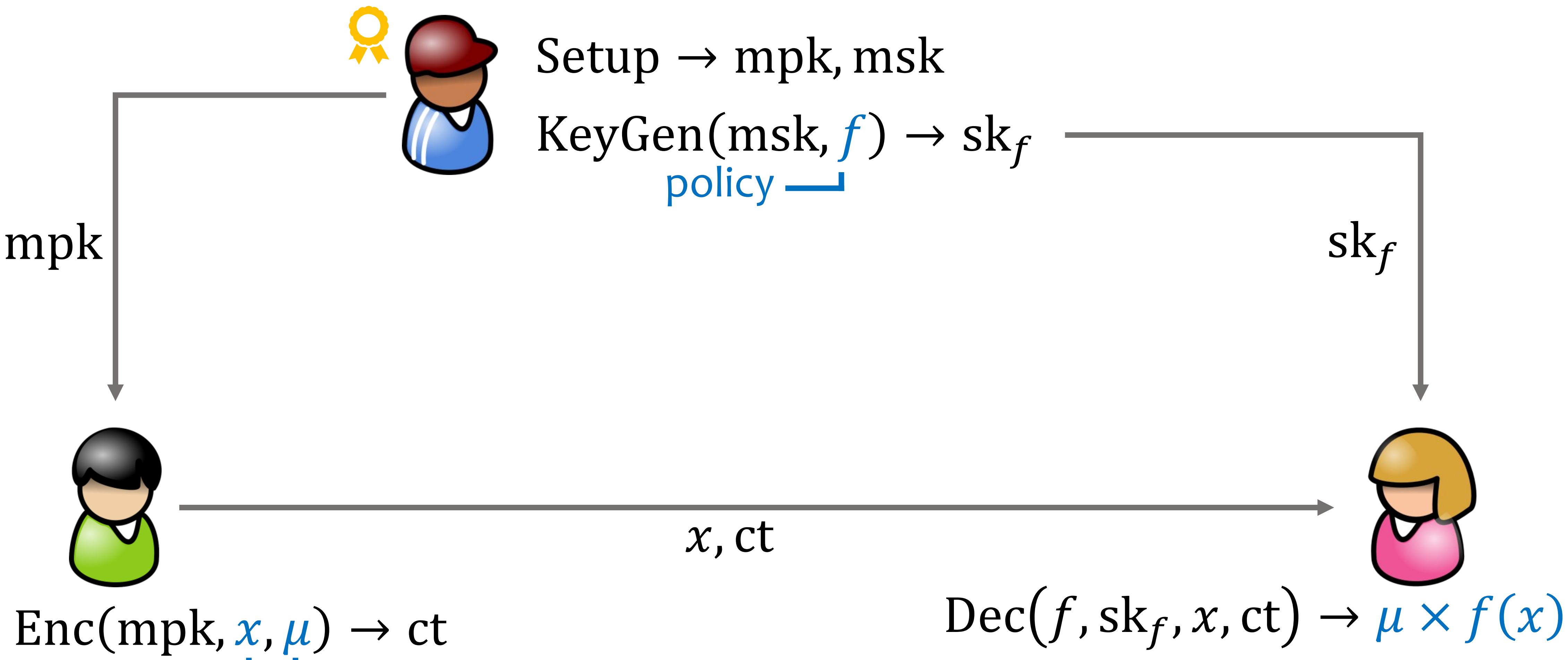
Succinct and Adaptively Secure ABE for ABP from k-Lin

