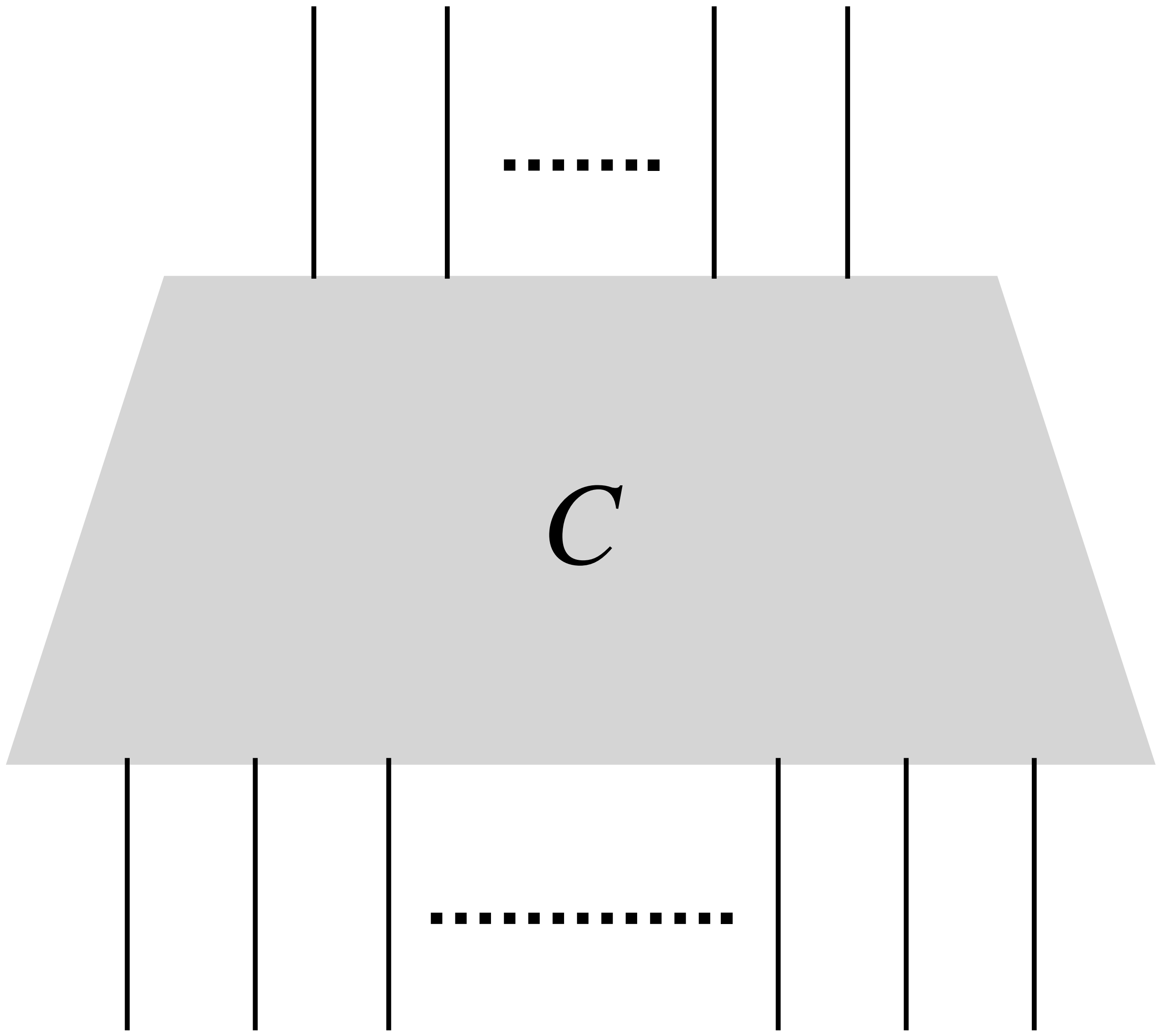
$\boxed{Delegate(k, x \| x') = Delegate(k_x, x')}$

Security

$$(x^*, \{Eval(k, x')\}_{x' \leq x^*}, \{x, Eval(k, x)\}, \{x, Delegate(k, x)\}) \approx_c (x^*, \{y_i^*\}_{i \in [|x^*|]}, \{x, Eval(k, x)\}, \{x, Delegate(k, x)\})$$

