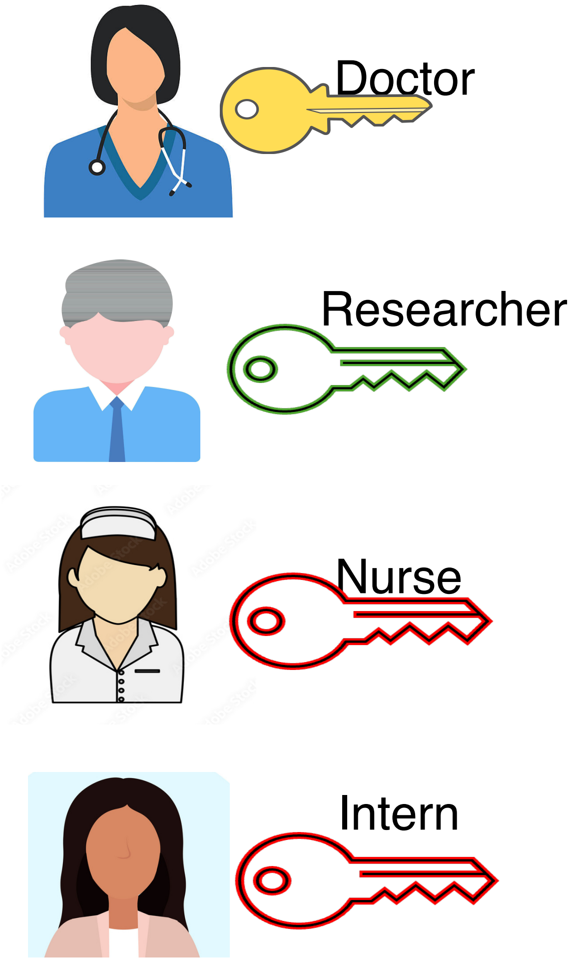
Attribute-Based Encryption for Turing Machines from Lattices

Shweta Agrawal (IIT Madras)
Simran Kumari (IIT Madras)
Shota Yamada (AIST Tokyo)

Travel Funded by Google India Travel Grant

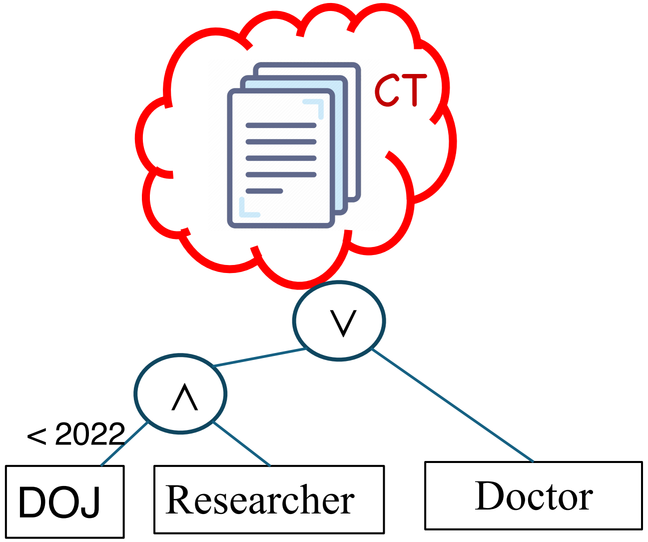
Attribute-Based Encryption[SW05,GPSW06]



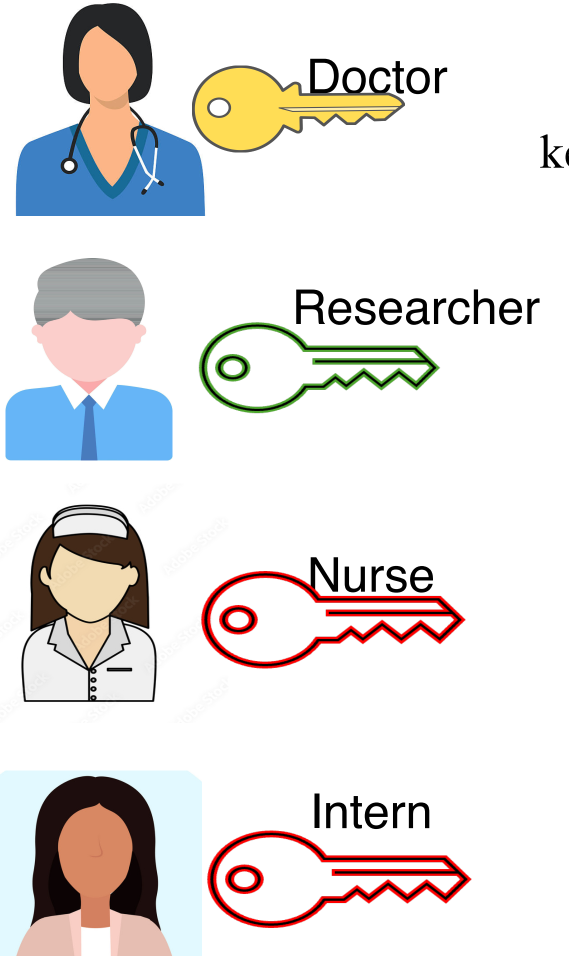
Fine grained version of a PKE Scheme

Enables access control on encrypted data

Attribute-Based Encryption[SW05,GPSW06]

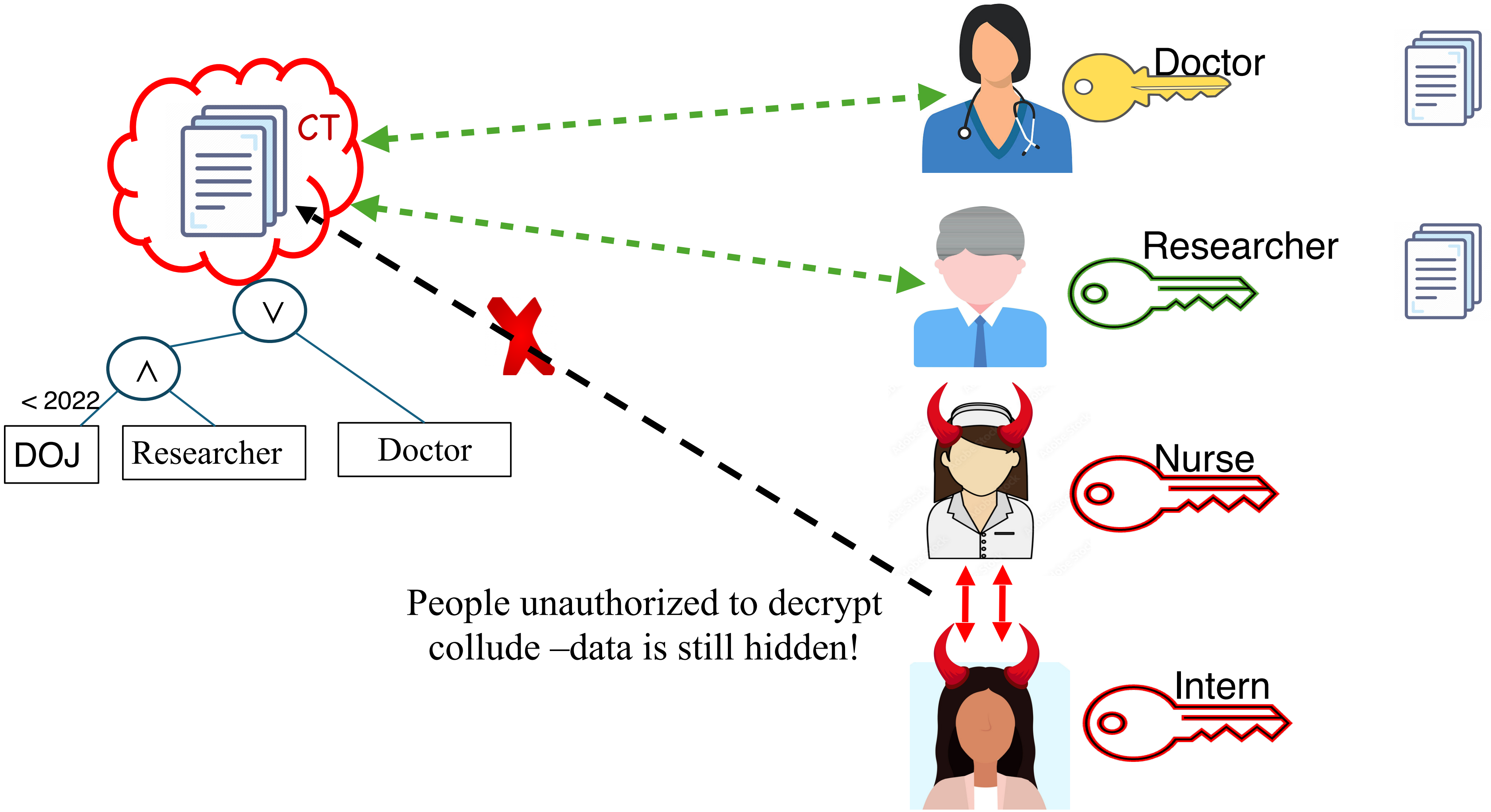


ct linked with “policy”—tells us who can decrypt!



