Adaptive Security of Constrained PRFs

Georg Fuchsbauer (IST Austria)
Momchil Konstantinov (LSGNT,UK)
Krzysztof Pietrzak (IST Austria)
Vanishree Rao (UCLA)

ASIACRYPT 2014

Overview

We consider adaptive/full security of recent primitive:

Constrained Pseudorandom Functions

[+] Introduce new proof technique and improve reduction of GGM from exponential loss to quasipolynomial loss

[−] Show that exponential loss for Boneh-Waters CPRF is inherent

Crypto 101

PRG

$\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is a (length-doubling) (ϵ, s)-secure pseudorandom generator (PRG) if for all $\mathcal{A}$, $|\mathcal{A}| \leq s$

$$|\Pr_{x \leftarrow U_\lambda}[\mathcal{A}(\mathsf{G}(x)) \to 1] - \Pr_{y \leftarrow U_{2\lambda}}[\mathcal{A}(y) \to 1]| \leq \epsilon$$

PRG

$\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is a (length-doubling) (ϵ, s)-secure pseudorandom generator (PRG) if for all $\mathcal{A}$, $|\mathcal{A}| \leq s$

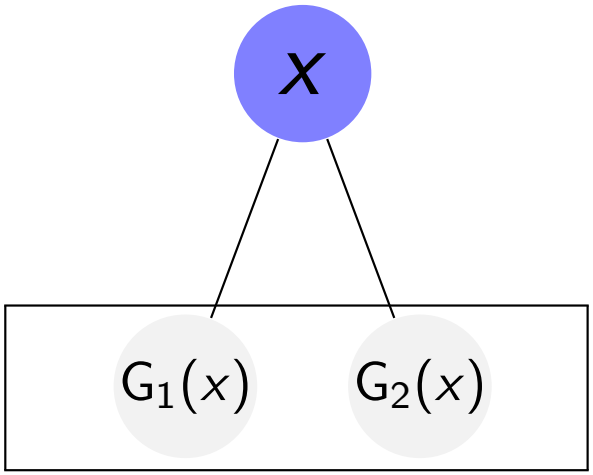
$$|\Pr_{x \leftarrow U_\lambda}[\mathcal{A}(\mathsf{G}(x)) \to 1] - \Pr_{y \leftarrow U_{2\lambda}}[\mathcal{A}(y) \to 1]| \leq \epsilon$$

PRF

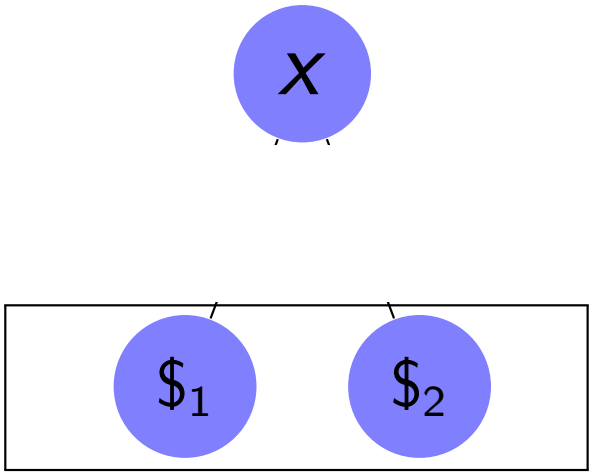
$\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a (ϵ, s, q)-secure pseudorandom function (PRF) if for all $\mathcal{A}$, $|\mathcal{A}| \leq s$, making $\leq q$ queries:

$$\left| \Pr_{K \leftarrow \mathcal{K}}[\mathcal{A}^{\mathsf{F}(K,.)} \to 1] - \Pr_{f \leftarrow \mathcal{F}[\mathcal{X},\mathcal{Y}]}[\mathcal{A}^{f(.)} \to 1] \right| \leq \epsilon$$

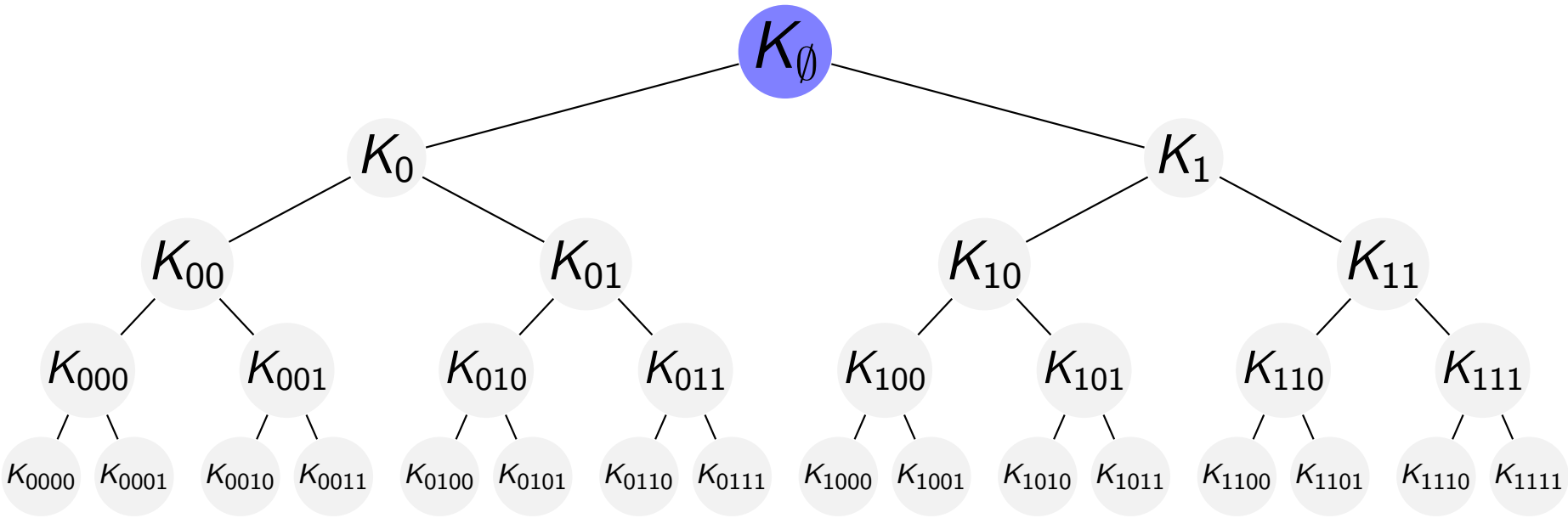
Goldreich-Goldwasser-Micali 84



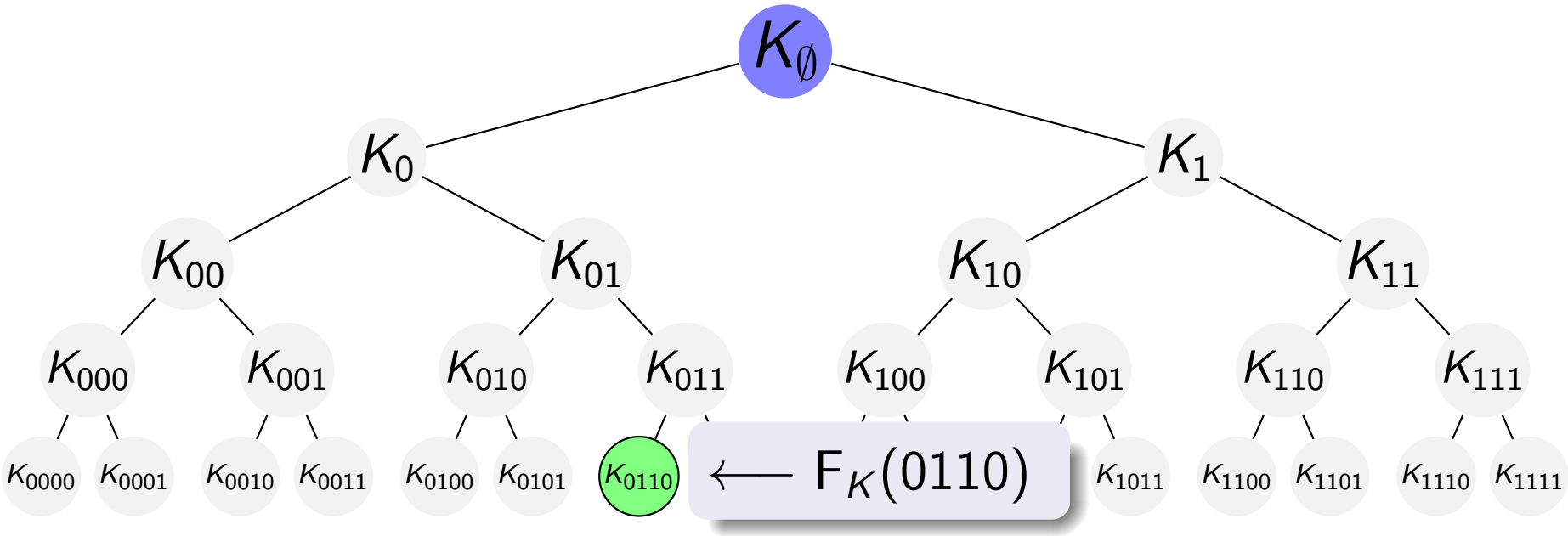
Goldreich-Goldwasser-Micali 84



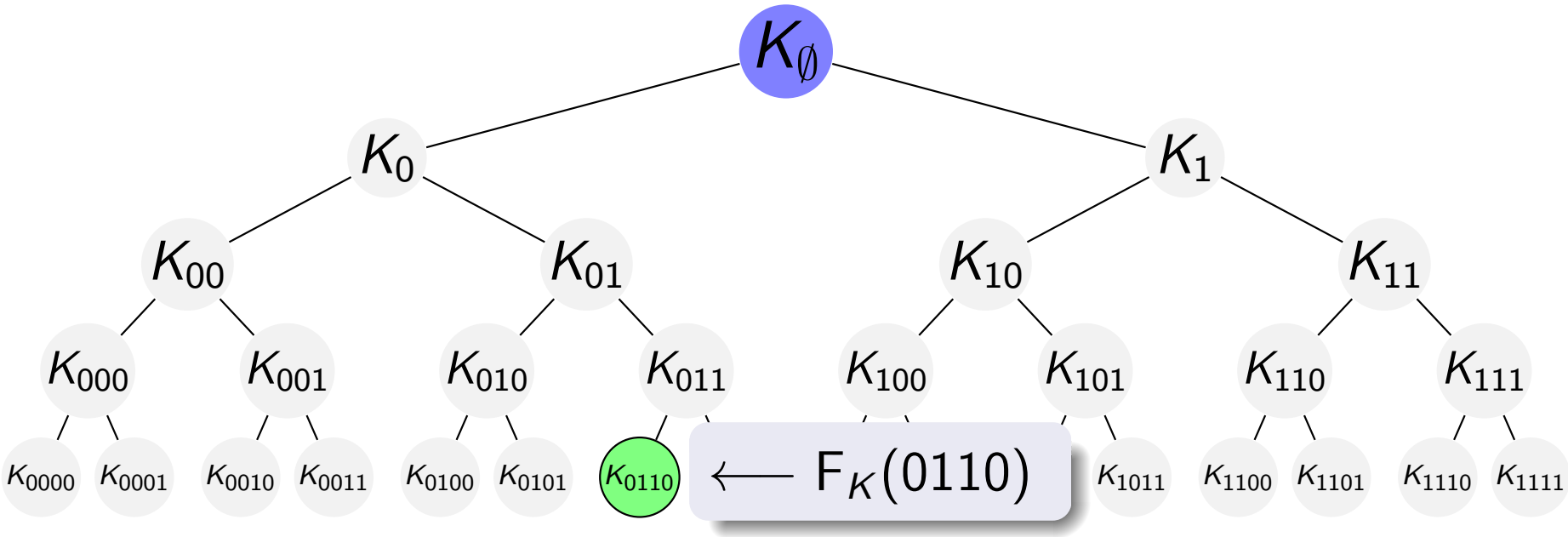
Goldreich-Goldwasser-Micali 84



Goldreich-Goldwasser-Micali 84



Goldreich-Goldwasser-Micali 84

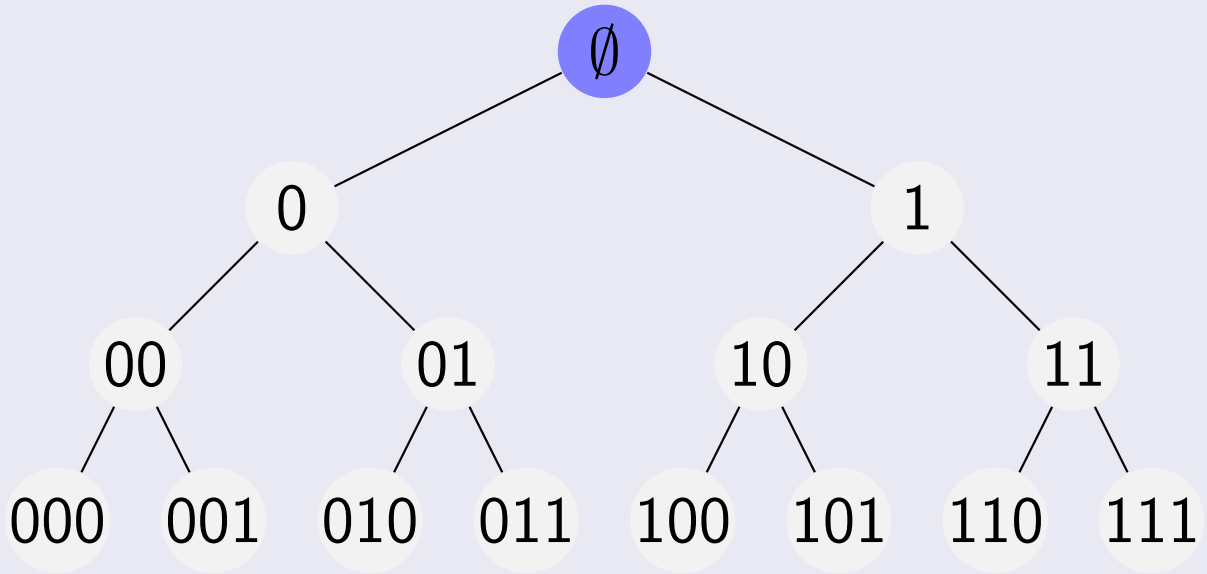


GGM PRF

- PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$
- $K = K_\emptyset \leftarrow \{0,1\}^n$
- $K_{x\|0} \| K_{x\|1} = G(K_x)$
- $F_K(x) = K_x$