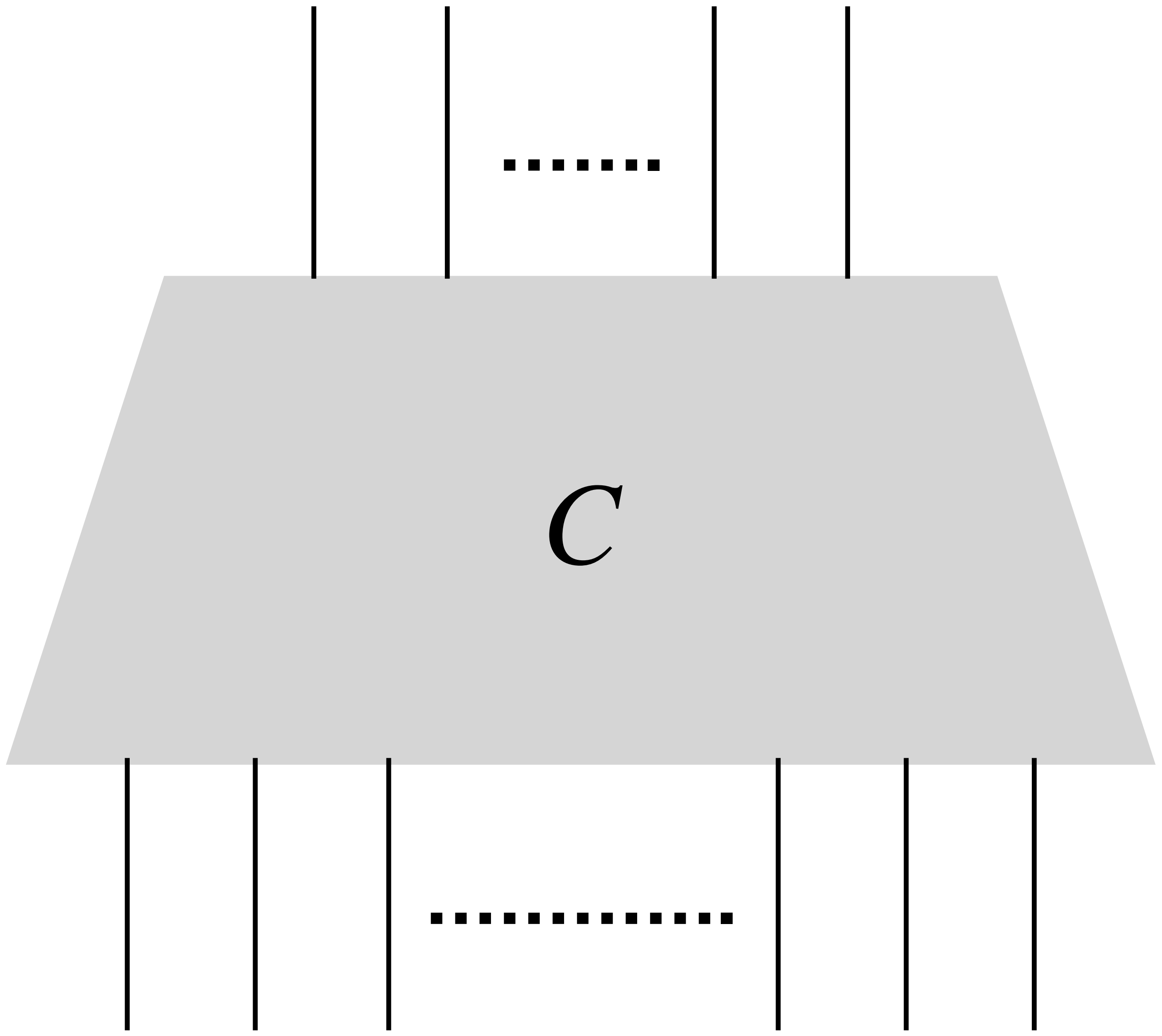
Garbling Scheme

Garbling Scheme

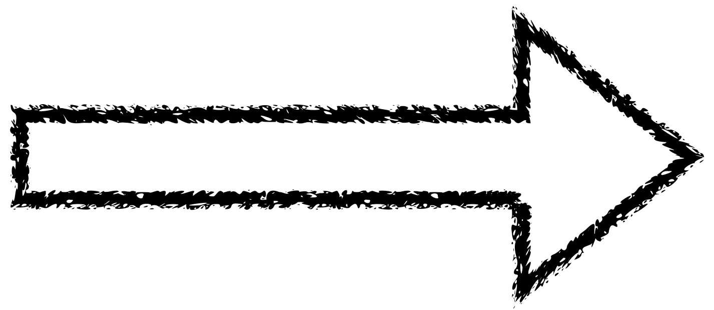


$C : \{0,1\}^k \rightarrow \{0,1\}^n$

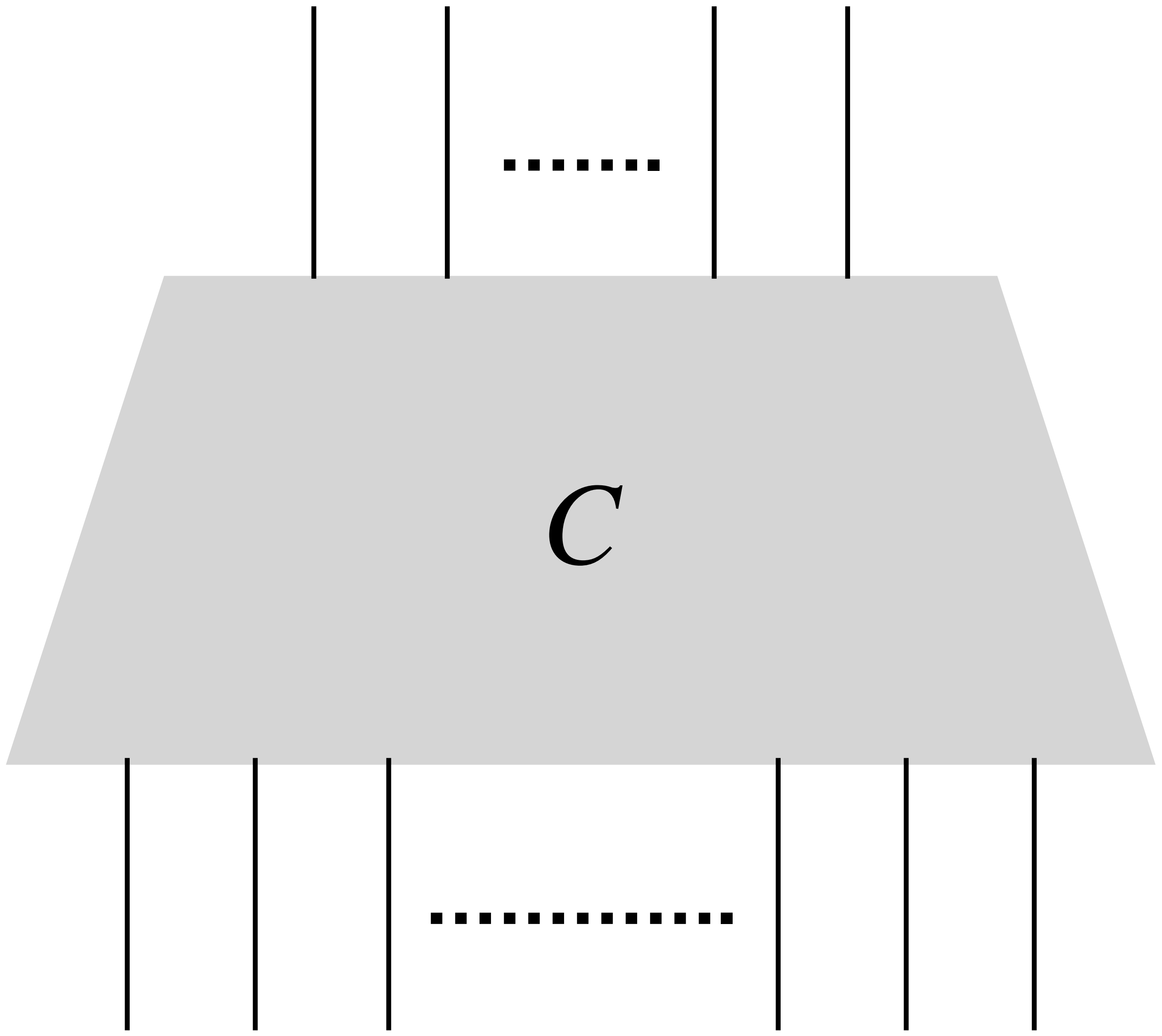
Garbling Scheme



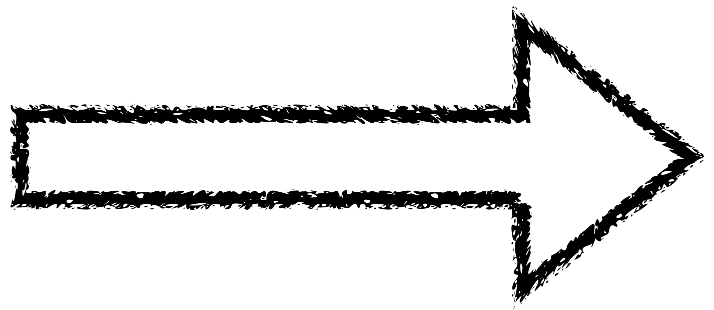
$C : \{0,1\}^k \to \{0,1\}^n$



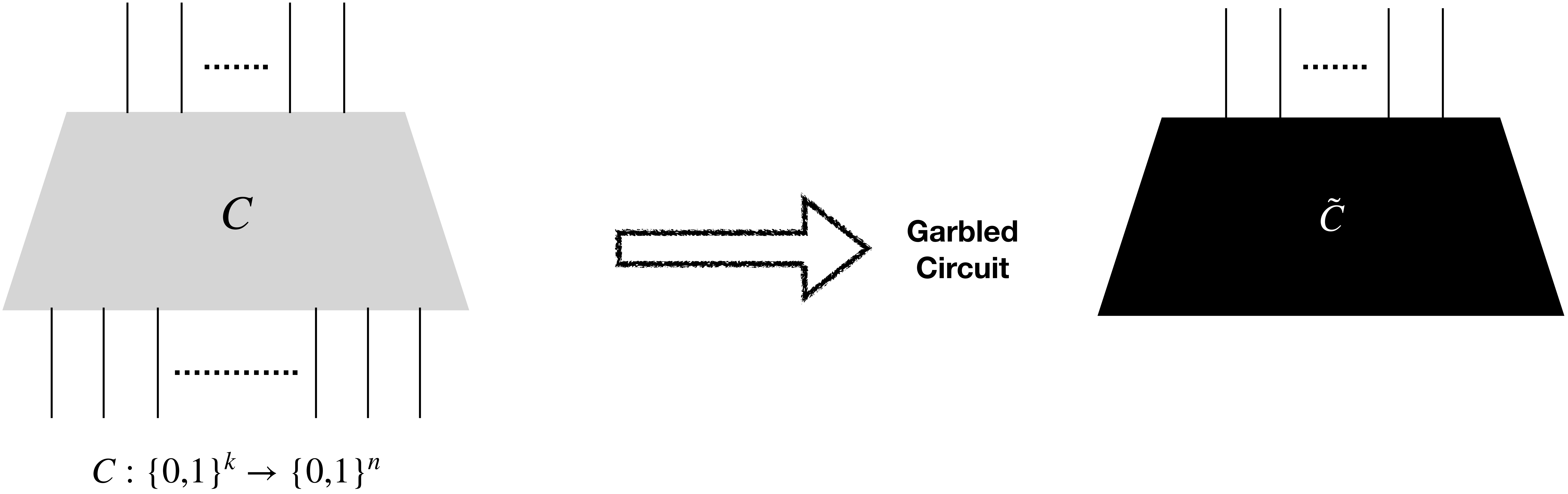
Garbling Scheme



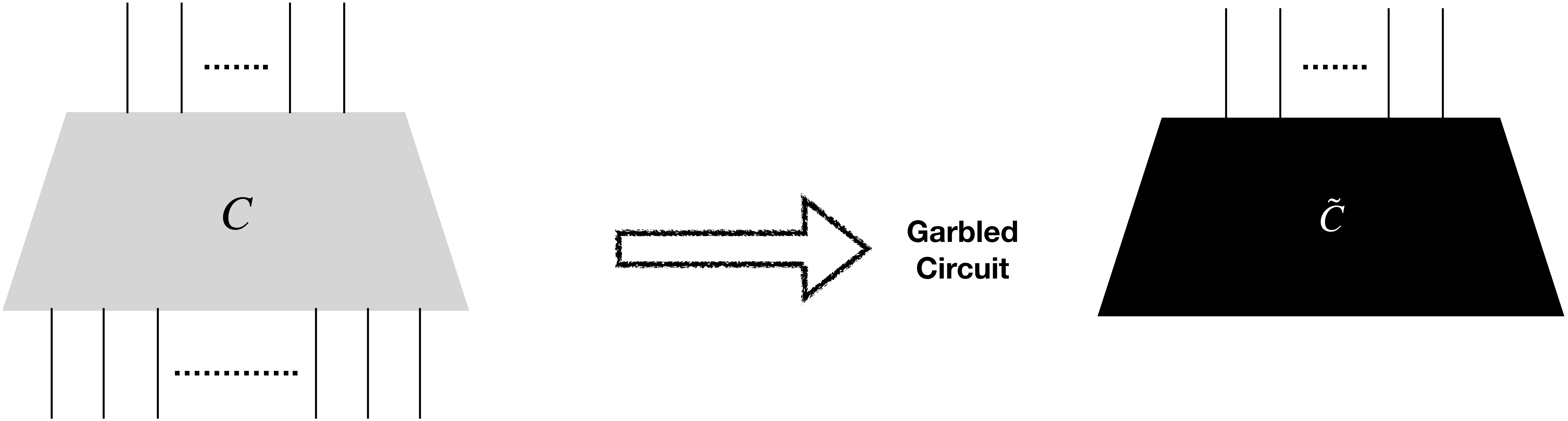
$C : \{0,1\}^k \to \{0,1\}^n$



Garbling Scheme



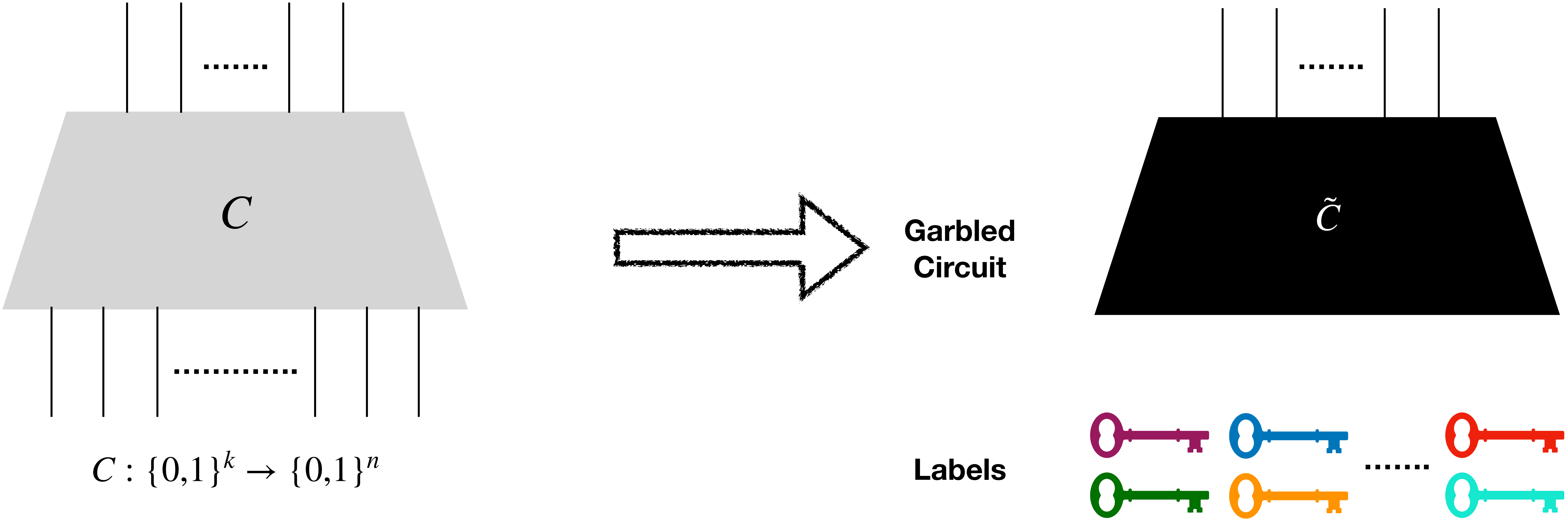
Garbling Scheme



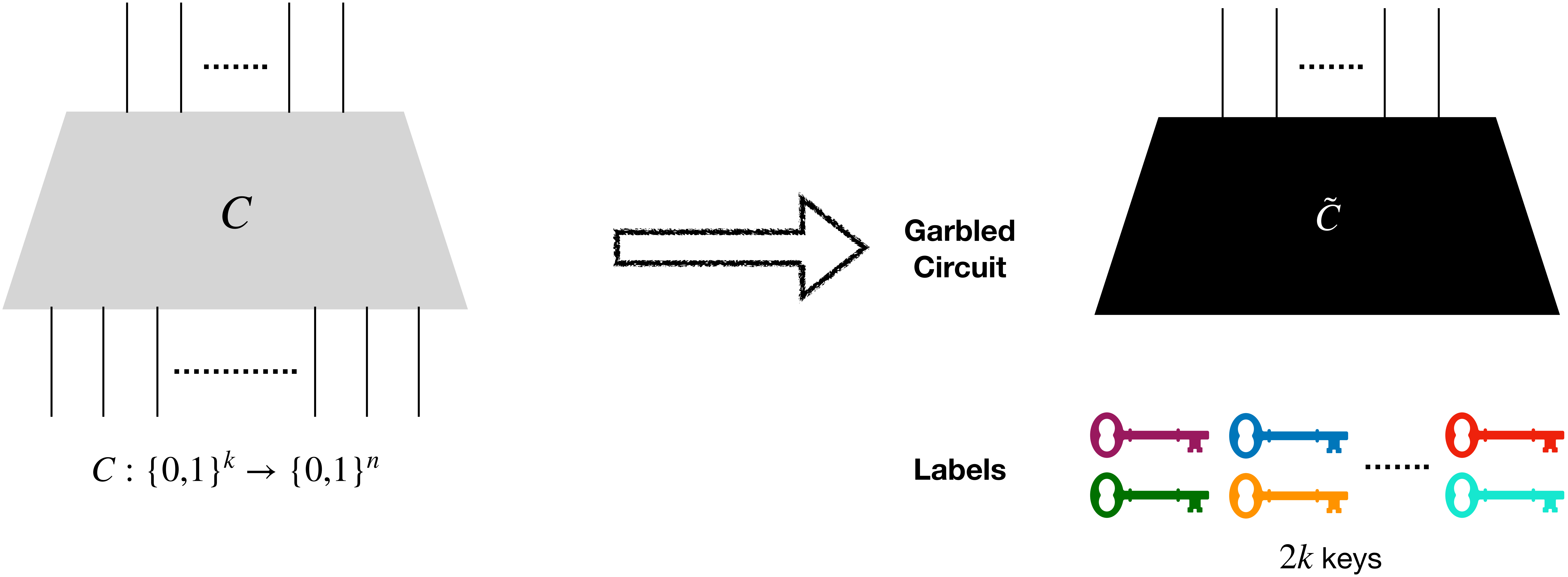
$C : \{0,1\}^k \to \{0,1\}^n$

Labels

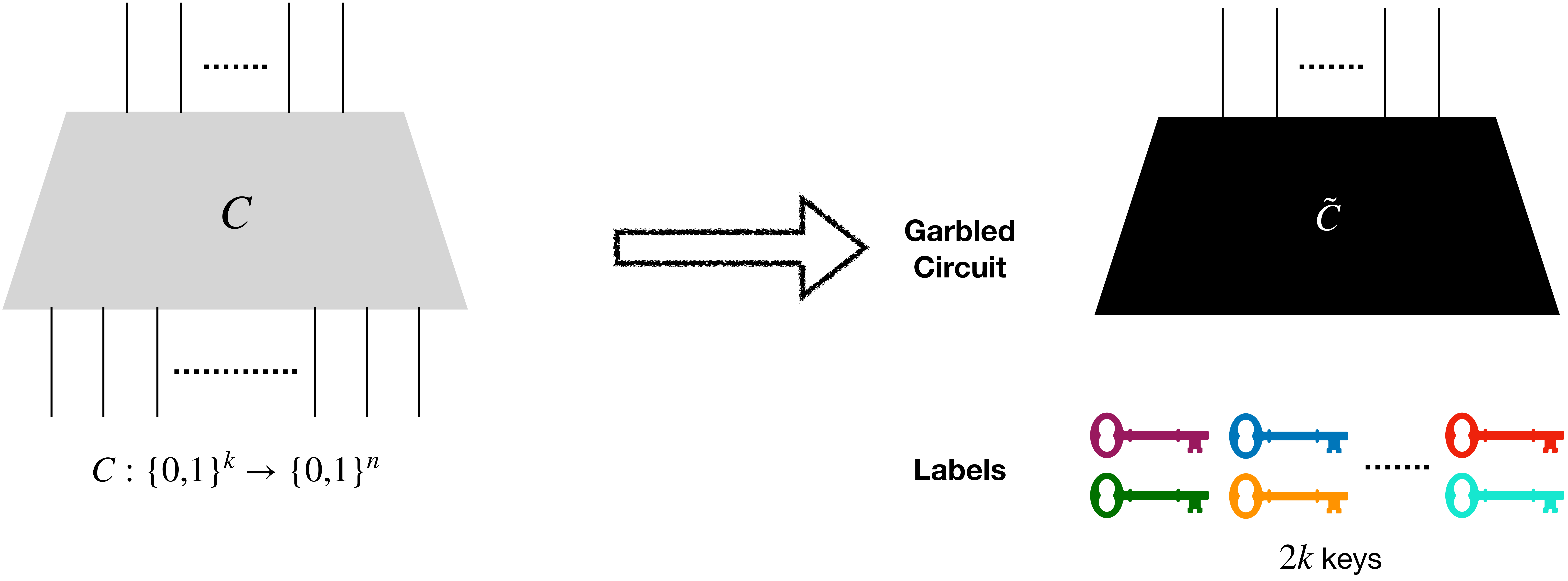
Garbling Scheme



Garbling Scheme

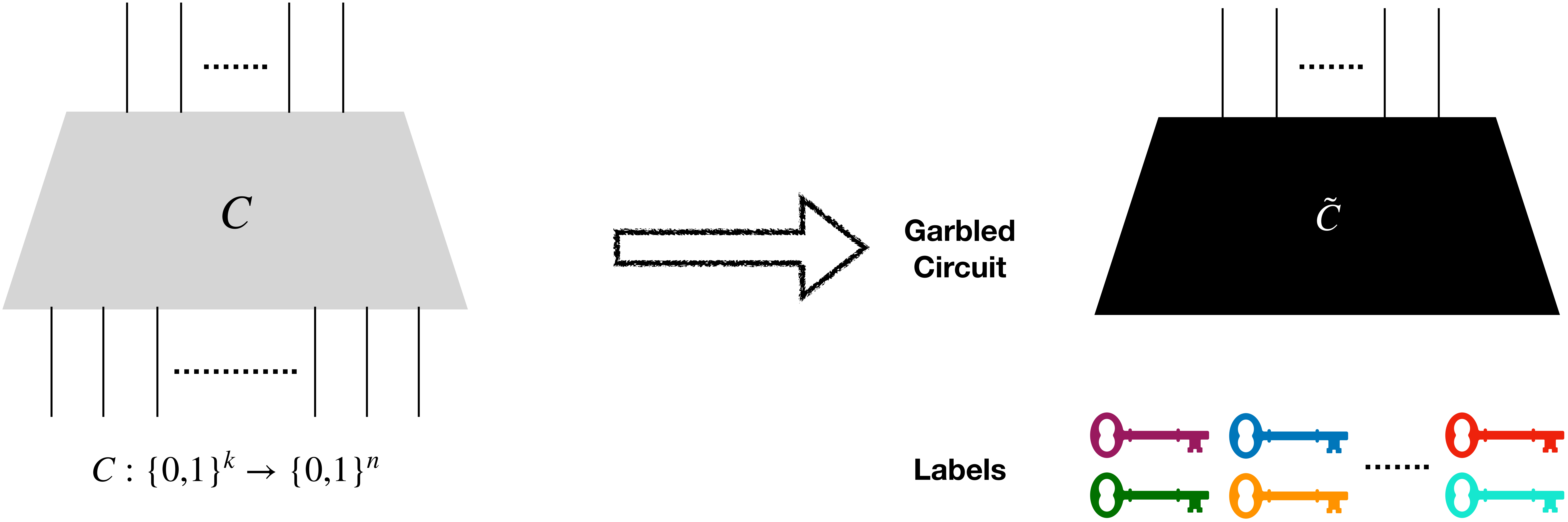


Garbling Scheme



Correctness - For any x, $C(x) = \tilde{C}($ .... $)$

Garbling Scheme



$C : \{0,1\}^k \to \{0,1\}^n$

$2k$ keys

Correctness - For any x, $C(x) = \tilde{C}($ .... $)$

Security - $\tilde{C}$ and $\{$ .... $\}$ reveals $C(x)$ and nothing more.

Setup

[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

[**Döttling-Garg'17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Setup

$\leftarrow S(1^\lambda)$

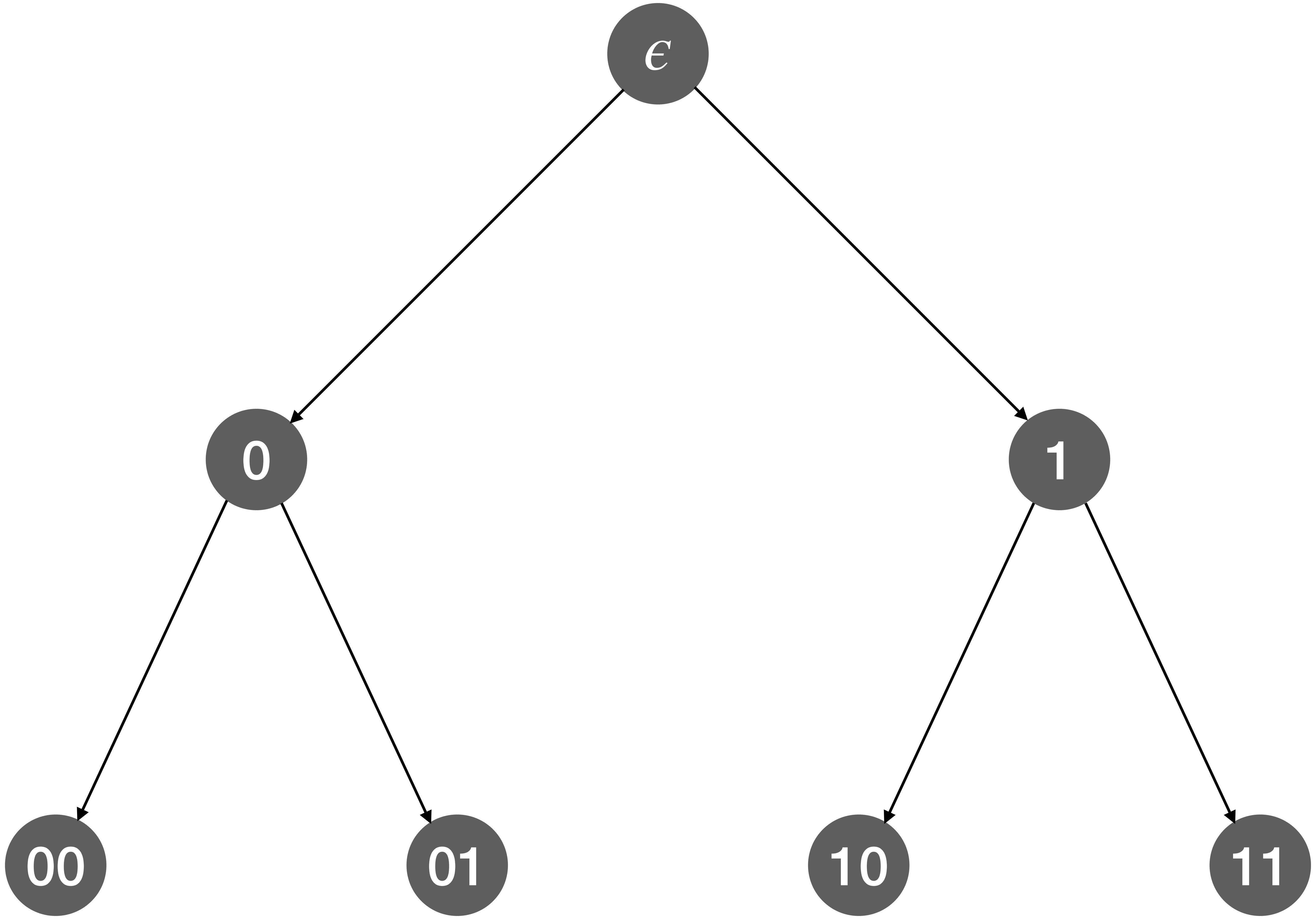
[**Döttling-Garg'17**]

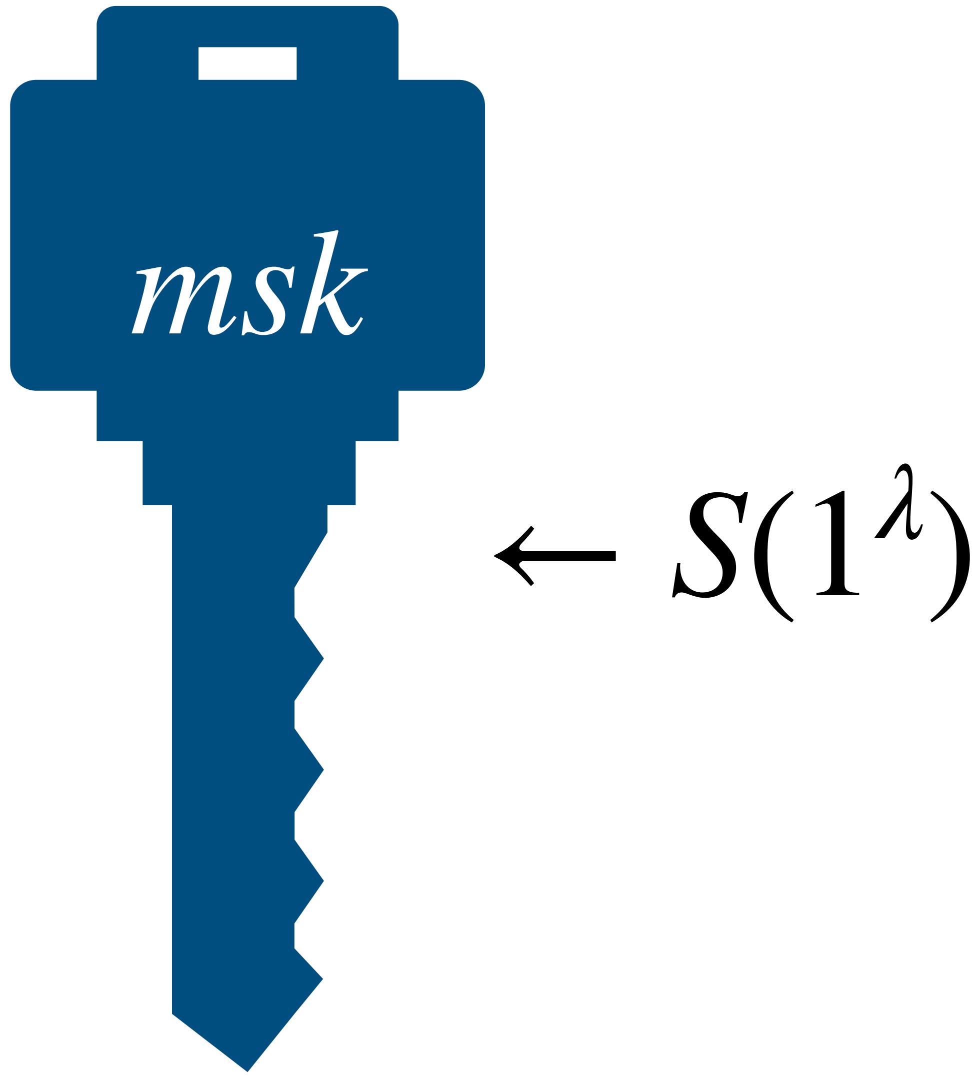
Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Setup





[**Döttling-Garg'17**]

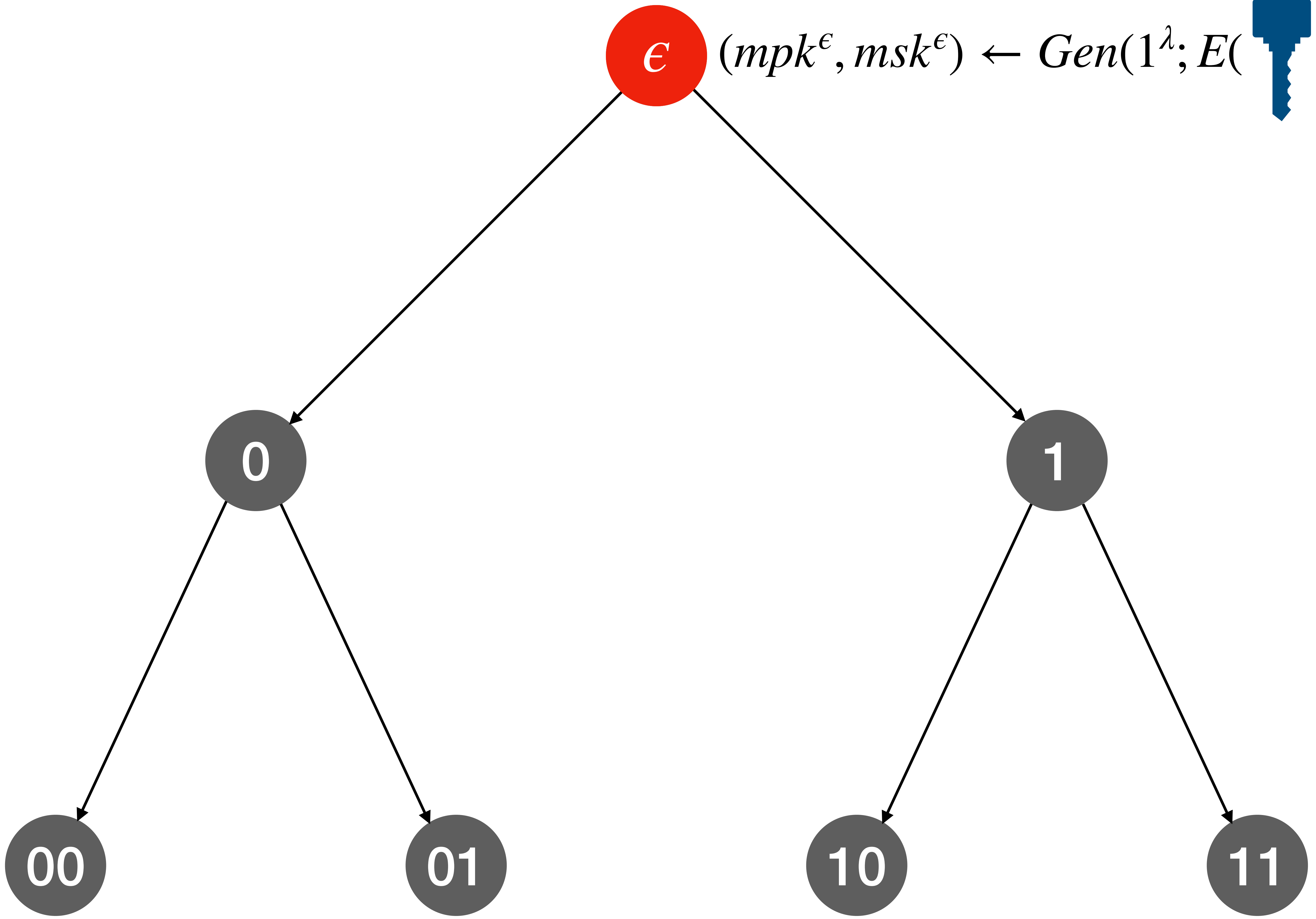
Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Setup

$(mpk^{\epsilon}, msk^{\epsilon}) \leftarrow Gen(1^{\lambda}; E(\quad, \epsilon))$



$\leftarrow S(1^{\lambda})$

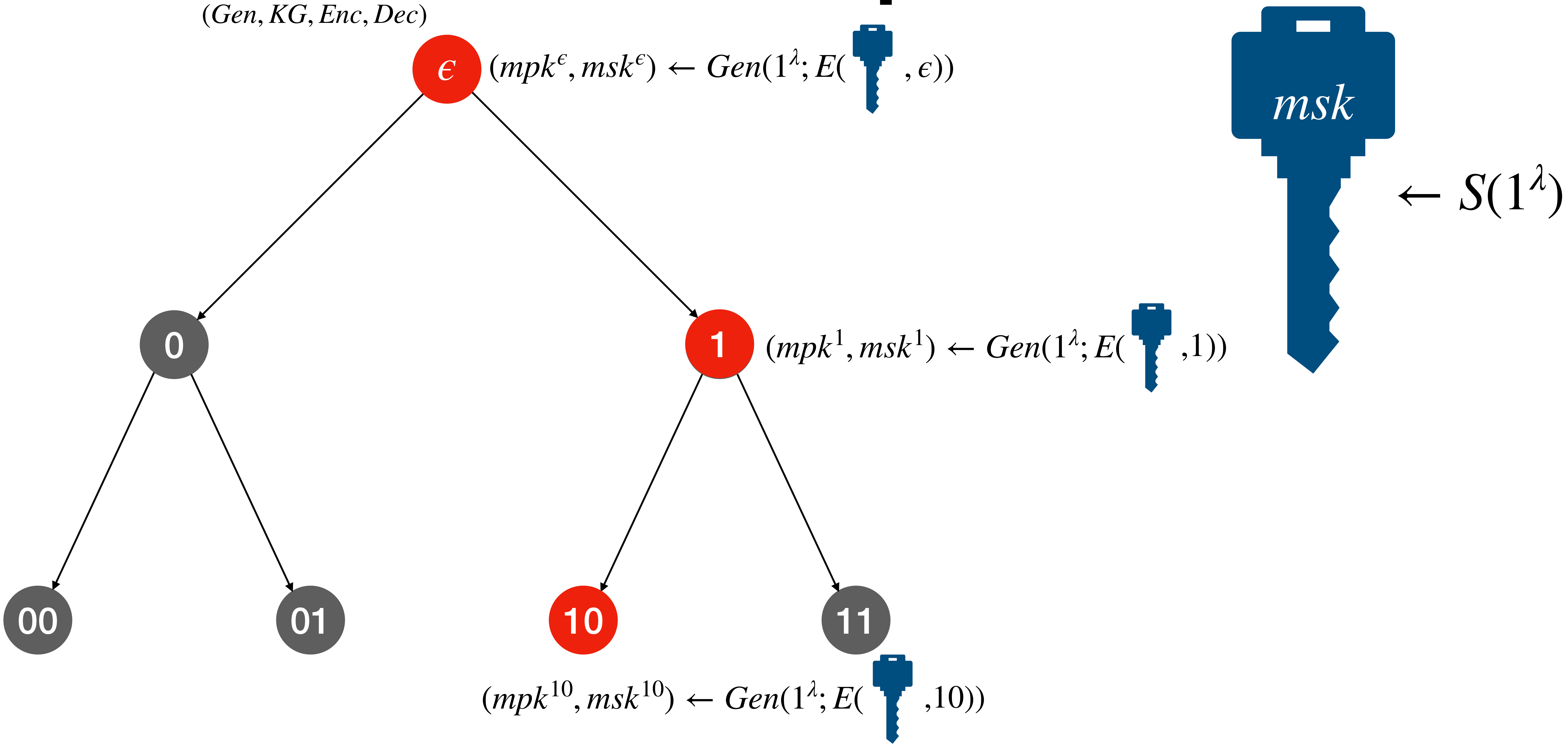
[**Döttling-Garg’17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

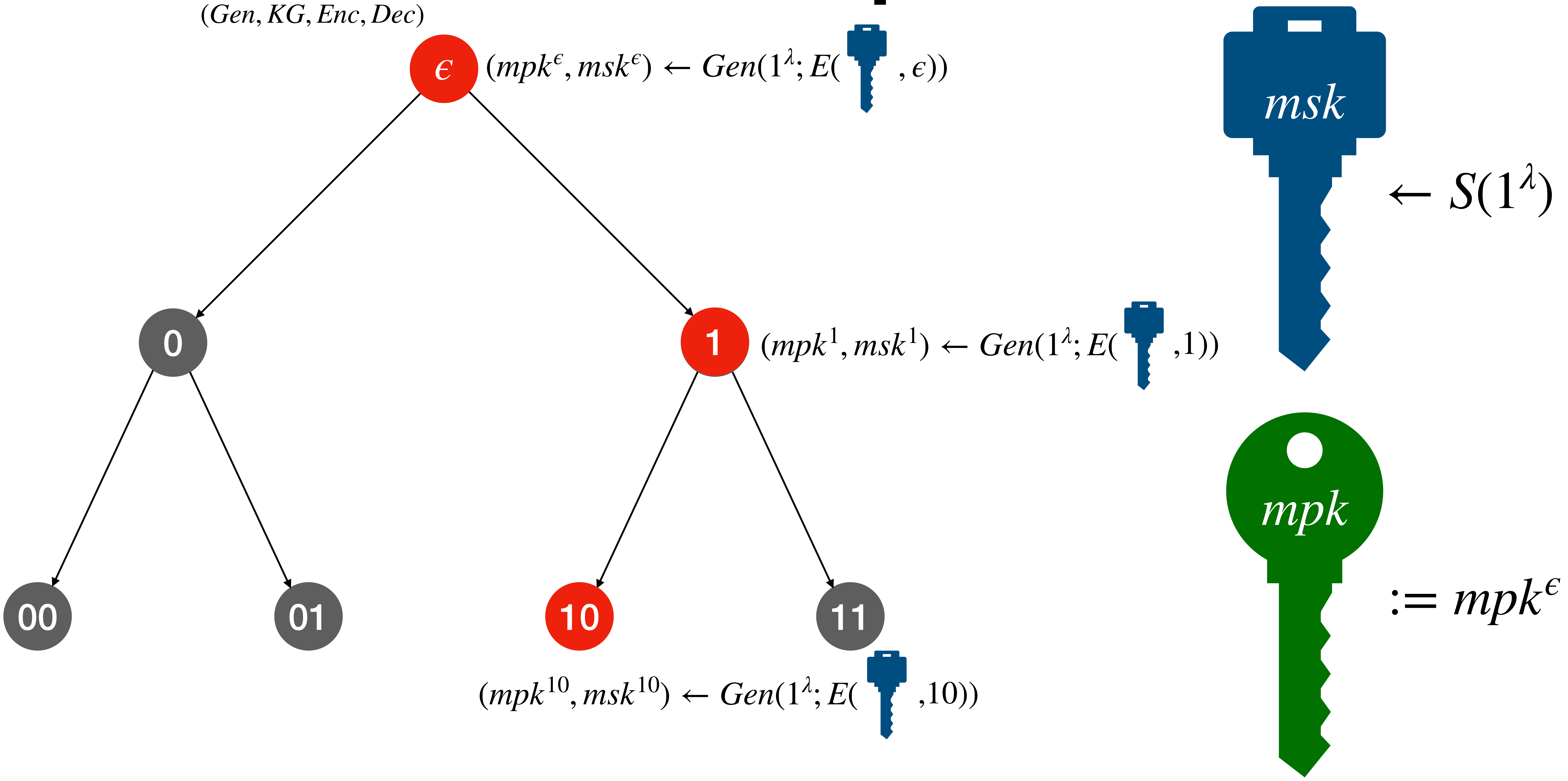
Setup



Setup

[Döttling-Garg'17]

