Compact FE for Unbounded Attribute-Weighted Sums for Logspace from SXDH

Pratish Datta
NTT Research

Tapas Pal
NTT SIL

Katsuyuki Takashima
Waseda University

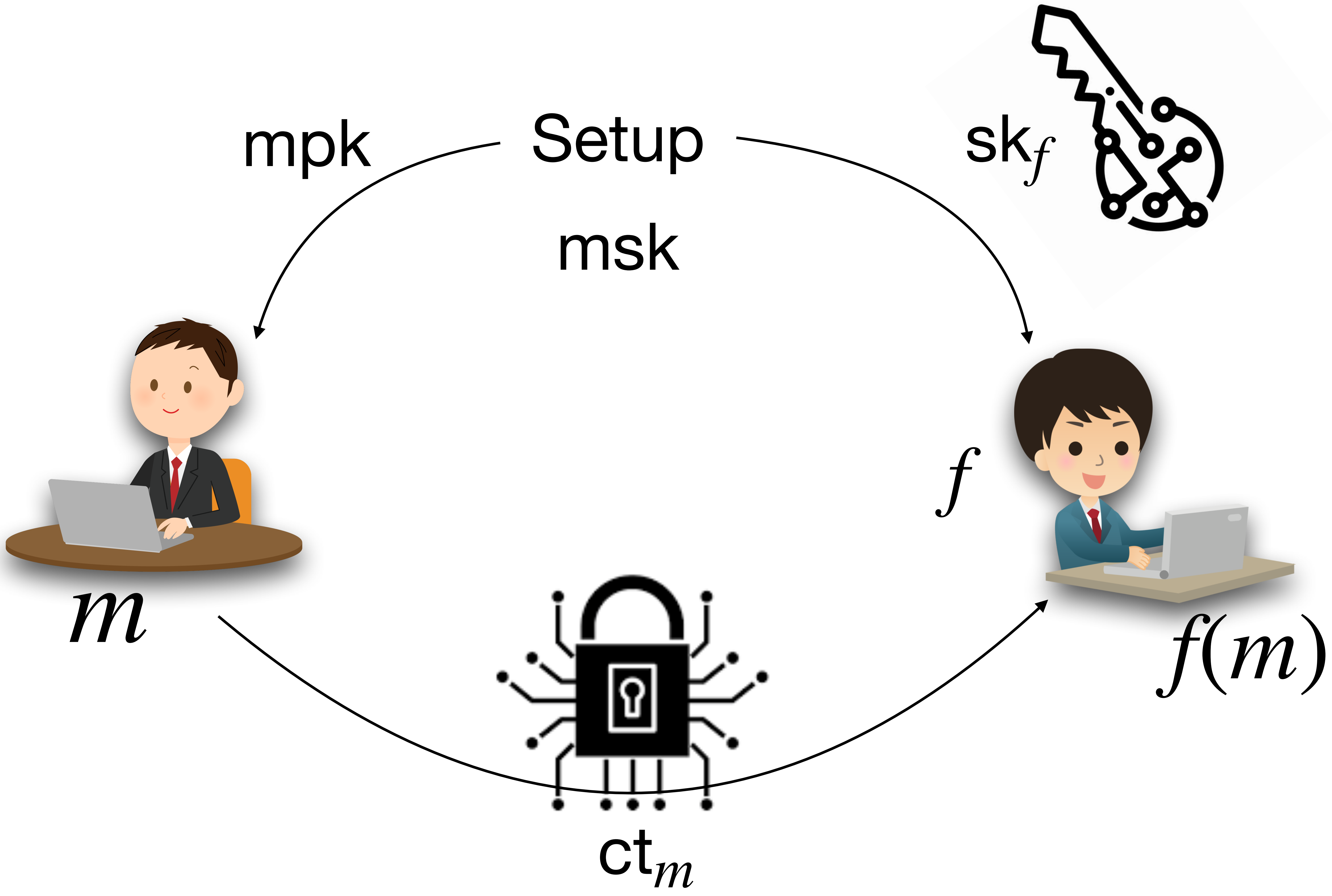
Functional Encryption [BSW11]

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$

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

$\mathsf{Enc}(\mathsf{mpk}, m) \to \mathsf{ct}_m$

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_m) \to f(m)$



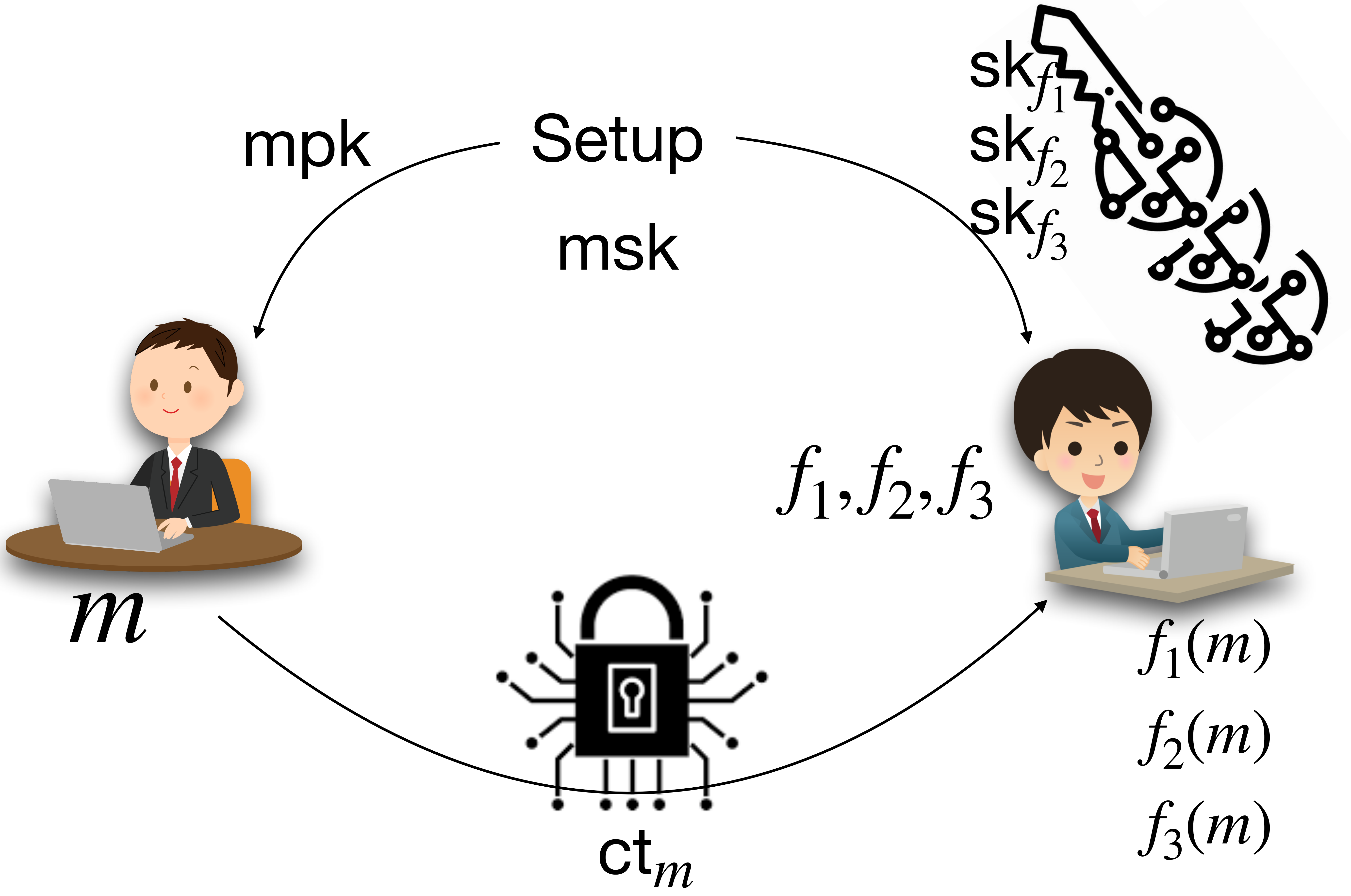
Simulation Security of FE [O'Niell11,BSW11]

Setup(1^λ) → (mpk, msk)

KeyGen(msk, f) → sk$_f$

Enc(mpk, m) → ct$_m$

Dec(sk$_f$, ct$_m$) → $f(m)$



Adaptive Simulation Security

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$

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

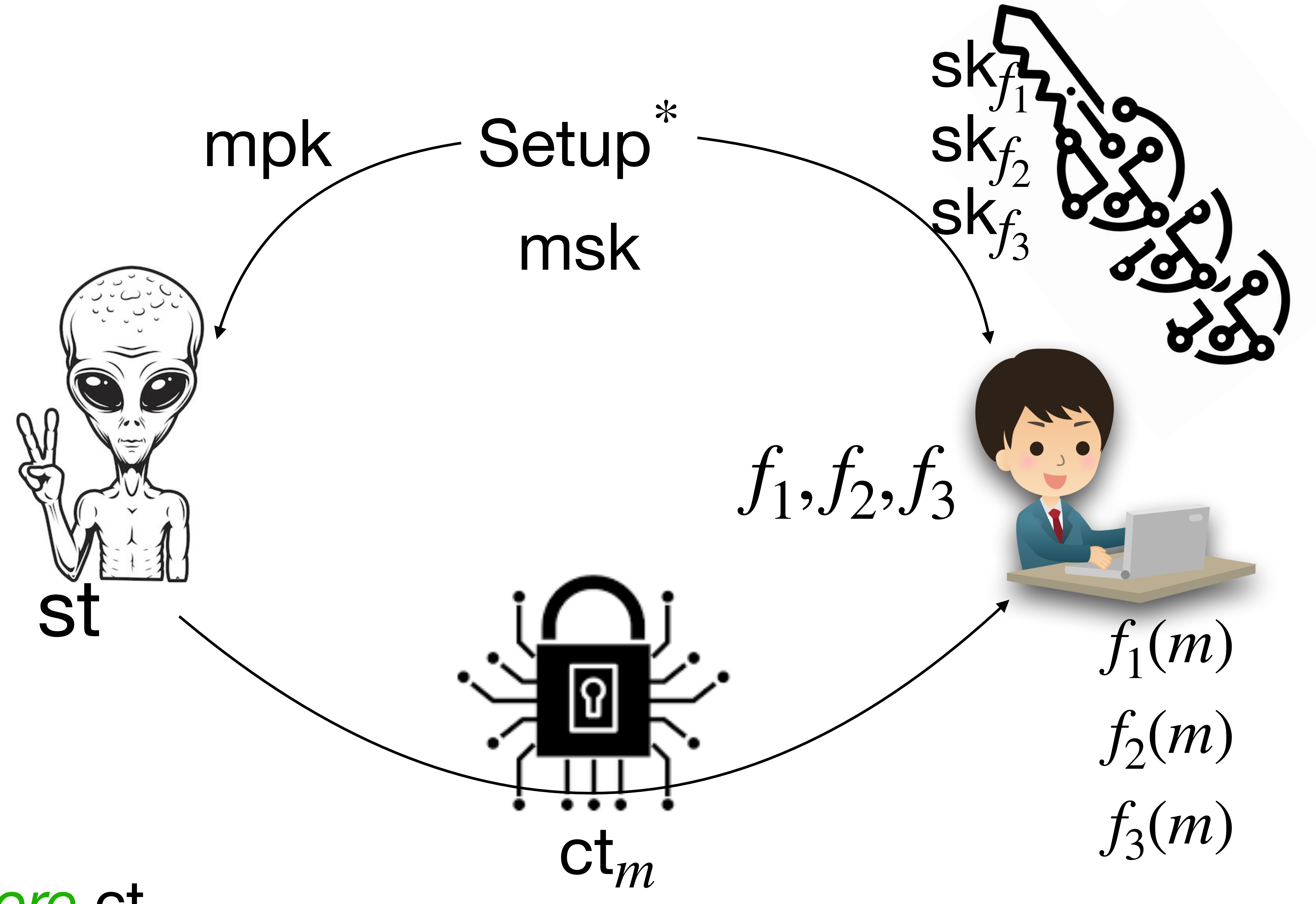
$\mathsf{Enc}(\mathsf{mpk}, m) \to \mathsf{ct}_m$

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_m) \to f(m)$

$\mathsf{Setup}*(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk}^*)$

$\mathsf{KeyGen}_0^*(\mathsf{msk}^*, f) \to \mathsf{sk}_f$

$\mathsf{Enc}^*(\mathsf{mpk}, \mathsf{msk}^*, \mathsf{st}) \to \mathsf{ct}_m$



before ct_m

st contains $\{f_i(m)\}_i$ for already queried $\{f_i\}_i$

Adaptive Simulation Security

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$

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

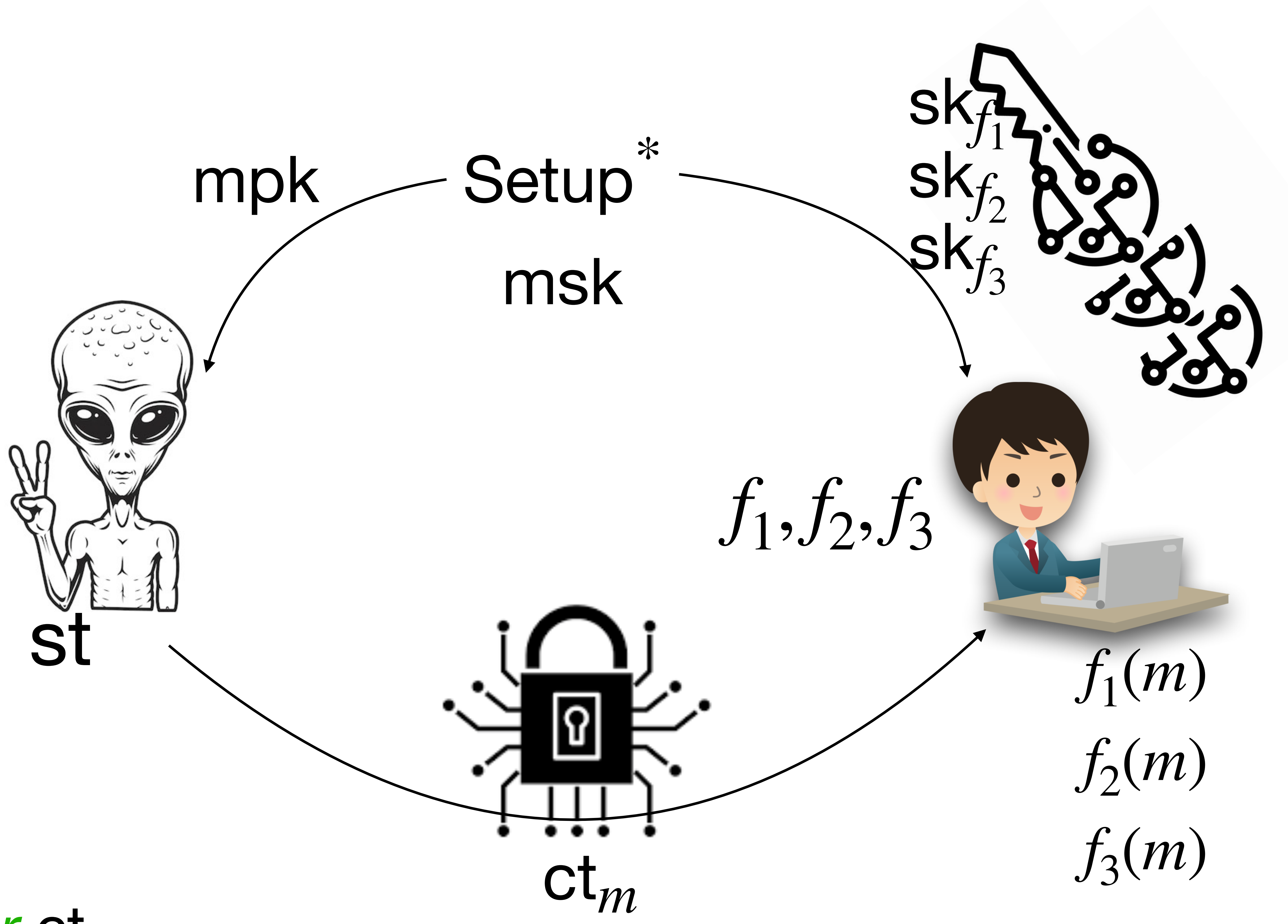
$\mathsf{Enc}(\mathsf{mpk}, m) \to \mathsf{ct}_m$

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_m) \to f(m)$

$\mathsf{Setup}*(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk}^*)$

$\mathsf{KeyGen}_1^*(\mathsf{msk}^*, f, f(m)) \to \mathsf{sk}_f$

$\mathsf{Enc}^*(\mathsf{mpk}, \mathsf{msk}^*, \mathsf{st}) \to \mathsf{ct}_m$



after ct_m

st contains $\{f_i(m)\}_i$ for already queried $\{f_i\}_i$

Adaptive Simulation Security

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$

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

$\mathsf{Enc}(\mathsf{mpk}, m) \to \mathsf{ct}_m$

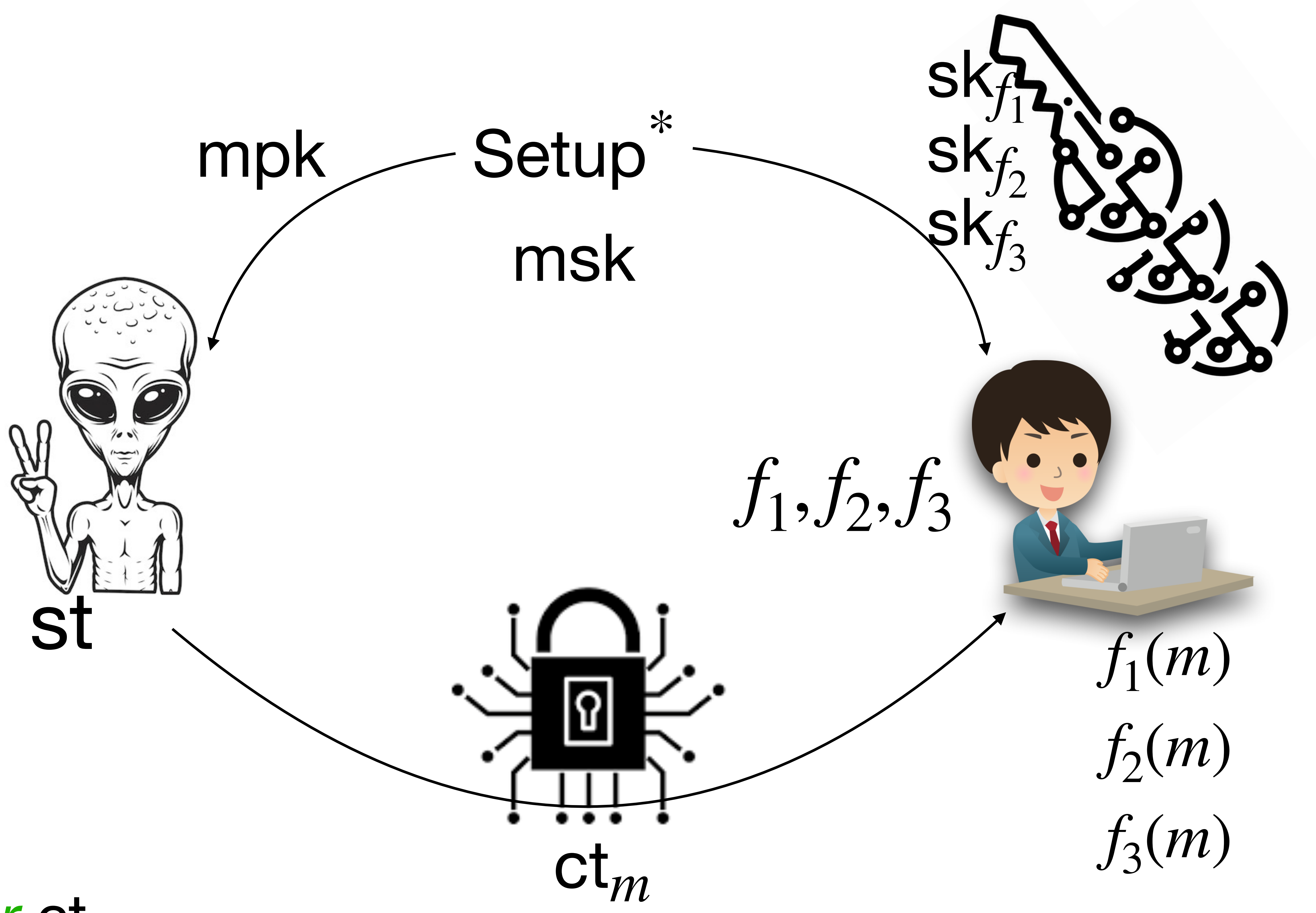
$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_m) \to f(m)$