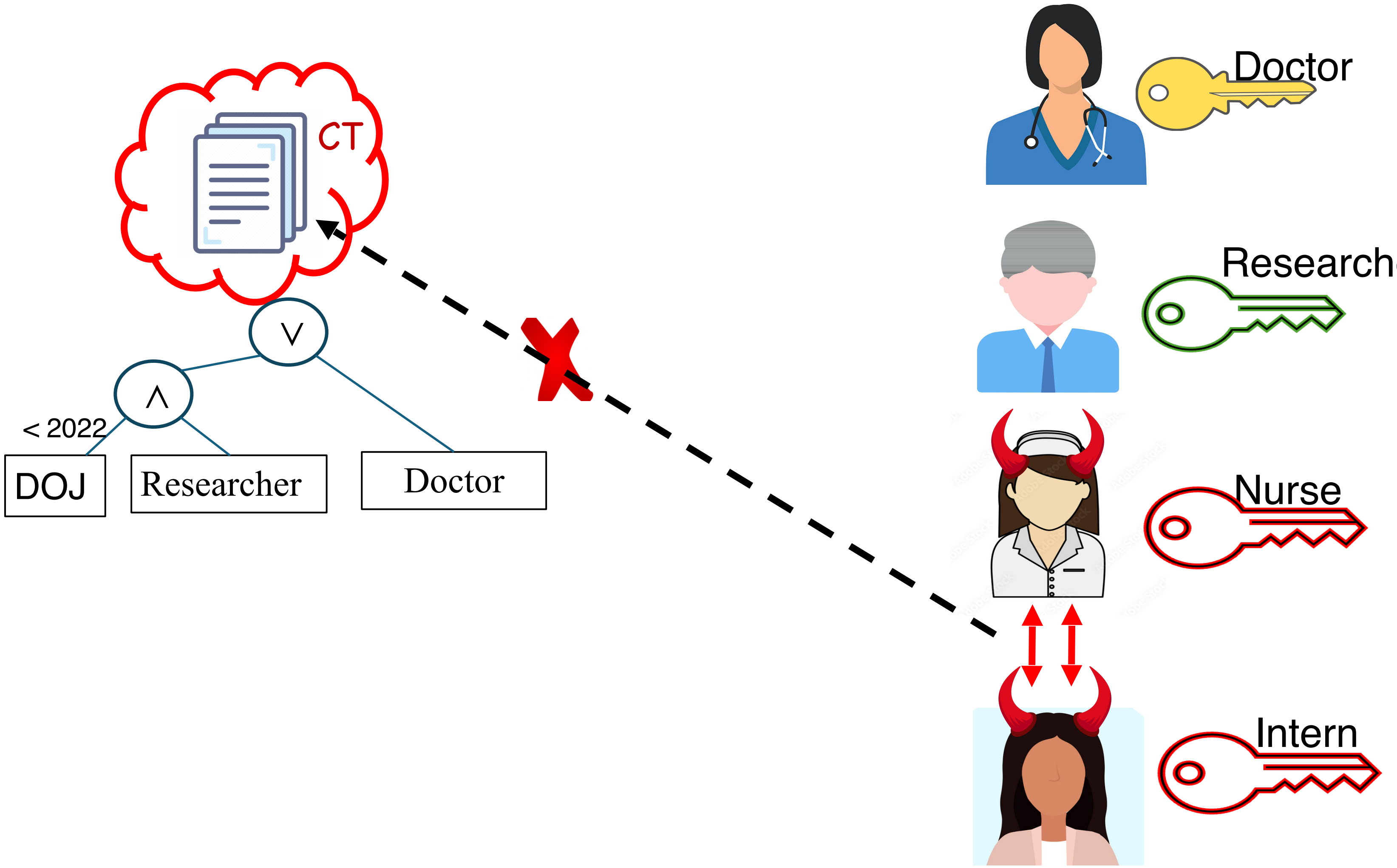
keys linked with “attributes”

Attribute-Based Encryption [SW05,GPSW06]



People unauthorized to decrypt collude –data is still hidden!

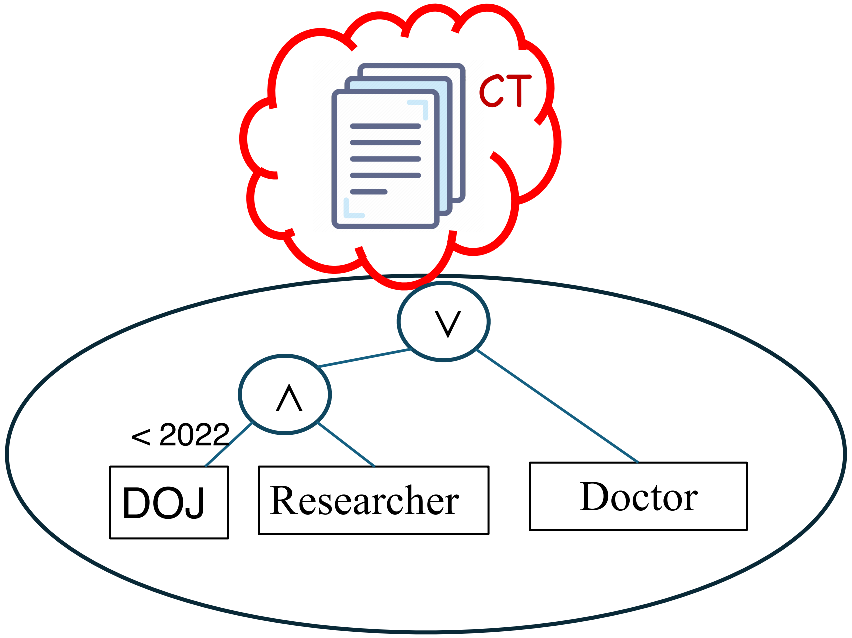
Attribute-Based Encryption[SW05,GPSW06]



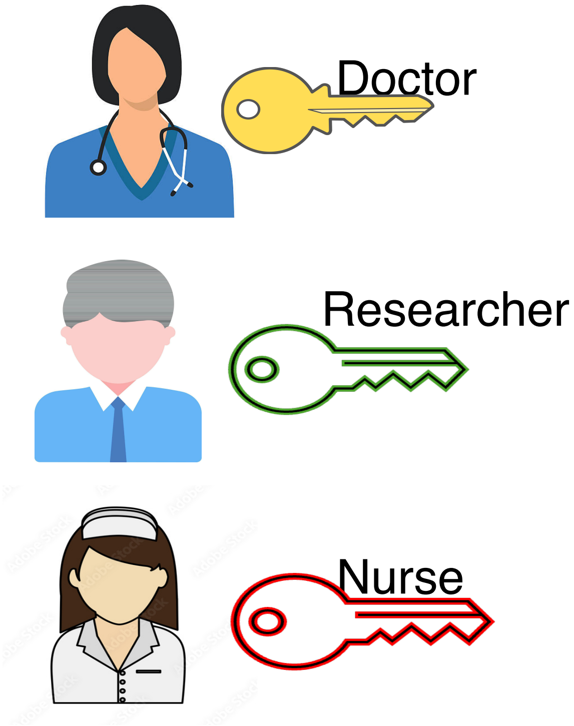
No collusion of unauthorized parties can break the security