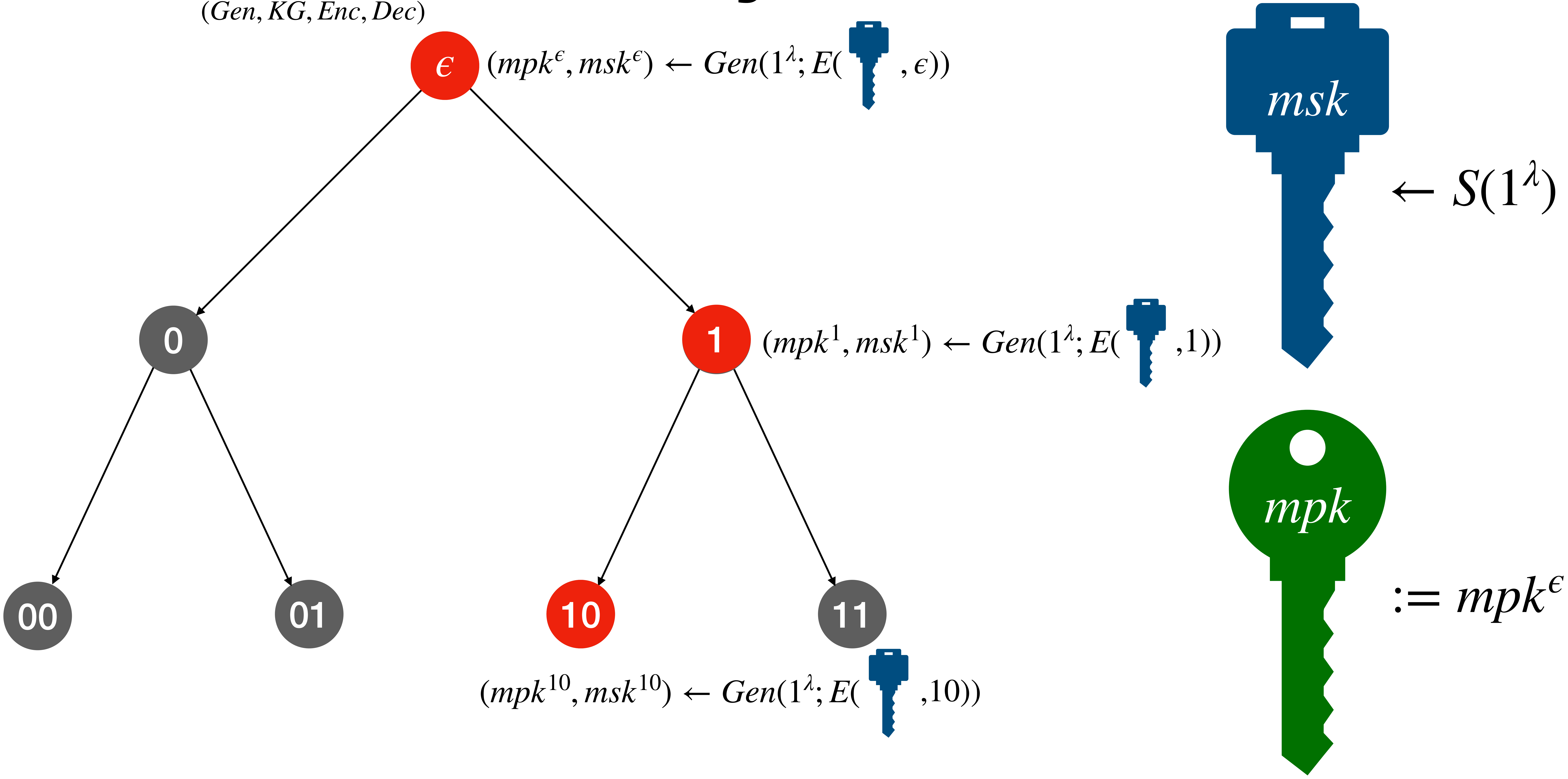
Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)



[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

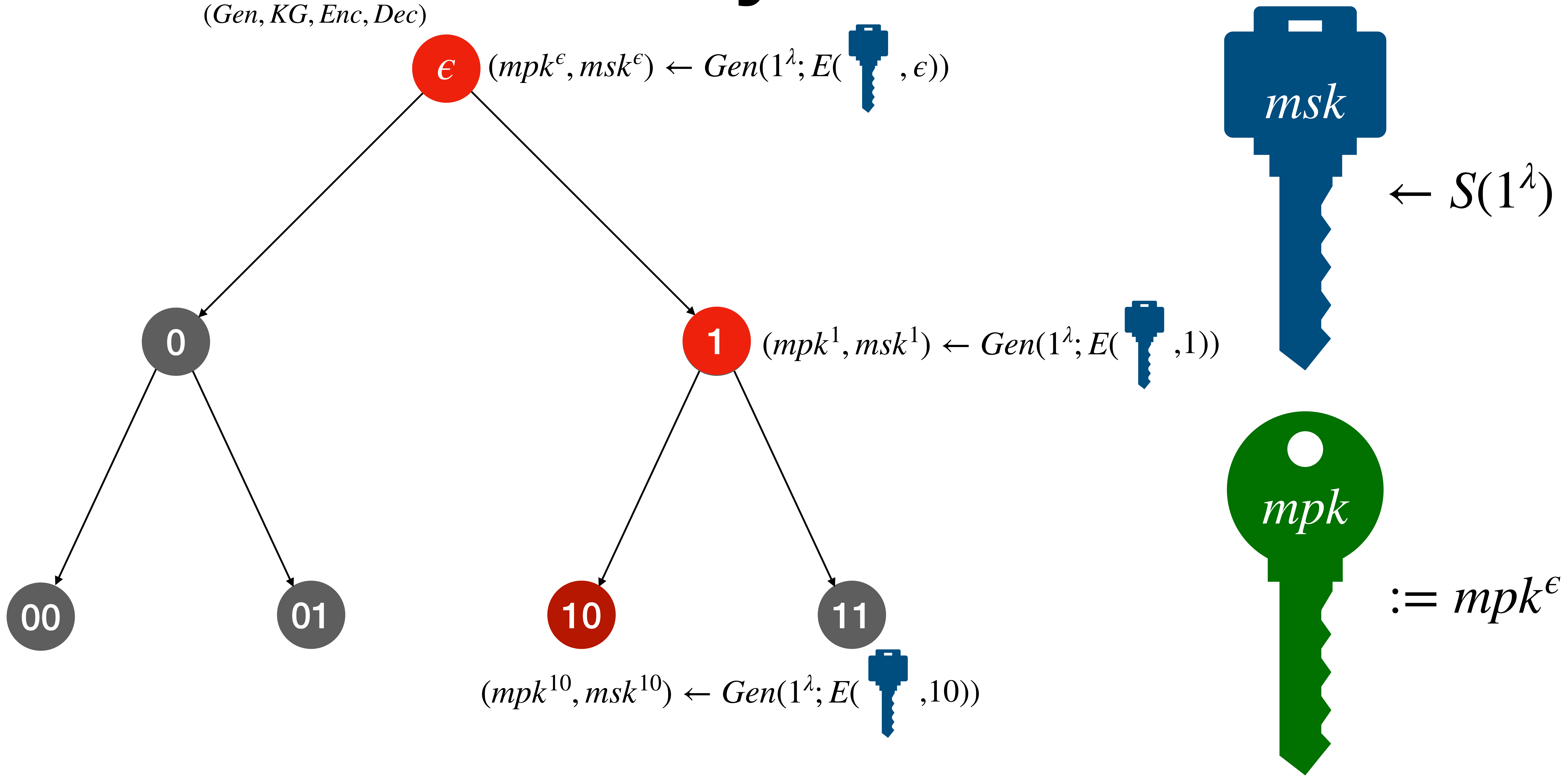
KeyGen



[**Döttling-Garg'17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

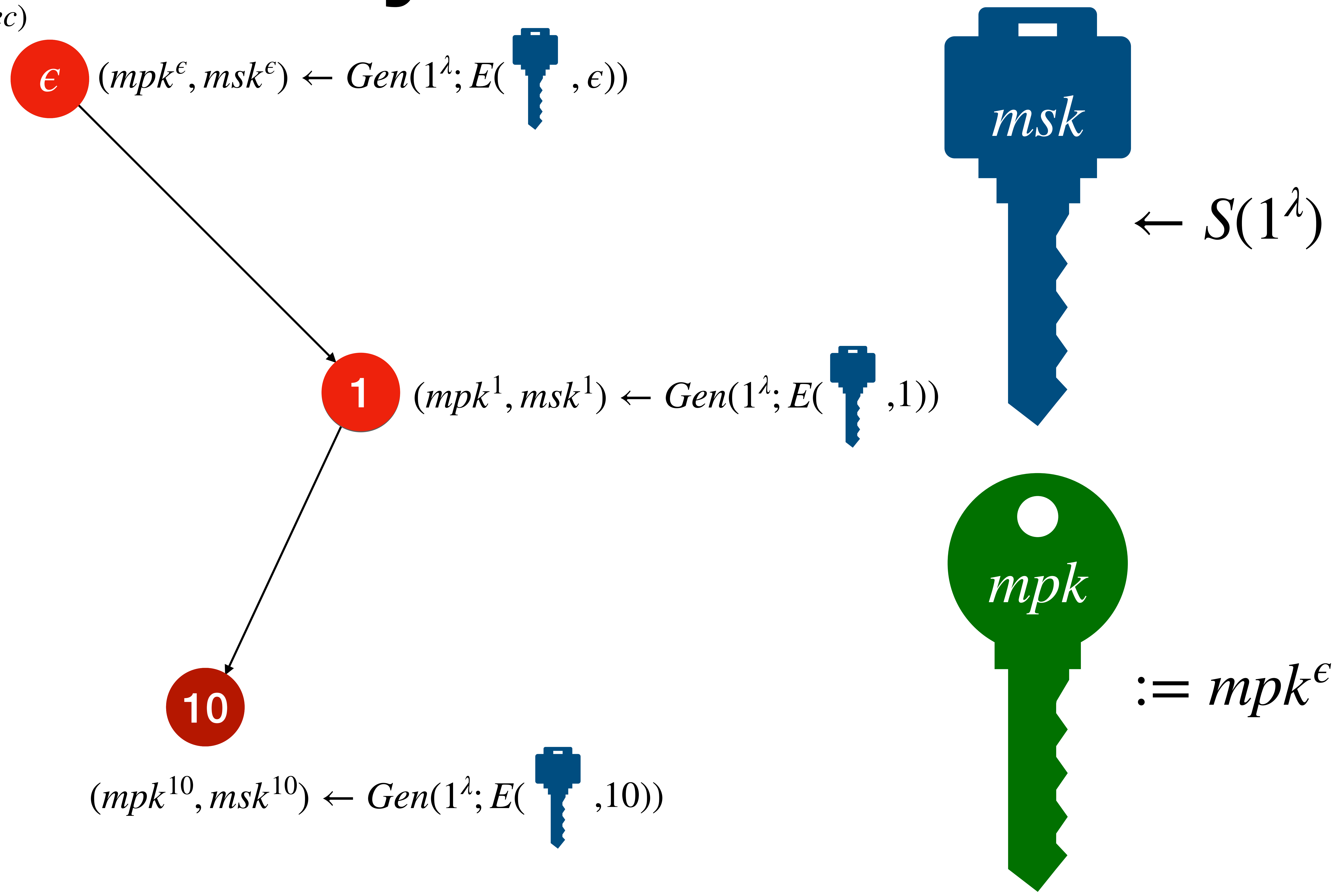
KeyGen



KeyGen

[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)



[Döttling-Garg'17]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

KeyGen

ϵ $(mpk^{\epsilon}, msk^{\epsilon}) \leftarrow Gen(1^{\lambda}; E($  $, \epsilon))$

1 $(mpk^{1}, msk^{1}) \leftarrow Gen(1^{\lambda}; E($  $, 1))$

$(mpk^{10}, msk^{10}) \leftarrow Gen(1^{\lambda}; E($  $, 10))$ 10

$\leftarrow S(1^{\lambda})$

$:= mpk^{\epsilon}$

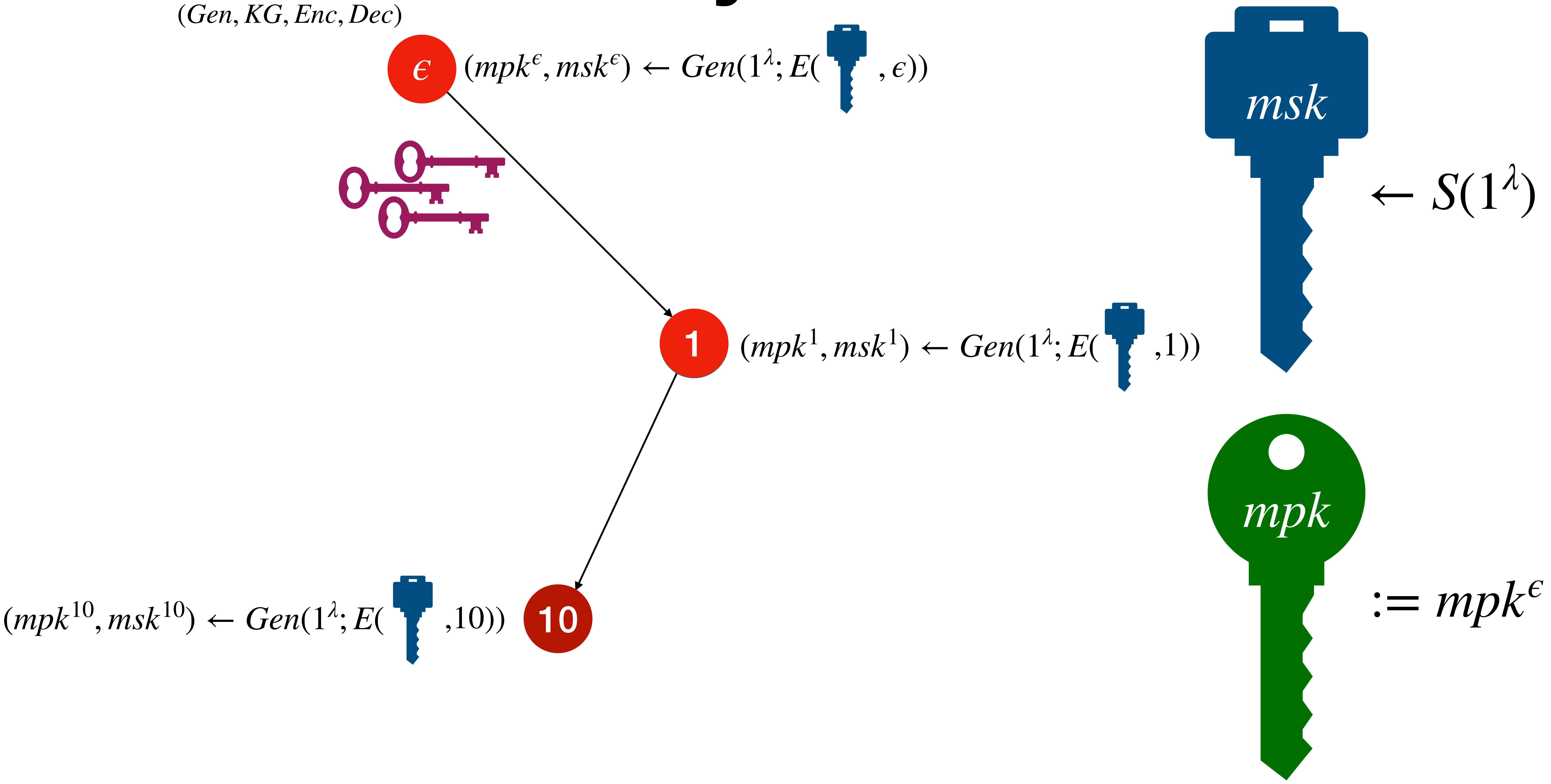
[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

KeyGen



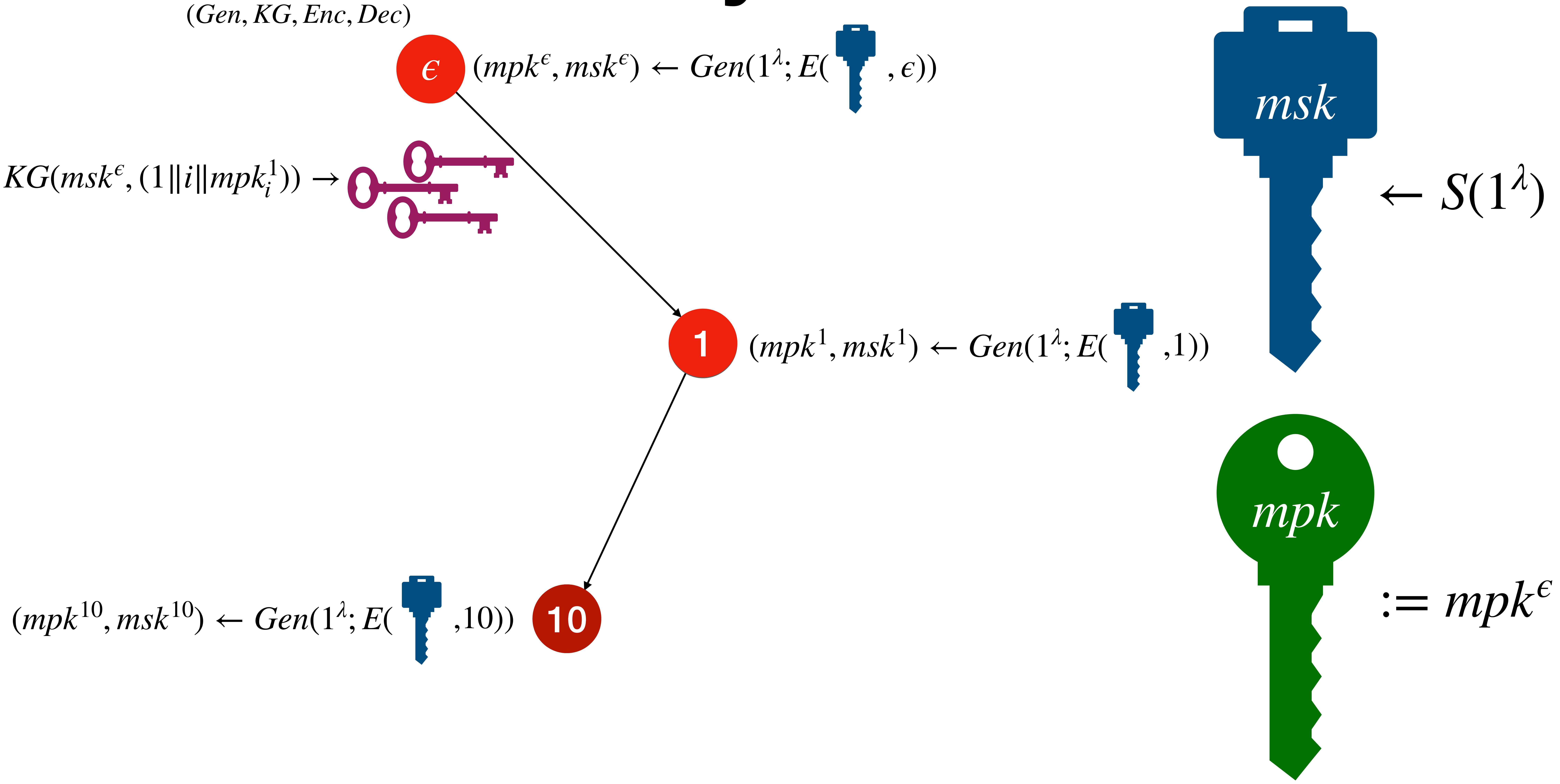
[**Döttling-Garg’17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

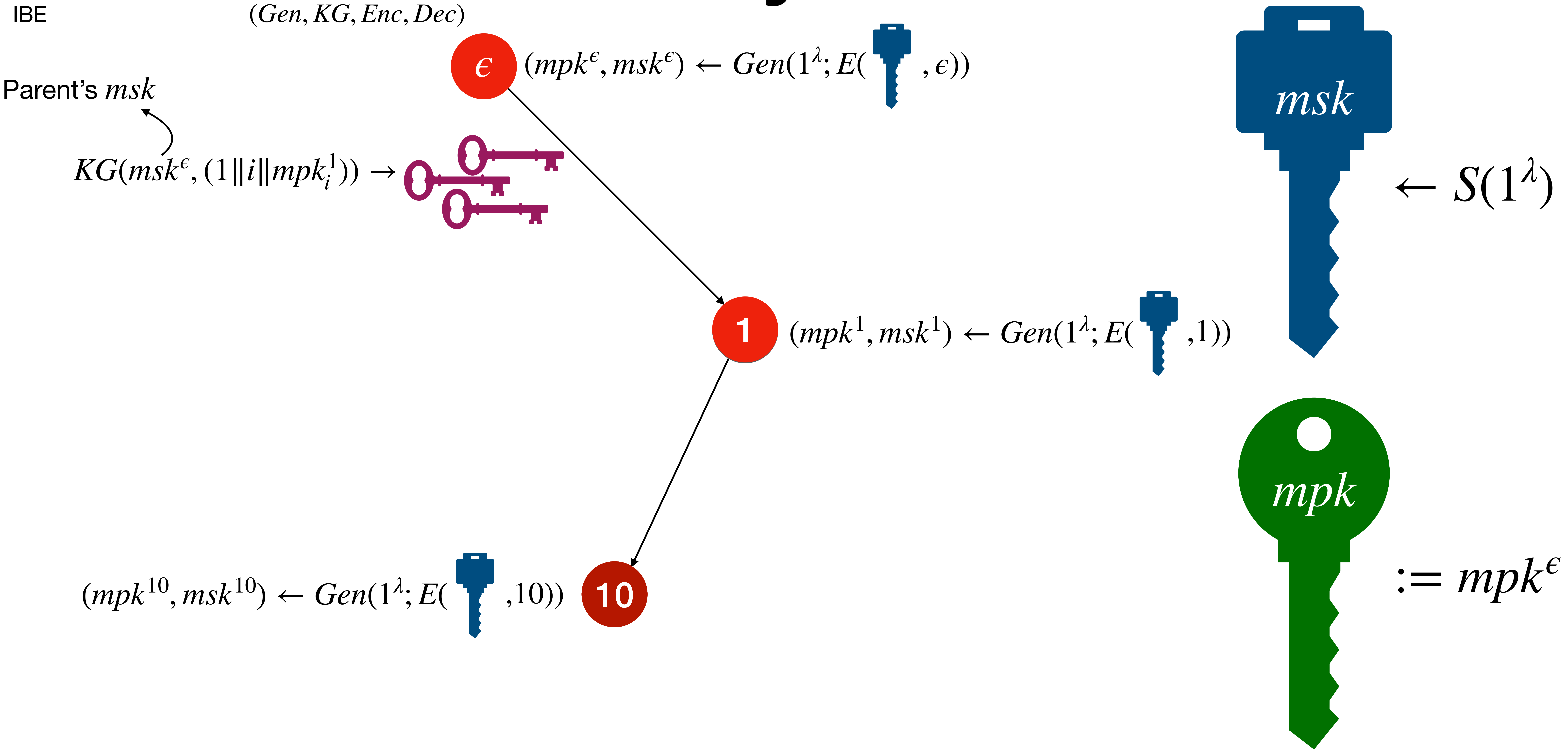
KeyGen



[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

KeyGen



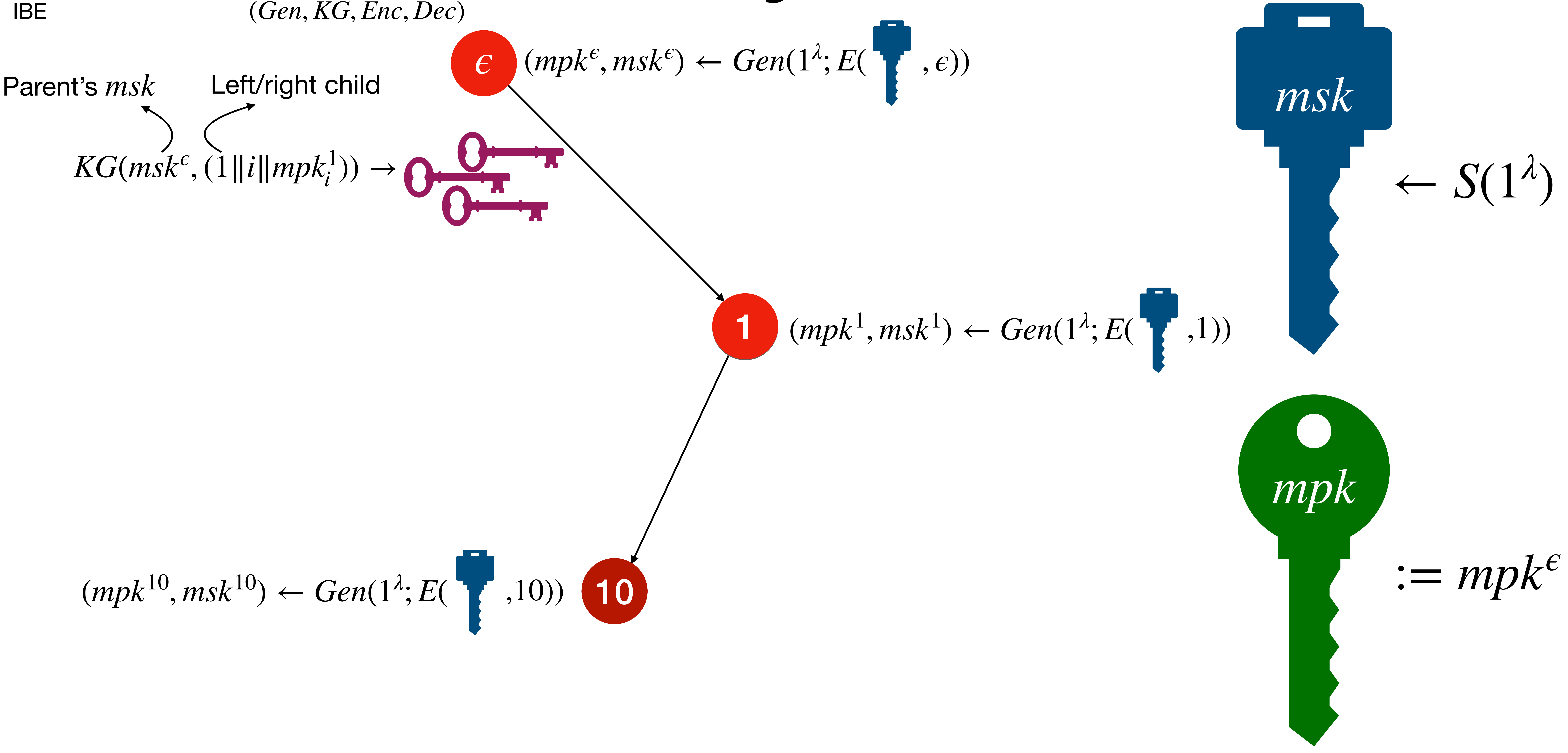
KeyGen

[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)



