How to Use (Plain) Witness Encryption: Registered ABE, Flexible Broadcast, and More

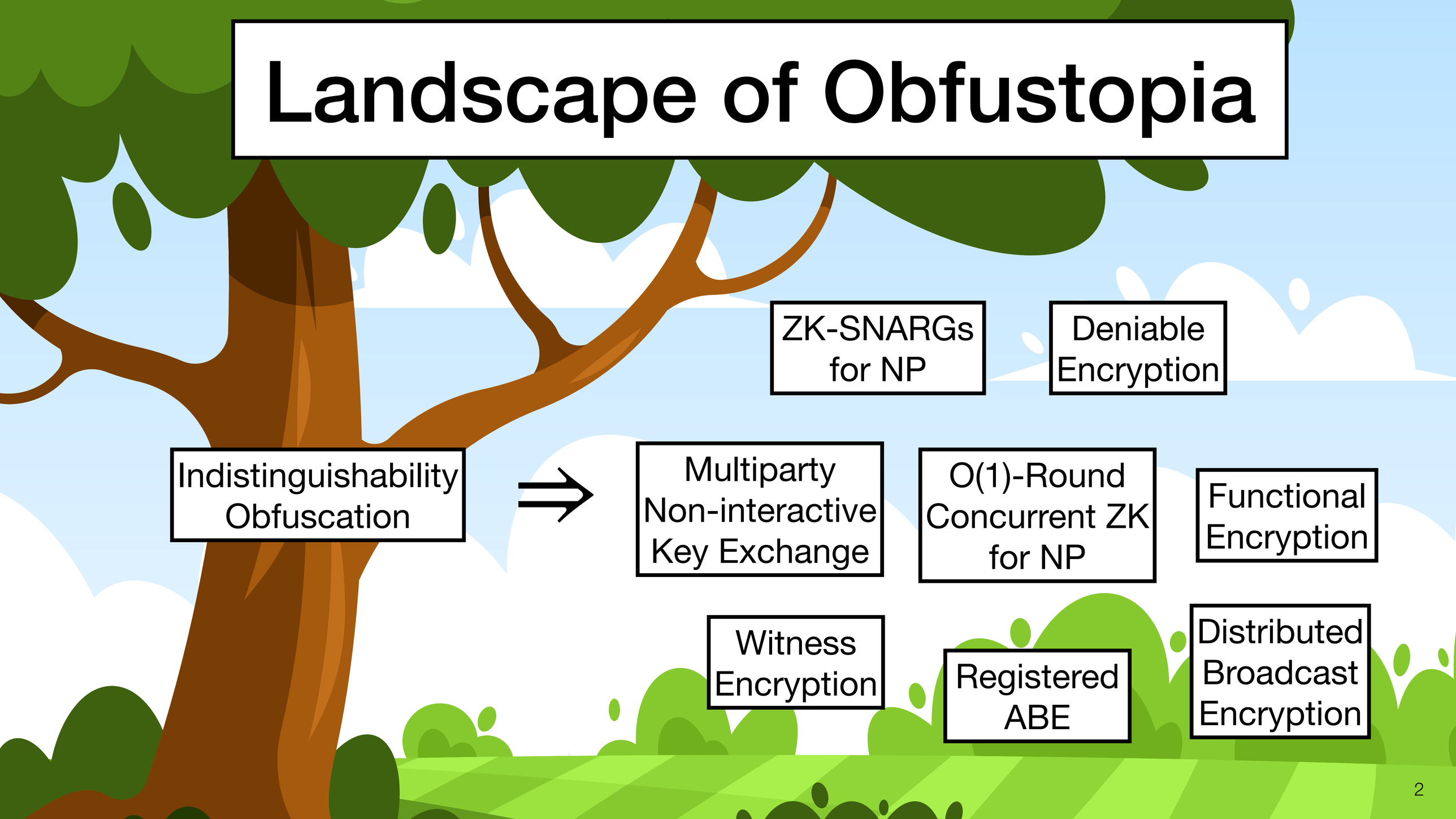
Cody Freitag[1], Brent Waters[2], David J. Wu[3]

[1]Boston University
[2]UT Austin and NTT Research
[3]UT Austin

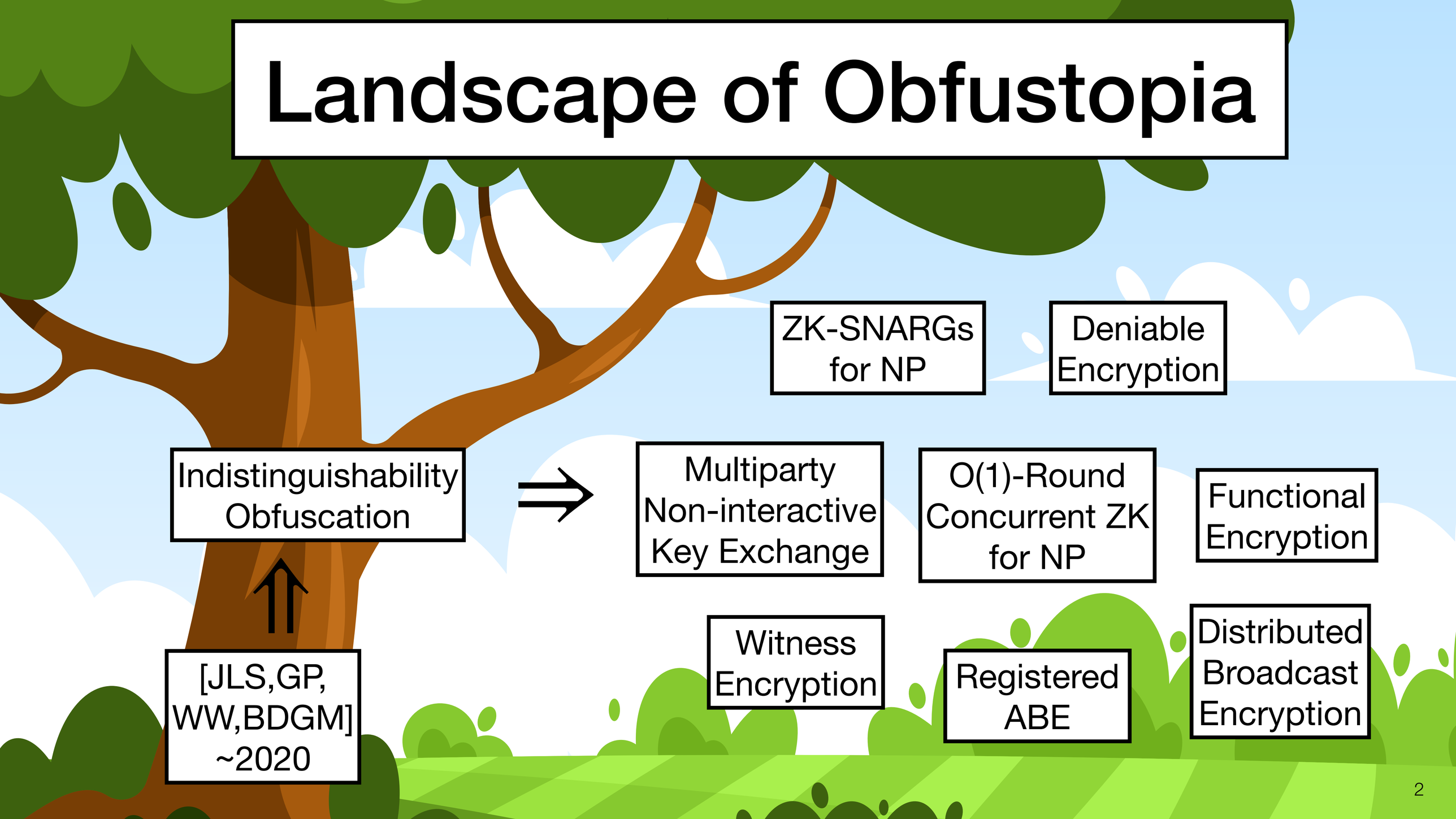
Landscape of Obfustopia

Indistinguishability Obfuscation

Landscape of Obfustopia



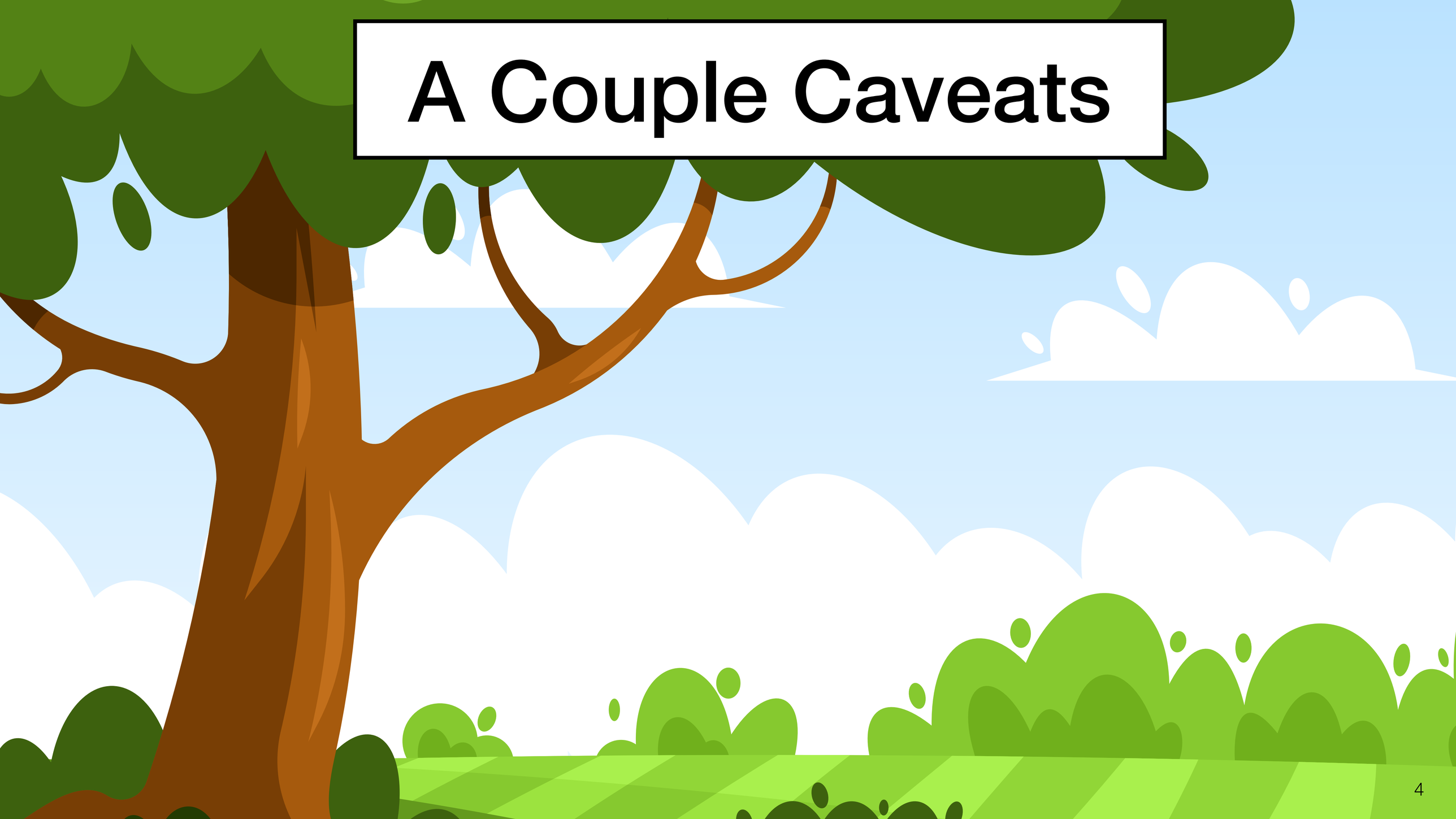
Landscape of Obfustopia



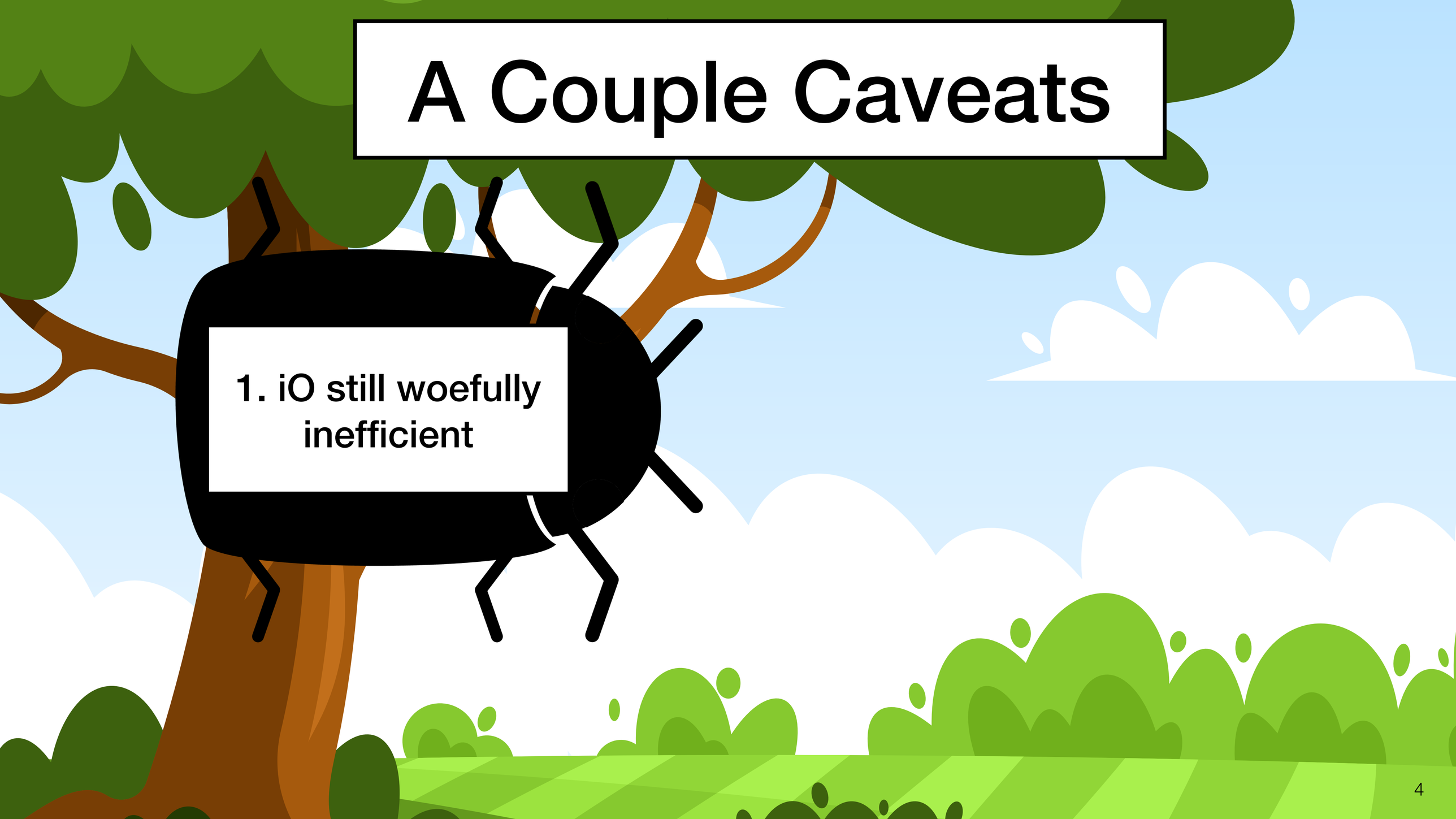
Landscape of Obfustopia

Where are the obfuscation startups?

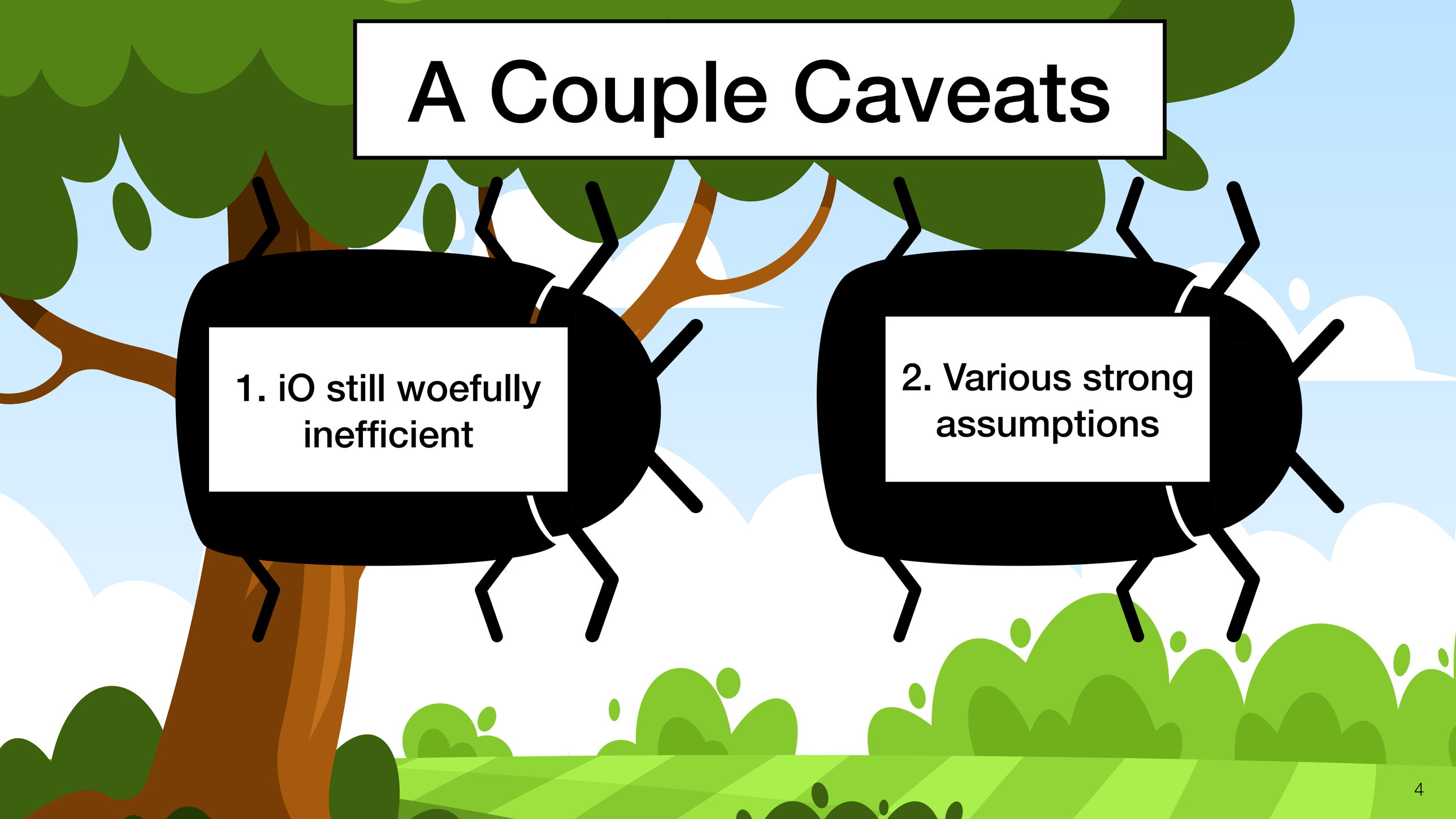
A Couple Caveats



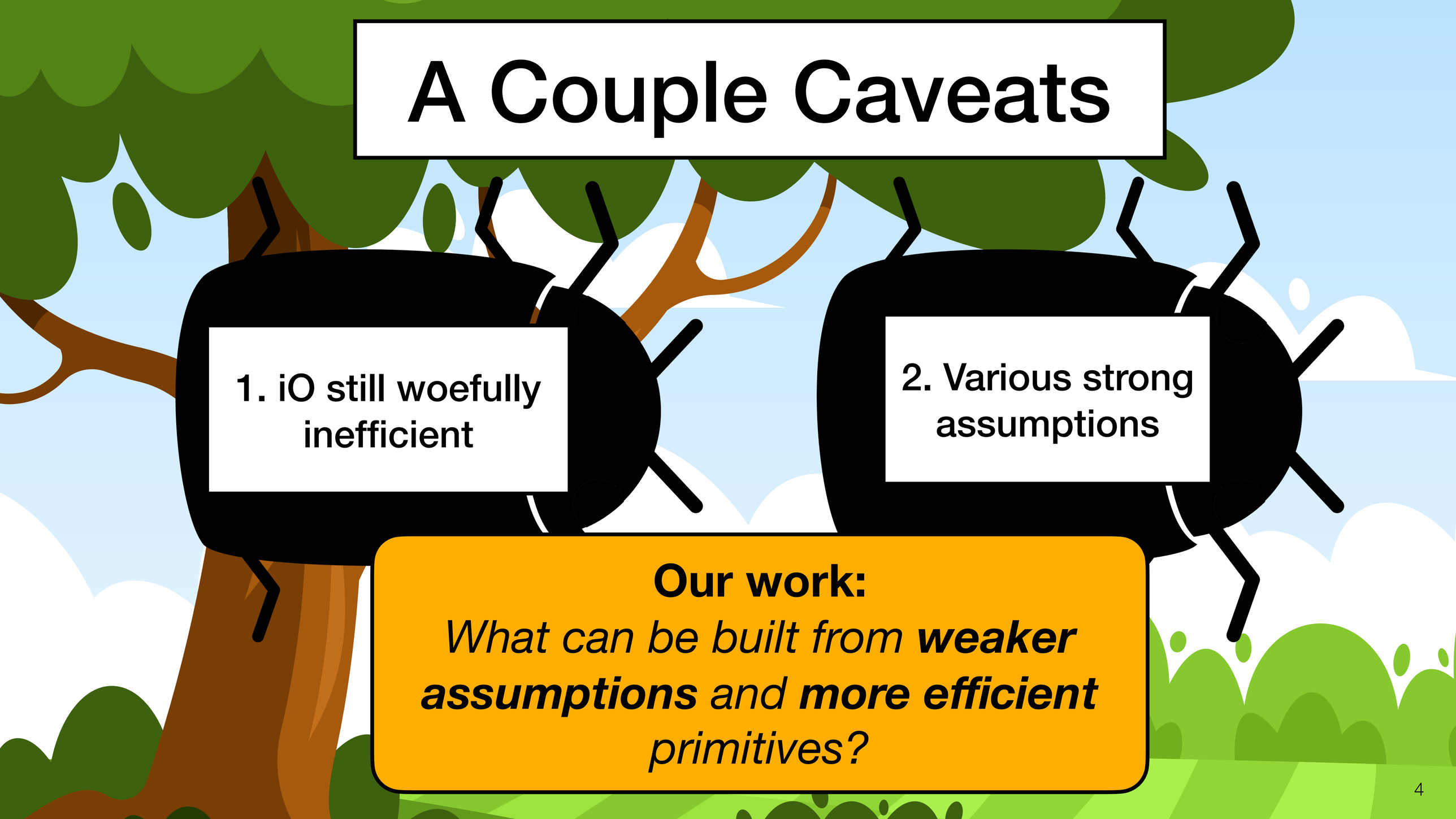
A Couple Caveats



A Couple Caveats

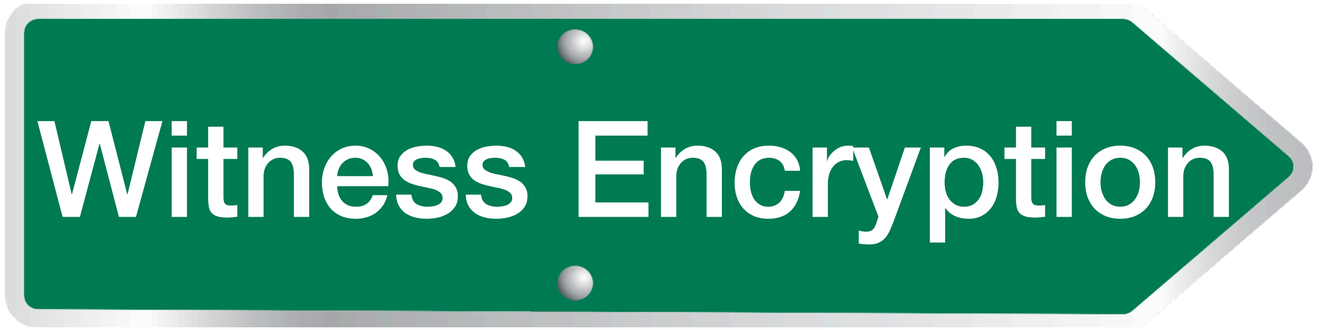


A Couple Caveats



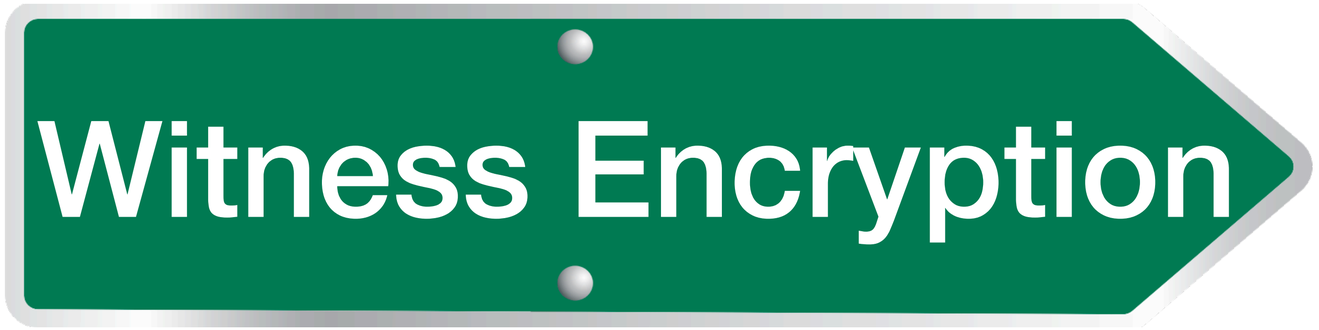
Our work:
What can be built from ***weaker assumptions*** *and* ***more efficient*** *primitives?*

A Promising Avenue:



[GGSW13]

A Promising Avenue:

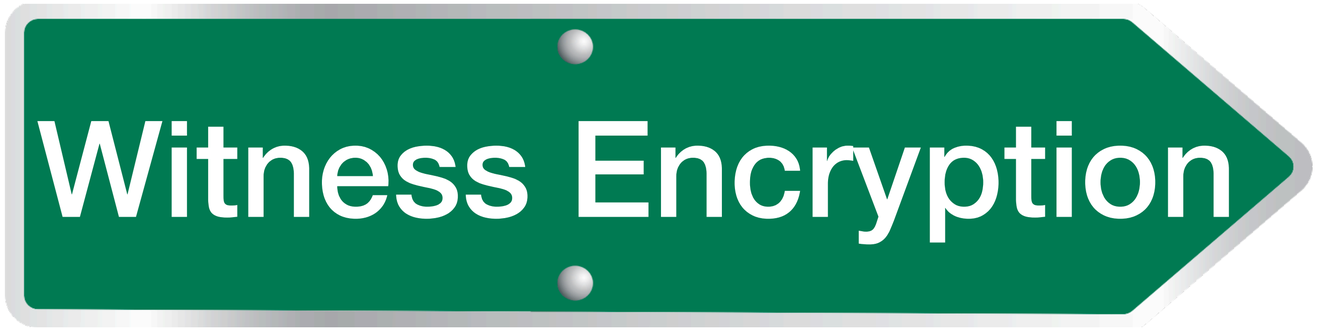


[GGSW13]

Series of recent works
[GMM17,BJKPW18,BIOW20,T22,VWW22]

1. New constructions from weaker assumptions

A Promising Avenue:



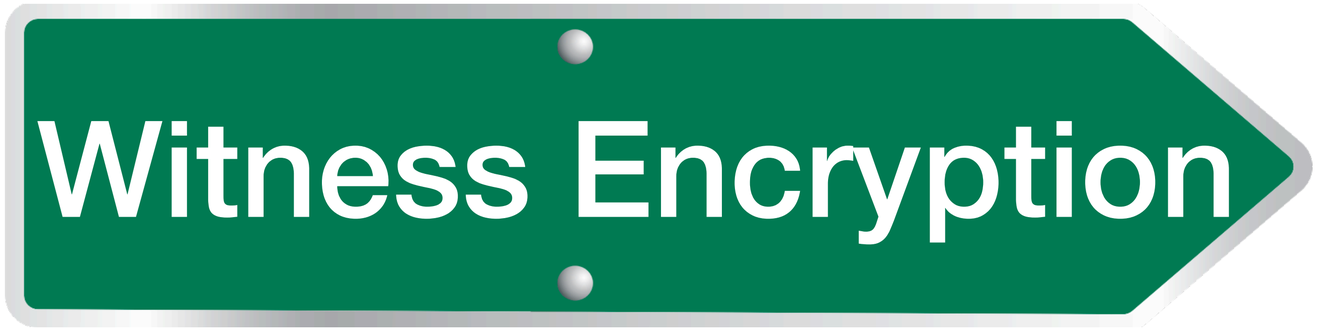
[GGSW13]

Series of recent works
[GMM17,BJKPW18,BIOW20,T22,VWW22]

1. New constructions from weaker assumptions

2. Simpler and more efficient than iO

A Promising Avenue:



[GGSW13]

Series of recent works
[GMM17,BJKPW18,BIOW20,T22,VWW22]

1. New constructions from weaker assumptions

2. Simpler and more efficient than iO

3. Provably "weaker" than iO

Our Work:
A New Framework for Witness Encryption

Our Work:
A New Framework for Witness Encryption

Existing iO Framework

[SW14,HW15]

Indistinguishability Obfuscation

Somewhere Stat. Binding Hash

Our Work:
A New Framework for Witness Encryption

Existing iO Framework

[SW14,HW15]

Indistinguishability Obfuscation

Somewhere Stat. Binding Hash

New WE Framework

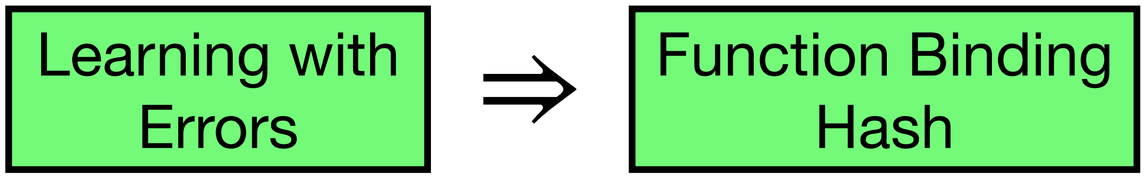
Witness Encryption

Function Binding Hash

New contribution!

