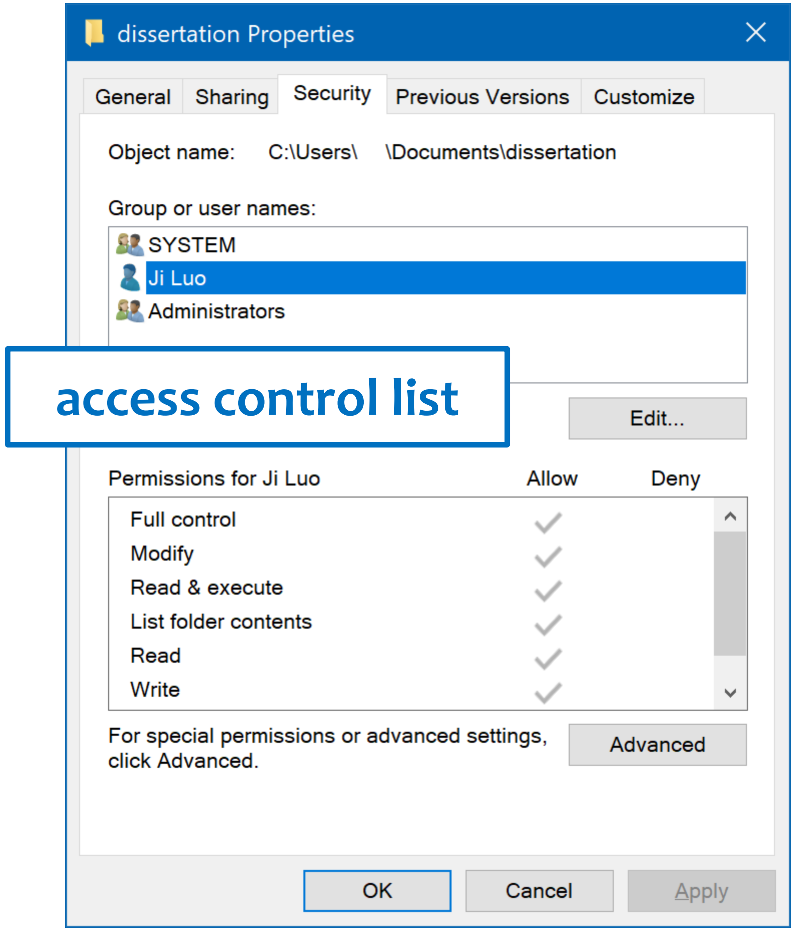
New Frontiers of Attribute-Based Encryption via a General Paradigm and More

罗辑 (Ji Luo)

based on joint work with
Yao-Ching Hsieh, Aayush Jain, Hanjun Li, Rachel Lin

Attribute-Based Encryption [SW05,GPSW06]

= access control, enforced by cryptography



dissertation Properties

Object name: C:\Users\ \Documents\dissertation

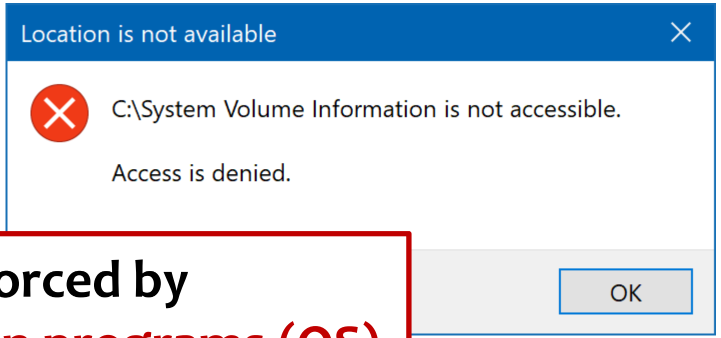
Group or user names:

- SYSTEM
- Ji Luo
- Administrators

Permissions for Ji Luo	Allow	Deny
Full control	✓	
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	

For special permissions or advanced settings, click Advanced.

access control list



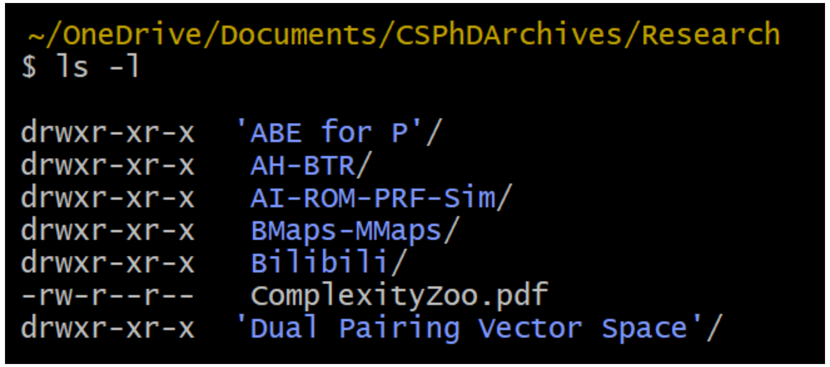
Location is not available

C:\System Volume Information is not accessible.

Access is denied.

only enforced by
checks in programs (OS)

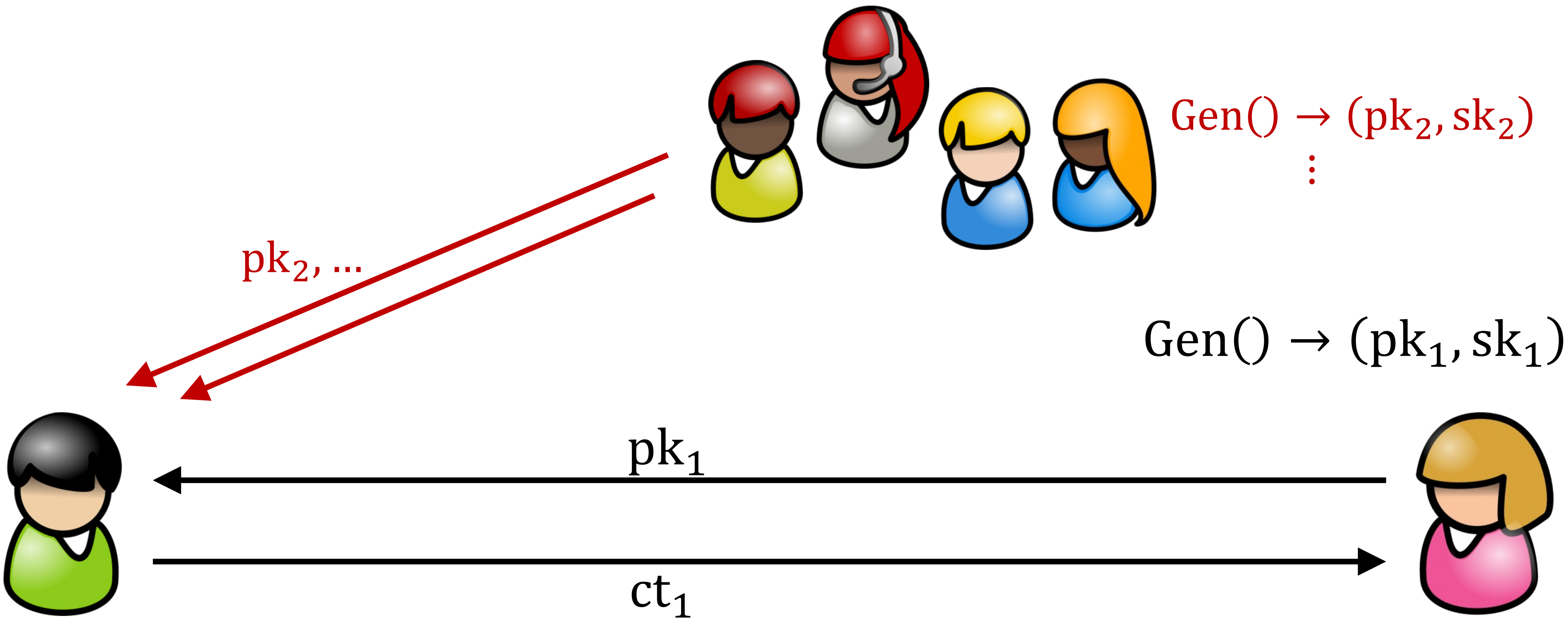
rwx rwx rwx
= permission bits



```
~/OneDrive/Documents/CSPhDArchives/Research
$ ls -l

drwxr-xr-x  'ABE for P'/
drwxr-xr-x   AH-BTR/
drwxr-xr-x   AI-ROM-PRF-Sim/
drwxr-xr-x   BMaps-MMaps/
drwxr-xr-x   Bilibili/
-rw-r--r--   ComplexityZoo.pdf
drwxr-xr-x  'Dual Pairing Vector Space'/
```

What about PKE for Access Control?



$\mathrm{Gen}() \rightarrow (\mathrm{pk}_2, \mathrm{sk}_2)$

$\vdots$

$\mathrm{pk}_2, \ldots$

$\mathrm{Gen}() \rightarrow (\mathrm{pk}_1, \mathrm{sk}_1)$

pk_1

ct_1

$\mathrm{Enc}(\mathrm{pk}_1, \mu) \rightarrow \mathrm{ct}_1$

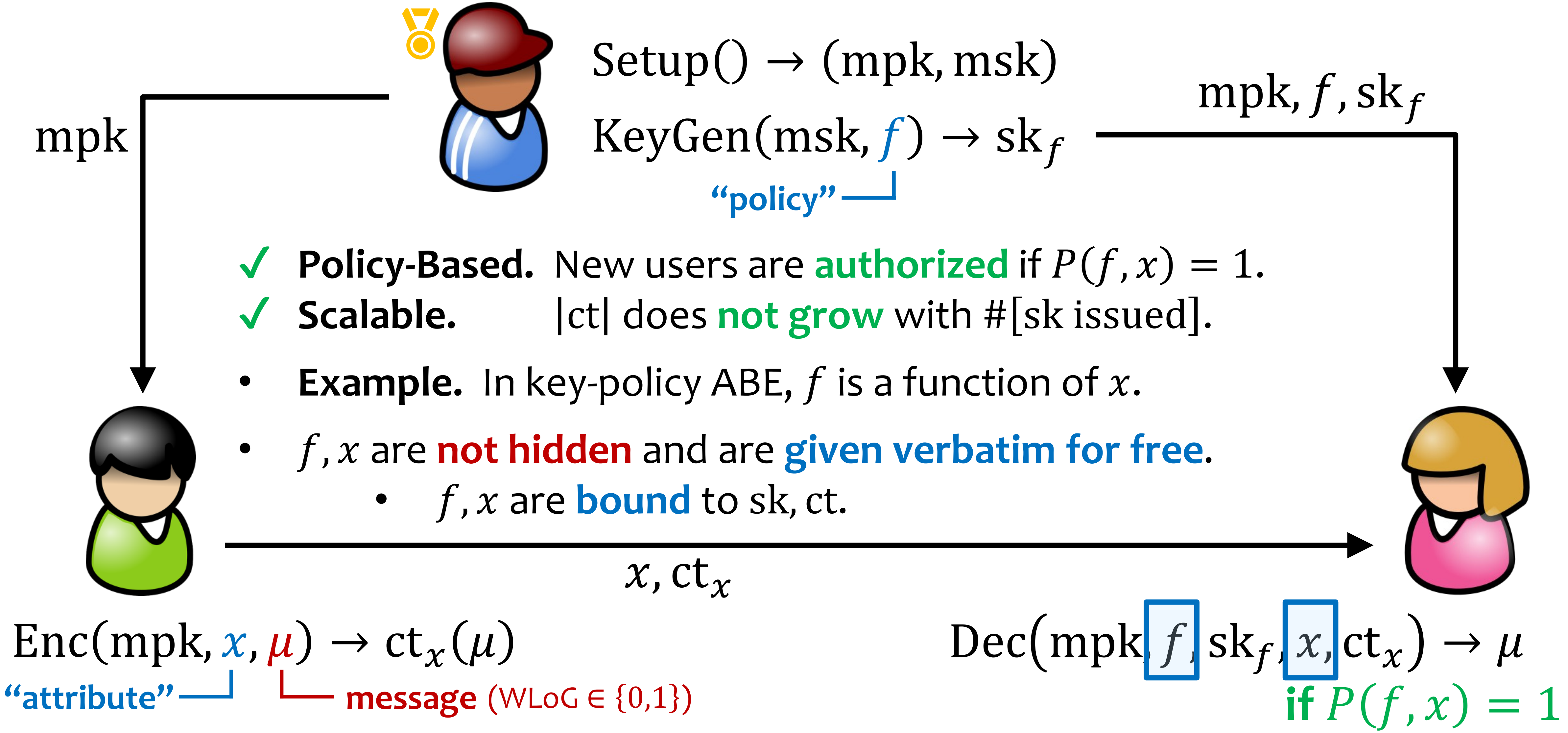
$\mathrm{Enc}(\mathrm{pk}_3, \mu) \rightarrow \mathrm{ct}_3$

$\vdots$

✗ new users always **un**authorized	✗ individual-based, not policy-based
✗ $\lvert\mathrm{ct}\rvert \propto \#[\mathrm{sk\ that\ can\ decrypt}]$	✗ not scalable

OS Analogy. New users (e.g., given suitable group membership) might be authorized.