[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

KeyGen

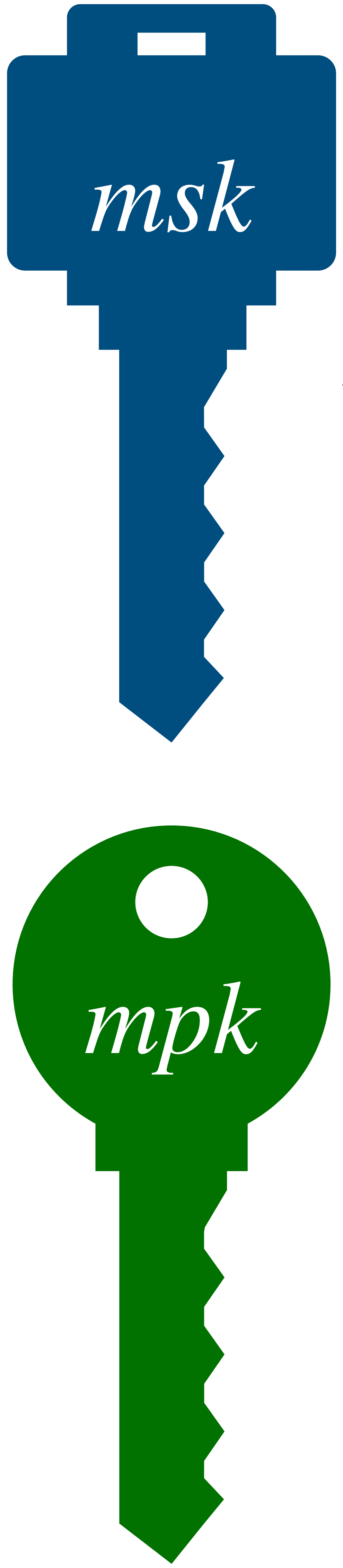
ϵ $(mpk^\epsilon, msk^\epsilon) \leftarrow Gen(1^\lambda; E(\quad, \epsilon))$

Parent's msk Left/right child

$KG(msk^\epsilon, (1\|\underbrace{i}_{i^{th} \text{ bit of child's } mpk}\|mpk_i^1)) \to$

1 $(mpk^1, msk^1) \leftarrow Gen(1^\lambda; E(\quad, 1))$

$(mpk^{10}, msk^{10}) \leftarrow Gen(1^\lambda; E(\quad, 10))$ 10



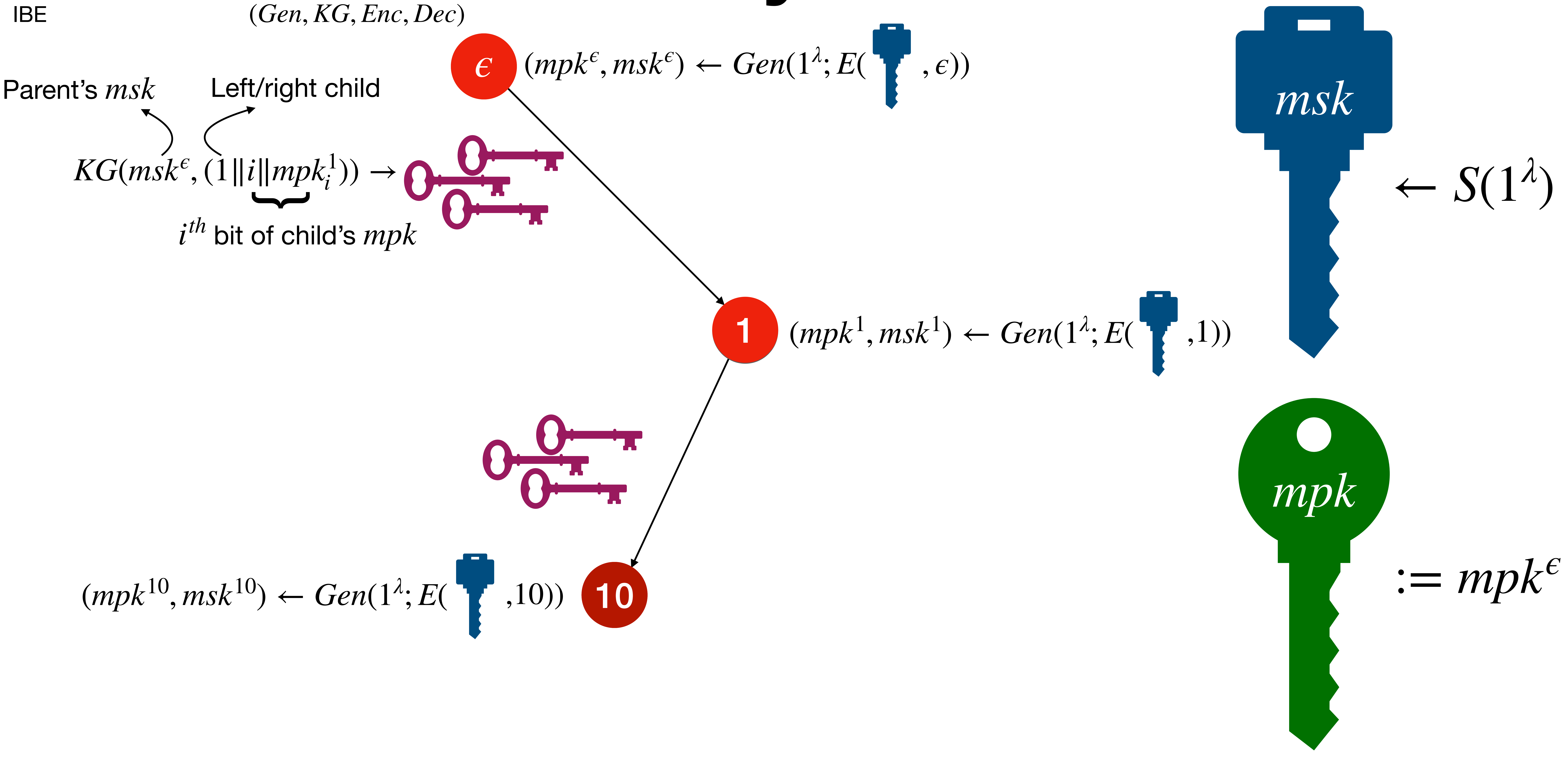
$\leftarrow S(1^\lambda)$

$:= mpk^\epsilon$

KeyGen

[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)



[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

KeyGen

ϵ $(mpk^{\epsilon}, msk^{\epsilon}) \leftarrow Gen(1^{\lambda}; E(\quad, \epsilon))$

Parent's msk Left/right child

$KG(msk^{\epsilon}, (1 \| \underbrace{i \| mpk_i^1}))\rightarrow$

i^{th} bit of child's mpk

1 $(mpk^1, msk^1) \leftarrow Gen(1^{\lambda}; E(\quad, 1))$

$KG(msk^1, (1 \| i \| mpk_i^{10}))\rightarrow$

$(mpk^{10}, msk^{10}) \leftarrow Gen(1^{\lambda}; E(\quad, 10))$ 10

msk $\leftarrow S(1^{\lambda})$

mpk $:= mpk^{\epsilon}$

KeyGen

[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

ϵ $(mpk^\epsilon, msk^\epsilon) \leftarrow Gen(1^\lambda; E(\;, \epsilon))$

Parent's msk Left/right child

$KG(msk^\epsilon, (1\|\underbrace{i\|mpk_i^1}_{i^{th} \text{ bit of child's } mpk})) \rightarrow$

1 $(mpk^1, msk^1) \leftarrow Gen(1^\lambda; E(\;, 1))$

$KG(msk^1, (1\|i\|mpk_i^{10})) \rightarrow$

$(mpk^{10}, msk^{10}) \leftarrow Gen(1^\lambda; E(\;, 10))$ 10

$msk \leftarrow S(1^\lambda)$

$mpk := mpk^\epsilon$

KeyGen

[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

ϵ $(mpk^{\epsilon}, msk^{\epsilon}) \leftarrow Gen(1^{\lambda}; E(\quad, \epsilon))$

Parent's msk Left/right child

$KG(msk^{\epsilon}, (1 \| \underbrace{i \|}_{} mpk_i^{1})) \rightarrow$

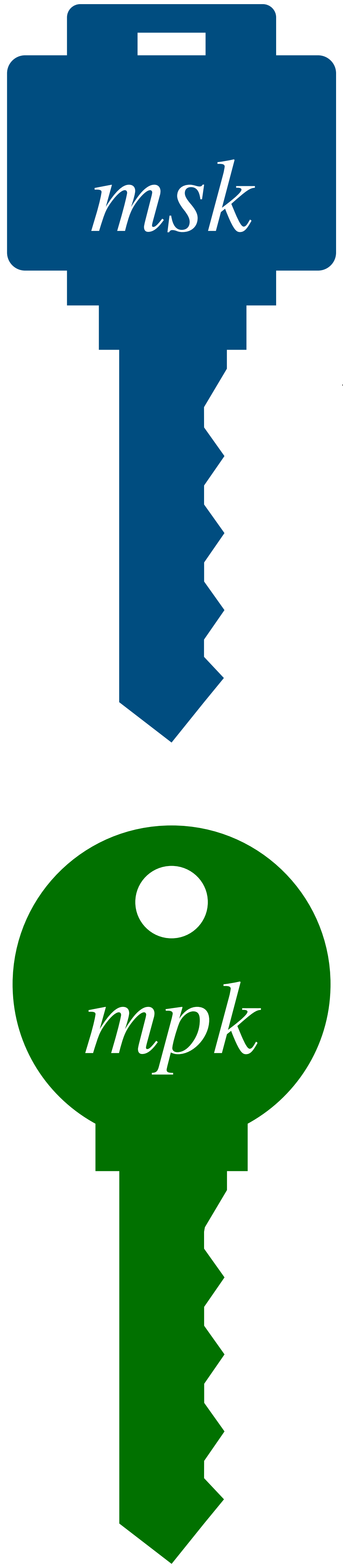
i^{th} bit of child's mpk

1 $(mpk^{1}, msk^{1}) \leftarrow Gen(1^{\lambda}; E(\quad, 1))$

$KG(msk^{1}, (1 \| i \| mpk_i^{10})) \rightarrow$

$(mpk^{10}, msk^{10}) \leftarrow Gen(1^{\lambda}; E(\quad, 10))$ 10

$KG(msk^{10}, \epsilon) \rightarrow$



$\leftarrow S(1^{\lambda})$

$:= mpk^{\epsilon}$

KeyGen

[Döttling-Garg'17]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

ϵ $(mpk^{\epsilon}, msk^{\epsilon}) \leftarrow Gen(1^{\lambda}; E(\quad, \epsilon))$

Parent's msk

Left/right child

$KG(msk^{\epsilon}, (1 \| \underbrace{i}_{i^{th} \text{ bit of child's } mpk} \| mpk_i^1)) \rightarrow$

1 $(mpk^1, msk^1) \leftarrow Gen(1^{\lambda}; E(\quad, 1))$

$KG(msk^1, (1 \| i \| mpk_i^{10})) \rightarrow$

$(mpk^{10}, msk^{10}) \leftarrow Gen(1^{\lambda}; E(\quad, 10))$ 10

$KG(msk^{10}, \epsilon) \rightarrow$

$msk \leftarrow S(1^{\lambda})$

$mpk := mpk^{\epsilon}$

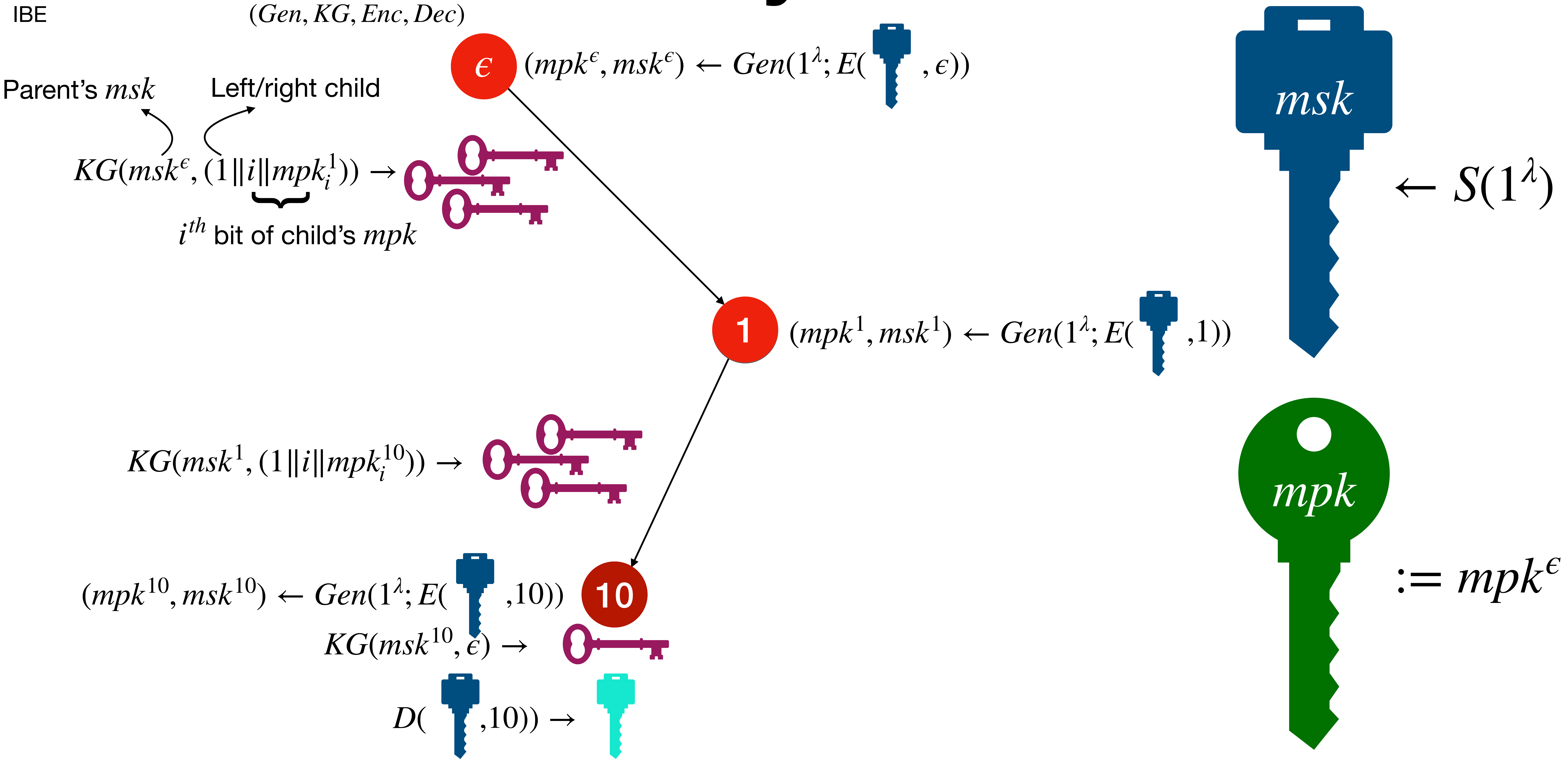
KeyGen

[Döttling-Garg'17]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)



Encryption

[**Döttling-Garg'17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

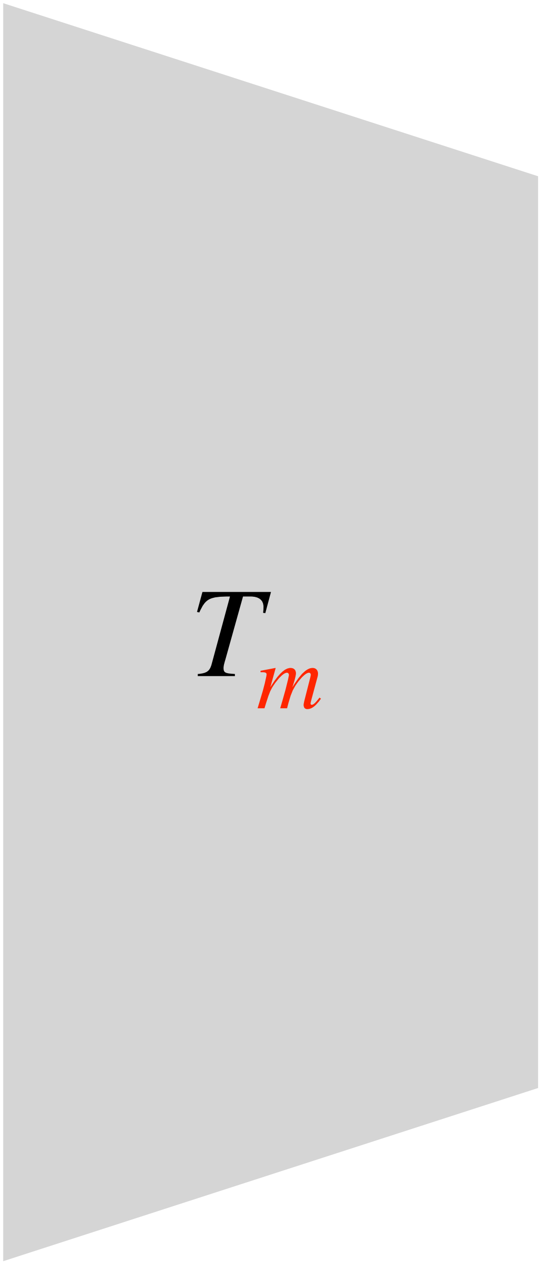
Encryption

 $:= mpk^{\epsilon}$

[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Encryption

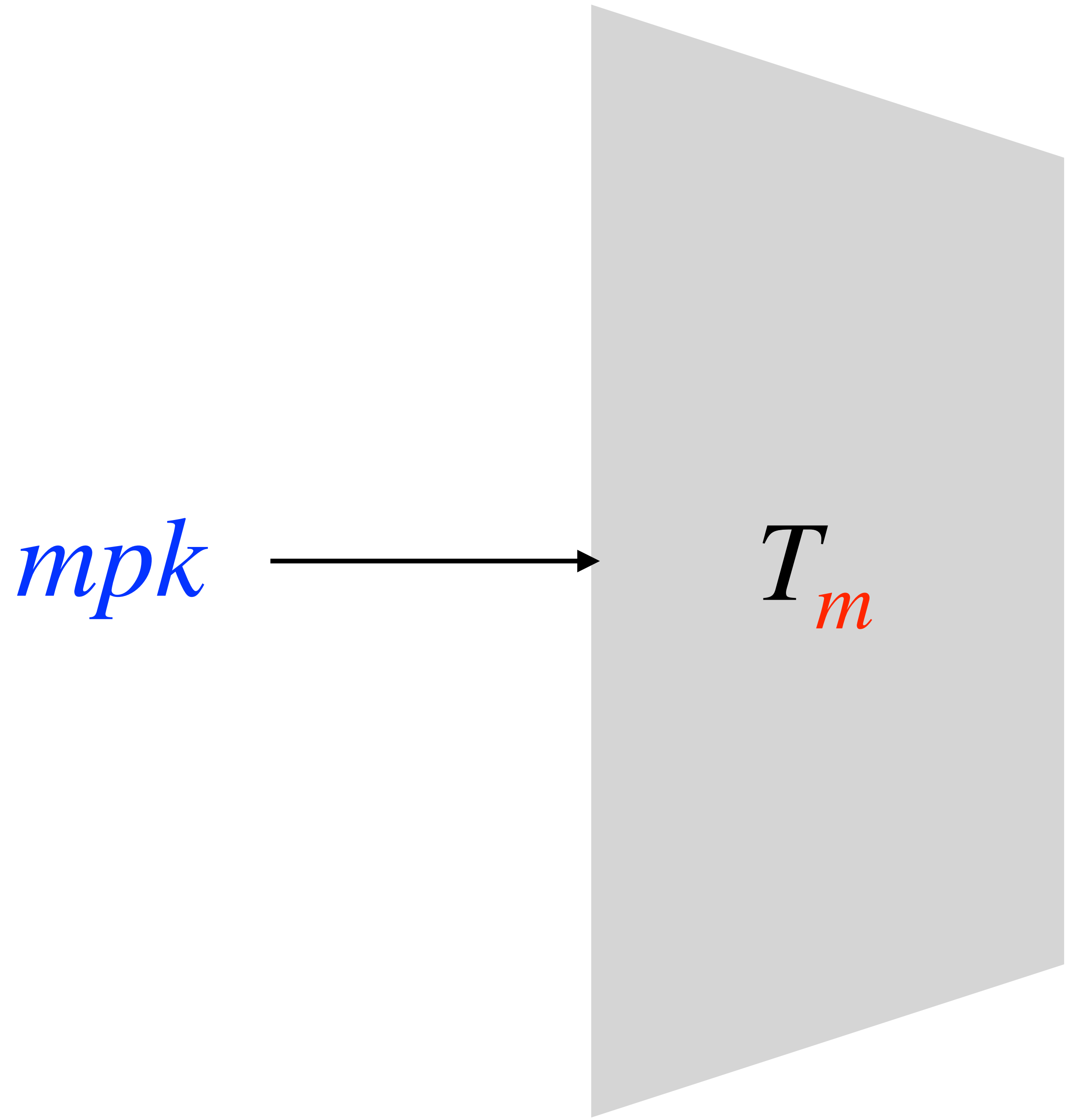


 $:= mpk^{\epsilon}$

[**Döttling-Garg’17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Encryption