Our Main Results

Theorem 1:



Our Main Results

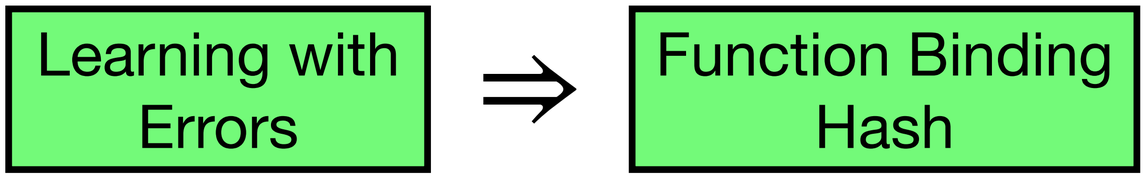
Theorem 1:

Learning with Errors $\Rightarrow$ Function Binding Hash

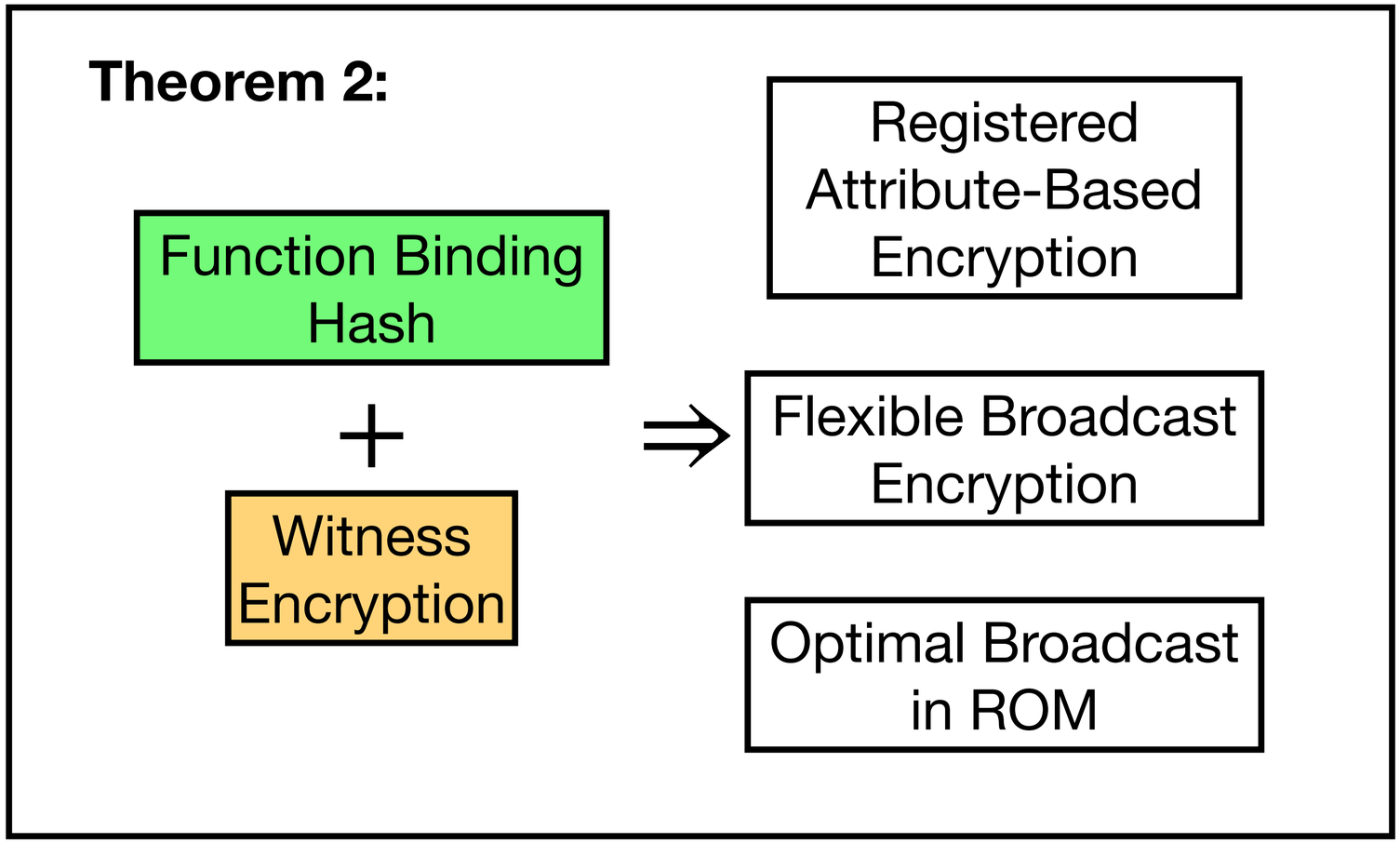
Similar techniques to SSB hash of [HW15]

Our Main Results

Theorem 1:

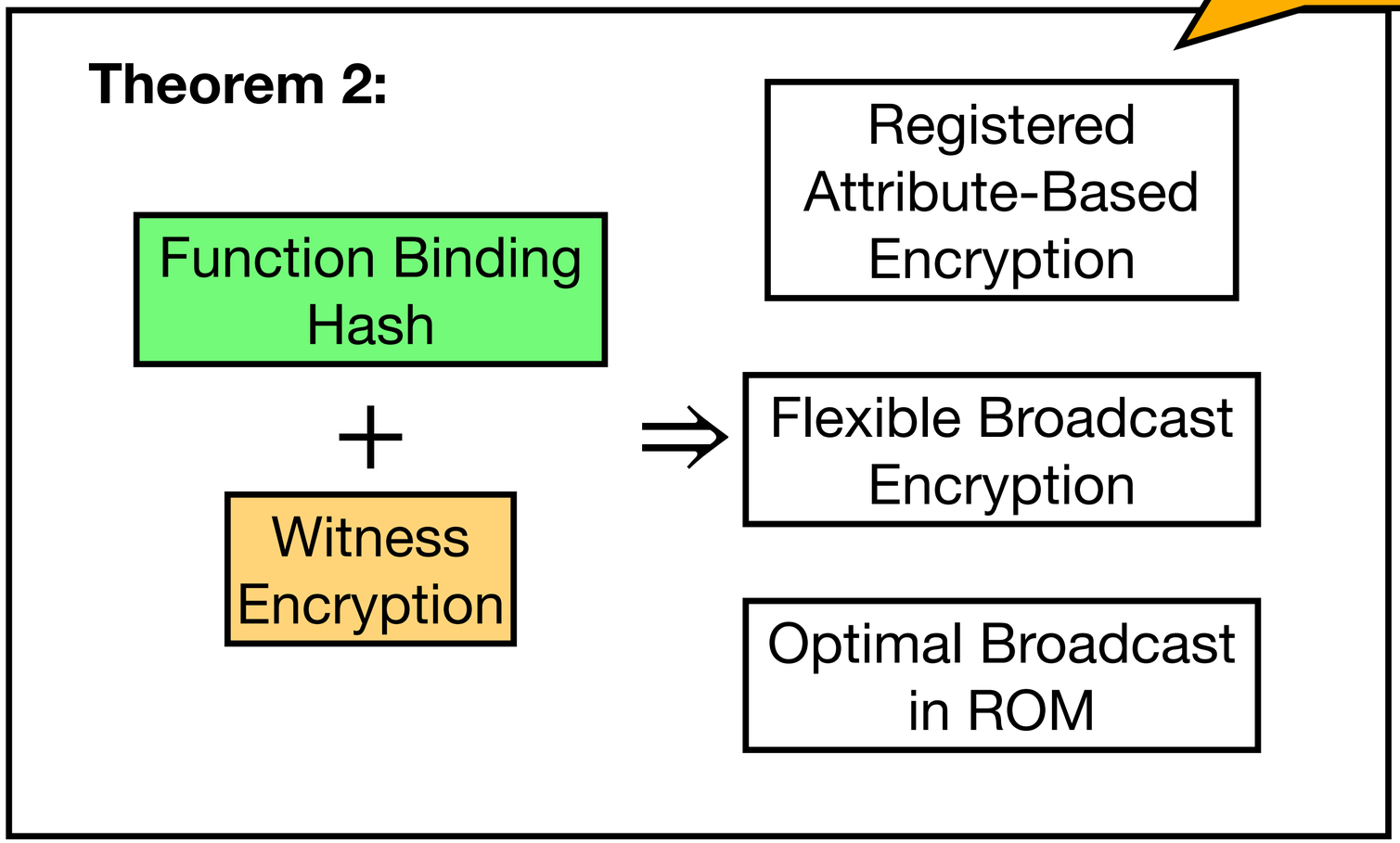


Our Main Results

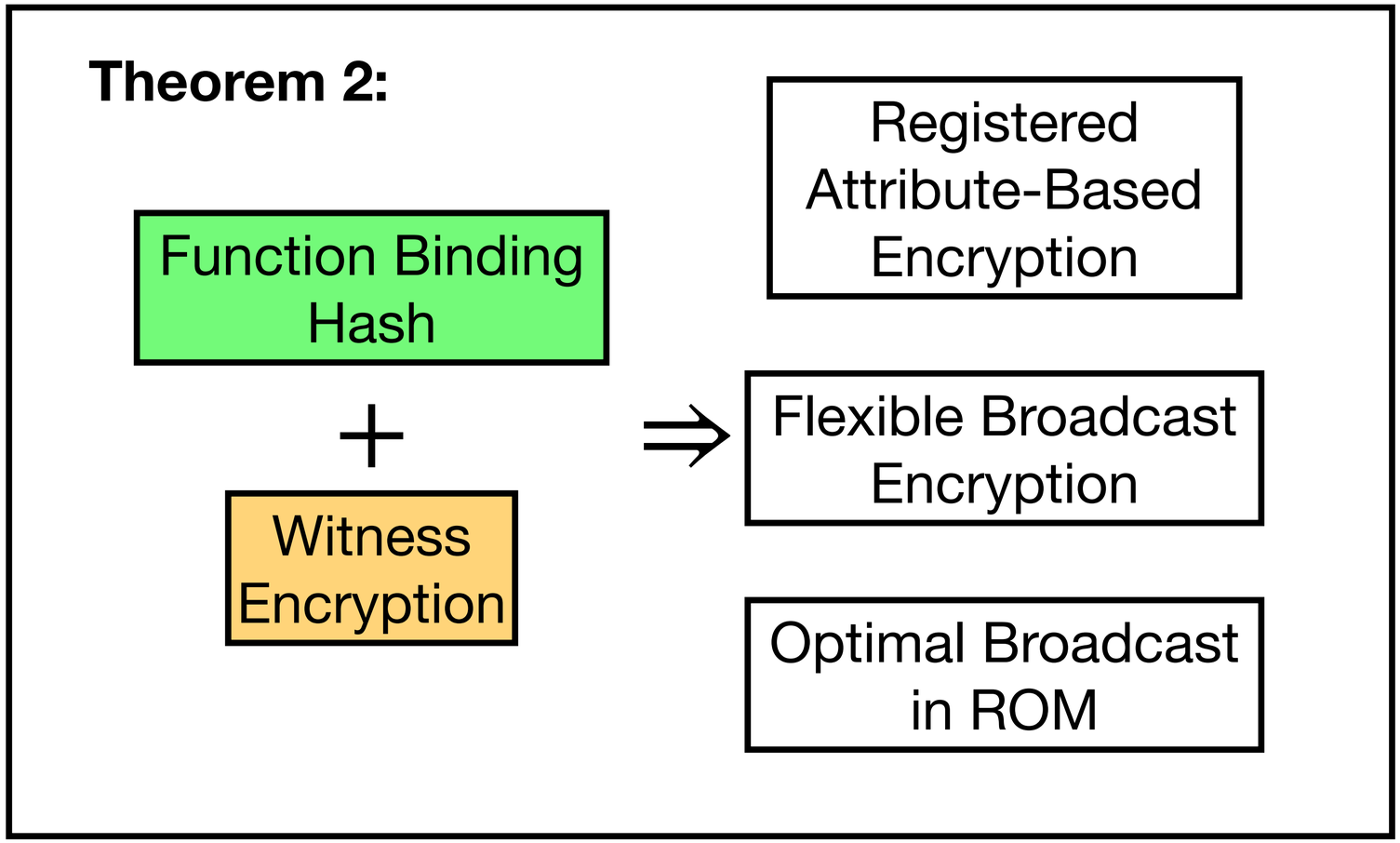


Our Main Results

Based on our new, general framework

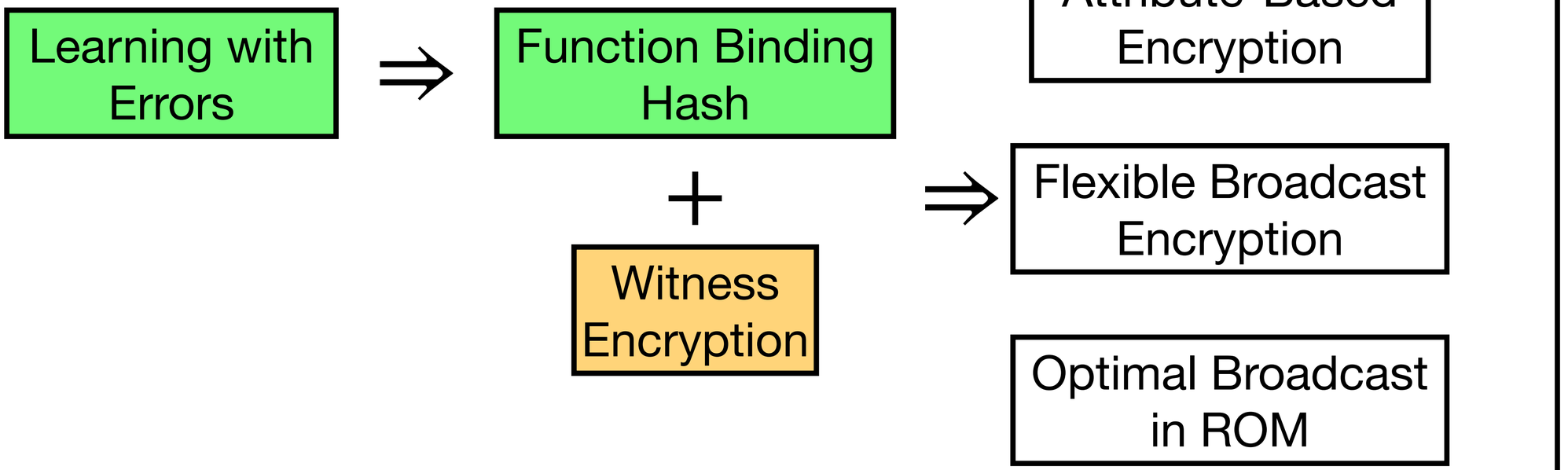


Our Main Results



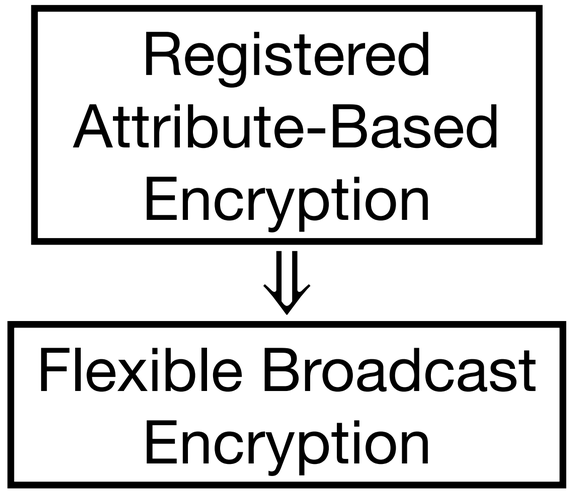
Our Main Results

Combined Theorem:

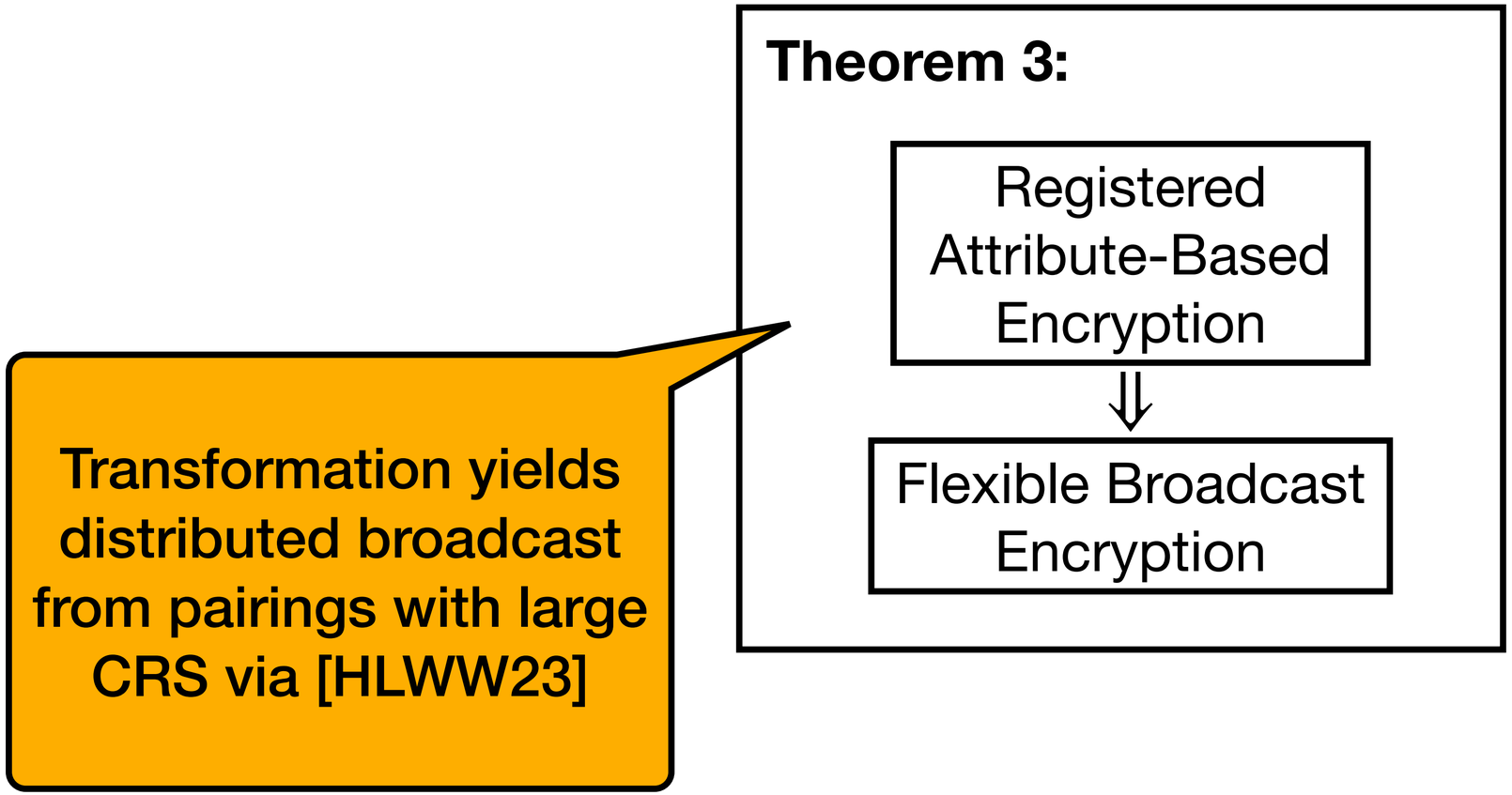


Our Main Results

Theorem 3:

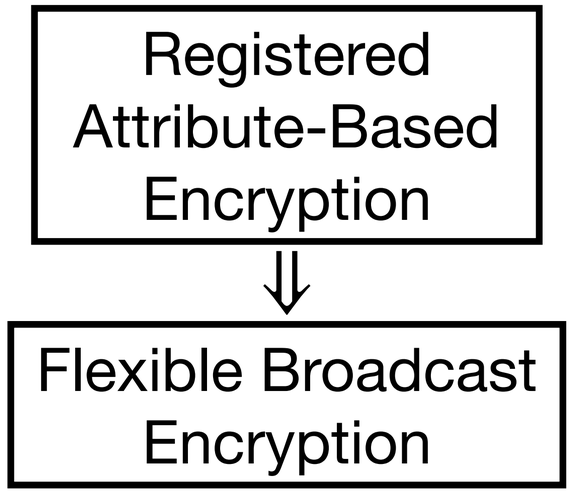


Our Main Results



Our Main Results

Theorem 3:



Detour into *Trustless* Cryptography

Detour into *Trustless* Cryptography

Functional Encryption: augment public-key encryption with fine-grained decryption capabilities [BW07,KSW08,BSW11,O'N10]

Detour into *Trustless* Cryptography

Functional Encryption: augment public-key encryption with fine-grained decryption capabilities [BW07,KSW08,BSW11,O'N10]

Limitation: secret keys are issued by a **central trusted authority**

Detour into *Trustless* Cryptography

Functional Encryption: augment public-key encryption with fine-grained decryption capabilities [BW07,KSW08,BSW11,O'N10]

Limitation: secret keys are issued by a **central trusted authority**

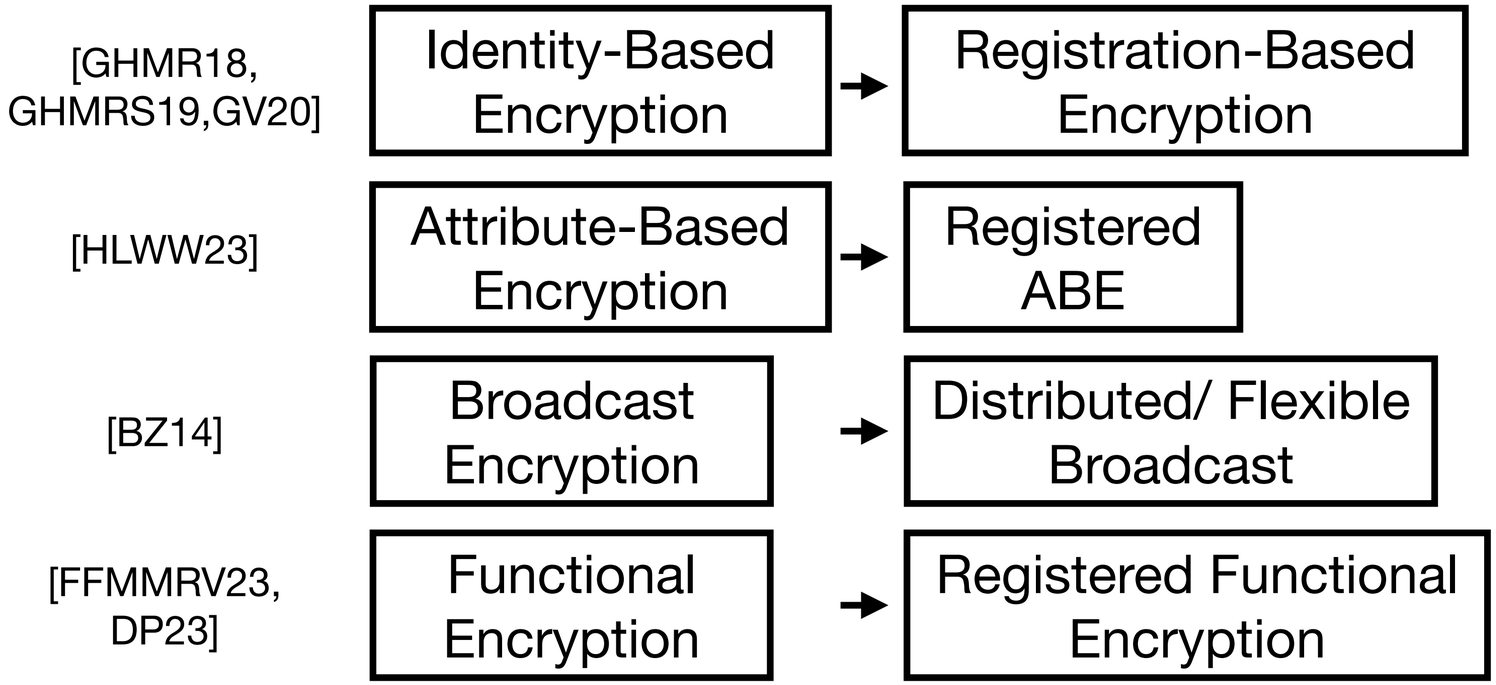
Recently: removing trust from functional encryption

Detour into *Trustless* Cryptography

Functional Encryption: augment public-key encryption with fine-grained decryption capabilities [BW07,KSW08,BSW11,O'N10]