Syntax and Correctness of ABE



$\mathsf{Setup}() \to (\mathsf{mpk}, \mathsf{msk})$

$\mathsf{KeyGen}(\mathsf{msk}, f) \to \mathsf{sk}_f$

"policy"

mpk

$\mathsf{mpk}, f, \mathsf{sk}_f$

✓ **Policy-Based.** New users are **authorized** if $P(f, x) = 1$.

✓ **Scalable.** $|\mathsf{ct}|$ does **not grow** with $\#[\mathsf{sk}\ \mathsf{issued}]$.

- **Example.** In key-policy ABE, f is a function of x.
- f, x are **not hidden** and are **given verbatim for free**.
  - f, x are **bound** to sk, ct.

x, ct_x

$\mathsf{Enc}(\mathsf{mpk}, x, \mu) \to \mathsf{ct}_x(\mu)$

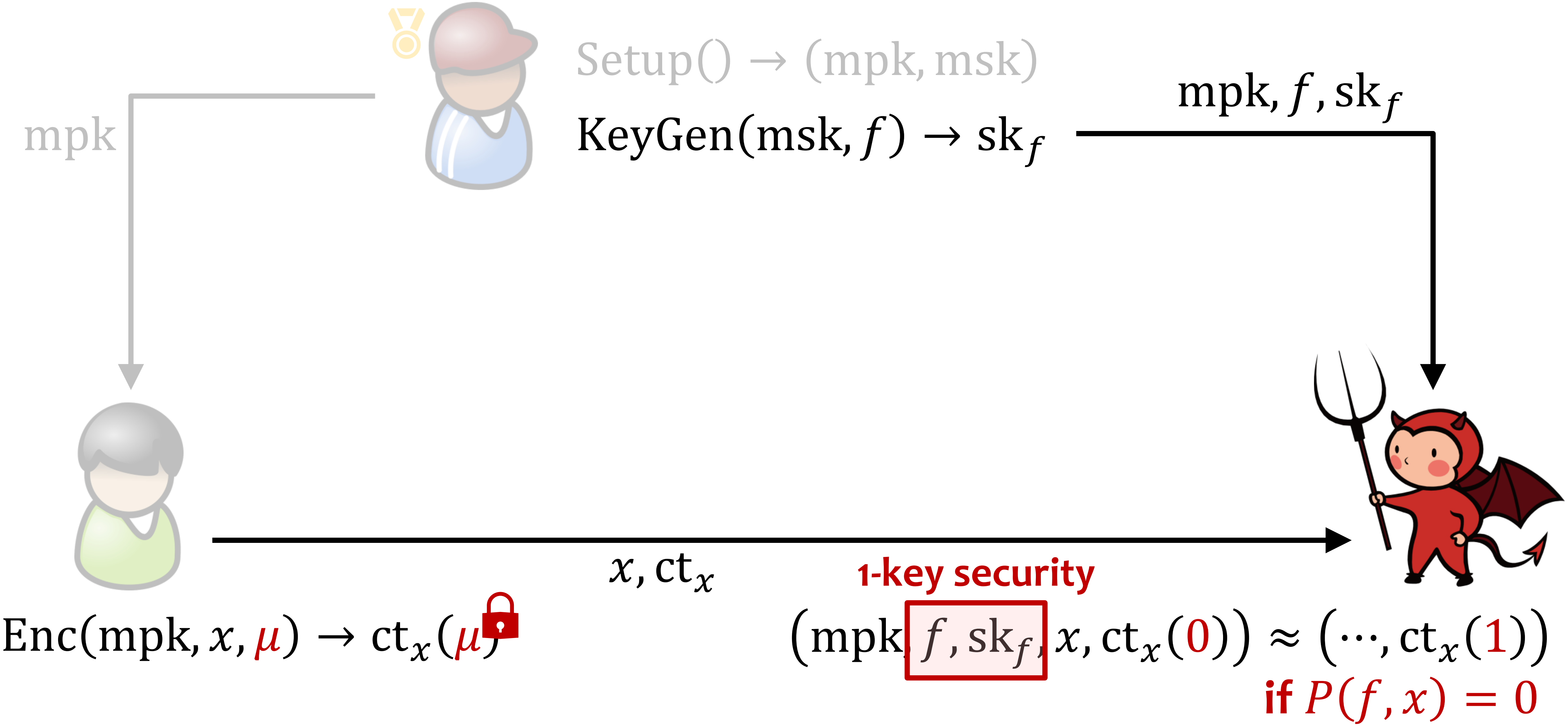
"attribute"

message (WLoG $\in \{0,1\}$)

$\mathsf{Dec}\big(\mathsf{mpk}, \boxed{f}, \mathsf{sk}_f, \boxed{x}, \mathsf{ct}_x\big) \to \mu$

if $P(f, x) = 1$

Security of ABE



$\text{Setup}() \to (\text{mpk}, \text{msk})$

$\text{KeyGen}(\text{msk}, f) \to \text{sk}_f$

mpk

$\text{mpk}, f, \text{sk}_f$

$\text{Enc}(\text{mpk}, x, \mu) \to \text{ct}_x(\mu)$ 🔒

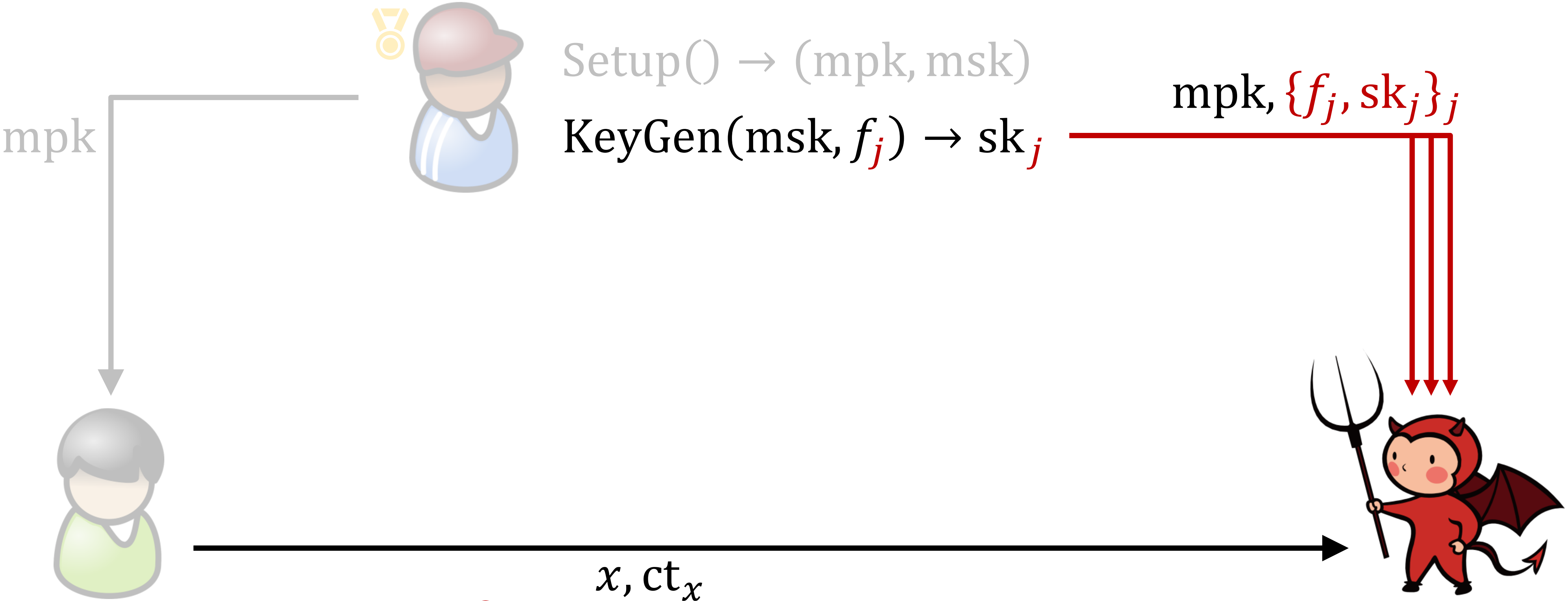
x, ct_x

1-key security

$$\left(\text{mpk}, \boxed{f, \text{sk}_f}, x, \text{ct}_x(0)\right) \approx \left(\cdots, \text{ct}_x(1)\right)$$

if $P(f, x) = 0$

Security of ABE – Collusion Resistance



$\mathsf{Setup}() \to (\mathsf{mpk}, \mathsf{msk})$

$\mathsf{KeyGen}(\mathsf{msk}, f_j) \to \mathsf{sk}_j$

$\mathsf{mpk}, \{f_j, \mathsf{sk}_j\}_j$

mpk

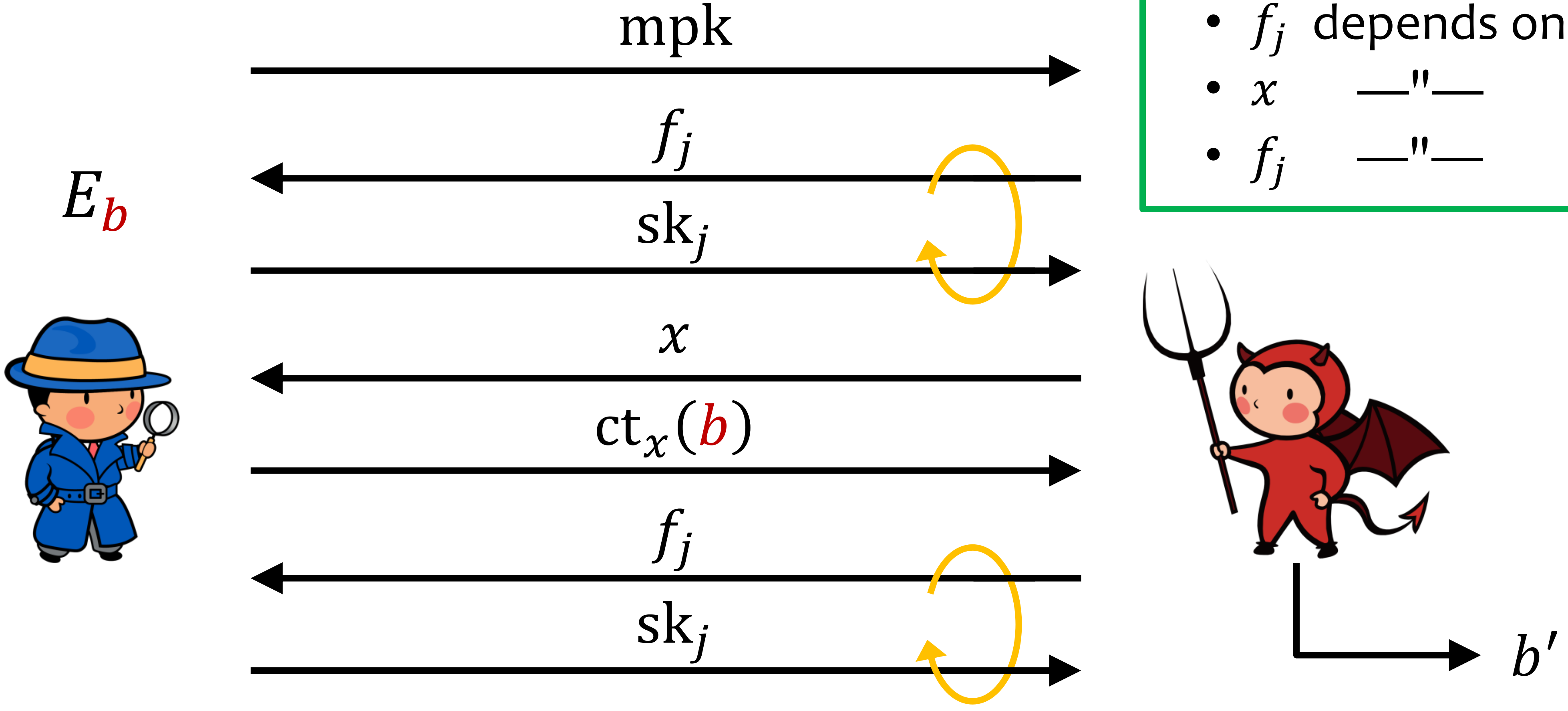
x, ct_x

$\mathsf{Enc}(\mathsf{mpk}, x, \mu) \to \mathsf{ct}_x(\mu)$ 🔒

$\left(\mathsf{mpk}, \{f_j, \mathsf{sk}_j\}_j, x, \mathsf{ct}_x(0)\right) \approx \left(\cdots, \mathsf{ct}_x(1)\right)$