[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Encryption

$$mpk \longrightarrow T_m \longrightarrow Enc(mpk, \epsilon, m)$$

 $:= mpk^{\epsilon}$

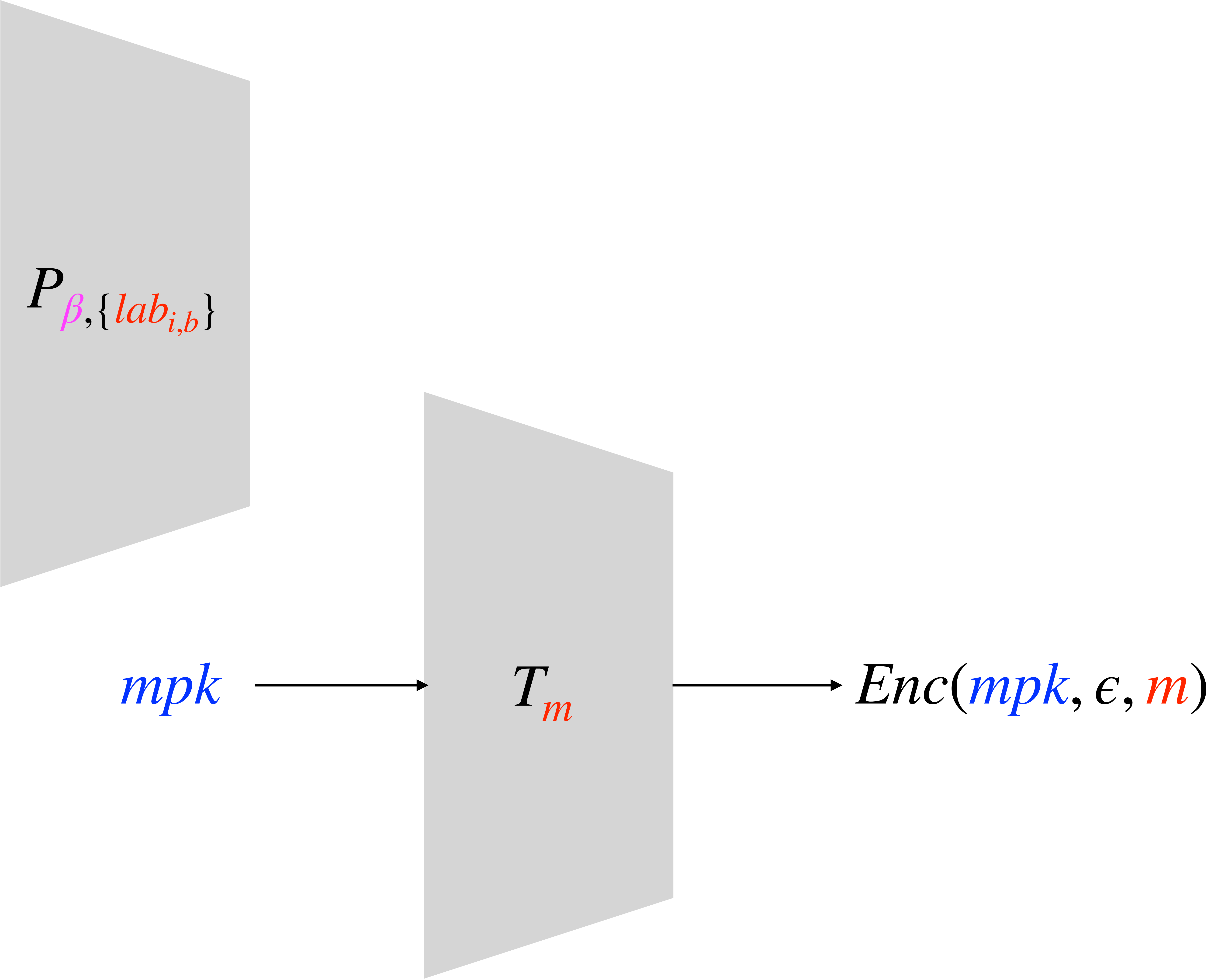
[**Döttling-Garg’17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Encryption

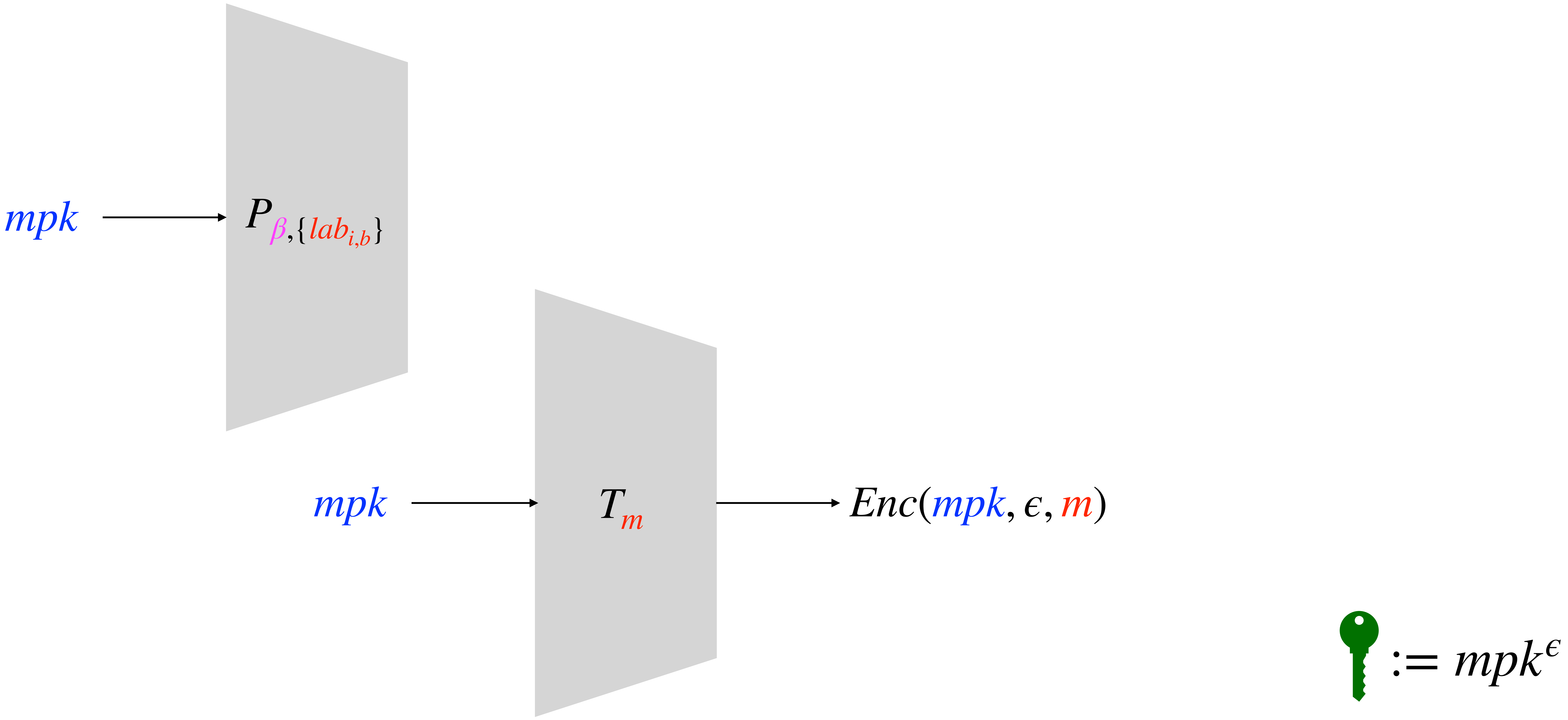


$:= mpk^{\epsilon}$

[**Döttling-Garg'17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Encryption



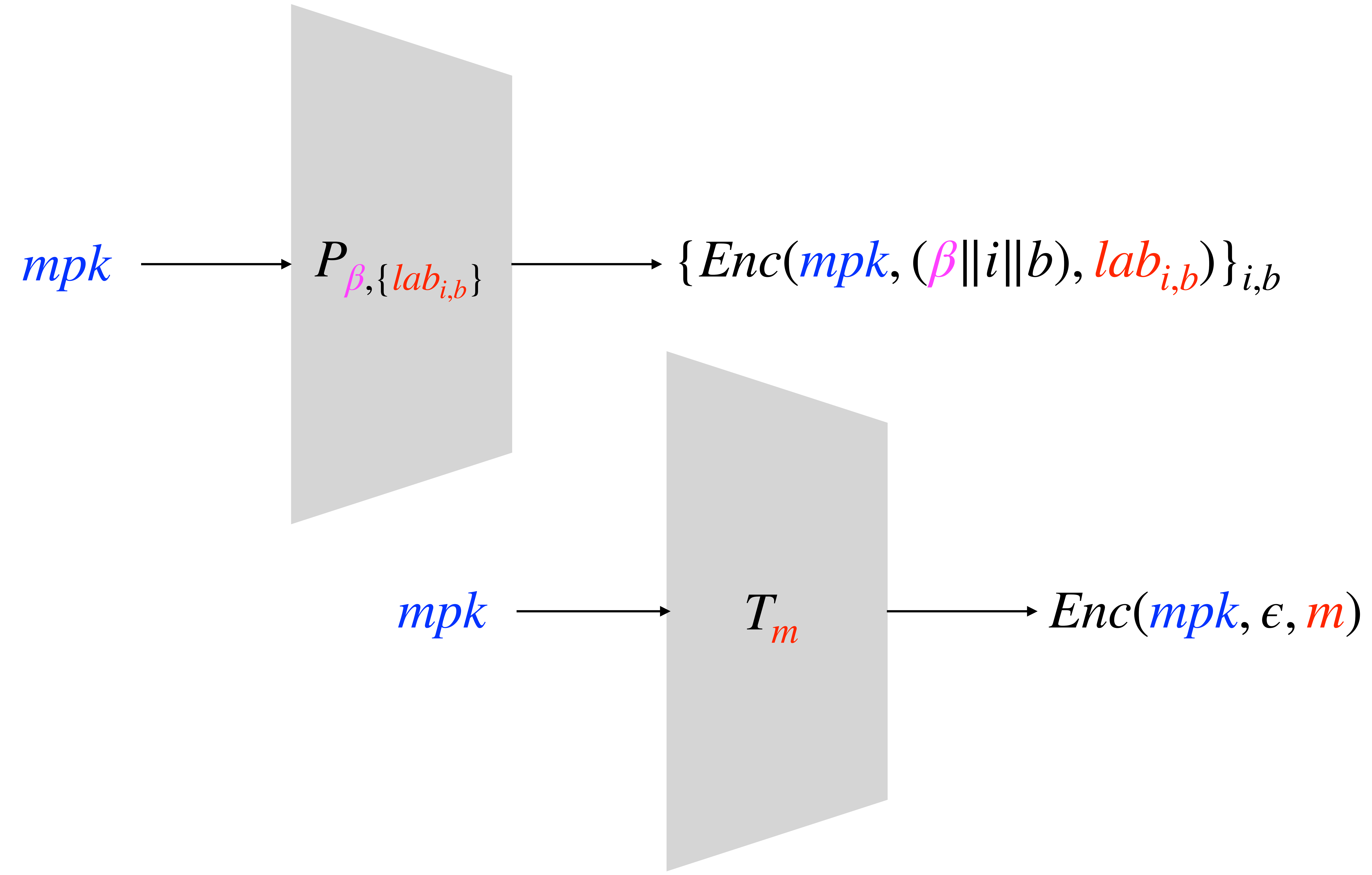
[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Encryption



 $:= mpk^{\epsilon}$

[Döttling-Garg'17]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Encryption

$$mpk \longrightarrow P_{\beta,\{lab_{i,b}\}} \longrightarrow \{Enc(mpk, (\beta \| i \| b), lab_{i,b})\}_{i,b}$$

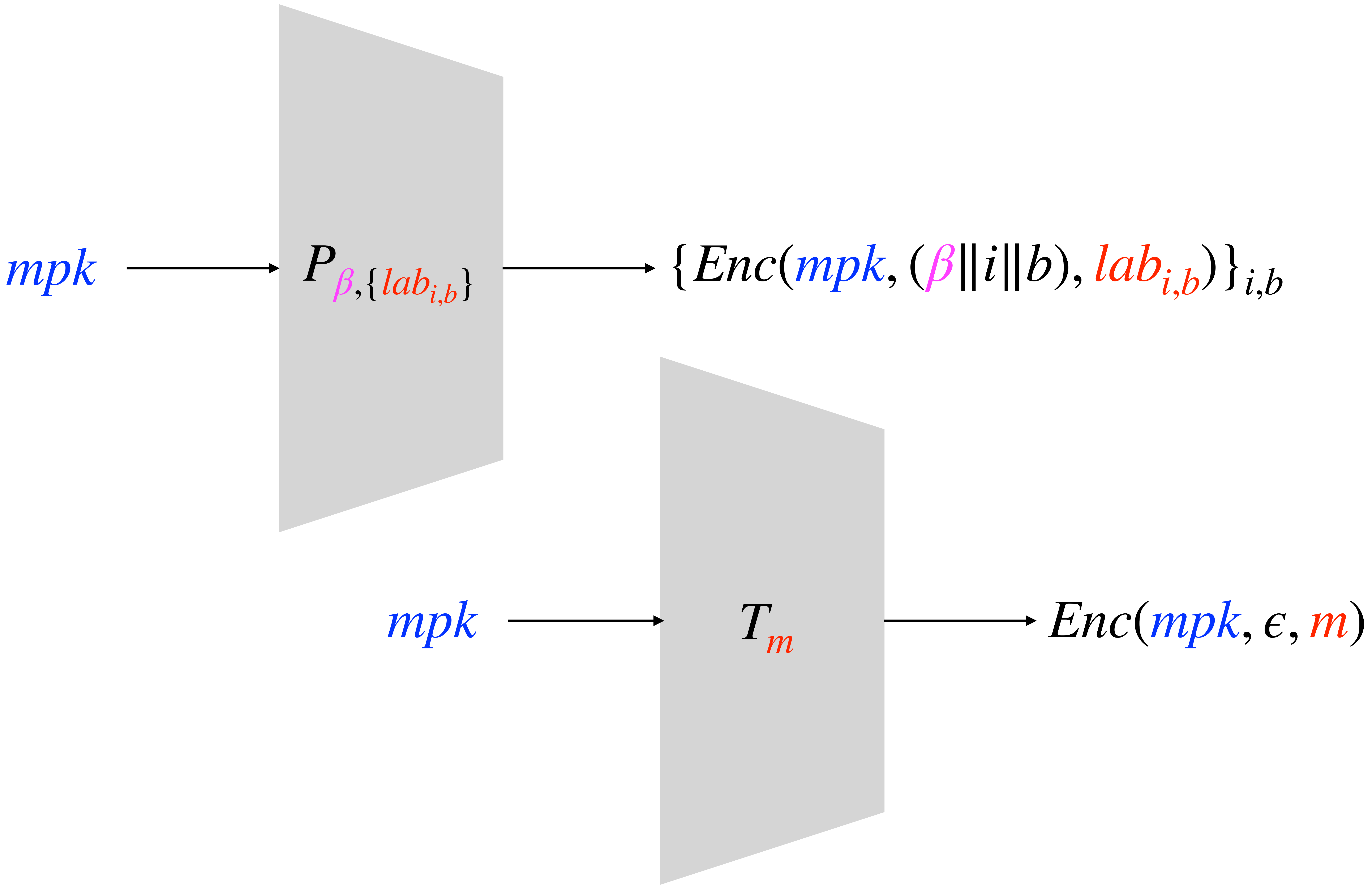
$$mpk \longrightarrow T_m \longrightarrow Enc(mpk, \epsilon, m)$$

$id = 10,$  $:= mpk^{\epsilon}$

Encryption

[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)



$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$id = 10,$  $:= mpk^{\epsilon}$

[**Döttling-Garg'17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Encryption

$mpk \longrightarrow P_{\beta,\{lab_{i,b}\}} \longrightarrow \{Enc(mpk, (\beta \| i \| b), lab_{i,b})\}_{i,b}$

$mpk \longrightarrow T_m \longrightarrow Enc(mpk, \epsilon, m)$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$id = 10,$  $:= mpk^{\epsilon}$

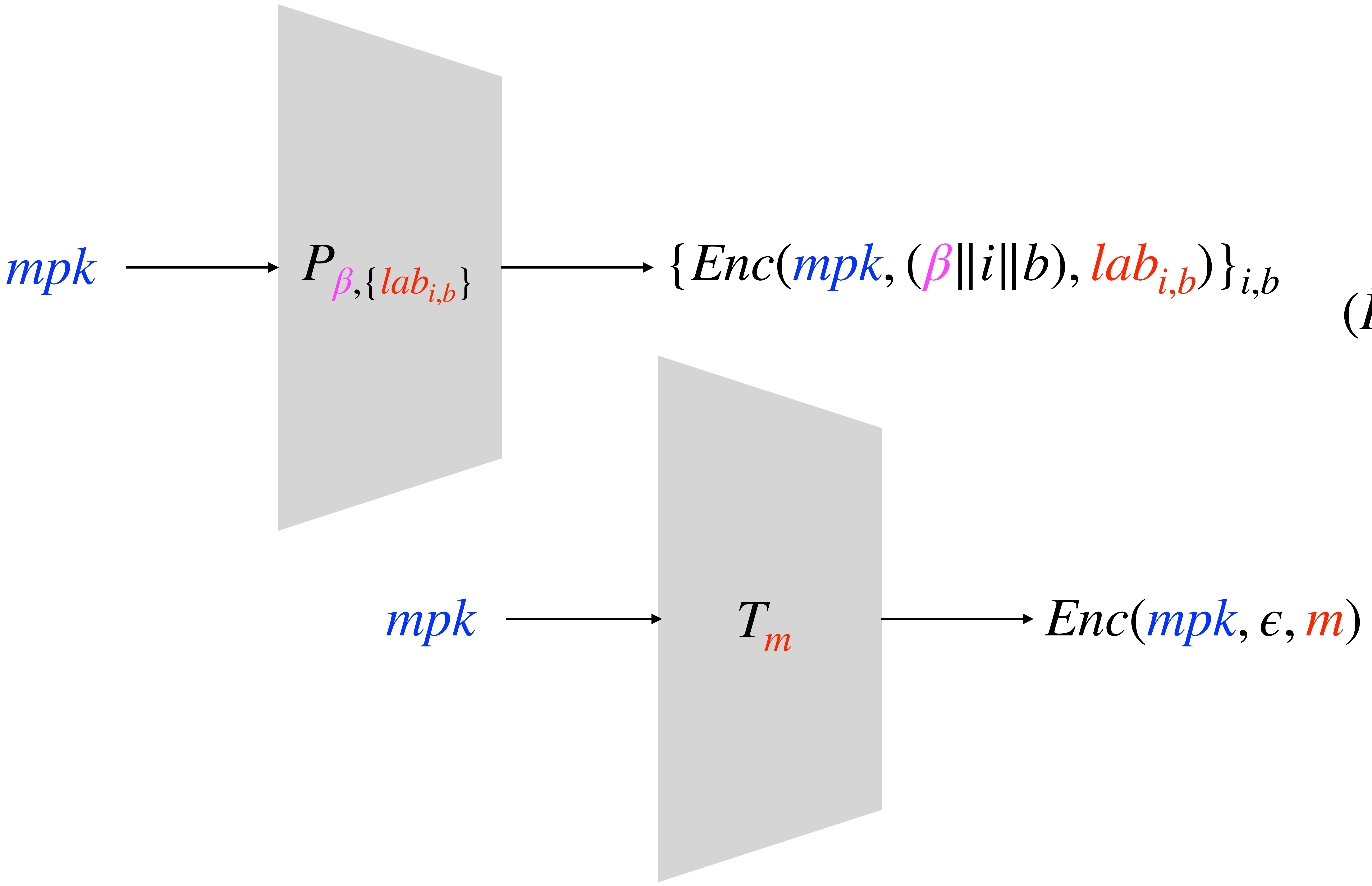
[**Döttling-Garg’17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Encryption



$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$id = 10,$  $:= mpk^{\epsilon}$

[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Encryption

$mpk \longrightarrow P_{\beta,\{lab_{i,b}\}} \longrightarrow \{Enc(mpk, (\beta \| i \| b), lab_{i,b})\}_{i,b}$

$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$

$mpk \longrightarrow T_m \longrightarrow Enc(mpk, \epsilon, m)$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$id = 10,$ $:= mpk^{\epsilon}$

[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Encryption

$(\tilde{Q}, LAB_{i,b}) \leftarrow Garb(P_{'1', \{Lab_{i,b}\}})$

$mpk \longrightarrow P_{\beta, \{lab_{i,b}\}} \longrightarrow \{Enc(mpk, (\beta \| i \| b), lab_{i,b})\}_{i,b}$

$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0', \{lab_{i,b}\}})$

$mpk \longrightarrow T_m \longrightarrow Enc(mpk, \epsilon, m)$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$id = 10,$  $:= mpk^{\epsilon}$

[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Encryption

$Return(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$

$(\tilde{Q}, LAB_{i,b}) \leftarrow Garb(P_{'1',\{Lab_{i,b}\}})$

$mpk \longrightarrow P_{\beta,\{lab_{i,b}\}} \longrightarrow \{Enc(mpk, (\beta\|i\|b), lab_{i,b})\}_{i,b}$

$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$

$mpk \longrightarrow T_m \longrightarrow Enc(mpk, \epsilon, m)$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$id = 10,$ $:= mpk^{\epsilon}$

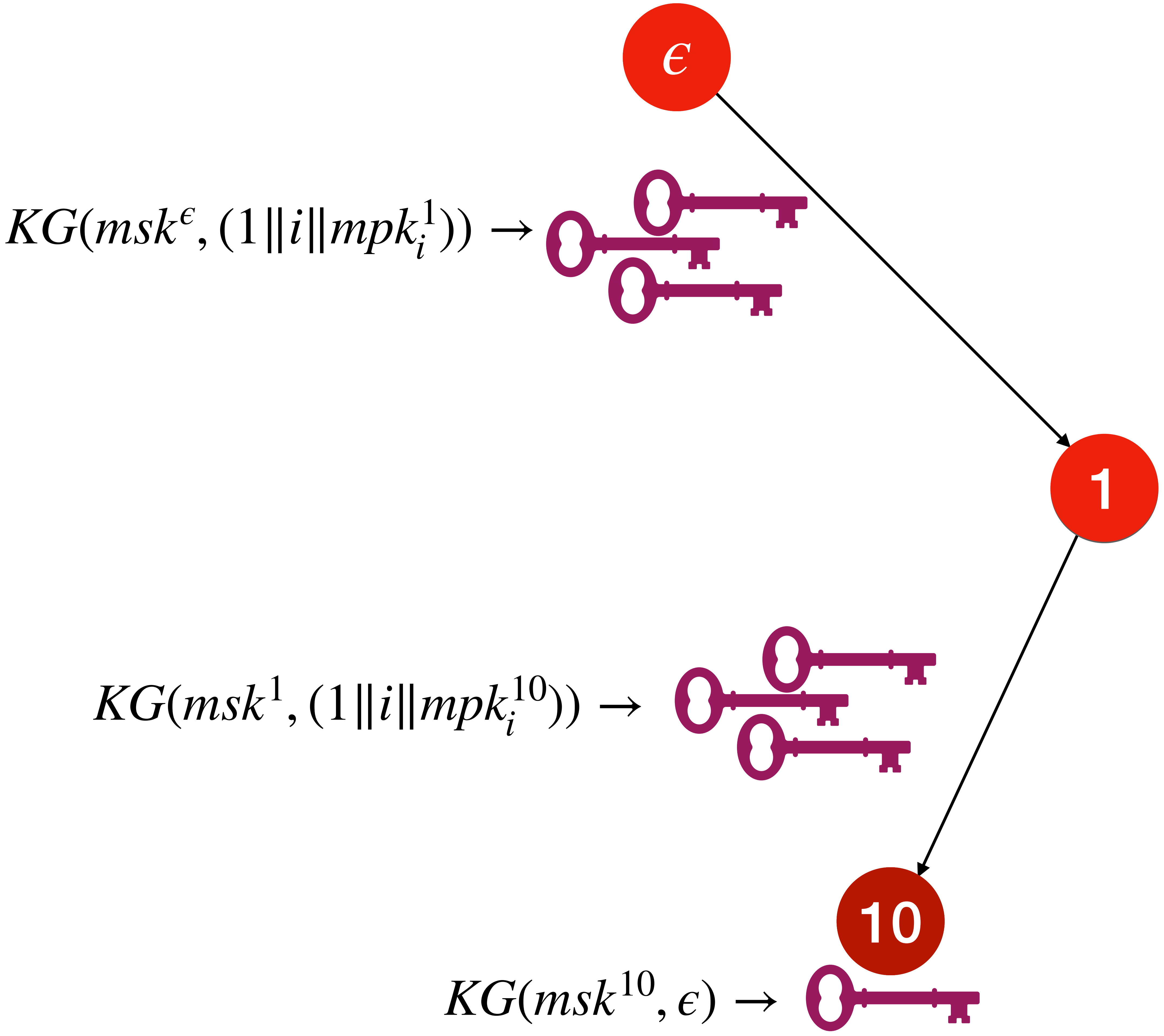
[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Decryption