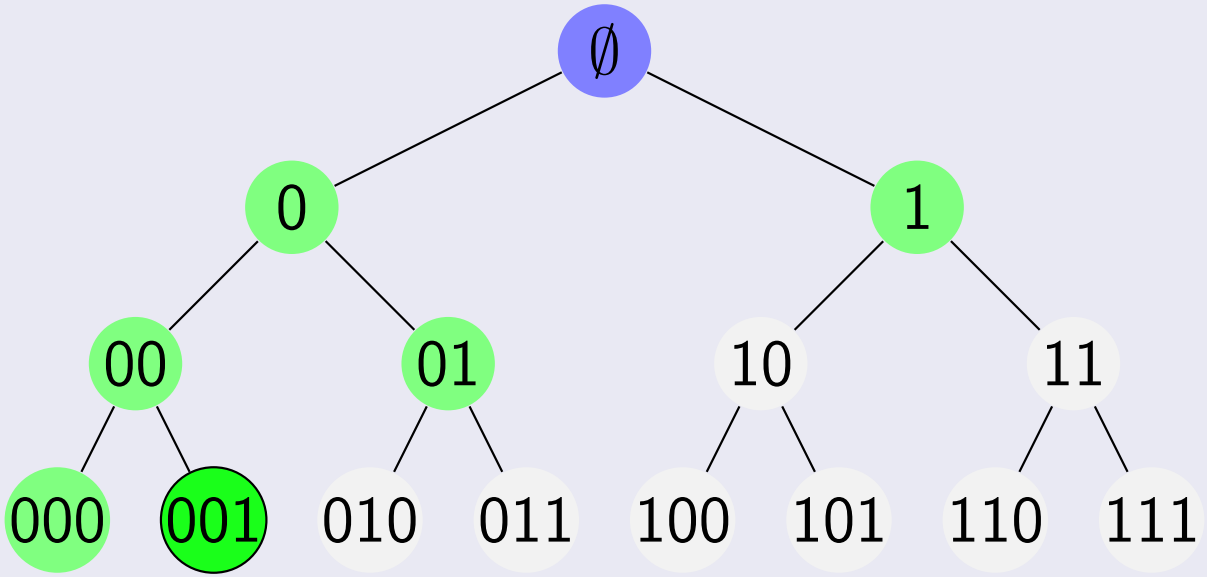
GGM hybrid argument

GGM



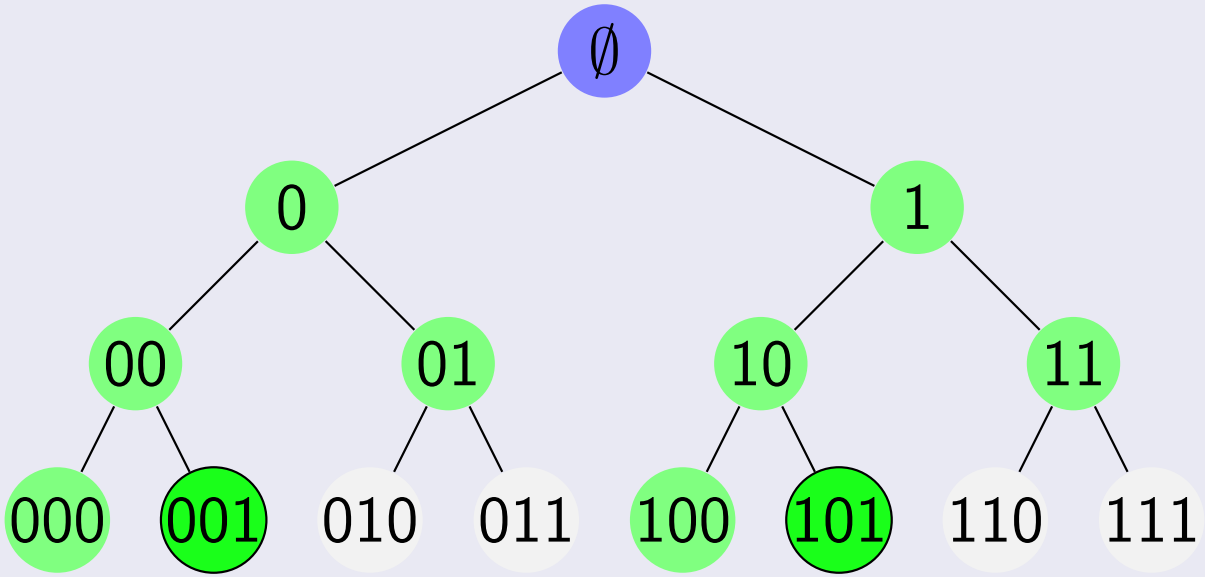
GGM hybrid argument

GGM: evaluation at 001



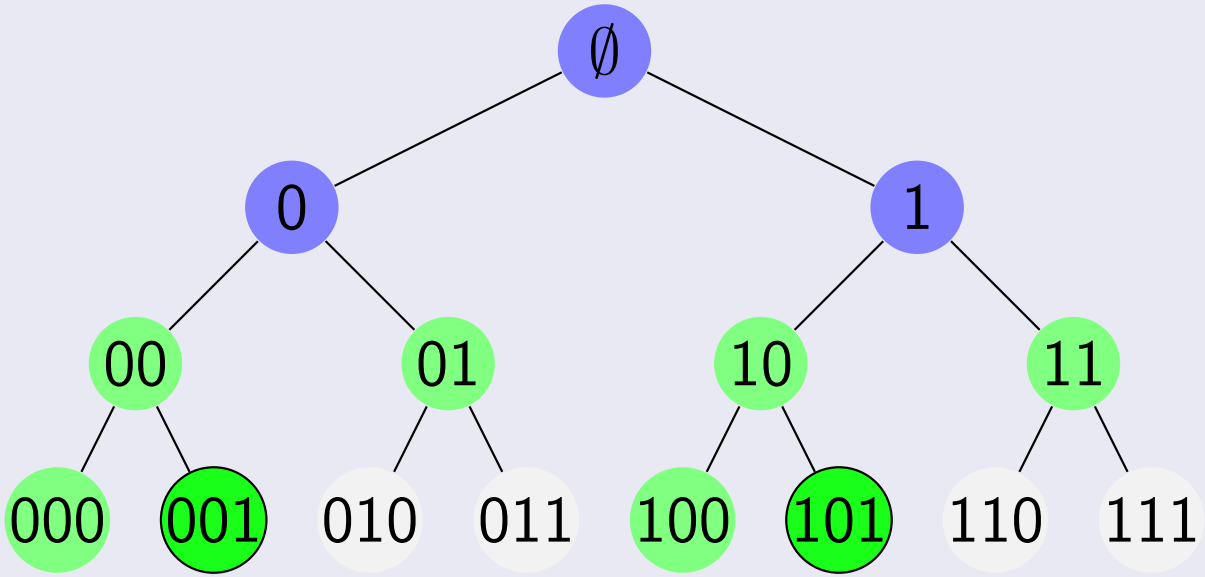
GGM hybrid argument

Hybrid H_0 (the real game)



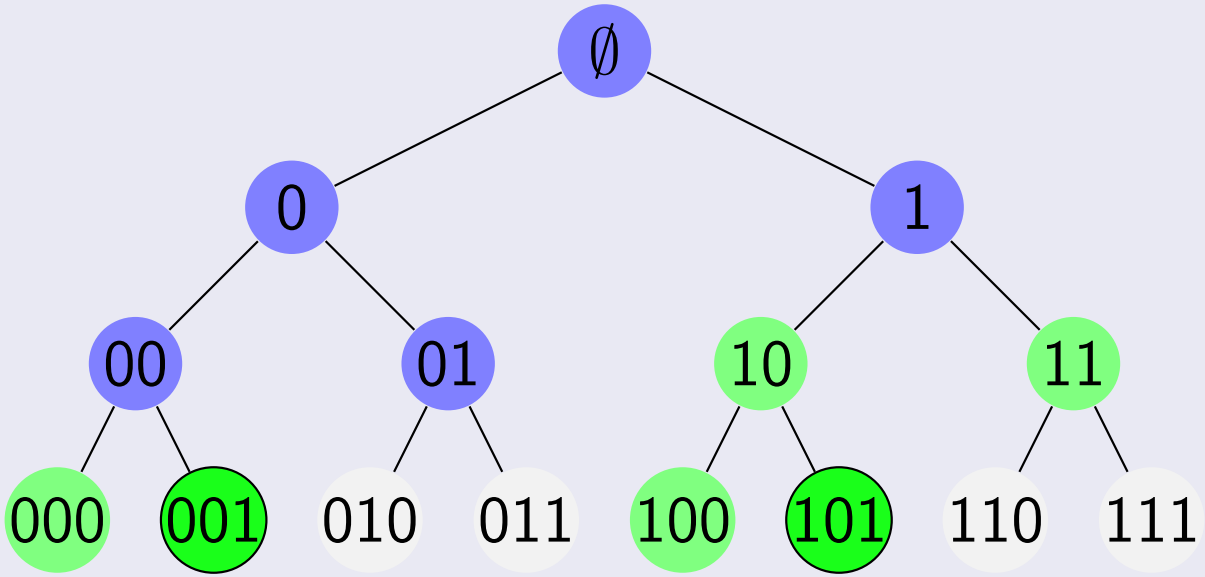
GGM hybrid argument

Hybrid H_1



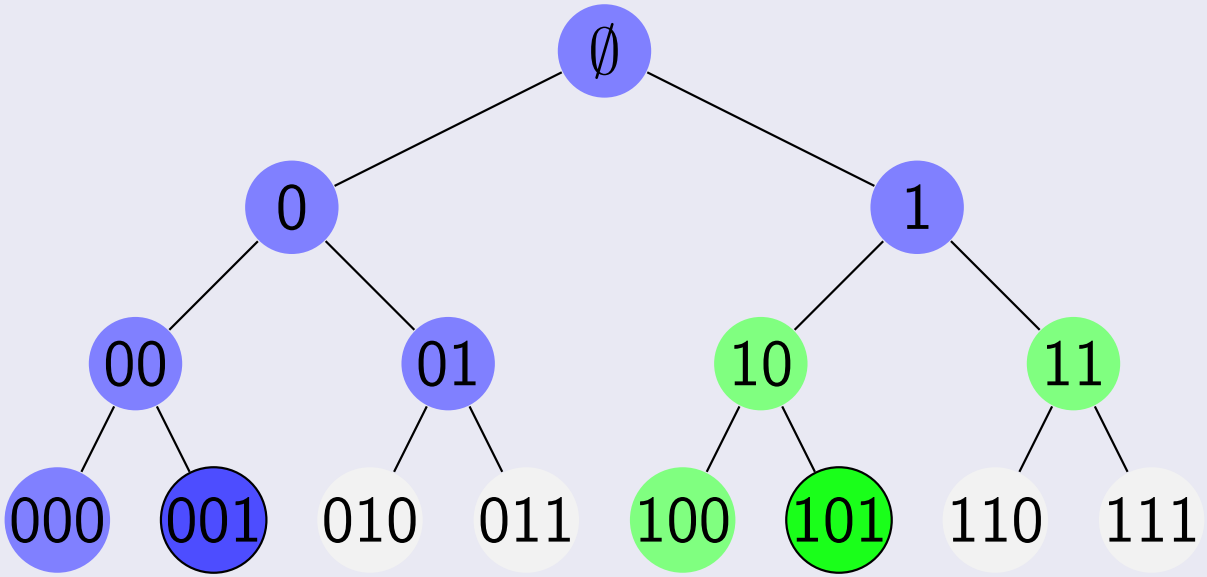
GGM hybrid argument

Hybrid H_2



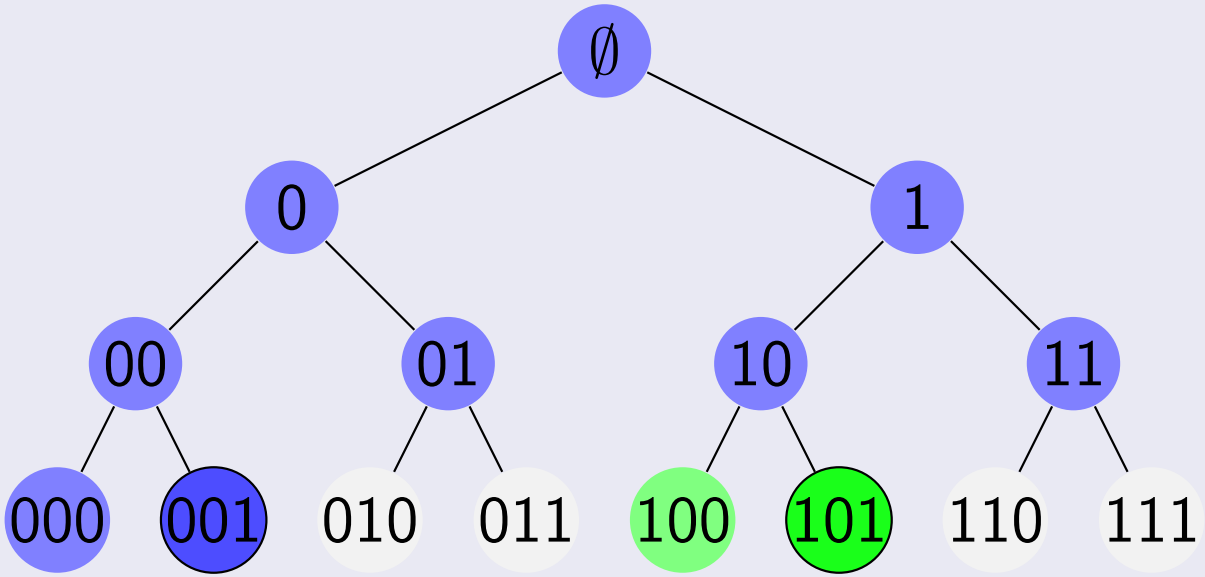
GGM hybrid argument

Hybrid H_3



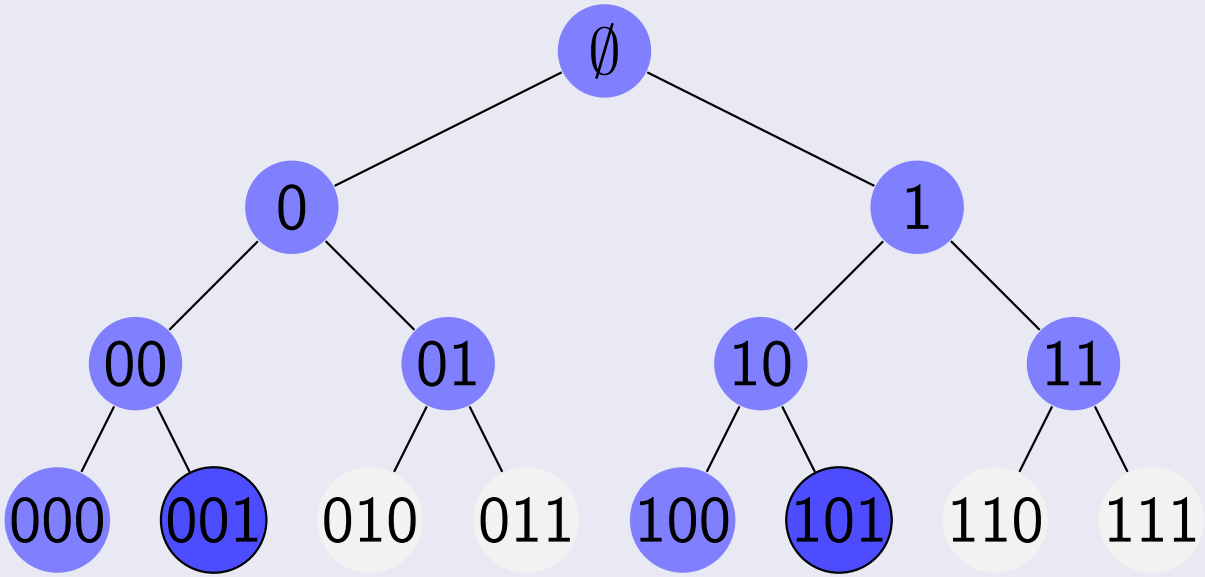
GGM hybrid argument

Hybrid H_4



GGM hybrid argument

Hybrid H_5 (the random game)



- $\mathrm{Adv}(H_0, H_{qn}) = \epsilon$ $\quad$ $q = \#\text{queries}$, n=input length.