if $P(f_j, x) = 0$ **for all** j

Security of ABE – Formal Definition

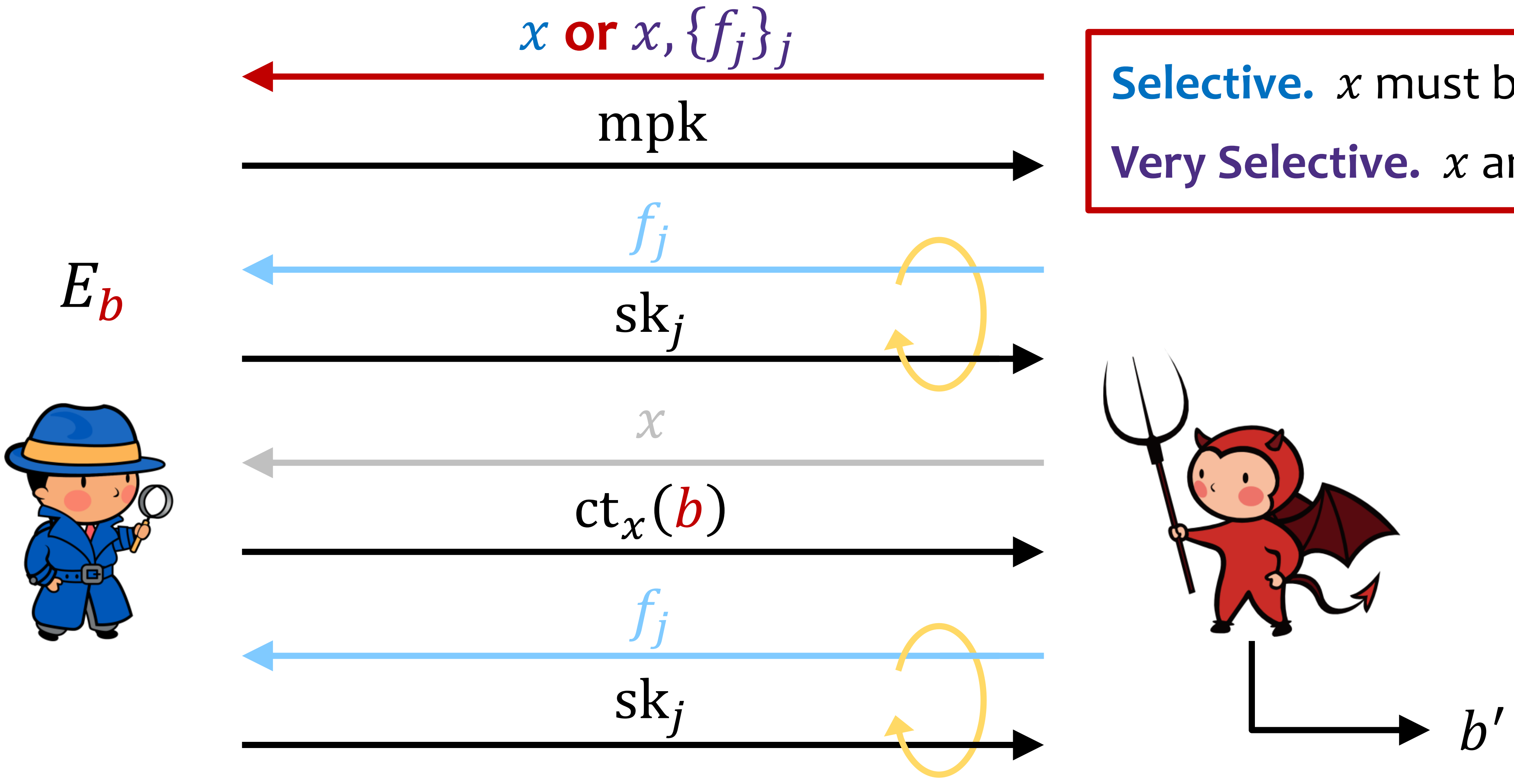


Adaptive Security.

- f_j depends on $\mathrm{mpk}, \mathrm{sk}_{<j}$
- x —"— $\mathrm{mpk}, \mathrm{sk}_{<J_1}$
- f_j —"— $\mathrm{mpk}, \mathrm{sk}_{<j}, \mathrm{ct}$

Security. $E_0 \approx E_1$ under the constraint of $P(f_j, x) = 0$ for all j.

Security of ABE – Weaker Notions

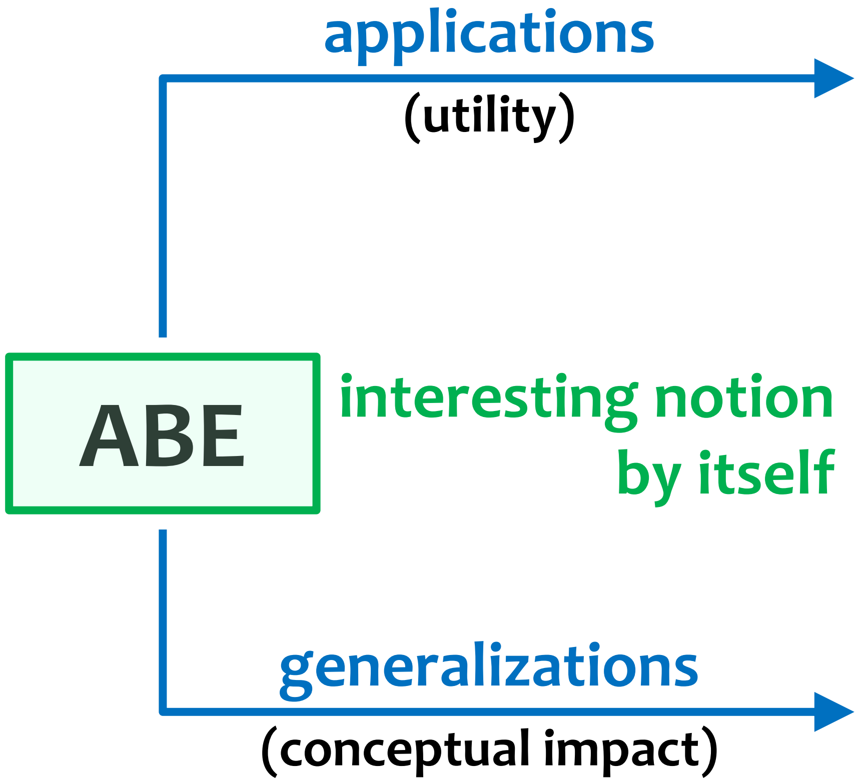


Selective. x must be chosen first.

Very Selective. x and all f_j's —"—.

Security. $E_0 \approx E_1$ under the constraint of $P(f_j, x) = 0$ for all j.

Why Study ABE?



applications

(utility)

ABE

interesting notion by itself

generalizations

(conceptual impact)

- access control

  audit logs [GPSW06] medical records [APGLPR11]

  private key distribution in cloud [Cloudflare17]

- verifiable delegated computation [PRV11]
- non-trivial witness encryption [BJKPW17]

- decentralization [C07,AYY22,HLWW22]

  multi-authority/input or registered

- stronger functionality [SBCSP07,BW07,BSW11]

  predicate / functional encryption

  ↑ connection to obfuscation [GGHRSW13,BV15,AJ15]

Pursuit of **Ends** – Desirata of ABE

Expressive. Supports rich class of policies.

circuits > formulae

RAM > TM > DFA

Succinct. Short mpk, sk, ct.

Recall $\mathrm{Dec}(\mathrm{mpk}, f, \boxed{\mathrm{sk}_f}, x, \boxed{\mathrm{ct}_x})$.

does not have to fully encode f, x

succinct

sk, ct **bound** to f, x (**not hiding**)

- think **hash / signature**
- possible that $|\mathrm{sk}| < |f|$, $|\mathrm{ct}| < |x|$

affects baseline

Efficient. Fast Dec (and Setup, KeyGen, Enc).

T_P = baseline for T_{Dec}

These objectives are intertwined 🪢 !

Strong Security. Adaptive > selective > very selective.

Weak Assumptions. Falsifiable > non-falsifiable.

Static > adversary-dependent (q-type).

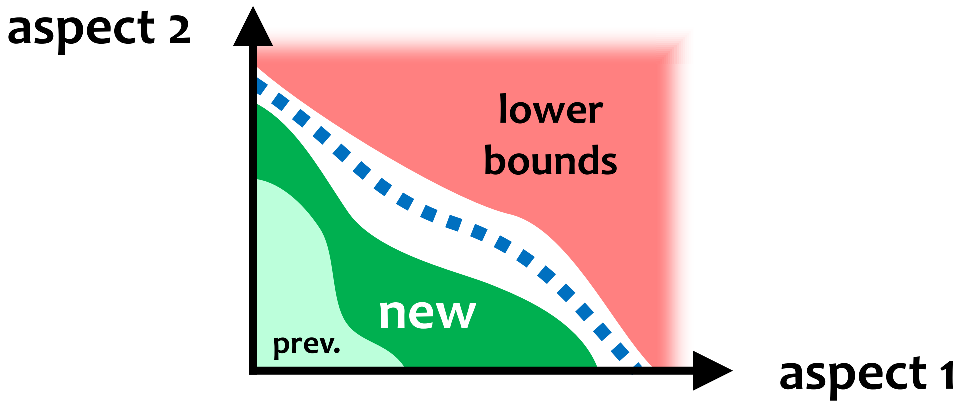
conceivable trade-off

- **same** construction
- proofs of **different** assumption ⇒ security

Pursuit of **Ends** – Multi-Objective Optimization

Expressive

Succinct



Goal. Characterize **curve of Pareto optimality**.

Efficient

Push the Frontier. Construct new schemes.

- **better** than previous in **at least one** aspect

  (wishful) better in **many** aspects

- some aspects are more prioritized

  (expressive, succinct)

Strong Security

Weak Assumptions

Encircle the Boundary. Prove trade-off lower bounds.

Pursuit of **Means**

Designing ABE schemes is… **not easy**!

general paradigm / framework ?