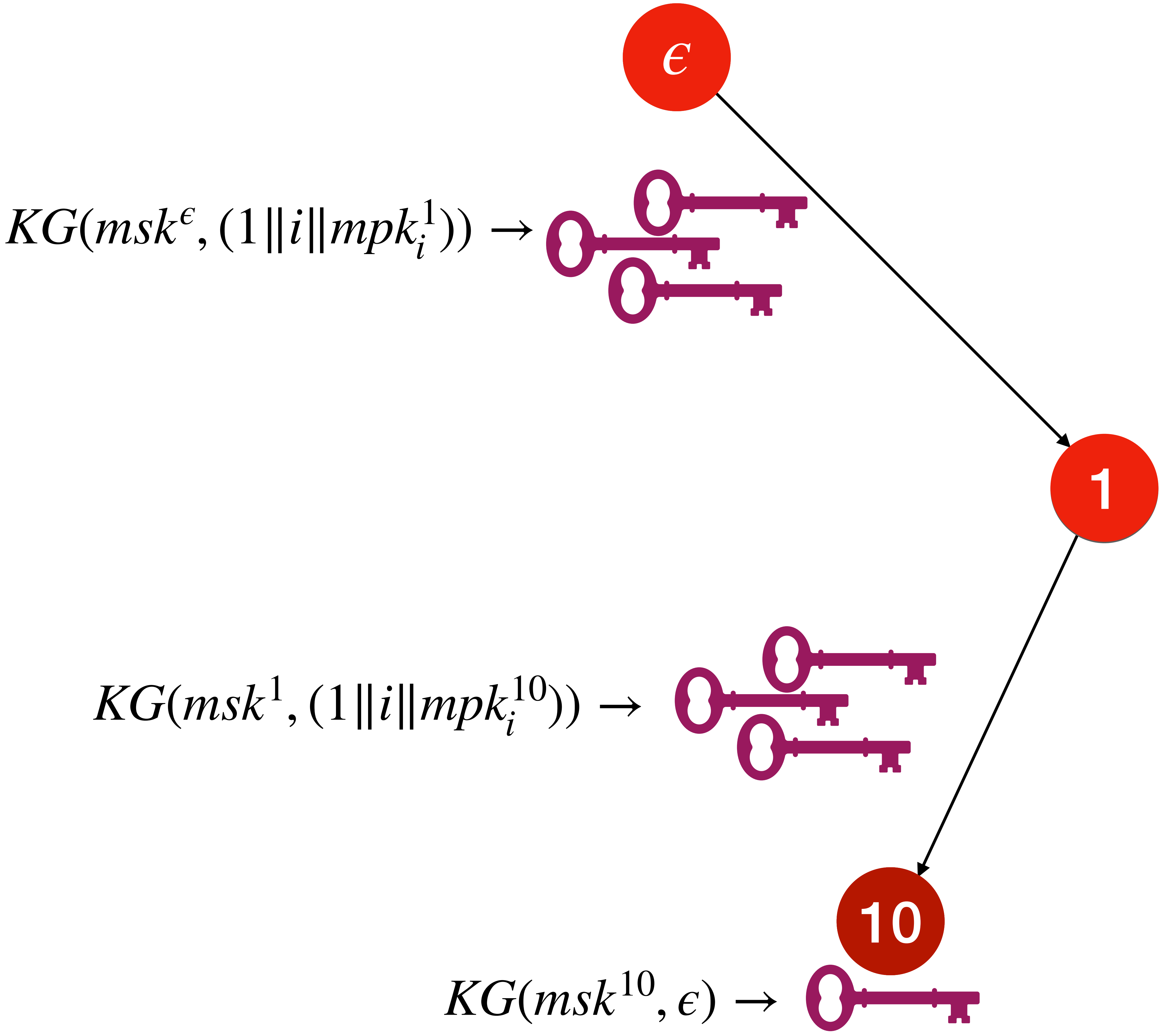


[**Döttling-Garg’17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$

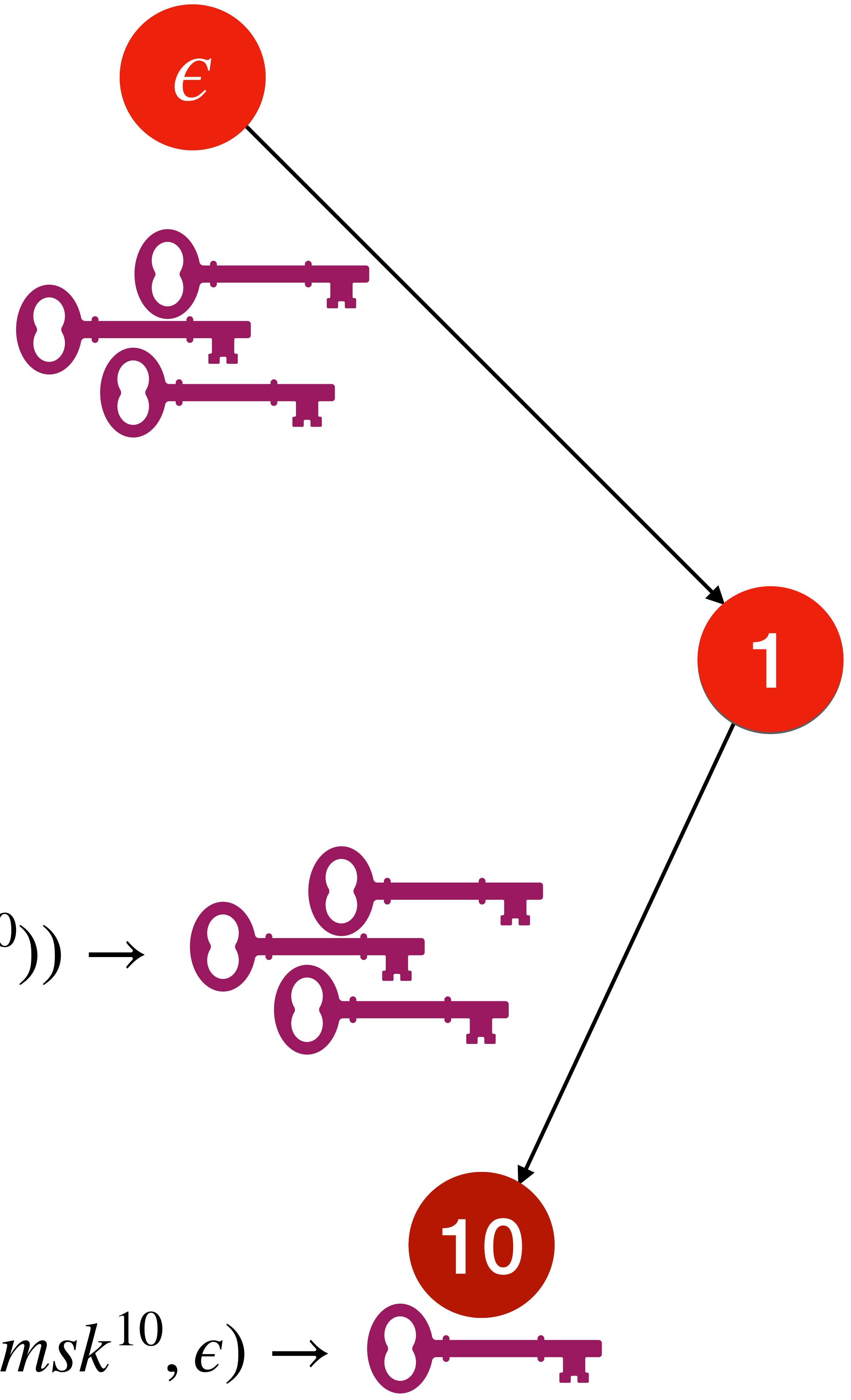


[Döttling-Garg'17]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$



$KG(msk^{\epsilon}, (1\|i\|mpk_i^1)) \rightarrow$

$KG(msk^{1}, (1\|i\|mpk_i^{10})) \rightarrow$

$KG(msk^{10}, \epsilon) \rightarrow$

[**Döttling-Garg’17**]

Delegatable PRF (S, E, D)

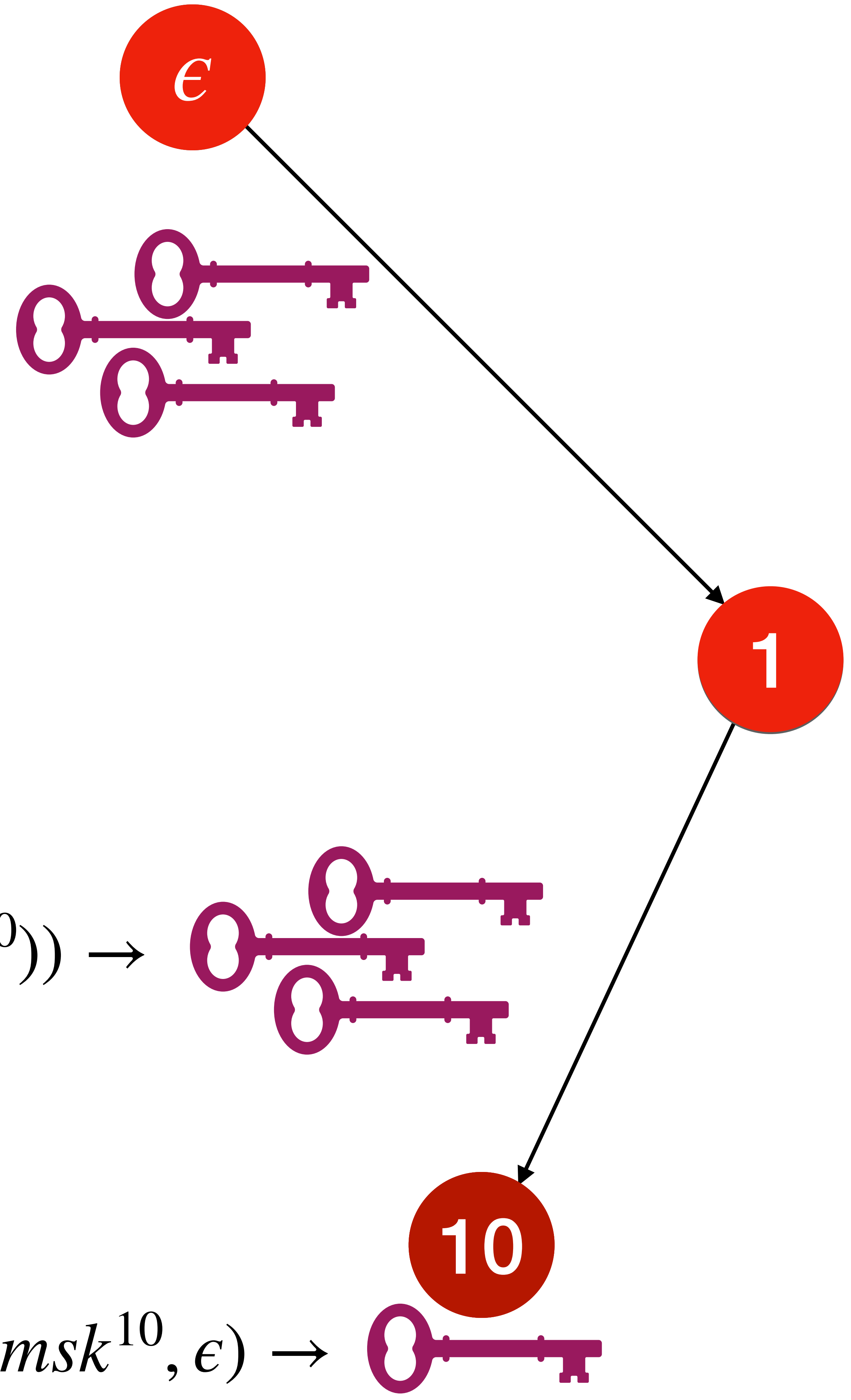
Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Decryption

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$$

$Eval$



$KG(msk^\epsilon, (1\|i\|mpk_i^1)) \rightarrow$

$KG(msk^1, (1\|i\|mpk_i^{10})) \rightarrow$

$KG(msk^{10}, \epsilon) \rightarrow$

[**Döttling-Garg'17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$

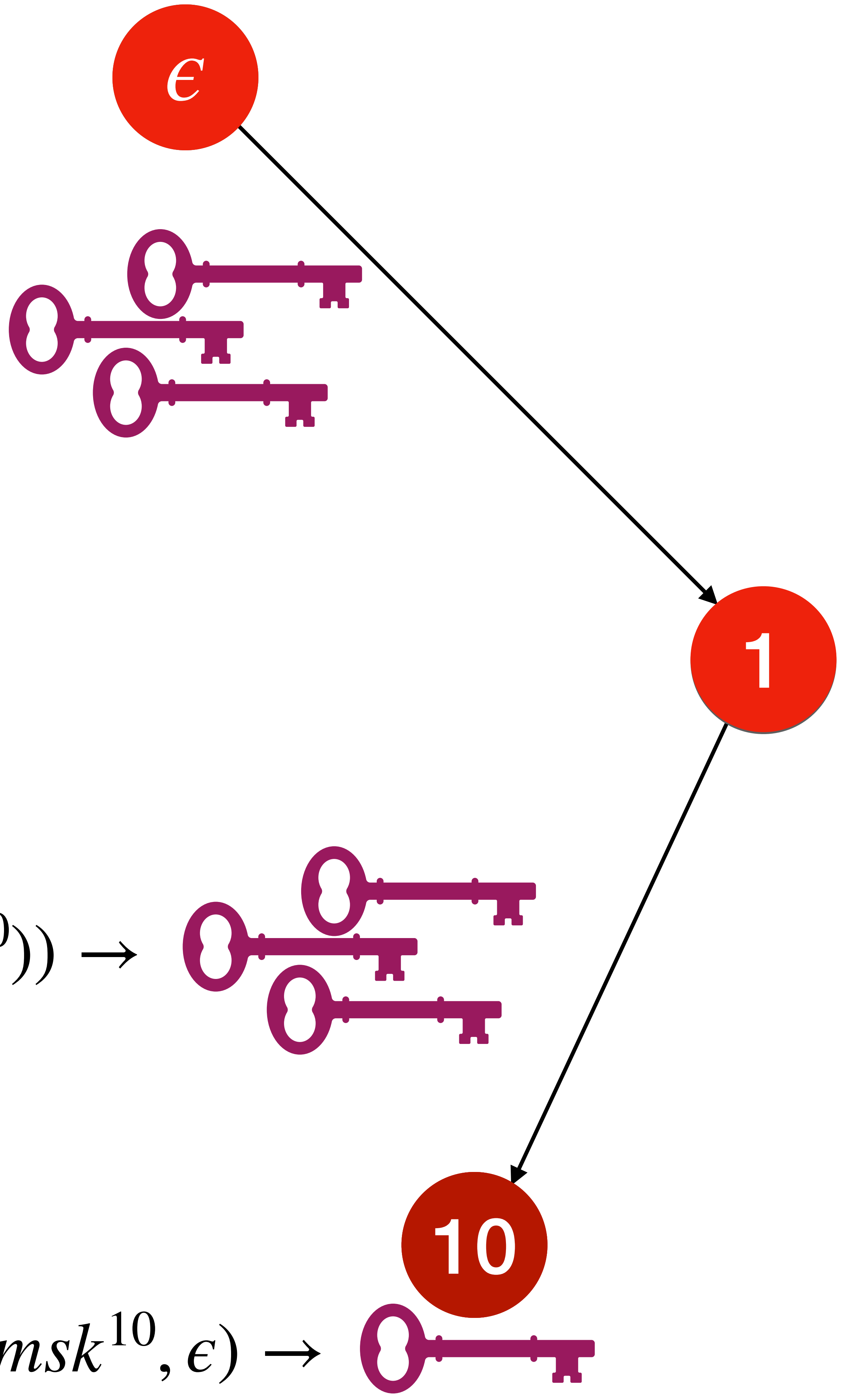
Eval

$CT = Enc(mpk^\epsilon, 1\|i\|b, Lab_{i,b})$

$KG(msk^\epsilon, (1\|i\|mpk_i^1)) \to$

$KG(msk^1, (1\|i\|mpk_i^{10})) \to$

$KG(msk^{10}, \epsilon) \to$



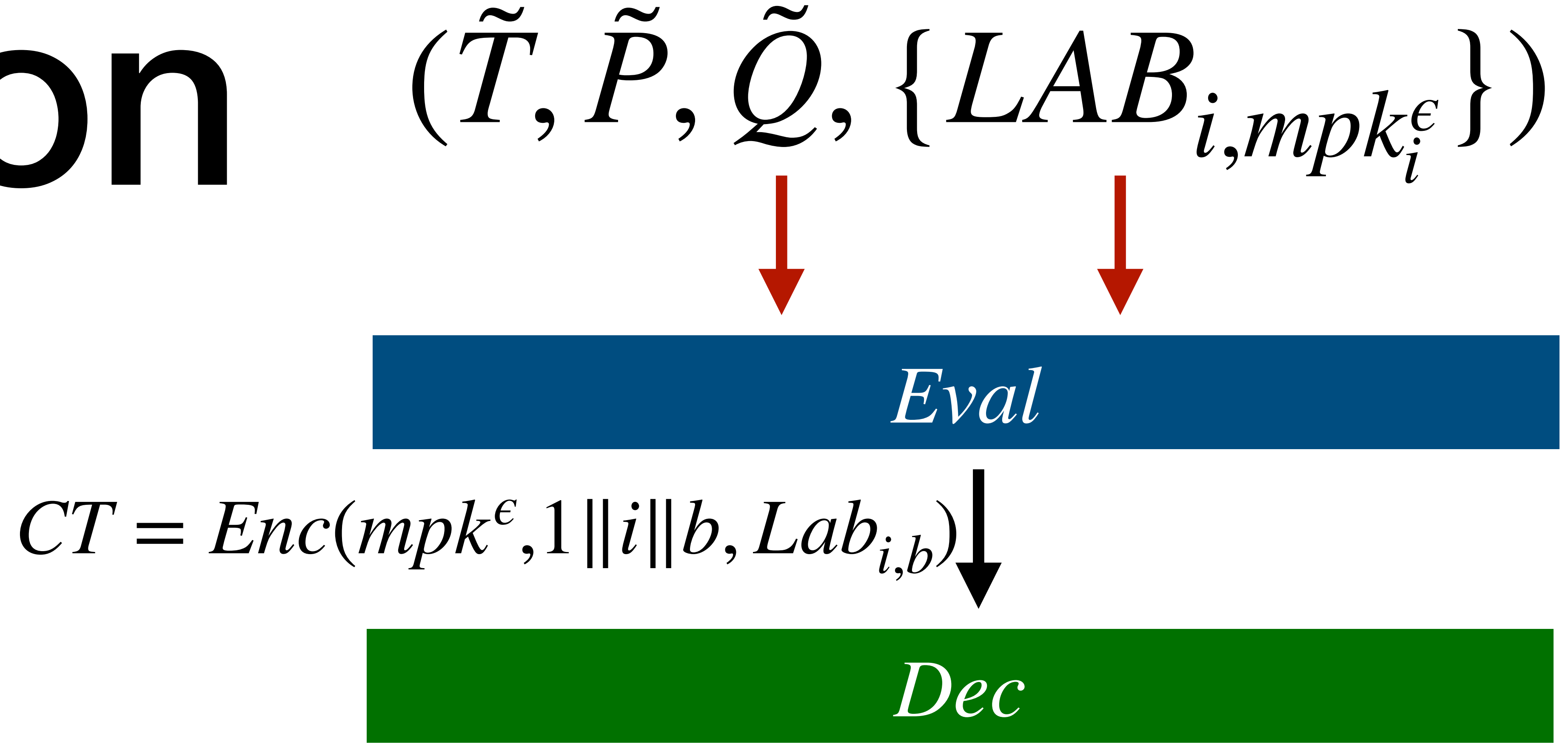
[**Döttling-Garg'17**]

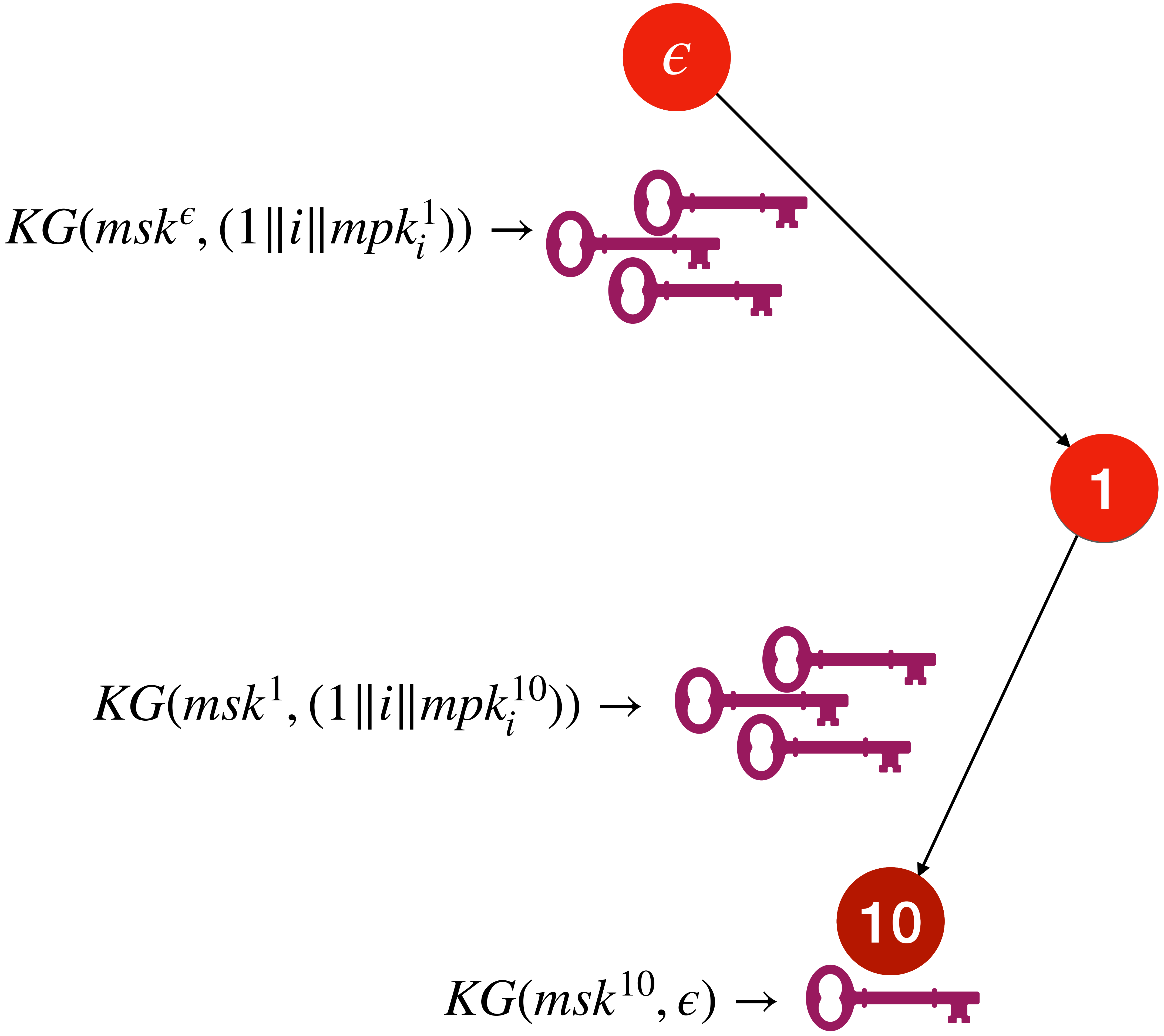
Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Decryption





[Döttling-Garg'17]