– Collusion Resistance

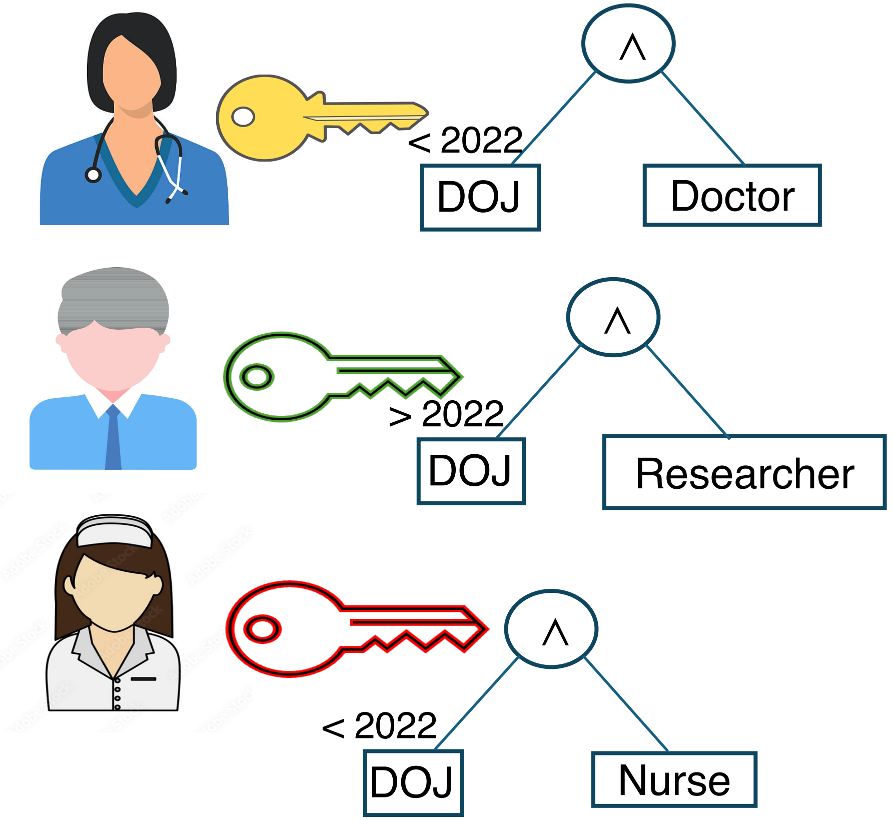
Ciphertext-Policy ABE (CP-ABE)



Policy associated with
ciphertext



Key-Policy ABE (KP-ABE)



Policy associated with keys

ABE Landscape

Uniform Model of Computation

Construction	Model	Assumptions
[Wat12]	DFA	Q-type assumptions on bilinear maps
[GWW19], [AMY19b]	DFA	static assumptions on bilinear maps
[LL20]	NL	pairings

ABE Landscape

Uniform Model of Computation

Construction	Model	Assumptions
[Wat12]	DFA	Q-type assumptions on bilinear maps
[GWW19], [AMY19b]	DFA	static assumptions on bilinear maps
[LL20]	NL	pairings
[GKP+13]	TM	Extractable WE, SNARKs
[AS17]	TM	iO
[AM18], [KNTY18]	TM	compact FE

ABE Landscape

Uniform Model of Computation
(Conjectured Post-Quantum regime)

Construction	Model	Assumptions
[AS17]	DFA (bounded-collision)	LWE
[AMY19a]	NFA (Secret-key)	LWE
[HLL24b]	DFA, L	LWE, Evasive LWE

ABE Landscape

Uniform Model of Computation
(Conjectured Post-Quantum regime)

Construction	Model	Assumptions
[AS17]	DFA (bounded-collision)	LWE
[AMY19a]	NFA (Secret-key)	LWE
[HLL24b]	DFA, L	LWE, Evasive LWE
This Work	**NL**	**LWE, Evasive LWE, Tensor LWE**

ABE Landscape

Uniform Model of Computation
(Conjectured Post-Quantum regime)

Construction	Model	Assumptions
[AS17]	DFA (bounded-collision)	LWE
[AMY19a]	NFA (Secret-key)	LWE
[HLL24b]	DFA, L	LWE, Evasive LWE
This Work	**NL**	**LWE, Evasive LWE, Tensor LWE**
This Work	**TM**	**LWE, Evasive LWE, Circular Tensor LWE**

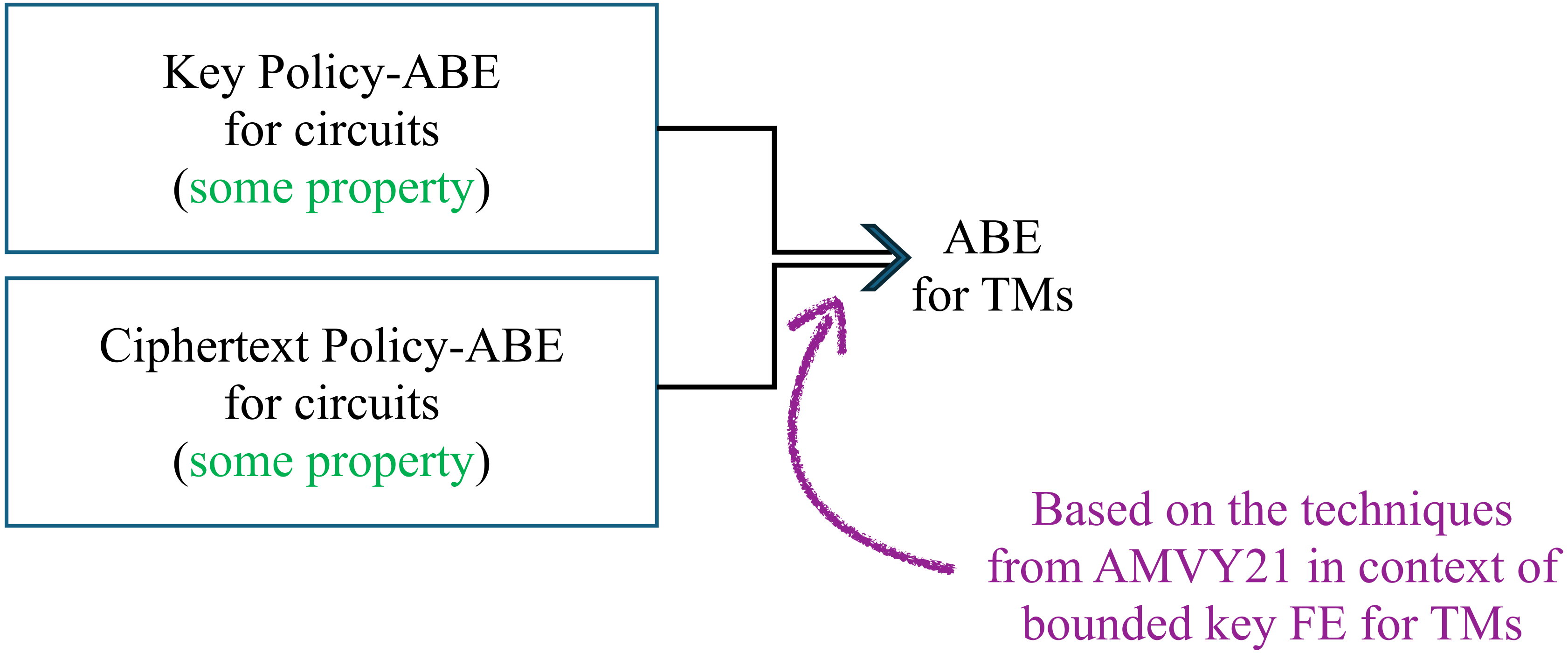
ABE for Turing Machines : Pathway [AKY24]

Key Policy-ABE
for circuits
(some property)