$$\approx_c$$

$\mathsf{Setup}*(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk}^*)$

$\mathsf{KeyGen}_1^*(\mathsf{msk}^*, f, f(m)) \to \mathsf{sk}_f$ *after* ct_m

$\mathsf{Enc}^*(\mathsf{mpk}, \mathsf{msk}^*, \mathsf{st}) \to \mathsf{ct}_m$



mpk

Setup^*

msk

sk_{f_1}

sk_{f_2}

sk_{f_3}

f_1, f_2, f_3

st

ct_m

$f_1(m)$

$f_2(m)$

$f_3(m)$

st contains $\{f_i(m)\}_i$ for already queried $\{f_i\}_i$

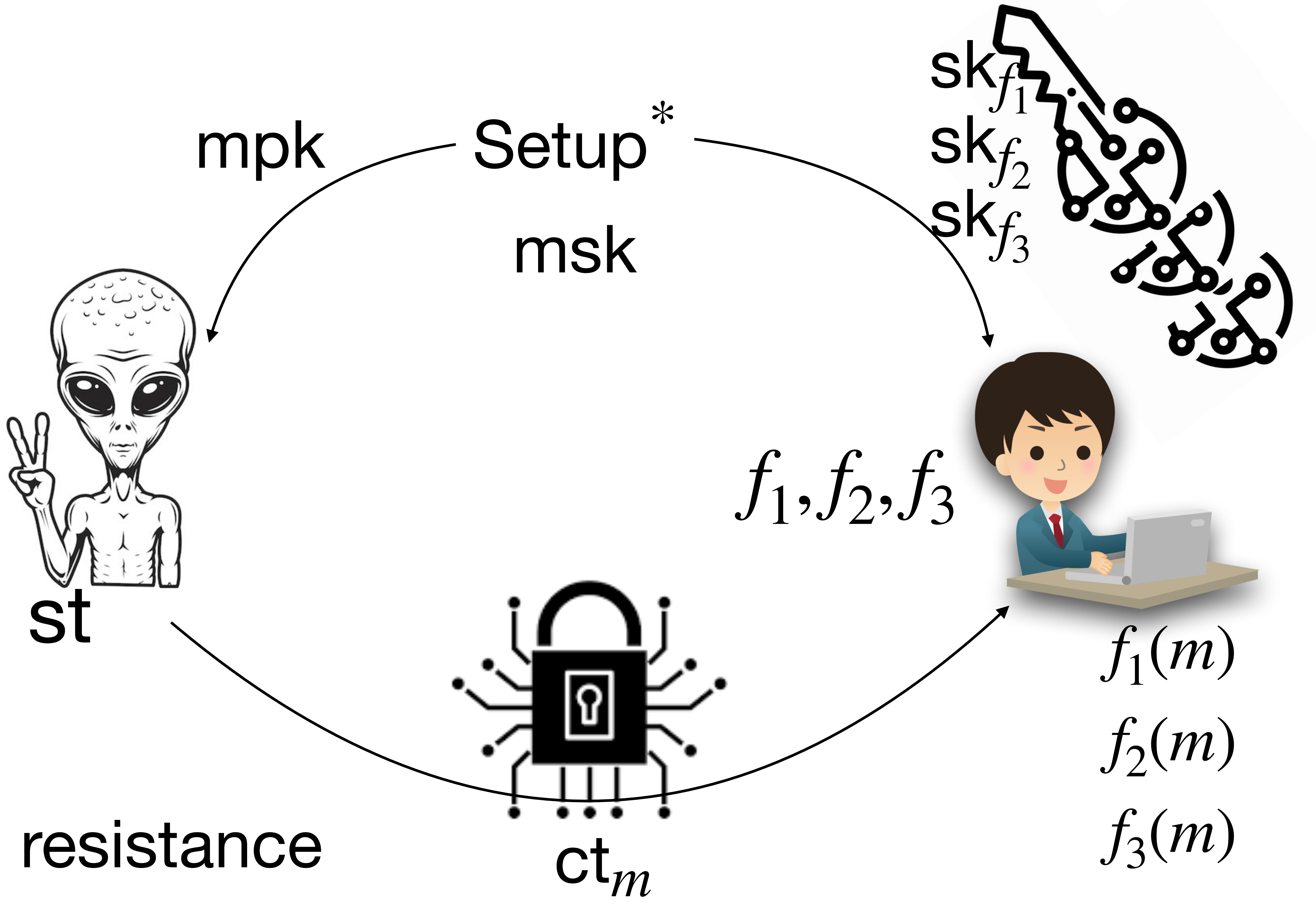
Adaptive Simulation Security: Bounded or Full Collusion resistance

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$

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

$\mathsf{Enc}(\mathsf{mpk}, m) \to \mathsf{ct}_m$

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_m) \to f(m)$



- *bounded* #sk_{f_i} → bounded collusion resistance
- *any* #sk_{f_i} → full collusion resistance

FE for Various Function Classes

General Class: TMs or All Circuits	
[GKP+13,GGG+14,BGJS15,AJ15, BKS16,AR17,AM18,AMVY21,JLS22….] • Inefficient and complex • bounded collusion-resistant • Assumptions: IO or SubExp LWE • Not yet practical	

FE for Various Function Classes

General Class: TMs or All Circuits	Specific Class: Linear, Quadratic or its variants
[GKP+13,GGG+14,BGJS15,AJ15, BKS16,AR17,AM18,AMVY21,JLS22….]	[ABDP15,ALS16,BCFG17,TT18,G20, GQ20,Wee20,ACGU20,AGW20,DP21,MKMS22….]
• Inefficient and complex	• Efficient and simple
• bounded collusion-resistant	• Full collusion-resistant
• Assumptions: IO or SubExp LWE	• Assumptions: DDH, k-Lin or LWE
• Not yet practical	• Applicable in practice

FE for Various Function Classes

General Class: TMs or All Circuits

[GKP+13,GGG+14,BGJS15,AJ15, BKS16,AR17,AM18,AMVY21,JLS22….]

- Inefficient and complex
- bounded collusion-resistant
- Assumptions: IO or SubExp LWE
- Not yet practical

Specific Class: Linear, Quadratic or its variants

[ABDP15,ALS16,BCFG17,TT18,G20, GQ20,Wee20,ACGU20,AGW20,DP21,MKMS22….]

- Efficient and simple
- Full collusion-resistant
- Assumptions: DDH, k-Lin or LWE
- Applicable in practice

- This work advances FE for a specific class

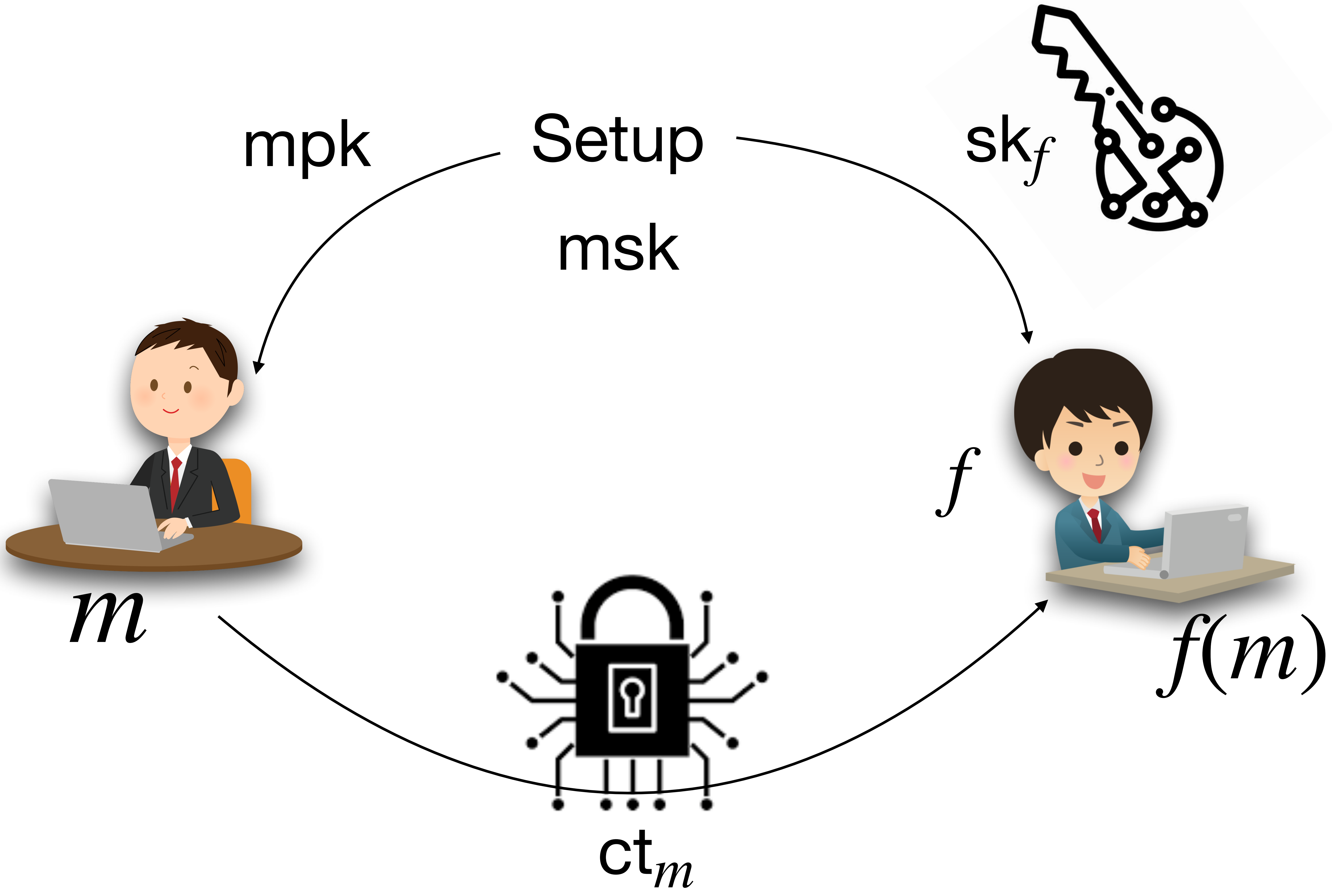
Functional Encryption for Attribute-Weighted Sums[AGW20]

$\text{Setup}(1^\lambda, 1^n, 1^m) \rightarrow (\text{mpk}, \text{msk})$

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

$\text{Enc}(\text{mpk}, m) \rightarrow \text{ct}_m$

$\text{Dec}(\text{sk}_f, \text{ct}_m) \rightarrow f(m)$



- Function: $f : \mathbb{Z}_p^m \rightarrow \mathbb{Z}_p^n$
- Message: $m = (\mathbf{x}, \mathbf{z}) \in \mathbb{Z}_p^{m \times n}$
- Output: $f(m) = f(\mathbf{x}) \cdot \mathbf{z}$

$\mathbf{x}$ is public, $\mathbf{z}$ is private

Functional Encryption for Attribute-Weighted Sums (AWS)

$$f(m) = f(\mathbf{x}) \cdot \mathbf{z}, \quad f \in \text{ABP}$$

Prior Works

- Function class = ABP
- Setup
  $|\text{mpk}| = O(|\mathbf{x}|, |\mathbf{z}|)$
- $|\text{ct}_m| = O(|\mathbf{z}|)$
- AD-SIM [D**P**21]
  $|\text{ct}_m| = O(|\mathbf{x}|, |\mathbf{z}|)$

Functional Encryption for Attribute-Weighted Sums (AWS)

$$f(m) = f(\mathbf{x}) \cdot \mathbf{z}, \quad f \in \text{ABP}$$

Prior Works

- Function class = ABP
- Setup
  $|\text{mpk}| = O(|\mathbf{x}|, |\mathbf{z}|)$
- $|\text{ct}_m| = O(|\mathbf{z}|)$
- AD-SIM [D**P**21]
  $|\text{ct}_m| = O(|\mathbf{x}|, |\mathbf{z}|)$

Applications

- IPFE: $f(\mathbf{x}) = \mathbf{y}$

  $f(m) = \mathbf{y} \cdot \mathbf{z}$
- ABE: $f(\mathbf{x}) = 1/0, \ \mathbf{z} = M$

  $f(m) = M$ if f satisfies $\mathbf{x}$
- AB-IPFE: $f(\mathbf{x}) = \mathbf{y}g(\mathbf{x})$

  $f(m) = \mathbf{y} \cdot \mathbf{z}$ if $\mathbf{x}$ satisfies g

Functional Encryption for Attribute-Weighted Sums (AWS)

$$f(m) = f(\mathbf{x}) \cdot \mathbf{z}, \quad f \in \text{ABP}$$

Prior Works

- Function class = ABP
- Setup
  $|\text{mpk}| = O(|\mathbf{x}|, |\mathbf{z}|)$
- $|\text{ct}_m| = O(|\mathbf{z}|)$
- AD-SIM [D**P**21]
  $|\text{ct}_m| = O(|\mathbf{x}|, |\mathbf{z}|)$

Applications

- IPFE: $f(\mathbf{x}) = \mathbf{y}$
  $f(m) = \mathbf{y} \cdot \mathbf{z}$
- ABE: $f(\mathbf{x}) = 1/0, \ \mathbf{z} = M$
  $f(m) = M$ if f satisfies $\mathbf{x}$
- AB-IPFE: $f(\mathbf{x}) = \mathbf{y}g(\mathbf{x})$
  $f(m) = \mathbf{y} \cdot \mathbf{z}$ if $\mathbf{x}$ satisfies g

Limitations

- non-uniform, non-dynamic
- bounded Setup
  $|\text{mpk}| \neq O(\lambda)$
- $|\text{ct}_m| \neq$ input-specific
- bounded FE:
  IPFE, ABE,…

Functional Encryption for Attribute-Weighted Sums (AWS)

$$f(m) = f(\mathbf{x}) \cdot \mathbf{z},\ f \in \text{TM}$$

	Applications	Limitations
• Function class = TM • Setup $\lvert\text{mpk}\rvert = O(\lvert\mathbf{x}\rvert, \lvert\mathbf{z}\rvert)$ • $\lvert\text{ct}_m\rvert = O(\lvert\mathbf{z}\rvert)$ • AD-SIM [D**P**21] $\lvert\text{ct}_m\rvert = O(\lvert\mathbf{x}\rvert, \lvert\mathbf{z}\rvert)$	• IPFE: $f(\mathbf{x}) = \mathbf{y}$ $f(m) = \mathbf{y} \cdot \mathbf{z}$ • ABE: $f(\mathbf{x}) = 1/0,\ \mathbf{z} = M$ $f(m) = M$ if f satisfies $\mathbf{x}$ • AB-IPFE: $f(\mathbf{x}) = \mathbf{y}g(\mathbf{x})$ $f(m) = \mathbf{y} \cdot \mathbf{z}$ if $\mathbf{x}$ satisfies g	• uniform, dynamic • bounded Setup $\lvert\text{mpk}\rvert \neq O(\lambda)$ • $\lvert\text{ct}_m\rvert \neq$ input-specific • bounded FE: IPFE, ABE,…

Functional Encryption for Attribute-Weighted Sums (AWS)

$$f(m) = f(\mathbf{x}) \cdot \mathbf{z},\ f \in \text{TM}$$

- Function class = TM
- Setup
  $|\text{mpk}| = O(\lambda)$
- $|\text{ct}_m| = O(|\mathbf{z}|)$
- AD-SIM [D**P**21]
  $|\text{ct}_m| = O(|\mathbf{x}|, |\mathbf{z}|)$

Applications

- IPFE: $f(\mathbf{x}) = \mathbf{y}$
  $f(m) = \mathbf{y} \cdot \mathbf{z}$
- ABE: $f(\mathbf{x}) = 1/0,\ \mathbf{z} = M$
  $f(m) = M$ if f satisfies $\mathbf{x}$
- AB-IPFE: $f(\mathbf{x}) = \mathbf{y}g(\mathbf{x})$
  $f(m) = \mathbf{y} \cdot \mathbf{z}$ if $\mathbf{x}$ satisfies g

Limitations

- uniform, dynamic
- unbounded Setup
  $|\text{mpk}| = O(\lambda)$
- $|\text{ct}_m| \neq$ input-specific
- bounded FE:
  IPFE, ABE,…

Functional Encryption for Attribute-Weighted Sums (AWS)

$$f(m) = f(\mathbf{x}) \cdot \mathbf{z},\ f \in \text{TM}$$

	Applications	Limitations
• Function class = TM • Setup $\lvert \text{mpk} \rvert = O(\lambda)$ • $\lvert \text{ct}_m \rvert = O(\lvert \mathbf{z} \rvert, \lvert \mathbf{x} \rvert)$ • AD-SIM [D**P**21] $\lvert \text{ct}_m \rvert = O(\lvert \mathbf{x} \rvert, \lvert \mathbf{z} \rvert)$	• IPFE: $f(\mathbf{x}) = \mathbf{y}$ $f(m) = \mathbf{y} \cdot \mathbf{z}$ • ABE: $f(\mathbf{x}) = 1/0,\ \mathbf{z} = M$ $f(m) = M$ if f satisfies $\mathbf{x}$ • AB-IPFE: $f(\mathbf{x}) = \mathbf{y}g(\mathbf{x})$ $f(m) = \mathbf{y} \cdot \mathbf{z}$ if $\mathbf{x}$ satisfies g	• uniform, dynamic • unbounded Setup $\lvert \text{mpk} \rvert = O(\lambda)$ • $\lvert \text{ct}_m \rvert$ = input-specific • bounded FE: IPFE, ABE,…

Functional Encryption for Attribute-Weighted Sums (AWS)

$$f(m) = f(\mathbf{x}) \cdot \mathbf{z},\ f \in \text{TM}$$

This Work

- Function class = TM
- Setup
  $|\text{mpk}| = O(\lambda)$
- $|\text{ct}_m| = O(|\mathbf{z}|, |\mathbf{x}|)$
- AD-SIM