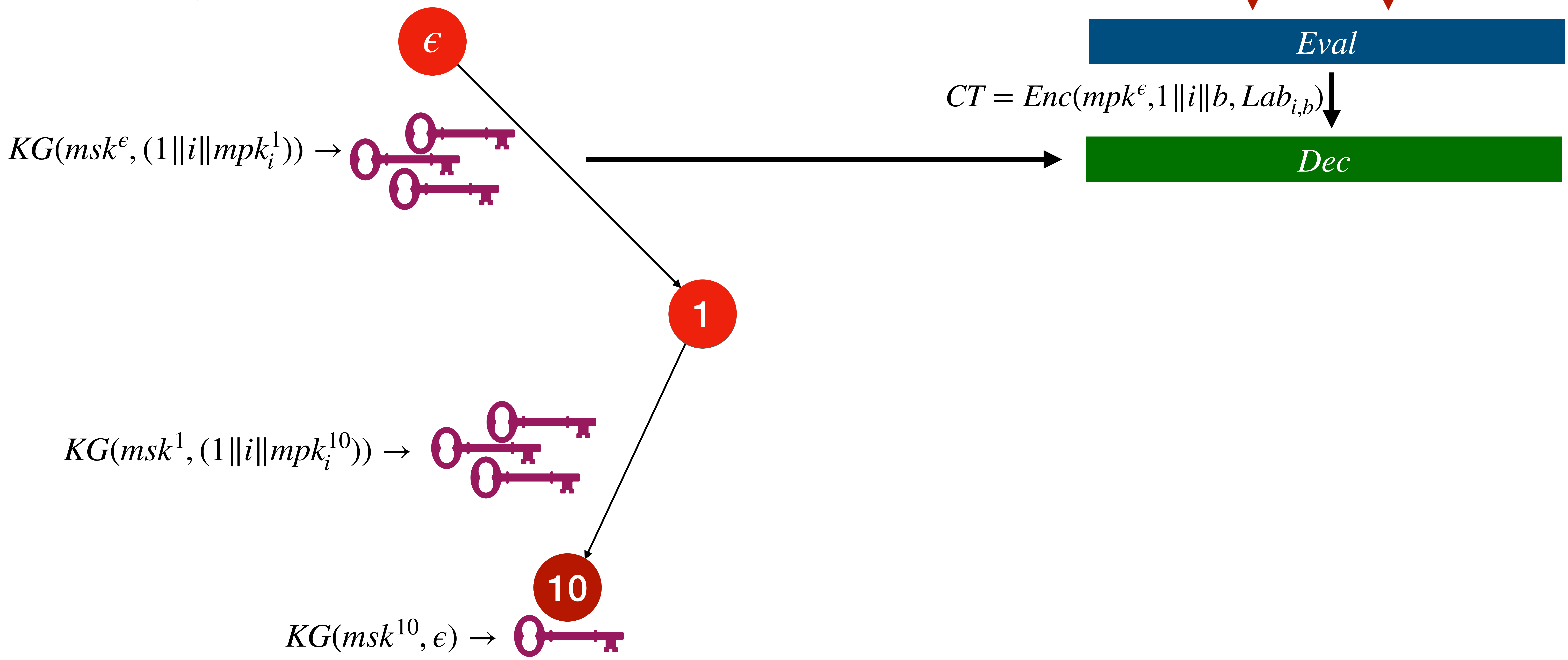
Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$



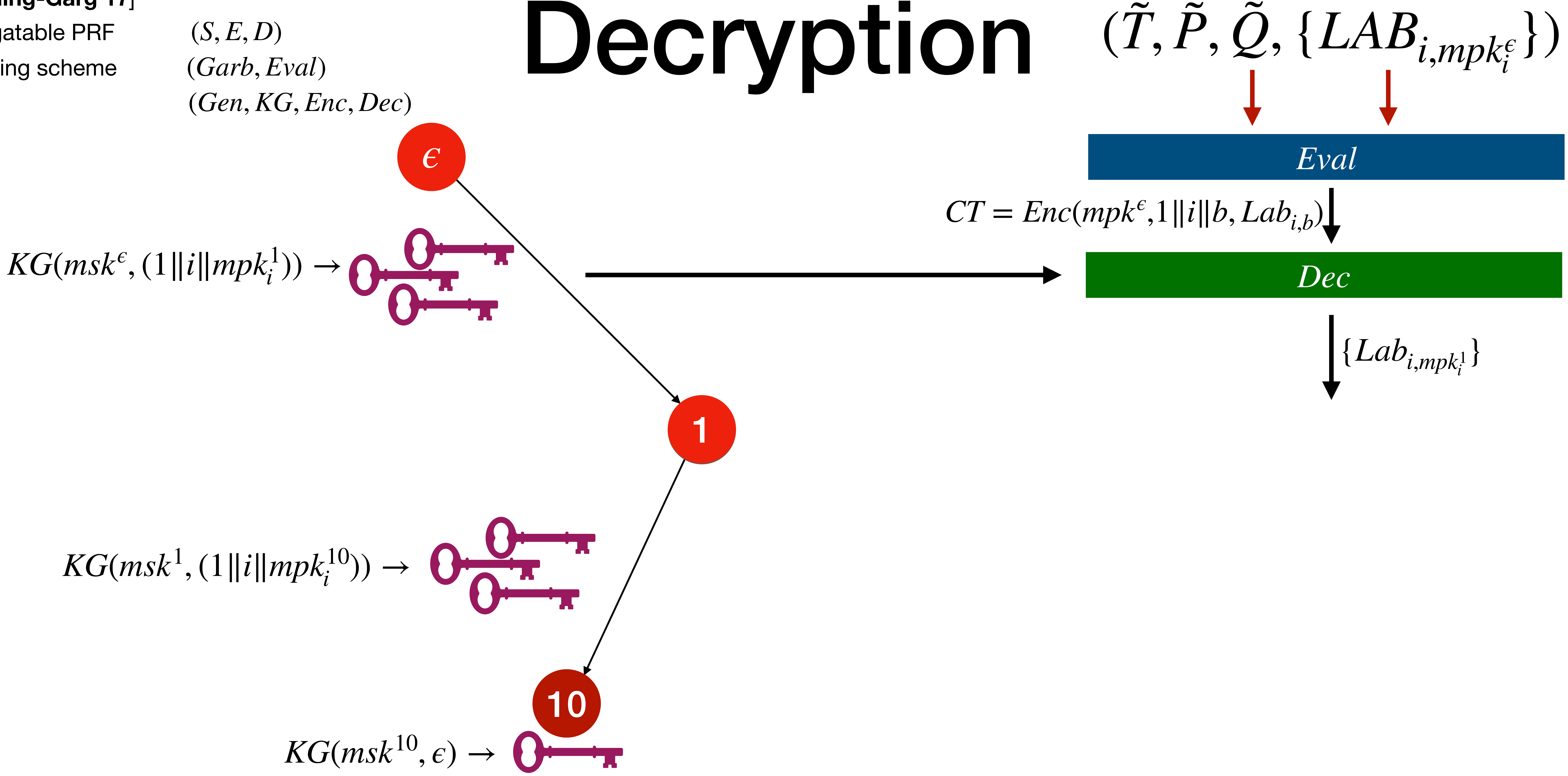
Decryption

[Döttling-Garg'17]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

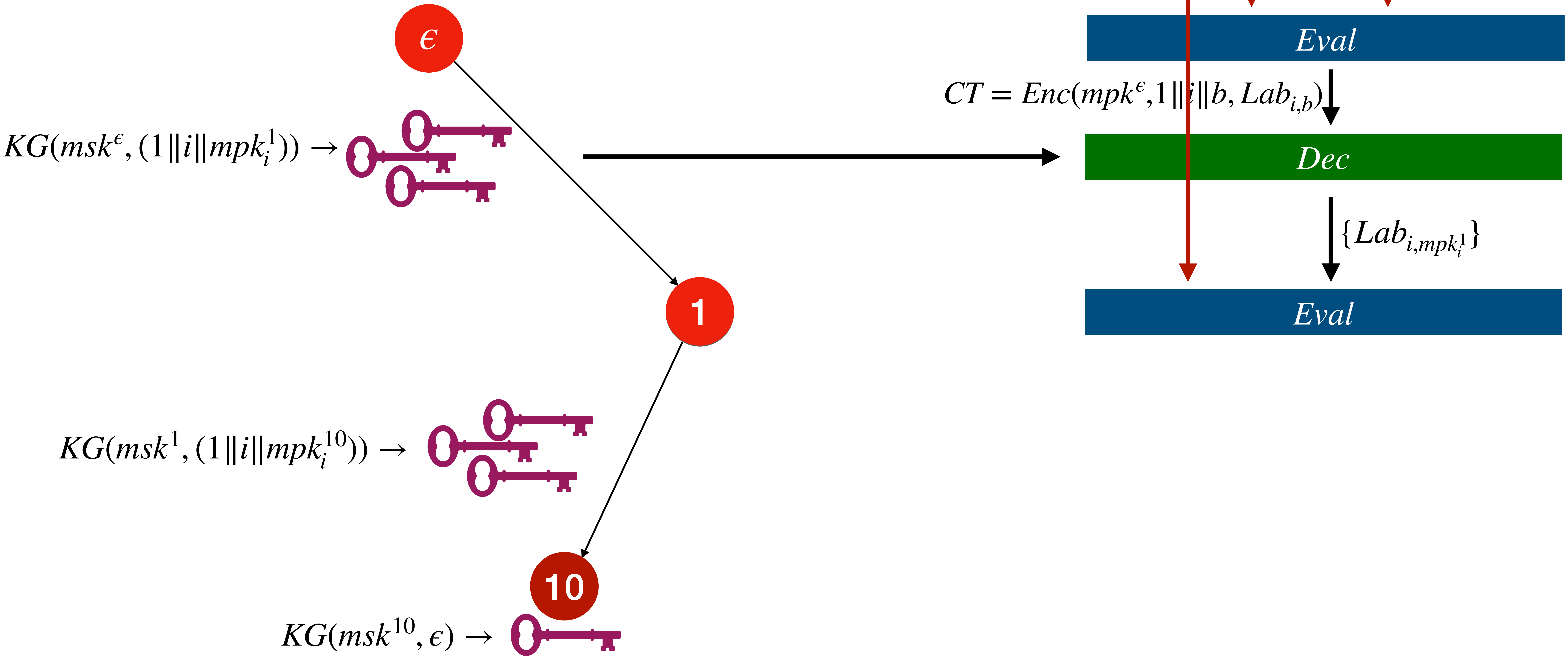


[**Döttling-Garg'17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$



Decryption

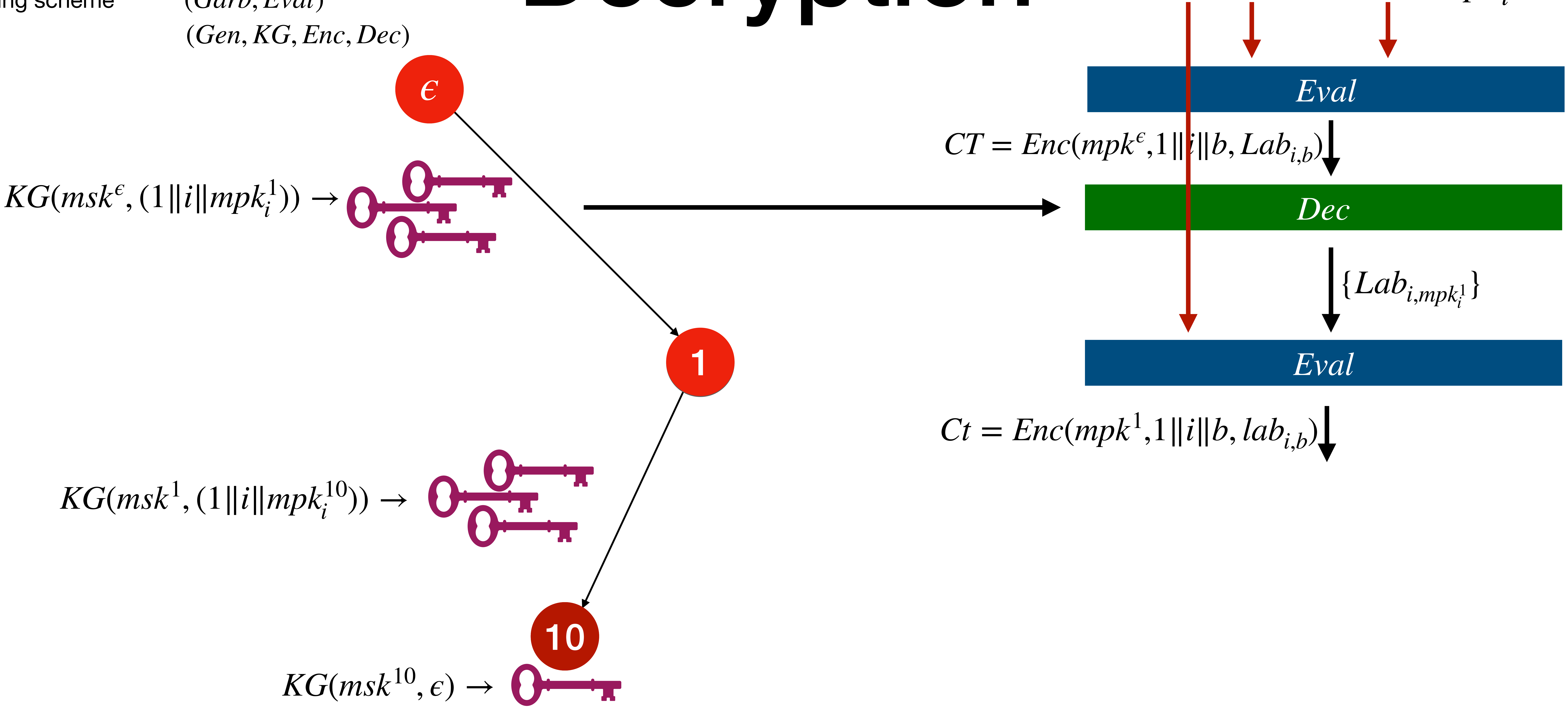
[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$

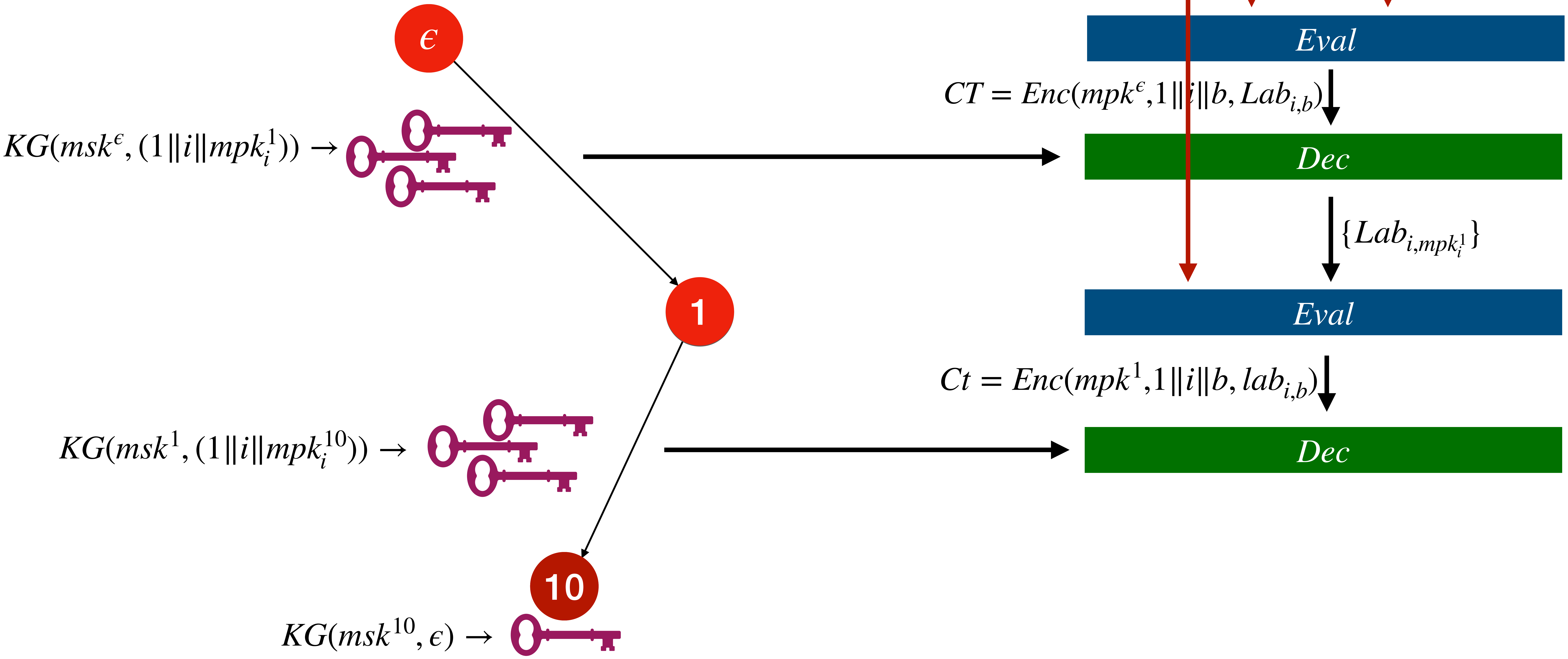


[**Döttling-Garg’17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$

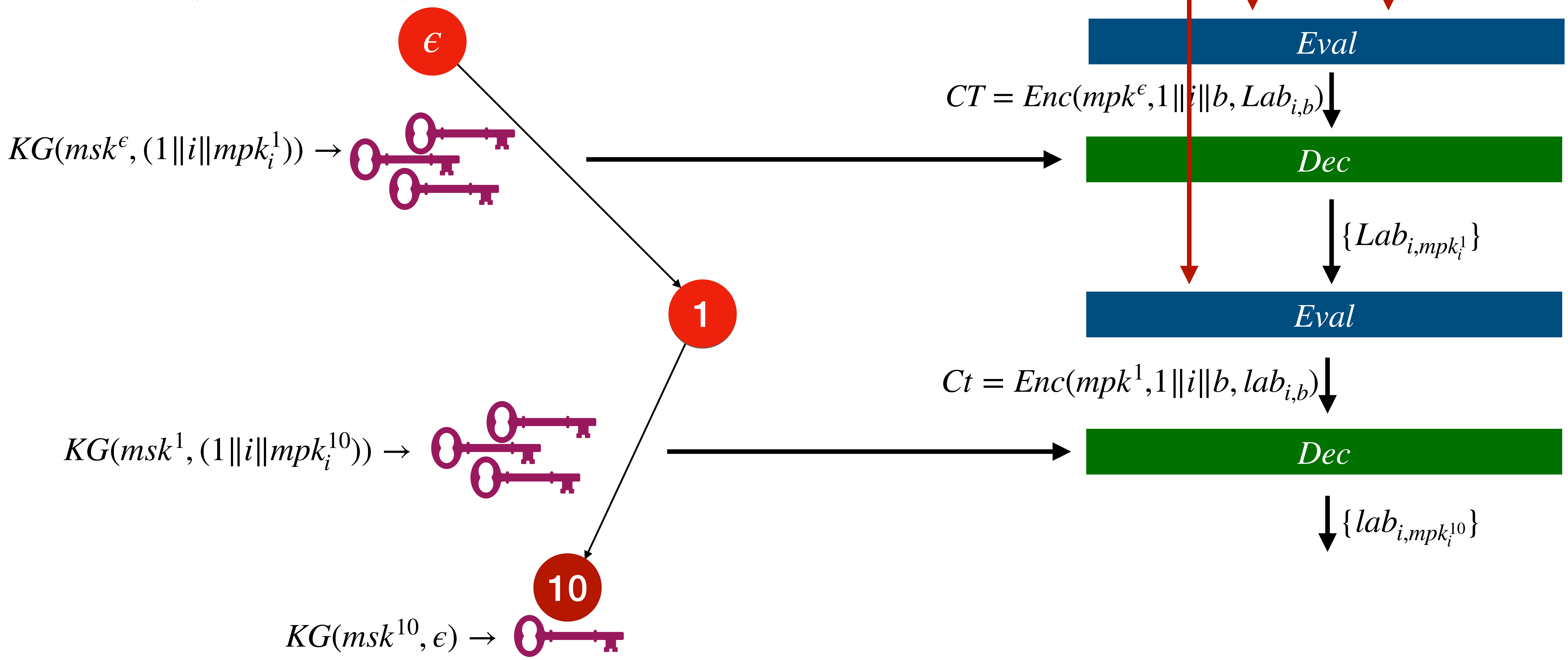


[**Döttling-Garg'17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$



[**Döttling-Garg’17**]

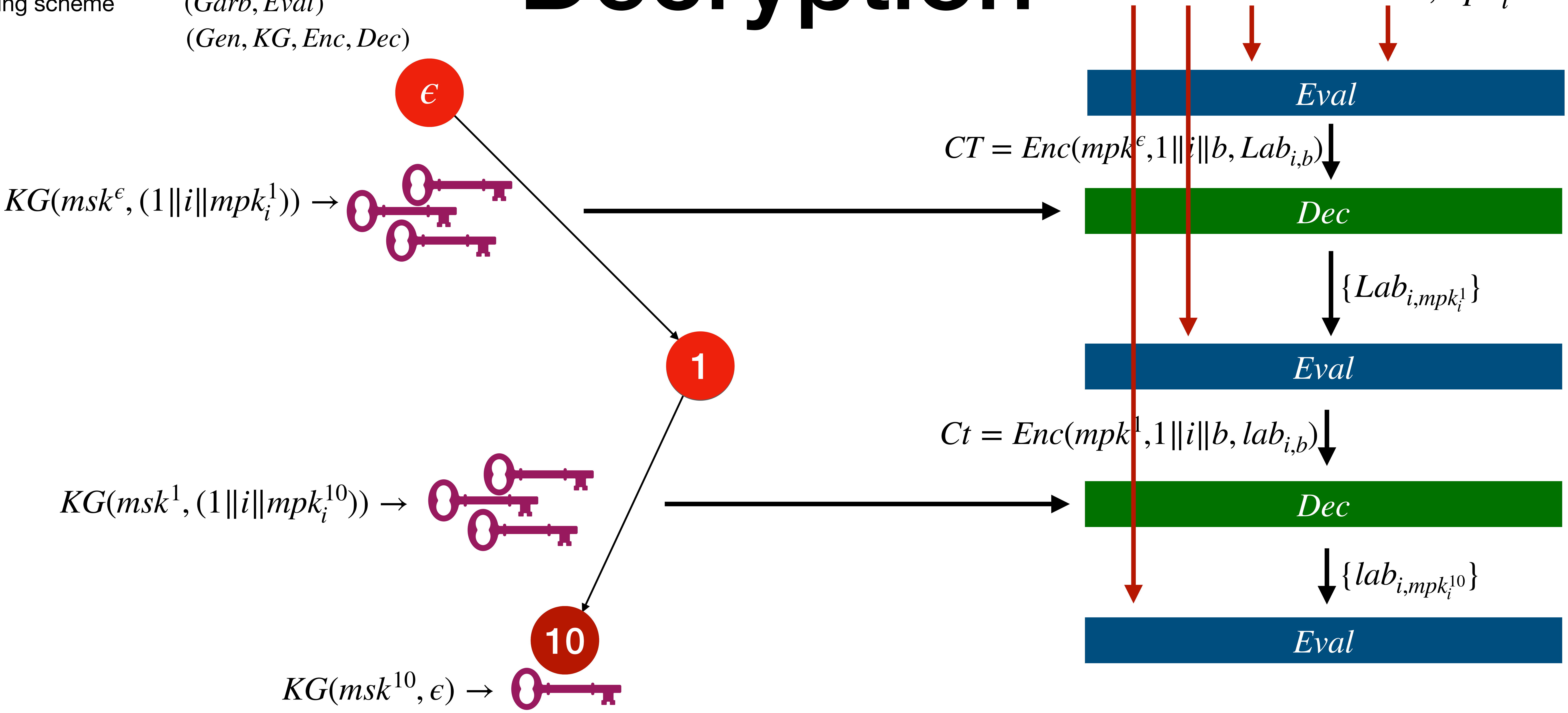
Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$