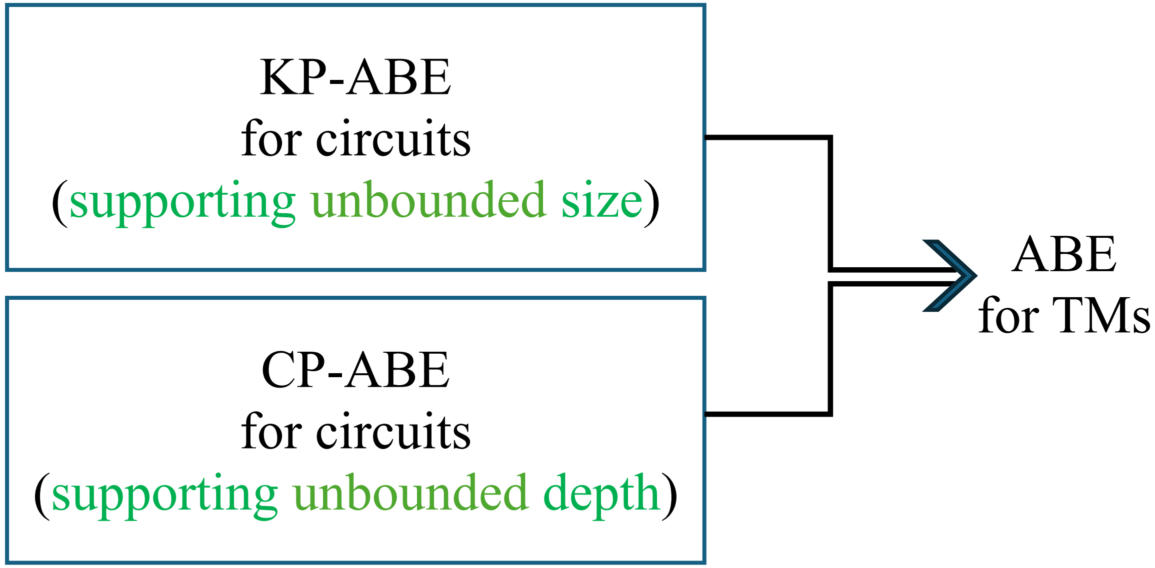
Ciphertext Policy-ABE
for circuits
(some property)

ABE
for TMs

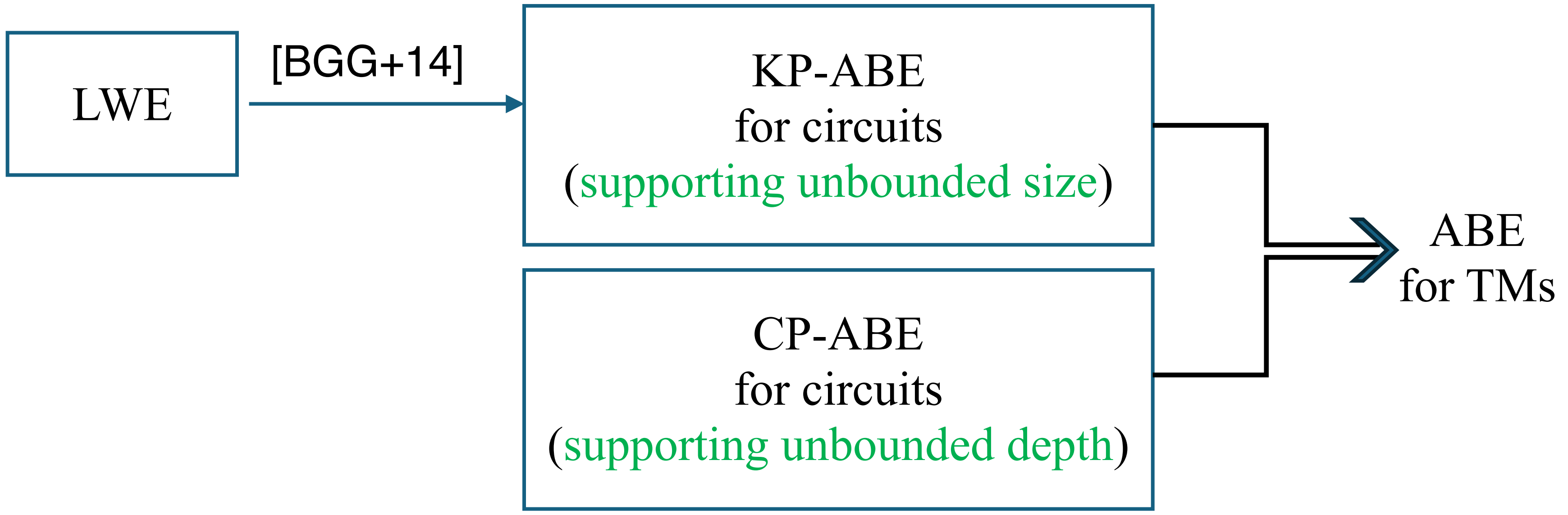
ABE for Turing Machines : Pathway [AKY24]



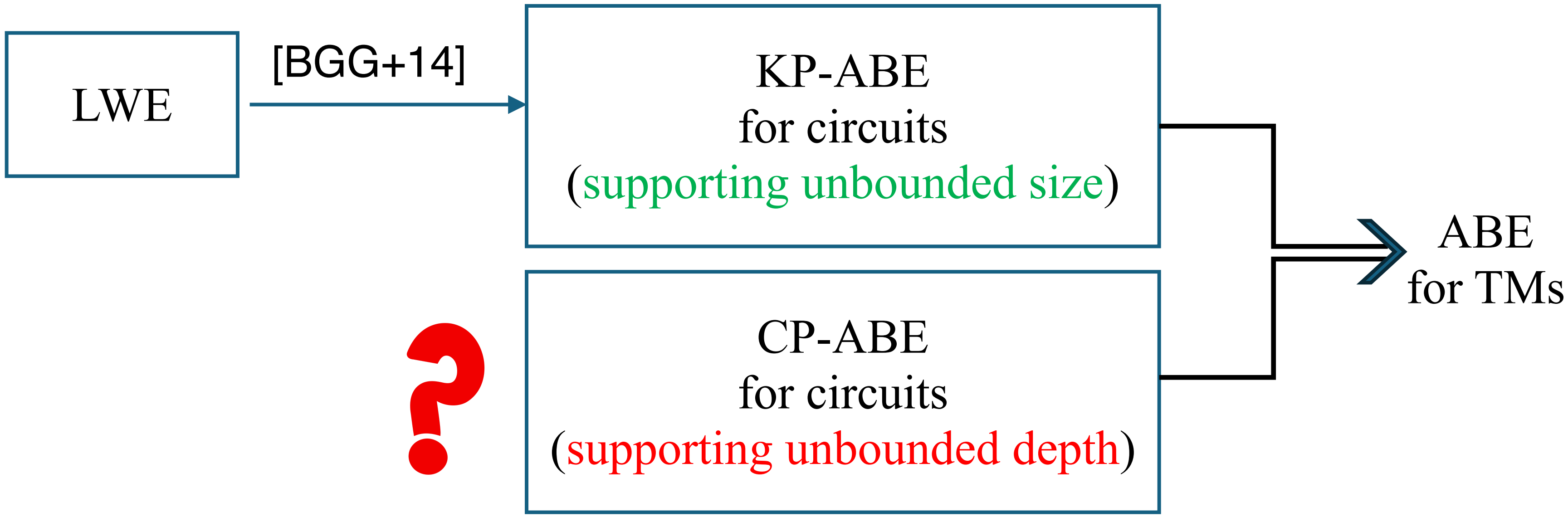
ABE for Turing Machines : Pathway [AKY24]



ABE for Turing Machines : Pathway [AKY24]



ABE for Turing Machines : Pathway [AKY24]