Limitation: secret keys are issued by a **central trusted authority**

Recently: removing trust from functional encryption



Detour into *Trustless* Cryptography

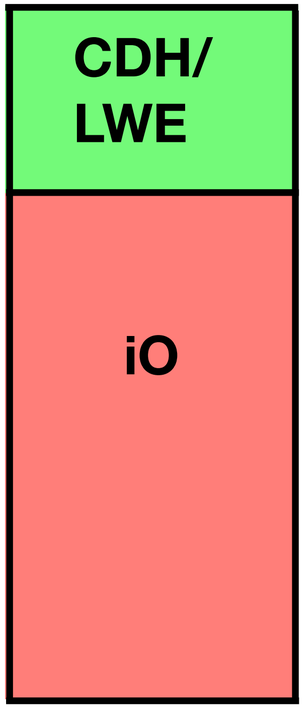
Functional Encryption: augment public-key encryption with fine-grained decryption capabilities [BW07,KSW08,BSW11,O'N10]

Limitation: secret keys are issued by a **central trusted authority**

Recently: removing trust from functional encryption

[GHMR18, GHMRS19,GV20]	Identity-Based Encryption	Registration-Based Encryption
[HLWW23]	Attribute-Based Encryption	Registered ABE
[BZ14]	Broadcast Encryption	Distributed/ Flexible Broadcast
[FFMMRV23, DP23]	Functional Encryption	Registered Functional Encryption

Previous:



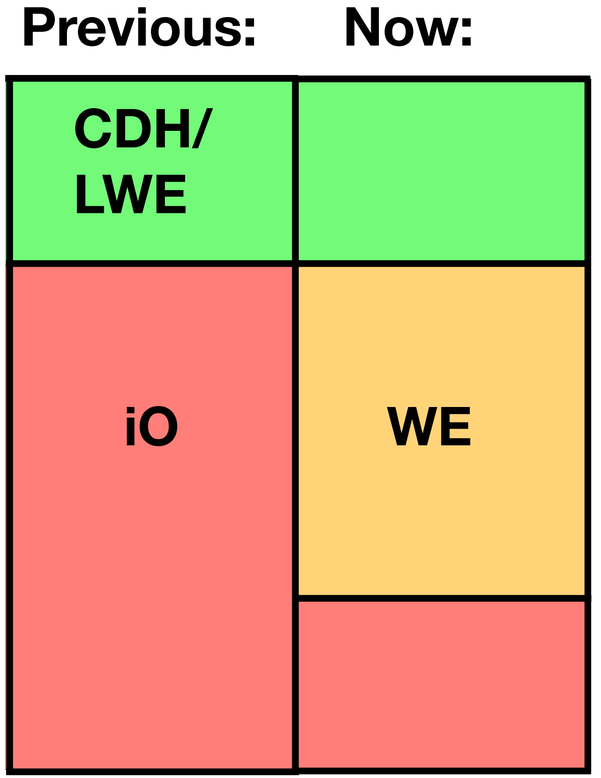
Detour into *Trustless* Cryptography

Functional Encryption: augment public-key encryption with fine-grained decryption capabilities [BW07,KSW08,BSW11,O'N10]

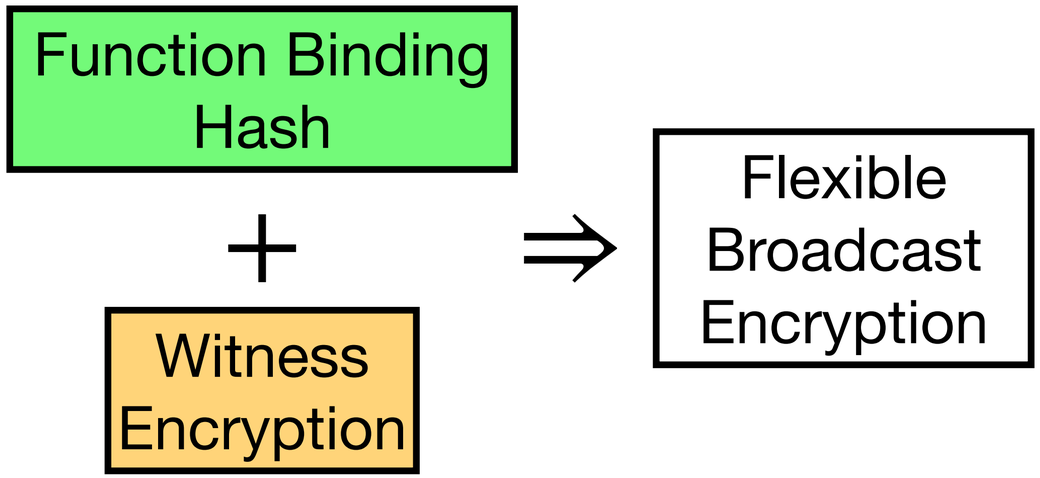
Limitation: secret keys are issued by a **central trusted authority**

Recently: removing trust from functional encryption

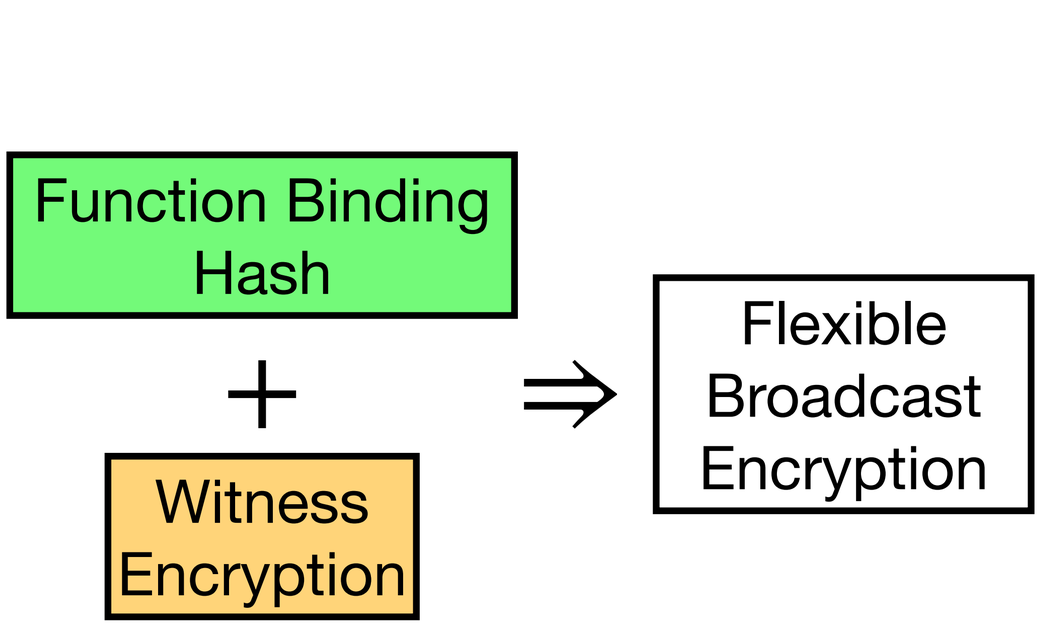
[GHMR18, GHMRS19,GV20]	Identity-Based Encryption	→ Registration-Based Encryption
[HLWW23]	Attribute-Based Encryption	→ Registered ABE
[BZ14]	Broadcast Encryption	→ Distributed/ Flexible Broadcast
[FFMMRV23, DP23]	Functional Encryption	→ Registered Functional Encryption



Plan for Today:
The Case of Flexible Broadcast

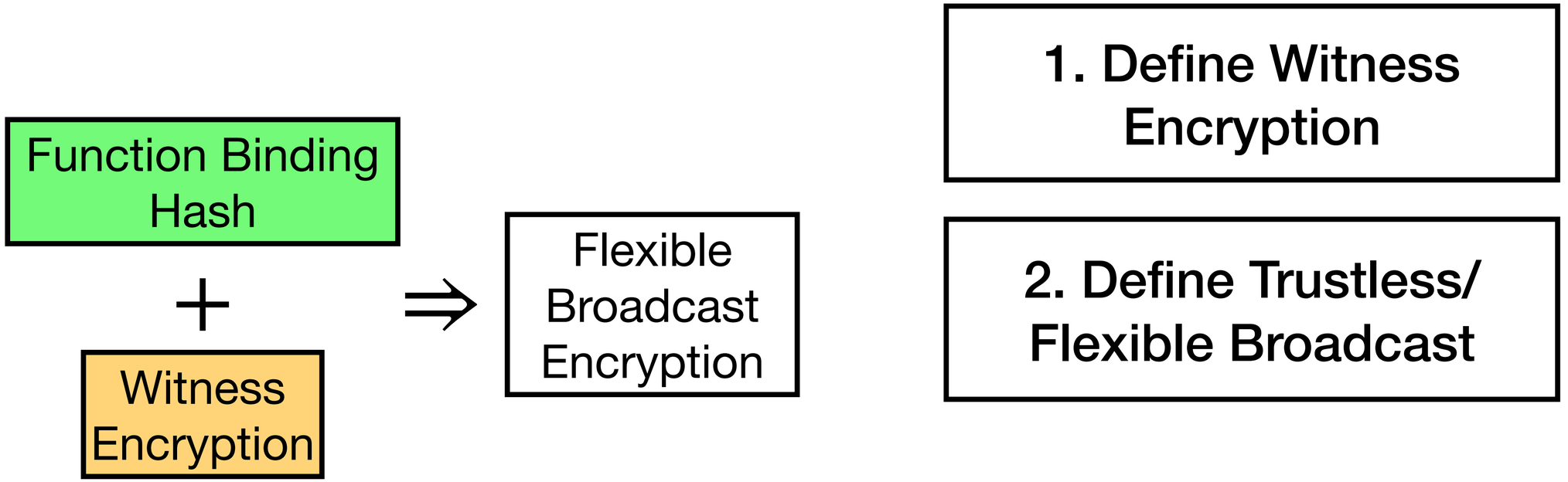


Plan for Today:
The Case of Flexible Broadcast

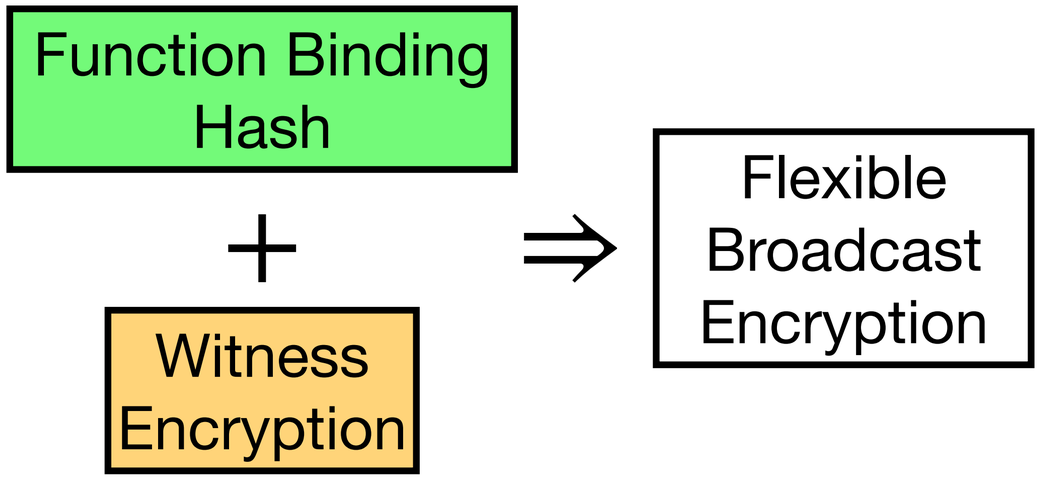


1. Define Witness Encryption

Plan for Today:
The Case of Flexible Broadcast



Plan for Today:
The Case of Flexible Broadcast



1. Define Witness Encryption

2. Define Trustless/ Flexible Broadcast

3. Show framework using Function Binding Hash

Witness Encryption

Witness Encryption

Syntax and Correctness:

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot)) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{ct}, w) \to \mathsf{msg}$ if $R_L(x, w) = 1$

Witness Encryption

Syntax and Correctness:

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot)) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{ct}, w) \to \mathsf{msg}$ if $R_L(x, w) = 1$

Security: If $x \notin L$,

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

$\approx \mathsf{Enc}(1^\lambda, 0, R_L(x, \cdot))$

Witness Encryption

Syntax and Correctness:

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot)) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{ct}, w) \to \mathsf{msg}$ if $R_L(x, w) = 1$

Security: If $x \notin L$,

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

$\approx \mathsf{Enc}(1^\lambda, 0, R_L(x, \cdot))$

Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$

Witness Encryption

Syntax and Correctness:

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot)) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{ct}, w) \to \mathsf{msg}$ if $R_L(x, w) = 1$

Security: If $x \notin L$,

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

$\approx \mathsf{Enc}(1^\lambda, 0, R_L(x, \cdot))$

Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$

Generalization of PKE

public key: x

secret key: w s.t. $R_L(x, w) = 1$

Witness Encryption

Syntax and Correctness:

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot)) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{ct}, w) \to \mathsf{msg}$ if $R_L(x, w) = 1$

Security: If $x \notin L$,

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

$\approx \mathsf{Enc}(1^\lambda, 0, R_L(x, \cdot))$

Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$

Generalization of PKE

public key: x

secret key: w s.t. $R_L(x, w) = 1$

Weak Form of Obfuscation

$\mathsf{ct} := \mathsf{ct}[R_L, x, \mathsf{msg}]$ is a program s.t.

$$\mathsf{ct}(w) \to \begin{cases} \mathsf{msg} & \text{if } R_L(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$$

Witness Encryption

Syntax and Correctness:

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot)) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{ct}, w) \to \mathsf{msg}$ if $R_L(x, w) = 1$

Security: If $x \notin L$,

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

$\approx \mathsf{Enc}(1^\lambda, 0, R_L(x, \cdot))$

Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$

Generalization of PKE

public key: x

secret key: w s.t. $R_L(x, w) = 1$

Weak Form of Obfuscation

$\mathsf{ct} := \mathsf{ct}[R_L, x, \mathsf{msg}]$ is a program s.t.

$$\mathsf{ct}(w) \to \begin{cases} \mathsf{msg} & \text{if } R_L(x, w) = 1 \\ \bot & \text{otherwise} \end{cases}$$

msg is comp. hidden if $x \notin L$

Witness Encryption

Syntax and Correctness:

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot)) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{ct}, w) \to \mathsf{msg}$ if $R_L(x, w) = 1$

Security: If $x \notin L$,

$\mathsf{Enc}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

$\approx \mathsf{Enc}(1^\lambda, 0, R_L(x, \cdot))$

Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|)$

Only hiding msg—no computation

Generalization of PKE

public key: x

secret key: w s.t. $R_L(x, w) = 1$

Weak Form of Obfuscation

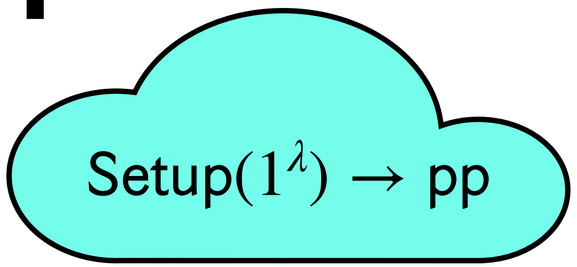
$\mathsf{ct} := \mathsf{ct}[R_L, x, \mathsf{msg}]$ is a program s.t.

$$\mathsf{ct}(w) \to \begin{cases} \mathsf{msg} & \text{if } R_L(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$$

msg is comp. hidden if $x \notin L$

Flexible Broadcast Encryption

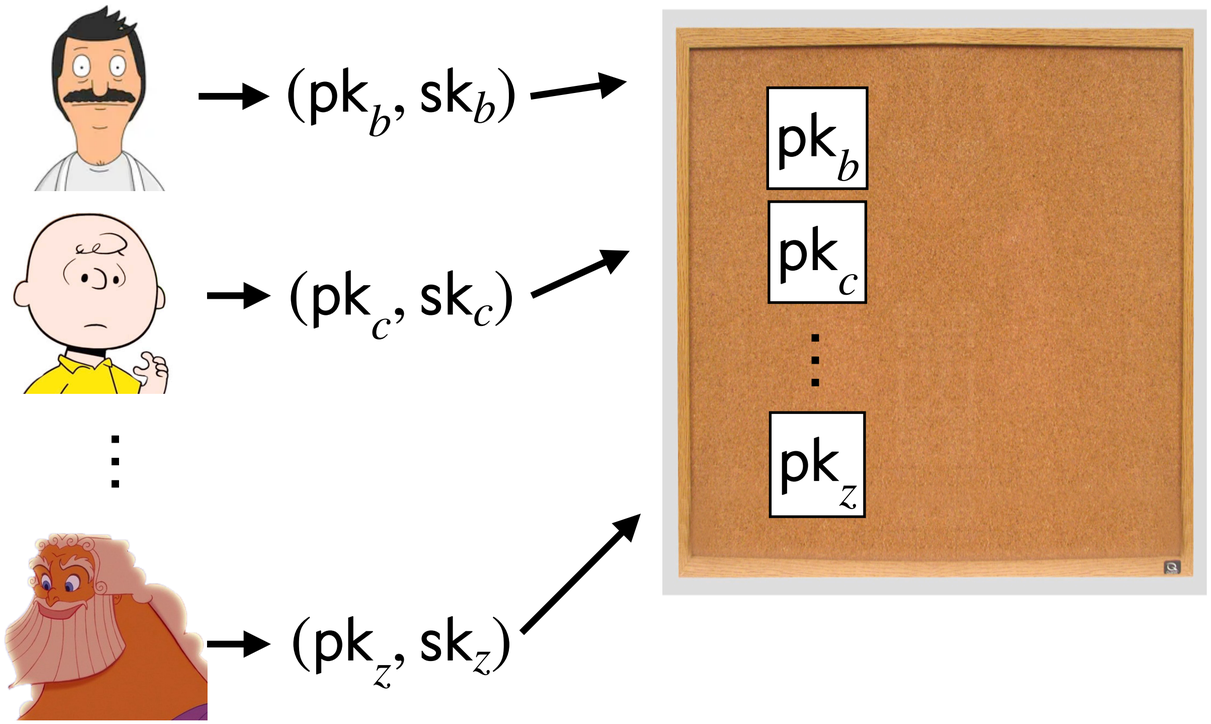
Flexible Broadcast Encryption



Flexible Broadcast Encryption

Setup(1^λ) → pp

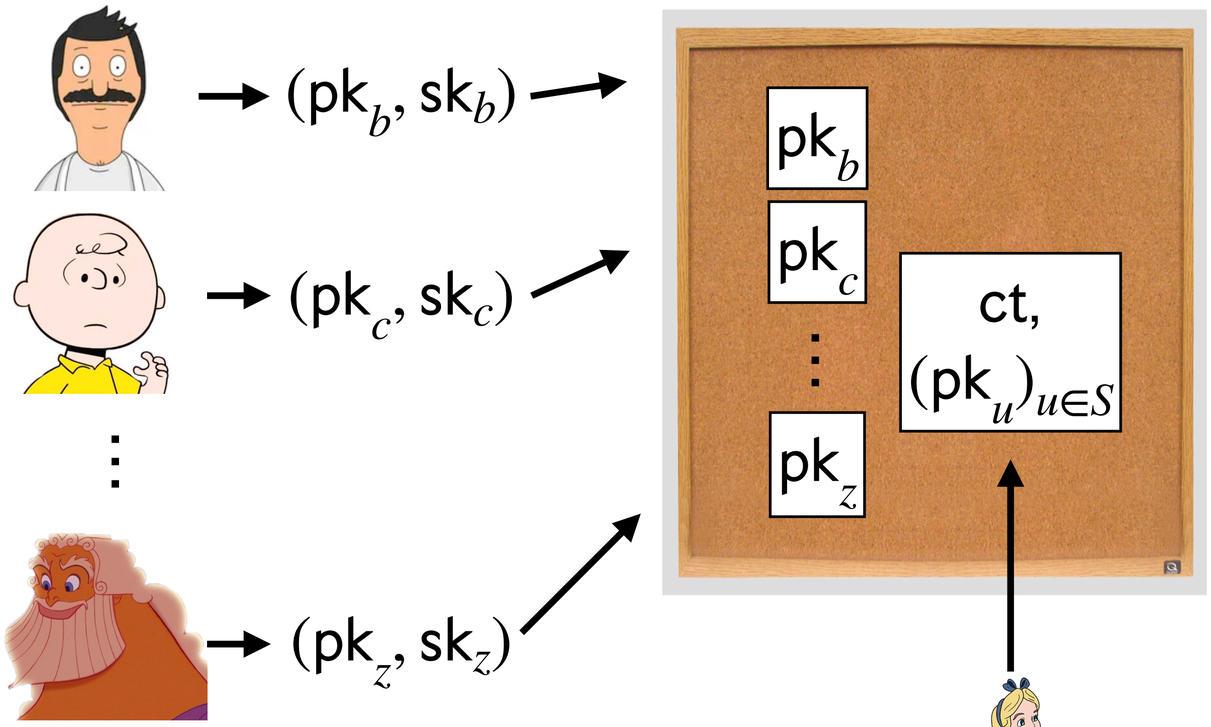
KeyGen(pp)



Flexible Broadcast Encryption

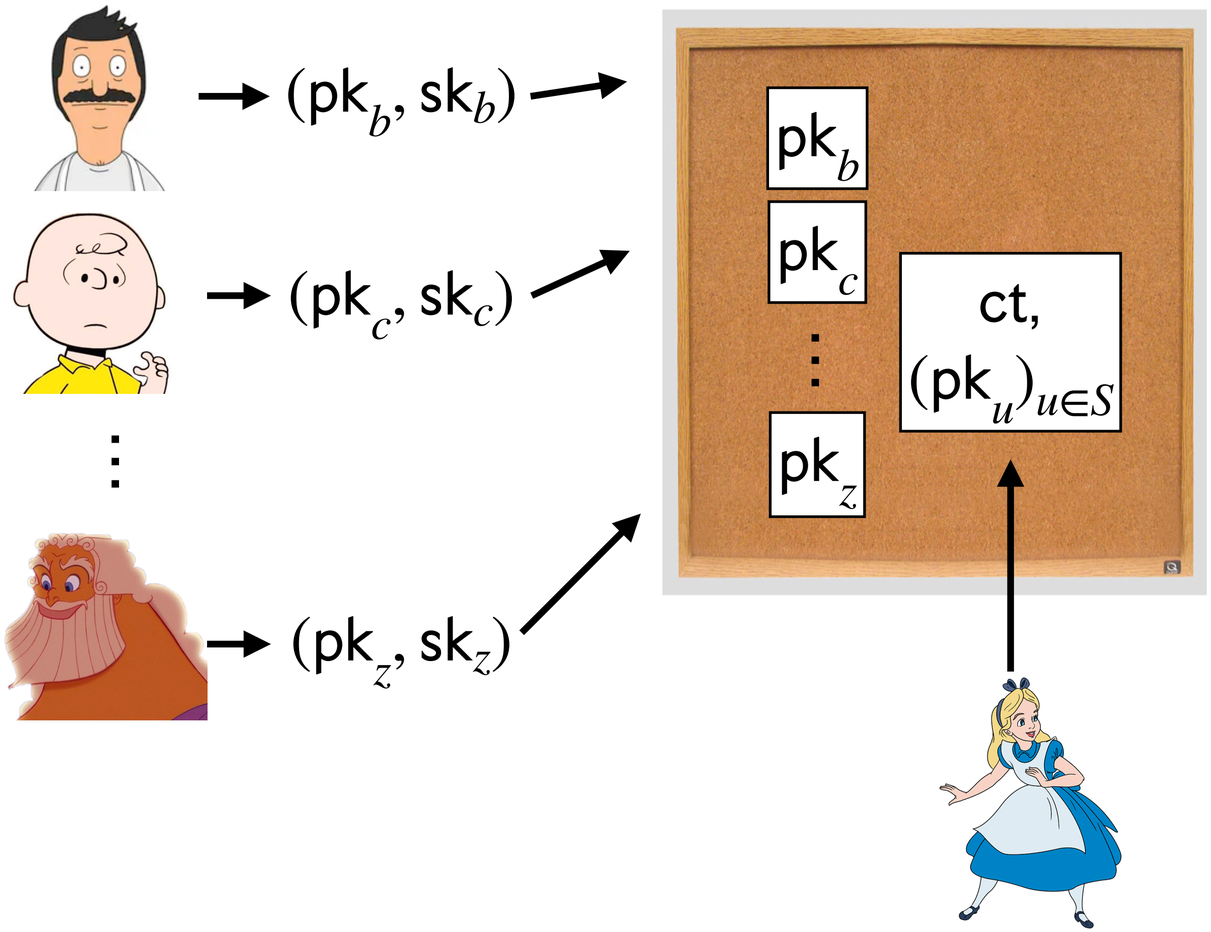
Setup(1^λ) → pp

KeyGen(pp)



Flexible Broadcast Encryption

KeyGen(pp)



$\text{Setup}(1^\lambda) \rightarrow \text{pp}$

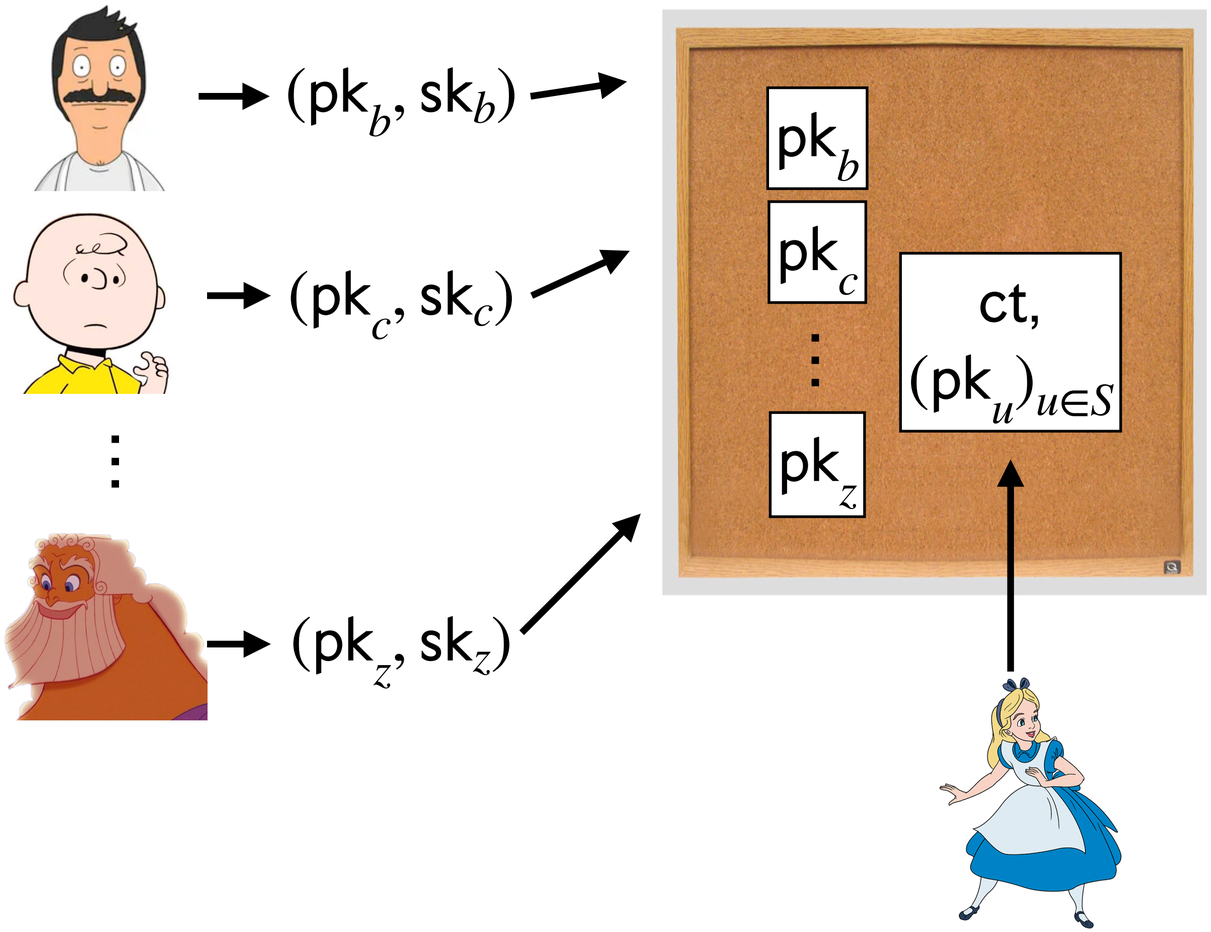
Syntax and Correctness:

$\text{Enc}(\text{pp}, \text{msg}, (\text{pk}_u)_{u \in S}) \rightarrow \text{ct}$

$\text{Dec}(\text{pp}, \text{ct}, \text{sk}_v, (\text{pk}_u)_{u \in S}) \rightarrow \text{msg}$ if $v \in S$

Flexible Broadcast Encryption

KeyGen(pp)



$\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$

Syntax and Correctness:

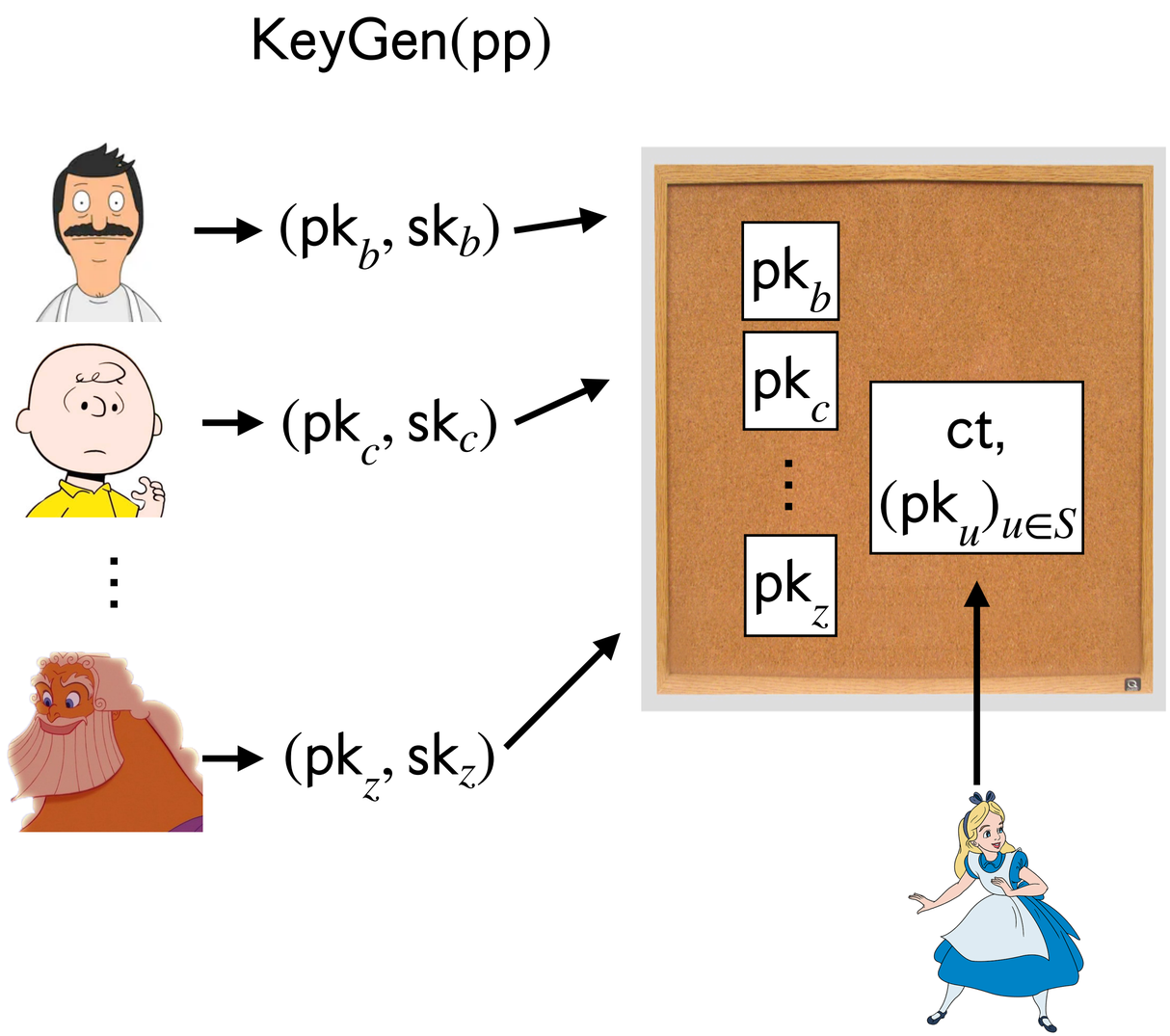
$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}_v, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{msg}$ if $v \in S$

≈**Security:** Without sk_v for $v \in S$,

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$
$\approx \mathsf{Enc}(\mathsf{pp}, 0, (\mathsf{pk}_u)_{u \in S})$

Flexible Broadcast Encryption



$\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$

Syntax and Correctness:

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}_v, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{msg}$ if $v \in S$

≈**Security:** Without sk_v for $v \in S$,

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$
$\approx \mathsf{Enc}(\mathsf{pp}, 0, (\mathsf{pk}_u)_{u \in S})$

Efficiency:

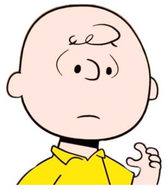
$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, \log |S|)$

Flexible Broadcast Encryption

Trustless: users generate own keys

KeyGen(pp)

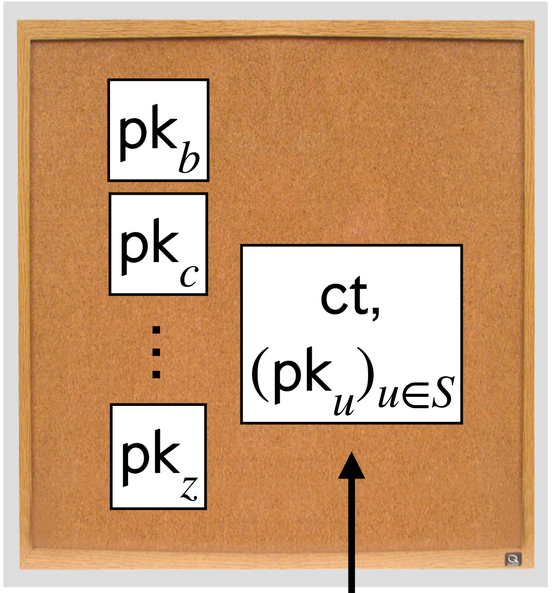
$\rightarrow (\mathsf{pk}_b, \mathsf{sk}_b) \rightarrow$

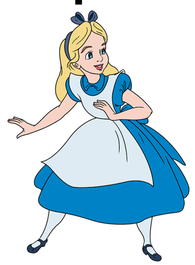


$\rightarrow (\mathsf{pk}_c, \mathsf{sk}_c) \rightarrow$

$\vdots$

$\rightarrow (\mathsf{pk}_z, \mathsf{sk}_z) \rightarrow$





Setup$(1^\lambda) \rightarrow$ pp

Syntax and Correctness:

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S}) \rightarrow \mathsf{ct}$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}_v, (\mathsf{pk}_u)_{u \in S}) \rightarrow \mathsf{msg}$ if $v \in S$

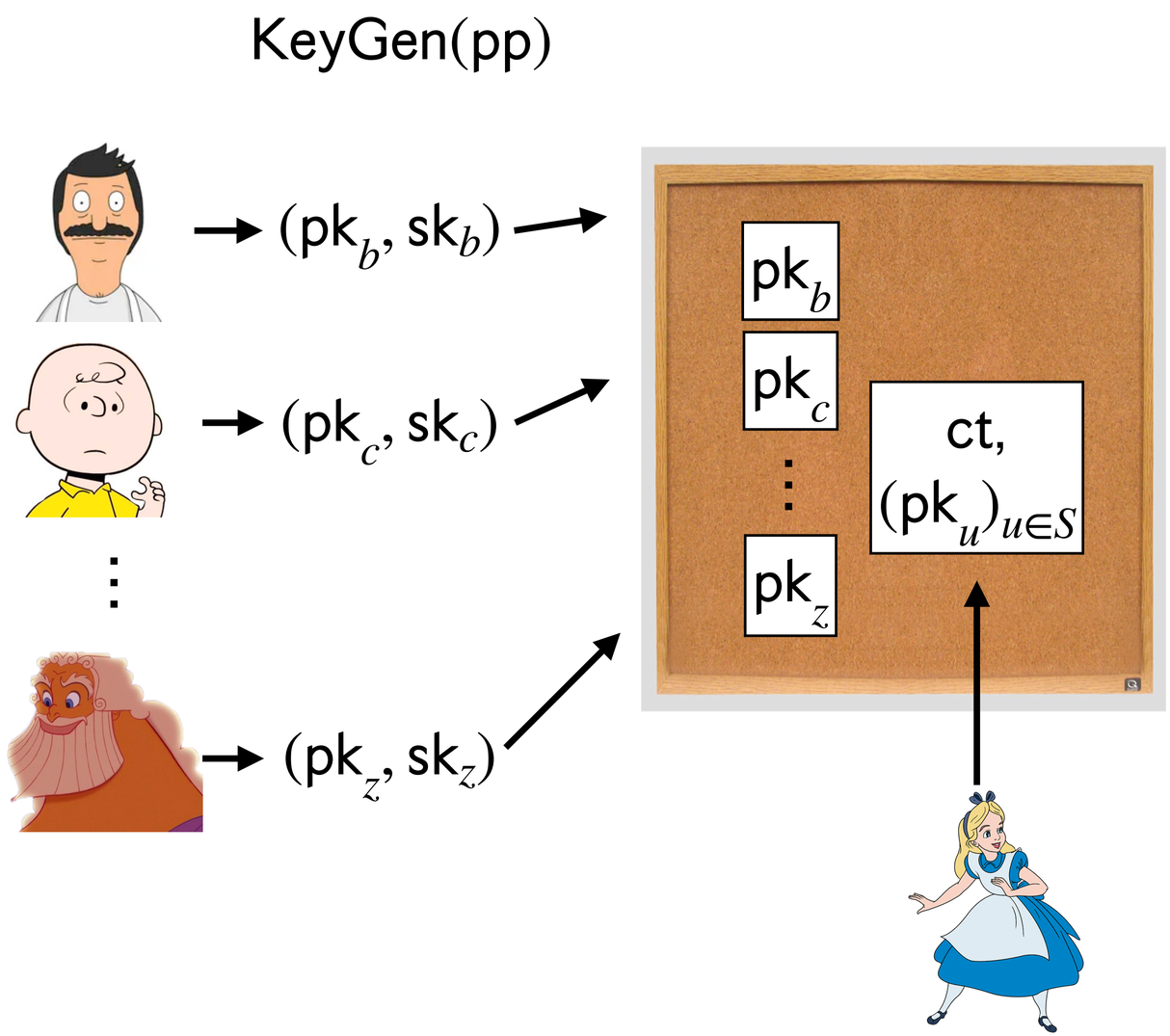
≈**Security:** Without sk_v for $v \in S$,

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$
$\approx \mathsf{Enc}(\mathsf{pp}, 0, (\mathsf{pk}_u)_{u \in S})$

Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, \log |S|)$

Flexible Broadcast Encryption



$\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$

Syntax and Correctness:

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}_v, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{msg}$ if $v \in S$

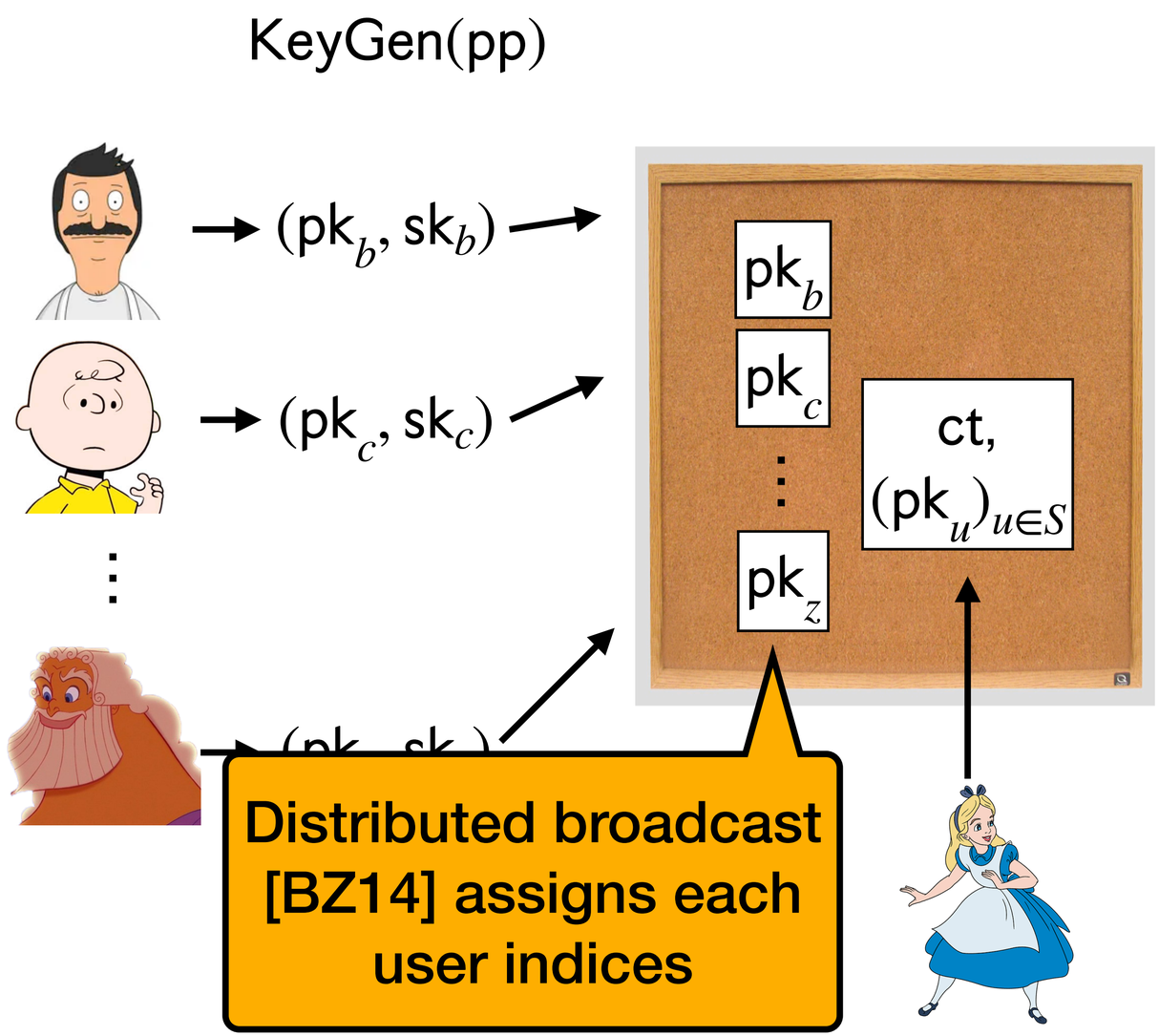
≈**Security:** Without sk_v for $v \in S$,

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$
$\approx \mathsf{Enc}(\mathsf{pp}, 0, (\mathsf{pk}_u)_{u \in S})$

Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, \log |S|)$

Flexible Broadcast Encryption



$\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$

Syntax and Correctness:

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}_v, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{msg}$ if $v \in S$

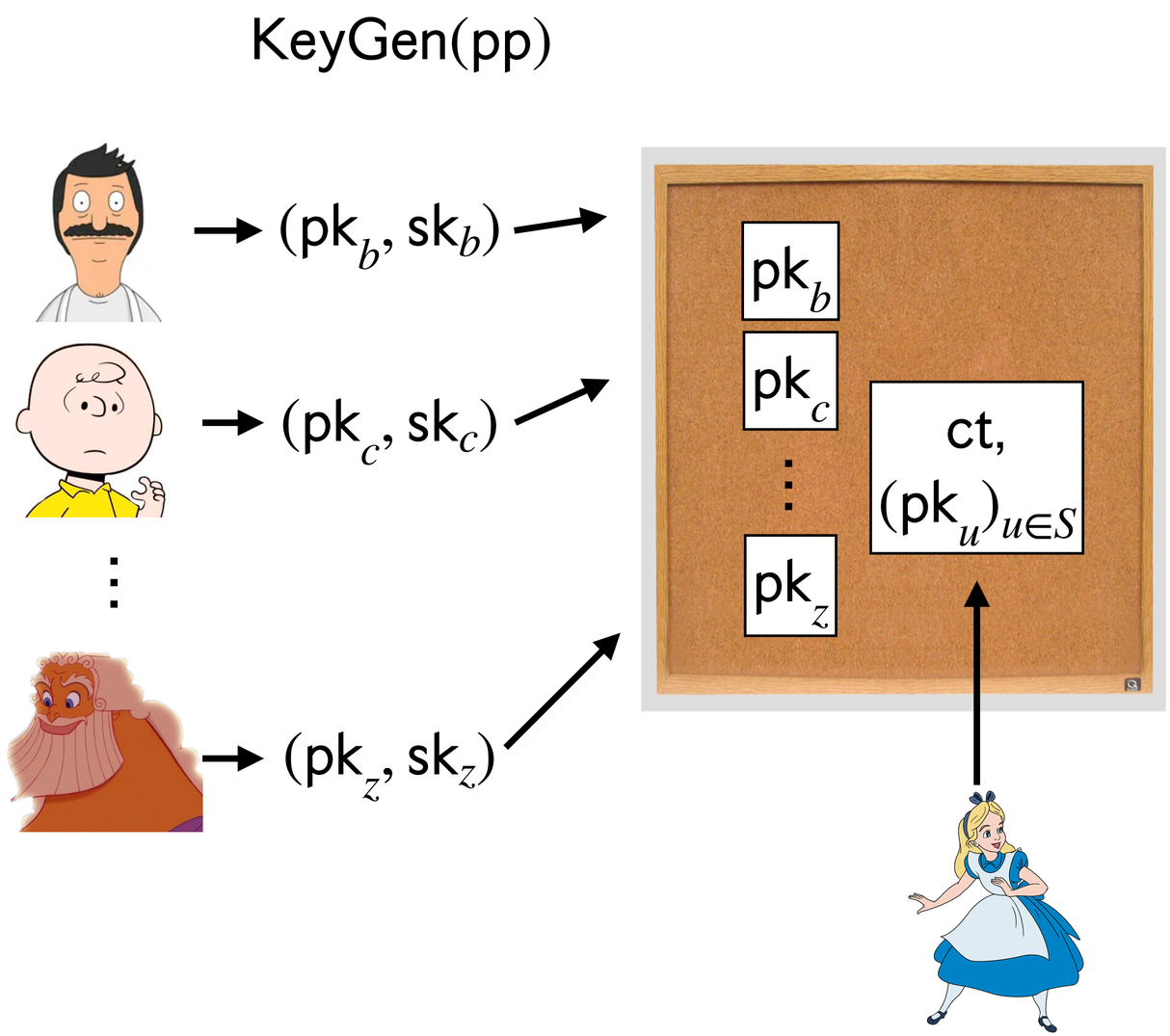
≈**Security:** Without sk_v for $v \in S$,

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$
$\approx \mathsf{Enc}(\mathsf{pp}, 0, (\mathsf{pk}_u)_{u \in S})$

Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, \log |S|)$

Flexible Broadcast Encryption



$\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$

Syntax and Correctness:

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{ct}$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{ct}, \mathsf{sk}_v, (\mathsf{pk}_u)_{u \in S}) \to \mathsf{msg}$ if $v \in S$

≈**Security:** Without sk_v for $v \in S$,

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$
$\approx \mathsf{Enc}(\mathsf{pp}, 0, (\mathsf{pk}_u)_{u \in S})$

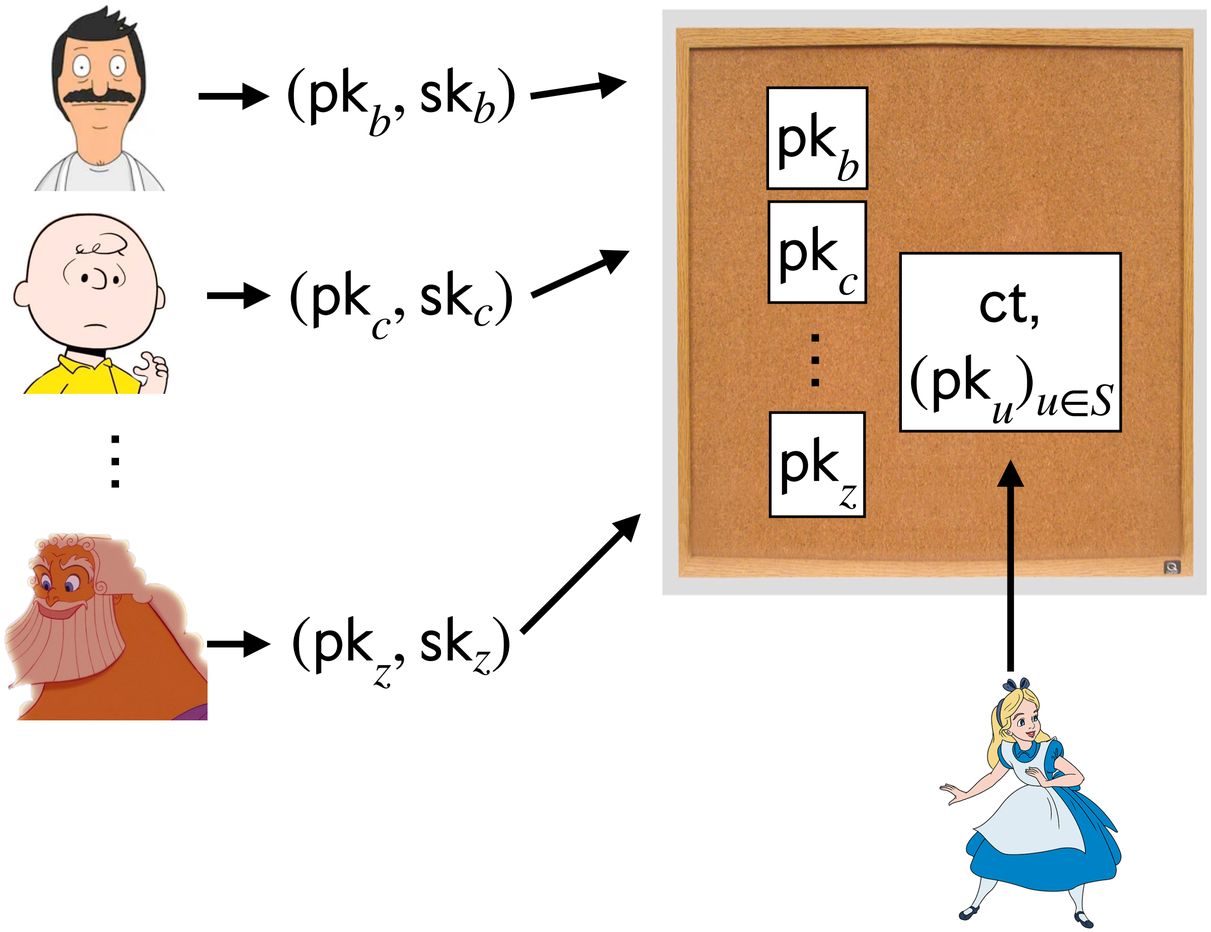
Efficiency:

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, \log |S|)$

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

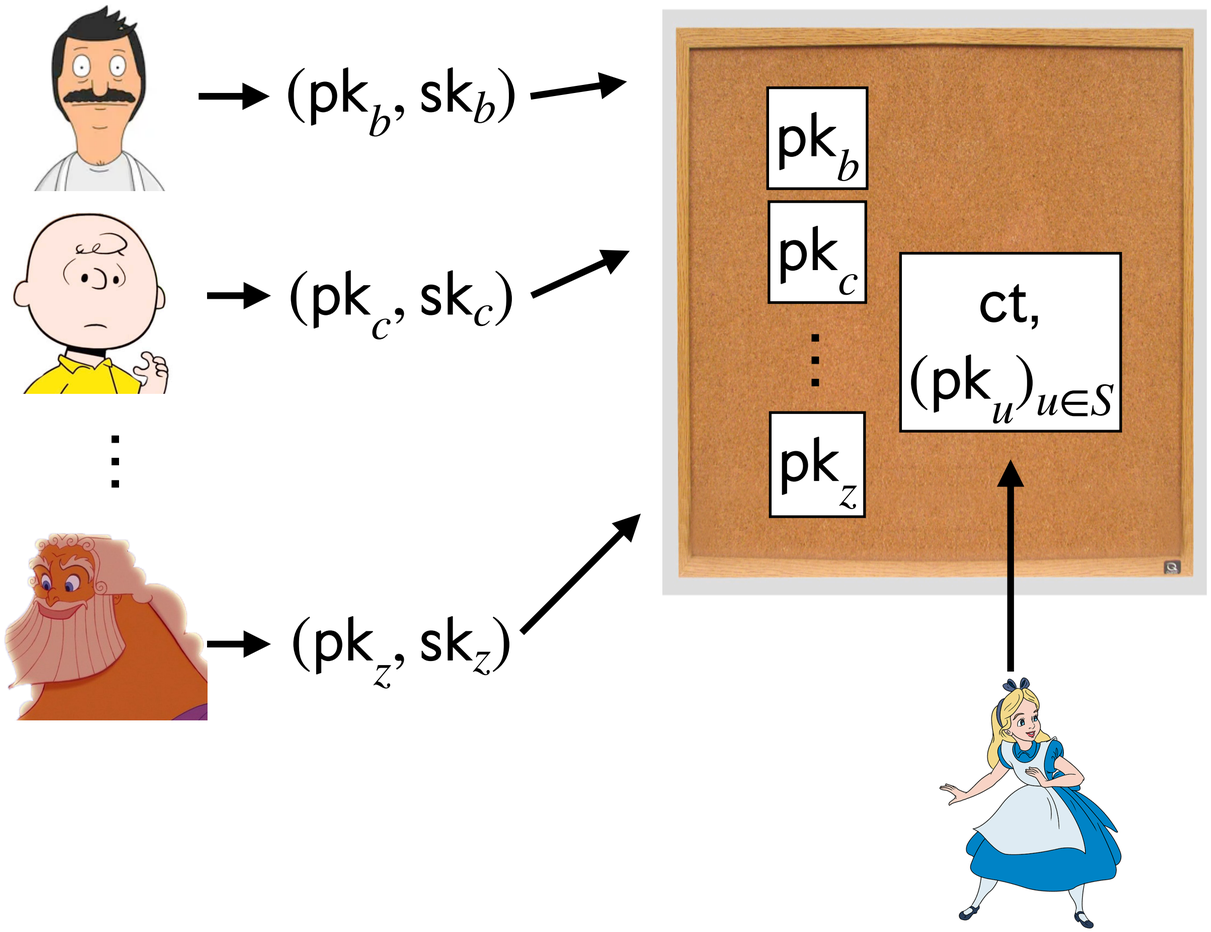
$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