WANTED

- **modular** – redistribute complexities
- **powerful** – new results
- **versatile** – flexible assumptions

Previously…

dual system encryption [W09] + refinements

- pair encoding [A14]
- predicate encoding [W14]

❏ **born for adaptive security**
⚠ **only instantiated with pairing**
⚠ **heavy in algebra details**

two-to-one recoding [GVW13]
key-homomorphic encryption [BGGHNSVV14]

⚠ **new results only from lattices**
⚠ **too few instantiations**

Organization

Part I. General Paradigm (ABE $\Leftarrow$ IPFE $\circ$ Garbling)

- 4 instantiations
  [LL20a,LL20b,LLL22,HLL24]

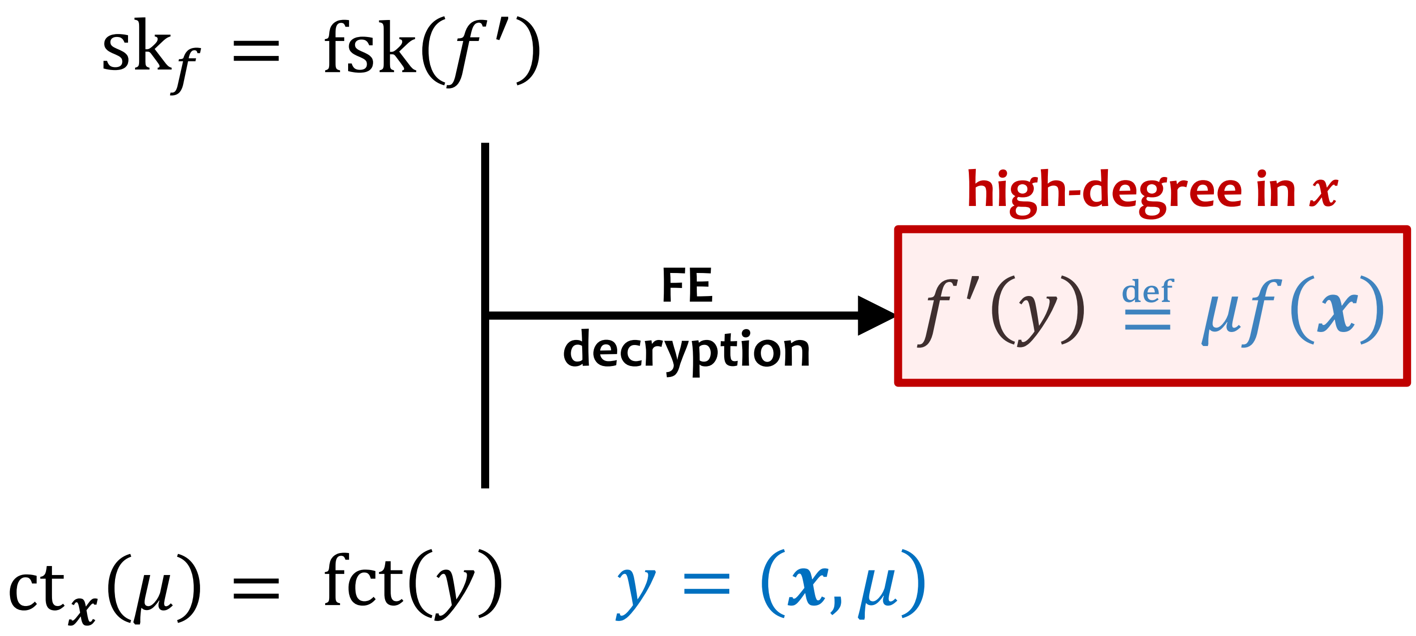
Part II. More

- ABE for circuits of unbounded depth from lattices
  [HLL23]
- first systematic study of
  optimal succinctness and efficiency for ABE
  [JLL23,L24]

Part I. **General Paradigm**

Somewhat technical, but less so than
the sum of all those separate talks.

ABE $\Leftarrow$ Functional Encryption



$\mathrm{sk}_f = \mathrm{fsk}(f')$

FE decryption

high-degree in $\boldsymbol{x}$

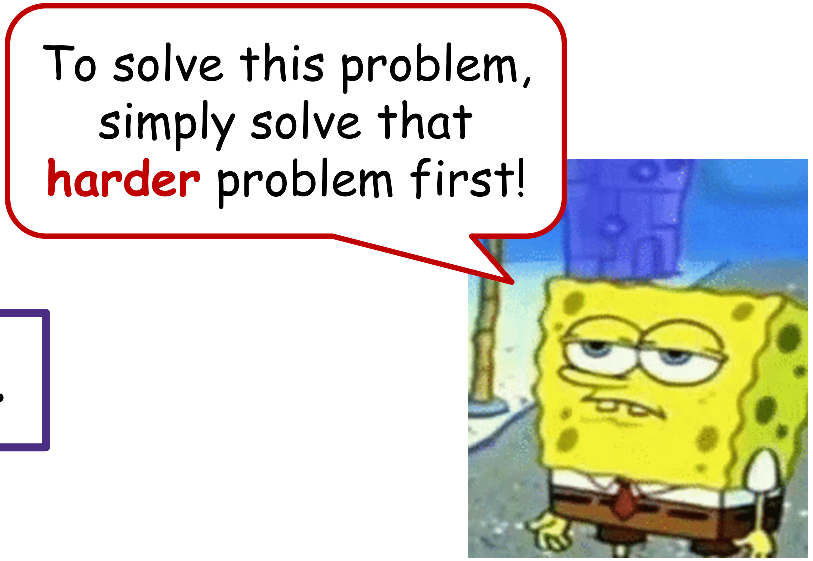
$f'(y) \overset{\text{def}}{=} \mu f(\boldsymbol{x})$

$\mathrm{ct}_{\boldsymbol{x}}(\mu) = \mathrm{fct}(y) \qquad y = (\boldsymbol{x}, \mu)$

Idea.

- Decompose into two phases (low-degree + high-degree).
- Use **FE** on **low-degree** only.

FE Security. Hides everything about y beyond $f'(y)$.


To solve this problem, simply solve that **harder** problem first!

Linear Garbling (Roughly) [Y82,Y86,AIK11,IW14]

1. $\mathrm{Garble}(f, \delta) \to (L_1, \dots, L_m)$
   - affine (low-degree) functions of x (**label functions**)
   - coefficients ($\boldsymbol{L}$'s) contain δ, randomness

Protect δ, randomness? Protect this process!

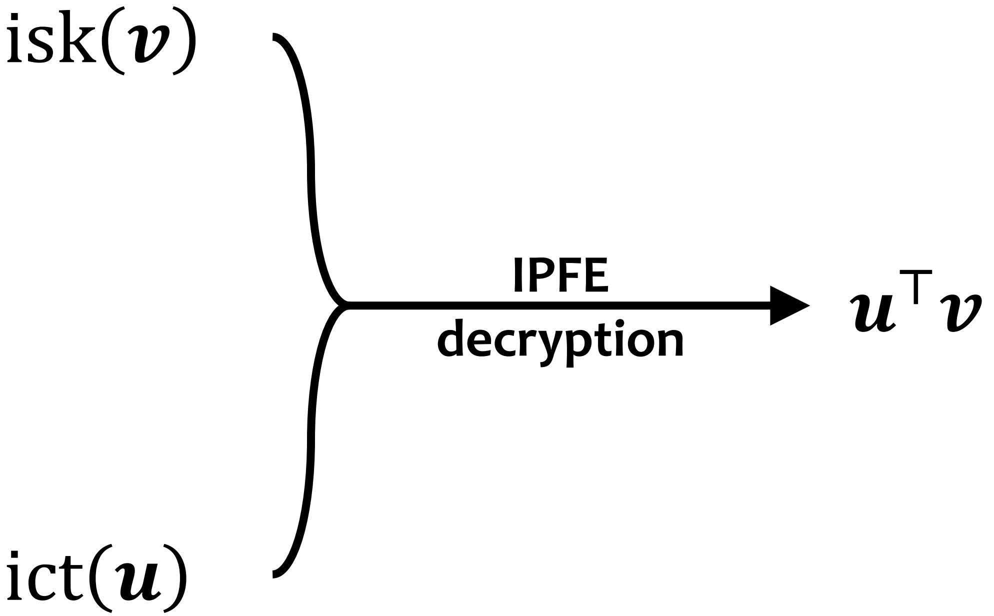
2. $\ell_1 = L_1(\boldsymbol{x}) = \boxed{\langle(1, \boldsymbol{x}), \boldsymbol{L}_1\rangle,}$ $\quad \dots, \quad$ $\ell_m = L_m(\boldsymbol{x}) = \boxed{\langle(1, \boldsymbol{x}), \boldsymbol{L}_m\rangle}$
   - **labels**

not hidden

3. $\mathrm{Eval}(f, \boxed{\boldsymbol{x},} \ell_1, \dots, \ell_m) \to \delta f(\boldsymbol{x})$
   - high-degree in $\boldsymbol{x}$

"ℓ's reveal nothing about δ beyond $\delta f(\boldsymbol{x})$"

Inner-Product FE (Roughly) [ABDP15]



$\mathrm{isk}(\boldsymbol{v})$

IPFE

decryption

$\boldsymbol{u}^\top \boldsymbol{v}$

$\mathrm{ict}(\boldsymbol{u})$

“isk, ict’s hide everything beyond the inner products”

ABE $\Leftarrow$ IPFE ∘ Garbling

$$\mathrm{sk}_f \;=\; \mathrm{isk}(\boldsymbol{L}_1)\,, \ldots, \mathrm{isk}(\boldsymbol{L}_m) \quad \text{(for } \delta\text{)}$$

$$\xrightarrow[\text{decryption}]{\text{IPFE}} \widehat{\delta f(\boldsymbol{x})} = L_1(\boldsymbol{x}), \ldots, L_m(\boldsymbol{x}) \xrightarrow[\text{evaluation}]{\text{garbling}} \delta f(\boldsymbol{x})$$

$$\mathrm{ct}_{\boldsymbol{x}}(\mu) \;=\; \mathrm{ict}(1, \boldsymbol{x}), \boxed{\delta \oplus \mu}$$

remove OTP
when $f(\boldsymbol{x}) = 1$

Composition of Security. (wishful)

- IPFE – only labels revealed
- garbling – only $\delta f(\boldsymbol{x})$ revealed
- δ is OTP for μ when $f(\boldsymbol{x}) = 0$

formalize properties that compose well

Security composition is tricky and sensitive to formalism.

Pairing Groups

- G_1, G_2, G_T groups of order p (prime)

  $G_i = \langle g_i \rangle$, additive, $[\![a]\!]_i \overset{\text{def}}{=} a g_i$

- $e: G_1 \times G_2 \to G_\mathrm{T}$ non-degenerate bilinear map

  $e(ag_1, bg_2) = abg_\mathrm{T}$, $[\![a]\!]_1 [\![b]\!]_2 = [\![ab]\!]_\mathrm{T}$

What is it good for cryptography?

Pairing = one-time, controlled multiplication.

✓ **Easy** $([\![a]\!]_1, b) \mapsto [\![ab]\!]_1$ and $([\![a]\!]_1, [\![b]\!]_2) \mapsto [\![ab]\!]_\mathrm{T}$.

DDH. $[\![a, b, ab]\!]_1 \approx [\![a, b, c]\!]_1$ for $a, b, c \overset{\$}{\leftarrow} \mathbb{Z}_p$.

✗ **Hard** $([\![a]\!]_1, [\![b]\!]_1) \mapsto [\![ab]\!]_\mathrm{T}$.

- Provides ***some*** protection for x in $[\![x]\!]_i$.
- Builds **IPFE** (*full* protection).

IPFE in [LL20a]

Recall. Garbling Eval after IPFE Dec.

Pairing-Based.

only linear operations with $[\![\cdot]\!]_{\mathrm{T}}$?

$$\mathrm{Dec}\big(\mathrm{isk}([\![\boldsymbol{v}]\!]_2),\ \mathrm{ict}([\![\boldsymbol{u}]\!]_1)\big) = [\![\boldsymbol{u}^\top \boldsymbol{v}]\!]_{\mathrm{T}}$$

Fact. Such IPFE can be built from k-Lin (**standard,** static assumption). [ALS16,W17,LV16,L17]

Function-Hiding.* (hides $\boldsymbol{u}, \boldsymbol{v}$)

$$\big(\mathrm{impk}, \{\mathrm{isk}(\boldsymbol{v}_{j0})\}_j, \{\mathrm{ict}(\boldsymbol{u}_{i0})\}_i\big) \approx \big(\mathrm{impk}, \{\mathrm{isk}(\boldsymbol{v}_{j1})\}_j, \{\mathrm{ict}(\boldsymbol{u}_{i1})\}_i\big)$$

Can compute $I \times J$ inner products $\boldsymbol{u}_{i?}^\top \boldsymbol{v}_{j?}$.

if $\boldsymbol{u}_{i0}^\top \boldsymbol{v}_{j0} = \boldsymbol{u}_{i1}^\top \boldsymbol{v}_{j1}$ for all i, j.

* not the full story, but good enough for now

Garbling in [LL20a]

More Linear Properties.

1. $\mathrm{Garble}(f, \delta; \boldsymbol{r}) \to (\boldsymbol{L}_1, \dots, \boldsymbol{L}_m)$
   linear in $(\delta, \boldsymbol{r})$
2. $\ell_j = \langle (1, \boldsymbol{x}), \boldsymbol{L}_j \rangle$
3. $\mathrm{Eval}(f, \boldsymbol{x}, \ell_1, \dots, \ell_m)$
   linear in $(\ell_1, \dots, \ell_m)$

Fact. Such garbling for arithmetic branching programs (ABP) exists. [IK00,IK02,IW14]

ABP = determinant of certain matrices

Security.* (distribution of $\ell_1, \dots, \ell_m$)

Point. This leads to **localized** label simulation.