Huijia (Rachel) Lin and Ji Luo

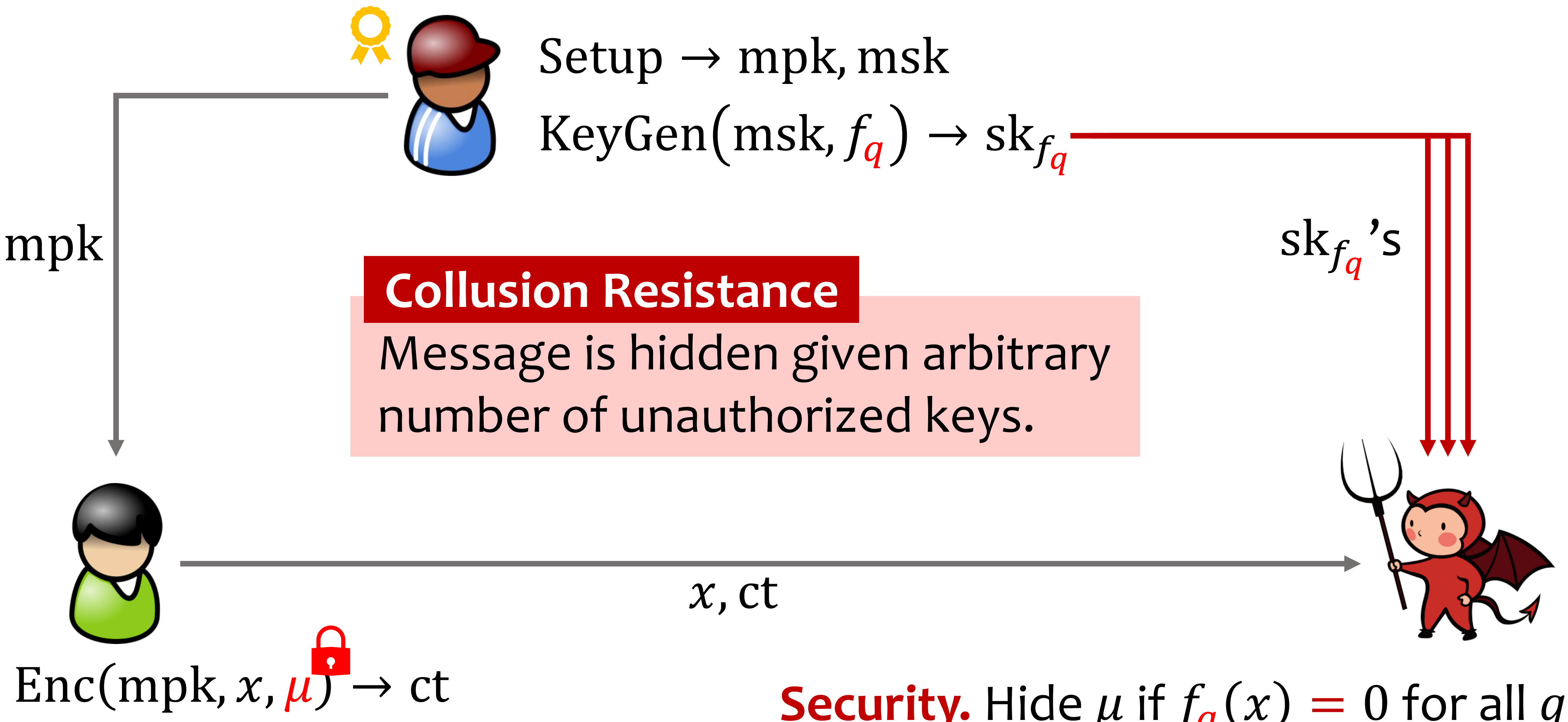
UNIVERSITY *of* WASHINGTON

Attribute-Based Encryption [SW05]



Correctness. Learn μ if $f(x) \neq 0$
(sk is authorized)

Attribute-Based Encryption [SW05]



Collusion Resistance

Message is hidden given arbitrary number of unauthorized keys.

Security. Hide μ if $f_q(x) = 0$ for all q (sk_q's are unauthorized)

Adaptive IND-CPA Security

Exp_b

$\xrightarrow{\mathrm{mpk}}$

$\xleftarrow{f_q}$

$\xrightarrow{\mathrm{sk}_q}$

$\xleftarrow{x, \mu_0, \mu_1}$

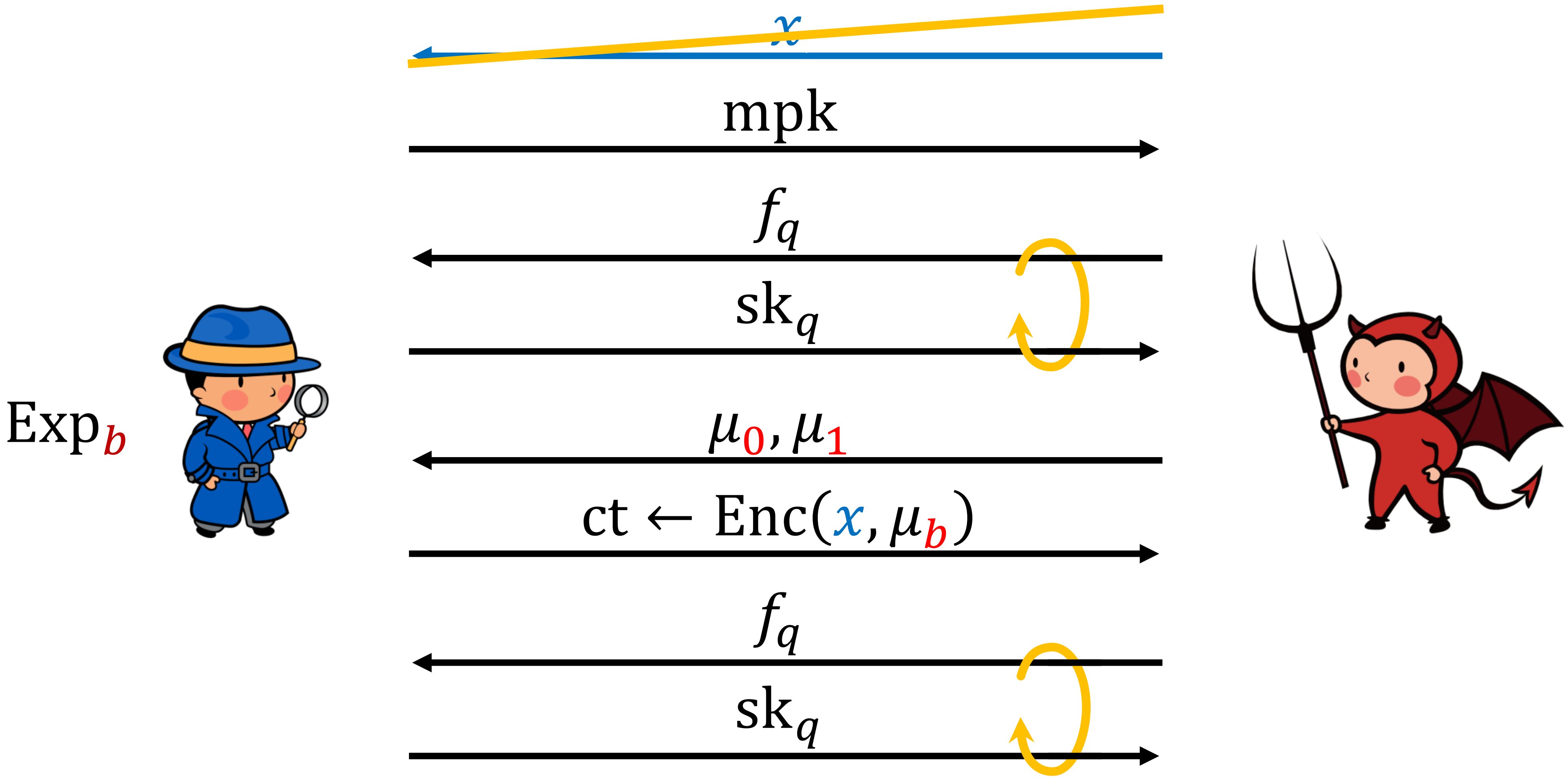
$\xrightarrow{\mathrm{ct} \leftarrow \mathrm{Enc}(x, \mu_b)}$

$\xleftarrow{f_q}$

$\xrightarrow{\mathrm{sk}_q}$

If for all queried keys $f_q(x) = 0$, then $\mathrm{Exp}_0 \approx \mathrm{Exp}_1$.