Applications

- IPFE: $f(\mathbf{x}) = \mathbf{y}$
  $f(m) = \mathbf{y} \cdot \mathbf{z}$
- ABE: $f(\mathbf{x}) = 1/0,\ \mathbf{z} = M$
  $f(m) = M$ if f satisfies $\mathbf{x}$
- AB-IPFE: $f(\mathbf{x}) = \mathbf{y}g(\mathbf{x})$
  $f(m) = \mathbf{y} \cdot \mathbf{z}$ if $\mathbf{x}$ satisfies g

Limitations

- uniform, dynamic
- unbounded Setup
  $|\text{mpk}| = O(\lambda)$
- $|\text{ct}_m| =$ input-specific
- bounded FE:
  IPFE, ABE,…

Functional Encryption for Attribute-Weighted Sums (AWS)

$$f(m) = f(\mathbf{x}) \cdot \mathbf{z},\ f \in \mathsf{TM}$$

This Work

- Function class = TM
- Setup
  $|\mathsf{mpk}| = O(\lambda)$
- $|\mathsf{ct}_m| = O(|\mathbf{z}|, |\mathbf{x}|)$
- AD-SIM

Applications

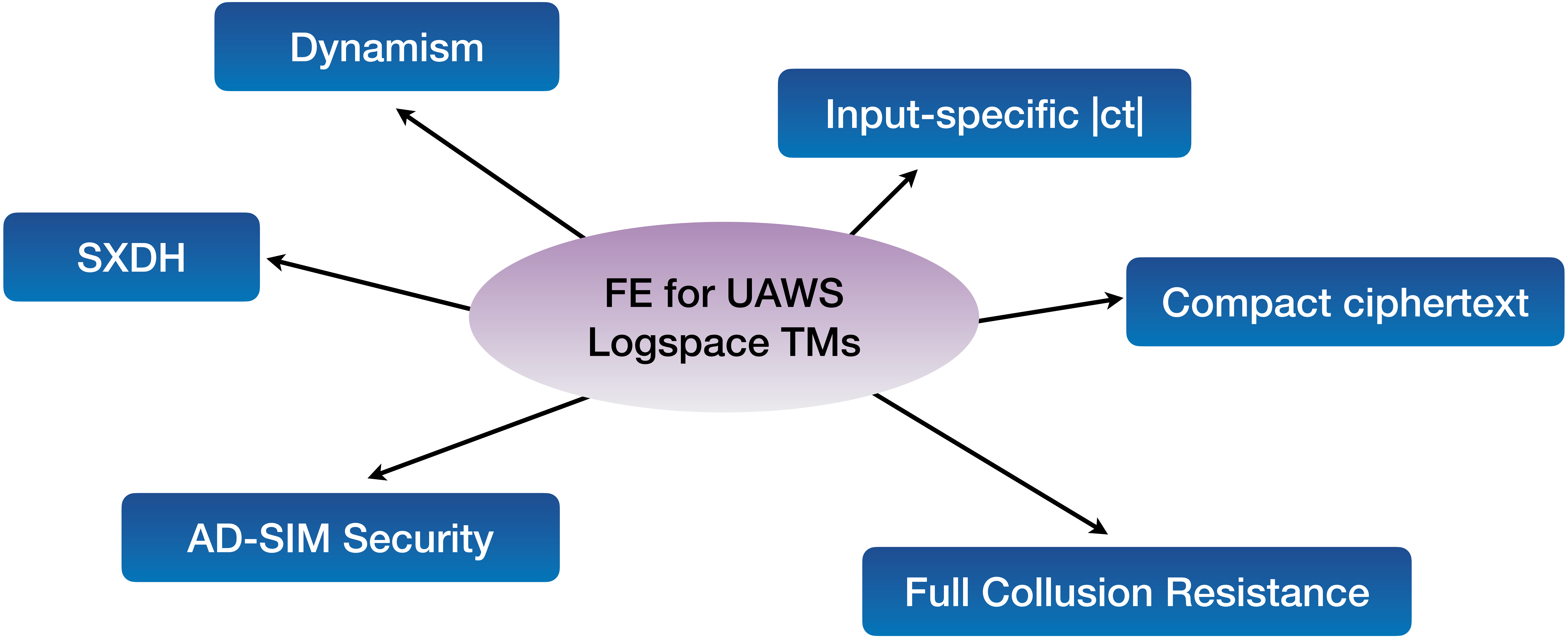
- UIPFE: $f(\mathbf{x}) = \mathbf{y}$
  $f(m) = \mathbf{y} \cdot \mathbf{z}$
- UABE: $f(\mathbf{x}) = 1/0,\ \mathbf{z} = M$
  $f(m) = M$ if f satisfies $\mathbf{x}$
- UAB-IPFE: $f(\mathbf{x}) = \mathbf{y}g(\mathbf{x})$
  $f(m) = \mathbf{y} \cdot \mathbf{z}$ if g satisfies $\mathbf{x}$

~~Limitations~~

- uniform, dynamic
- unbounded Setup
  $|\mathsf{mpk}| = O(\lambda)$
- $|\mathsf{ct}_m|$ = input-specific
- Unbounded FE:
  UIPFE, UABE

Summary of our Results

- Define FE for Unbounded Attribute-Weighted Sums (UAWS)



FE for UAWS
Logspace TMs

Dynamism

Input-specific |ct|

SXDH

Compact ciphertext

AD-SIM Security

Full Collusion Resistance

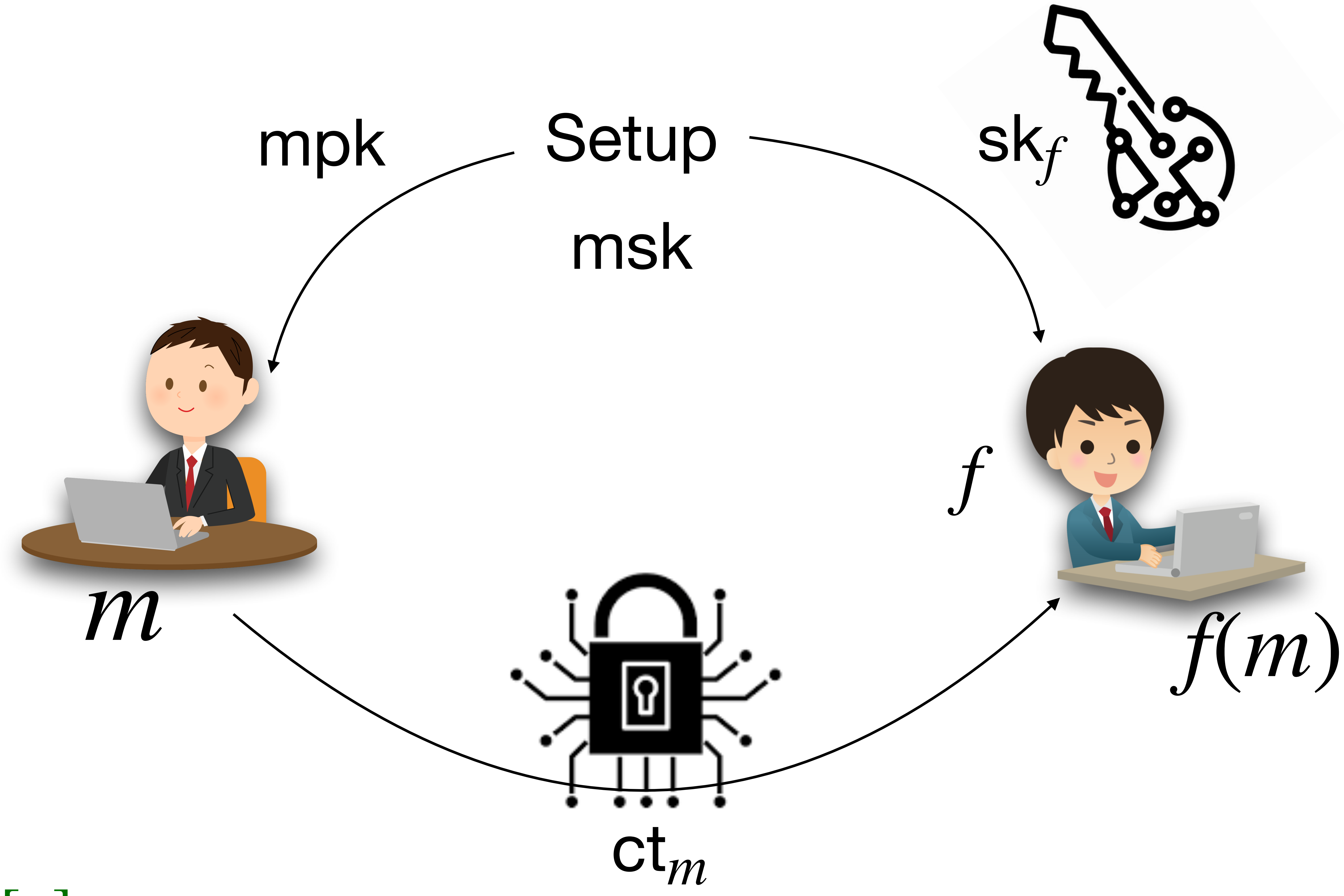
How to Define: FE for Unbounded AWS (UAWS)

Setup$(1^\lambda) \to$ (mpk, msk)

KeyGen(msk, f) $\to$ sk$_f$

Enc(mpk, m) $\to$ ct$_m$

Dec(sk$_f$, ct$_m$) $\to f(m)$



mpk

Setup

msk

sk$_f$

m

f

$f(m)$

ct$_m$

- Function: $f = (M_k)_{k \in I}$ s.t. $M_k \in$ TM
- Message: $m = (\mathbf{x}, \mathbf{z}) \in \{0,1\}^* \times \mathbb{Z}_p^n$
- Output: $f(m) = \sum_{k \in I} M_k(\mathbf{x}) z_k$ iff $I \subseteq [n]$

$\mathbf{x}$ is public, $\mathbf{z}$ is private

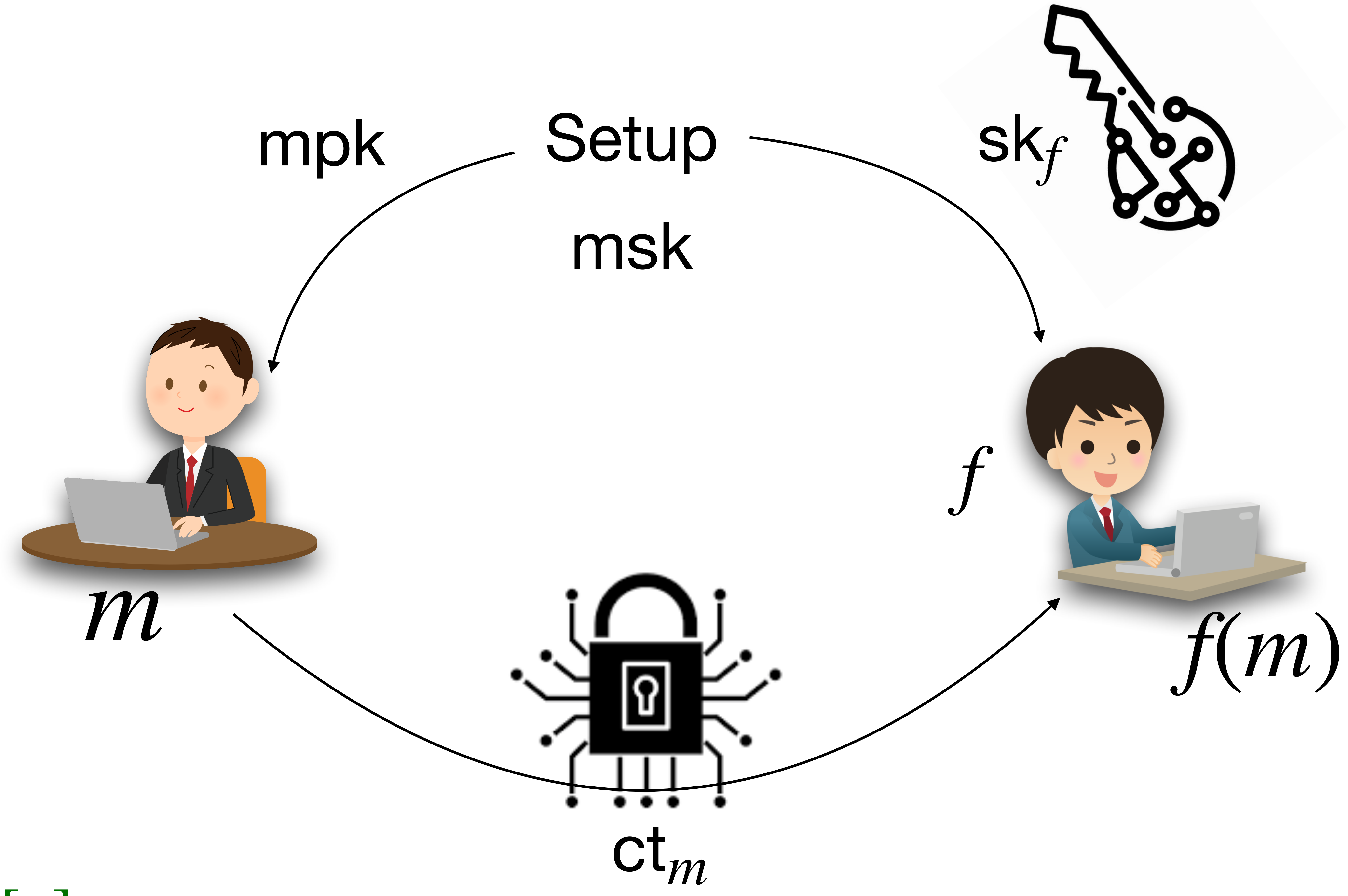
How to Define: FE for Unbounded AWS (UAWS)

Setup$(1^\lambda) \rightarrow$ (mpk, msk)

KeyGen(msk, f) $\rightarrow$ sk$_f$

Enc(mpk, m) $\rightarrow$ ct$_m$

Dec(sk$_f$, ct$_m$) $\rightarrow f(m)$



- Function: $f = (M_k)_{k \in I}$ s.t. $M_k \in$ TM
- Message: $m = (\mathbf{x}, \mathbf{z}) \in \{0,1\}^* \times \mathbb{Z}_p^n$
- Output: $f(m) = \sum_{k \in I} M_k(\mathbf{x}) z_k$ iff $I \subseteq [n]$

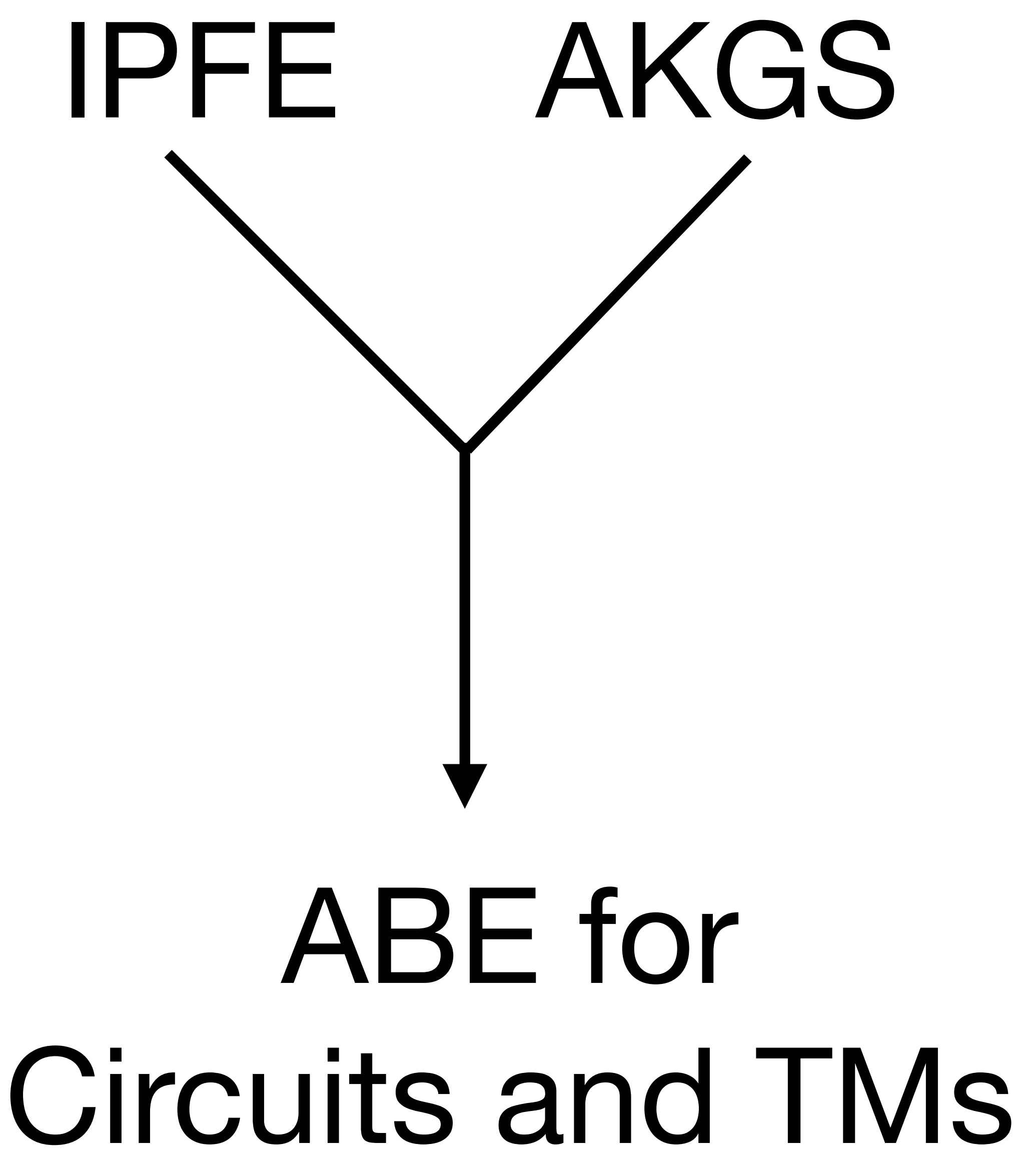
$\mathbf{x}$ is public, $\mathbf{z}$ is private

$$f = (M_1, M_2, M_4),\ m = (\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3, z_4, z_5))$$

$$f(m) = M_1(\mathbf{x})z_1 + M_2(\mathbf{x})z_2 + M_4(\mathbf{x})z_4$$

Roadmap towards FE for UAWS

[LL20]