$\Rightarrow$ $\mathrm{Adv}(H_i, H_{i+1}) \geq \epsilon/qn$ for some i

$\Rightarrow$ break G with adv. $\geq \epsilon/qn$

Constrained PRFs [BW13]

constrained PRF

$\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a constrained PRF for set system $\mathcal{S} \subseteq 2^{\mathcal{X}}$ if there is

- constrained key space $\mathcal{K}_c$
- $\mathsf{F.cstr}\colon \mathcal{K} \times \mathcal{S} \to \mathcal{K}_c$
- $\mathsf{F.eval}\colon \mathcal{K}_c \times \mathcal{X} \to \mathcal{Y}$, s.t.

Constrained PRFs [BW13]

constrained PRF

$\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a constrained PRF for set system $\mathcal{S} \subseteq 2^{\mathcal{X}}$ if there is

- constrained key space $\mathcal{K}_c$
- $\mathsf{F.cstr}\colon \mathcal{K} \times \mathcal{S} \to \mathcal{K}_c$
- $\mathsf{F.eval}\colon \mathcal{K}_c \times \mathcal{X} \to \mathcal{Y}$, s.t.

$$k_S \leftarrow \mathsf{F.cstr}(k, S)$$

$$\Rightarrow \quad \mathsf{F.eval}(k_S, x) = \begin{cases} \mathsf{F}(k, x) & \text{if } x \in S \\ \perp & \text{otherwise} \end{cases}$$

Constrained PRFs [BW13]

constrained PRF

$\mathsf{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a constrained PRF for set system $\mathcal{S} \subseteq 2^{\mathcal{X}}$ if there is

- constrained key space $\mathcal{K}_c$
- $\mathsf{F.cstr}\colon \mathcal{K} \times \mathcal{S} \to \mathcal{K}_c$
- $\mathsf{F.eval}\colon \mathcal{K}_c \times \mathcal{X} \to \mathcal{Y}$, s.t.

$$k_S \leftarrow \mathsf{F.cstr}(k, S)$$

$$\Rightarrow \quad \mathsf{F.eval}(k_S, x) = \begin{cases} \mathsf{F}(k, x) & \text{if } x \in S \\ \perp & \text{otherwise} \end{cases}$$

$\Rightarrow$ F should look random where one cannot evaluate

Adaptive vs. selective security

Adaptive security experiment

- $k \leftarrow \mathcal{K}$, $b \leftarrow \{0,1\}$
- $\mathcal{A}$ can query $\mathsf{F.cstr}(k, \cdot)$
- $\mathcal{A}$ submits $x^* \in \{0,1\}^n$
- $\mathcal{A}$ gets $\begin{cases} \mathsf{F}(k, x^*) & \text{if } b = 1 \\ y \leftarrow \mathcal{Y} & \text{if } b = 0 \end{cases}$
- $\mathcal{A}$ outputs b'

Adaptive vs. selective security

Adaptive security experiment

- $k \leftarrow \mathcal{K}$, $b \leftarrow \{0,1\}$
- $\mathcal{A}$ can query $\mathsf{F.cstr}(k, \cdot)$
- $\mathcal{A}$ submits $x^* \in \{0,1\}^n$
- $\mathcal{A}$ gets $\begin{cases} \mathsf{F}(k, x^*) & \text{if } b = 1 \\ y \leftarrow \mathcal{Y} & \text{if } b = 0 \end{cases}$
- $\mathcal{A}$ outputs b'

$\mathcal{A}$ must choose $x^* \notin \bigcup S_i$, $\{S_i\}$ queried to F.cstr

Adaptive vs. selective security

Adaptive security experiment

- $k \leftarrow \mathcal{K}$, $b \leftarrow \{0,1\}$
- $\mathcal{A}$ can query $\mathsf{F.cstr}(k, \cdot)$
- $\mathcal{A}$ submits $x^* \in \{0,1\}^n$
- $\mathcal{A}$ gets $\begin{cases} \mathsf{F}(k, x^*) & \text{if } b = 1 \\ y \leftarrow \mathcal{Y} & \text{if } b = 0 \end{cases}$
- $\mathcal{A}$ outputs b'

F is (ϵ, s, q)-secure if

for all q-query $\mathcal{A}$, $|\mathcal{A}| \leq s$: $\left| \Pr[b' = b] - \frac{1}{2} \right| \leq \epsilon$

Adaptive vs. selective security

Selective security experiment

- $k \leftarrow \mathcal{K}$, $b \leftarrow \{0,1\}$
- $\mathcal{A}$ submits $x^* \in \{0,1\}^n$
- $\mathcal{A}$ can query $\mathsf{F.cstr}(k, \cdot)$
- $\mathcal{A}$ gets $\begin{cases} \mathsf{F}(k, x^*) & \text{if } b = 1 \\ y \leftarrow \mathcal{Y} & \text{if } b = 0 \end{cases}$
- $\mathcal{A}$ outputs b'

F is (ϵ, s, q)-selectively secure if

for all q-query $\mathcal{A}$, $|\mathcal{A}| \leq s$: $\left| \Pr[b' = b] - \frac{1}{2} \right| \leq \epsilon$

Adaptive vs. selective security

Selective security experiment

- $k \leftarrow \mathcal{K}$, $b \leftarrow \{0,1\}$
- $\mathcal{A}$ submits $x^* \in \{0,1\}^n$
- $\mathcal{A}$ can query $\mathsf{F.cstr}(k, \cdot)$
- $\mathcal{A}$ gets $\begin{cases} \mathsf{F}(k, x^*) & \text{if } b = 1 \\ y \leftarrow \mathcal{Y} & \text{if } b = 0 \end{cases}$
- $\mathcal{A}$ outputs b'

Complexity leveraging

$\mathcal{A}$ breaks adaptive security with adv. ϵ

$\Rightarrow$

can break selective security with adv. $\epsilon/2^n$.

F is (ϵ, s, q)-selectively secure if

for all q-query $\mathcal{A}$, $|\mathcal{A}| \leq s$: $\left|\Pr[b' = b] - \frac{1}{2}\right| \leq \epsilon$

GGM as constrained PRF

[BW13] Boneh, Waters: *Constrained Pseudorandom Functions and Their Applications.* Asiacrypt 2013

[KPTZ13] Kiayias, Papadopoulos, Triandopoulos, Zacharias: *Delegatable Pseudorandom Functions and Applications.* CCS 2013

[BGI14] Boyle, Goldwasser, Ivan: *Functional Signatures and Pseudorandom Functions.* PKC'14

- GGM is prefix-constrained PRF
- ie constrained PRF for set system

$$\mathcal{S} = \left\{ S_p \mid p \in \{0,1\}^{\leq n} \right\}$$

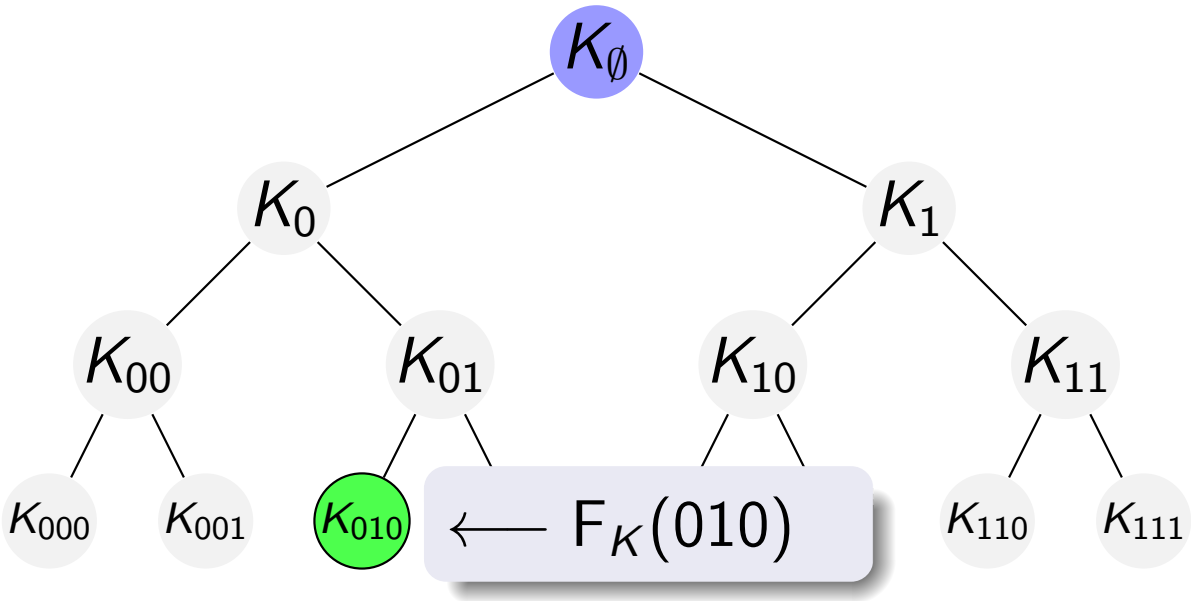
- with $S_p = \left\{ p \| z \;\middle|\; z \in \{0,1\}^{n-|p|} \right\}$

GGM as constrained PRF

[BW13] Boneh, Waters: *Constrained Pseudorandom Functions and Their Applications.* Asiacrypt 2013

[KPTZ13] Kiayias, Papadopoulos, Triandopoulos, Zacharias: *Delegatable Pseudorandom Functions and Applications.* CCS 2013

[BGI14] Boyle, Goldwasser, Ivan: *Functional Signatures and Pseudorandom Functions.* PKC'14

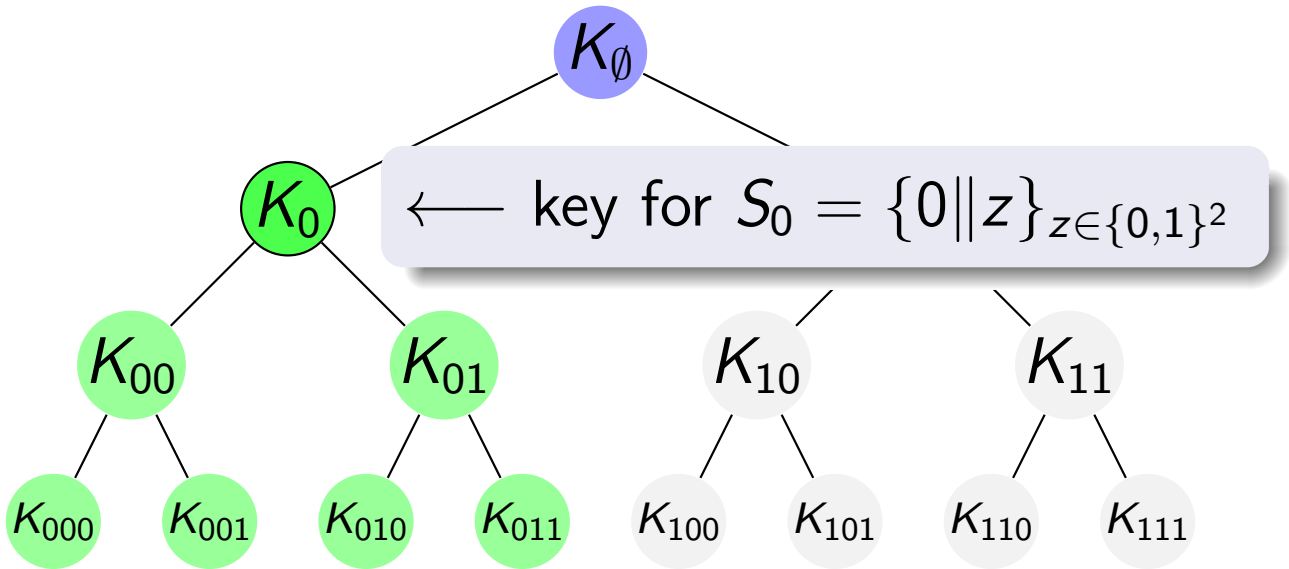


GGM as constrained PRF

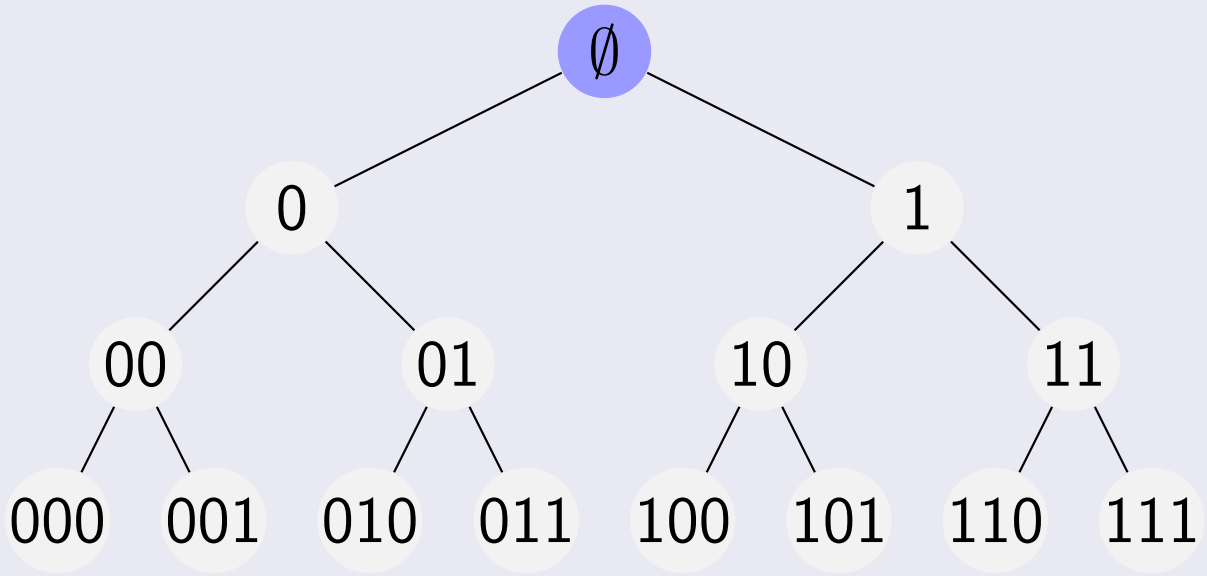
[BW13] Boneh, Waters: *Constrained Pseudorandom Functions and Their Applications.* Asiacrypt 2013