~~(Weaker) Selective~~ IND-CPA Security



If for all queried keys $f_q(x) = 0$, then $\mathsf{Exp}_0 \approx \mathsf{Exp}_1$.

Efficiency

How **succinct** can ABE ciphertexts be?

most schemes: $|\mathrm{ct}| = \mathrm{poly}(\lambda)\,|x| + |\mu|$

KEM trick

Note that x is **public,**

possible to have $|\mathrm{ct}| = \underbrace{\mathrm{poly}(\lambda)}_{\text{succinct}} + |\mu|$

succinct

Recall that $\mathrm{Dec}(f, \mathrm{sk}, x, \mathrm{ct}) \to \mu$

stored & transferred in the clear

Our Results

			plaintext in G_T
★ **Succinct**	$\lvert\mathrm{ct}\rvert =$	$4\lvert G_2\rvert + \lvert G_T\rvert$	$(2k+2)\lvert G_2\rvert + \lvert G_T\rvert$
★ Expressive		ABP	ABP
★ Adaptive		✔	✔
★ Standard Assumption		SXDH	MDDH_k

Plus, CP-ABE with succinct keys: $\lvert\mathrm{sk}\rvert = (3k+4)\lvert G_1\rvert$

Related Works

	$\lvert\mathrm{ct}\rvert$	**policies**	**adaptive**	**assumption**
Att16	18	MSP	✓	q-type
ZGT+16	$4k$	MSP		k-Lin ✓
TA20	$2k+2$	NC¹	✓	MDDH_k ✓
ours	$2k+3$	ABP	✓	MDDH_k ✓

ABP: arithmetic, includes NC^1

Framework of [LL20, Eurocrypt]

computational tool

function-hiding
Inner-**P**roduct
Functional **E**ncryption

non-succinct IPFE sk/ct

information-theoretic tool

Arithmetic **K**ey
Garbling **S**cheme

special randomized encoding

1-ABE = 1-key
1-ciphertext
secret-key

ABE

Framework of This Work

gradually simulation-secure
public-key IPFE
(**no** function-hiding)
succinct IPFE sk

Arithmetic **K**ey
Garbling **S**cheme

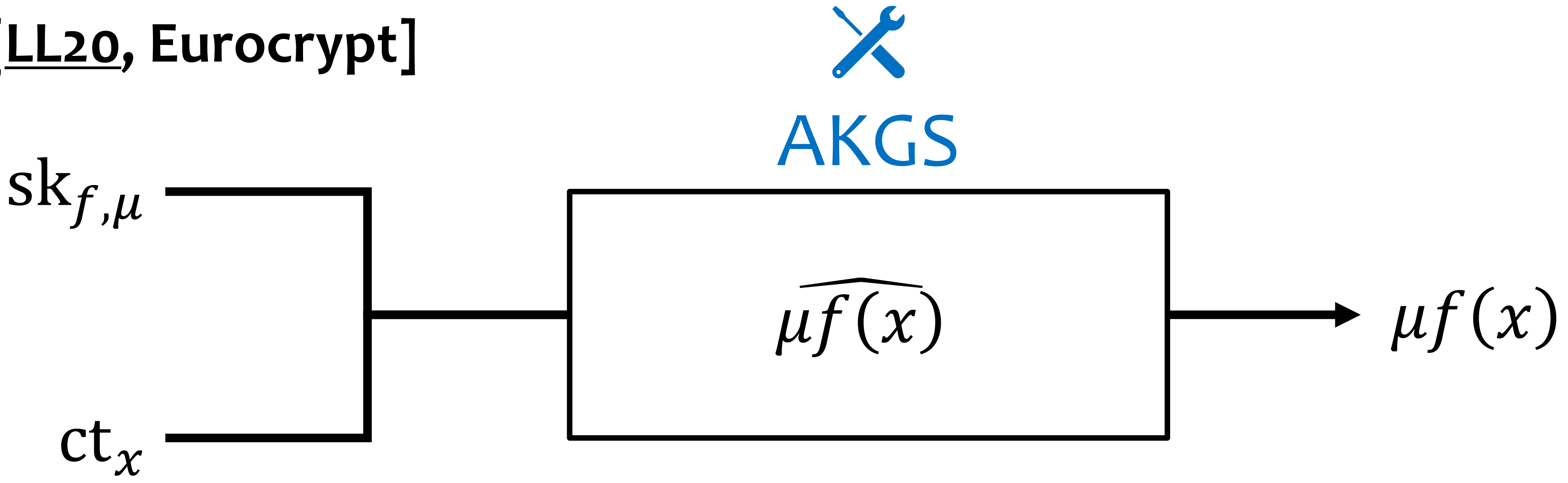
CP-1-ABE = 1-key, public-key

dual system / hash proof

KP-ABE

1-ABE via AKGS and IPFE

Idea from [LL20, Eurocrypt]



Secure. $\widehat{\mu f(x)}$ hides μ if $f(x) = 0$.
It does **not** hide f, x.

compute using IPFE $\Longrightarrow$ **Simple.** AKGS is **linear** in x.