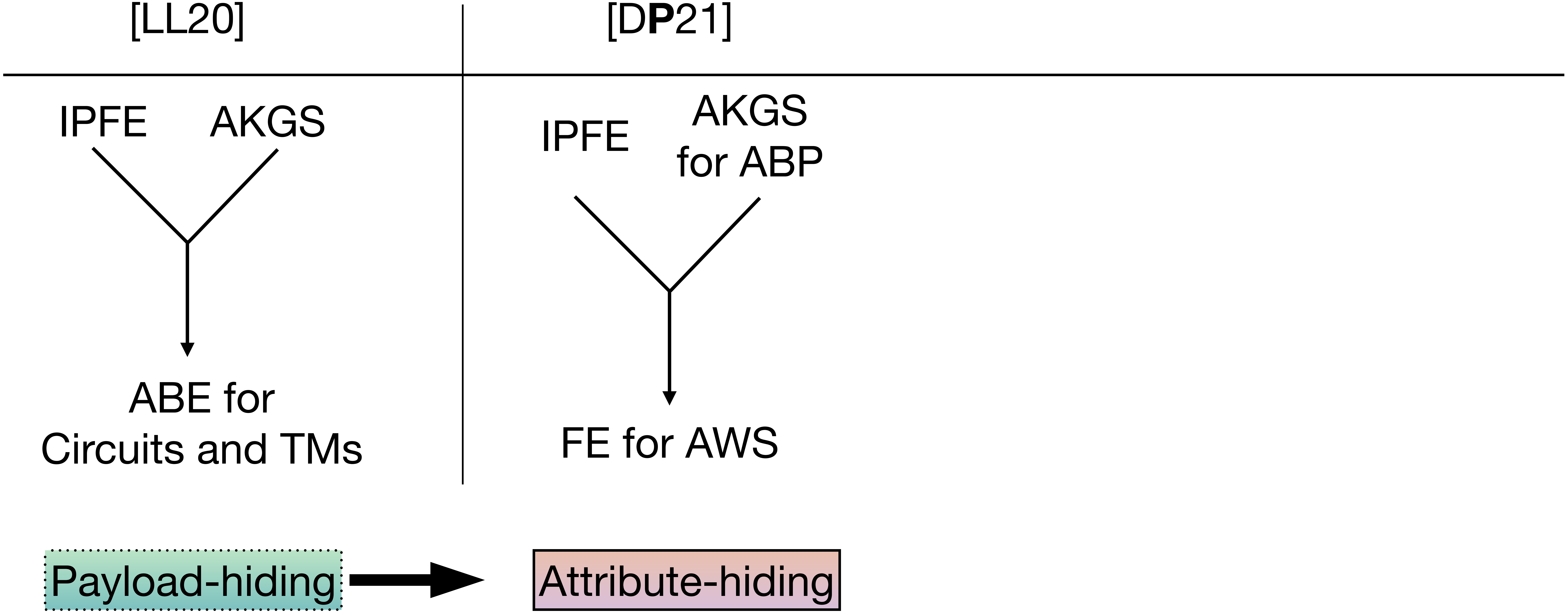
IPFE

AKGS

ABE for
Circuits and TMs

Payload-hiding

Roadmap towards FE for UAWS



[LL20]

IPFE

AKGS

ABE for Circuits and TMs

[D**P**21]

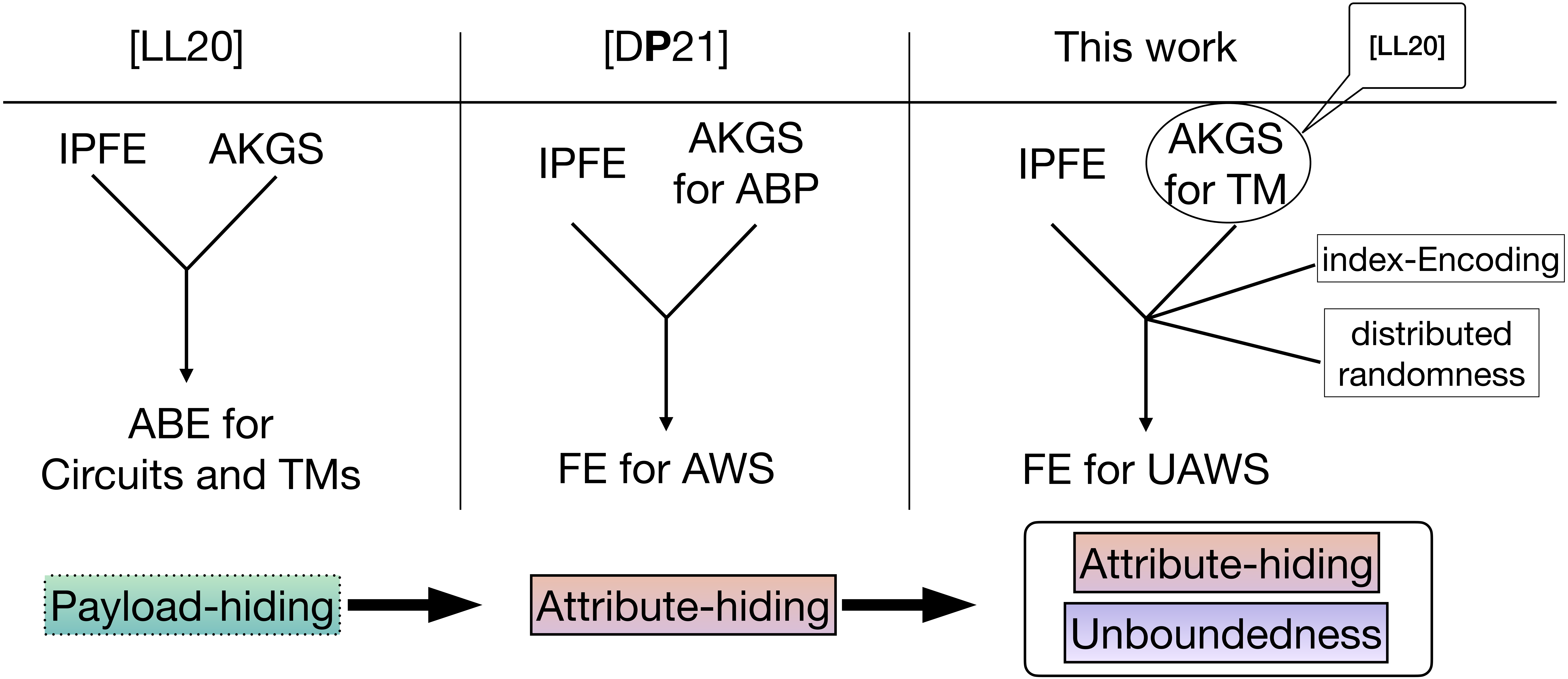
IPFE

AKGS for ABP

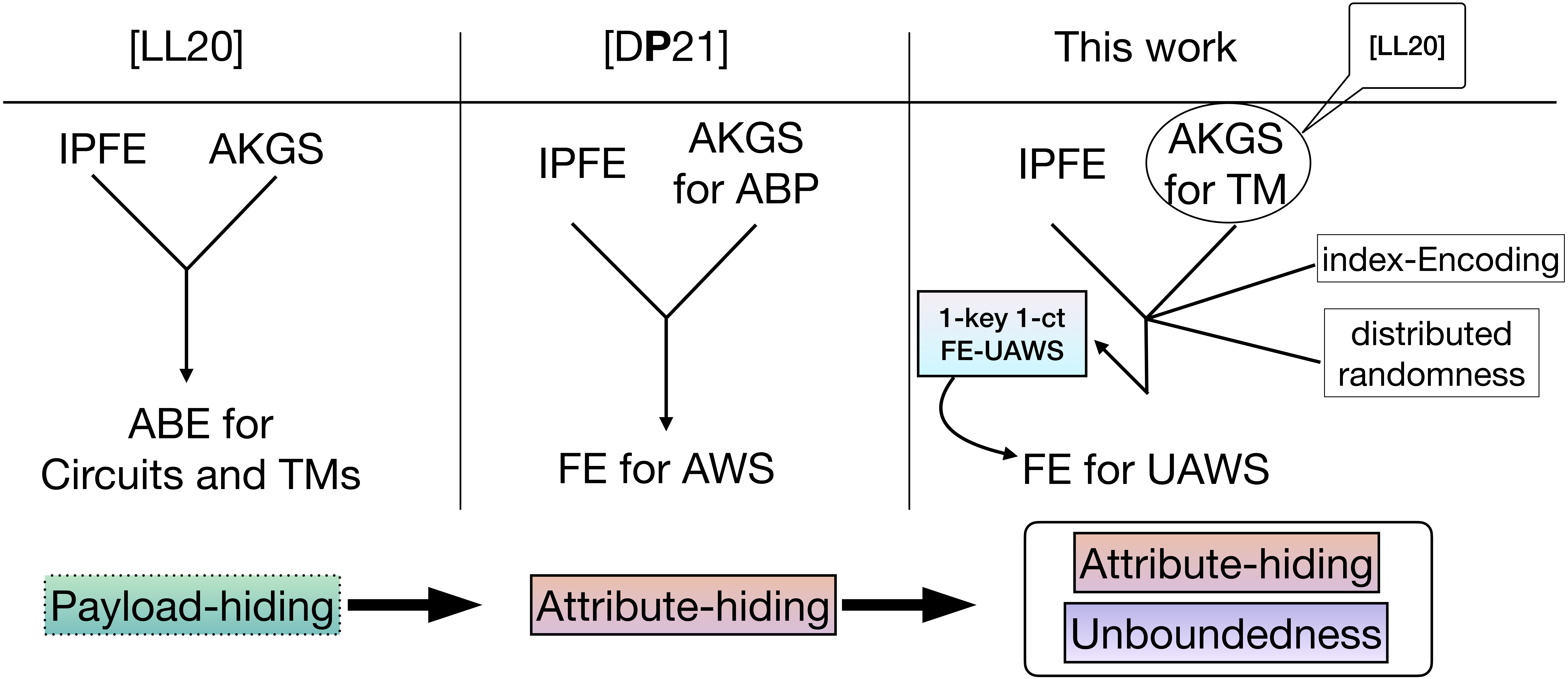
FE for AWS

Payload-hiding → Attribute-hiding

Roadmap towards FE for UAWS



Roadmap towards FE for UAWS



[LL20]

IPFE AKGS

ABE for Circuits and TMs

[D**P**21]

IPFE AKGS for ABP

FE for AWS

This work

[LL20]

IPFE AKGS for TM

index-Encoding

distributed randomness

1-key 1-ct FE-UAWS

FE for UAWS

Payload-hiding

Attribute-hiding

Attribute-hiding

Unboundedness

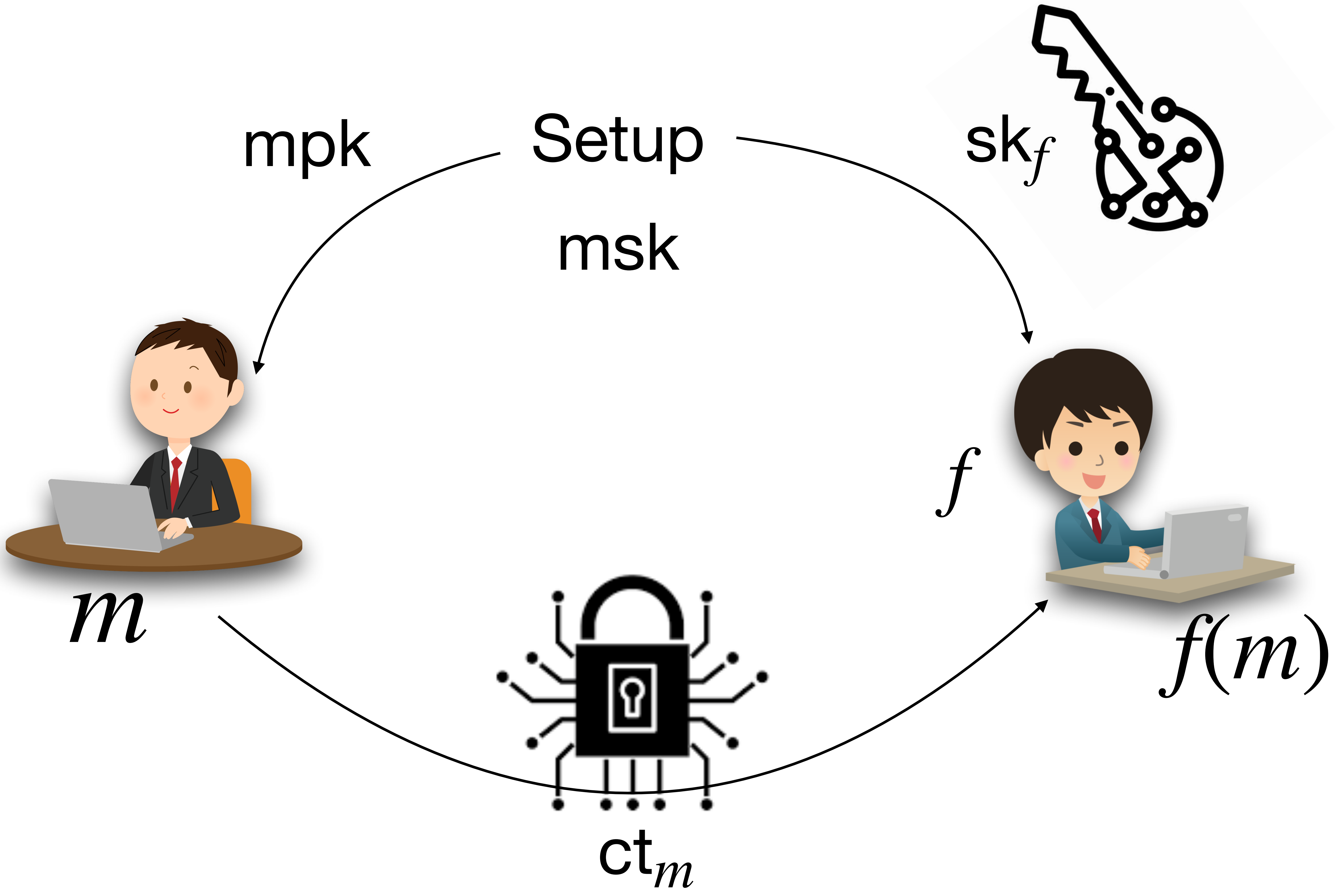
Inner Product Functional Encryption (IPFE)

$\mathsf{Setup}(1^\lambda, 1^n) \rightarrow \mathsf{msk}$

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

$\mathsf{Enc}(\mathsf{msk}, m) \rightarrow \mathsf{ct}_m$

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_m) \rightarrow f(m)$



- Functions: $f = \mathbf{y} \in \mathbb{Z}_p^n$
- Message: $m = \mathbf{x} \in \mathbb{Z}_p^n$
- Output: $f(m) = \mathbf{x} \cdot \mathbf{y}$

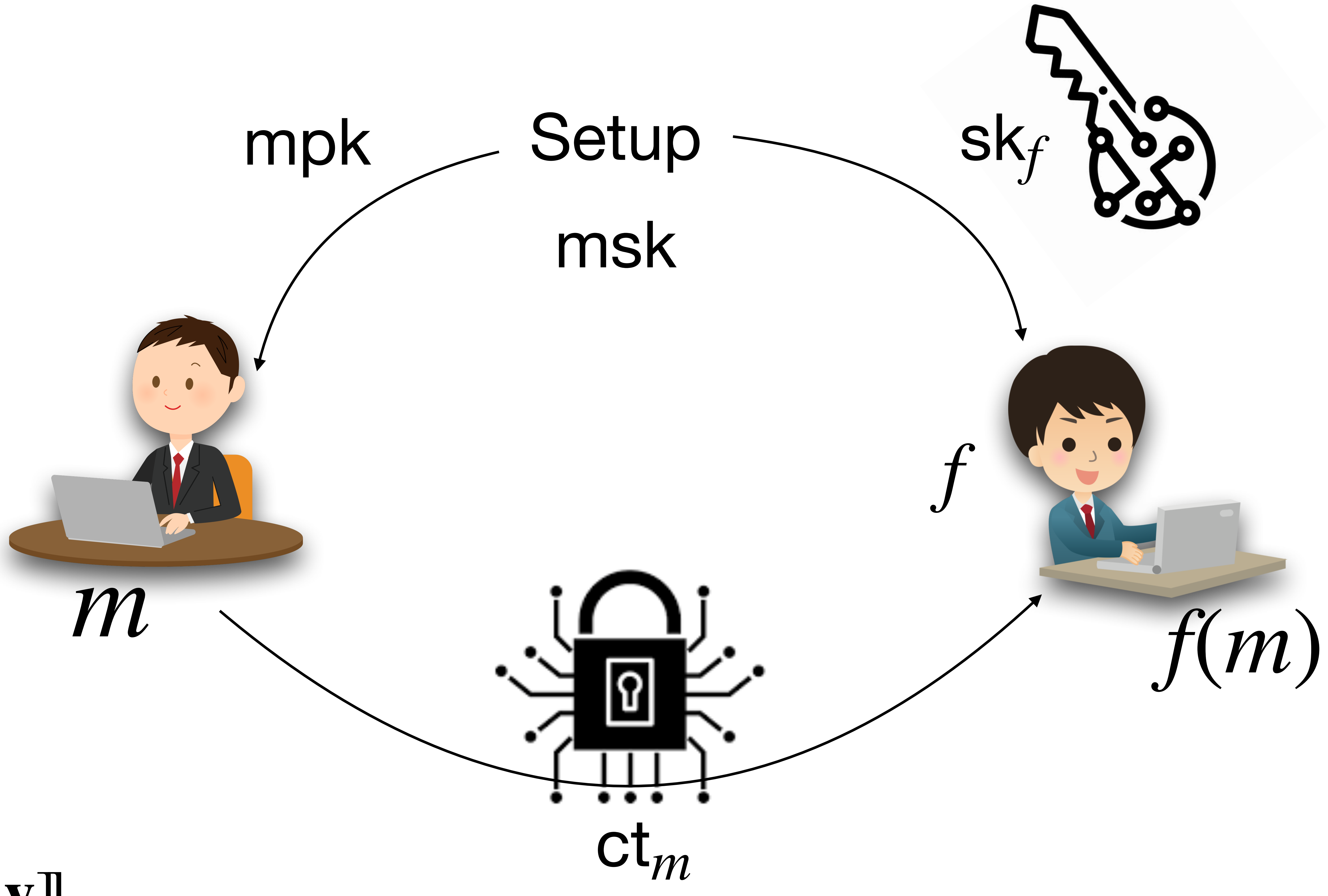
Inner Product Functional Encryption (IPFE)

$\mathsf{Setup}(1^\lambda, 1^n) \rightarrow \mathsf{msk}$

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

$\mathsf{Enc}(\mathsf{msk}, m) \rightarrow \mathsf{ct}_m$

$\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct}_m) \rightarrow f(m)$



$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$

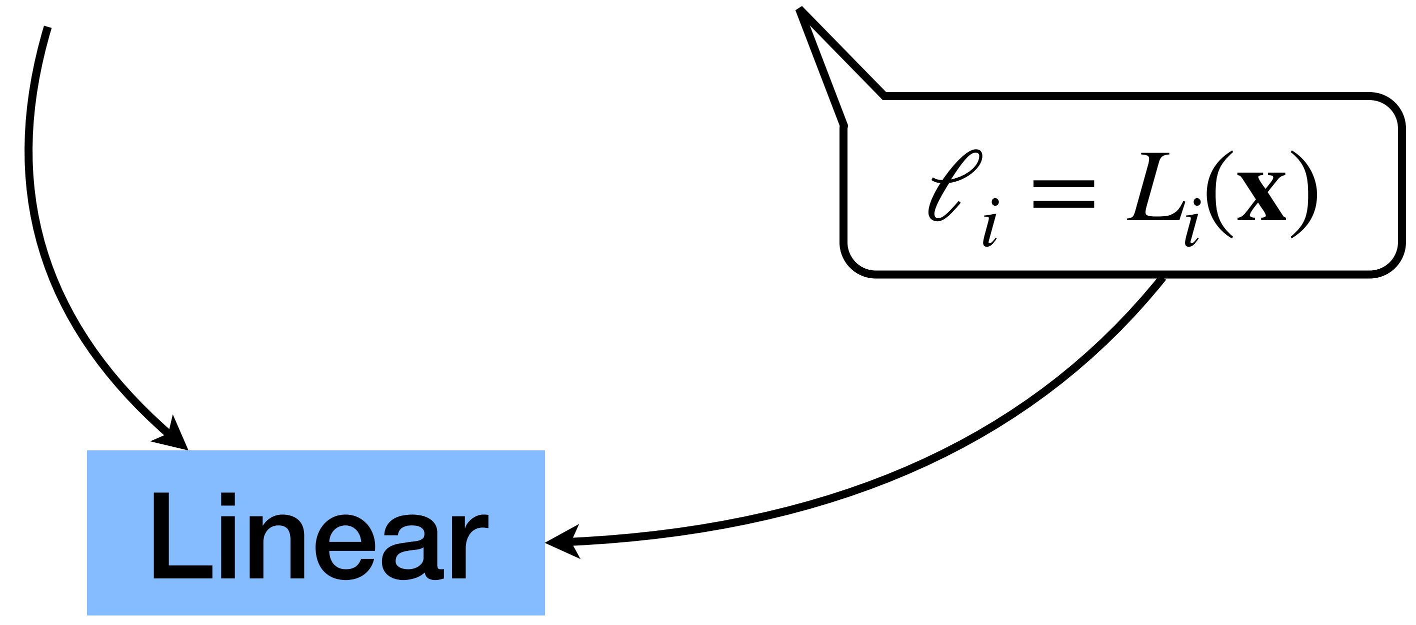
- Functions: $f = [\![\mathbf{y}]\!]_2 \in \mathbb{G}_2^n$
- Message: $m = [\![\mathbf{x}]\!]_1 \in \mathbb{G}_1^n$
- Output: $f(m) = e([\![\mathbf{x}]\!]_1, [\![\mathbf{y}]\!]_2) = [\![\mathbf{x} \cdot \mathbf{y}]\!]_T$