[KPTZ13] Kiayias, Papadopoulos, Triandopoulos, Zacharias: *Delegatable Pseudorandom Functions and Applications.* CCS 2013

[BGI14] Boyle, Goldwasser, Ivan: *Functional Signatures and Pseudorandom Functions.* PKC'14

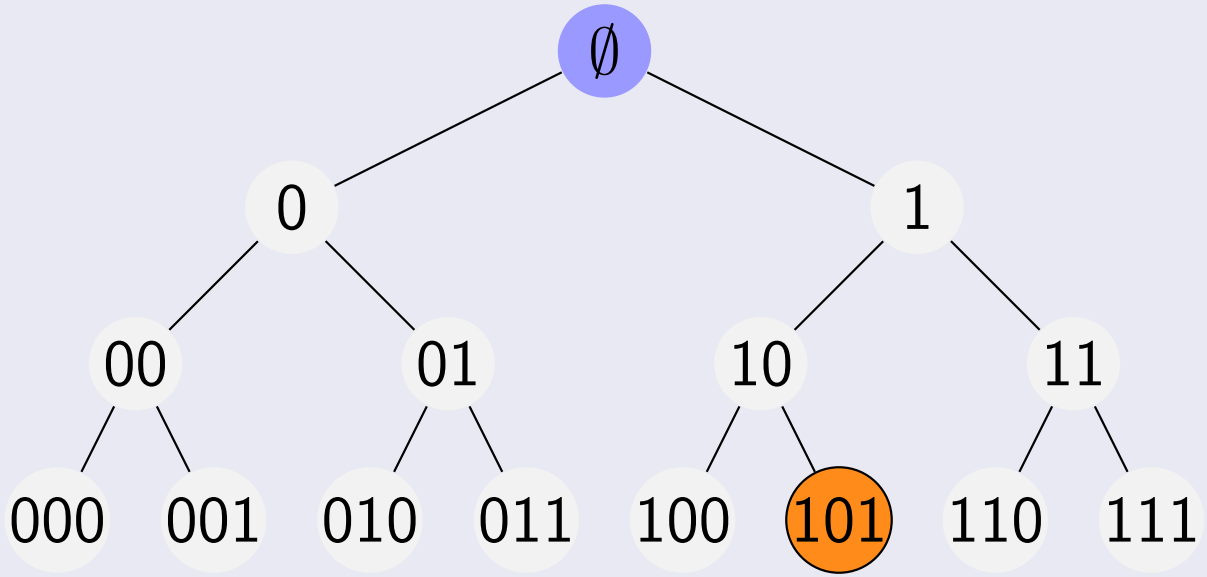


Proving selective security



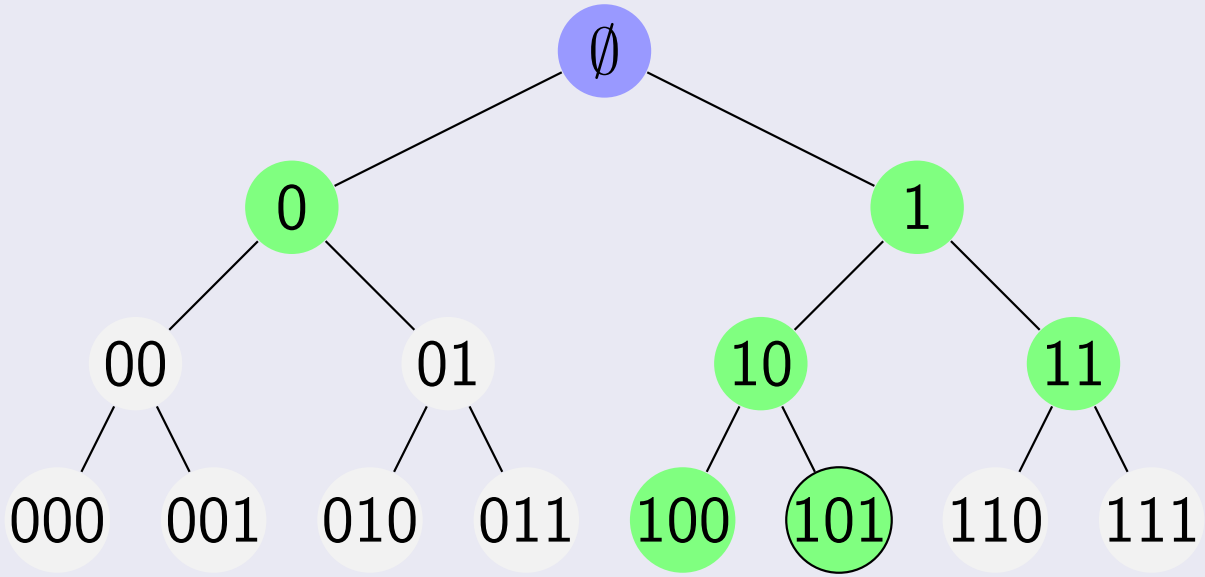
Proving selective security

Submitted challenge



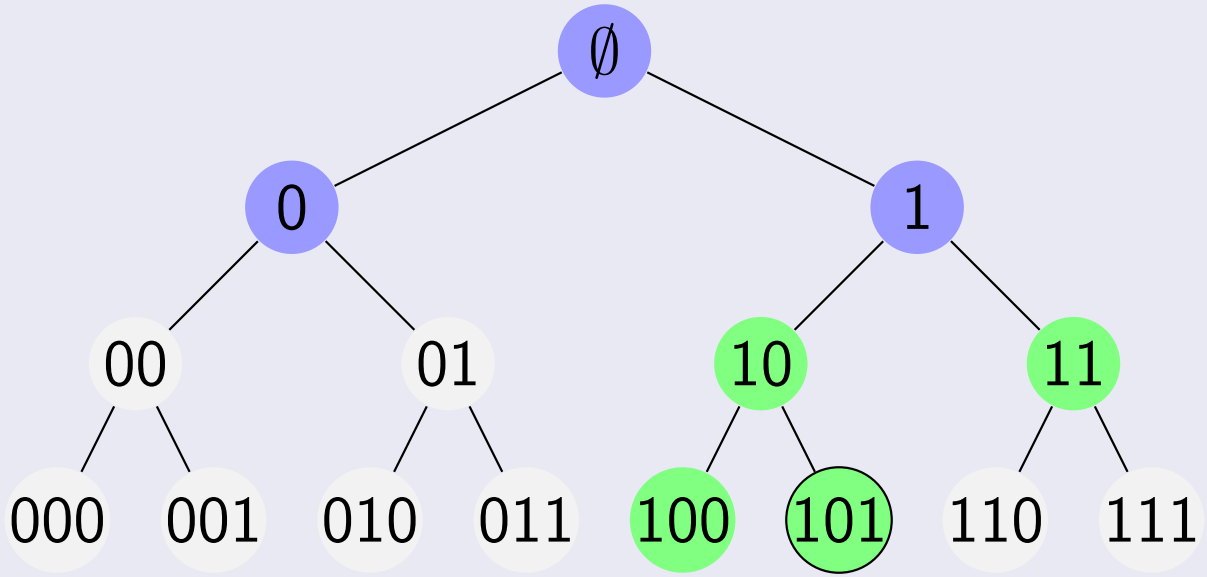
Proving selective security

Hybrid H_0 (real game)