Arithmetic Key Garbling Scheme

linear functions of x

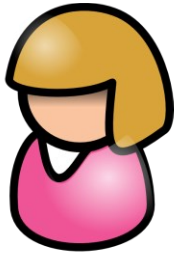
1. Label functions: $L_1, \ldots, L_m \leftarrow \mathrm{Garble}(f, \mu; r)$
2. Garblings: $\ell_1, \ldots, \ell_m = L_1(x), \ldots, L_m(x)$

a.k.a. "labels"

$f: \mathbb{Z}_p^n \to \mathbb{Z}_p$

$x \in \mathbb{Z}_p^n$

$\xrightarrow{f, x, \ell_1, \ldots, \ell_m}$



$\mathrm{Eval}(f, x, \ell_1, \ldots, \ell_m) = \mu f(x)$

linear in labels
(possible thanks to partial hiding)

Security (partial hiding).

$\mathrm{Sim}(\underbrace{f, x}_{\text{not hidden}}, \mu f(x)) \to \ell_1, \ldots, \ell_m$

Group-Based IPFE

$$\left.\begin{array}{l}\mathrm{isk} \leftarrow \mathrm{KeyGen}(\mathrm{msk}, \boldsymbol{v}) \\ \mathrm{ict} \leftarrow \mathrm{Enc}(\mathrm{msk}, \boldsymbol{u})\end{array}\right\} \xrightarrow{\mathrm{Dec}} [\![\langle \boldsymbol{u}, \boldsymbol{v} \rangle]\!] = g^{\langle \boldsymbol{u}, \boldsymbol{v} \rangle} \in G$$

Block Vector Notation

$$\left.\begin{array}{l}\mathrm{isk}(\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \boldsymbol{v}_3) \\ \mathrm{ict}(\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \boldsymbol{u}_3)\end{array}\right\} \rightarrow \langle \boldsymbol{u}_1, \boldsymbol{v}_1 \rangle + \langle \boldsymbol{u}_2, \boldsymbol{v}_2 \rangle + \langle \boldsymbol{u}_3, \boldsymbol{v}_3 \rangle$$

IND-CPA reveals $\langle \boldsymbol{u}, \boldsymbol{v} \rangle, \boldsymbol{v}$, hides $\boldsymbol{u}$;
can be **public-key** with **succinct** isk.

Function-Hiding reveals $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$, hides $\boldsymbol{u}, \boldsymbol{v}$;
only **secret-key**, **no succinctness**.

1-ABE via AKGS and IPFE

$$L_1, \ldots, L_m \leftarrow \mathrm{Garble}(f, \mu)$$

labels in the exponent

$$\mathrm{sk}_{f,\mu} = \{\mathrm{isk}(L_j)\}_{j\in[m]}$$

must be hidden

grows with $|x|$

$$\mathrm{ct}_x = \mathrm{ict}(x)$$

IPFE

Dec

$$[\![\ell_j = L_j(x)]\!]$$

AKGS

Eval

✔ **linear**

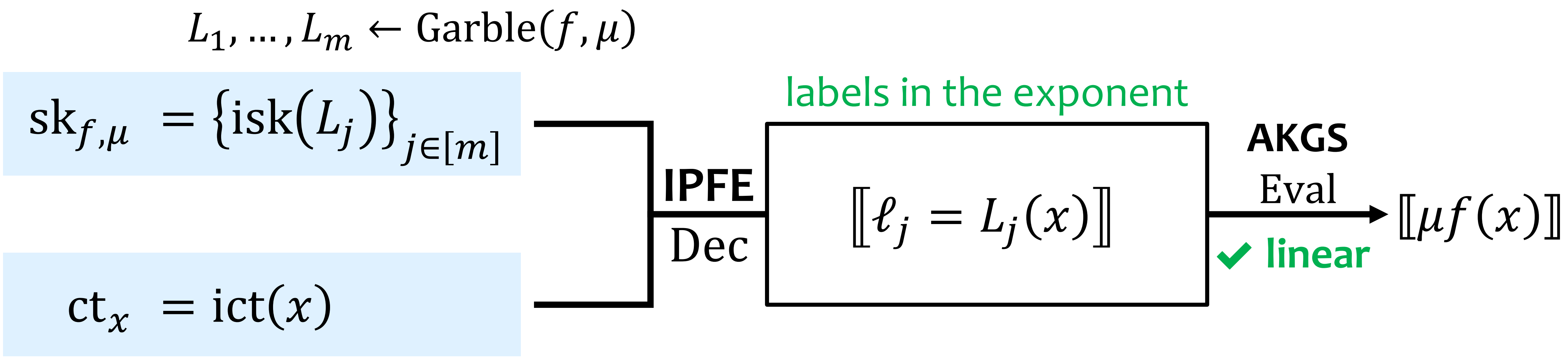
$$[\![\mu f(x)]\!]$$

Proof needs **function-hiding**,
but **FH $\Longrightarrow$ non-succinct.**

Intuitions for Security.

- IPFE $\Longrightarrow$ only ℓ_j's are revealed
- AKGS $\Longrightarrow$ only $\mu f(x)$ is revealed

1-ABE via AKGS and IPFE



Idea. Use (public-key) IPFE without function-hiding.

must hide L_j's for security $\Longrightarrow$ L_j's in IPFE **ciphertext**, x in IPFE **key**

1-ABE via AKGS and IPFE

$$L_1, \ldots, L_m \leftarrow \mathrm{Garble}(f, \mu)$$

$$\mathrm{ct}_{f,\mu} = \{\mathrm{ict}(L_j)\}_{j \in [m]}$$

$$\mathrm{sk}_x = \mathrm{isk}(x)$$

IPFE

Dec

labels in the exponent

$$[\![\ell_j = L_j(x)]\!]$$

AKGS

Eval

✔ **linear**

$$[\![\mu f(x)]\!]$$

public-key IPFE $\Longrightarrow$ public-key **CP-**1-ABE

succinct isk $\Longrightarrow$ succinct sk in CP-1-ABE

(eventually $\Longrightarrow$ succinct ct in full-fledged KP-ABE)