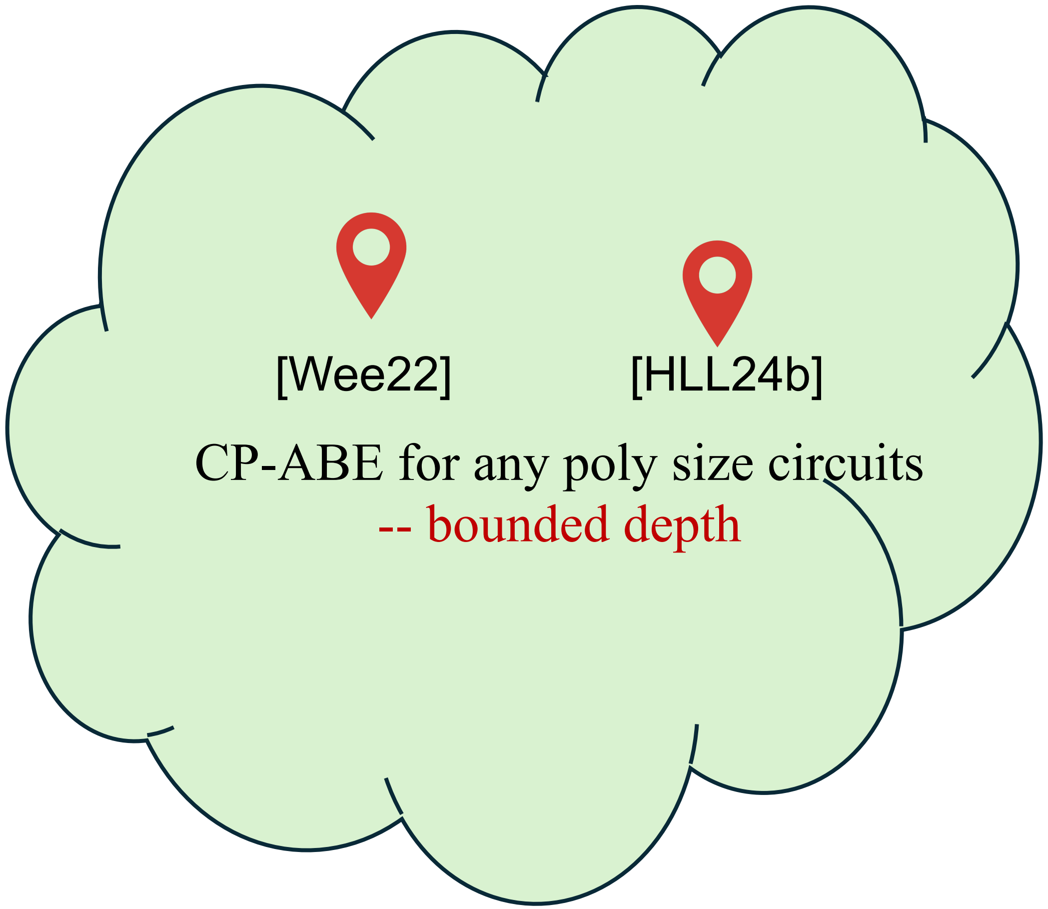
CP-ABE Landscape

[Wee22]

CP-ABE for any poly size circuits
-- bounded depth

new lattice assumptions
(Evasive LWE and Tensor LWE)

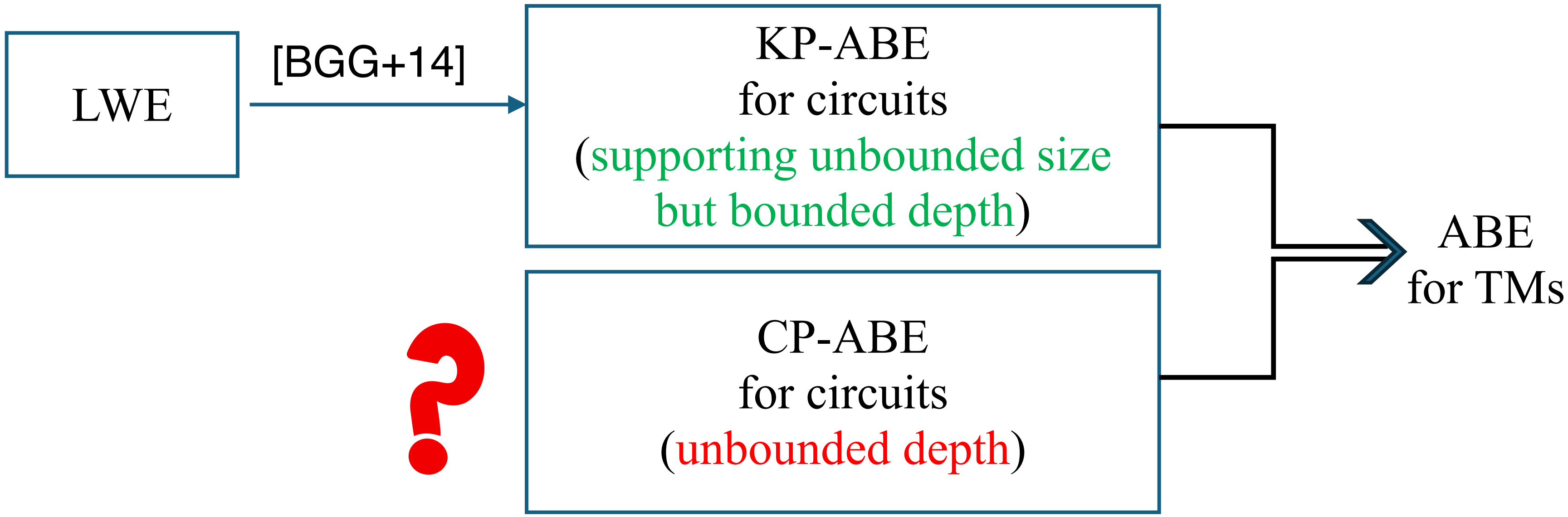
CP-ABE Landscape



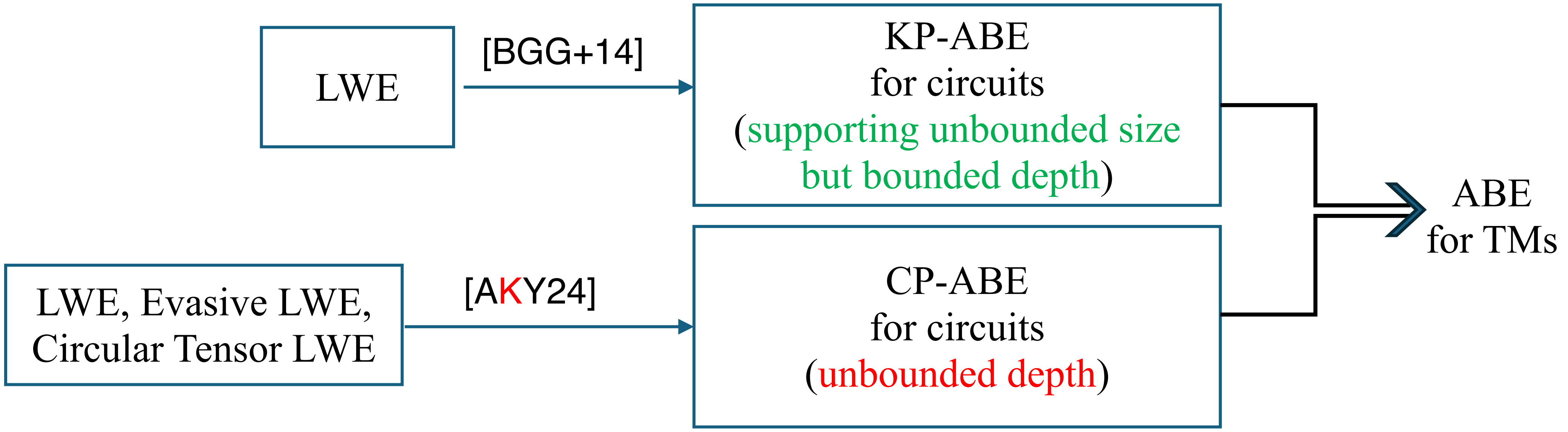
new lattice assumptions
(Evasive LWE and Tensor ~~L~~WE)

variant

ABE for Turing Machines : Pathway



ABE for Turing Machines : Pathway



Unbounded CP-ABE [AKY24] : Outline

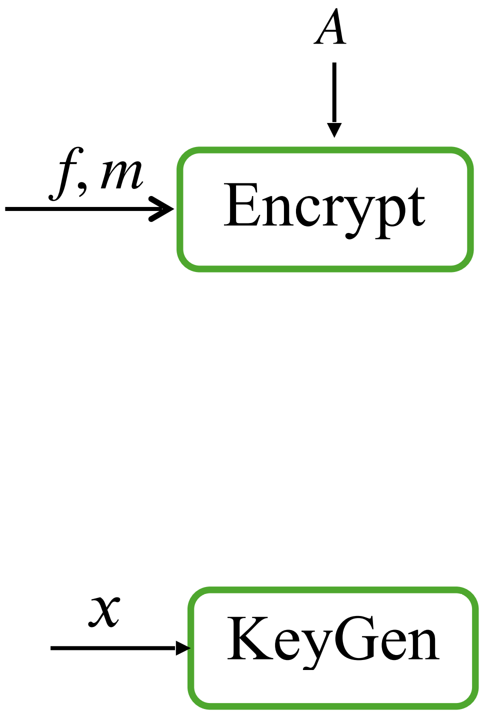
starting point of our scheme

Wee's CP-ABE[Wee22]