FH-IPFE: $\{\mathsf{sk}_{\mathbf{y}_0}, \mathsf{ct}_{\mathbf{x}_0}\} \approx_c \{\mathsf{sk}_{\mathbf{y}_1}, \mathsf{ct}_{\mathbf{x}_1}\}$ if $[\![\mathbf{x}_0 \cdot \mathbf{y}_0]\!]_T = [\![\mathbf{x}_1 \cdot \mathbf{y}_1]\!]_T$

Arithmetic Key Garbling Scheme over $\mathbb{Z}_p$ [IW14]

Garble$(f, z, \beta) \to (L_1, \ldots, L_t), \quad f : \mathbb{Z}_p^m \to \mathbb{Z}_p$

Eval$(f, \mathbf{x}, \ell_1, \ldots, \ell_t) \to zf(\mathbf{x}) + \beta$

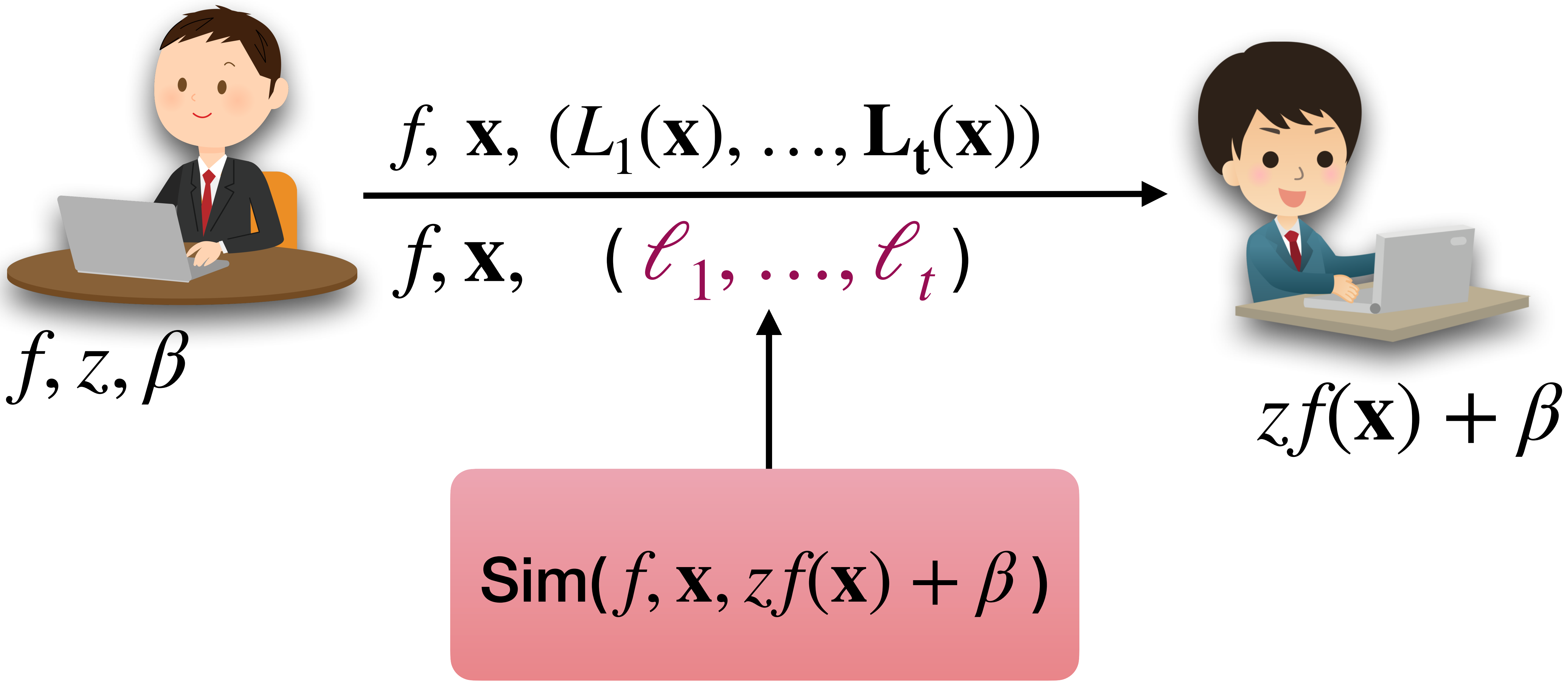


$\ell_i = L_i(\mathbf{x})$

Linear

Sim$(f, \mathbf{x}, zf(\mathbf{x}) + \beta) \to (\ell_1, \ldots, \ell_t)$

$f, \mathbf{x}$ are public, $\quad z, \beta$ are private



$f,\ \mathbf{x},\ (L_1(\mathbf{x}), \ldots, \mathbf{L_t}(\mathbf{x}))$

$f, \mathbf{x}, \quad (\ \ell_1, \ldots, \ell_t\ )$

f, z, β

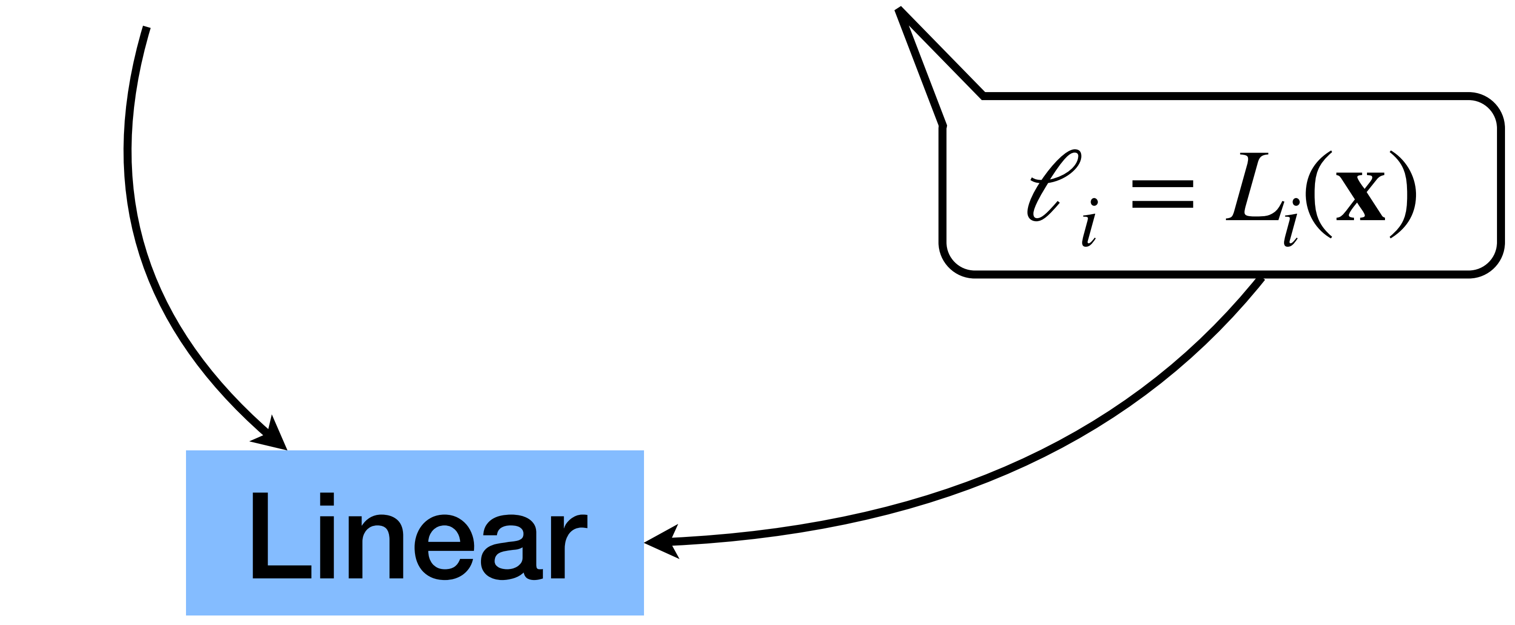
$zf(\mathbf{x}) + \beta$

Sim$(f, \mathbf{x}, zf(\mathbf{x}) + \beta\ )$

Arithmetic Key Garbling Scheme: Piecewise Security [LL20]

Garble$(f, z, \beta) \to (L_1, \ldots, L_t)$, $\quad f : \mathbb{Z}_p^m \to \mathbb{Z}_p$

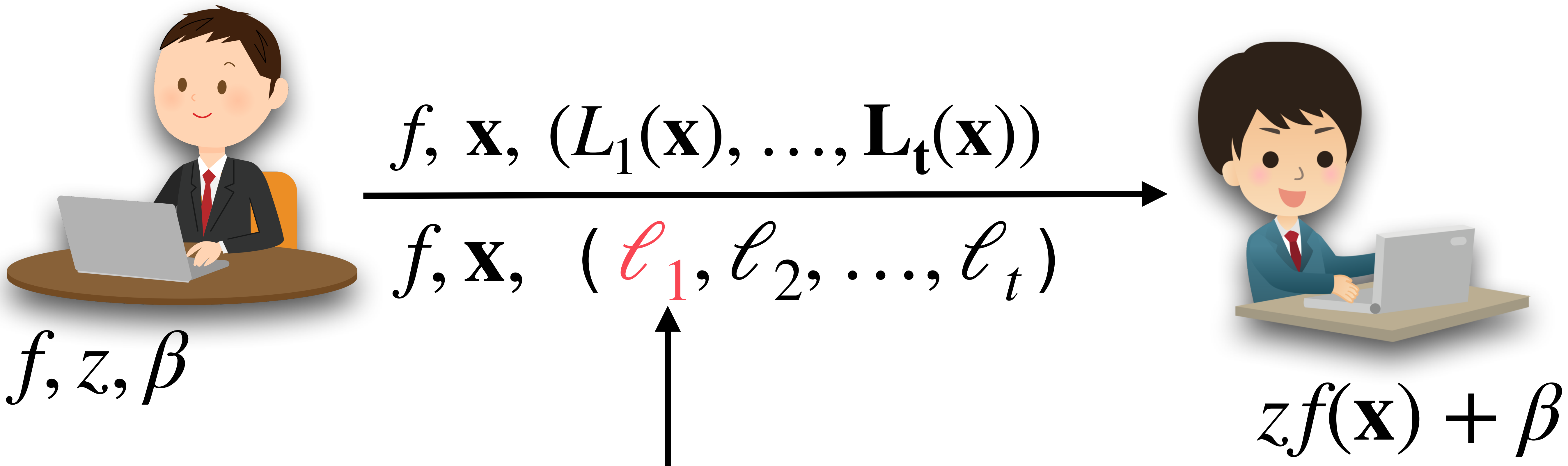
Eval$(f, \mathbf{x}, \ell_1, \ldots, \ell_t) \to zf(\mathbf{x}) + \beta$



$\ell_i = L_i(\mathbf{x})$

Linear

$f, \mathbf{x}$ are public, $\quad z, \beta$ are private



f, z, β

$f,\ \mathbf{x},\ (L_1(\mathbf{x}), \ldots, \mathbf{L}_{\mathbf{t}}(\mathbf{x}))$

$f, \mathbf{x},\ \ (\ell_1, \ell_2, \ldots, \ell_t)$

$zf(\mathbf{x}) + \beta$

RevSamp$(f, \mathbf{x}, zf(\mathbf{x}) + \beta, \ell_2, \ldots, \ell_t)$

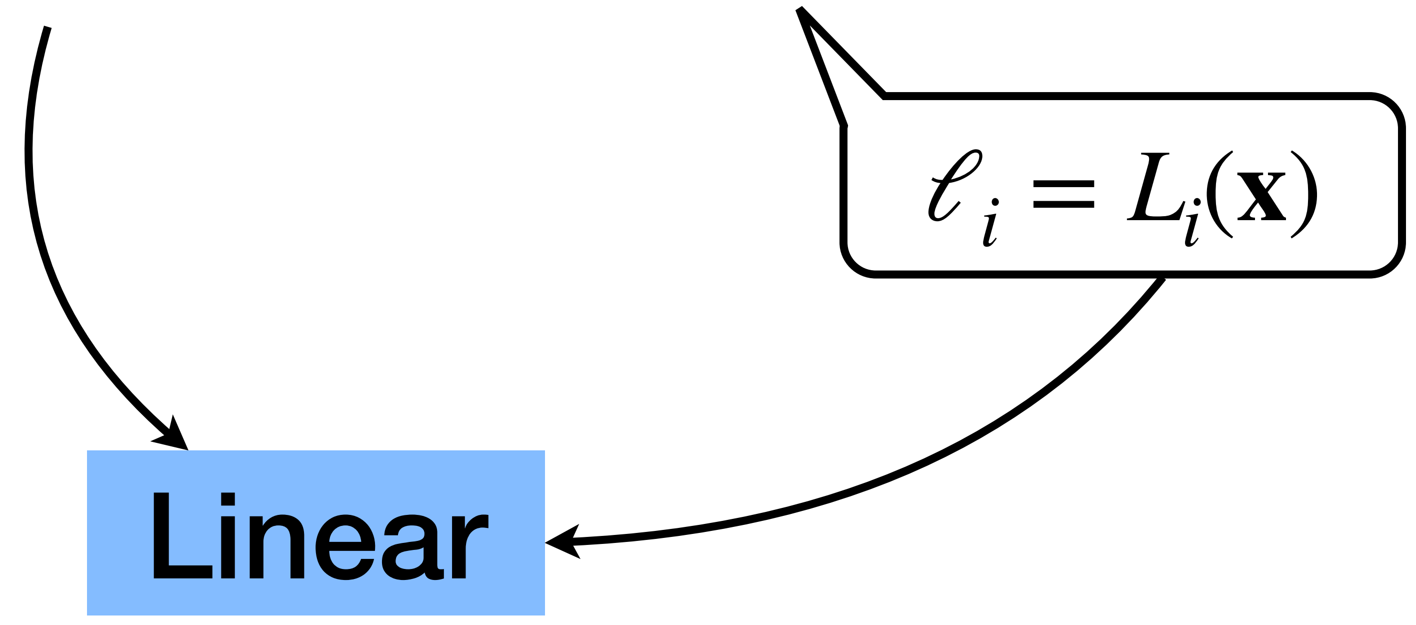
Sim$(f, \mathbf{x}, zf(\mathbf{x}) + \beta) \to (\ell_1, \ldots, \ell_t)$

RevSamp$(f, \mathbf{x}, zf(\mathbf{x}) + \beta, \ell_2, \ldots, \ell_t) \to \ell_1$

Arithmetic Key Garbling Scheme: Piecewise Security [LL20]

Garble$(f, z, \beta) \to (L_1, \ldots, L_t)$, $f : \mathbb{Z}_p^m \to \mathbb{Z}_p$

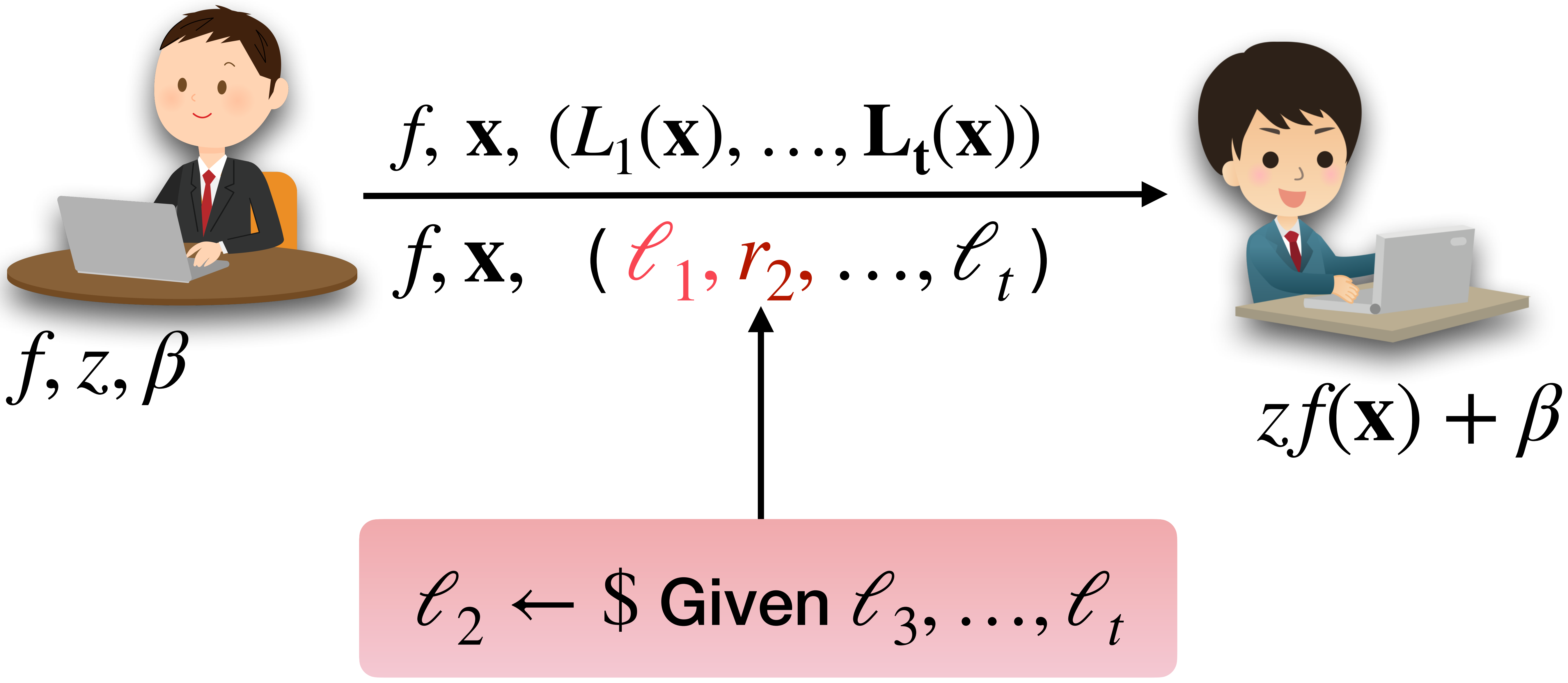
Eval$(f, \mathbf{x}, \ell_1, \ldots, \ell_t) \to zf(\mathbf{x}) + \beta$