Fact. [ALS16] public-key IPFE has **succinct key.**

1-ABE via AKGS and IPFE

$$L_1, \dots, L_m \leftarrow \mathrm{Garble}(f, \mu)$$

$$\mathrm{ct}_{f,\mu} = \{\mathrm{ict}(L_j)\}_{j \in [m]}$$

$$\mathrm{sk}_x = \mathrm{isk}(x)$$

IPFE

Dec

labels in the exponent

$$[\![\ell_j = L_j(x)]\!]$$

AKGS

Eval

✔ **linear**

$$[\![\mu f(x)]\!]$$

"Selective" Security (x then f). **Easy**

Adaptive Security (f then x). **Very Tricky**

Security: x then f

IND-CPA of IPFE

AKGS Security

sk_x	$\mathrm{isk}(\ x \quad 1\ )$		$\mathrm{isk}(\ x \quad 1\ )$		$\mathrm{isk}(\ x \quad 1\ )$
		$\approx$		$\equiv$	
$\mathsf{ct}_{f,\mu}$	$\{\ \mathrm{ict}(\ L_j \quad 0\ )\ \}$		$\{\ \mathrm{ict}(\ 0 \quad \ell_j\ )\ \}$		$\{\ \mathrm{ict}(\ 0 \quad \ell_j\ )\ \}$
	$\{L_j\} \leftarrow \mathrm{Garble}(f, \mu)$		$\{L_j\} \leftarrow \mathrm{Garble}(f, \mu)$ $\{\ell_j\} \leftarrow \{L_j(x) = \langle L_j, x\rangle\}$		$\{\ell_j\} \leftarrow \mathrm{Sim}(f, x, \underbrace{\boxed{\mu f(x)}}_{=0})$

Want. μ is hidden.

Done. μ is hidden.

What about f then x?

Security: f then x (Naïve Attempt)

IND-CPA of IPFE

$\mathsf{ct}_{f,\mu}$	$\{\, \mathsf{ict}(\ L_j \quad 0 \ ) \,\}$	$\times$	$\{\, \mathsf{ict}(\ 0 \quad \ell_j\ ) \,\}$
sk_x	$\mathsf{isk}(\ x \quad 1\ )$		$\mathsf{isk}(\ x \quad 1\ )$

$\{L_j\} \leftarrow \mathsf{Garble}(f, \mu)$

$\{L_j\} \leftarrow \mathsf{Garble}(f, \mu)$
$\{\ell_j\} \leftarrow \{L_j(x) = \langle L_j, x \rangle\}$

x is unknown when generating $\mathsf{ct}_{f,\mu}$.

Want. μ is hidden.

Security: f then x (Naïve Attempt)

$$\mathrm{ct}_{f,\mu} \quad \begin{matrix} \mathrm{ict}(& L_1 & 0 & 0 & \cdots & 0 &) \\ \mathrm{ict}(& L_2 & 0 & 0 & \cdots & 0 &) \\ & \vdots & \vdots & \vdots & \ddots & \vdots & \\ \mathrm{ict}(& L_m & 0 & 0 & \cdots & 0 &) \end{matrix} \qquad \begin{matrix} \mathrm{ict}(& 0 & 1 & 0 & \cdots & 0 &) \\ \mathrm{ict}(& 0 & 0 & 1 & \cdots & 0 &) \\ & \vdots & \vdots & \vdots & \ddots & \vdots & \\ \mathrm{ict}(& 0 & 0 & 0 & \cdots & 1 &) \end{matrix}$$

$$\mathrm{sk}_x \quad \mathrm{isk}(\ x \ \ 0 \ \ 0 \ \cdots \ 0\ ) \qquad \mathrm{isk}(\ x \ \ \ell_1 \ \ \ell_2 \ \cdots \ \ell_m\ )$$

too many values hardwired

using FH to hide hardwired labels in key

either $\Longrightarrow$ non-succinct key

Hardwiring Less: Piecewise Security [LL20, EC]

$$L_1, \dots, L_m \leftarrow \mathrm{Garble}(f, \mu)$$

Labels are **marginally random** given **subsequent label functions**.

For $j > 1$ and all x:

$$\left(L_j(x), L_{j+1}, \dots, L_m\right) \equiv \left(\$, L_{j+1}, \dots, L_m\right).$$

piecewise security

linear constraint over ℓ_j's

ℓ_1 can be solved from $\mathrm{Eval}(f, x, \ell_1, \dots, \ell_m) = \mu f(x)$.

$$\ell_1 \leftarrow \mathrm{RevCompute}(f, x, \mu f(x), \ell_2, \dots, \ell_m) .$$

Fact. There exists piecewise secure AKGS for ABP [IW14].

Hardwiring Less: Special Simulation Structure

$$\{\ \ell_1 \leftarrow L_1(x) \quad \ell_2 \leftarrow L_2(x) \quad \ell_3 \leftarrow L_3(x) \quad \cdots \quad \ell_m \leftarrow L_m(x)\ \}$$

$$\equiv\ \{\ \ell_1 \leftarrow \mathrm{RC}(\cdots) \quad \ell_2 \leftarrow \$ \quad \ell_3 \leftarrow \$ \quad \cdots \quad \ell_m \leftarrow \$\ \}$$

	real	**simulation**	
$\mathrm{ct}_{f,\mu}$	$\mathrm{ict}(\ L_1 \quad 0 \quad 0\ )$	$\mathrm{ict}(\ 0 \quad 0 \quad 1\ )$	no need for x
	$\{\ \mathrm{ict}(\ L_j \quad 0 \quad 0\ )\ \}_{j>1}$	$\{\ \mathrm{ict}(\ 0 \quad \ell_j \quad 0\ )\ \}_{j>1}$	$\boxed{\ell_j \leftarrow \$}$
sk_x	$\mathrm{isk}(\ x \quad 1 \quad 0\ )$	$\mathrm{isk}(\ x \quad 1 \quad \ell_1\ )$	$\ell_1 \leftarrow \mathrm{RC}(\cdots)$