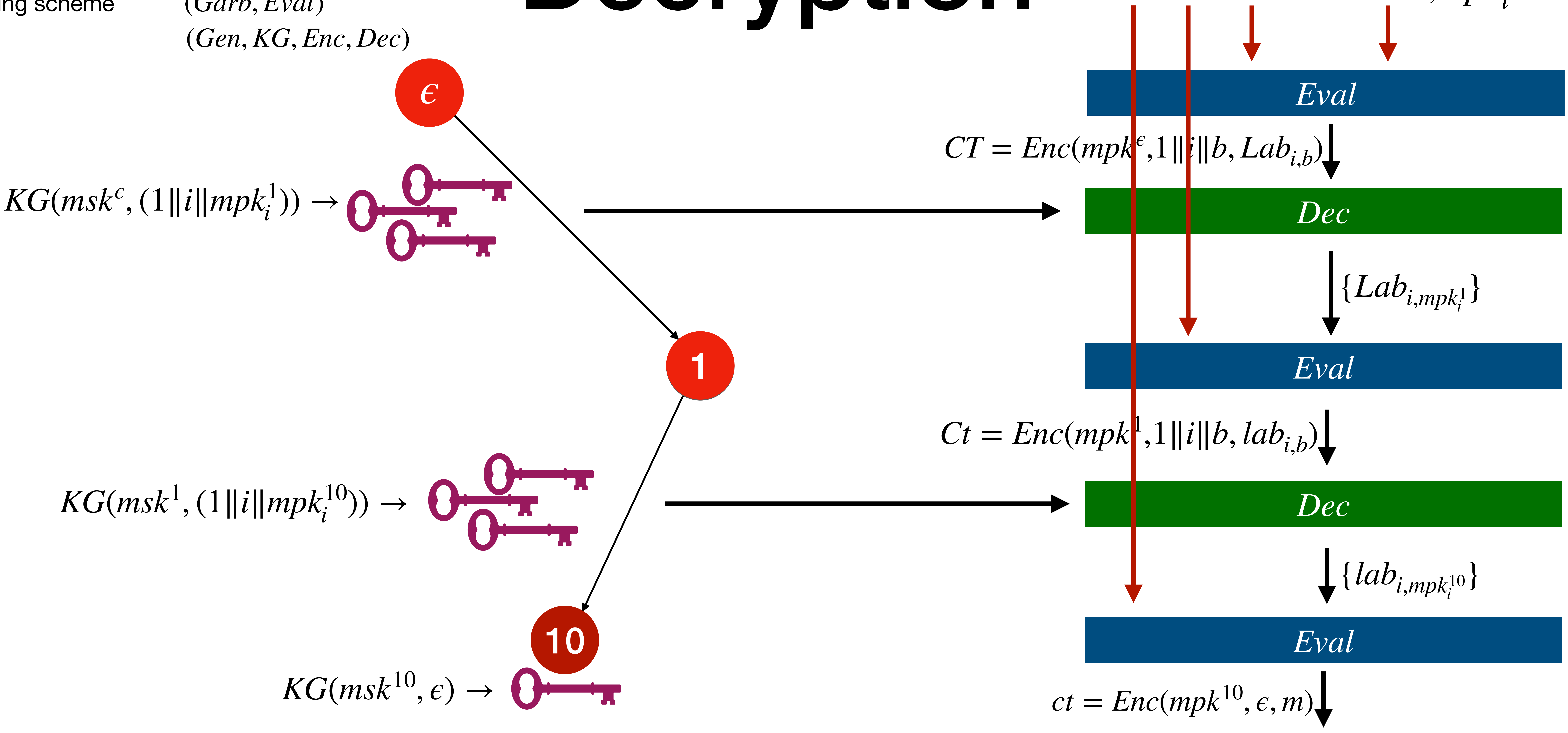


[**Döttling-Garg’17**]

Delegatable PRF	(S, E, D)
Garbling scheme	$(Garb, Eval)$
IBE	(Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$



[**Döttling-Garg'17**]

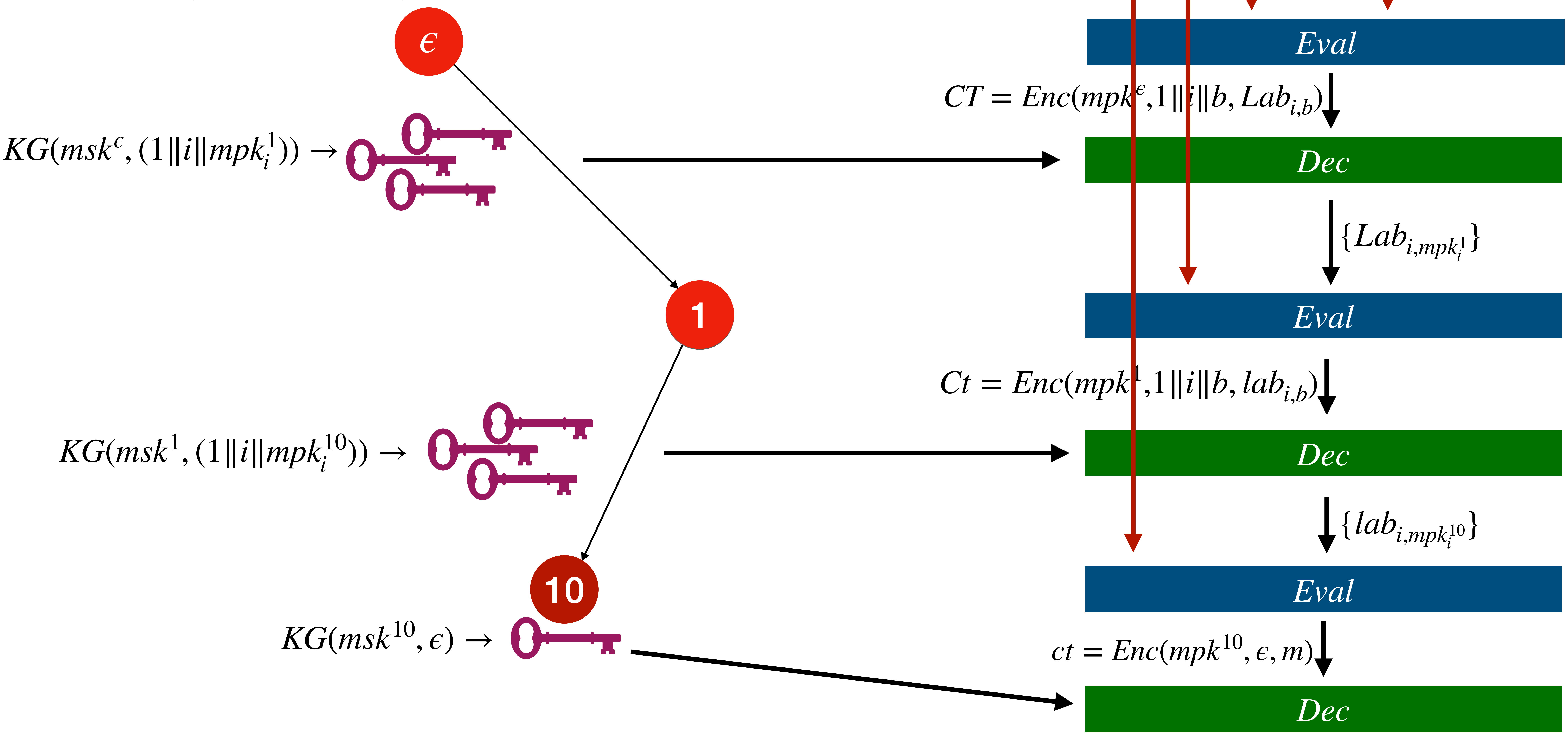
Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Decryption

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$



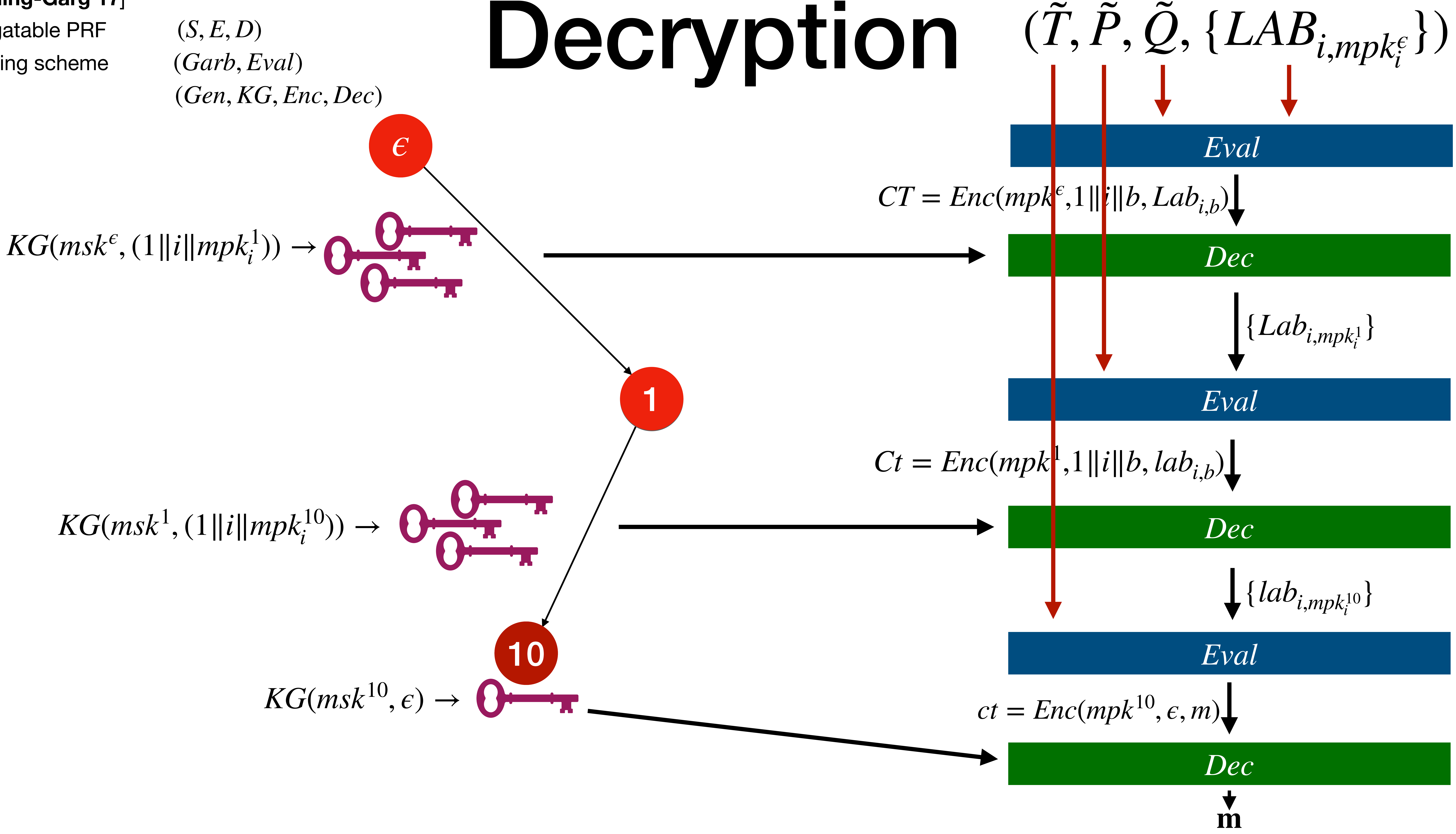
[**Döttling-Garg'17**]

Delegatable PRF (S, E, D)

Garbling scheme $(Garb, Eval)$

IBE (Gen, KG, Enc, Dec)

Decryption



Selective Security of DG17

[Döttling-Garg'17]

Selective Security of DG17

[Döttling-Garg'17]

$(\tilde{Q}, LAB_{i,b}) \leftarrow Garb(P_{'1',\{Lab_{i,b}\}})$

$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$

Selective Security of DG17

[Döttling-Garg'17]

$(\tilde{Q}, LAB_{i,b}) \leftarrow Garb(P_{'1',\{Lab_{i,b}\}})$

1. Replace PRF evaluations with truly random

$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$$

Selective Security of DG17

[Döttling-Garg'17]

$$(\tilde{Q}, LAB_{i,b}) \leftarrow Garb(P_{'1',\{Lab_{i,b}\}})$$

1. Replace PRF evaluations with truly random
    1. Use simulators of GC

$$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$$

$$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$$

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$$

Selective Security of DG17

[Döttling-Garg'17]

$(\tilde{Q}, LAB_{i,mpk_i^{\epsilon}}) \leftarrow Sim(\{CT_{i,b}\})$

1. Replace PRF evaluations with truly random
    1. Use simulators of GC

$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$$

Selective Security of DG17

[Döttling-Garg'17]

1. Replace PRF evaluations with truly random
    1. Use simulators of GC
    2. Replace IBE ciphertexts

$$(\tilde{Q}, LAB_{i,mpk_i^\epsilon}) \leftarrow Sim(\{CT_{i,b}\})$$

$$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$$

$$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$$

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$$

Selective Security of DG17

[Döttling-Garg'17]

$(\tilde{Q}, LAB_{i,mpk_i^\epsilon}) \leftarrow Sim(\{CT_{i,b}\})$

1. Replace PRF evaluations with truly random

$CT_{i,1-mpk_i^1} = Enc(mpk^\epsilon, (1\|i\|1-mpk_i^1), \mathbf{0})$

   1. Use simulators of GC
   2. Replace IBE ciphertexts

$(\tilde{P}, Lab_{i,b}) \leftarrow Garb(P_{'0',\{lab_{i,b}\}})$

$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$$

Selective Security of DG17

[Döttling-Garg'17]

1. Replace PRF evaluations with truly random
    1. Use simulators of GC
    2. Replace IBE ciphertexts

$$(\tilde{Q}, LAB_{i,mpk_i^{\epsilon}}) \leftarrow Sim(\{CT_{i,b}\})$$

$$CT_{i,1-mpk_i^1} = Enc(mpk^{\epsilon}, (1\|i\|1 - mpk_i^1), \mathbf{0})$$

$$(\tilde{P}, Lab_{i,mpk_i^1}) \leftarrow Sim(\{Ct_{i,b}\})$$

$$Ct_{i,1-mpk_i^{10}} = Enc(mpk^{\epsilon}, (1\|i\|1 - mpk_i^{10}), \mathbf{0})$$

$$\tilde{T}, \{lab_{i,b}\} \leftarrow Garb(T_m)$$

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$$

Selective Security of DG17

[Döttling-Garg'17]

1. Replace PRF evaluations with truly random
    1. Use simulators of GC
    2. Replace IBE ciphertexts

$(\tilde{Q}, LAB_{i,mpk_i^\epsilon}) \leftarrow Sim(\{CT_{i,b}\})$

$CT_{i,1-mpk_i^1} = Enc(mpk^\epsilon, (1\|i\|1 - mpk_i^1), \mathbf{0})$

$(\tilde{P}, Lab_{i,mpk_i^1}) \leftarrow Sim(\{Ct_{i,b}\})$

$Ct_{i,1-mpk_i^{10}} = Enc(mpk^\epsilon, (1\|i\|1 - mpk_i^{10}), \mathbf{0})$

$\tilde{T}, \{lab_{i,mpk_i^{10}}\} \leftarrow Sim(Enc(mpk^{10}, \epsilon, m))$

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^\epsilon}\})$$

Selective Security of DG17

[Döttling-Garg'17]

1. Replace PRF evaluations with truly random
    1. Use simulators of GC
    2. Replace IBE ciphertexts
    3. Security follows from IBE

$$(\tilde{Q}, LAB_{i,mpk_i^{\epsilon}}) \leftarrow Sim(\{CT_{i,b}\})$$

$$CT_{i,1-mpk_i^1} = Enc(mpk^{\epsilon}, (1\|i\|1 - mpk_i^1), \mathbf{0})$$

$$(\tilde{P}, Lab_{i,mpk_i^1}) \leftarrow Sim(\{Ct_{i,b}\})$$

$$Ct_{i,1-mpk_i^{10}} = Enc(mpk^{\epsilon}, (1\|i\|1 - mpk_i^{10}), \mathbf{0})$$

$$\tilde{T}, \{lab_{i,mpk_i^{10}}\} \leftarrow Sim(Enc(mpk^{10}, \epsilon, m))$$

$$(\tilde{T}, \tilde{P}, \tilde{Q}, \{LAB_{i,mpk_i^{\epsilon}}\})$$

Challenges for Adaptive Security

Challenges for Adaptive Security

- DG17 is selective because we need to *selectively puncture* id^* in GGM-style PRF tree for building delegatable PRFs.

Challenges for Adaptive Security

- DG17 is selective because we need to *selectively puncture* id^* in GGM-style PRF tree for building delegatable PRFs.
    - Hofheinz, Kastner, and Klein used *rewinding-based* proof technique to prove adaptive security of GGM PRF.

Challenges for Adaptive Security

- DG17 is selective beca… …ely puncture id^* in GGM-style PRF tree for buil…
    - Hofheinz, Kastner, a… …proof technique to prove adaptive security…

Does this/similar techniques help us in building HIBE?

Adaptive HIBE

Adaptive HIBE

We show adaptive delegatable PRFs are indeed suffices for proving DG17 is adaptively secure!!!

Adaptive HIBE

We show adaptive delegatable PRFs are indeed suffices for proving DG17 is adaptively secure!!!

Surprisingly, the original construction needs no changes!!!

Adaptive HIBE

We show adaptive delegatable PRFs are indeed suffices for proving DG17 is adaptively secure!!!

Surprisingly, the original construction needs no changes!!!

This may explain why there were no improvements to DG17-HIBE in the last decade.

Adaptive HIBE

We show adaptive delegatable PRFs are indeed suffices for proving DG17 is adaptively secure!!!

Surprisingly,
the original
construction ne
change

This may
explain why
were no
ents to DG17-
he last decade.

The proof is
NOT the same as
DG17!!!

Why DG17 Proof Still Fails?

Why DG17 Proof Still Fails?

- DG17 proof hardwires garbled circuits based on challenge identity id^*.

Why DG17 Proof Still Fails?

- DG17 proof hardwires garbled circuits based on challenge identity id^*.
- In adaptive security, id^* is not known during pre-challenge phase.

Why DG17 Proof Still Fails?

- DG17 proof hardwires garbled circuits based on challenge identity id^*.
- In adaptive security, id^* is not known during pre-challenge phase.
- Cannot simulate garbling step-by-step as before — the simulation would fail because all $id^*[1:i]$ must be **correctly** guessed in pre-challenge phase.

Why DG17 Proof Still Fails?

- DG17 proof hardwires garbled circuits based on challenge identity id^*.
- In adaptive security, id^* is not known during pre-challen
- Cannot simulate garbling step-by-step as before – d
  fail because all $id^*[1 : i]$ must be **correctly** guesse
  phase.

Adversary
may obtain
keys related to
$mpk^{id^*[1:i]}$ and $msk^{id^*[1:i]}$

Why DG17 Proof Still Fails?

- DG17 proof hardwires garbled circuits based on challenge identity id^*.
- In adaptive security, id^* is not known during pre-challen
- Cannot simulate garbling step-by-step as before – d fail because all $id^*[1:i]$ must be **correctly** guesse phase.
- Need a **new proof approach** to support adaptive adversaries!!!

Adversary may obtain keys related to $mpk^{id^*[1:i]}$ and $msk^{id^*[1:i]}$

Why DG17 Proof Still Fails?

- DG17 proof hardwires garbled circuits based on challenge identity id^*.
- In adaptive security, id^* is not known during pre-challen
- Cannot simulate garbling step-by-step as before – d fail because all $id^*[1:i]$ must be **correctly** guesse phase.
- Need a **new proof approach** to support adaptive adversaries!!!

Adversary may obtain keys related to $mpk^{id^*[1:i]}$ and $msk^{id^*[1:i]}$

Nested Hybrids & Pebbling-style strategy

New Proof Strategy – High Level Sketch

New Proof Strategy – High Level Sketch

- Game 0 : Regular HIBE adaptive security game

New Proof Strategy – High Level Sketch

- Game 0 : Regular HIBE adaptive security game
- Game 1 : All HIBE secret keys are generated using msk.

New Proof Strategy – High Level Sketch

$KeyGen(msk, id_1 \| id_2) = Delegate(sk_{id_1}, id_2)$

- Game 0 : Regular HIBE adaptive security game
- Game 1 : All HIBE secret keys are generated using msk.

New Proof Strategy – High Level Sketch

$KeyGen(msk, id_1 \| id_2) = Delegate(sk_{id_1}, id_2)$

- Game 0 : Regular HIBE adaptive security game
- Game 1 : All HIBE secret keys are generated using msk.
- Game i : Simulate the first i garbled circuits.

New Proof Strategy – High Level Sketch

$KeyGen(msk, id_1 \| id_2) = Delegate(sk_{id_1}, id_2)$

- Game 0 : Regular HIBE adaptive security game
- Game 1 : All HIBE secret keys are generated using msk.
- Game i : Simulate the first i garbled circuits.
- Last Game : Simulate the final garbled circuit that encrypts m.

New Proof Strategy – High Level Sketch

- Game 0 : Regular HIBE adaptive security game

$KeyGen(msk, id_1 \| id_2) = Delegate(sk_{id_1}, id_2)$

- Game 1 : All HIBE secret keys are generated using msk.
- Game i : Simulate the first i garbled circuits.
- Last Game : Simulate the final garbled circuit that encrypts m.

Finally, security follows from IBE security.

New Proof Strategy – High Level Sketch

$KeyGen(msk, id_1 \| id_2) = Delegate(sk_{id_1}, id_2)$

- Game 0 : Regular HIBE adaptive security game
- Game 1 : All HIBE secret keys are generated using msk.
- $\boxed{\text{Game } i \text{ : Simulate the first } i \text{ garbled circuits.}}$
- Last Game : Simulate the final garbled circuit that encrypts m.

Finally, security follows from IBE security.

High Level Sketch - Game i

High Level Sketch - Game i

- Step 1 : Simulate the i^{th} garbled circuit.

High Level Sketch - Game i

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.

High Level Sketch - Game i

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1 : i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.

High Level Sketch - Game i

How to handle pre-challenge queries???

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.

High Level Sketch - Game i

How to handle pre-challenge queries???

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.
- Step 4 : Use PRF evaluation at node $id^*[1:i]$

High Level Sketch - Game i

- Step 1 : Simulate the i^{th} garbled circuit.

How to handle pre-challenge queries???

- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.
- Step 4 : Use PRF evaluation at node $id^*[1:i]$

Only need to sample a single pair of IBE keys randomly!!!

High Level Sketch - Game i

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.
- Step 4 : Use PRF evaluation at node $id^*[1:i]$

How to handle pre-challenge queries???

Pebbling strategy
[Hemenway-Jafargholi-Ostrovsky-Scafuro-Wichs'16]
[Jafargholi-Wichs'16]

Only need to sample a single pair of IBE keys randomly!!!

High Level Sketch - Game i

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.
- Step 4 : Use PRF evaluation at node $id^*[1:i]$

High Level Sketch - Game i

How to guess $id^*[1:i]$???

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.
- Step 4 : Use PRF evaluation at node $id^*[1:i]$

High Level Sketch - Game i

How
to guess
$id^*[1:i]$???

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.
- Step 4 : Use PRF evaluation at node $id^*[1:i]$

Nested Hybrid

[Hohenberger-Waters’10,Fuchsbauer-Konstantinov-Pietrzak-Rao’14,Fuchsbauer-Jafargholi-Pietrzak’15]

High Level Sketch - Game i

How to guess $id^*[1:i]$???

- Step 1 : Simulate the i^{th} garbled circuit.
- Step 2 : Replace PRF evaluation at node $id^*[1:i]$ with truly random.
- Step 3 : Use IBE security to switch half of the ciphertexts.
- Step 4 : Use PRF evaluation at node $id^*[1:i]$

Nested Hybrid → Depends on q - first key query that overlaps with $id^*[1:i]$

[Hohenberger-Waters'10,Fuchsbauer-Konstantinov-Pietrzak-Rao'14,Fuchsbauer-Jafargholi-Pietrzak'15]

Final Remarks

Final Remarks

- In this work, we showed adaptive HIBE is achievable from selective IBE.

Final Remarks

- In this work, we showed adaptive HIBE is achievable from selective IBE.
  - Tighter security loss?

Final Remarks

- In this work, we showed adaptive HIBE is achievable from selective IBE.
  - Tighter security loss? Quadratic

Final Remarks

- In this work, we showed adaptive HIBE is achievable from selective IBE.
  - Tighter security loss? Quadratic
  - Practically efficient constructions?

Final Remarks

- In this work, we showed adaptive HIBE is achievable from selective IBE.
  - Tighter security loss? Quadratic
  - Practically efficient constructions? Avoid garbling schemes

Final Remarks

- In this work, we showed adaptive HIBE is achievable from selective IBE.
    - Tighter security loss? Quadratic
    - Practically efficient constructions? Avoid garbling schemes
    - Can we use these techniques in other settings such as Delegatable ABE?

Final Remarks

- In this work, we showed adaptive HIBE is achievable from selective IBE.
  - Tighter security loss? Quadratic
  - Practically efficient constructions? Avoid garbling schemes
  - Can we use these techniques in other settings such as Delegatable ABE? Current work by [Goyal-Yadugiri'25]

Thank You

https://mahe94.github.io

Funded by the European Union (ERC, LACONIC, 101041207). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.