1. $\ell_2, \dots, \ell_m$ are jointly random.
2. ℓ_1 is uniquely determined by $f, x, \delta f(\boldsymbol{x}), \ell_2, \dots, \ell_m$ due to **evaluation correctness**, i.e.,
   $$\mathrm{Eval}(f, x, \ell_1, \ell_2, \dots, \ell_m) = \delta f(\boldsymbol{x}),$$
   a linear constraint on ℓ_1.

* not the full story, but good enough for now

Instantiating the Paradigm in [LL20a]

$$\mathrm{sk}_f = \mathrm{isk}(\llbracket \boldsymbol{L}_1 \rrbracket_2), \ldots, \mathrm{isk}(\llbracket \boldsymbol{L}_m \rrbracket_2)$$

$$\xrightarrow[\textbf{decryption}]{\textbf{IPFE}} \widehat{\delta f(\boldsymbol{x})} = \llbracket \ell_1 \rrbracket_{\mathrm{T}}, \ldots, \llbracket \ell_m \rrbracket_{\mathrm{T}} \xrightarrow[\textbf{linear in } \ell\textbf{'s}]{\text{Eval}} \llbracket \delta f(\boldsymbol{x}) \rrbracket_{\mathrm{T}}$$

$$\mathrm{ct}_{\boldsymbol{x}}(\mu) = \mathrm{ict}(\llbracket 1, \boldsymbol{x} \rrbracket_1)$$

Selective Security in [LL20a]

✓ independent of δ

ict ($1, \boldsymbol{x}$)	IPFE $\approx$	ict ($1, \boldsymbol{x}$)	garbling $\equiv$	ict ($1, \boldsymbol{x}$)
isk ($\boldsymbol{L}_1$)		isk ($\ell_1, \mathbf{0}$)		isk ($\ell_1, \mathbf{0}$)
isk ($\boldsymbol{L}_2$)		isk ($\ell_2, \mathbf{0}$)		isk ($\ell_2, \mathbf{0}$)
$\vdots$		$\vdots$		$\vdots$
isk ($\boldsymbol{L}_m$)		isk ($\ell_m, \mathbf{0}$)		isk ($\ell_m, \mathbf{0}$)
		$\ell_j = \langle (1, \boldsymbol{x}), \boldsymbol{L}_j \rangle$		$\ell_2, \ldots, \ell_m = \$$

solve

$$\mathrm{Eval}(f, \boldsymbol{x}, \ell_1, \ell_2, \ldots, \ell_m) = \delta f(\boldsymbol{x})$$

for ℓ_1 $= 0$ **(constraint)**

Problem with Adaptive Security in [LL20a]

x not known at this point ↓

isk ($\boldsymbol{L}_1$)	isk ($\ell_1, \mathbf{0}$)	isk ($\ell_1, \mathbf{0}$)
isk ($\boldsymbol{L}_2$)	isk ($\ell_2, \mathbf{0}$)	isk ($\ell_2, \mathbf{0}$)
$\vdots$	$\vdots$	$\vdots$
isk ($\boldsymbol{L}_m$)	isk ($\ell_m, \mathbf{0}$)	isk ($\ell_m, \mathbf{0}$)
$\xrightarrow{\boldsymbol{x}}$	$\ell_j = \langle (1, \boldsymbol{x}), \boldsymbol{L}_j \rangle$	$\ell_2, \dots, \ell_m = \$$
ict ($1, \boldsymbol{x}$)	ict ($1, \boldsymbol{x}$)	ict ($1, \boldsymbol{x}$)

cannot solve equation dependent on x →

solve

$$\boxed{\mathrm{Eval}(f, \boldsymbol{x}, \ell_1, \ell_2, \dots, \ell_m) = 0}$$

for ℓ_1

Fixing Adaptive Security in [LL20a]

$\mathrm{isk}\,(\ \boldsymbol{L}_1 \quad 0\ )$

$\mathrm{isk}\,(\ \boldsymbol{L}_2 \quad 0\ )$

$\vdots$

$\mathrm{isk}\,(\ \boldsymbol{L}_m \quad 0\ )$

$\xrightarrow{\boldsymbol{x}}$

$\mathrm{ict}\,(\ 1, \boldsymbol{x} \quad 0\ )$

many steps *

$\approx$

✓ can be generated
✓ independent of δ

$\mathrm{isk}\,(\ 0, \mathbf{0} \quad 1\ )$

$\mathrm{isk}\,(\ \ell_2, \mathbf{0} \quad 0\ )$

$\vdots$

$\mathrm{isk}\,(\ \ell_m, \mathbf{0} \quad 0\ )$

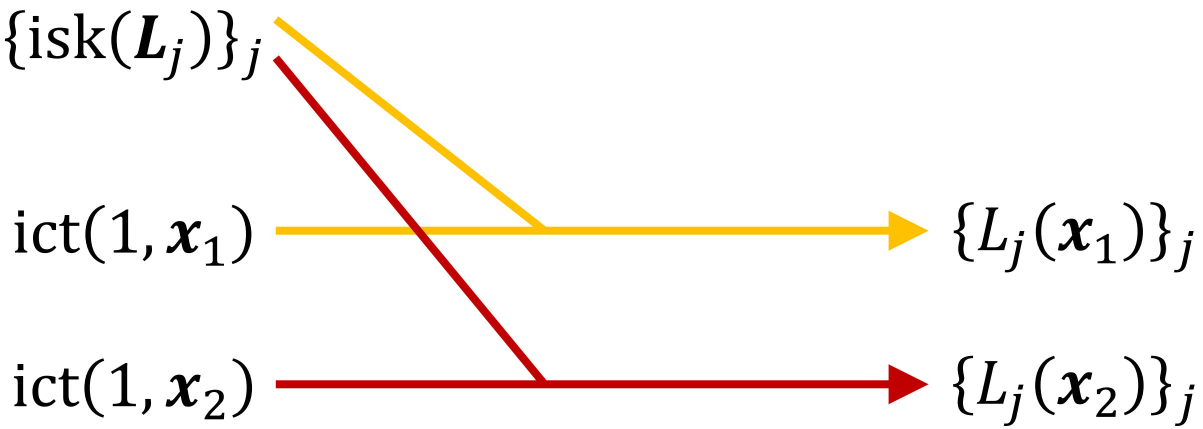
$\ell_2, \dots, \ell_m = \$$

$\mathrm{ict}\,(\ 1, \boldsymbol{x} \quad \ell_1\ )$

solve
$\mathrm{Eval}(f, \boldsymbol{x}, \ell_1, \ell_2, \dots, \ell_m) = 0$
for ℓ_1

* untold part of garbling security

Multi-Ciphertext Security in [LL20a]

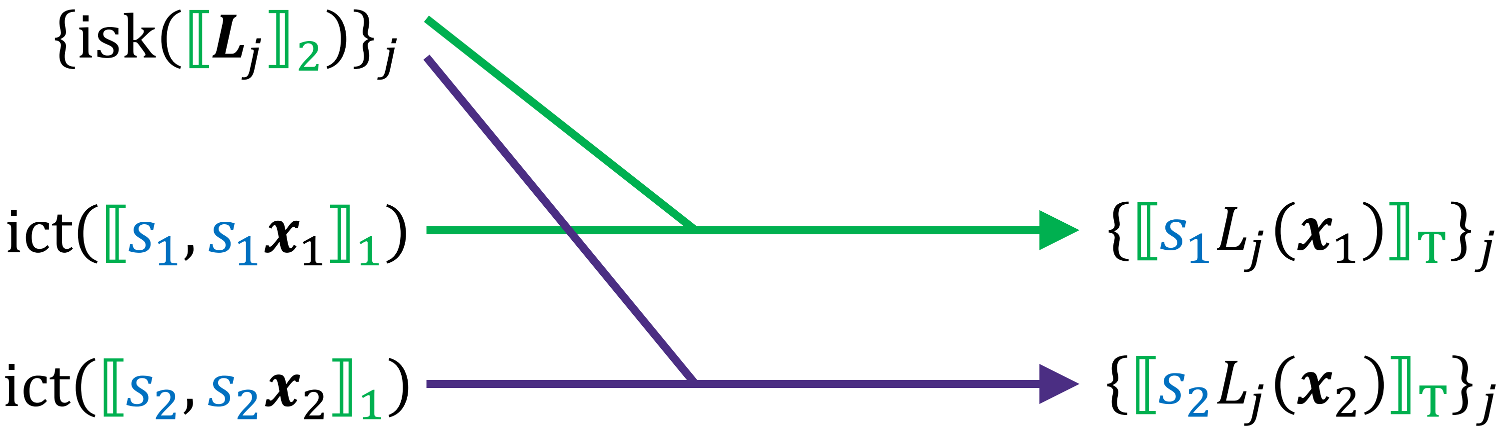


$\{\mathrm{isk}(\boldsymbol{L}_j)\}_j$

$\mathrm{ict}(1, \boldsymbol{x}_1)$ → $\{L_j(\boldsymbol{x}_1)\}_j$

$\mathrm{ict}(1, \boldsymbol{x}_2)$ → $\{L_j(\boldsymbol{x}_2)\}_j$

✗ Garbling security breaks if label functions are reused!



$\{\mathrm{isk}([\![\boldsymbol{L}_j]\!]_2)\}_j$

$\mathrm{ict}([\![s_1, s_1\boldsymbol{x}_1]\!]_1)$ → $\{[\![s_1 L_j(\boldsymbol{x}_1)]\!]_{\mathrm{T}}\}_j$

$\mathrm{ict}([\![s_2, s_2\boldsymbol{x}_2]\!]_1)$ → $\{[\![s_2 L_j(\boldsymbol{x}_2)]\!]_{\mathrm{T}}\}_j$

✓ DDH ensures $\{\boldsymbol{L}_j, \{s_i\boldsymbol{L}_j\}_i\}_j$ looks fresh in group.

ABE for Uniform in [LL20a]

Previous. input length of f is fixed (non-uniform model)

Now. more flexible (e.g., NFA)

- sk_Γ for regular expression Γ
  (works with ***all possible*** input length)
- $\mathrm{ct}_{\boldsymbol{x}}$ for input string $\boldsymbol{x}$
  (works with ***all possible*** reg.exp. size)

Same Paradigm.

- garbling for NFA, NL
- use IPFE to compute garbling
- proof guided by same idea

simple idea, complex execution,
IPFE helpful in managing proof

Tweaks. garbling size $\Theta(|\Gamma| \cdot |\boldsymbol{x}|)$

- sk = $\Theta(|\Gamma|)$ many isk's
- ct = $\Theta(|\boldsymbol{x}|)$ many ict's

make ***every pair*** of decryption useful!

Achievements of [LL20a]

ABE for Non-Uniform. ABP, adaptive, standard assumptions.

- **Previous.** puts bound on program size upon Setup [LOSTW10],
  or non-adaptive [GSPW06],
  or non-standard assumptions [LW12].
- **Previous, Concurrent.**
  for Boolean formula / branching programs [KW19,GW20].

ABE for Uniform. (N)L, (linear-size) N/DFA, adaptive, standard assumptions.

- **Previous.** for DFA,
  non-adaptive or large components or non-standard assumptions
  [W12,A14,AMY19,GWW19].
- **Concurrent.** [GW20]
  for DFA, same achievements;
  for NFA, non-adaptive.

* comparison only with pairing-based schemes

Power of Paradigm Exhibited by [LL20a]

one method solving **many** open problems (pairing-based)

- adaptive ABE for arithmetic computation / DFA
- ABE for NFA

almost the **end game** of adaptive standard ABE from pairing

- small remaining gap between selective/adaptive ABE
  (arithmetic **span** program vs A**B**P)
- **still the only** known adaptive ABE for NFA, L, NL
  (for ABP, improved in [LL20b])

✗ Size of garbling with our security notion is tightly related to ABP size. [Luo20汉]

reused in the **future**

- next-up in this talk
- same IPFE / garbling used for AB-FE for ABP, L [DP21,DPT22]

Remember Succinctness?

$|\mathrm{sk}_f| < |f|$?

$\mathrm{sk}_f = \mathrm{isk}(\boldsymbol{L}_1), \ldots, \mathrm{isk}(\boldsymbol{L}_m)$

IPFE

$\approx$

(x then f)

$\mathrm{isk}(\boldsymbol{\ell}_1, \mathbf{0}), \ldots, \mathrm{isk}(\boldsymbol{\ell}_m, \mathbf{0})$

- has $m = |f|$ objects (isk's)
- has $\geq m$ bits of (garbling) randomness

must hide garbling randomness

(non-hiding – more difficult for proof)

use non-hiding isk to *bind* to $\boldsymbol{x}$

$|\mathrm{ct}_x| < |\boldsymbol{x}|$?