Modified Proof with Less Hardwiring

	real $\xrightarrow{\text{IPFE}}$	**hybrid 1** $\xrightarrow{\text{one by one}}$	**simulation**
$\mathsf{ct}_{f,\mu}$	$\mathsf{ict}(L_1\ 0\ 0\ 0)$	$\mathsf{ict}(0\ 0\ 1\ 0)$	$\mathsf{ict}(0\ 0\ 1\ 0)$
	$\{\mathsf{ict}(L_j\ 0\ 0\ 0)\}_{j>1}$	$\{\mathsf{ict}(L_j\ 0\ 0\ 0)\}_{j>1}$	$\{\mathsf{ict}(0\ \ell_j\ 0\ 0)\}_{j>1}$
sk_x	$\mathsf{isk}(x\ 1\ 0\ 0)$	$\mathsf{isk}(x\ 1\ \ell_1\ 0)$	$\mathsf{isk}(x\ 1\ \ell_1\ 0)$

$\ell_j \leftarrow L_j(x)$ ⟶ $\ell_j \leftarrow \$$

$\ell_1 \leftarrow L_1(x) = \mathsf{RC}(\cdots)$ $\qquad$ $\ell_1 \leftarrow \mathsf{RC}(\cdots)$

Next Step. Switch label functions L_j to random labels ℓ_j one by one.

Modified Proof with Less Hardwiring

	hybrid j	$\xrightarrow{\text{IPFE}}$	$\xrightarrow{\text{IPFE}}$ hybrid $j+1$
$\mathsf{ct}_{f,\mu}$	$\mathsf{ict}(\ 0\ \ 0\ \ 1\ \ 0\ )$		
	$_{j'<j}\{\ \mathsf{ict}(\ 0\ \ \ell_{j'}\ \ 0\ \ 0\ )\ \}$		
	$\mathsf{ict}(\ L_j\ \ 0\ \ 0\ \ 0\ )$	$\mathsf{ict}(\ 0\ \ 0\ \ 0\ \ 1\ )$	$\mathsf{ict}(\ 0\ \ \ell_j\ \ 0\ \ 0\ )$
	$_{j'>j}\{\ \mathsf{ict}(L_{j'}\ \ 0\ \ 0\ \ 0\ )\ \}$		
sk_x	$\mathsf{isk}(\ x\ \ 1\ \ \ell_1\ \ 0\ )$	$\mathsf{isk}(\ x\ \ 1\ \ \ell_1\ \ \ell_j\ )$	$\mathsf{isk}(\ x\ \ 1\ \ \ell_1\ \ 0\ )$

$$\ell_j \leftarrow L_j(x) \xrightarrow[\text{randomness}]{\text{marginal}} \ell_j \leftarrow \$$$

✓ only two values hardwired in key

✗ using FH to hide hardwired labels in key

Replacing Function-Hiding: Simulation Security

Simulation Security of Public-Key IPFE **satisfied by [ALS16, ALMT20]**

real		**simulation**
impk		$\widetilde{\mathrm{impk}}$
$\{\mathrm{isk}(\boldsymbol{v}_q)\}_{q\le Q_1}$		$\{\widetilde{\mathrm{isk}}(\,\boldsymbol{v}_q \mid \emptyset\,)\}_{q\le Q_1}$
$\mathrm{ict}(\boldsymbol{u})$	$\approx$	$\widetilde{\mathrm{ict}}(\,\emptyset \mid \langle \boldsymbol{u}, \boldsymbol{v}_1\rangle, \dots, \langle \boldsymbol{u}, \boldsymbol{v}_{Q_1}\rangle\,)$
$\{\mathrm{isk}(\boldsymbol{v}_q)\}_{Q_1<q\le Q}$		$\{\widetilde{\mathrm{isk}}(\,\boldsymbol{v}_q \mid \boxed{\langle \boldsymbol{u}, \boldsymbol{v}_q\rangle})\}_{Q_1<q\le Q}$

hardwires values in key

Naturally generalizes to **any constant number** of simulated ciphertexts.

One-Liner. Simulator uses an inner product **only when** it can be decrypted.

Previous Proof with Function-Hiding

	real	hybrid 1	simulation
	$\xrightarrow{\text{IPFE}}$	$\xrightarrow{\text{one by one}}$	
$\mathrm{ct}_{f,\mu}$	$\mathrm{ict}(L_1\ \ 0\ \ 0\ \ 0)$	$\mathrm{ict}(0\ \ 0\ \ 1\ \ 0)$	$\mathrm{ict}(0\ \ 0\ \ 1\ \ 0)$
	$\{\mathrm{ict}(L_j\ \ 0\ \ 0\ \ 0)\}_{j>1}$	$\{\mathrm{ict}(L_j\ \ 0\ \ 0\ \ 0)\}_{j>1}$	$\{\mathrm{ict}(0\ \ \ell_j\ \ 0\ \ 0)\}_{j>1}$
sk_x	$\mathrm{isk}(x\ \ 1\ \ 0\ \ 0)$	$\mathrm{isk}(x\ \ 1\ \ \ell_1\ \ 0)$	$\mathrm{isk}(x\ \ 1\ \ \ell_1\ \ 0)$

$\ell_j \leftarrow L_j(x)$ ⟶ $\ell_j \leftarrow \$$

$\ell_1 \leftarrow L_1(x) = \mathrm{RC}(\cdots)$ $\qquad$ $\ell_1 \leftarrow \mathrm{RC}(\cdots)$