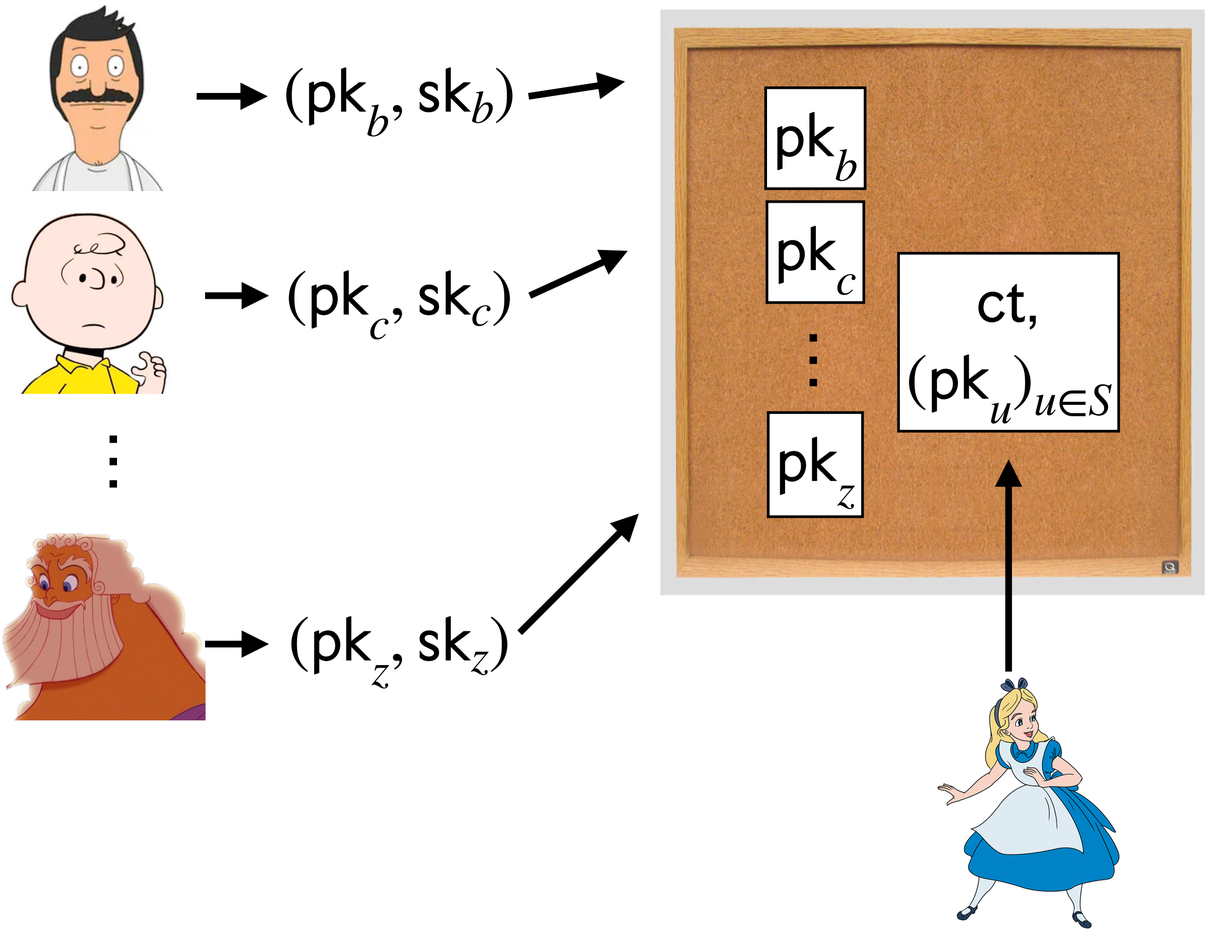
where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

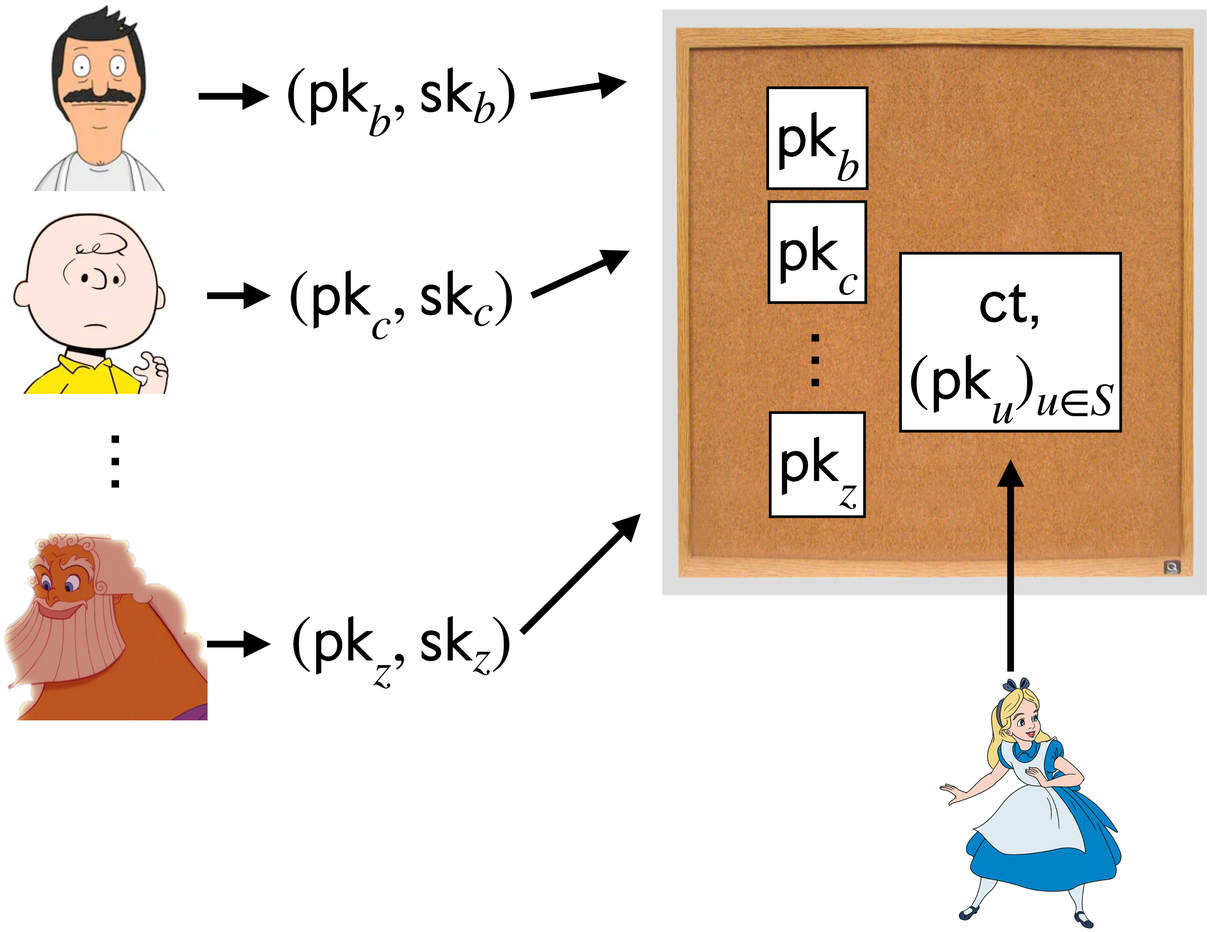
$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

Correct? ✓

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

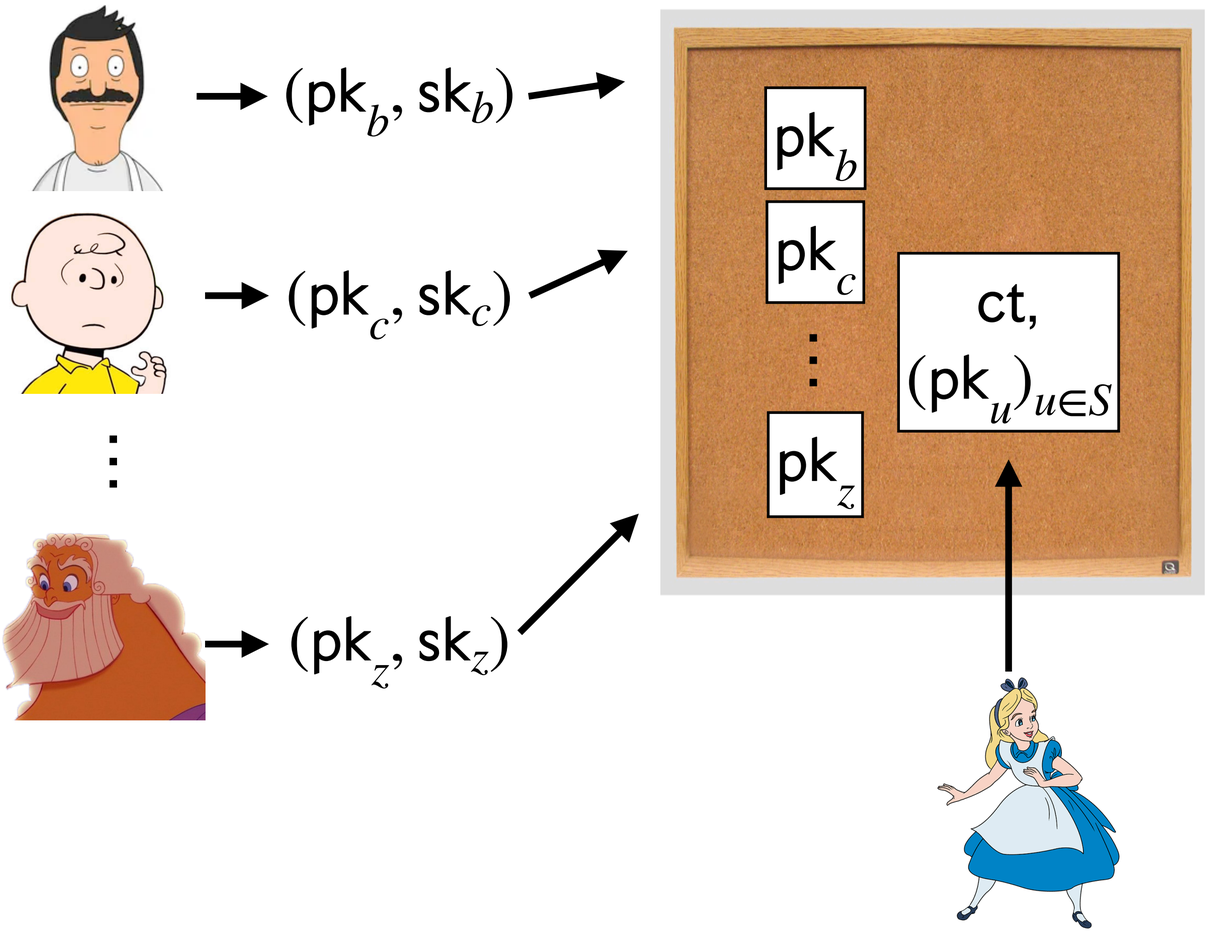
$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

Correct? ✓ **Secure?**

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

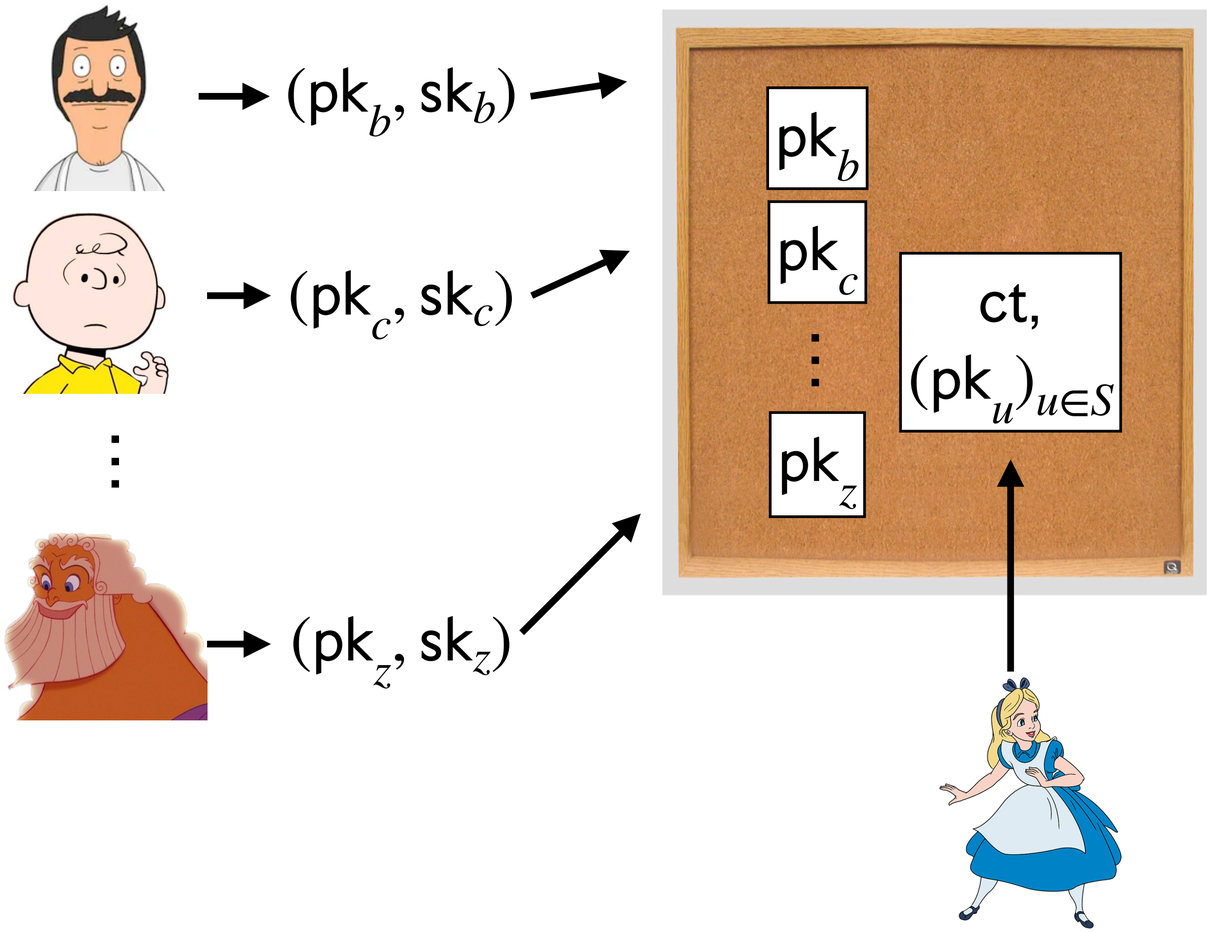
Correct? ✓ **Secure?**

Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

Correct? ✓ **Secure?**

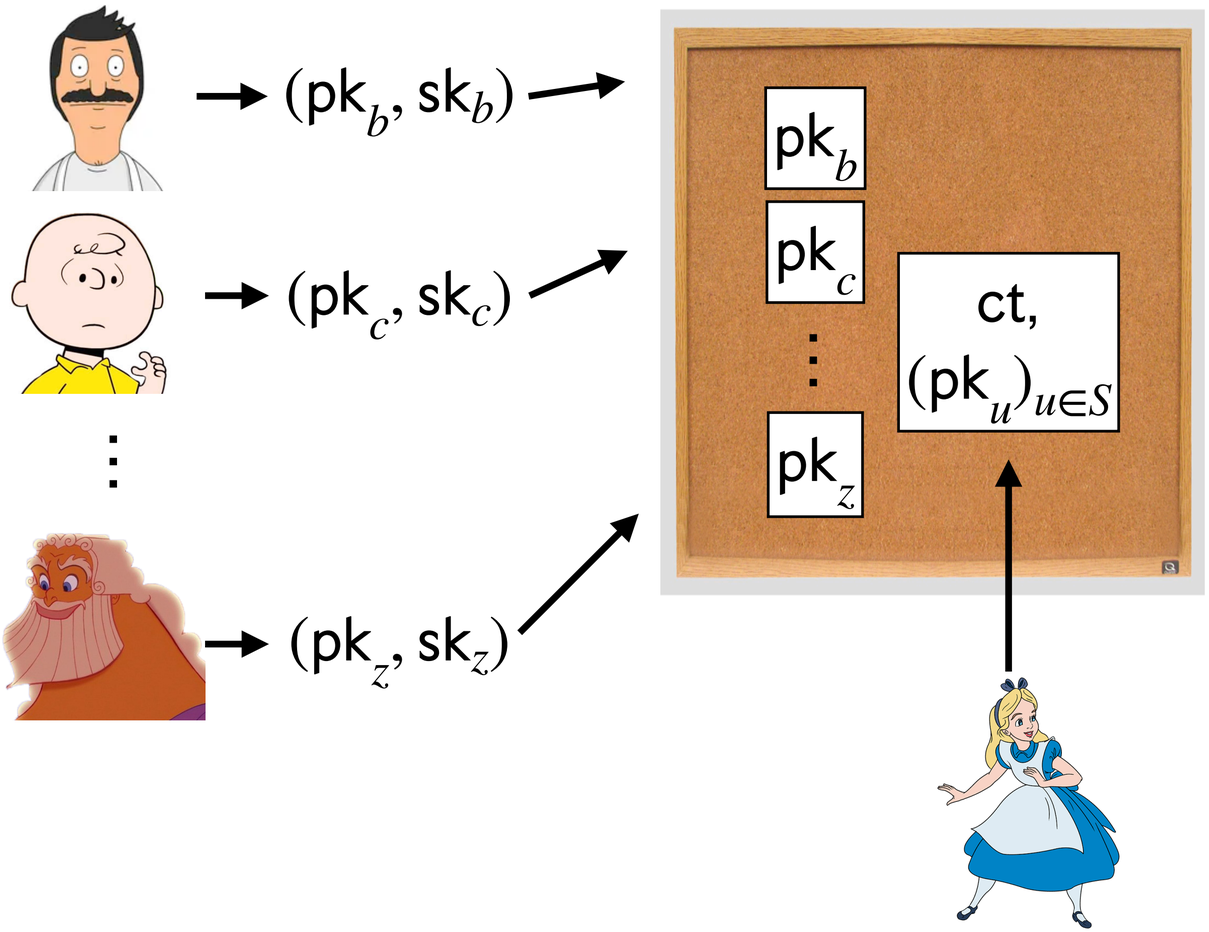
Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

Hybrid H_2: switch $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, 0, R_L(x, \cdot))$

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

Correct? ✓ **Secure?**

Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

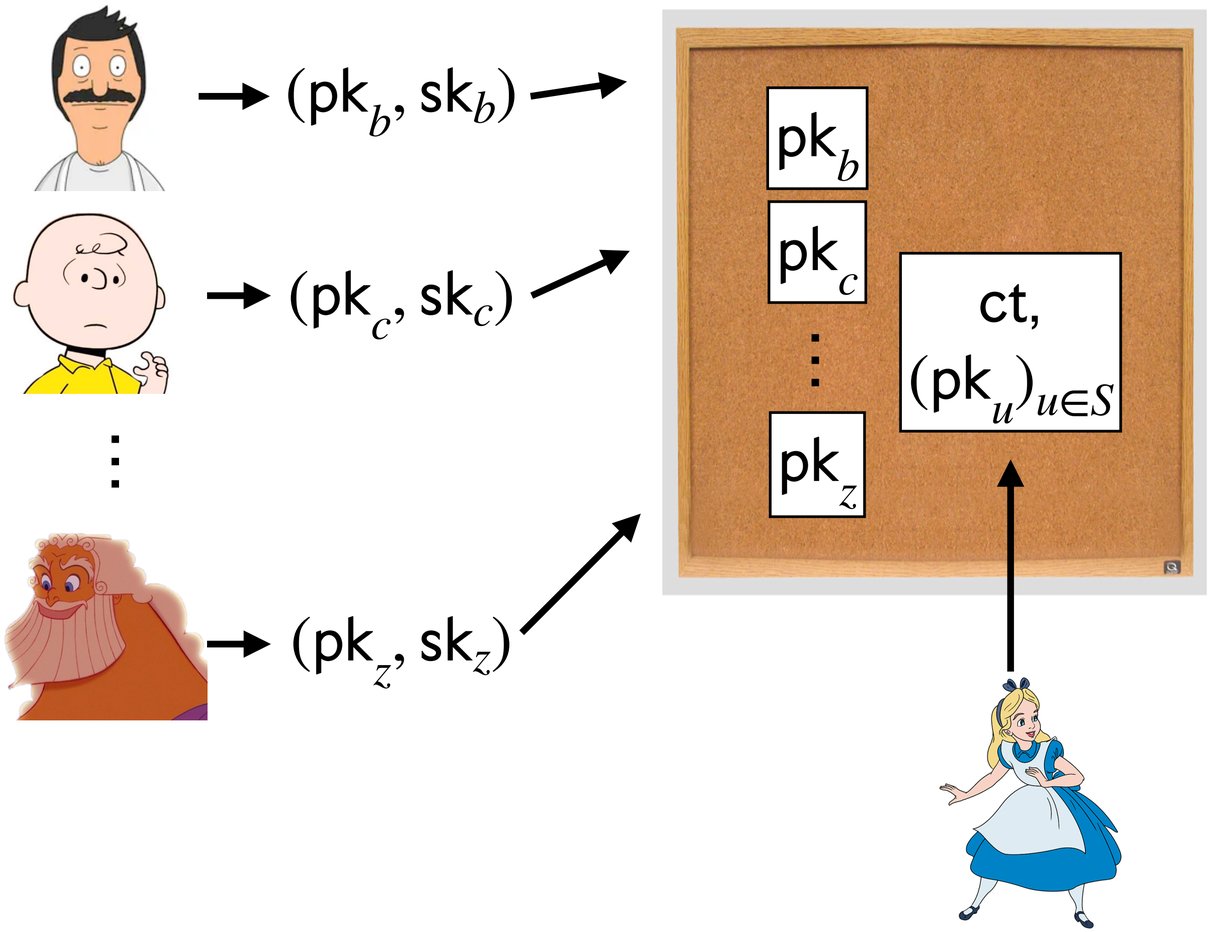
Hybrid H_2: switch $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, 0, R_L(x, \cdot))$

Relies on $x \notin L$ in H_1.

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

Correct? ✓ **Secure?** ✓

Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

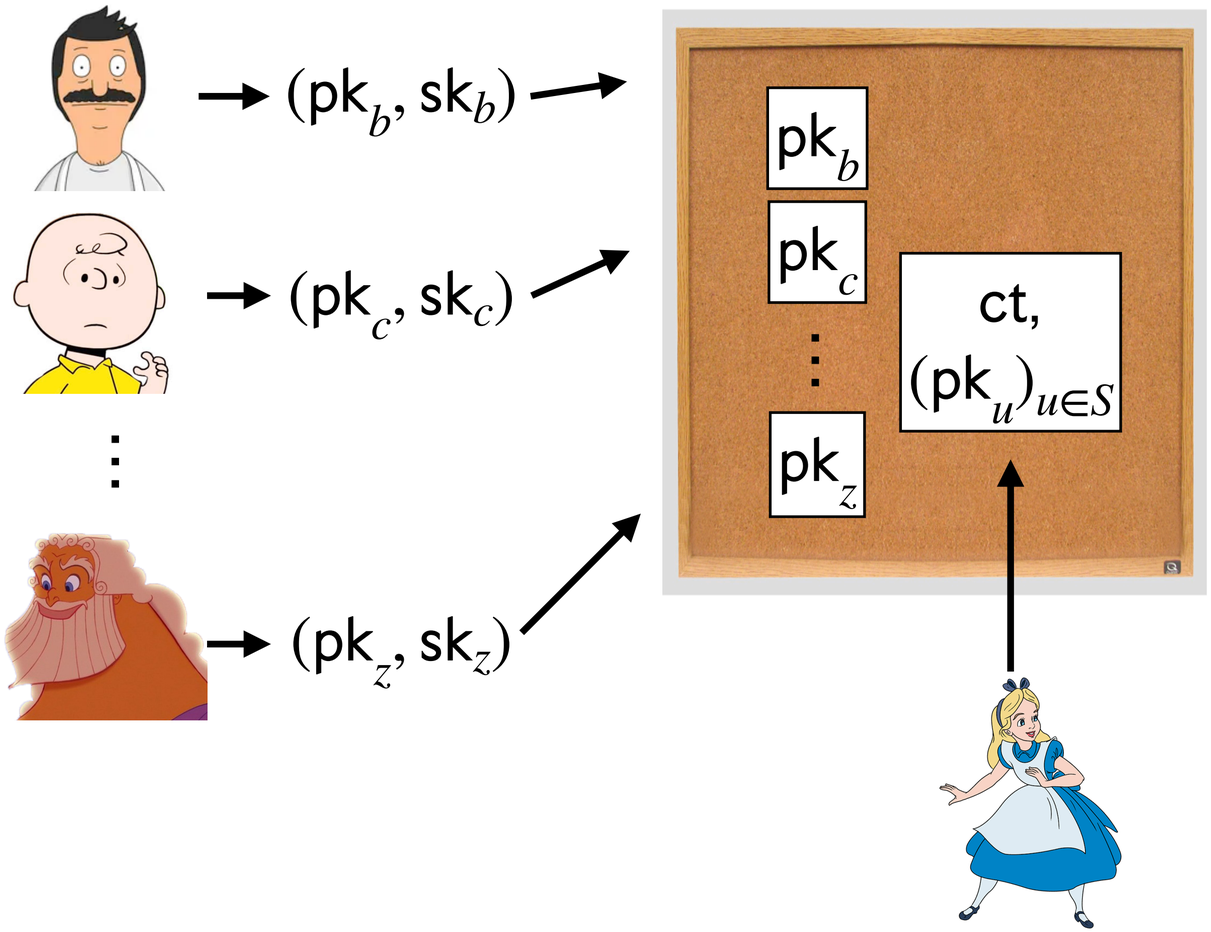
Hybrid H_2: switch $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, 0, R_L(x, \cdot))$

Relies on $x \notin L$ in H_1.