$\mathrm{ct}_{\boldsymbol{x}}(\mu) = \mathrm{ict}(1, \boldsymbol{x})$

$\mathrm{ict}(1, \boldsymbol{x})$

- IPFE (hiding) security $\Longrightarrow$ $|\mathrm{ict}| \geq |\boldsymbol{x}|$

nothing to hide

no hiding required for "x then f" case

Using IPFE with Succinct Keys

$\mathrm{ct}_f(\mu) = \mathrm{ict}(\boldsymbol{L}_1), \ldots, \mathrm{ict}(\boldsymbol{L}_m)$

$\mathrm{ict}(0, \mathbf{0}, 1)$,
$\mathrm{ict}(\ell_2, \mathbf{0}), \ldots, \mathrm{ict}(\ell_m, \mathbf{0})$

many steps?

$\approx$

(f then x)

Two values hardwired during proof.

$\xrightarrow{\boldsymbol{x}}$

$\mathrm{sk}_{\boldsymbol{x}} = \mathrm{isk}(1, \boldsymbol{x})$

$\boxed{\mathrm{isk}(1, \boldsymbol{x}, \ell_1)}$

cannot hardwire ℓ_1
by changing vector

- ✓ $|\mathrm{isk}| = O(1)$
- ⚠ no hiding
- □ CP-1-ABE

Solution. IPFE with ***simulation security***.

(**stronger formulation compatible with proof**)

IPFE with Simulation Security

$$\begin{array}{ccc}
\mathrm{impk} & & \widetilde{\mathrm{impk}} \\[2em]
\{\, \mathrm{isk}(\ \boldsymbol{v}_j\ )\, \} & & \{\, \widetilde{\mathrm{isk}}\,(\overbrace{\ \boldsymbol{v}_j \ \big|\ \ \bot\ \ }^{\text{input to simulator}})\, \} \\[1em]
\mathrm{ict}(\ \boldsymbol{u}\ ) & \approx & \widetilde{\mathrm{ict}}\,(\ \bot\ \big|\ \{\boldsymbol{u}^\top \boldsymbol{v}_j\}_{j<J_1}\ ) \\[1em]
\{\, \mathrm{isk}(\ \boldsymbol{v}_j\ )\, \} & & \{\, \widetilde{\mathrm{isk}}\,(\ \boldsymbol{v}_j\ \big|\ \boldsymbol{u}^\top \boldsymbol{v}_j\ )\, \}
\end{array}$$

At every moment, (adaptive)
input to simulator is
whatever is **intended**
to be revealed.

Constructions. [LL20b]

- Generically from any selectively secure IPFE.
- Direct by modifying [ALS16] (better efficiency).

Stronger Formulation. [LL20b]

1. Can simulate up to T ciphertexts.
   (T tunable at Setup, affects component sizes)
2. Can **do/undo** simulation for any ict **in the presence of other** $\widetilde{\mathrm{ict}}$'s.

Using Simulation-Secure IPFE in [LL20b]

$$\mathrm{ct}_f(\mu) = \mathrm{ict}(\boldsymbol{L}_1), \ldots, \mathrm{ict}(\boldsymbol{L}_m)$$

$$\widetilde{\mathrm{ict}}(\bot|\bot),$$
$$\mathrm{ict}(\ell_2, \mathbf{0}), \ldots, \mathrm{ict}(\ell_m, \mathbf{0})$$

many steps

$\approx$

(f then x)

$\xrightarrow{\boldsymbol{x}}$

$$\mathrm{sk}_{\boldsymbol{x}} = \mathrm{isk}(1, \boldsymbol{x})$$

$$\widetilde{\mathrm{isk}}(1, \boldsymbol{x}|\ell_1)$$

✓ $|\mathrm{isk}| = O(T)$ with $T = 2$

❑ CP-1-ABE

Multi-key security? KP-ABE?

- CP-1-ABE + dual system [W09] ⟹ KP-ABE
- KP-ABE ⟹ KP-1-ABE (trivial)
- KP-1-ABE + dual system ⟹ CP-ABE*

* a factor of 2 shaved off in sizes compared to usual implementation of dual system, somehow…

Summary of [LL20b]

Achievements in Succinct ABE. ABP, adaptive, standard assumptions.

- **Part with x is Succinct.** ct_x in KP-ABE, sk_x in CP-ABE.
- **Previous.** only (natively) for Boolean computation,
  non-adaptive or non-standard assumptions [A16,ZGTCLQC16].
- **Concurrent.**
  for Boolean formulae [AT20]
  only 1 fewer group element in ct for KP.

What about the Paradigm? (not fully within paradigm)

- **"Ablation Study" of Roles of IPFE.** By comparing [LL20a] with literature…
  computing garbling (new IPFE);
  rerandomizing garbling (dual system).
- **Learn in Abstraction, Improve by Breaking It.**
  paradigm = bridge to reach the goal?

Moving Beyond Noiseless Garbling

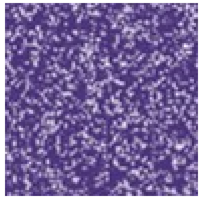
Part with x is Succinct. ct_x in KP-ABE, sk_x in CP-ABE.

Part with f?

Fact. **Size of noiseless linear garbling** tightly related to **span program size** [B84,M87,BDHM92,KW93] (linear algebraic computation, **low-depth**).

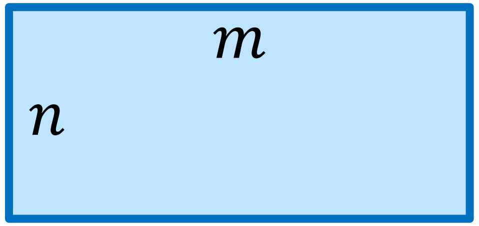
Noiseless

- cannot make f-part **succinct**
- does not handle **high** depth

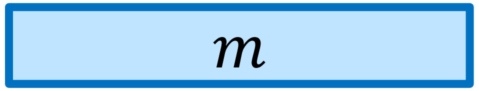


Let's try allowing noises!

Attribute Encoding from Lattices [BGGHNSVV14,GV15]



$$\boldsymbol{A} = (\boldsymbol{A}_1, \dots, \boldsymbol{A}_{|\boldsymbol{x}|}) \in \mathbb{Z}_q^{n\times|\boldsymbol{x}|m} \xrightarrow[\text{for circuit } C]{\text{EvalC}} \boldsymbol{A}_C \in \mathbb{Z}_q^{n\times m}$$



$$\boldsymbol{s}^\top(\boldsymbol{A} - \boldsymbol{x} \otimes \boldsymbol{G}) + \boldsymbol{e}^\top \xrightarrow[\text{for } C \text{ and } \boldsymbol{x}]{\text{EvalCX}} \boldsymbol{s}^\top(\boldsymbol{A}_C - C(\boldsymbol{x}) \cdot \boldsymbol{G}) + \boldsymbol{e}_C^\top$$

$$\uparrow$$

$$= \left(\boldsymbol{s}^\top(\boldsymbol{A}_1 - \boldsymbol{x}[1] \cdot \boldsymbol{G}) + \boldsymbol{e}_1^\top, \dots\right)$$

- **homomorphic** encoding
- sizes depend on depth d of C, **not size**
- noise growth is **exponential** in d

*** What is $\boldsymbol{G}$?**
Some fixed, publicly known matrix – details not needed for now.

Noisy Linear Garbling from Attribute Encoding

Public Parameters. $\boldsymbol{A}$, short $\boldsymbol{z}$

Think binary x.

Labels. $\boldsymbol{s}^\top(\boldsymbol{A} - \boldsymbol{x} \otimes \boldsymbol{G}) + \boldsymbol{e}^\top = \utilde{\boldsymbol{s}^\top(\boldsymbol{A} - \boldsymbol{x} \otimes \boldsymbol{G})} = \boldsymbol{c}^\top$ (wavy = noises)

Evaluation.

1. $\boldsymbol{c}^\top \xrightarrow{\text{EvalCX}} \utilde{\boldsymbol{s}^\top(\boldsymbol{A}_C - C(\boldsymbol{x}) \cdot \boldsymbol{G})} = \boldsymbol{c}_C^\top$
2. output $\boldsymbol{c}_C^\top \boldsymbol{z}$

$$= \utilde{\boldsymbol{s}^\top \boldsymbol{A}_C \boldsymbol{z}} - C(\boldsymbol{x}) \cdot \utilde{\boldsymbol{s}^\top \boldsymbol{G} \boldsymbol{z}}$$

- $C(\boldsymbol{x}) = 0$, then just the secret
- $C(\boldsymbol{x}) = 1$, then $\utilde{\boldsymbol{s}^\top \boldsymbol{G} \boldsymbol{z}}$ is OTP to hide secret *

Changes.

- "$P(C, x) = \neg C(\boldsymbol{x})$" – recover secret when $C(\boldsymbol{x}) = 0$
- secret is $\utilde{\boldsymbol{s}^\top \boldsymbol{A}_C \boldsymbol{z}}$

* not the full story, but good enough for now

Using Noisy Linear Garbling in [LLL22]

$|\mathsf{ct}_f| = O(|\boldsymbol{x}|^2) < |f|$

$$\mathsf{ct}_f(\mu) = \mathsf{ict}([\![\boldsymbol{s}^\top \boldsymbol{A}_1 + \boldsymbol{e}_1^\top,\ \boldsymbol{s}^\top(\boldsymbol{A}_1 - \boldsymbol{G}) + \boldsymbol{e}_1^\top]\!]_1), \ldots$$

✓ polynomial noise for log-depth circuits (formulae)

$$\xrightarrow[\text{decryption}]{\text{IPFE}} \boxed{[\![r(}\boldsymbol{s}^\top(\boldsymbol{A}_1 - \boldsymbol{x}[1]\cdot\boldsymbol{G}) + \boxed{\boldsymbol{e}_1^\top)}]\!]_{\mathrm{T}}, \ldots \xrightarrow{\text{Eval}} [\![\mu]\!]_{\mathrm{T}}$$

DDH cannot rerandomize *small noises*

generic group model + stronger IPFE security

$|\mathsf{sk}_x| = O(1)$

$$\mathsf{sk}_{\boldsymbol{x}} = \mathsf{isk}(r[\![\neg\boldsymbol{x}, \boldsymbol{x}]\!]_2)$$

- selects $\boldsymbol{s}^\top\boldsymbol{A}_1$ or $\boldsymbol{s}^\top(\boldsymbol{A}_1 - \boldsymbol{G})$ etc.
- DDH-style rerandomization with r

Generic Group Model [S97,M05]

Standard Model.

- arbitrary computation on group element represented in bits
- certain computational problem is hard

Generic Group Model. intuitive although strong

- only operations via group-theoretic interfaces
  - addition, negation, zero-testing
  - pairing
- (equivalently) adversary capability

  = zero-test any **linear** function of $(\llbracket 1, \boldsymbol{w}_1 \rrbracket_1 \otimes \llbracket 1, \boldsymbol{w}_2 \rrbracket_2, \llbracket \boldsymbol{w}_{\mathrm{T}} \rrbracket_{\mathrm{T}})$

more control of adversarial behavior
$\Longrightarrow$ easier to write proofs

Saw $w_1 = \llbracket a \rrbracket_1, w_2 = \llbracket b \rrbracket_2, w_{\mathrm{T}} = \llbracket c \rrbracket_{\mathrm{T}}$.
Define $L(r, s) = r - s$. Ask $L(\boldsymbol{w}_1 \otimes \boldsymbol{w}_2, \boldsymbol{w}_{\mathrm{T}}) \overset{?}{=} 0$.

Yes/no. (Tests whether $ab = c$.)