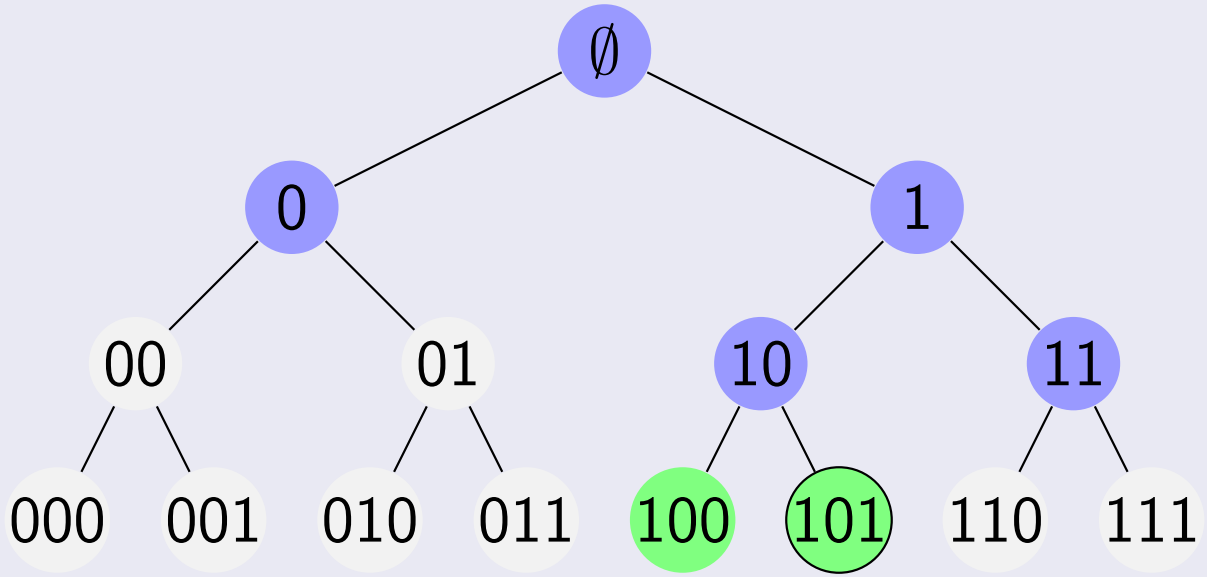
Proving selective security

Hybrid H_1



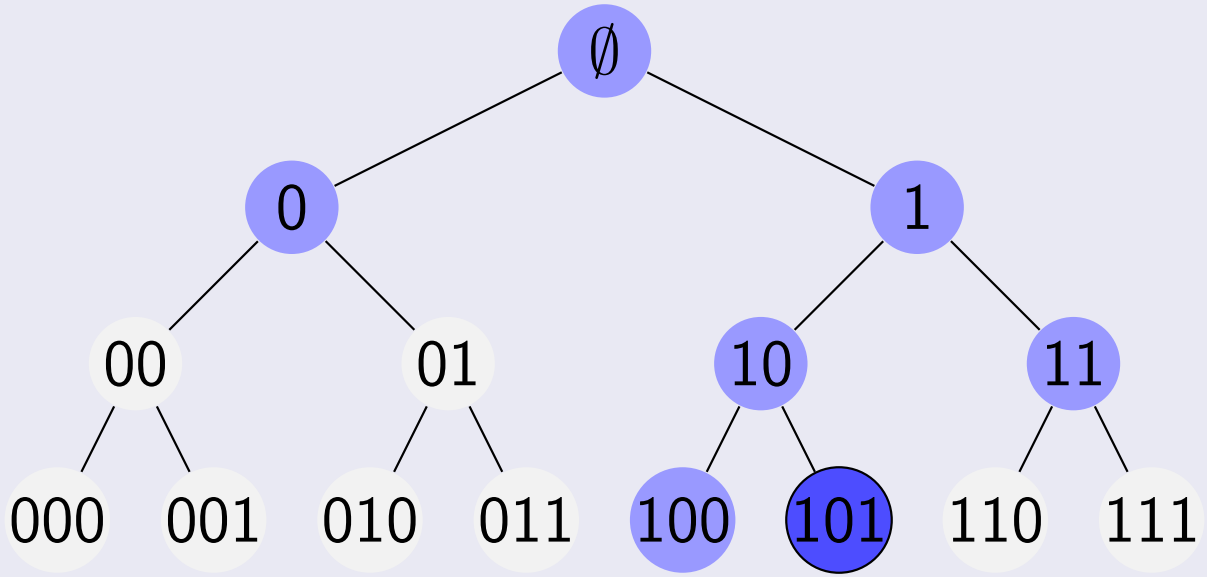
Proving selective security

Hybrid H_2



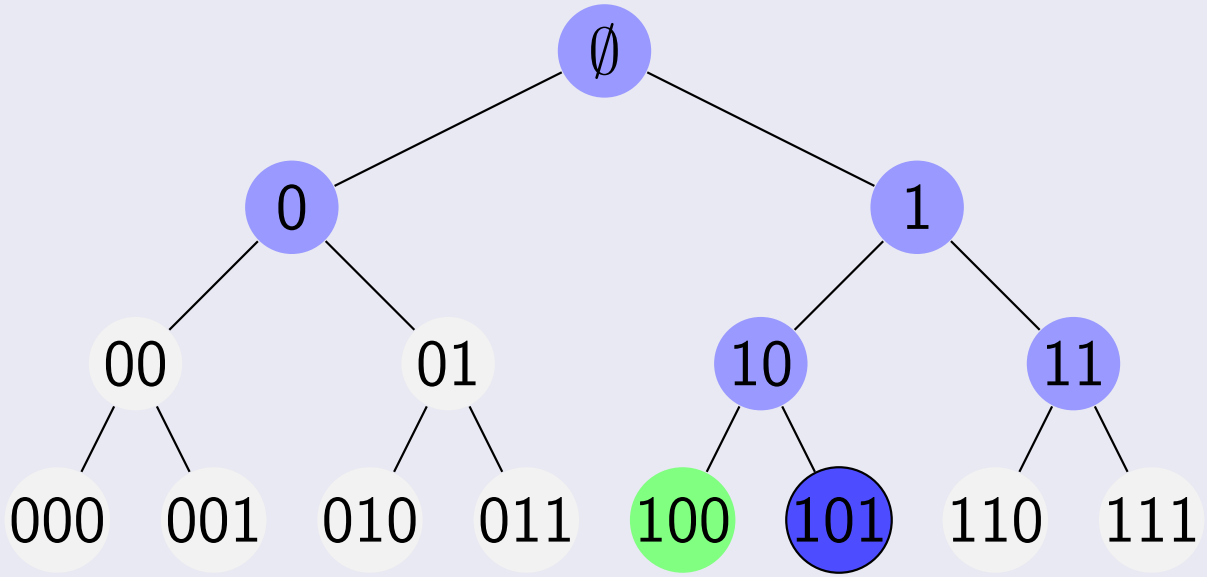
Proving selective security

Hybrid H_3



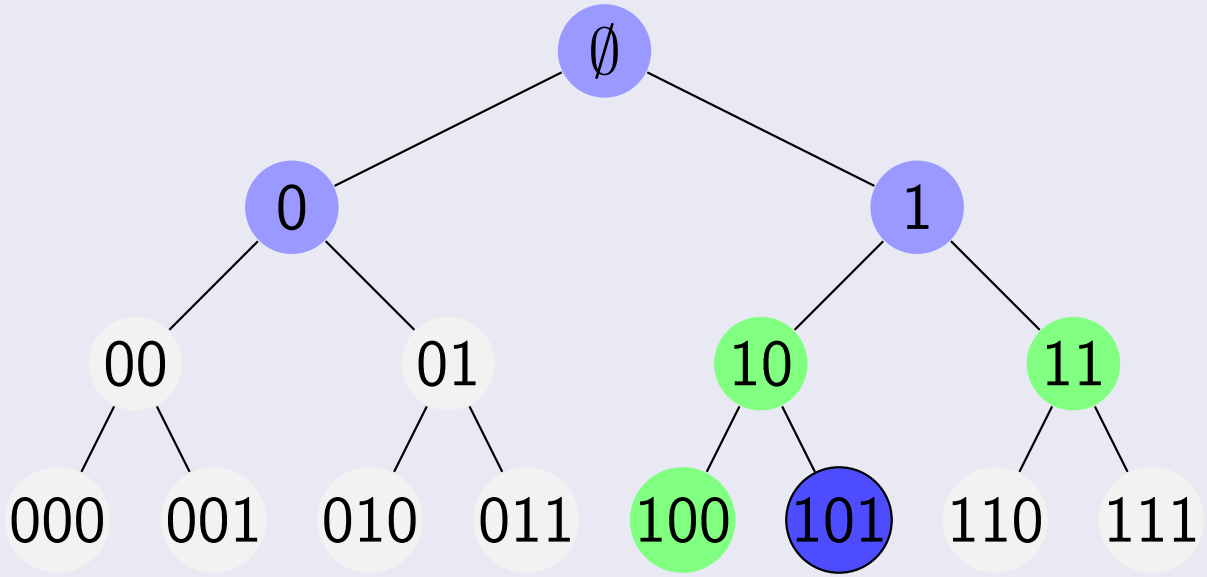
Proving selective security

Hybrid H_4



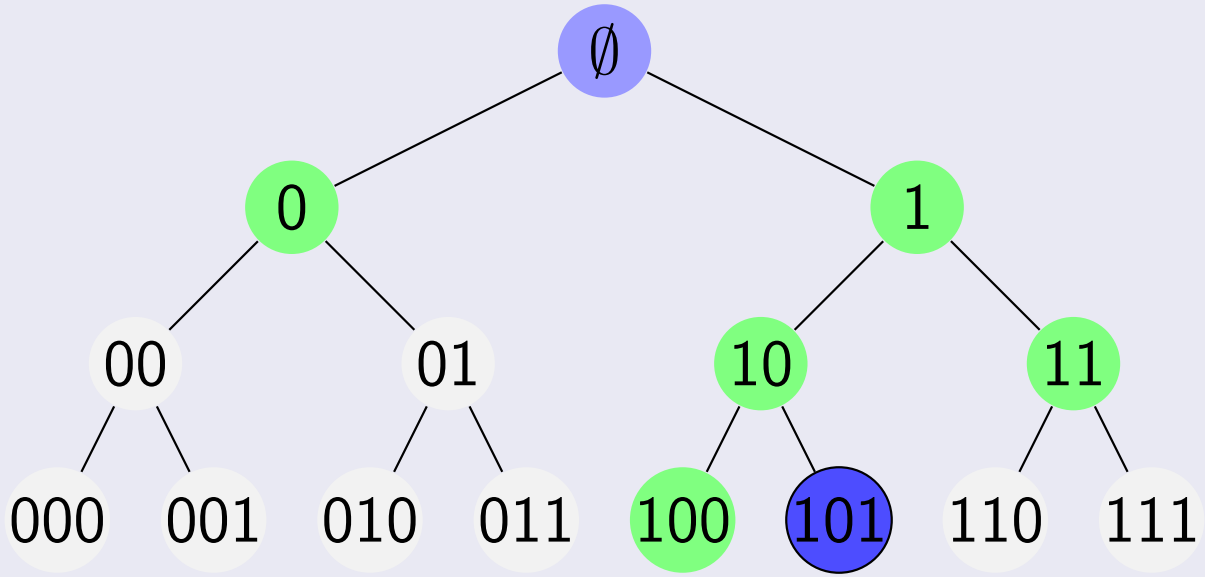
Proving selective security

Hybrid H_5



Proving selective security

Hybrid H_6 (random game)

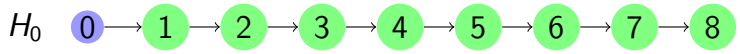


- $\mathsf{Adv}(H_0, H_{2n}) = \epsilon$

$\Rightarrow$ $\mathsf{Adv}(H_i, H_{i+1}) \geq \epsilon/2n$

$\Rightarrow$ $\mathsf{Adv}(\mathsf{G}(U_\lambda), U_{2\lambda}) \geq \epsilon/2n$

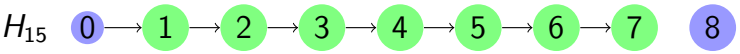
Proving adaptive security using leveraging



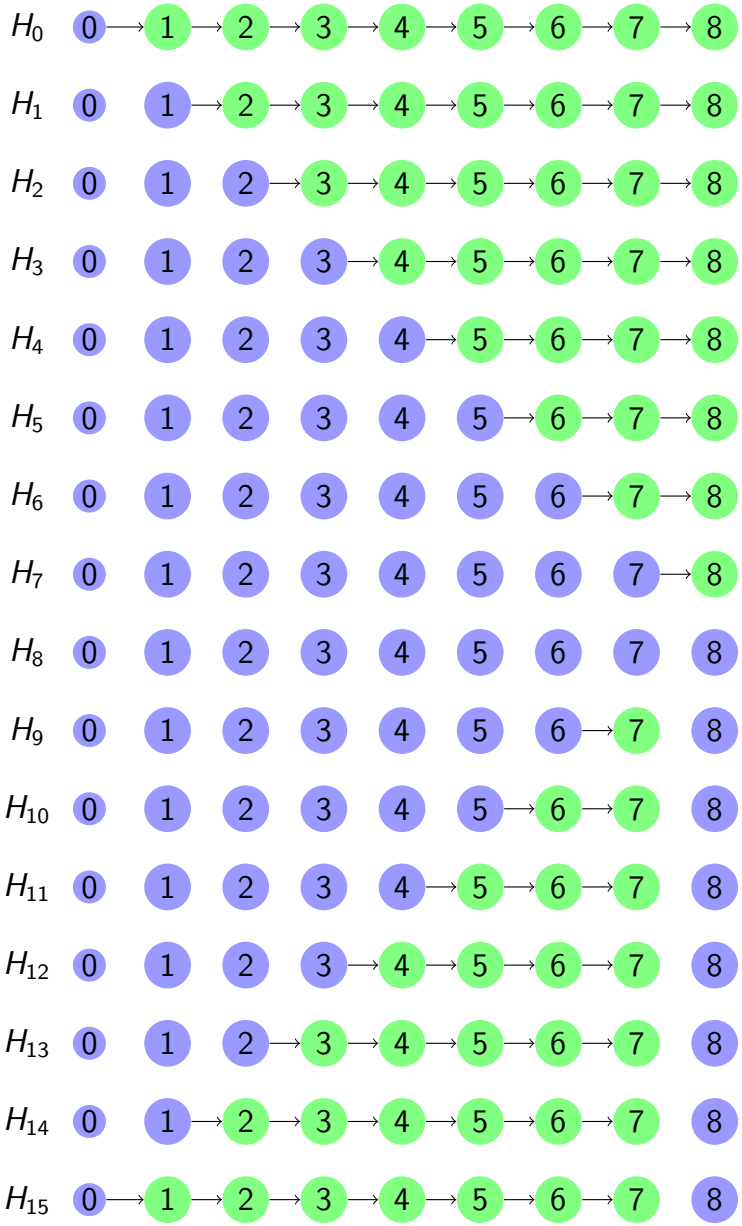
Proof

- Leveraging: guess challenge

$$\epsilon \rightarrow \frac{\epsilon}{2^n}$$



Proving adaptive security using leveraging



Proof

- Leveraging: guess challenge
- Hybrid Argument

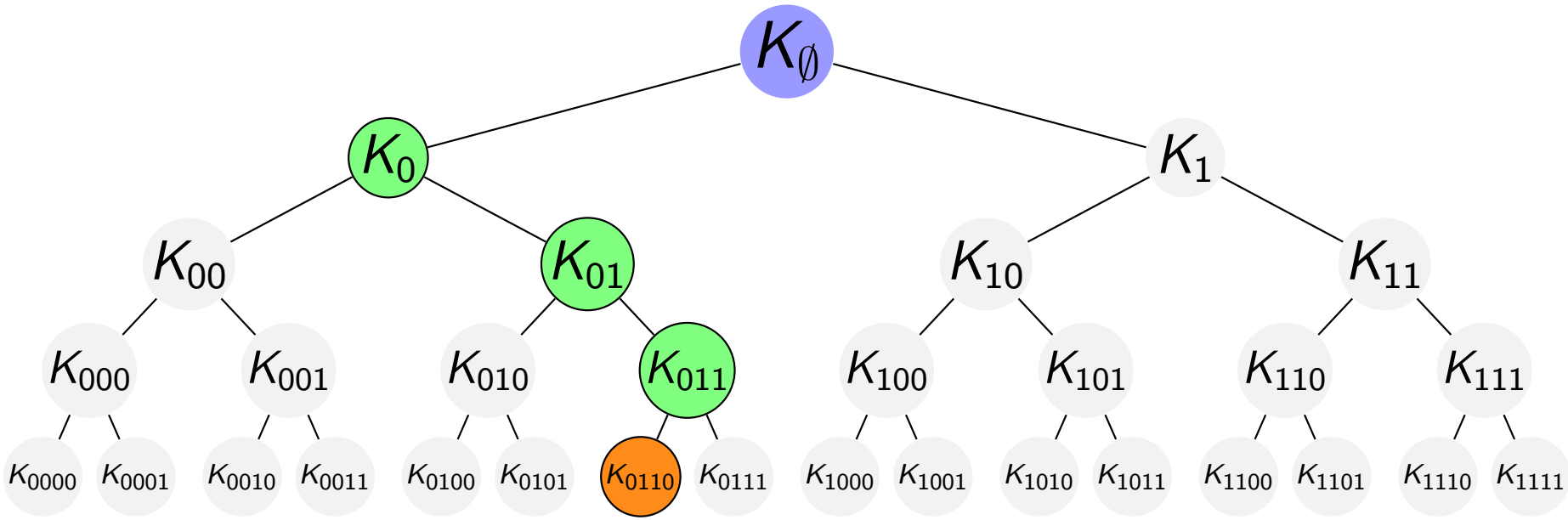
$$\epsilon \rightarrow \frac{\epsilon}{2^n} \rightarrow \frac{\epsilon}{2^n \cdot 2n}$$

GGM Constrained PRF

Selective proof strategy

- need to embed random points on path to x^*
- but don't know x^* $\Rightarrow$ could guess x^*

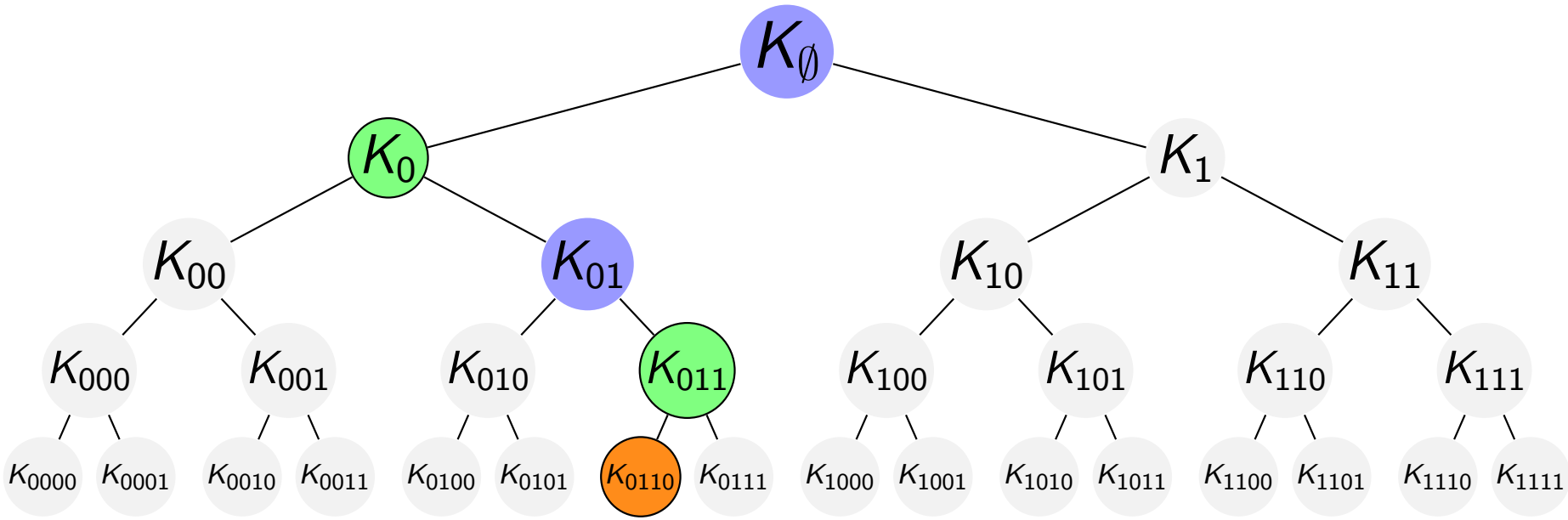
  $\Rightarrow$ lose 2^n



GGM Constrained PRF

Halving the game

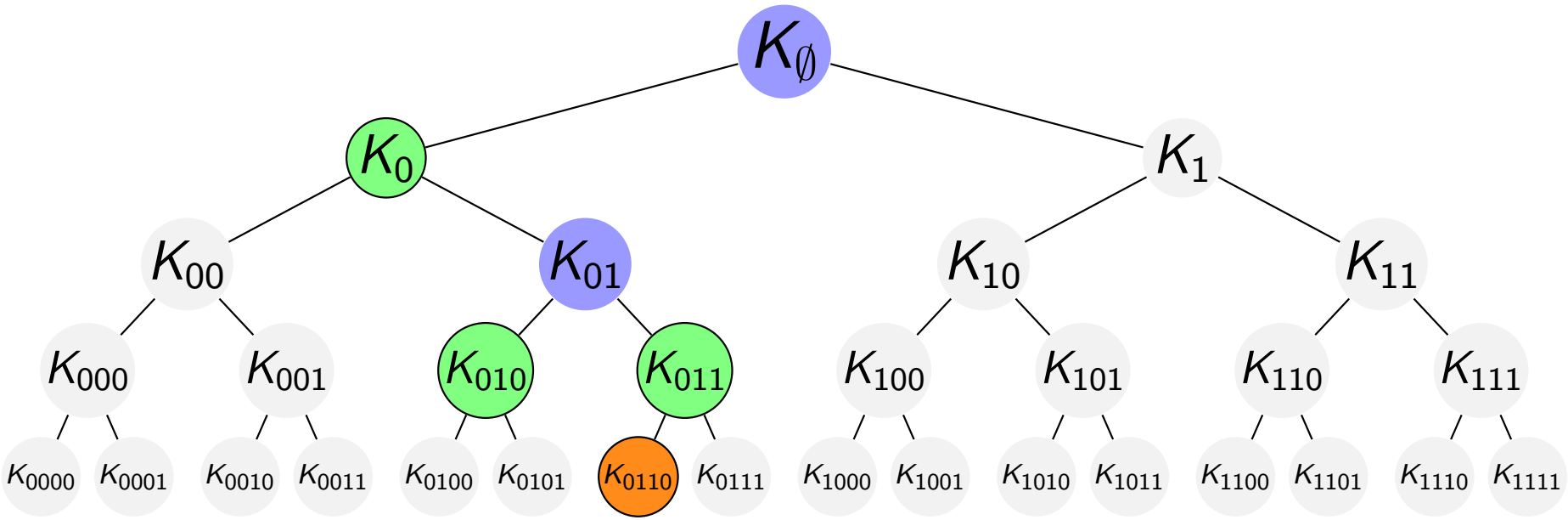
- embed random point on half the way
- wait until x^* is queried



GGM Constrained PRF

Halving the game

- embed random point on half the way
- wait until x^* is queried
- problem: what if there was a query before?