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

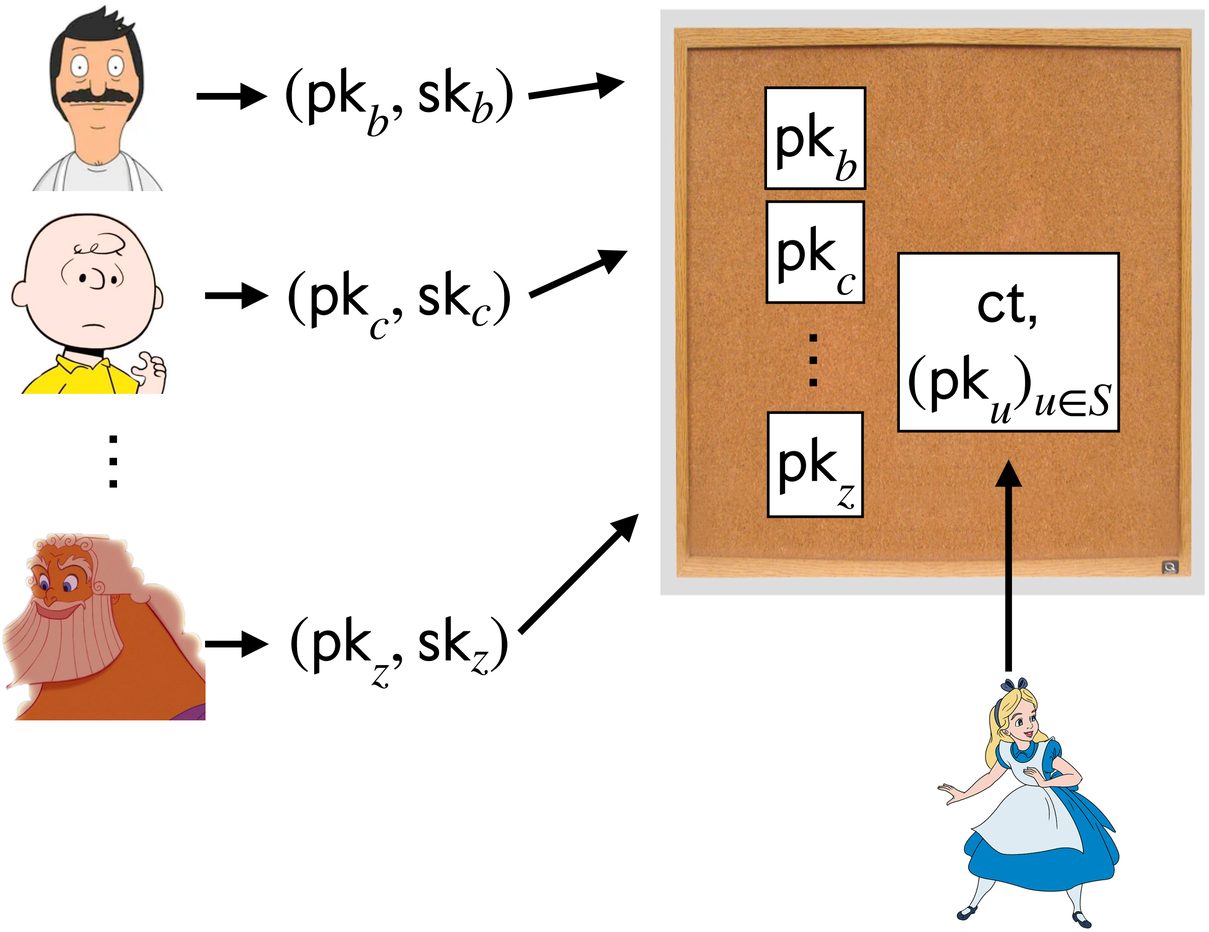
$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

Correct? ✓ **Secure?** ✓ **Efficient?**

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

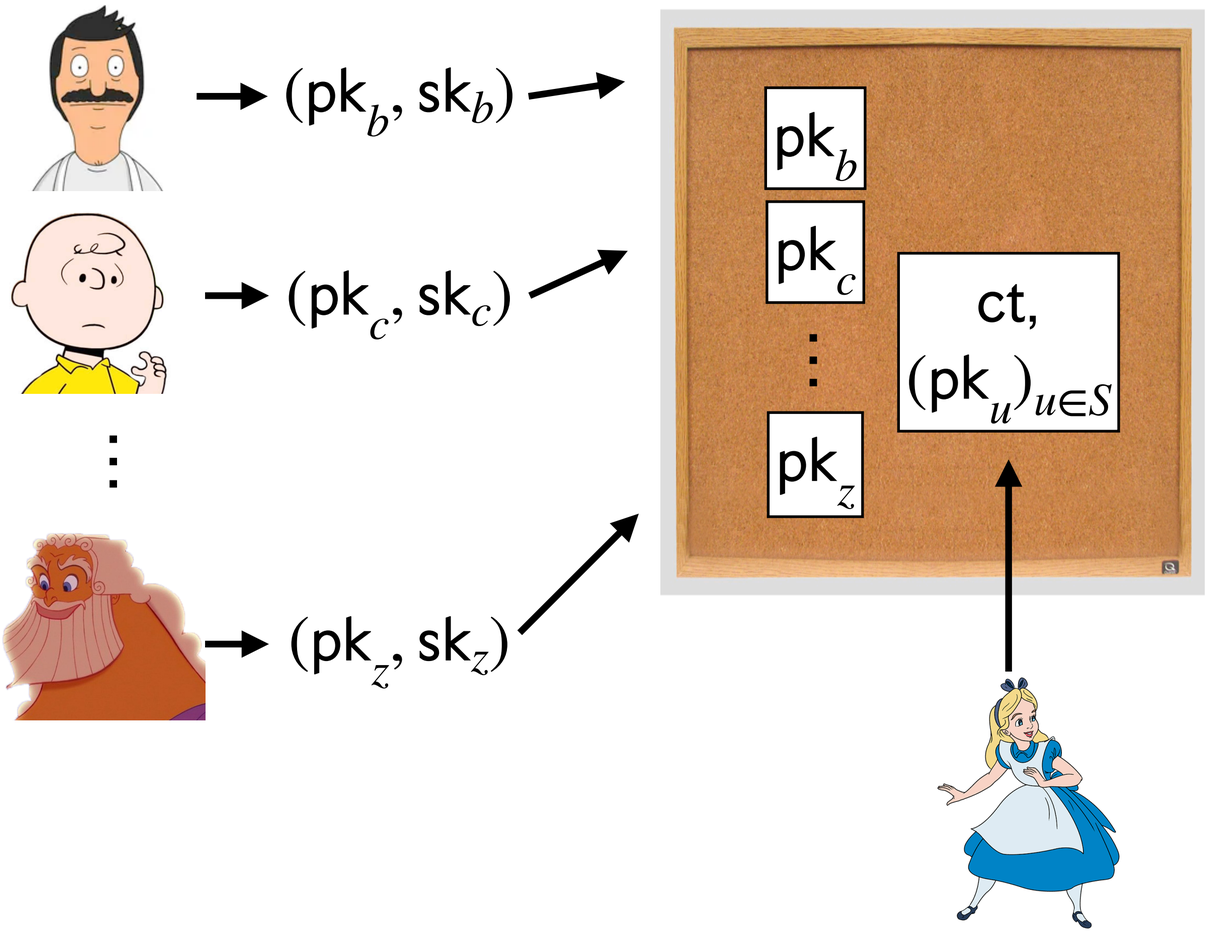
Correct? ✓ **Secure?** ✓ **Efficient?**

$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

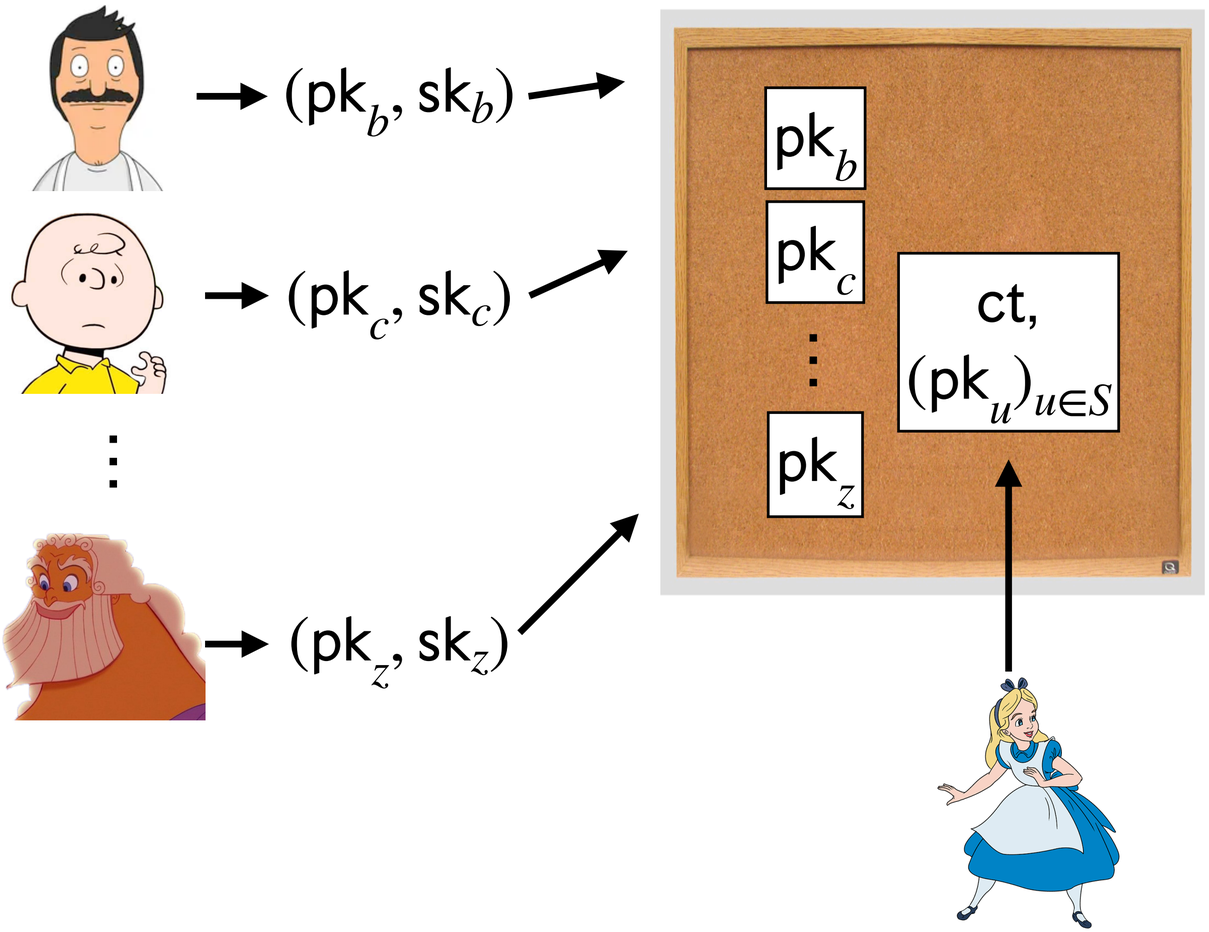
Correct? ✓ **Secure?** ✓ **Efficient?**

$$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$$
$$= \mathsf{poly}(\lambda, |\mathsf{msg}|, |S|)$$

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

Correct? ✓ **Secure?** ✓ **Efficient?**

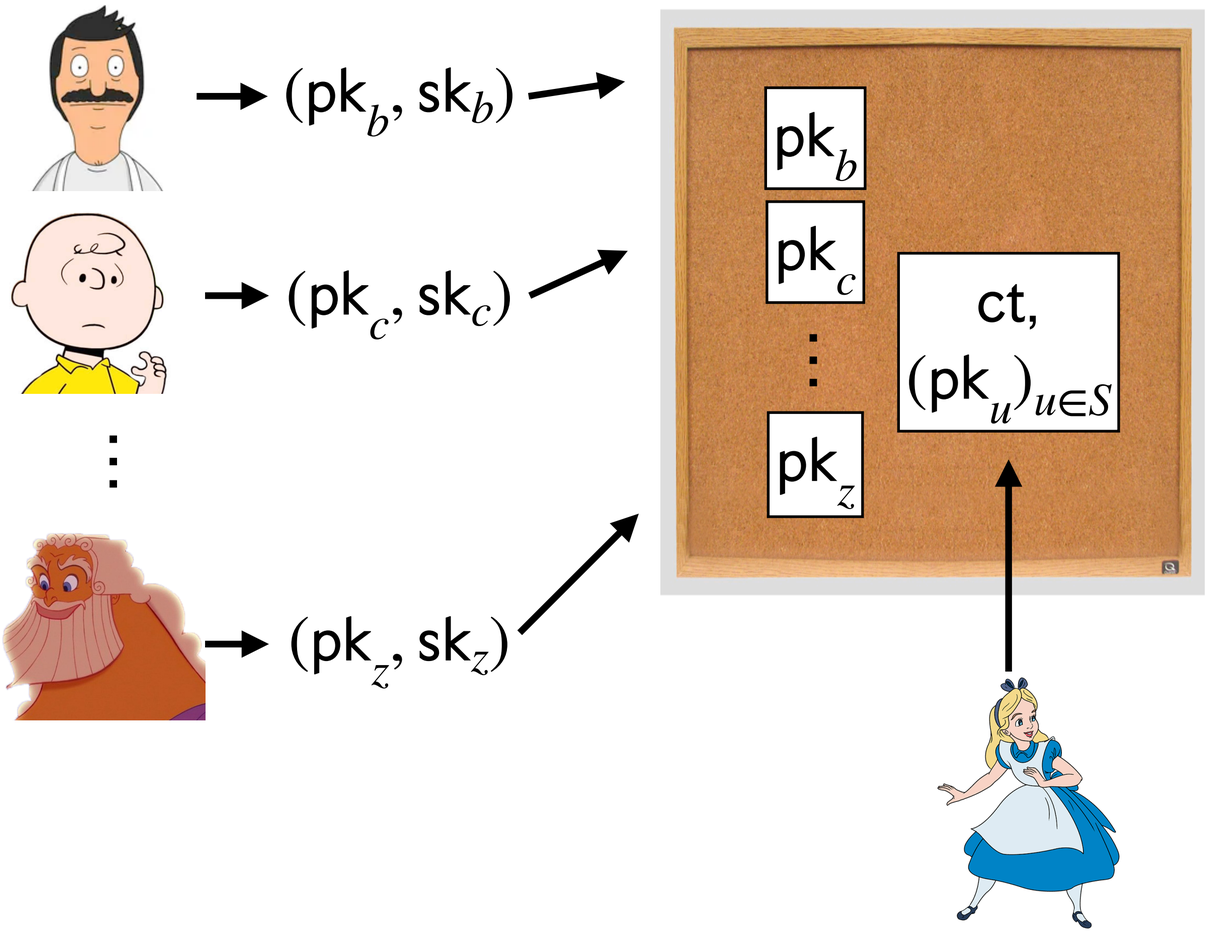
$$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$$
$$= \mathsf{poly}(\lambda, |\mathsf{msg}|, |S|)$$

Need log |S|

Construction: Attempt 1

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = \mathsf{pk}_{\mathsf{PKE}}$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = (\mathsf{pk}_u)_{u \in S}$, $w = \mathsf{sk}_v$

$R_L(x, w) \to (\exists v \in S, \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v))$

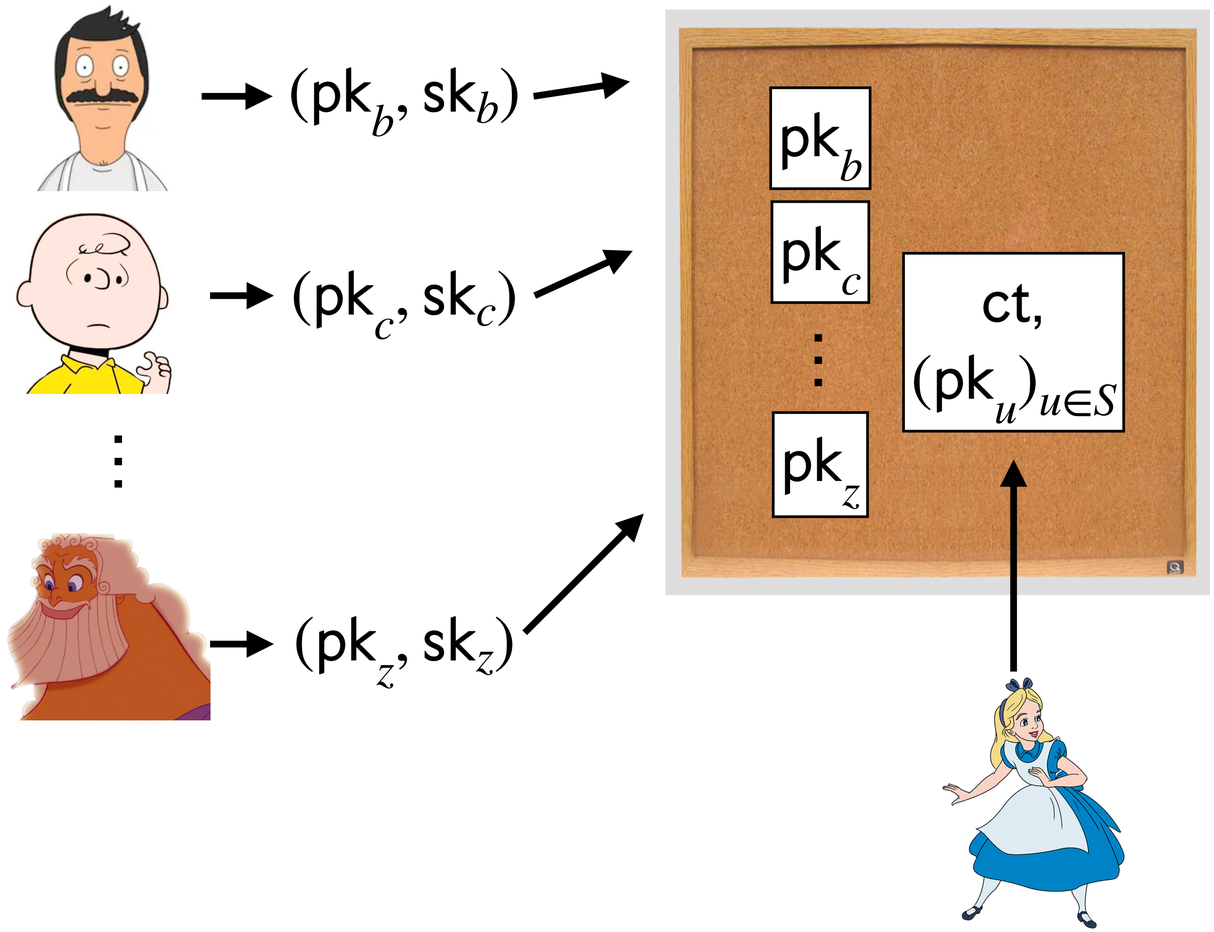
Correct? ✓ **Secure?** ✓ **Efficient?** ✗

$$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$$
$$= \mathsf{poly}(\lambda, |\mathsf{msg}|, |S|)$$

Need log |S|

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$



$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

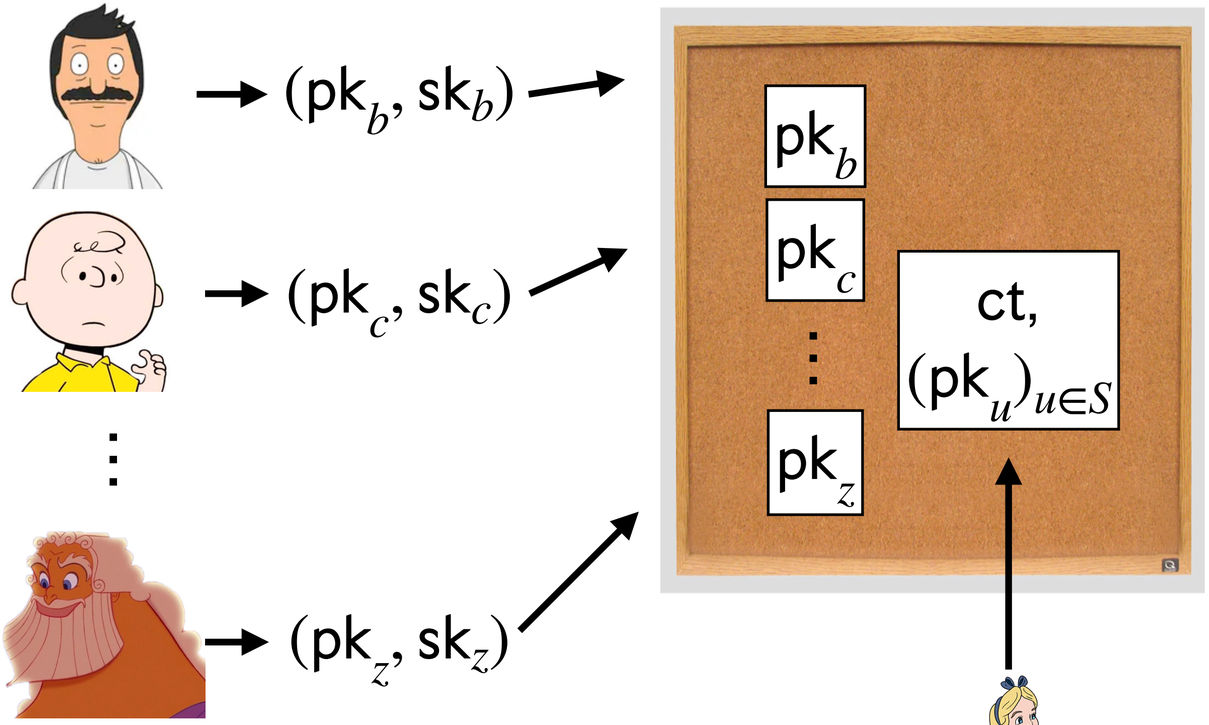
Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$



Merkle Tree—hash function with local opening

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

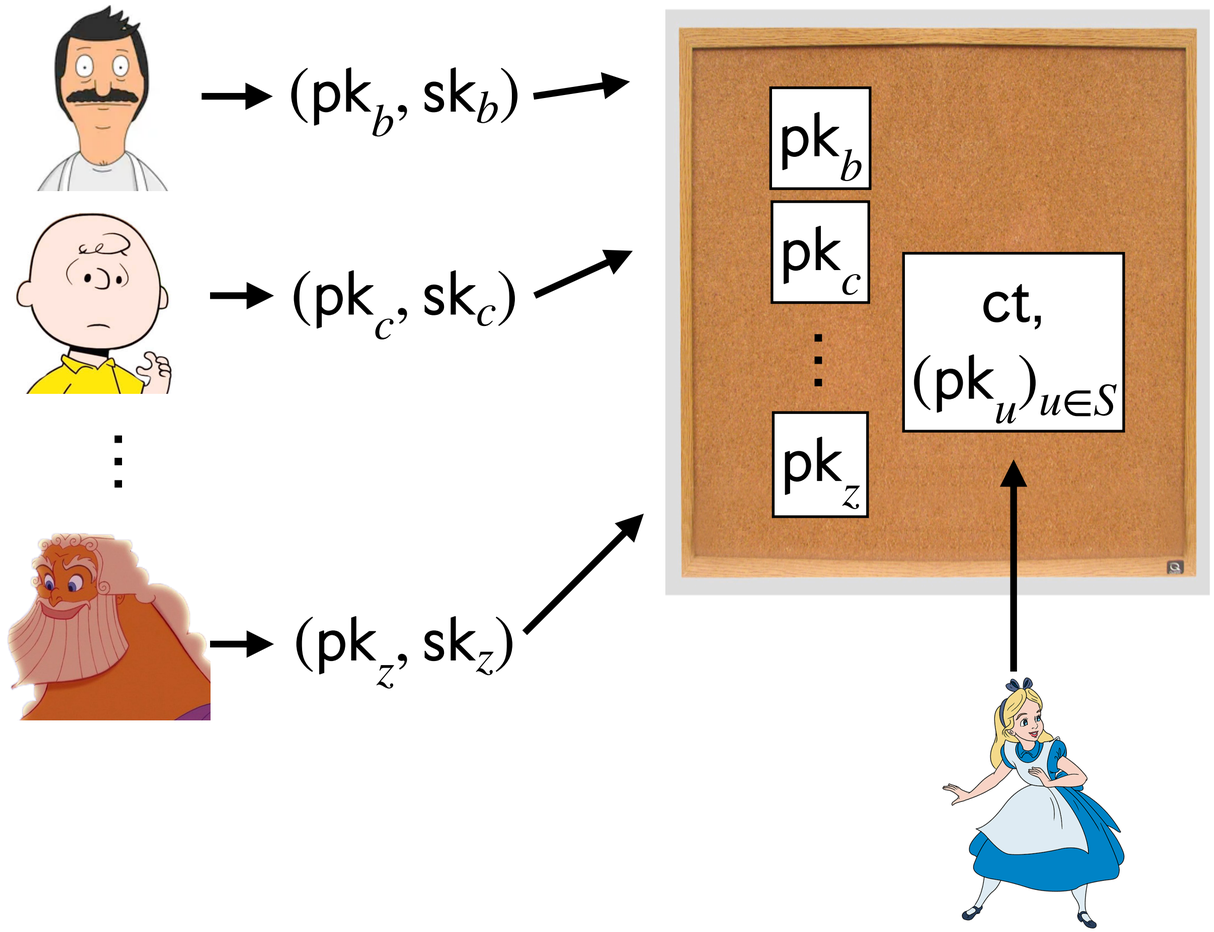
Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$



$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

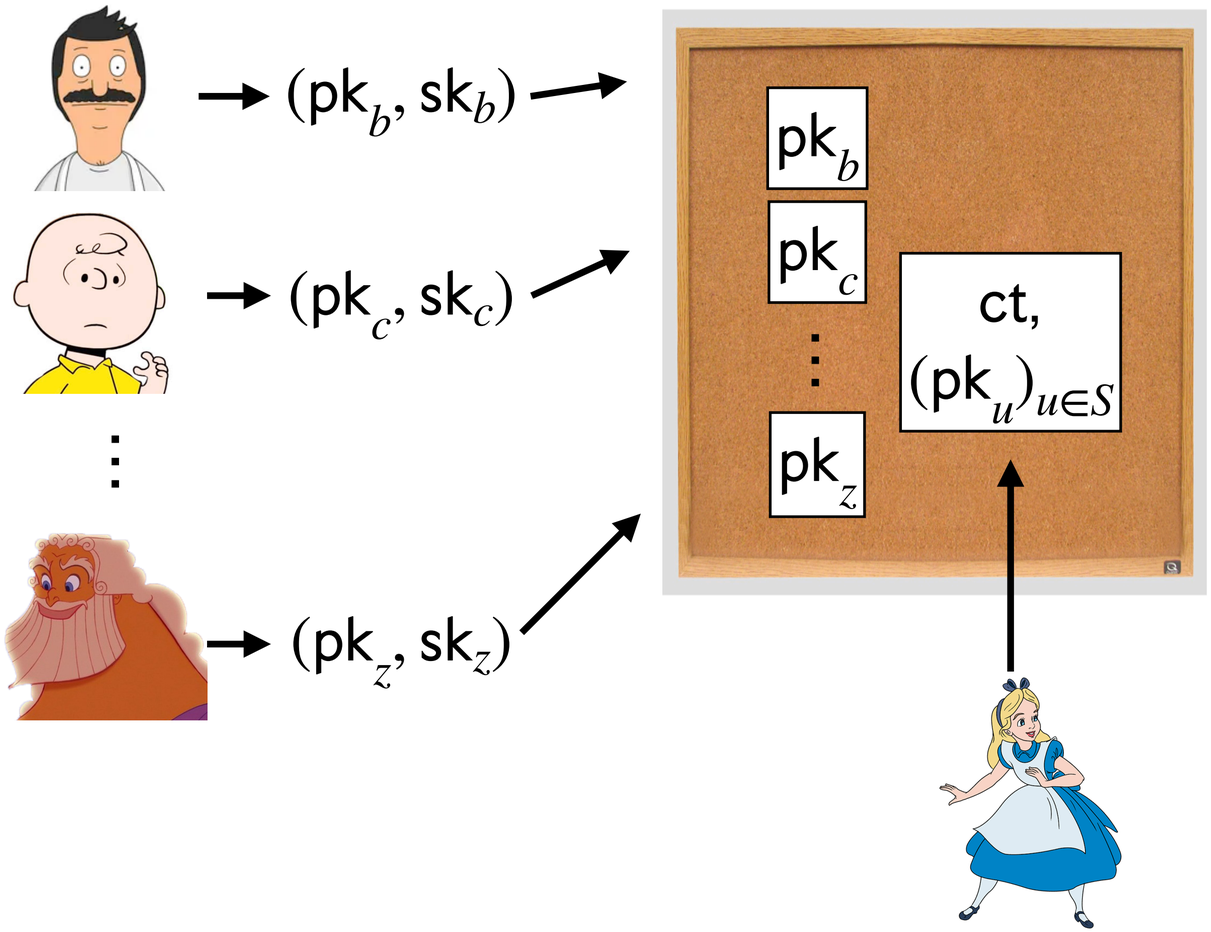
where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