✗ using FH to hide hardwired ℓ_1 in key

Modified Proof with Simulation

	real $\xrightarrow{\text{IPFE Simulation}}$	hybrid 1 $\xrightarrow{\text{one by one}}$	simulation
$\mathrm{ct}_{f,\mu}$	$\mathrm{ict}(L_1\ \ 0)$	$\widetilde{\mathrm{ict}}(\ \emptyset\ \vert\ \emptyset\ )$	$\widetilde{\mathrm{ict}}(\ \emptyset\ \vert\ \emptyset\ )$
	$\{\mathrm{ict}(L_j\ \ 0)\}_{j>1}$	$\{\mathrm{ict}(L_j\ \ 0\ \ )\}_{j>1}$	$\{\mathrm{ict}(0\ \ \ell_j\ \ )\}_{j>1}$
sk_x	$\mathrm{isk}(x\ \ 1)$	$\widetilde{\mathrm{isk}}(x\ \ 1\ \vert\ \ell_1\ )$	$\widetilde{\mathrm{isk}}(x\ \ 1\ \vert\ \ell_1\ )$

$\ell_j \leftarrow L_j(x)$ ⟶ $\ell_j \leftarrow \$$

$\ell_1 \leftarrow L_1(x) = \mathrm{RC}(\cdots)$ $\qquad$ $\ell_1 \leftarrow \mathrm{RC}(\cdots)$

✔ simulating with ℓ_1 hardwired in key

Modified Proof with Simulation

hybrid j $\xrightarrow{\text{Simulate one more ciphertext}}$ $\xrightarrow{\text{Simulate one fewer ciphertext}}$ **hybrid $j+1$**

	hybrid j		hybrid $j+1$
$\mathrm{ct}_{f,\mu}$	$\widetilde{\mathrm{ict}}(\ \emptyset \mid \emptyset\ )$		
	$\{\mathrm{ict}(\ 0\ \ \ell_{j'}\ )\}_{j'<j}$		
	$\mathrm{ict}(\ L_j\ \ 0\ )$	$\widetilde{\mathrm{ict}}(\ \emptyset \mid \emptyset\ )$	$\mathrm{ict}(\ 0\ \ \ell_j\ )$
	$\{\mathrm{ict}(L_{j'}\ \ 0\ )\}_{j'>j}$		
sk_x	$\widetilde{\mathrm{isk}}(\ x\ \ 1 \mid \ \ell_1\ )$	$\widetilde{\mathrm{isk}}(\ x\ \ 1 \mid \ell_1, \ell_j)$	$\widetilde{\mathrm{isk}}(\ x\ \ 1 \mid \ \ell_1\ )$

$$\ell_j \leftarrow L_j(x) \xrightarrow[\text{randomness}]{\text{marginal}} \ell_j \leftarrow \$$$

✓ only two simulated ciphertexts

Modified Proof with Simulation

	hybrid j		hybrid $j+1$
$\mathrm{ct}_{f,\mu}$	$\widetilde{\mathrm{ict}}(\ \emptyset \mid \emptyset\ )$	$\widetilde{\mathrm{ict}}(\ \emptyset \mid \emptyset\ )$	$\widetilde{\mathrm{ict}}(\ \emptyset \mid \emptyset\ )$
	$_{j'<j}\{\ \mathrm{ict}(\ 0\ \ \ell_{j'}\ )\ \}$		
	$\mathrm{ict}(\ L_j\ \ 0\ )$	$\widetilde{\mathrm{ict}}(\ \emptyset \mid \emptyset\ )$	$\mathrm{ict}(\ 0\ \ \ell_j\ )$
	$_{j'>j}\{\ \mathrm{ict}(L_{j'}\ \ 0\ )\ \}$		
sk_x	$\widetilde{\mathrm{isk}}(\ x\ \ 1 \mid \ell_1\ )$	$\widetilde{\mathrm{isk}}(\ x\ \ 1 \mid \ell_1, \ell_j)$	$\widetilde{\mathrm{isk}}(\ x\ \ 1 \mid \ell_1\ )$

Simulate one more ciphertext?

Simulate one fewer ciphertext?

(Usual) Simulation real ≈ simulation

Needed in Proof real ≈ simulating **one** ≈ simulating **two**

Insufficiency of (Usual) Simulation Security

Simulation Security for 2 Ciphertexts

$$\{\mathrm{ict}(\boxed{\boldsymbol{u}_1}), \mathrm{ict}(\boldsymbol{u}_2), \mathrm{isk}(\boldsymbol{v})\} \approx \{\widetilde{\mathrm{ict}}(\emptyset), \widetilde{\mathrm{ict}}(\emptyset), \widetilde{\mathrm{isk}}(\boldsymbol{v} \mid \langle \boldsymbol{v}, \boxed{\boldsymbol{u}_1} \rangle, \langle \boldsymbol{v}, \boldsymbol{u}_2 \rangle)\}$$

$$\overset{?}{\approx}$$

$$\{\mathrm{ict}(\boxed{\boldsymbol{u}_1}), \mathrm{ict}(\boldsymbol{u}_2), \mathrm{isk}(\boldsymbol{v})\} \approx \{\widetilde{\mathrm{ict}}(\emptyset), \mathrm{ict}(\boldsymbol{u}_2), \widetilde{\mathrm{isk}}(\boldsymbol{v} \mid \langle \boldsymbol{v}, \boxed{\boldsymbol{u}_1} \rangle)\}$$

To use hybrid argument, must know $\boldsymbol{u}_1$ (even before simulating keys)! ☹