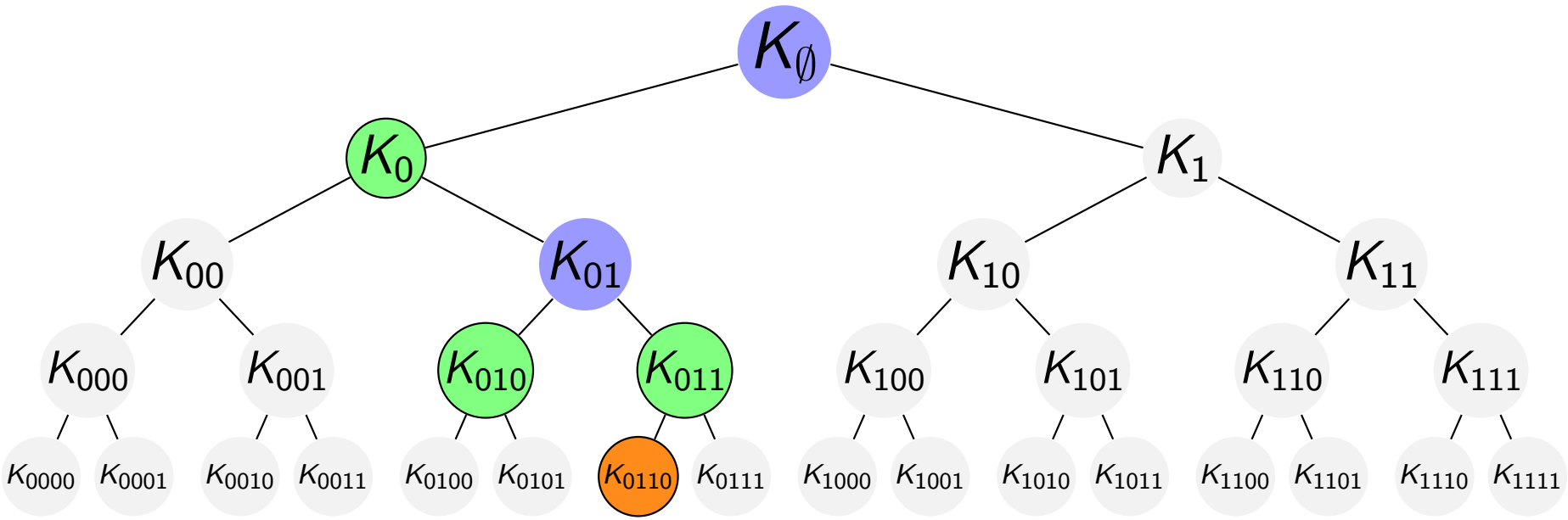


GGM Constrained PRF

New proof strategy

- guess which query is first with prefix $x^*[1 \dots n/2]$

  $\Rightarrow$ lose $1/q$

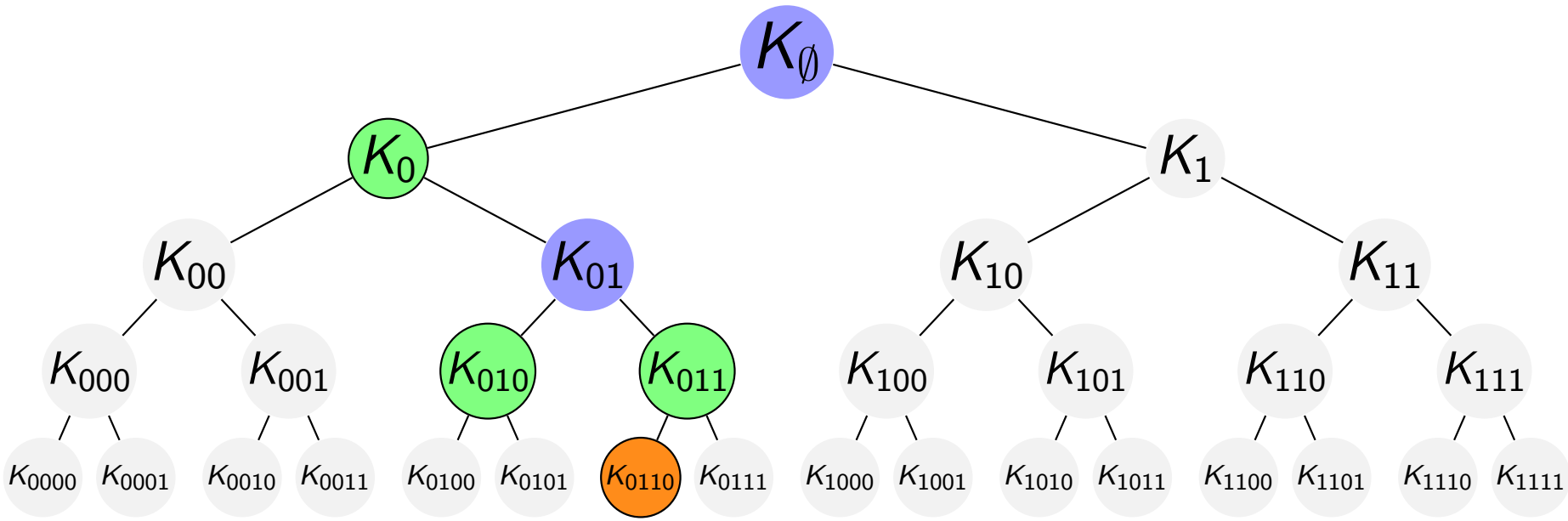


GGM Constrained PRF

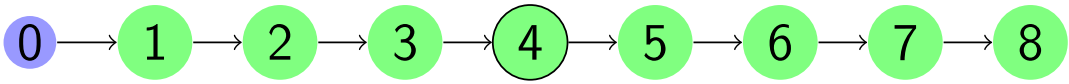
New proof strategy

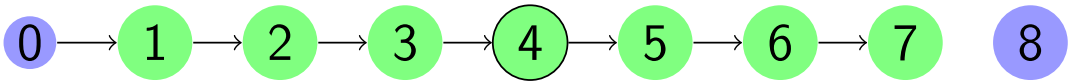
- guess which query is first with prefix $x^*[1 \dots n/2]$

  $\Rightarrow$ lose $1/q$
- if guess correct, can simulate correctly



Proving adaptive security by nesting



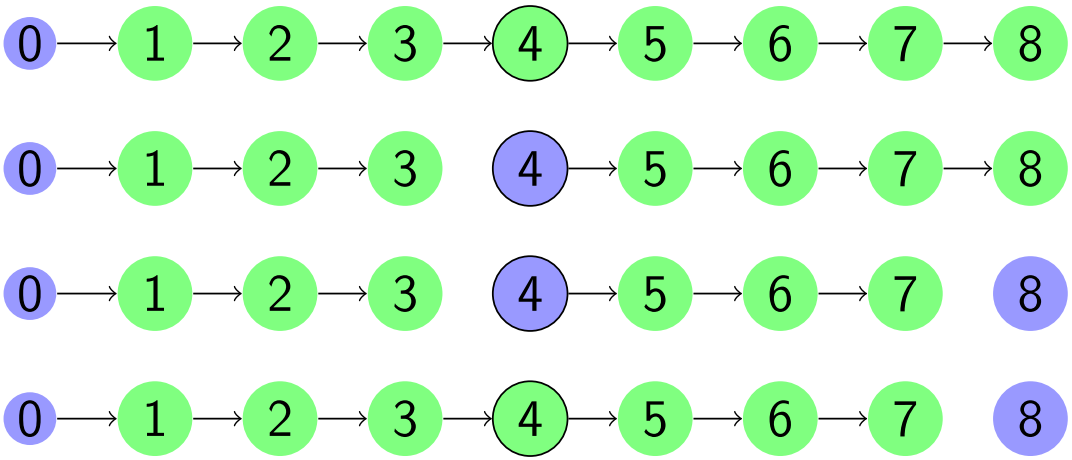


Proof

1. Guess first query that agrees with x^* on 4-prefix.

$$\epsilon \longrightarrow \frac{\epsilon}{q}$$

Proving adaptive security by nesting

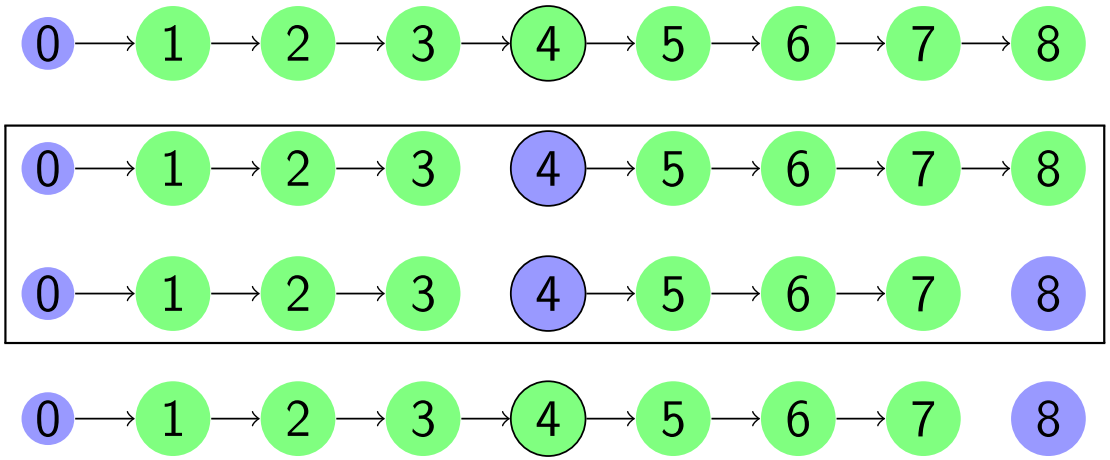


Proof

1. Guess first query that agrees with x^* on 4-prefix.

$$\epsilon \longrightarrow \frac{\epsilon}{q}$$

Proving adaptive security by nesting

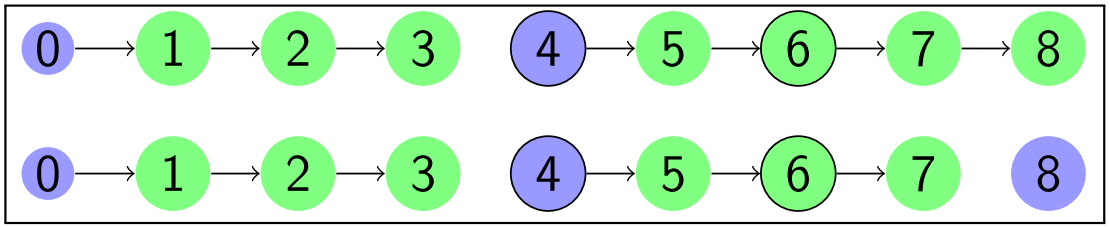


Proof

2. Hybrid argument.

$$\epsilon \rightarrow \frac{\epsilon}{q} \rightarrow \frac{\epsilon}{3q}$$

Proving adaptive security by nesting

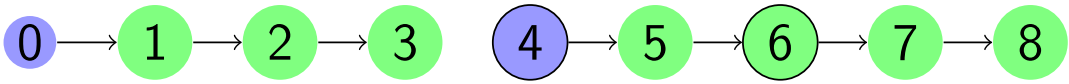


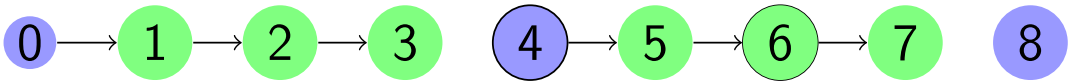
Proof

1. Guess first query that agrees with x^* on 6-prefix.

$$\epsilon \to \frac{\epsilon}{q} \to \frac{\epsilon}{3q} \to \frac{\epsilon}{3q^2}$$

Proving adaptive security by nesting



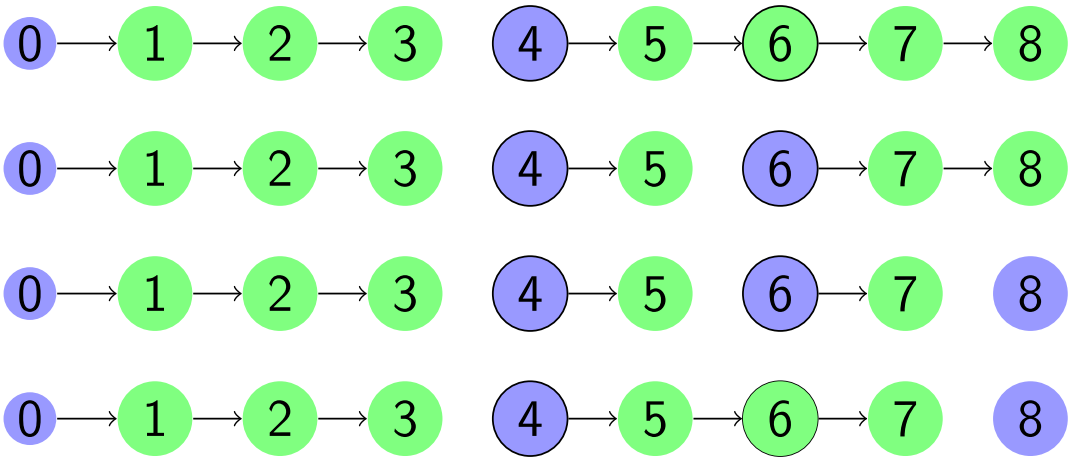


Proof

1. Guess first query that agrees with x^* on 6-prefix.

$$\epsilon \to \frac{\epsilon}{q} \to \frac{\epsilon}{3q} \to \frac{\epsilon}{3q^2}$$