To understand: the reason for *bounded* depth

Wee's CP-ABE [Wee22]

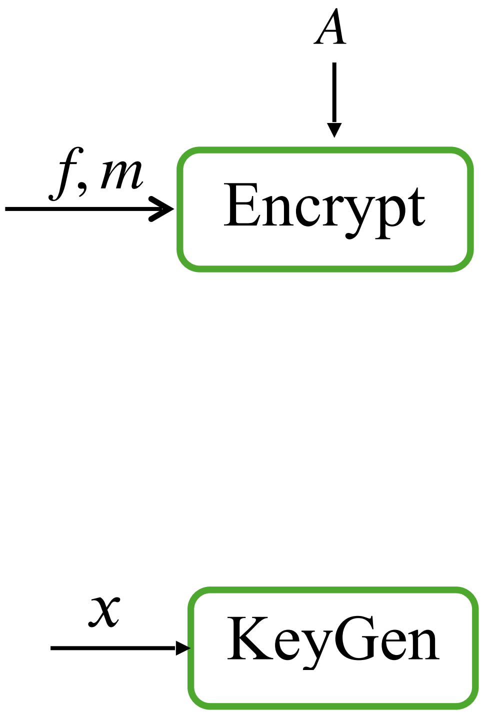
Randomised attribute encoding : key idea towards CP-ABE



$$\boxed{s} \; \boxed{A - x \otimes G} \otimes \boxed{r} + \boxed{err}$$

Wee's CP-ABE[Wee22]

Randomised attribute encoding : key idea towards CP-ABE



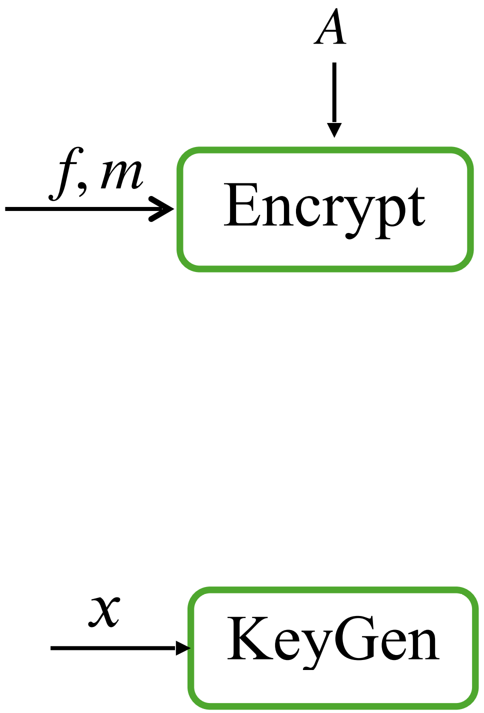
$$\boxed{s}\ \boxed{A - x \otimes G} \otimes \boxed{r} + \boxed{err}$$

enc randomness

Keygen randomness

Wee's CP-ABE[Wee22]

Randomised attribute encoding : key idea towards CP-ABE



$$s \quad A - x \otimes G \quad \otimes \quad r \quad + \quad err$$

appears somewhere in decryption!

Wee's CP-ABE [Wee22]

Randomised attribute encoding : key idea towards CP-ABE

A

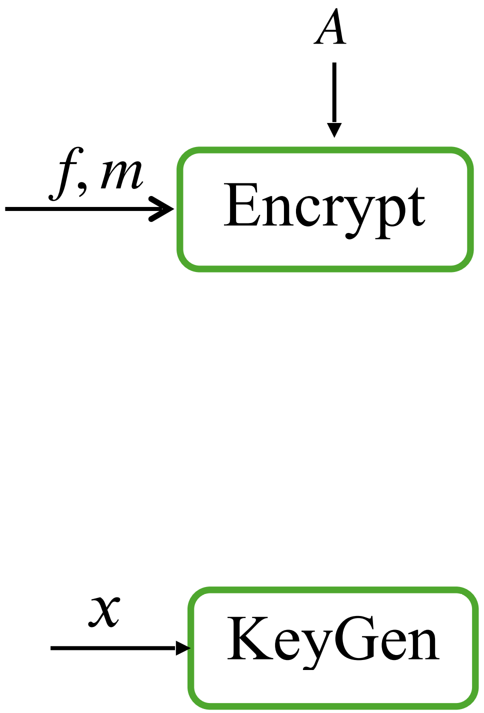
$\xrightarrow{f, m}$ Encrypt

$$\left(\boxed{s} \ \boxed{A - x \otimes G} \otimes \boxed{r} + \boxed{err} \right) \boxed{H_{f,x} \otimes I}$$

$\xrightarrow{x}$ KeyGen

BGG+14 matrices $H_{f,x}$, H_f

Wee's CP-ABE [Wee22]



$$\boxed{s}\ \boxed{A' - f(x)G} \otimes \boxed{r} + \boxed{err \cdot H_{f,x}}$$

Homomorphism preserved under randomisation!

Wee's CP-ABE [Wee22]

$$\xrightarrow{f,\,m} \boxed{\text{Encrypt}} \quad \text{(input } A \text{ from above)}$$

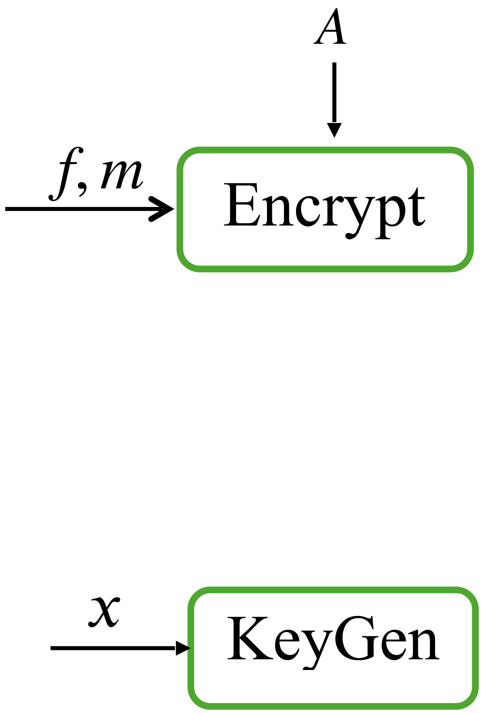
$$\xrightarrow{x} \boxed{\text{KeyGen}}$$

$$\boxed{s} \; \boxed{A' - f(x)G} \otimes \boxed{r} + \boxed{err \cdot H_{f,x}}$$

randomised homomorphic encoding

$\boldsymbol{H_{f,x}}, \boldsymbol{H_f}$: norm grows exp wrt the depth of f !!

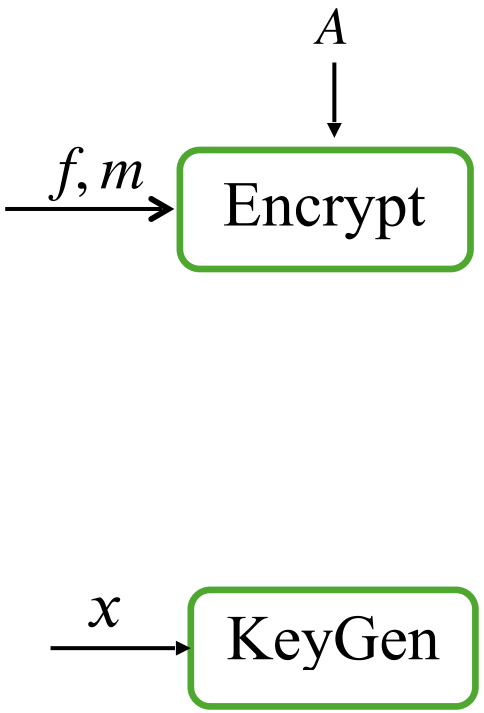
Wee's CP-ABE [Wee22]



$$\boxed{s}\ \boxed{A' - f(x)G} \otimes \boxed{r} + \boxed{err \cdot H_{f,x}}$$

Error grows exp w.r.t depth of f !!