$\ell_i = L_i(\mathbf{x})$

Linear

$f, \mathbf{x}$ are public, z, β are private



f, z, β

$f,\ \mathbf{x},\ (L_1(\mathbf{x}), \ldots, \mathbf{L_t}(\mathbf{x}))$

$f, \mathbf{x},\ (\ell_1, r_2, \ldots, \ell_t)$

$zf(\mathbf{x}) + \beta$

$\ell_2 \leftarrow \$$ Given $\ell_3, \ldots, \ell_t$

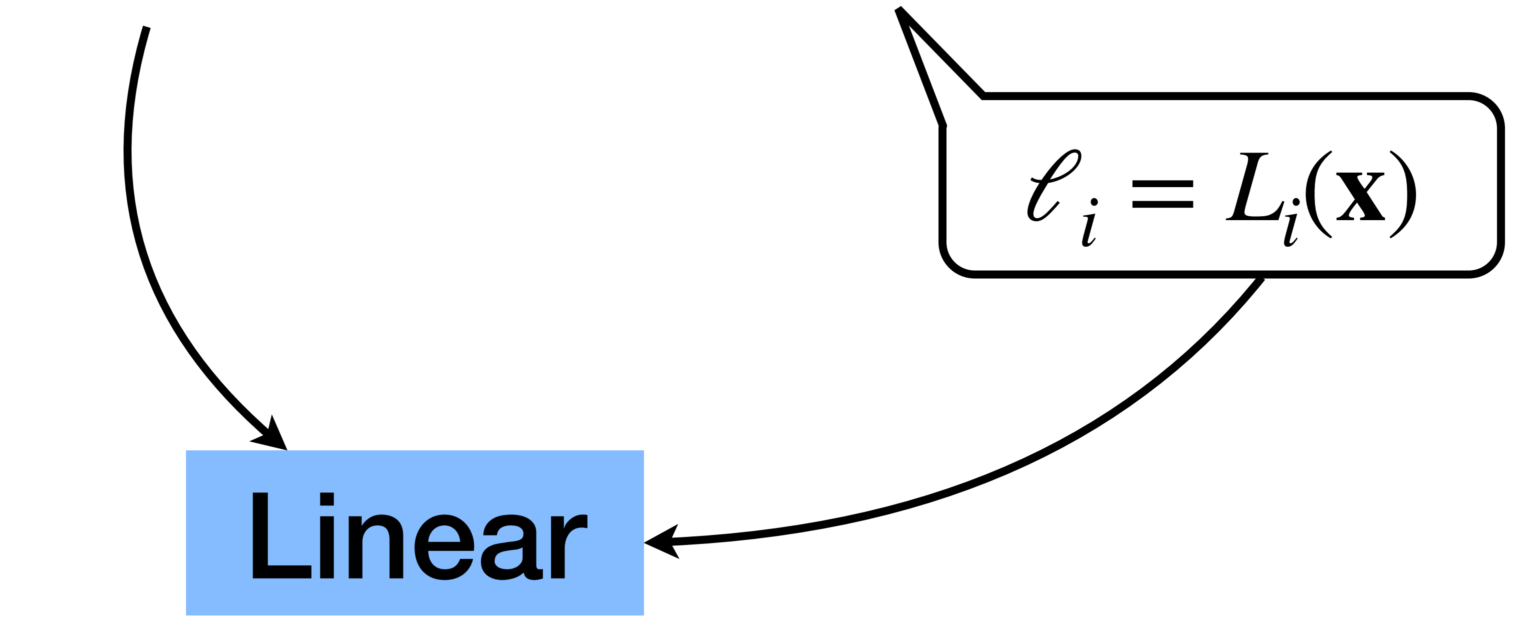
Sim$(f, \mathbf{x}, zf(\mathbf{x}) + \beta) \to (\ell_1, \ldots, \ell_t)$

RevSamp$(f, \mathbf{x}, zf(\mathbf{x}) + \beta, \ell_2, \ldots, \ell_t) \to \ell_1$

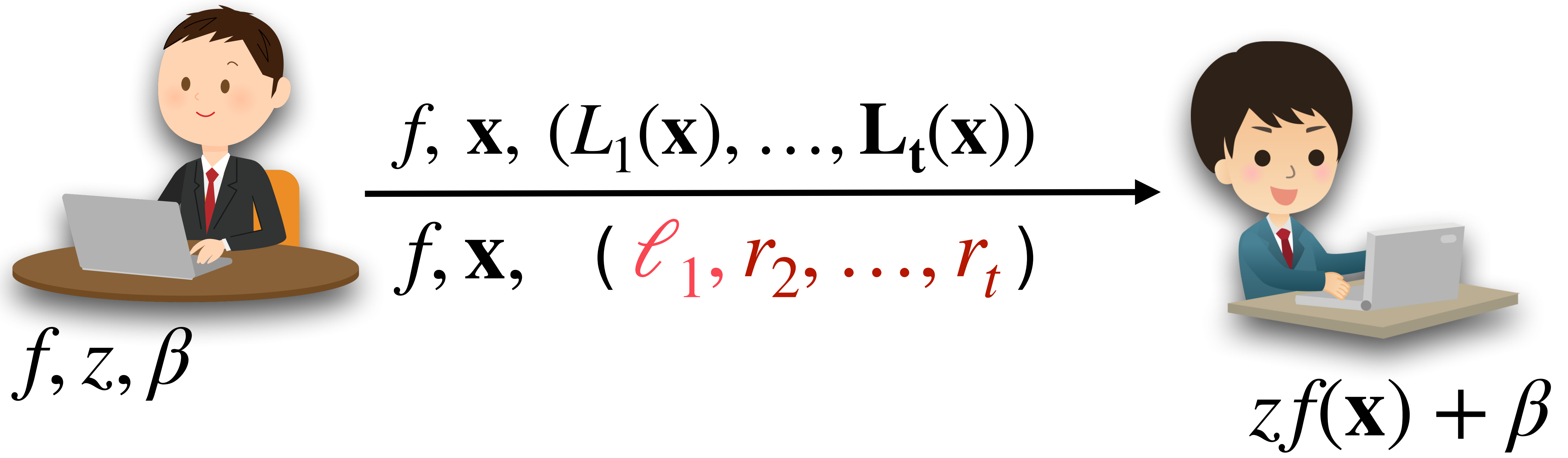
Arithmetic Key Garbling Scheme: Piecewise Security [LL20]

$\text{Garble}(f, z, \beta) \rightarrow (L_1, \ldots, L_t), \quad f : \mathbb{Z}_p^m \rightarrow \mathbb{Z}_p$

$\text{Eval}(f, \mathbf{x}, \ell_1, \ldots, \ell_t) \rightarrow zf(\mathbf{x}) + \beta$



$f, \mathbf{x}$ are public, $\quad z, \beta$ are private



$f, \mathbf{x}, (L_1(\mathbf{x}), \ldots, \mathbf{L}_\mathbf{t}(\mathbf{x}))$
$f, \mathbf{x}, \quad (\ell_1, r_2, \ldots, r_t)$
f, z, β
$zf(\mathbf{x}) + \beta$

$\text{Sim}(f, \mathbf{x}, zf(\mathbf{x}) + \beta) \rightarrow (\ell_1, \ldots, \ell_t)$

$\Uparrow$

Piecewise Security

$\text{RevSamp}(f, \mathbf{x}, zf(\mathbf{x}) + \beta, \ell_2, \ldots, \ell_t) \rightarrow \ell_1$

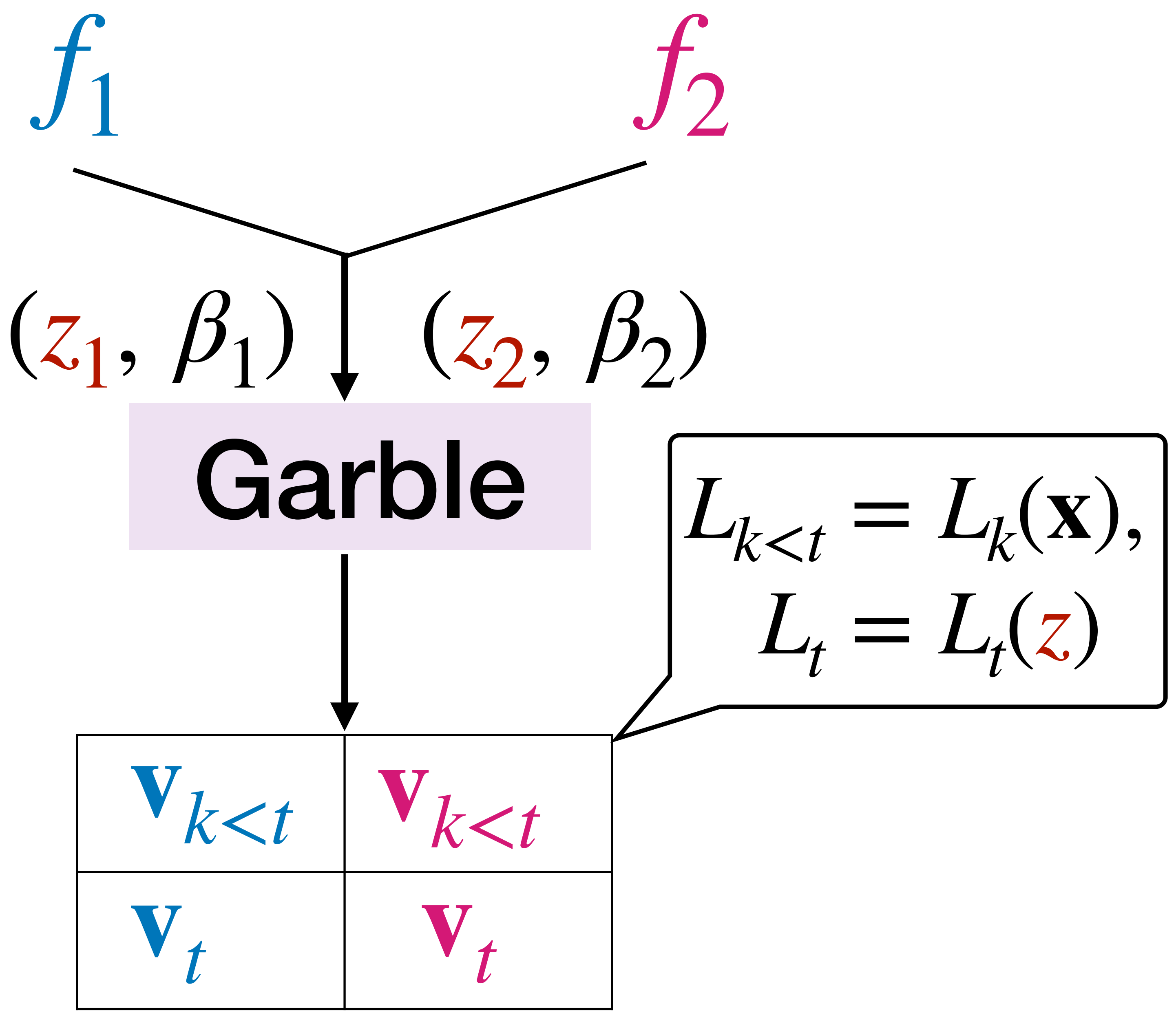
Marginal Randomness: $\ell_{j>1} \leftarrow \$$, Given $\ell_{j+1}, \ldots, \ell_t$

Core Idea of [D**P**21] for FE-AWS for ABPs

function	input	output
$f = (f_1, f_2)$ s.t. $f_k : \mathbb{Z}_p^n \to \mathbb{Z}_p$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2))$	$z_1 f_1(\mathbf{x}) + z_2 f_2(\mathbf{x})$

Core Idea of [D**P**21] for FE-AWS for ABPs

function	input	output
$f = (f_1, f_2)$ s.t. $f_k : \mathbb{Z}_p^n \to \mathbb{Z}_p$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2))$	$z_1 f_1(\mathbf{x}) + z_2 f_2(\mathbf{x})$

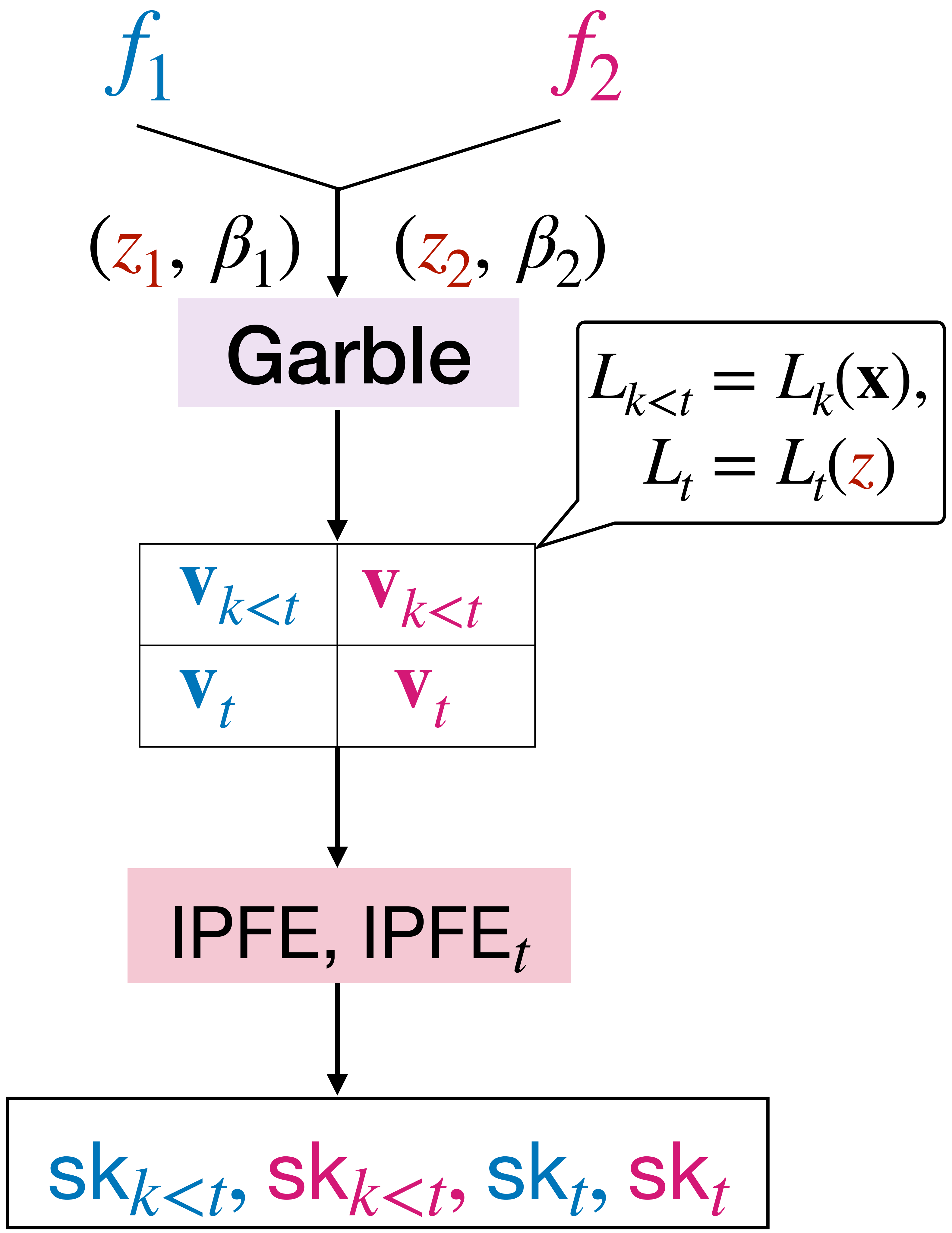


$L_{k<t} = L_k(\mathbf{x})$,
$L_t = L_t(z)$

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

Core Idea of [D**P**21] for FE-AWS for ABPs

function	input	output
$f = (\ f_1, f_2\ )$ s.t. $f_k : \mathbb{Z}_p^n \to \mathbb{Z}_p$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2))$	$z_1 f_1(\mathbf{x}) + z_2 f_2(\mathbf{x})$



f_1 f_2

$(z_1,\ \beta_1)$ $(z_2,\ \beta_2)$

Garble

$L_{k<t} = L_k(\mathbf{x}),$
$L_t = L_t(z)$

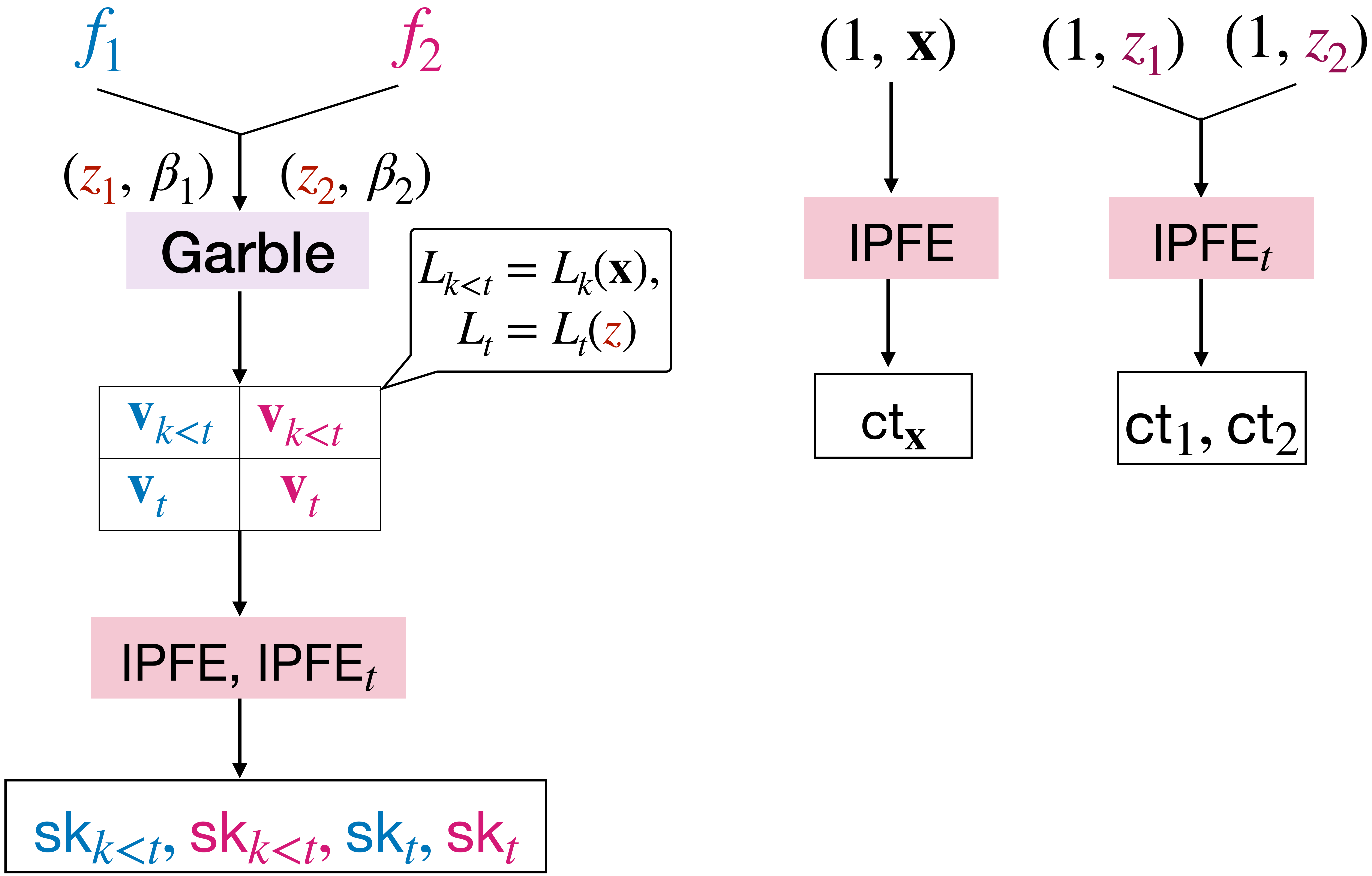
$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