Proving adaptive security by nesting

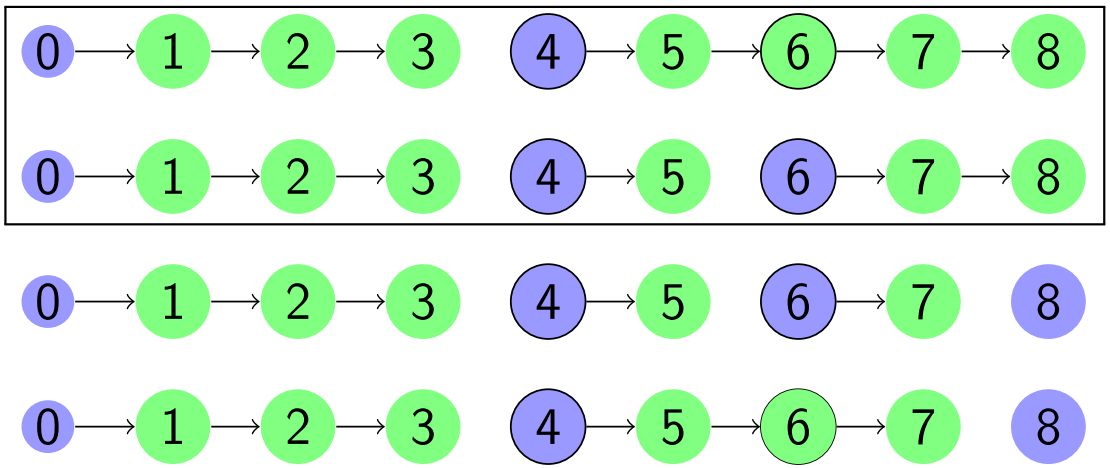


Proof

1. Guess first query that agrees with x^* on 6-prefix.

$$\epsilon \to \frac{\epsilon}{q} \to \frac{\epsilon}{3q} \to \frac{\epsilon}{3q^2}$$

Proving adaptive security by nesting

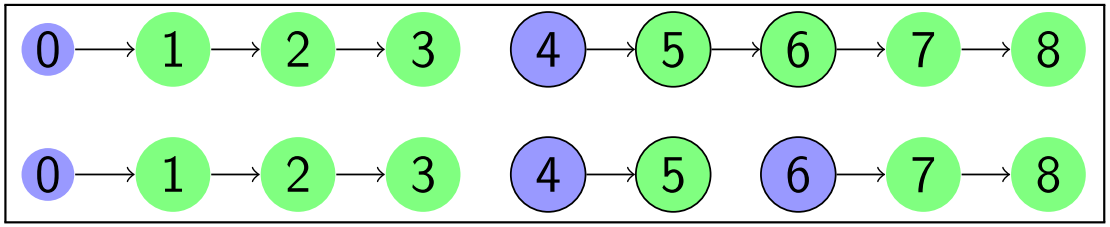


Proof

2. Hybrid argument.

$$\epsilon \to \frac{\epsilon}{q} \to \frac{\epsilon}{3q} \to \frac{\epsilon}{3q^2} \to \frac{\epsilon}{3^2 q^2}$$

Proving adaptive security by nesting

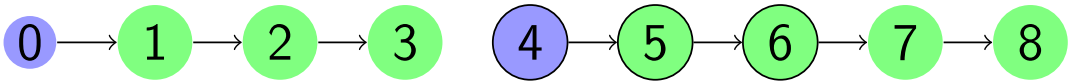


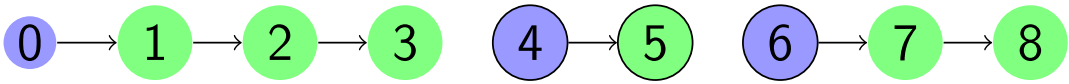
Proof

1. Guess first query that agrees with x^* on 5-prefix.

$$\epsilon \to \frac{\epsilon}{q} \to \frac{\epsilon}{3q} \to \frac{\epsilon}{3q^2} \to \frac{\epsilon}{3^2q^2} \to \frac{\epsilon}{3^2q^3}$$

Proving adaptive security by nesting



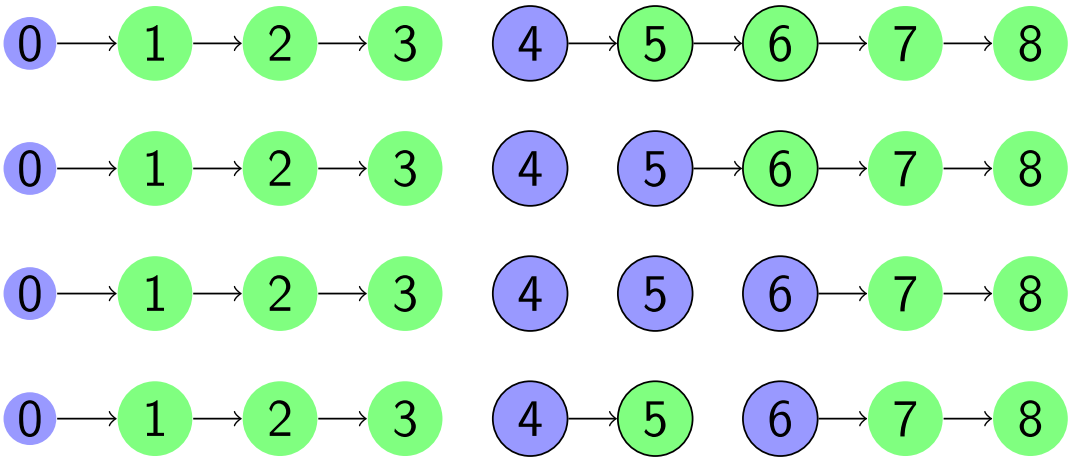


Proof

1. Guess first query that agrees with x^* on 5-prefix.

$$\epsilon \to \frac{\epsilon}{q} \to \frac{\epsilon}{3q} \to \frac{\epsilon}{3q^2} \to \frac{\epsilon}{3^2q^2} \to \frac{\epsilon}{3^2q^3}$$

Proving adaptive security by nesting

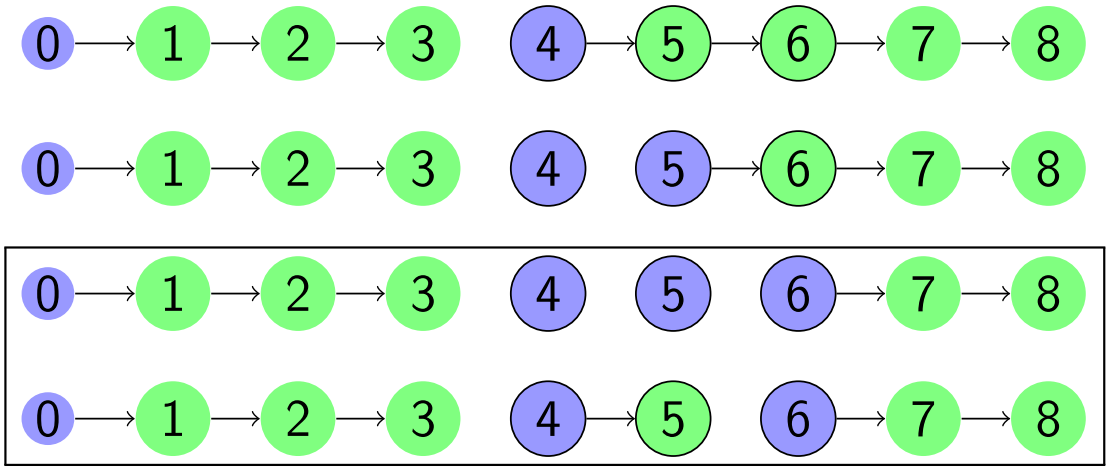


Proof

1. Guess first query that agrees with x^* on 5-prefix.

$$\epsilon \to \frac{\epsilon}{q} \to \frac{\epsilon}{3q} \to \frac{\epsilon}{3q^2} \to \frac{\epsilon}{3^2q^2} \to \frac{\epsilon}{3^2q^3}$$

Proving adaptive security by nesting



Proof

2. Hybrid argument.

$$\epsilon \to \frac{\epsilon}{q} \to \frac{\epsilon}{3q} \to \frac{\epsilon}{3q^2} \to \frac{\epsilon}{3^2q^2} \to \frac{\epsilon}{3^2q^3} \to \frac{\epsilon}{(3q)^{\log n}}$$

Our result

Theorem

If $\mathsf{G}\colon \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ *is an* $(\epsilon_\mathsf{G}, s_\mathsf{G})$*-secure PRG then (for any* n, q*)* $\mathsf{GGM}^\mathsf{G}\colon \{0,1\}^n \to \{0,1\}^\lambda$ *is a* **fully** (ϵ, s, q)*-secure constrained PRF for* $\mathcal{S}_{\mathsf{pre}}$*, where*

$$\epsilon = \epsilon_\mathsf{G} \cdot (3q)^{\log n} \qquad s = s_\mathsf{G} - O(q \cdot n \cdot |\mathsf{G}|)$$

Related work

Adaptive security for CPRFs

Poly-time reductions for

- *puncturable* ($\subsetneq$ prefix-fixing) PRF using indistinguishability obfuscation (iO)

[HKW14] Hohenberger, Koppula, Waters. *Adaptively secure puncturable pseudorandom functions in the standard model.* IACR eprint 2014/521

Related work

Adaptive security for CPRFs

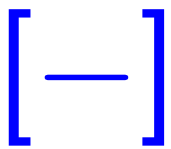
Poly-time reductions for

- *puncturable* ($\subsetneq$ prefix-fixing) PRF using indistinguishability obfuscation (iO)
- *circuit-constrained* PRF using iO in the random-oracle model.

[Hof14] Hofheinz. *Fully secure constrained pseudorandom functions using random oracles.* IACR eprint 2014/372

[HKW14] Hohenberger, Koppula, Waters. *Adaptively secure puncturable pseudorandom functions in the standard model.* IACR eprint 2014/521

[HKKW14] Hofheinz, Kamath, Koppula, Waters: *Adaptively Secure Constrained Pseudorandom Functions.* IACR eprint 2014/720



The Boneh-Waters bit-fixing PRF

The Boneh-Waters bit-fixing PRF

- Construction based on multilinear maps
- Proved (tightly) selectively secure under multilinear DDH

The Boneh-Waters bit-fixing PRF

The Boneh-Waters bit-fixing PRF

- Construction based on multilinear maps
- Proved (tightly) selectively secure under multilinear DDH

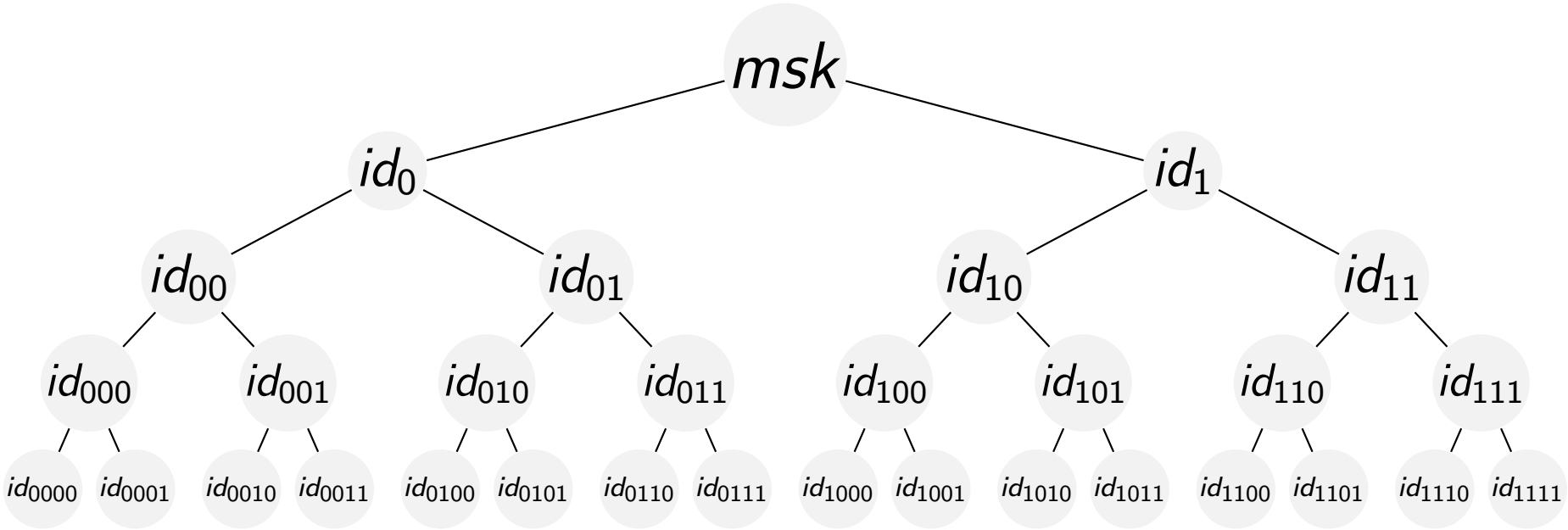
Our results

- We show that exponential loss for adaptive security is inherent (even when restricted to prefix-fixing)
- Re-use a technique by Lewko and Waters, who show exponential loss for certain HIBE systems

[LW14] Lewko, Waters: *Why proving HIBE systems secure is difficult.* EUROCRYPT 2014