Very Strong IPFE Simulation Security of [LLL22]

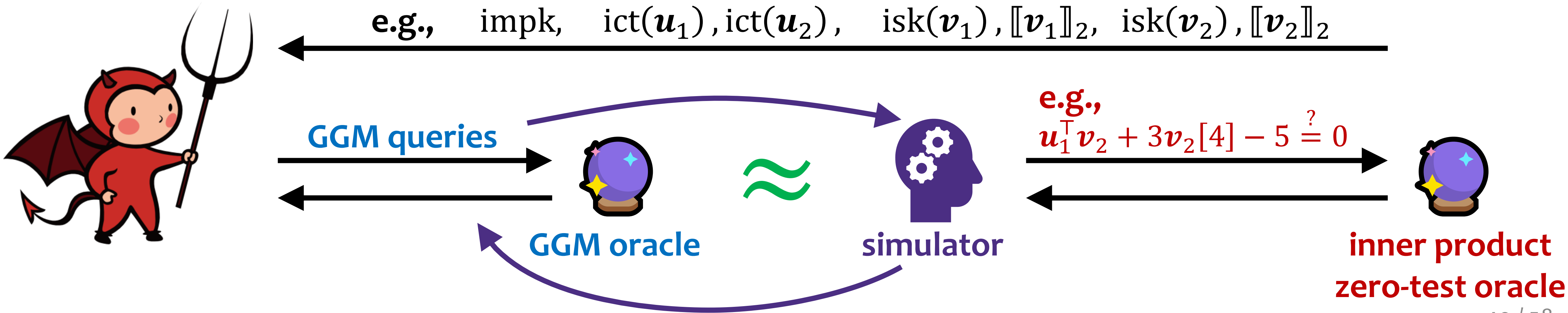
Generic Group Model.

> *"All you can do is to **zero-test** linear functions of **pairing results** (and target group elements)."*

IPFE Simulation in GGM.

> *"All you can do is to **zero-test** linear functions of **inner products** (and key vectors)."*

✓ proven for [ABDP15]
✓ rerandomization with r now **works**



e.g., $\mathrm{impk}, \quad \mathrm{ict}(\boldsymbol{u}_1), \mathrm{ict}(\boldsymbol{u}_2), \quad \mathrm{isk}(\boldsymbol{v}_1), [\![\boldsymbol{v}_1]\!]_2, \quad \mathrm{isk}(\boldsymbol{v}_2), [\![\boldsymbol{v}_2]\!]_2$

e.g., $\boldsymbol{u}_1^{\top}\boldsymbol{v}_2 + 3\boldsymbol{v}_2[4] - 5 \overset{?}{=} 0$

Summary of [LLL22]

Doubly Succinct CP-ABE. for Boolean **formulae** (log-depth circuits)

- $|\mathrm{sk}_{\boldsymbol{x}}| = O(1) < |\boldsymbol{x}|$ and $|\mathrm{ct}_f| = O(|\boldsymbol{x}|^2) < |f|$
- **first** ABE with non-trivial **double** succinctness

- **Previous.** [AY20] CP-ABE from pairing + lattices
  [AWY20] —"— with [LL20a] IPFE (not **doubly** succinct)

More Succinct KP-ABE. for Boolean **circuits** with $|\mathrm{sk}_C| = O(1)$

- **Previous.** [BGGHNSVV14] with $|\mathrm{sk}_C| = \mathrm{poly}(d)$
- **Later.** [CW23] from just LWE

Versality of Paradigm. Can **combine** pairing and lattices.

Lattices, not Pairing

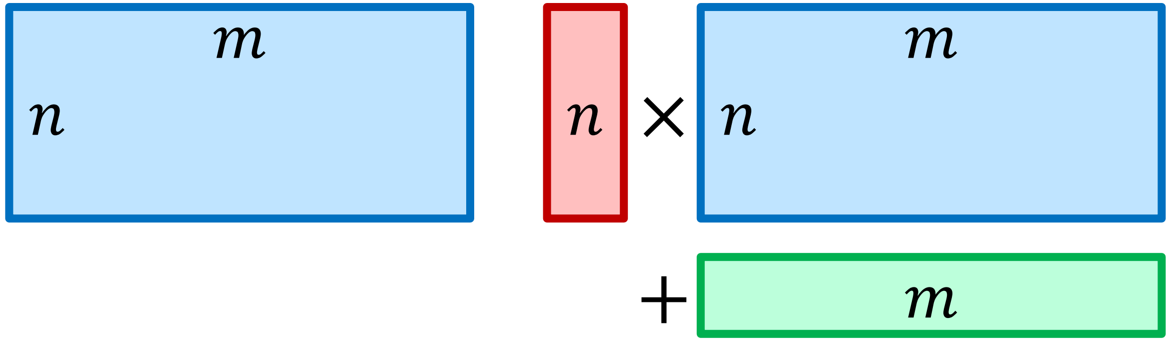
Why not pairing?

- not post-quantum secure
- noise must be polynomially bounded

Learning with Errors (LWE).

protection for $\boldsymbol{s}$

$$\boldsymbol{A}, \qquad \boldsymbol{s}^\top \boldsymbol{A} + \boldsymbol{e}^\top \qquad \approx \qquad \boldsymbol{A}, \ \$$$



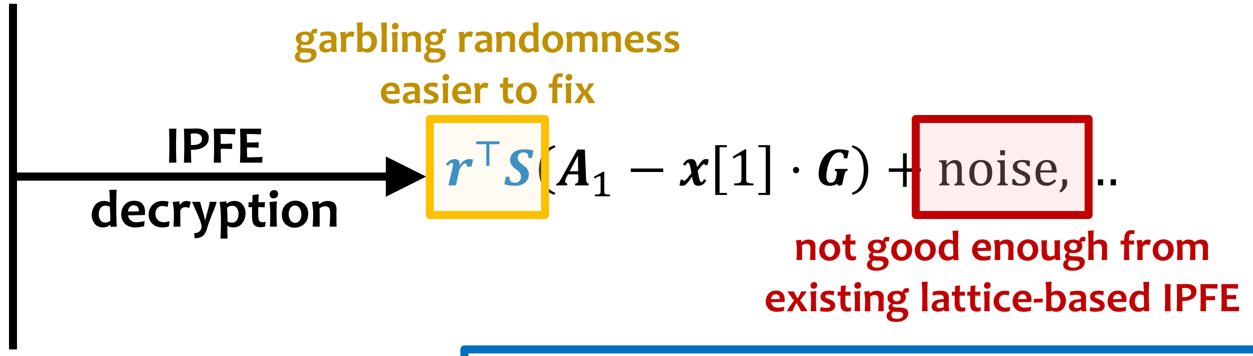
- presumably post-quantum
- OK with somewhat large noise

✓ builds some IPFE

✗ IPFE insufficient for ABE

Rerandomization with Lattice-Based IPFE

$$\mathrm{ct}_f(\mu) = \mathrm{ict}(\boldsymbol{S}\boldsymbol{A}_1,\ \boldsymbol{S}(\boldsymbol{A}_1 - \boldsymbol{G}))\,, \ldots$$



IPFE decryption

garbling randomness easier to fix

$$\boxed{\boldsymbol{r}^\top\boldsymbol{S}}(\boldsymbol{A}_1 - \boldsymbol{x}[1]\cdot\boldsymbol{G}) + \boxed{\text{noise},}\ \ldots$$

not good enough from existing lattice-based IPFE

$$\mathrm{sk}_{\boldsymbol{x}} = \mathrm{isk}(\boldsymbol{r}^\top(\neg\boldsymbol{x}, \boldsymbol{x}))$$

Goal. lattice-based IPFE giving good noises

Lattice Trapdoors [MP12] and Evasive LWE [W22,T22]

“$\boldsymbol{K} = \boldsymbol{B}^{-1}(\boldsymbol{P})$”

trapdoor of $\boldsymbol{B}$ = information about $\boldsymbol{B}$ for solving $\boldsymbol{B}\boldsymbol{K} = \boldsymbol{P}$ for **small** $\boldsymbol{K}$, given any $\boldsymbol{P}$

$$(\boldsymbol{s}^\top \boldsymbol{B} + \boldsymbol{e}^\top) \cdot \boldsymbol{B}^{-1}(\boldsymbol{P}) = \boldsymbol{s}^\top \boldsymbol{P} + \boldsymbol{e}^\top \boldsymbol{K}$$

$$\boxed{\boldsymbol{s}^\top \boldsymbol{B} \cdot \boldsymbol{B}^{-1}(\boldsymbol{P}) = \boldsymbol{s}^\top \boldsymbol{P}}$$

- **controlled multiplication**
- **makes LWE fail**
  (no protection for $\boldsymbol{s}$)

$\boldsymbol{s}^\top \boldsymbol{B} \cdot \boldsymbol{B}^{-1}(\boldsymbol{0})$ small
$\$ \cdot \boldsymbol{B}^{-1}(\boldsymbol{0})$ random

Evasive LWE. (conditional protection for $\boldsymbol{s}$)

*“The **only meaningful way** to use $\boldsymbol{B}^{-1}(\boldsymbol{P})$ is to **multiply it to** $\boldsymbol{s}^\top \boldsymbol{B}$ and ignore noise correlation.”*

$$\text{if} \quad (\boldsymbol{B}, \boldsymbol{P},\ \boldsymbol{s}^\top \boldsymbol{B} + \boldsymbol{e}_{\mathrm{B}}^\top,\ \boldsymbol{s}^\top \boldsymbol{P} + \boldsymbol{e}_{\mathrm{P}}^\top) \approx (\boldsymbol{B}, \boldsymbol{P},\ \$,\ \$)$$

$$\text{then} \quad (\boldsymbol{B}, \boldsymbol{P},\ \boldsymbol{s}^\top \boldsymbol{B} + \boldsymbol{e}_{\mathrm{B}}^\top,\ \boldsymbol{B}^{-1}(\boldsymbol{P})\ ) \approx (\boldsymbol{B}, \boldsymbol{P},\ \$,\ \boldsymbol{B}^{-1}(\boldsymbol{P}))$$

Evasive LWE and Evasive IPFE [HLL24]

IPFE (full protection) ⇐

Corporate needs you to find the differences between this picture and this picture.
They're the same picture.

Pairing. controlled multiplication

DDH or GGM. some protection

Lattice Trapdoor. controlled multiplication

LWE + Evasive LWE. some protection

⇒ Evasive IPFE

suffices for ABE *

Functionality.

$$\mathrm{Dec}(\mathrm{impk}, \boldsymbol{v}, \mathrm{isk}(\boldsymbol{v}), \mathrm{ict}(\boldsymbol{u})) = \boldsymbol{u}^\top \boldsymbol{v} + e_{\mathrm{Dec}}$$

Security.

$I \times J$ fresh noises

if $\{\boldsymbol{u}_i^\top \boldsymbol{v}_j + e_{ij}\}_{ij} \approx \{\$\}_{ij}$

then

$$(\mathrm{impk}, \{\boldsymbol{v}_j, \mathrm{isk}(\boldsymbol{v}_j)\}_j, \{\mathrm{ict}(\boldsymbol{u}_i)\}_i)$$
$$\approx (\mathrm{impk}, \{\boldsymbol{v}_j, \mathrm{isk}(\boldsymbol{v}_j)\}_j, \{\mathrm{ict}(\$)\}_i)$$

* not the full story, but good enough for now

Achievements of [HLL24]

Lattice-Based CP-ABE. for circuits (using garbling from [BGGHNSVV14])

- from LWE + evasive LWE
- **Previous.** [BV20] no security proof
  [W22] from LWE + tensor LWE + evasive LWE

ABE for Uniform. for DFA, L (using garbling from [LL20a])

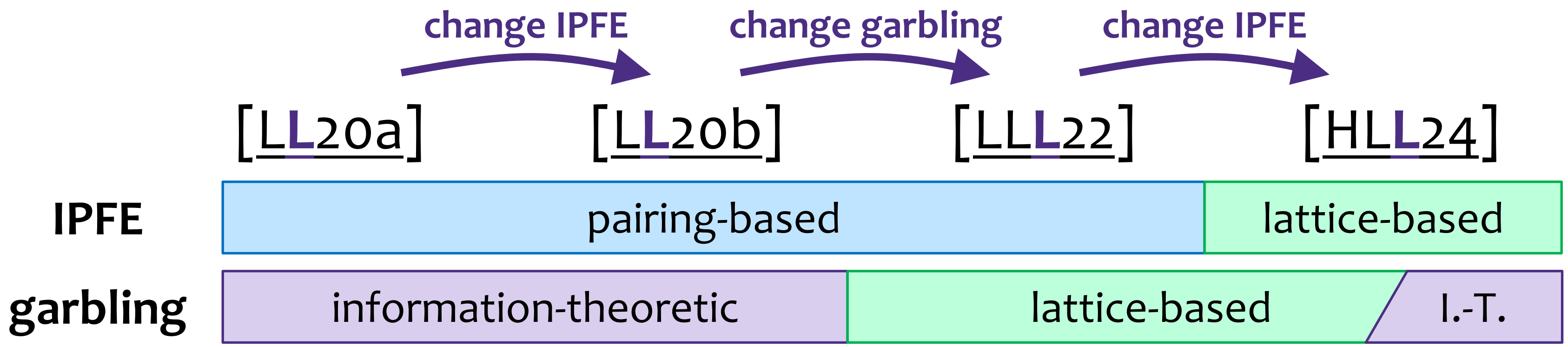
- **first** lattice-based public-key ABE for uniform
- **Previous.** [AS17,W22] against bounded collusion
  [AMY19] secret-key ABE for NFA
  [W22] no security proof

Summary of Paradigm

ABE $\Leftarrow$ IPFE $\circ$ Garbling

Composition of Security.