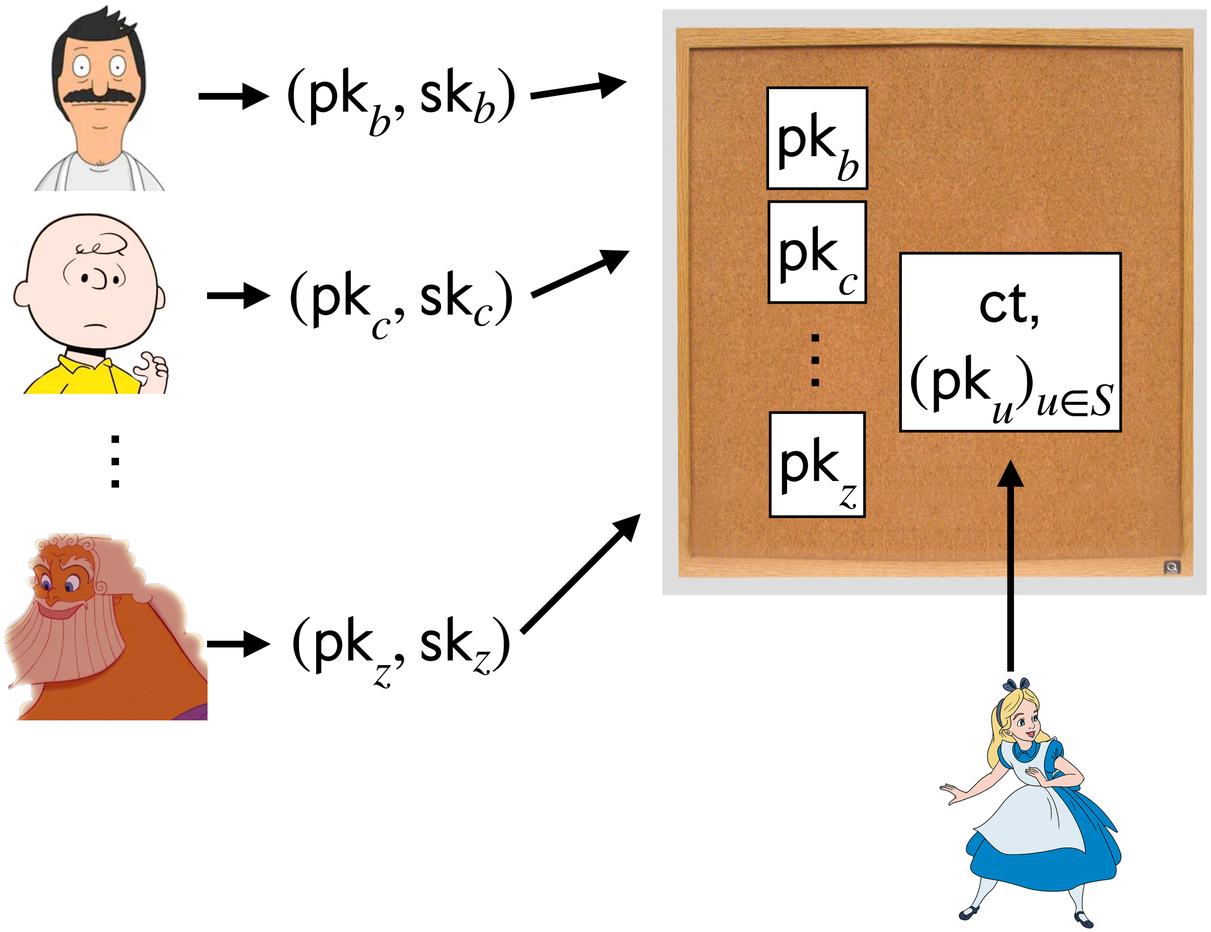
$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

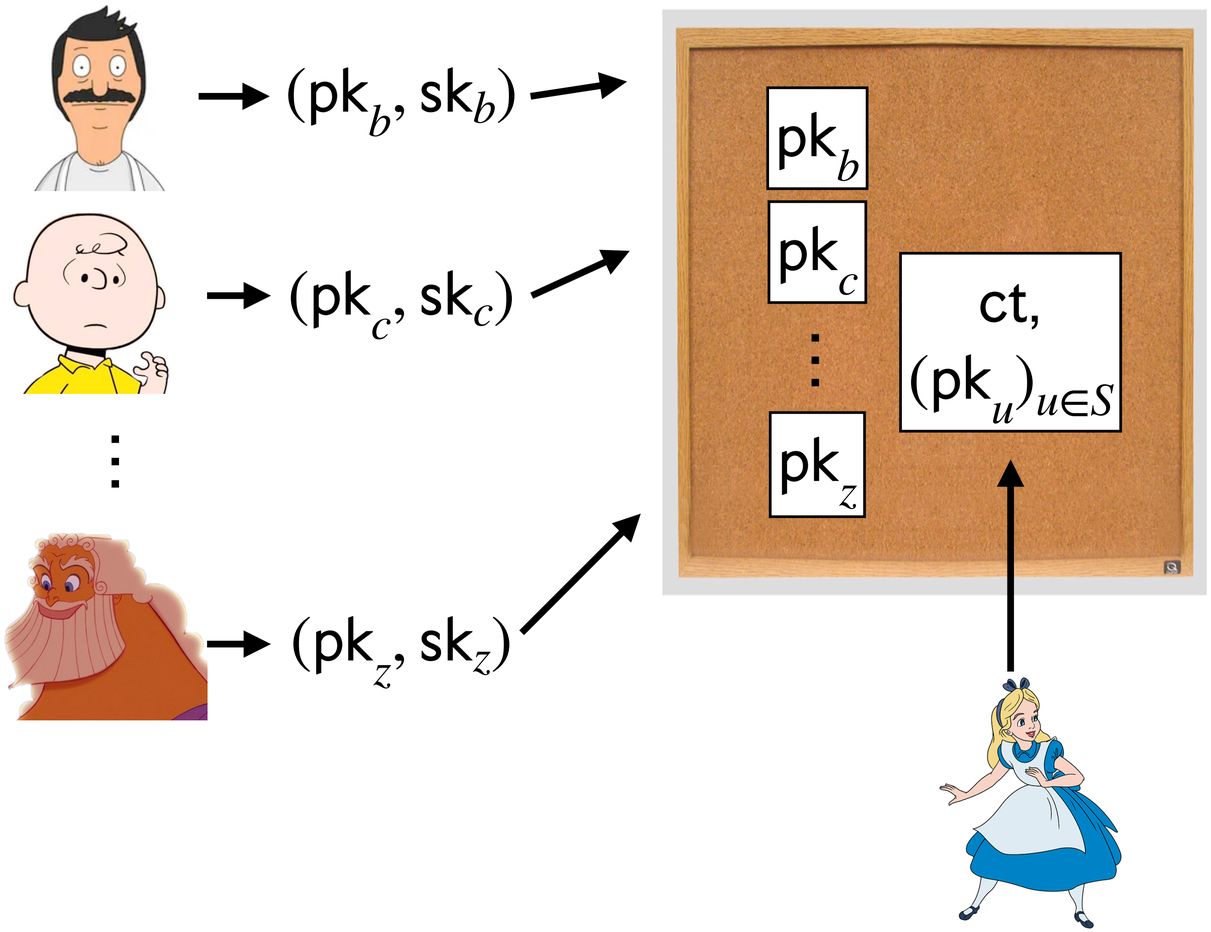
$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓ **Efficient?**

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

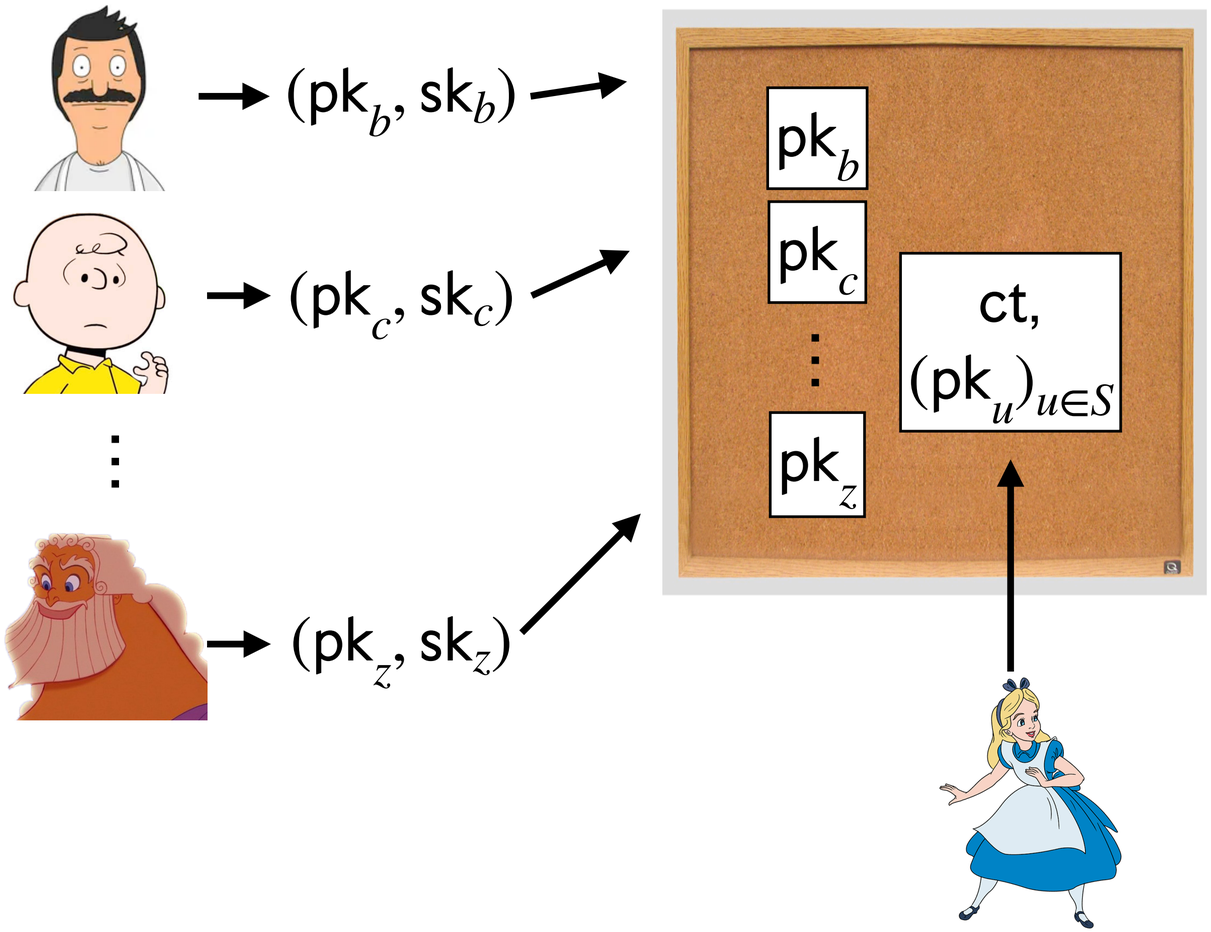
Correct? ✓ **Efficient?**

$$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$$
$$= \mathsf{poly}(\lambda, |\mathsf{msg}|, \log |S|)$$

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

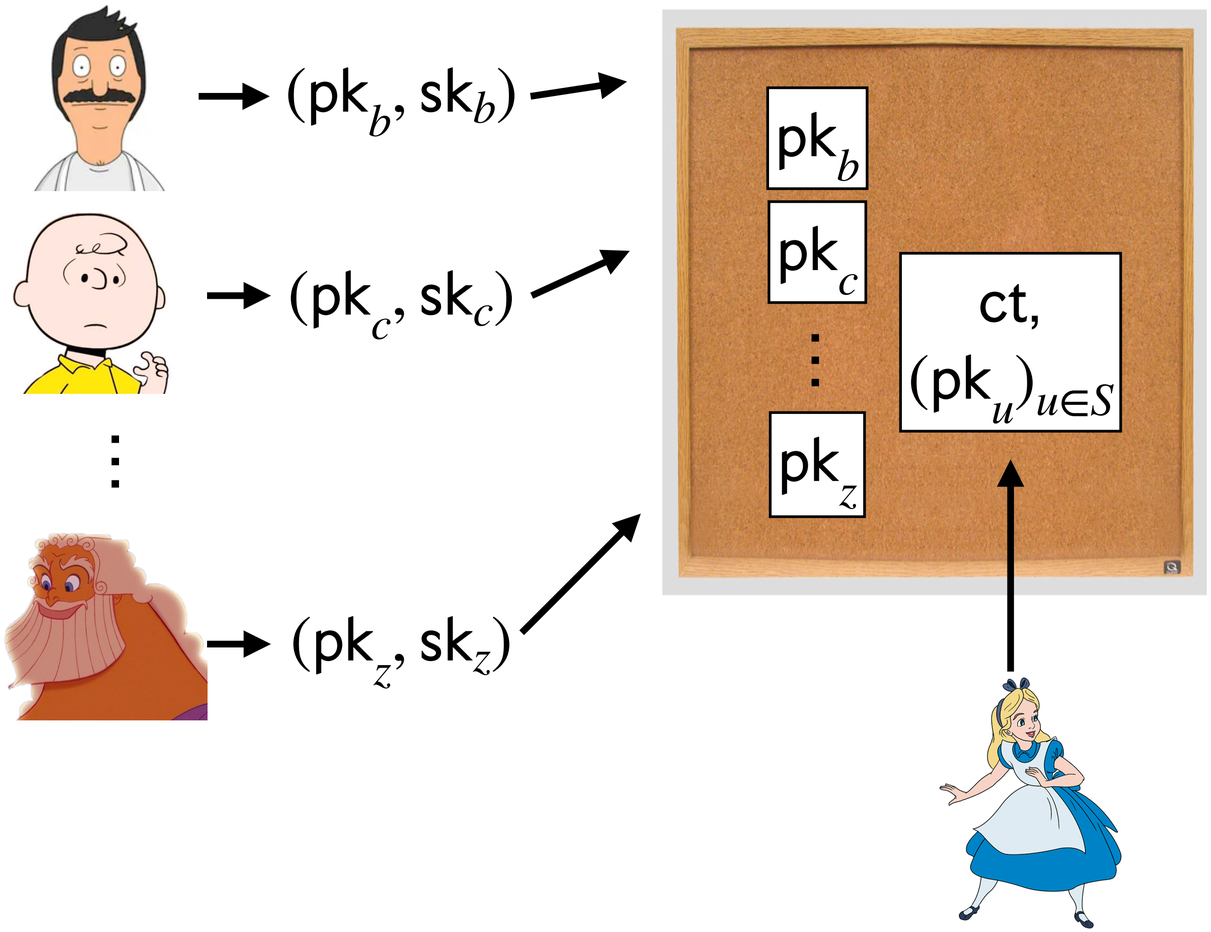
Correct? ✓ **Efficient?** ✓

$$|\mathsf{ct}| = \mathsf{poly}(\lambda, |\mathsf{msg}|, |R_L|)$$
$$= \mathsf{poly}(\lambda, |\mathsf{msg}|, \log |S|)$$

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

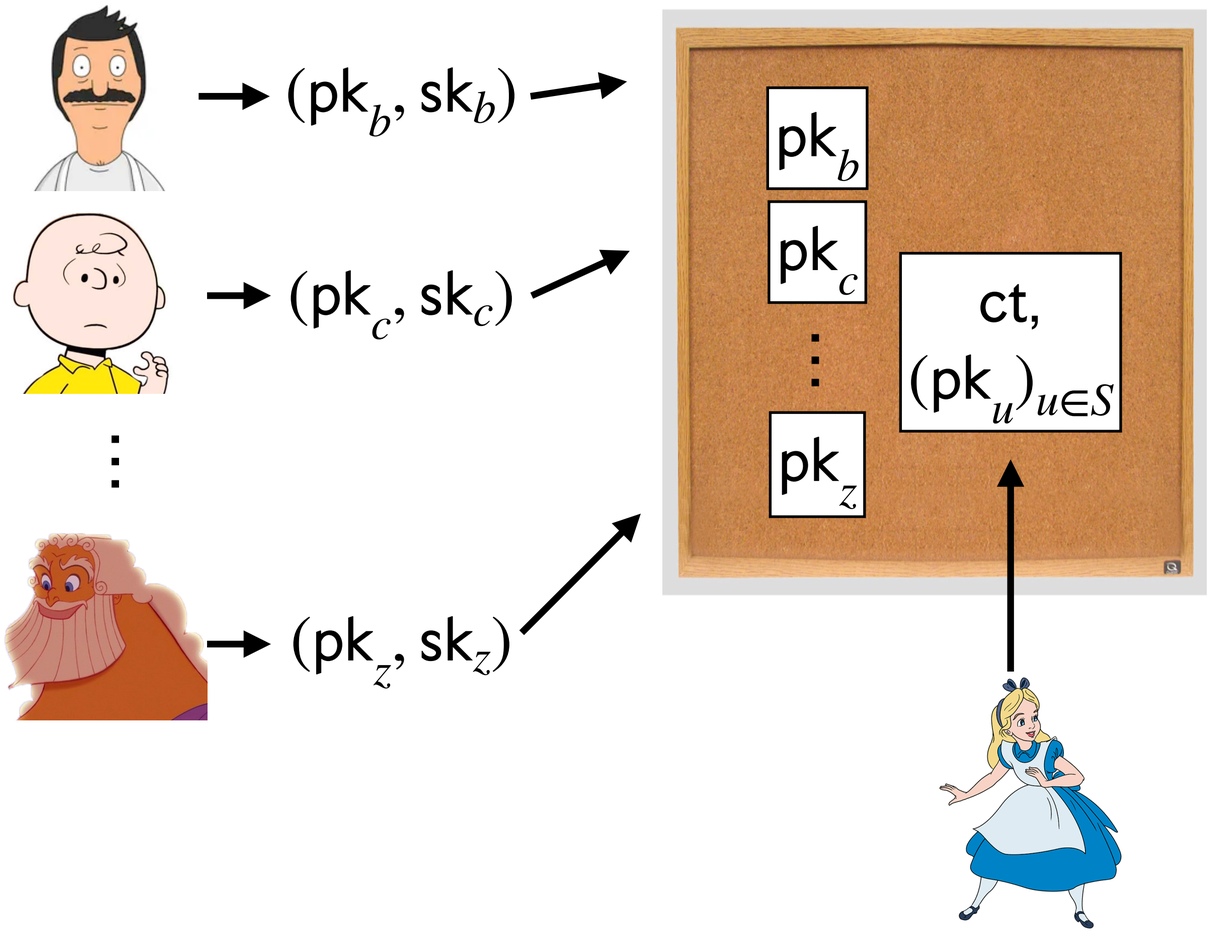
$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓ **Efficient?** ✓ **Secure?**

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

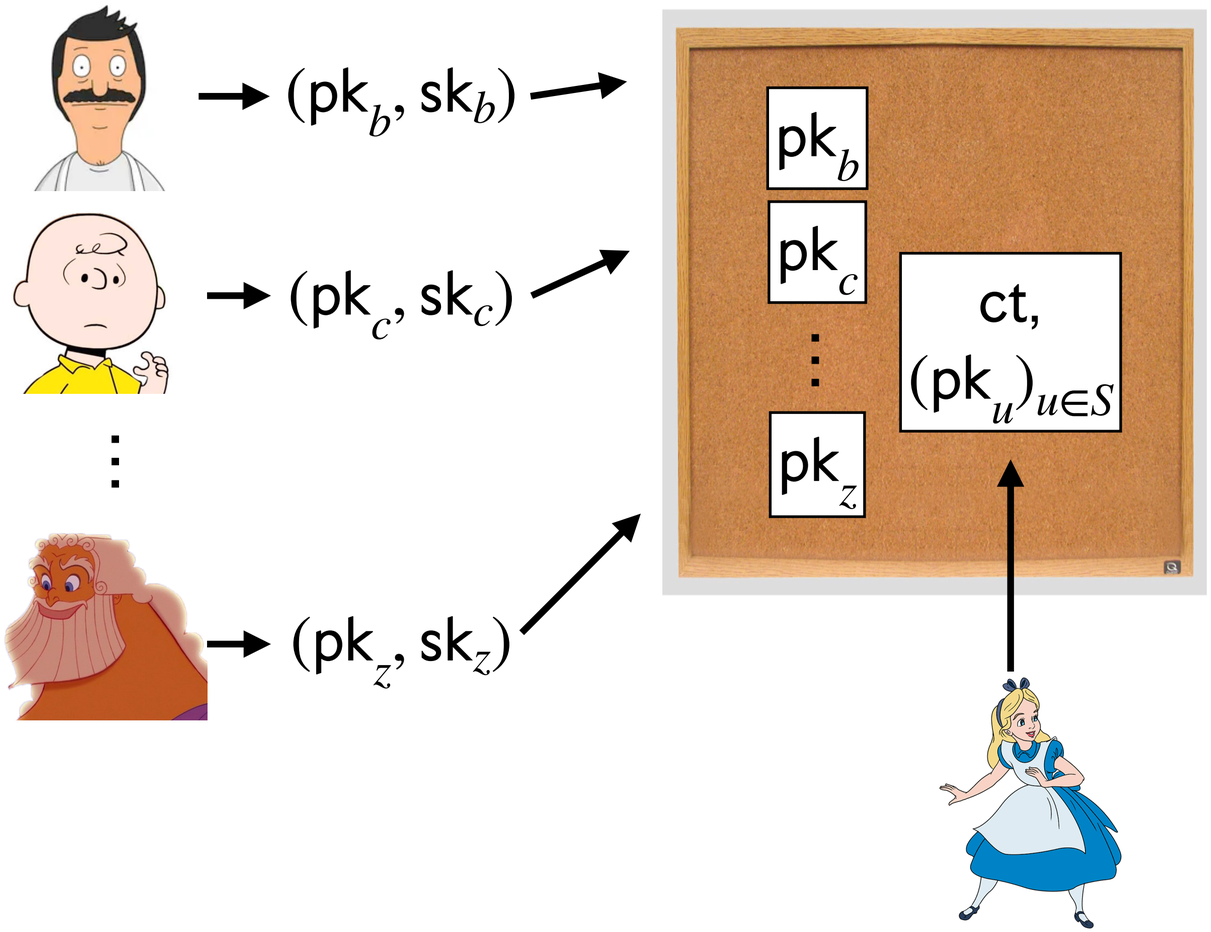
Correct? ✓ **Efficient?** ✓ **Secure?**

Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓ **Efficient?** ✓ **Secure?**

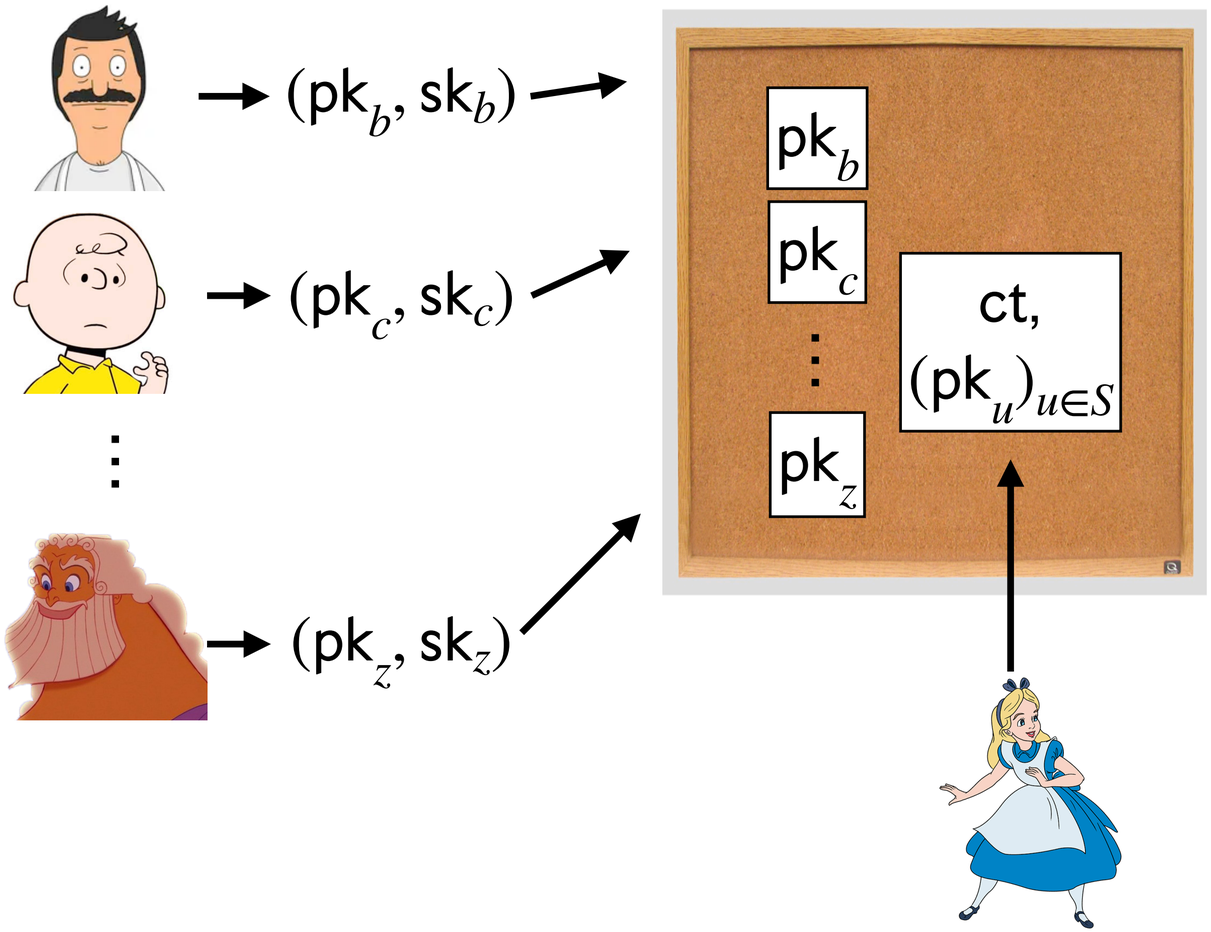
Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

Likely is the case that $x \in L$!

Attempt 2: Efficient Set Membership!

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{MT}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{MT}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓ **Efficient?** ✓ **Secure?** ✗

Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

Likely is the case that $x \in L$!

Solution: Function Binding Hash

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Recall: SSB hash binds you to a ***single input*** x_i

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Recall: SSB hash binds you to a ***single input*** x_i

Key idea: Instead bind to a ***function*** of the inputs!

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Recall: SSB hash binds you to a ***single input*** x_i

Key idea: Instead bind to a ***function*** of the inputs!

If the function output is small, no impossibility!

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Recall: SSB hash binds you to a ***single input*** x_i

Key idea: Instead bind to a ***function*** of the inputs!

Formally, if $\exists \pi$, $\mathsf{VerOpen}(\mathsf{dig}, x_i', i, \pi) = 1$,
there *must* exist an extension of values $x_j', j \neq i$ s.t.

$$f(x_1, \ldots, x_n) = f(x_1', \ldots, x_n')$$

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Recall: SSB hash binds you to a ***single input*** x_i

Key idea: Instead bind to a ***function*** of the inputs!

Formally, if $\exists \pi, \mathsf{VerOpen}(\mathsf{dig}, x_i', i, \pi) = 1$,
there *must* exist an extension of values $x_j', j \neq i$ s.t.

$$f(x_1, \ldots, x_n) = f(x_1', \ldots, x_n')$$

Can only open ***consistent*** values!