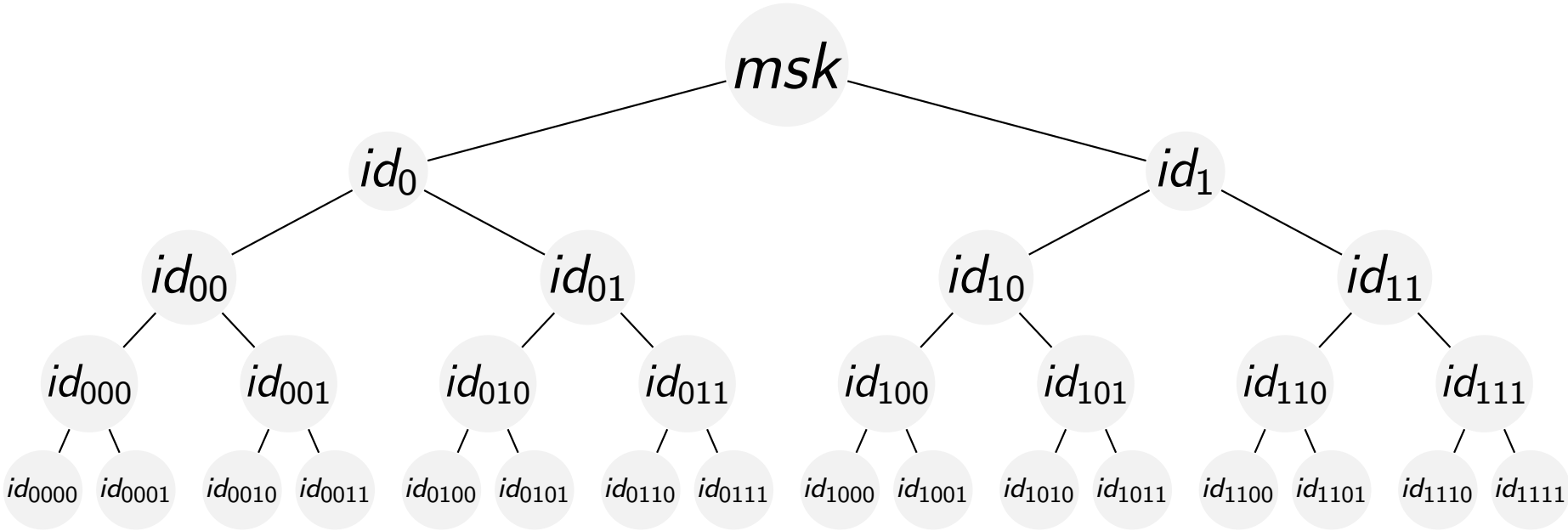
Lewko-Waters: Why proving HIBE is difficult

Hierarchical IBE:



Lewko-Waters: Why proving HIBE is difficult

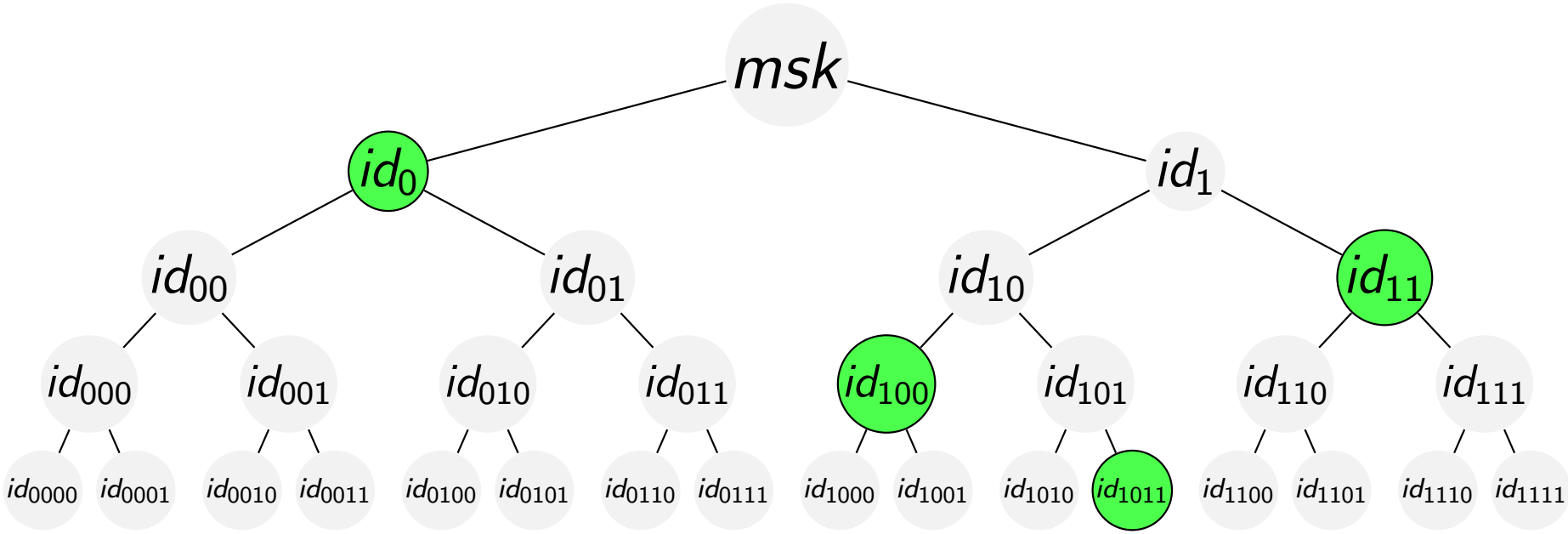
Hierarchical IBE:



- [LW14] show that partitioning proofs must lose exponential factor in depth of HIBE:

Lewko-Waters: Why proving HIBE is difficult

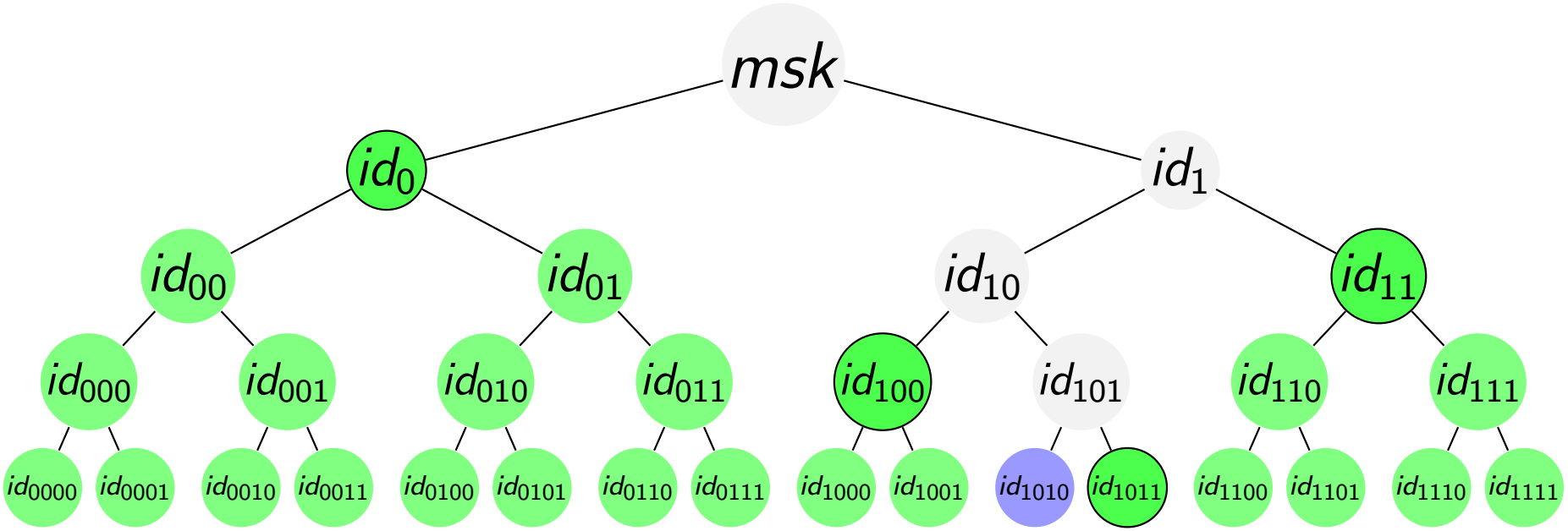
Hierarchical IBE:



- [LW14] show that partitioning proofs must lose exponential factor in depth of HIBE:
- $\mathcal{A}$ could query keys to decrypt for any but one id.

Lewko-Waters: Why proving HIBE is difficult

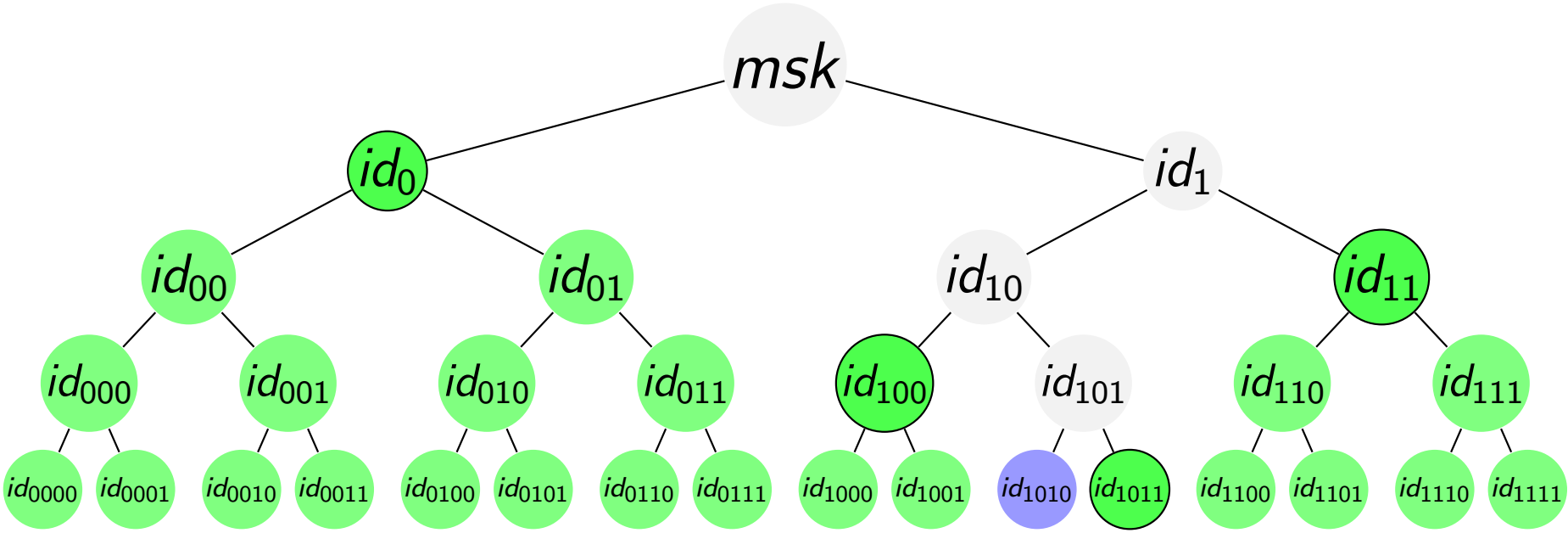
Hierarchical IBE:



- [LW14] show that partitioning proofs must lose exponential factor in depth of HIBE:
- $\mathcal{A}$ could query keys to decrypt for any but one id.
- $\Rightarrow$ reduction must guess id

Lewko-Waters: Why proving HIBE is difficult

Hierarchical IBE:



- [LW14] show that partitioning proofs must lose

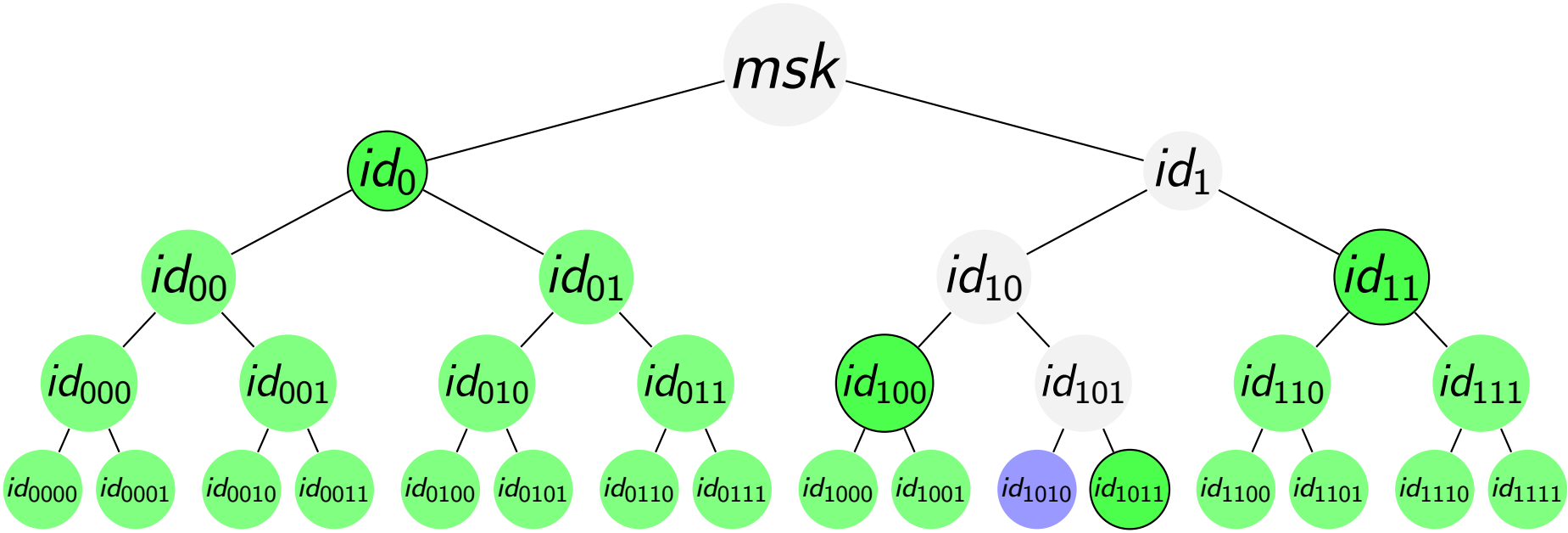
Condition

Keys have to be checkable w.r.t. pk

- $\Rightarrow$ reduction must guess id

Lewko-Waters: Why proving HIBE is difficult

Hierarchical IBE:



- [LW14] show that partitioning proofs must lose

Condition

Keys have to be checkable w.r.t. pk

- $\Rightarrow$ reduction must guess id

Result

Simple reductions for HIBE must lose factor exp. in depth

FKPV: Why proving Boneh-Waters is difficult

Applied to Boneh-Waters PRF

- There is no public key!

$\Rightarrow$ A makes 2 fingerprint queries for constrained keys

$\Rightarrow$ these fix the secret key

- Keys are checkable w.r.t. fingerprint keys using pairings

FKPV: Why proving Boneh-Waters is difficult

Applied to Boneh-Waters PRF

- There is no public key!

$\Rightarrow$ A makes 2 fingerprint queries for constrained keys

$\Rightarrow$ these fix the secret key

- Keys are checkable w.r.t. fingerprint keys using pairings

Theorem

Let $\Pi(\lambda)$ be a decisional problem such that no algorithm running in time $t = poly(\lambda)$ has an advantage non-negligible in λ. Let R be a simple $(t, \epsilon, q, \delta, t')$ reduction from Π to unpredictability of the Boneh-Waters prefix-constrained PRF with domain $\{0,1\}^n$, with both t, t' polynomial in λ, and $q \geq n - 1$. Then δ vanishes exponentially as a function of n (up to terms that are negligible in λ).

Thank you! ☺