Wee's CP-ABE[Wee22]



$$\boxed{s}\ \boxed{A' - f(x)G} \otimes \boxed{r} + \boxed{e_{large}}$$

Error grows exp w.r.t depth of f !!
Bounded Params w.r.t depth of f!

Our Goal

$$s \quad A' - f(x)G \otimes r + e_{large}$$

randomised homomorphic encoding with *large* error

$$s \quad A' - f(x)G \otimes r + e_{small}$$

randomised homomorphic encoding with *small* error

Our Goal

$$\boxed{s(I \otimes r)}\ \boxed{A' - f(x)G} + \boxed{e_{large}}$$

$$\Downarrow$$

$$\boxed{s(I \otimes r)}\ \boxed{A' - f(x)G} + \boxed{e_{small}}$$

Further homomorphic evaluation

HLL Unbounded Algorithms [HLL24]

$\boxed{s} \boxed{A' - f(x)G} + \boxed{e_{large}}$

homomorphic encoding with *large* error

encryption secret

$\boxed{s} \boxed{A' - f(x)G} + \boxed{e_{small}}$

homomorphic encoding with *small* error

HLL Unbounded Algorithms [HLL24]

$$s\,(A' - f(x)G) + e_{large}$$

homomorphic encoding with
large error

In the context of
unbounded depth KP-ABE

$$s\,(A' - f(x)G) + e_{small}$$

homomorphic encoding with
small error

HLL Unbounded Algorithms [HLL24]

$$\boxed{s}\ \boxed{A' - f(x)G} + \boxed{e_{large}}$$

homomorphic encoding with *large* error

1. Noise Removal

2. Structure Restoration

$$\boxed{s}\ \boxed{A' - f(x)G} + \boxed{e_{small}}$$

homomorphic encoding with *small* error

Our Goal

$$\boxed{s(I \otimes r)}\ \boxed{A' - f(x)G} + \boxed{e_{large}}$$

randomised homomorphic encoding with *large* error

Randomised secret

$$\boxed{s(I \otimes r)}\ \boxed{A' - f(x)G} + \boxed{e_{small}}$$

randomised homomorphic encoding with *small* error

encryption secret

keygen secret

Our Goal

$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}}$ randomised homomorphic encoding with *large* error

How ?

$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{small}}$ randomised homomorphic encoding with *small* error

Our Goal

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}}$$

Take the HLL path?

1. Noise Removal
2. Structure Restoration

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{small}}$$

Our Approach

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow \boxed{\begin{array}{c}\text{HLL}\\ \text{Noise Removal}\end{array}} \longrightarrow \boxed{func(s_r)} - \boxed{s_r}\ \boxed{f(x)G}$$

randomised homomorphic
encoding with *large* error

Noiseless encoding

Our Approach

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow \boxed{\begin{array}{c}\text{HLL}\\ \text{Noise Removal}\end{array}} \longrightarrow \boxed{func(s_r)} - \boxed{s_r}\ \boxed{f(x)G}$$

Works if the secret is small

Our Approach

$$s_r \left(A' - f(x)G \right) + e_{large} \longrightarrow \text{HLL Noise Removal} \longrightarrow func(s_r) + s_r \, f(x)G$$

Undesirable mask

$func$: publicly computable

Our Approach

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow \boxed{\begin{matrix}\text{HLL}\\ \text{Noise Removal}\end{matrix}} \longrightarrow \underbrace{\boxed{func(s_r)}}_{\text{Undesirable mask}} - \boxed{s_r}\ \boxed{f(x)G}$$

Wanted: $\approx \boxed{s_r}\ \boxed{A'}$

Our Approach

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow$$

HLL
Noise Removal

$$\longrightarrow \boxed{func(s_r)} - \boxed{s_r}\ \boxed{f(x)G}$$

$sk : s_r$

$S_r = hct(s_r)$

fhe circular ct

HLL
Structure Restoration

Our Approach

$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow$ HLL Noise Removal $\longrightarrow \boxed{func(s_r)} - \boxed{s_r}\ \boxed{f(x)G}$

$S_r = hct(s_r)$

s_r

circular ct

HLL Structure Restoration

$\boxed{s_r(A - S_r \otimes G) + e}$

ct randomness

Our Approach

s_r $A' - f(x)G$ + e_{large} → HLL Noise Removal → $func(s_r)$ − s_r $f(x)G$

$S_r = hct(s_r)$

s_r

circular ct

$s_r(A - S_r \otimes G) + e$

encoding of S_r

HLL Structure Restoration

Our Approach

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow \boxed{\begin{array}{c}\text{HLL}\\ \text{Noise Removal}\end{array}} \longrightarrow \boxed{func(s_r)} - \boxed{s_r}\ \boxed{f(x)G}$$

$$S_r = hct(s_r)$$

s_r

$$\boxed{s_r(A - S_r \otimes G) + e}$$

$$\boxed{\begin{array}{c}\text{HLL}\\ \text{Structure Restoration}\end{array}} \longrightarrow \approx \boxed{s_r}\ \boxed{A_{func}} - \boxed{func(s_r)}$$

Our Approach

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow \boxed{\begin{matrix}\text{HLL}\\ \text{Noise Removal}\end{matrix}} \longrightarrow \boxed{func(s_r)} - \boxed{s_r}\ \boxed{f(x)G}$$

$$S_r = hct(s_r)$$

s_r

$$\boxed{s_r(A - S_r \otimes G) + e}$$

$$\boxed{\begin{matrix}\text{HLL}\\ \text{Structure Restoration}\end{matrix}} \longrightarrow \approx \boxed{s_r}\ \boxed{A_{func}} - \boxed{func(s_r)}$$

publicly computable

Our Approach

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow$$

HLL
Noise Removal

$$\longrightarrow \boxed{\cancel{func}(s_r)} - \boxed{s_r}\ \boxed{f(x)G}$$

$$S_r = hct(s_r) \qquad s_r$$

$$\boxed{s_r(A - S_r \otimes G) + e}$$

HLL
Structure Restoration

$$\longrightarrow\ \approx \boxed{s_r}\ \boxed{A_{func}} - \boxed{\cancel{func}(s_r)}$$

Our Approach

$$\boxed{s_r}\ \boxed{A' - f(x)G} + \boxed{e_{large}} \longrightarrow \boxed{\text{HLL Noise Removal}} \longrightarrow \boxed{\cancel{func}(s_r)} - \boxed{s_r}\ \boxed{f(x)G}$$