- IPFE – only rerandomized garblings revealed
- assumption – garblings are properly rerandomized
- garbling – secret/message hidden if unauthorized



change IPFE

change garbling

change IPFE

[LL20a] [LL20b] [LLL22] [HLL24]

IPFE: pairing-based | lattice-based

garbling: information-theoretic | lattice-based | I.-T.

Modular. **Hides** most ***raw*** usage of computational assumptions into IPFE and garbling security.

Powerful. Achieves various ABE with **better** properties.

Versatile. Works with pairing, lattice, or pairing + lattice.

Open Questions from Part I

- Gap between **selective/adaptive** ABE from **static** pairing assumptions

  (arithmetic span program vs ABP)

- **CP**-ABE for circuits from **falsifiable** lattice assumptions
- ABE for **DFA** from **falsifiable** lattice assumptions

  (Evasive LWE is non-falsifiable.)

Part II. **More**

Nothing technical now,
just the results and the messages.

Bounded and Unbounded (KP-ABE)

Recall. $\boldsymbol{s}^\top(\boldsymbol{A} - \boldsymbol{x} \otimes \boldsymbol{G}) + \boldsymbol{e}^\top \xrightarrow{\text{EvalCX}} \boldsymbol{s}^\top(\boldsymbol{A}_C - C(\boldsymbol{x}) \cdot \boldsymbol{G}) + \boldsymbol{e}_C^\top$

noise growth **exponential** in depth d of C

computation in $\mathbb{Z}_q$ – only works when $d = O(\log q)$

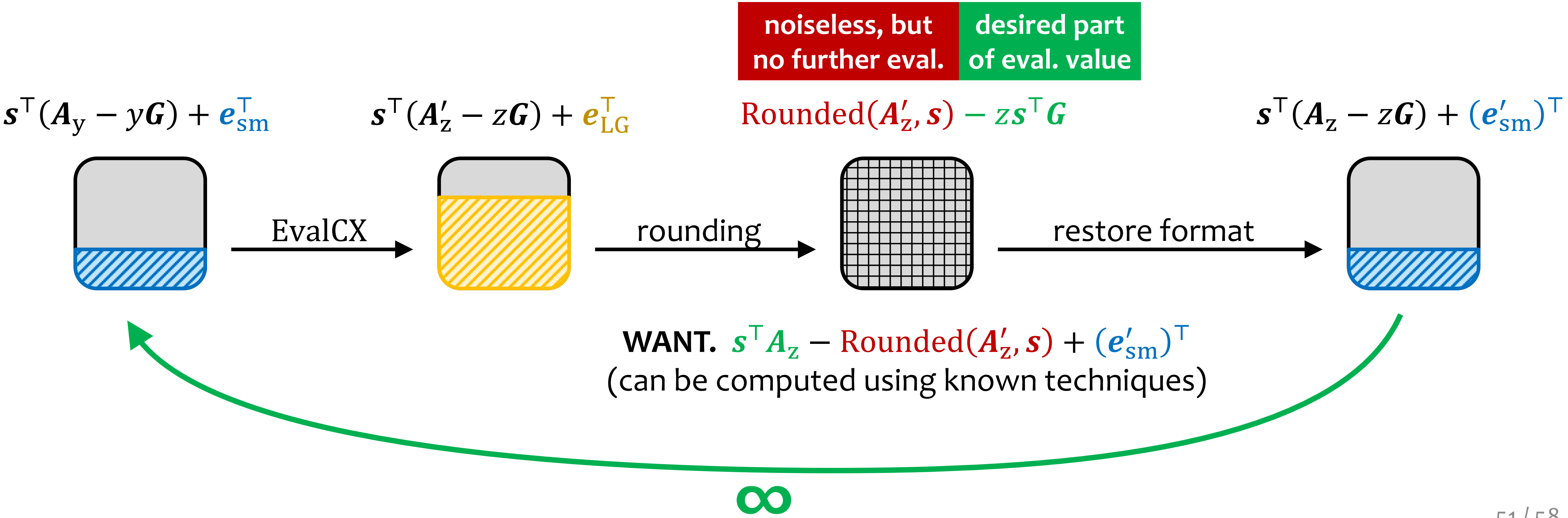
- q is often chosen upon Setup.
- q **must** be chosen upon Enc.
  - $\text{ct}_{\boldsymbol{x}}$ contains elements in $\mathbb{Z}_q$ (attribute encoding).
  - This forces $d \le |\text{ct}|$,
    so $\text{ct}_{\boldsymbol{x}}$ **cannot** work with sk_C if $d > |\text{ct}|$,
    even if $C(\boldsymbol{x}) = 0$.
  - "$\text{ct}_{\boldsymbol{x}}$ places an upper bound on d," (**depth-bounded**)
    even though $\boldsymbol{x}$ has nothing to do with d.

WANT. No upper bound on d from mpk, ct ("**depth-unbounded**").

Unbounded Evaluation for Attribute Encoding

Idea. (similar to *fully homomorphic encryption*)

- Start from ***somewhat small*** noise.
- Perform some evaluation. Noise becomes ***somewhat large***.
- **Reduce noise** to ***somewhat small*** before it ***overflows***.
- Rinse and repeat.



noiseless, but no further eval.

desired part of eval. value

$\boldsymbol{s}^\top(\boldsymbol{A}_\mathrm{y} - y\boldsymbol{G}) + \boldsymbol{e}_\mathrm{sm}^\top$

EvalCX

$\boldsymbol{s}^\top(\boldsymbol{A}'_\mathrm{z} - z\boldsymbol{G}) + \boldsymbol{e}_\mathrm{LG}^\top$

rounding

$\mathrm{Rounded}(\boldsymbol{A}'_\mathrm{z}, \boldsymbol{s}) - z\boldsymbol{s}^\top\boldsymbol{G}$

restore format

$\boldsymbol{s}^\top(\boldsymbol{A}_\mathrm{z} - z\boldsymbol{G}) + (\boldsymbol{e}'_\mathrm{sm})^\top$

WANT. $\boldsymbol{s}^\top\boldsymbol{A}_\mathrm{z} - \mathrm{Rounded}(\boldsymbol{A}'_\mathrm{z}, \boldsymbol{s}) + (\boldsymbol{e}'_\mathrm{sm})^\top$

(can be computed using known techniques)

∞

Achievements and Open Questions [HLL23]

KP-ABE for circuits of unbounded depth.

- long-standing open problem
- from circular LWE + evasive circular LWE (circular = encrypt $\boldsymbol{s}$ using $\boldsymbol{s}$)

- **Previous.** [BGGHNSVV14] for bounded depth
- **Concurrent.** [CW23] for bounded depth with $|\mathrm{sk}_C| = O(1)$

Related Primitives. (with depth bound removed, from circular LWE)

Open Questions. depth-unbounded KP-ABE
from **falsifiable** (no “evasive”) lattice assumptions

Dream and Actual Versions of ABE [JLL23]

Previous. ABE for ABP, NL, circuits…

ABE for RAM (best model for real-world programs)

$$|\mathrm{sk}_f| = O(1)$$

$$|\mathrm{ct}_x| = O(1)$$

$$T_{\mathrm{Dec}} = O(T_{\mathrm{RAM},f,x})$$

possible that $T_{\mathrm{RAM},f,x} < |x|$ (think binary search)

from functional encryption for circuits
with *arbitrarily bad* efficiency… ↓

$\lvert\mathrm{sk}_f\rvert$	$\lvert\mathrm{ct}_x\rvert$	T_{Dec}
$O(1)$	$O(1)$	$O(T + \lvert f\rvert + \lvert x\rvert)$
$\lvert f\rvert + O(1)$	$O(1)$	$O(T + \lvert x\rvert)$
$O(1)$	$\lvert x\rvert + O(1)$	$O(T + \lvert f\rvert)$
$\lvert f\rvert + O(1)$	$\lvert x\rvert + O(1)$	$O(T)$

Are we (am I) stupid,
or is it some necessary evil?

YOU CAN (NOT) OPTIMIZE [L24]

Theorem. For any secure ABE supporting $P(f = i,\ \ x = \boldsymbol{R}) = \boldsymbol{R}[i]$, it holds that

$$|\mathrm{ct}_x| \cdot T_{\mathrm{Dec}} = \Omega(|x|).$$

Similar trade-off lower bound holds between $|\mathrm{sk}_f|$ and T_{Dec}.

$\Longrightarrow$ Schemes of [JLL23] are **Pareto-optimal.**

Fact. Schemes of [JLL23] can be modified into

$$|\mathrm{sk}_f| = |f|^{\alpha} + O(1), \quad |\mathrm{ct}_x| = |x|^{\beta} + O(1),$$
$$T_{\mathrm{Dec}} = O\big(T \times (|f|^{1-\alpha} + |x|^{1-\beta})\big),$$

for any constants $0 < \alpha, \beta < 1$.

$\Longrightarrow$ The trade-off lower bound is **tight** for $T = O(1)$.

Achievements of [JLL23,L24]

New Agenda.

- multi-objective optimization
- quest for Pareto-optimality

Trade-Off Lower Bounds for ABE.

- first such bounds
- **Message.** Maybe succinctness is not worth it
  if we must pay dearly for each decryption?

Constructions.

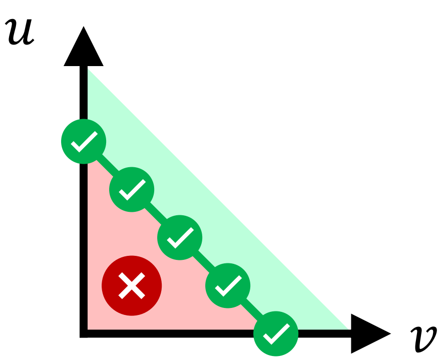
- down-to-constant optimization
- Pareto-optimal

Open Questions from [JLL23,L24]

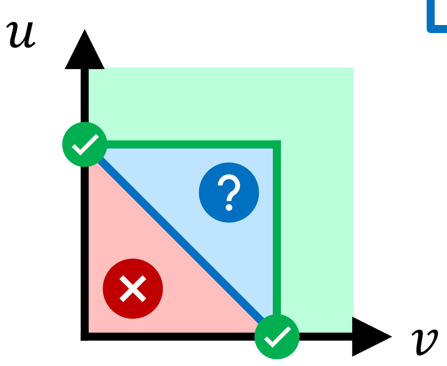
- fully pin down the Pareto frontier for general T

$|\mathsf{ct}_x| = O(|x|^u)$
$T_{\mathsf{Dec}} = O(T + |x|^v + \cdots)$

($T = O(1)$)



(general T)



- Is "f, x verbatim for free" the correct cost model?

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}'_f, \mathsf{ct}'_x)$

from verbatim-for-free model. $\mathsf{ct}'_x = (x, \mathsf{ct}_x)$
cannot achieve $T_{\mathsf{Dec}} = O(T)$ with $|\mathsf{ct}'_x| = |x| + O(1)$.

other implementation achieving the goal? (ct'_x encoding x in some clever way)

Acknowledgments

advisors.
Rachel, Stefano.

other and former committee members.
Paul Beame, Gaku Liu, Anup Rao, Cynthia Vinzant.

coauthors.
Ivan Damgård, Sabine Oechsner, Peter Scholl, Mark Simkin,
Shengyu Zhao, Tingfung Lau, Eric I-Chao Chang, Yan Xu,
Rachel Lin, Hanjun Li, Junqing Gong, Hoeteck Wee,
Aayush Jain, Daniel Wichs, Yao-Ching Hsieh, Yevgeniy Dodis.

CSE members, former teachers, cohorts, friends.

mom, dad.