IPFE, IPFE$_t$

$\mathsf{sk}_{k<t}, \mathsf{sk}_{k<t}, \mathsf{sk}_t, \mathsf{sk}_t$

Core Idea of [D**P**21] for FE-AWS for ABPs

function	input	output
$f = (f_1, f_2)$ s.t. $f_k : \mathbb{Z}_p^n \to \mathbb{Z}_p$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2))$	$z_1 f_1(\mathbf{x}) + z_2 f_2(\mathbf{x})$



f_1 f_2

(z_1, β_1) (z_2, β_2)

Garble

$L_{k<t} = L_k(\mathbf{x})$,
$L_t = L_t(z)$

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

IPFE, IPFE$_t$

sk$_{k<t}$, sk$_{k<t}$, sk$_t$, sk$_t$

$(1, \mathbf{x})$

IPFE

ct$_{\mathbf{x}}$

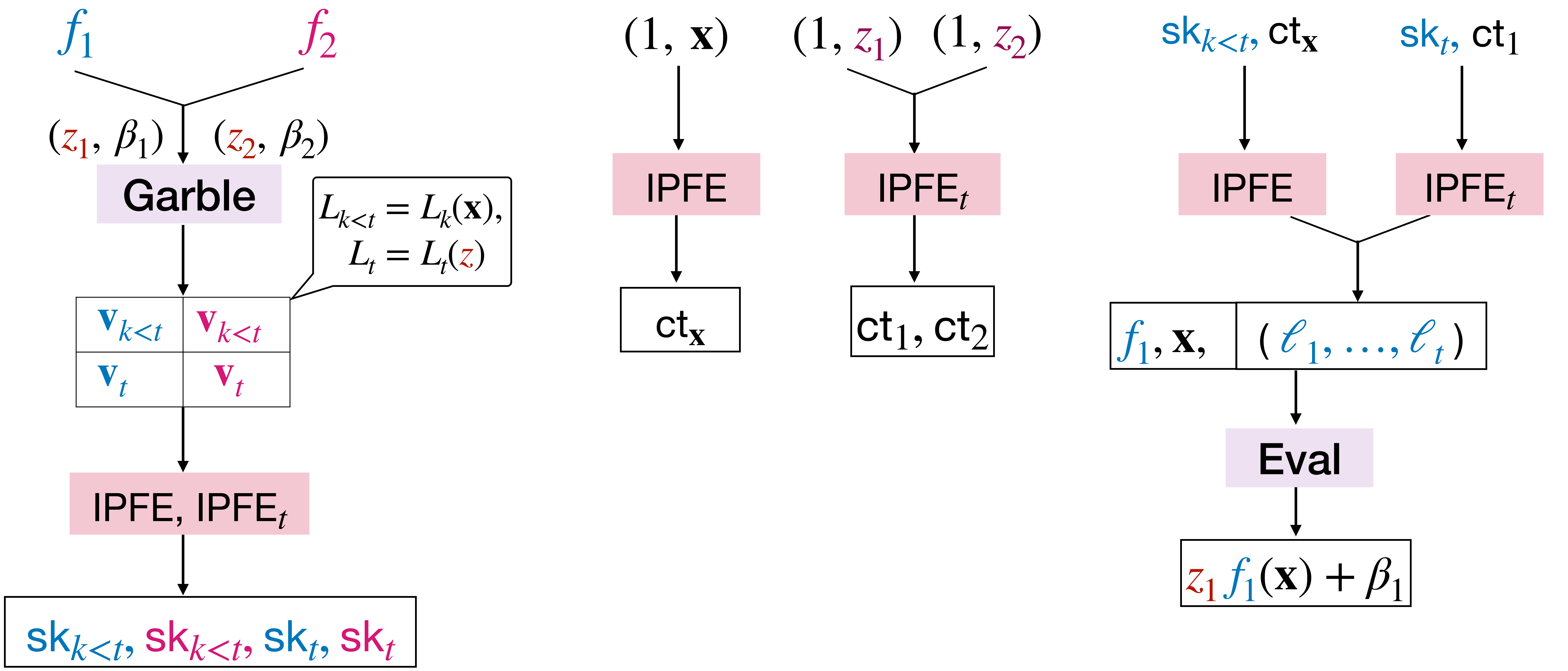
$(1, z_1)$ $(1, z_2)$

IPFE$_t$

ct$_1$, ct$_2$

Core Idea of [D**P**21] for FE-AWS for ABPs

function	input	output
$f = (f_1, f_2)$ s.t. $f_k : \mathbb{Z}_p^n \to \mathbb{Z}_p$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2))$	$z_1 f_1(\mathbf{x}) + z_2 f_2(\mathbf{x})$



f_1 f_2

(z_1, β_1) (z_2, β_2)

Garble

$L_{k<t} = L_k(\mathbf{x})$,
$L_t = L_t(z)$

$\mathbf{v}_{k<t}$ $\mathbf{v}_{k<t}$
$\mathbf{v}_t$ $\mathbf{v}_t$

IPFE, IPFE$_t$

sk$_{k<t}$, sk$_{k<t}$, sk$_t$, sk$_t$

$(1, \mathbf{x})$

IPFE

ct$_\mathbf{x}$

$(1, z_1)$ $(1, z_2)$

IPFE$_t$

ct$_1$, ct$_2$

sk$_{k<t}$, ct$_\mathbf{x}$ sk$_t$, ct$_1$

IPFE IPFE$_t$

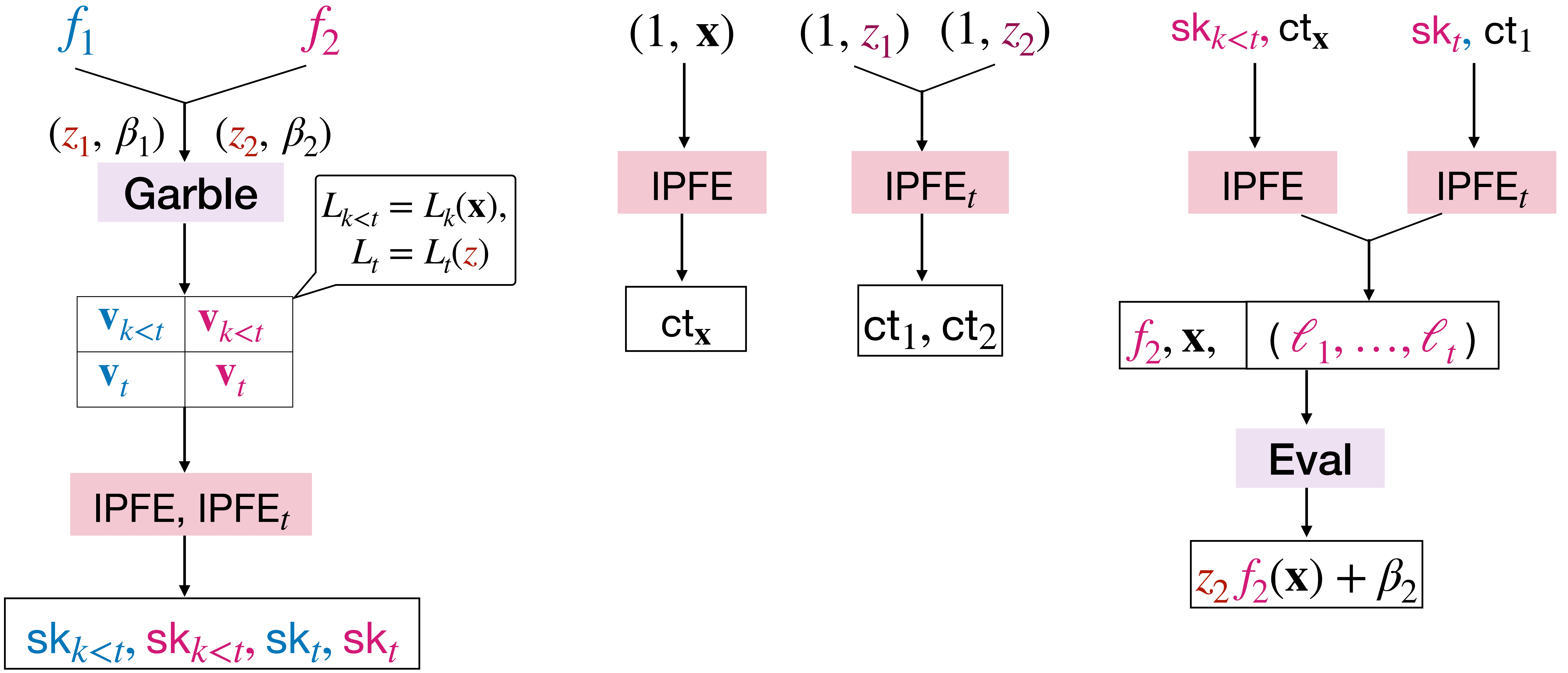
$f_1, \mathbf{x},$ $(\ell_1, \ldots, \ell_t)$

Eval

$z_1 f_1(\mathbf{x}) + \beta_1$

Core Idea of [D**P**21] for FE-AWS for ABPs

function	input	output
$f = (f_1, f_2)$ s.t. $f_k : \mathbb{Z}_p^n \to \mathbb{Z}_p$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2))$	$z_1 f_1(\mathbf{x}) + z_2 f_2(\mathbf{x})$



f_1 f_2

(z_1, β_1) (z_2, β_2)

Garble

$L_{k<t} = L_k(\mathbf{x})$,
$L_t = L_t(z)$

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

IPFE, IPFE$_t$

sk$_{k<t}$, sk$_{k<t}$, sk$_t$, sk$_t$

$(1, \mathbf{x})$

IPFE

ct$_\mathbf{x}$

$(1, z_1)$ $(1, z_2)$

IPFE$_t$

ct$_1$, ct$_2$

sk$_{k<t}$, ct$_\mathbf{x}$ sk$_t$, ct$_1$

IPFE IPFE$_t$

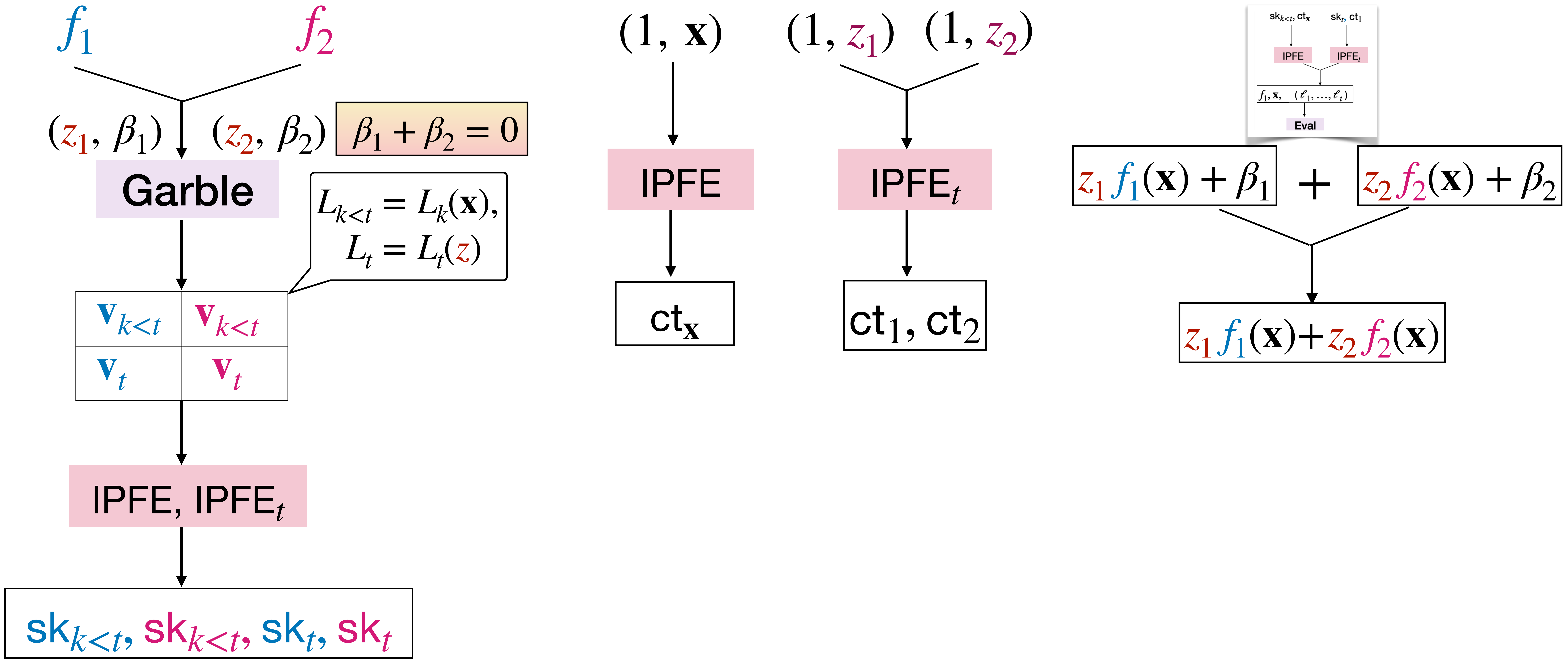
$f_2, \mathbf{x},$ $(\ell_1, \ldots, \ell_t)$

Eval

$z_2 f_2(\mathbf{x}) + \beta_2$

Core Idea of [D**P**21] for FE-AWS for ABPs

function	input	output
$f = (f_1, f_2)$ s.t. $f_k : \mathbb{Z}_p^n \to \mathbb{Z}_p$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2))$	$z_1 f_1(\mathbf{x}) + z_2 f_2(\mathbf{x})$



f_1 f_2

(z_1, β_1) (z_2, β_2) $\beta_1 + \beta_2 = 0$

Garble

$L_{k<t} = L_k(\mathbf{x})$,
$L_t = L_t(z)$

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

IPFE, IPFE$_t$

$\mathsf{sk}_{k<t}, \mathsf{sk}_{k<t}, \mathsf{sk}_t, \mathsf{sk}_t$

$(1, \mathbf{x})$ → IPFE → $\mathsf{ct}_\mathbf{x}$

$(1, z_1)$ $(1, z_2)$ → IPFE$_t$ → $\mathsf{ct}_1, \mathsf{ct}_2$

$\mathsf{sk}_{k<t}, \mathsf{ct}_\mathbf{x}$
$\mathsf{sk}_t, \mathsf{ct}_1$
IPFE
IPFE$_t$
$f_1, \mathbf{x}$, $(\ell_1, \ldots, \ell_t)$
Eval

$z_1 f_1(\mathbf{x}) + \beta_1$ + $z_2 f_2(\mathbf{x}) + \beta_2$

$z_1 f_1(\mathbf{x}) + z_2 f_2(\mathbf{x})$

Our Idea for FE-UAWS for TMs

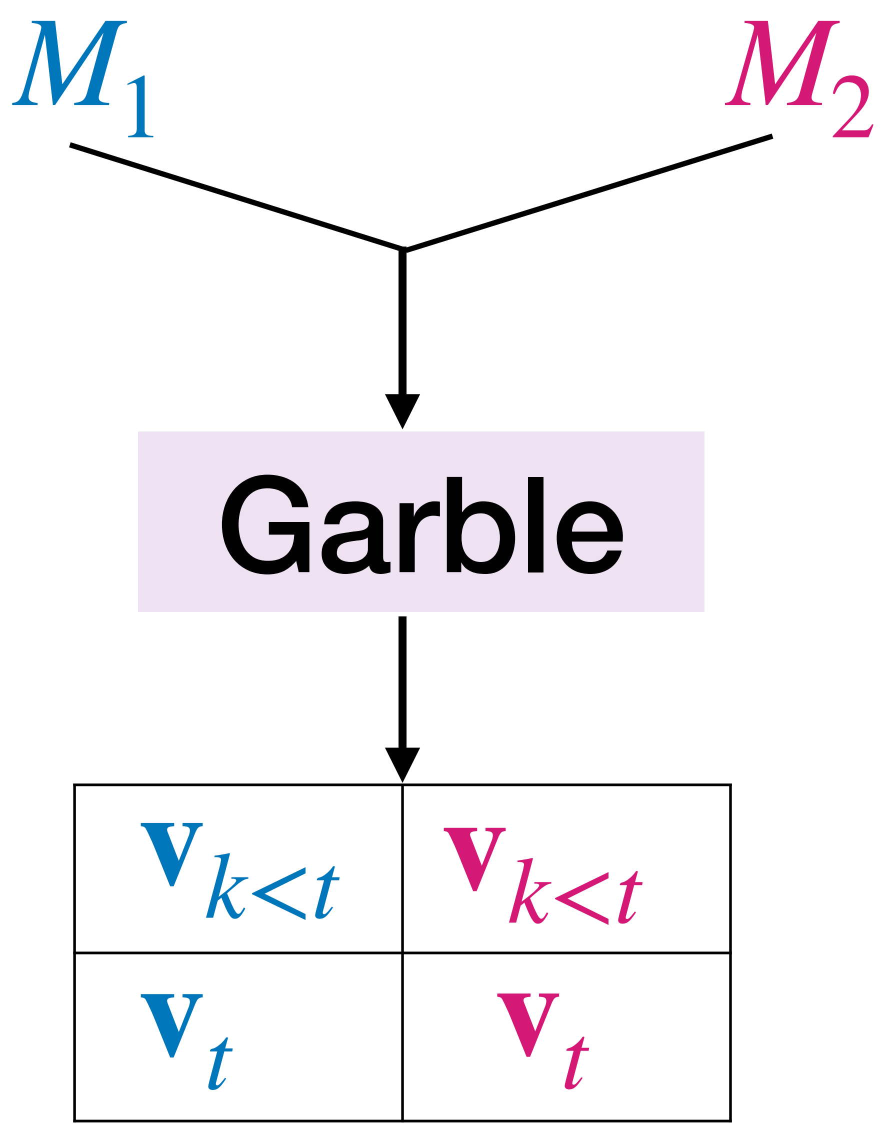
function	input	output
$M = (M_k)_{k\in I}$ s.t. $M_k : \{0,1\}^* \to \{0,1\}$	$(\mathbf{x}, \mathbf{z} = (z_1, \ldots, z_n))$	$\sum_{k\in I} M_k(\mathbf{x}) z_k$