Needed in Proof

No concept of $\boldsymbol{u}_1$! ☺

$$\{\widetilde{\mathrm{ict}}_1(\emptyset), \mathrm{ict}_2(\boldsymbol{u}_2), \widetilde{\mathrm{isk}}(\boldsymbol{v} \mid \boxed{\text{desired inner product with } \mathrm{ict}_1})\}$$

$$\approx \{\widetilde{\mathrm{ict}}_1(\emptyset), \widetilde{\mathrm{ict}}_2(\emptyset), \widetilde{\mathrm{isk}}(\boldsymbol{v} \mid \boxed{\text{desired inner product with } \mathrm{ict}_1}, \langle \boldsymbol{u}_2, \boldsymbol{v} \rangle)\}$$

Gradual Simulation Security

T-Ciphertext Simulation

$$\widetilde{\text{impk}} \qquad\qquad \widetilde{\text{impk}}$$

Adversary supplies input to simulator!

$$\{\widetilde{\text{isk}}(\cdots), \widetilde{\text{ict}}(\cdots)\} \qquad\qquad \{\widetilde{\text{isk}}(\cdots), \widetilde{\text{ict}}(\cdots)\}$$

Simulate one more ciphertext

$$\text{ict}(\boldsymbol{u}) \approx \widetilde{\text{ict}}(\emptyset|\cdots)$$

$$\{\widetilde{\text{isk}}(\boldsymbol{v}_q|\cdots,\cdots), \widetilde{\text{ict}}(\cdots)\} \qquad \{\widetilde{\text{isk}}(\boldsymbol{v}_q|\cdots,\langle\boldsymbol{u},\boldsymbol{v}_q\rangle,\cdots), \widetilde{\text{ict}}(\cdots)\}$$

Insert inner products at the right position.

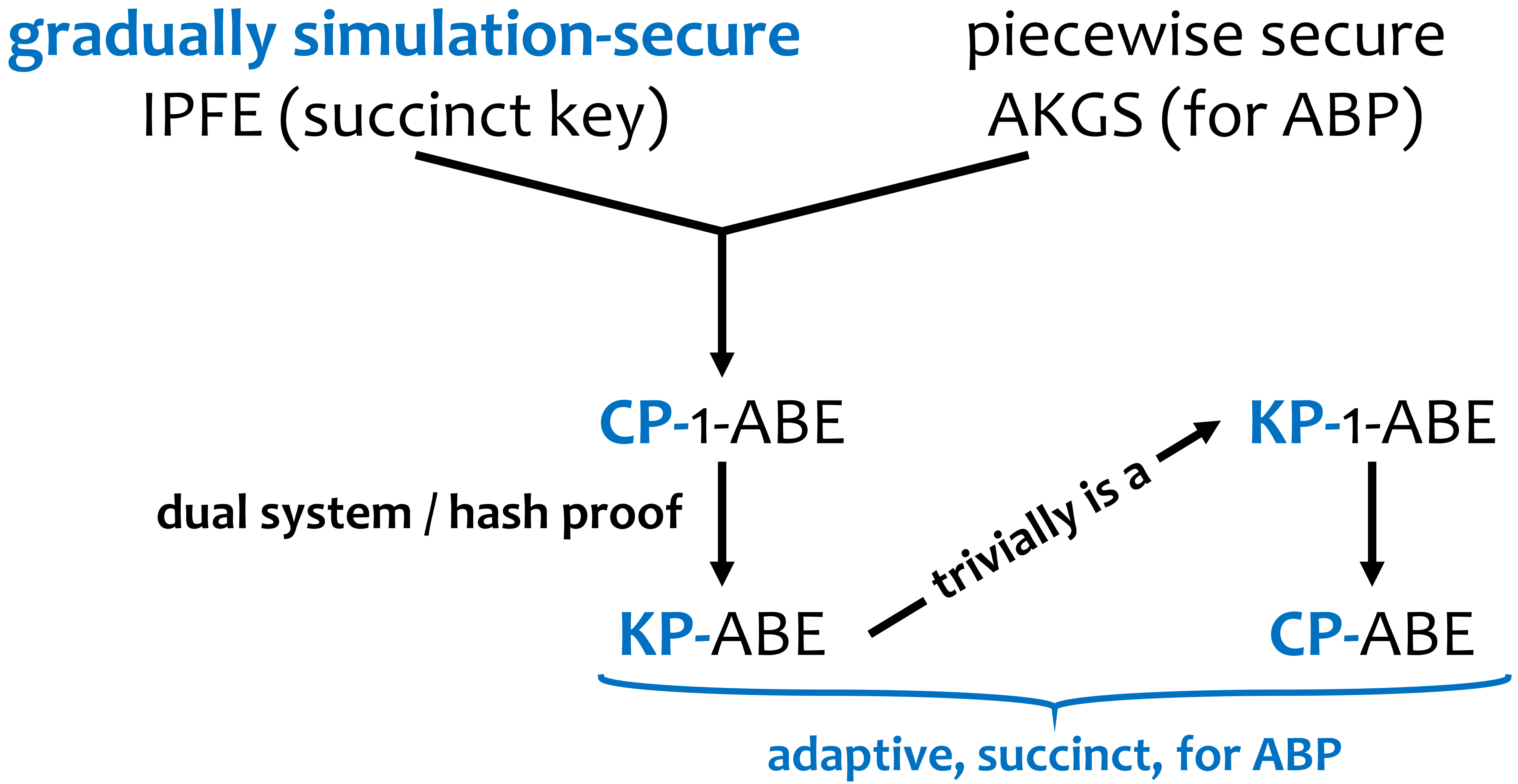
Constraint. $\#[\widetilde{\text{ict}}\text{'s}] \le T$ in the right hybrid.

Gradual Simulation Security

Theorem. [ALS16] can be modified for gradual simulation security.

Key size **only** grows with #[simulated ciphertexts] $= T$,
not vector dimension (~ attribute length).
We only need $T = 2$ ($\Longrightarrow$ **succinctness**).

Bonus Fact. Generic transformation (preserving key succinctness):
selective IND-CPA → (adaptive) gradual simulation security.



Thank you! ia.cr/2020/1139