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Recall: SSB hash binds you to a ***single input*** x_i

Key idea: Instead bind to a ***function*** of the inputs!

For our application, will bind to the function

$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n} \mathsf{Dec}(\mathsf{sk}_{\mathsf{PKE}}, x_i)$$

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Recall: SSB hash binds you to a ***single input*** x_i

Key idea: Instead bind to a ***function*** of the inputs!

For our application, will bind to the function

$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n} \mathsf{Dec}(\mathsf{sk}_{\mathsf{PKE}}, x_i)$$

Any Enc(1) is ***inconsistent*** with all leaves Enc(0)!

Solution: Function Binding Hash

Consider $\mathsf{dig} = \mathsf{Hash}(\mathsf{hk}, (x_1, \ldots, x_n))$

Recall: SSB hash binds you to a ***single input*** x_i

Key idea: Instead bind to a ***function*** of the inputs!

For our application, will bind to the function

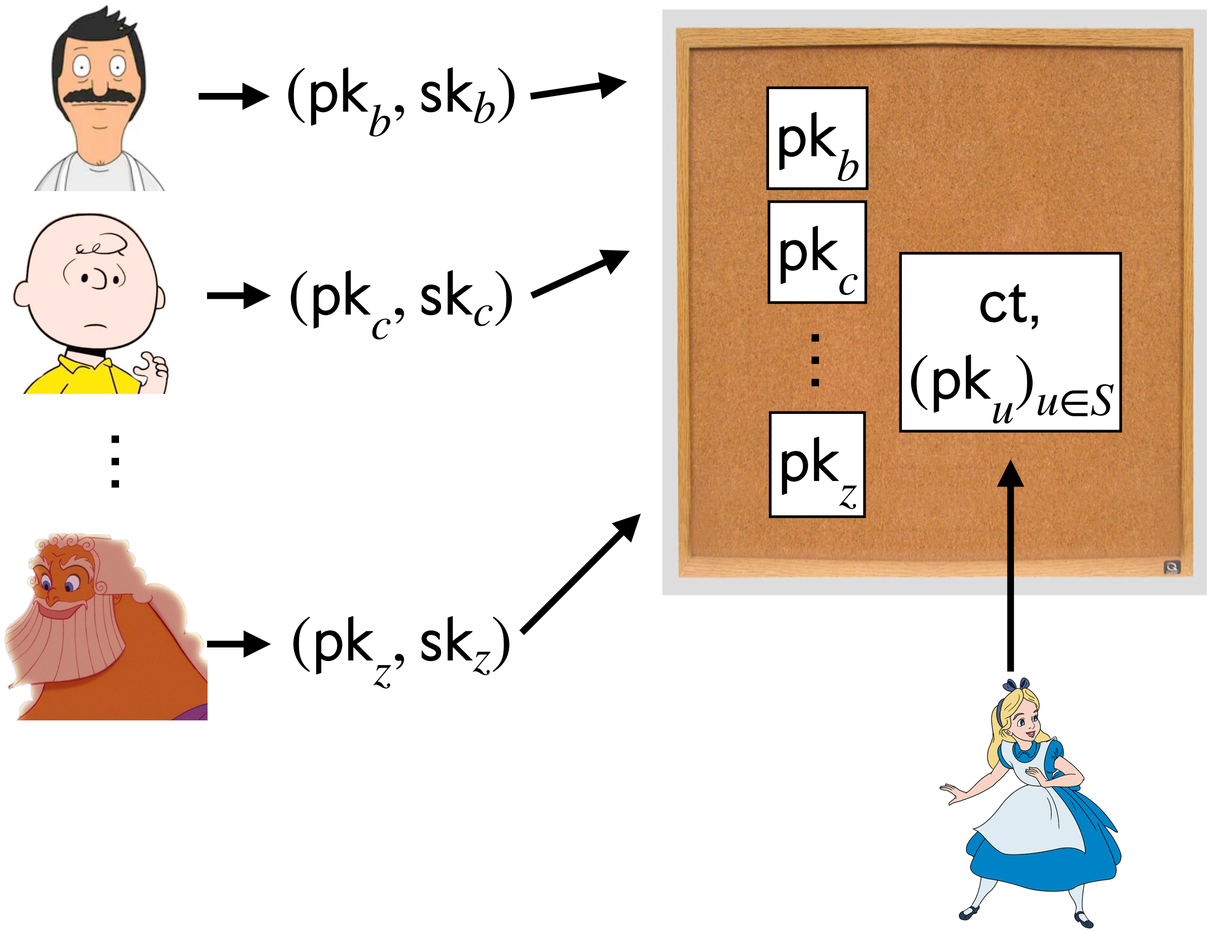
$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n} \mathsf{Dec}(\mathsf{sk}_{\mathsf{PKE}}, x_i)$$

Construction briefly: MT with FHE following [HW15],
- compute $\mathsf{Dec}(\mathsf{sk}_{\mathsf{PKE}}, \cdot)$ at leaves then
- homomorphically propagate OR of children.

Final Attempt: Using Function Binding Hash

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{FBH}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

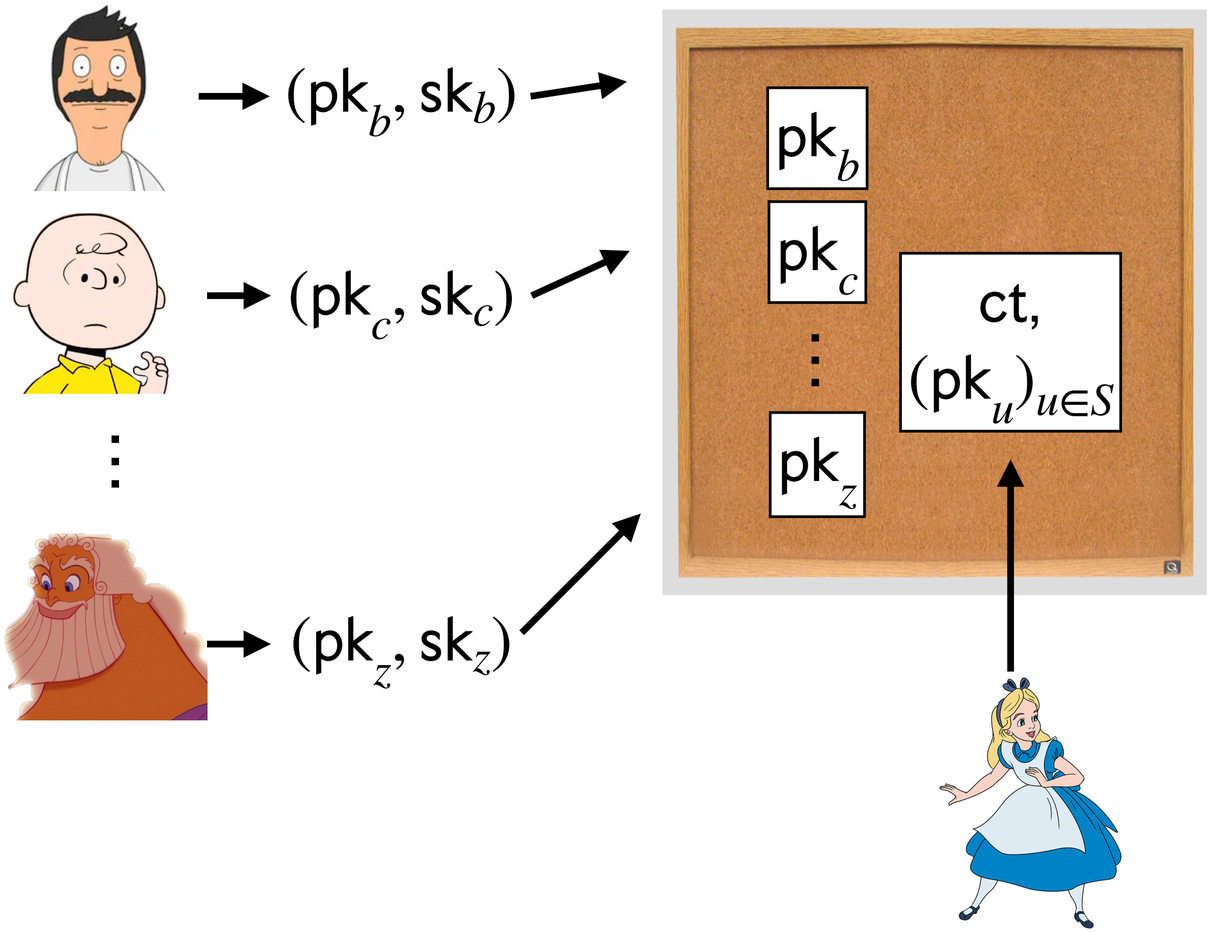
where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{FBH}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Final Attempt: Using Function Binding Hash

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{FBH}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

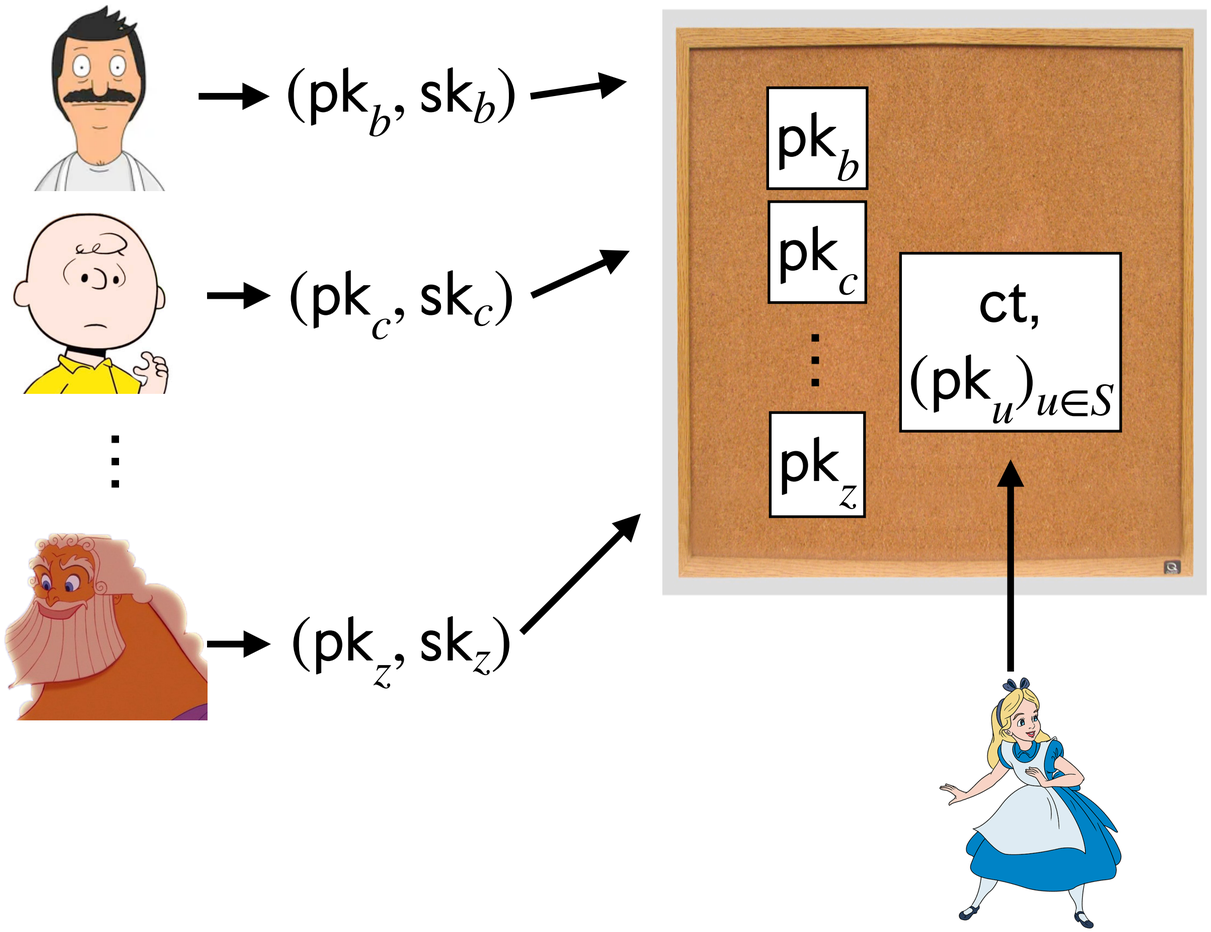
$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{FBH}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓ **Efficient?** ✓

Final Attempt: Using Function Binding Hash

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{FBH}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

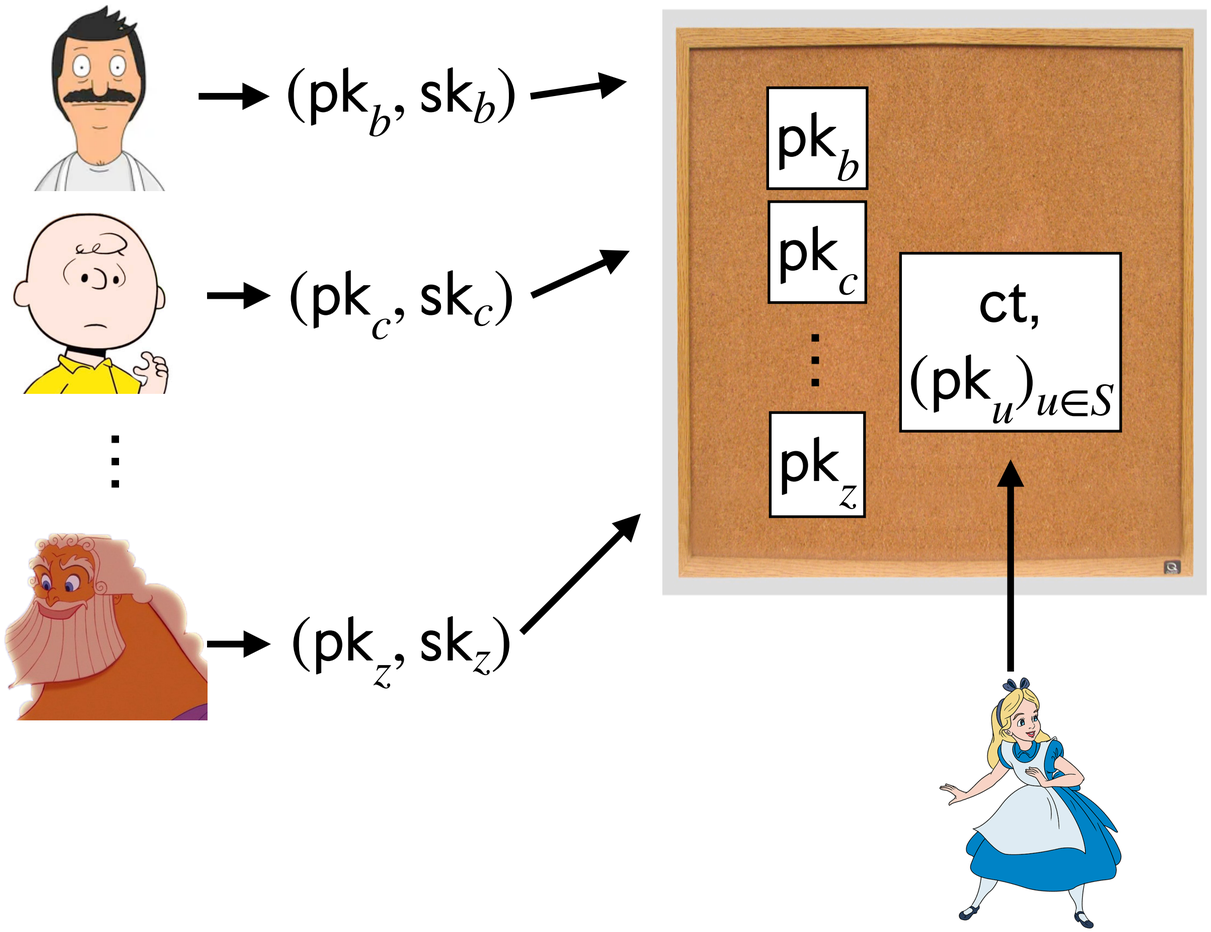
$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{FBH}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓ **Efficient?** ✓ **Secure?**

Final Attempt: Using Function Binding Hash

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{FBH}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{FBH}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

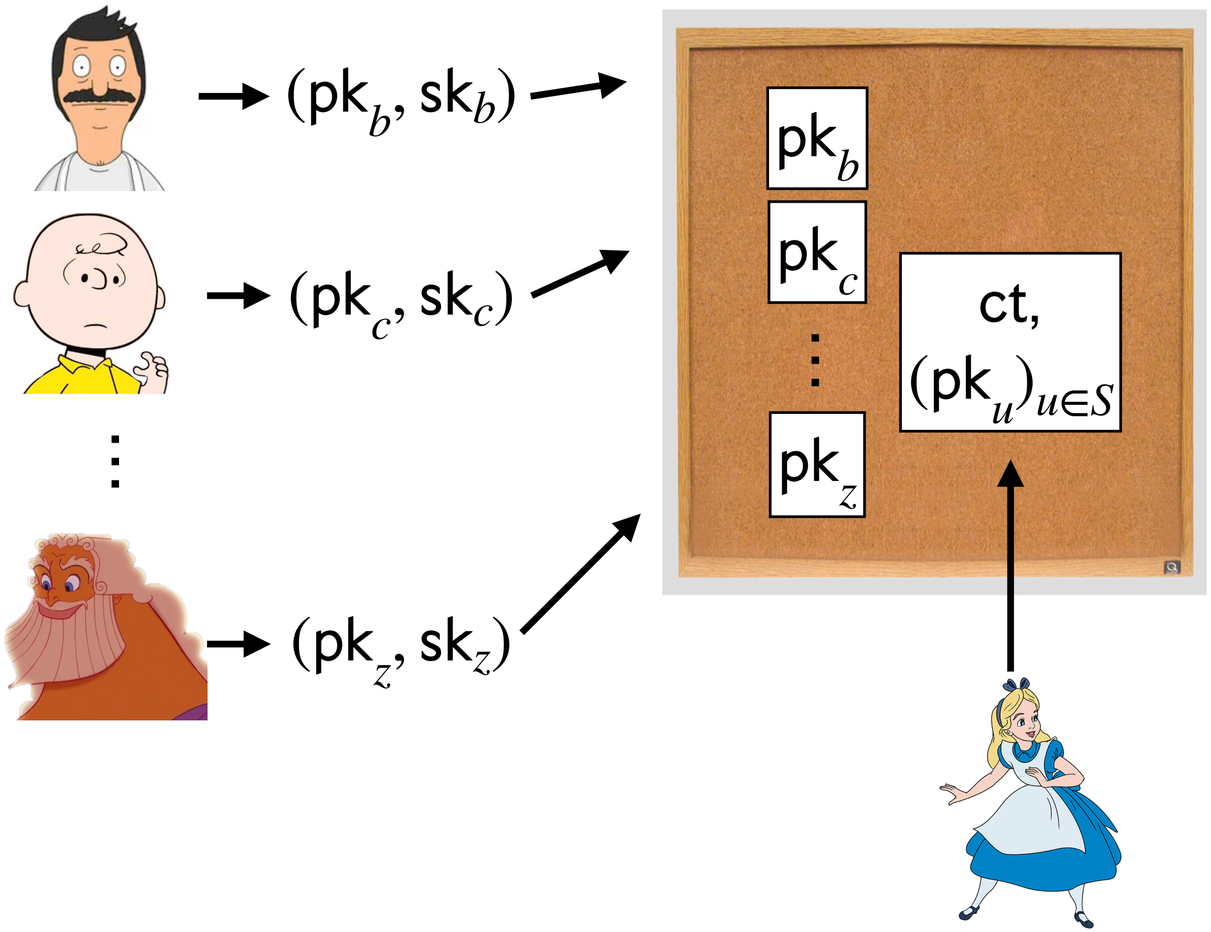
Correct? ✓ **Efficient?** ✓ **Secure?**

Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

Final Attempt: Using Function Binding Hash

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{FBH}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{FBH}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓ **Efficient?** ✓ **Secure?**

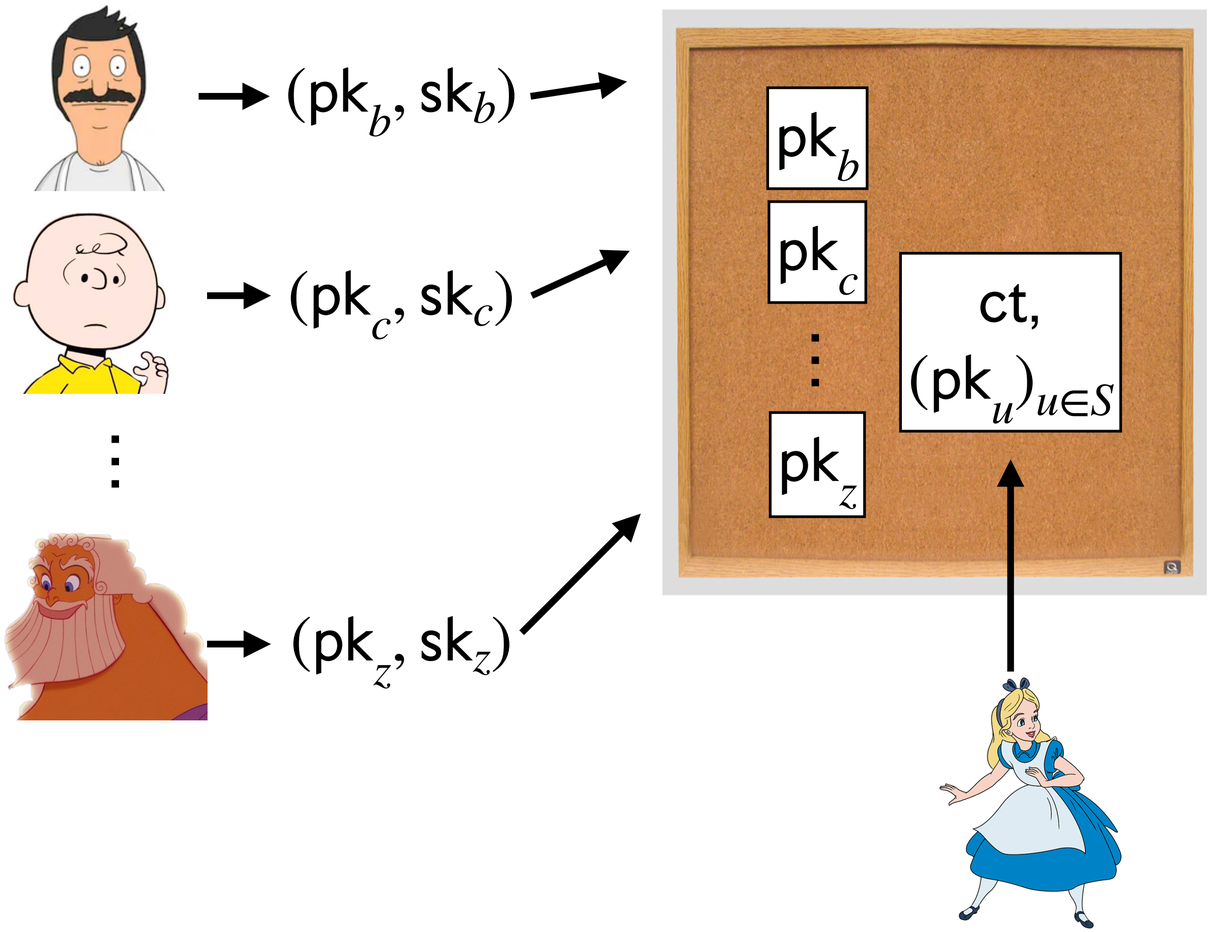
Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

By FBH security, $x \notin L$!

Final Attempt: Using Function Binding Hash

$\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{Enc}_{\mathsf{PKE}}(1; r), r)$

$\mathsf{pp} = (\mathsf{pk}_{\mathsf{PKE}}, \mathsf{hk})$



$\mathsf{Enc}(\mathsf{pp}, \mathsf{msg}, (\mathsf{pk}_u)_{u \in S})$:

Compute $\mathsf{dig} \leftarrow \mathsf{Hash}_{\mathsf{FBH}}(\mathsf{hk}, (\mathsf{pk}_u)_{u \in S})$

Output $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{WE}}(1^\lambda, \mathsf{msg}, R_L(x, \cdot))$

where $x = \mathsf{dig}$, $w = (\mathsf{sk}_v, \mathsf{pk}_v, i, \pi)$

$$R_L(x, w) \to \begin{cases} \mathsf{pk}_v = \mathsf{Enc}_{\mathsf{PKE}}(1; \mathsf{sk}_v) \\ \wedge\ \mathsf{VerOpen}_{\mathsf{FBH}}(\mathsf{dig}, \mathsf{pk}_v, i, \pi) \end{cases}$$

Correct? ✓ **Efficient?** ✓ **Secure?** ✓

Hybrid H_1: $\forall u \in S$, switch $\mathsf{pk}_u = \mathsf{Enc}_{\mathsf{PKE}}(0; r)$

By FBH security, $x \notin L$!

To Recap

New Framework for Using Witness Encryption

Witness Encryption

+

Function Binding Hash

⇒

Registered Attribute-Based Encryption

Flexible Broadcast Encryption

Optimal Broadcast in ROM

New contribution!

To Recap

New Framework for Using Witness Encryption

Witness Encryption

+

Function Binding Hash

New contribution!

⇒

Registered Attribute-Based Encryption

⇓

Flexible Broadcast Encryption

Optimal Broadcast in ROM

To Recap

New Framework for Using Witness Encryption

Witness Encryption

+

Function Binding Hash

New contribution!

⇒

Registered Attribute-Based Encryption

⇓

Flexible Broadcast Encryption

Optimal Broadcast in ROM

Details in paper!

To Recap

New Framework for Using Witness Encryption

Witness Encryption

+

Function Binding Hash

New contribution!

⇒

Registered Attribute-Based Encryption

⇓

Flexible Broadcast Encryption

Optimal Broadcast in ROM

To Recap

New Framework for Using Witness Encryption

Witness Encryption

+

Function Binding Hash

New contribution!

⇒

Registered Attribute-Based Encryption

⇓

Flexible Broadcast Encryption

Trustless Encryption Systems

Optimal Broadcast in ROM

Open Problems

Open Problems

New constructions of function-binding hash functions

- Constructions without LWE?
- For other function families?
- Impossibility for (general) function classes?

Open Problems

New constructions of function-bin

- Constructions without LWE?
- For other function families?
- Impossibility for (general) function classes?

Our FHE construction generalizes to thresholds of predicates!

Open Problems

New constructions of function-binding hash functions

- Constructions without LWE?
- For other function families?
- Impossibility for (general) function classes?

Open Problems

New constructions of function-binding hash functions
- Constructions without LWE?
- For other function families?
- Impossibility for (general) function classes?

New applications of function-binding hash functions
- with or without witness encryption

Open Problems

New constructions of function-binding hash functions
- Constructions without LWE?
- For other function families?
- Impossibility for (general) function classes?

New applications of function-binding hash functions
- with or without witness encryption

Concurrent Crypto paper define related “Predicate Extractable Hash” with applications to SNARGs

Open Problems

New constructions of function-binding hash functions
- Constructions without LWE?
- For other function families?
- Impossibility for (general) function classes?

New applications of function-binding hash functions
- with or without witness encryption

Thanks!

New Framework for Using Witness Encryption

Witness Encryption

+

Function Binding Hash

New contribution!

⇒

Registered Attribute-Based Encryption

⇓

Flexible Broadcast Encryption

Trustless Encryption Systems

Optimal Broadcast in ROM