s_r

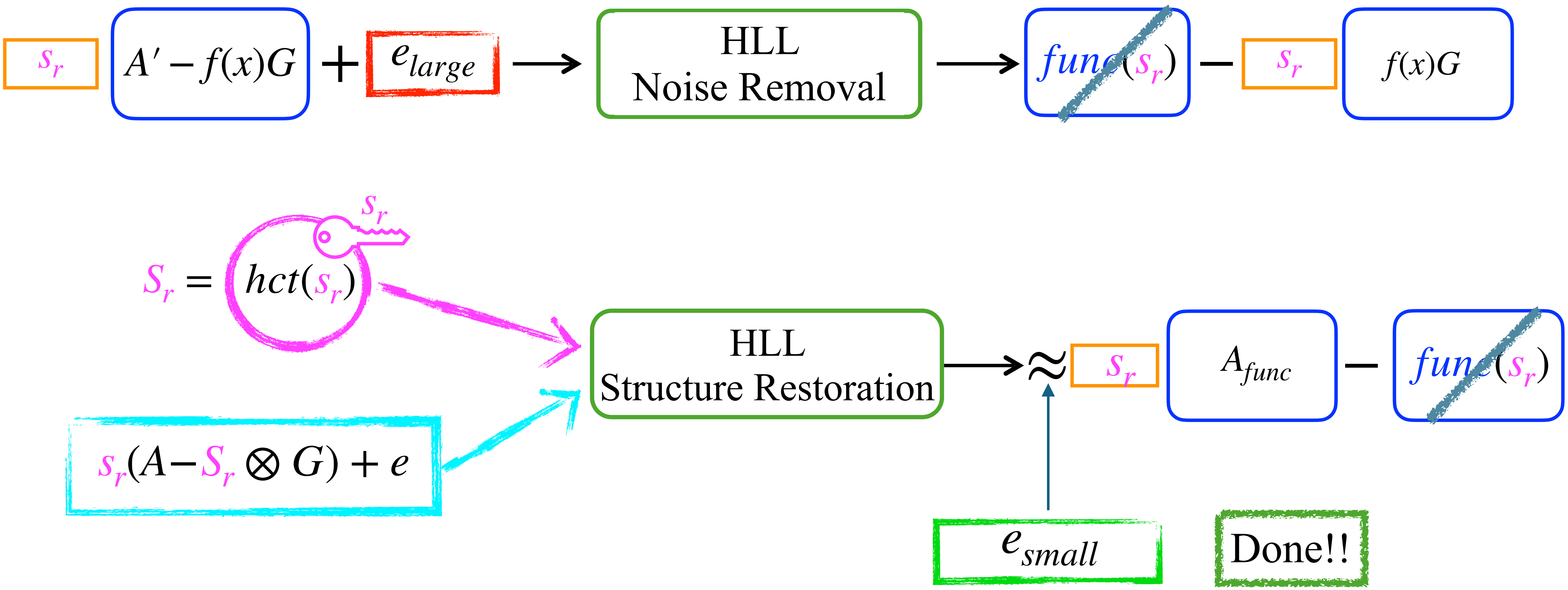
$$S_r = hct(s_r)$$

$$\boxed{s_r(A - S_r \otimes G) + e}$$

$$\boxed{\text{HLL Structure Restoration}} \longrightarrow \approx \boxed{s_r}\ \boxed{A_{func}} - \boxed{\cancel{func}(s_r)}$$

$$\boxed{e_{small}}$$

Our Approach



Our Approach

s : encryption randomness
r : keygen randomness

s_r

$S_r = hct(s_r)$

$s_r(A - S_r \otimes G) + e$

?

HLL
Structure Restoration

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$S_r = hct(s_r)$

$s_r(A - S_r \otimes G) + e$

?

HLL
Structure Restoration

! Can't give $S_r = hct_{s_r}(s_r)$ from encryption !

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$S_r = hct(s_r)$

$\boxed{s_r(A - S_r \otimes G) + e}$

HLL
Structure Restoration

Compute $S_r = hct_{s_r}(s_r)$ from $hct_s(s)$?

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$S_r = hct(s_r)$

$\boxed{s_r(A - S_r \otimes G) + e}$

?

$\boxed{\text{HLL Structure Restoration}}$

Compute $S_r = hct_{s_r}(s_r)$ from $hct_s(s)$?

! Can randomise the underlying message via homo. evaluation.
Randomising secret key is infeasible !! !

Our Approach

s : encryption randomness
r : keygen randomness

$S_r = hct(s_r)$

HLL
Structure Restoration

$s_r(A - S_r \otimes G) + e$

?

Compute $S_r = hct_{s_r}(s_r)$ from $hct_s(s)$?

! Randomising secret key is infeasible !! !

✘ Key Shrinking [BV11, BGV14]
Key Expansion [CM15,MW16]

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$S_r = hct(s_r)$

$s_r(A - S_r \otimes G) + e$

?

HLL
Structure Restoration

! Additional hurdles in computing encoding of S_r !

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$S_r = hct(s_r)$

HLL
Structure Restoration

?

$s_r(A - S_r \otimes G) + e$

! Additional hurdles in computing encoding of S_r !

✗ Can't apply tensors/evasive to randomise the enc randomness

Our Approach

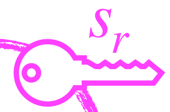
s : encryption randomness
r : keygen randomness

s_r

$$S_r = hct(s_r)$$

Our Approach

s : encryption randomness
r : keygen randomness



$$S_r = hct(s_r)$$

Obvious Fact:
FHE ciphertexts are good for computing on the *underlying messages*, not on the underlying keys

Recall

$$\left(\boxed{s} \boxed{A - x \otimes G} + \boxed{err} \right) \boxed{H_{f,x}}$$

Homomorphic Eval on attribute encoding

Recall

$$\left(\boxed{s} \; \boxed{A - x \otimes G} + \boxed{err} \right) \boxed{H_{f,x}}$$

requires f , x

Recall

$$\left(\boxed{s} \;\; \boxed{A - \boxed{hct(x)} \otimes G} + \boxed{err} \right)$$

Want: To hide x .

Recall

$$\left(\boxed{s} \boxed{A - \boxed{hct(x)} \otimes G} + \boxed{err} \right)$$

Want: To hide x .

-Use the GSW13 FHE scheme

Recall

$$X = hct(x)$$

$$\left(\boxed{s} \boxed{A - \boxed{X} \otimes G} + \boxed{err} \right)$$

Want: To hide x and compute f on x !

Recall

$X = hct(x)$

$$\left(\boxed{s} \; \boxed{A - \boxed{X} \otimes G} + \boxed{err} \right) \boxed{H_{f,X}}$$

requires f , X

Want: To hide x and compute f on x !

$H_{f,x}$ only gives $f(hct(x))$

Automatic Decryption [BTVW17]

s ← FHE sk

encryption randomness ← $\left(\boxed{s}\; \boxed{A - \boxed{X} \otimes G} + \boxed{err} \right)$

$$X = hct(x)$$

Want: To hide x and compute f on x !

Automatic Decryption [BTVW17]

s ← FHE sk

encryption randomness ← s

$$s\left(\boxed{A - \boxed{X} \otimes G} + err\right) \quad \boxed{H[FHE.Eval_f, X]}$$

$X = hct(x)$

requires $FHE.Eval_f$, X

Automatic Decryption [BTVW17]

$\color{red}{s} \longleftarrow$ FHE sk

$$\approx \boxed{\color{red}{s}}\ \boxed{A'} - \boxed{\color{red}{s}}\ \boxed{\boxed{\color{red}{hct(f(x))}} \cdot G}$$

FHE Eval$(f, \color{red}{hct(x)})$

Automatic Decryption [BTVW17]

$$\approx \boxed{s} \boxed{A'} - \boxed{f(x) \cdot G}$$

Computed f on $\boxed{hct(x)}$!

Automatic Decryption [BTVW17]

$$\approx \boxed{s} \boxed{A'} - \boxed{f(x) \cdot G}$$

Computed f on $\boxed{hct(x)}$!

Also the key idea in

HLL
Structure Restoration

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$S_r = hct(s_r)$

t

$T = \boxed{hct(s)} \longrightarrow$ Homom. Evaluation + Automatic Dec

Our Approach

s : encryption randomness

r : keygen randomness

s_r

$S_r = hct(s_r)$

t

$T = \boxed{hct(s)}$

$\boxed{t(A - T \otimes G) + e}$

$\longrightarrow$ Homom. Evaluation + Automatic Dec

Our Approach

s : encryption randomness
r : keygen randomness

$S_r = hct(s_r)$

$T = \boxed{hct(s)}$

$\boxed{t(A - T \otimes G) + e}$

$\longrightarrow$ Homom. Evaluation + Automatic Dec

$F[r]$ ← By keygen

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$S_r = hct(s_r)$

t

$T = \boxed{hct(s)}$

$\boxed{t(A - T \otimes G) + e}$

$\longrightarrow$ Homom. Evaluation + Automatic Dec $\longrightarrow$ $F[r](s) = S_r$

$\uparrow$

$F[r]$ $\leftarrow$ By keygen

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$$S_r = hct(s_r)$$

t

$$T = \boxed{hct(s)}$$

$$\boxed{t(A - T \otimes G) + e}$$

Homom. Evaluation
+ Automatic Dec

$F[r]$ By keygen

$S_r + mask$

Our Approach

s : encryption randomness
r : keygen randomness

s_r

$$S_r = hct(s_r)$$

t

$$T = \boxed{hct(s)}$$

$$\boxed{t(A - T \otimes G) + e}$$

Homom. Evaluation
+ Automatic Dec

$F[r]$ By keygen

$$S_r + mask$$

Use Evasive LWE to cancel the mask!

Summary

- We construct the first CP-ABE scheme supporting unbounded depth circuits from lattices. Security proof requires a new assumption 'circular tensor LWE' along with LWE and Evasive LWE.

Summary

- We construct the first CP-ABE scheme supporting unbounded depth circuits from lattices. Security proof requires a new assumption 'circular tensor LWE' along with LWE and Evasive LWE.

- We construct the first ABE scheme for NL and all Turing Machines from lattices.