Our Idea for FE-UAWS for TMs

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$

Our Idea for FE-UAWS for TMs

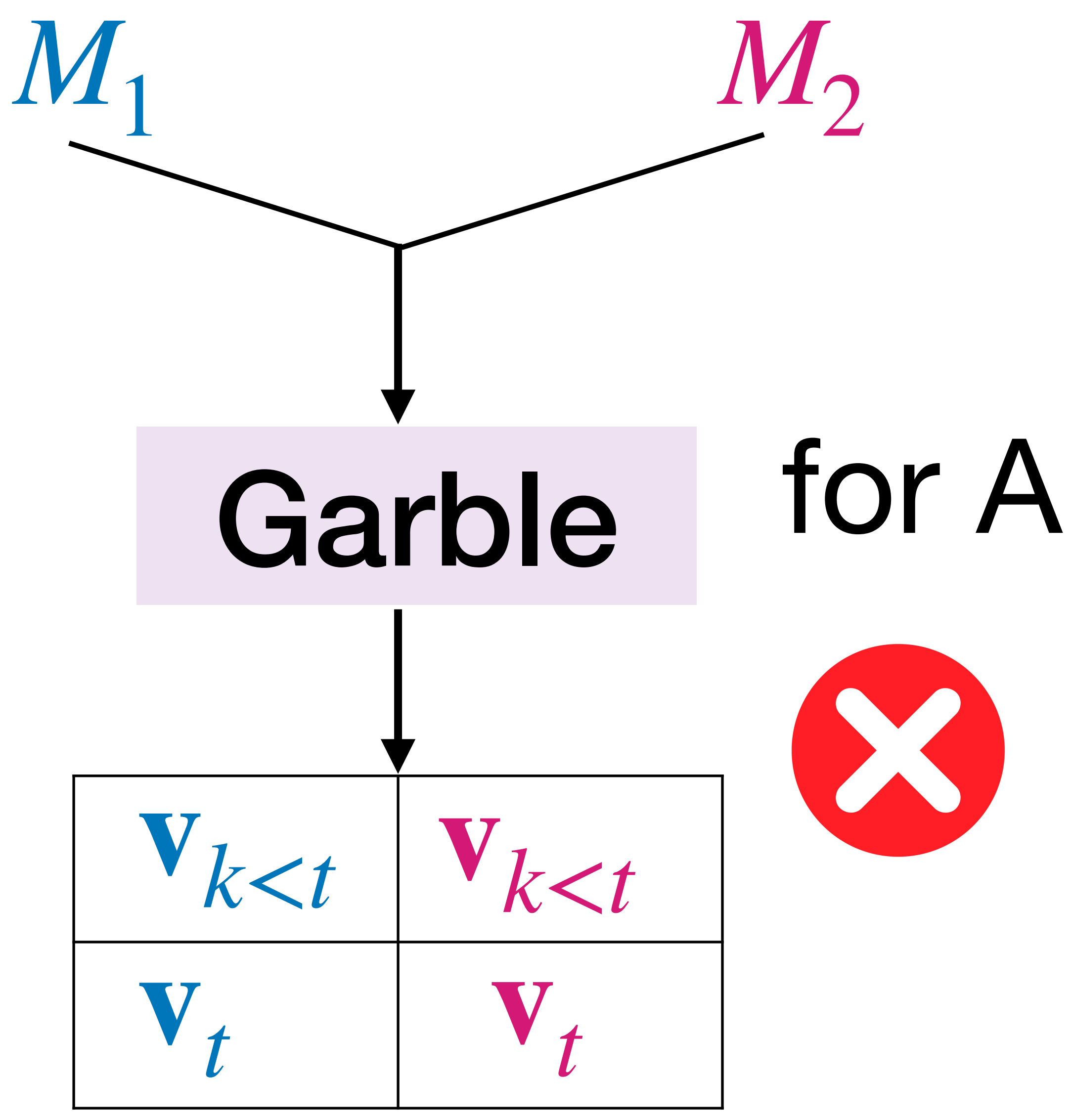
function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

Our Idea for FE-UAWS for TMs

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

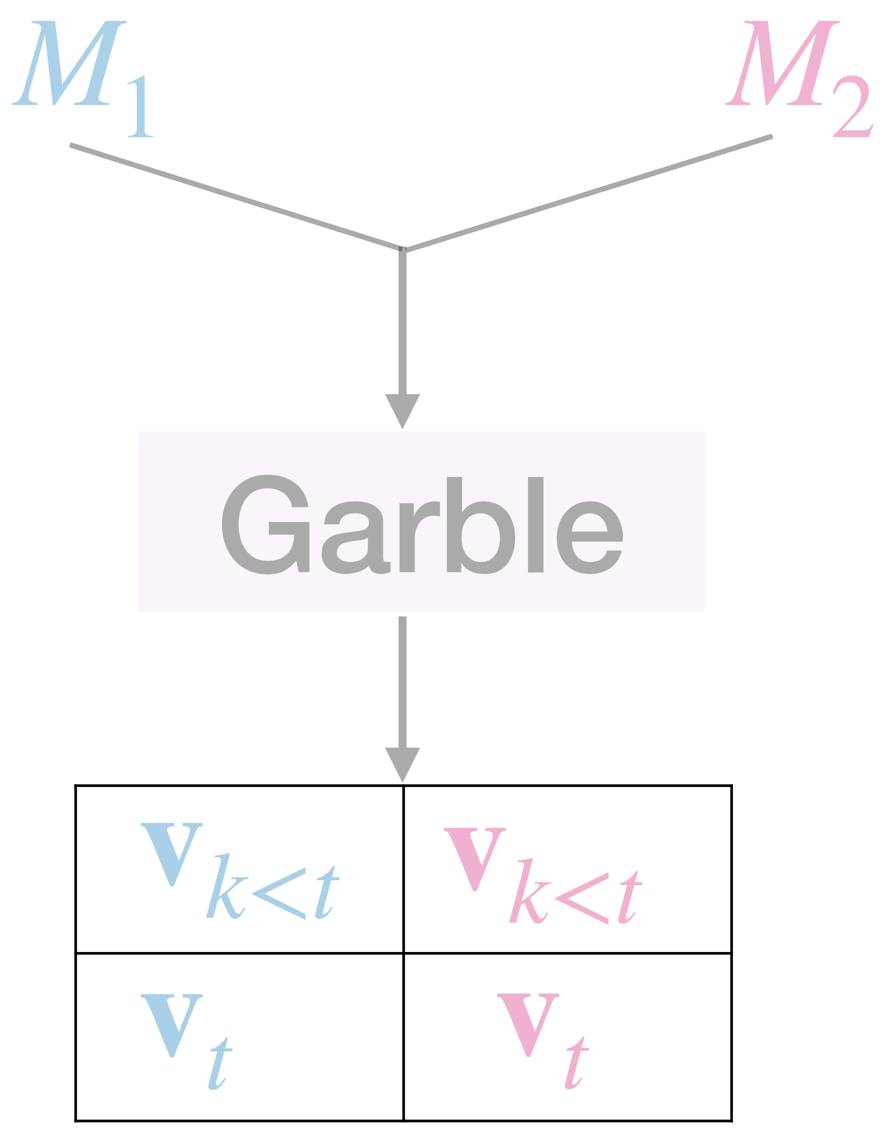
Garble for ABPs ❌

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

AKGS for TM computation [LL20]

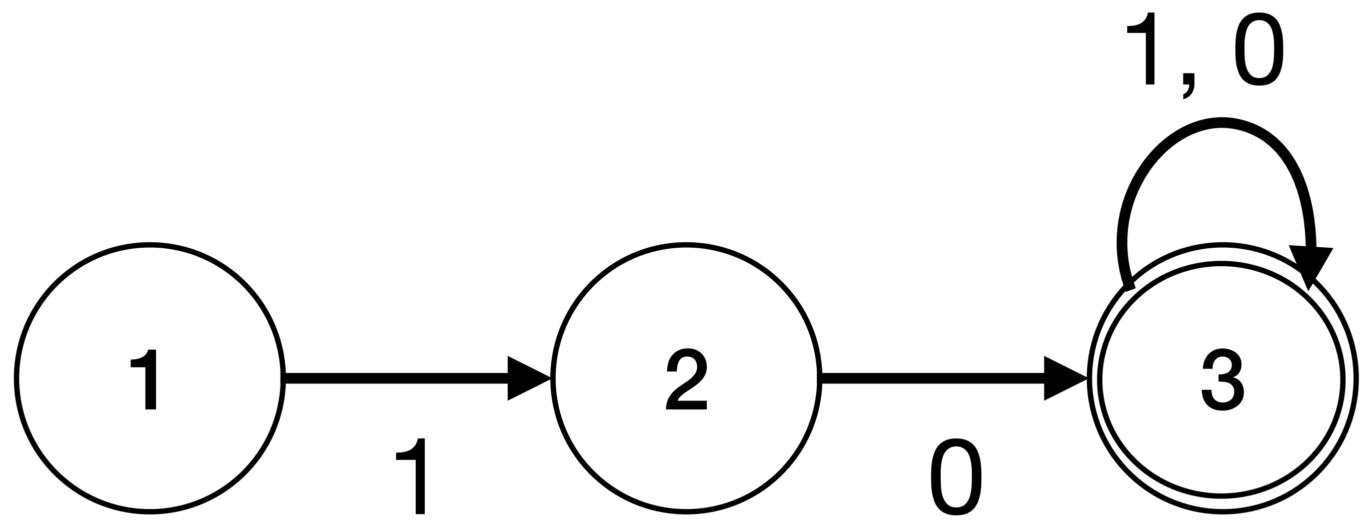
Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

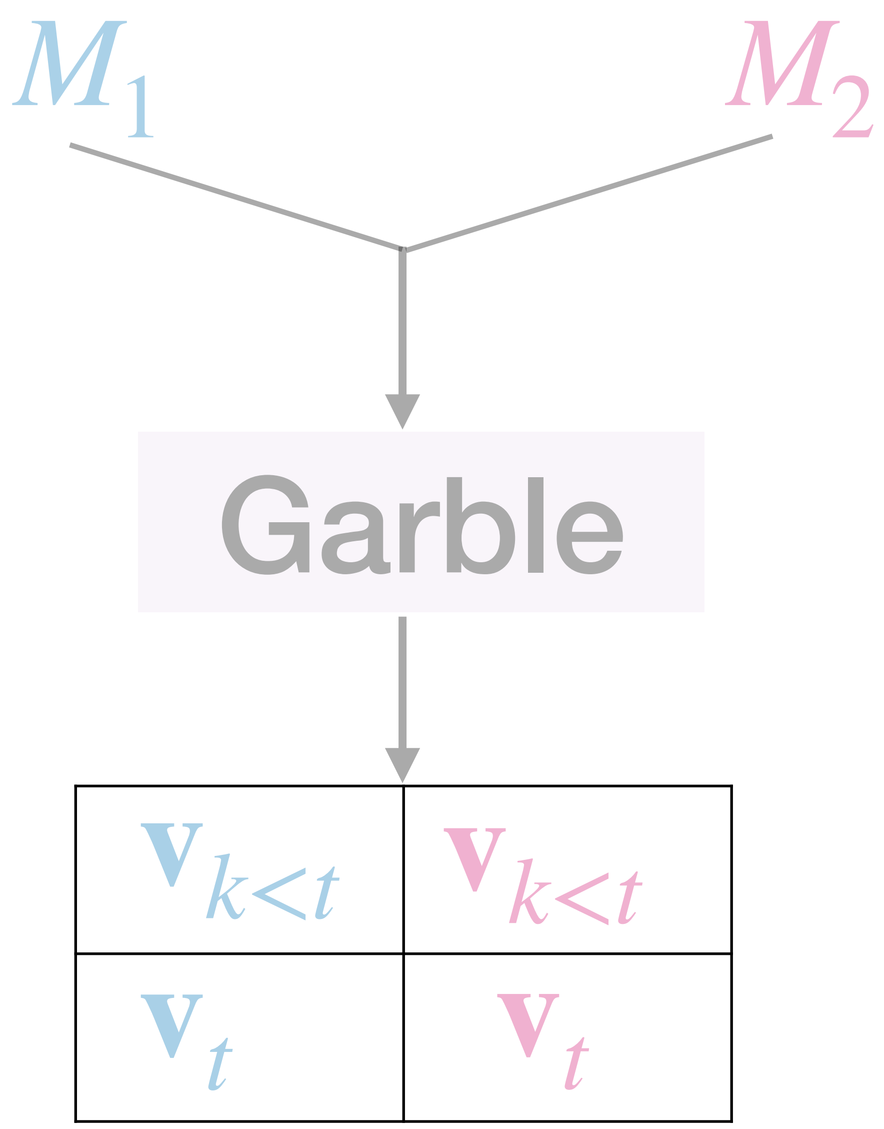
Deterministic Finite Automaton (DFA)



- ✦ states $\{1,2,\ldots,Q\}$
  - initial state 1
  - accepting state q_{acc}
- ✦ transition function δ

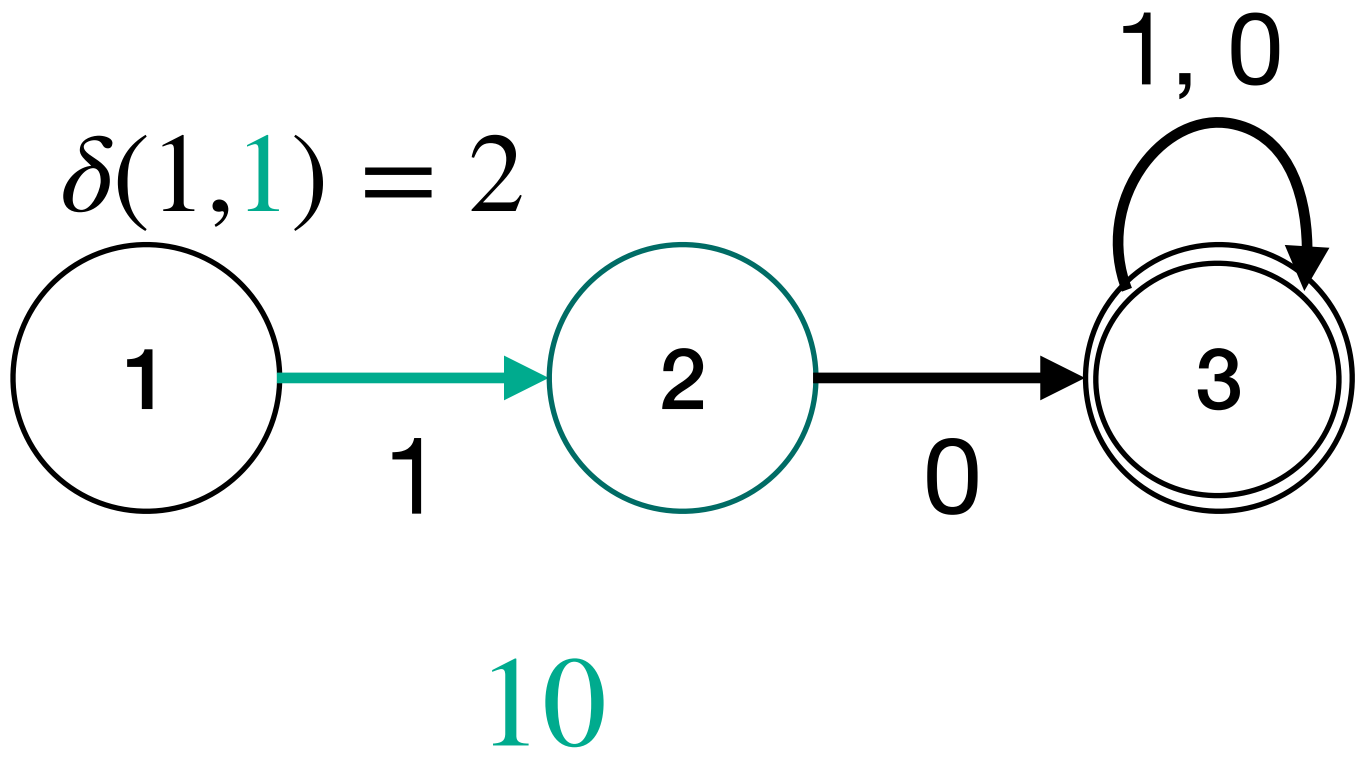
Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



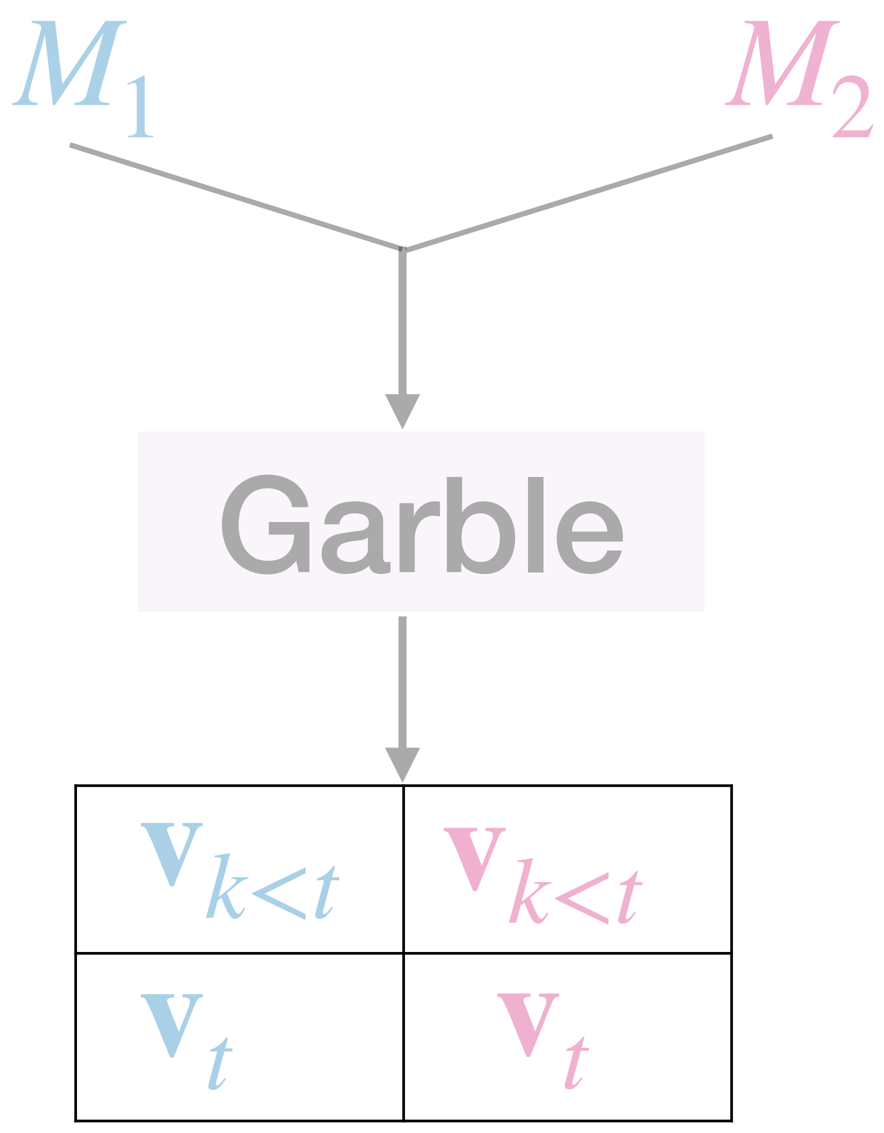
Deterministic Finite Automaton (DFA)

- ✦ states $\{1,2,\ldots,Q\}$
  - initial state 1
  - accepting state q_{acc}
- ✦ transition function δ



Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$

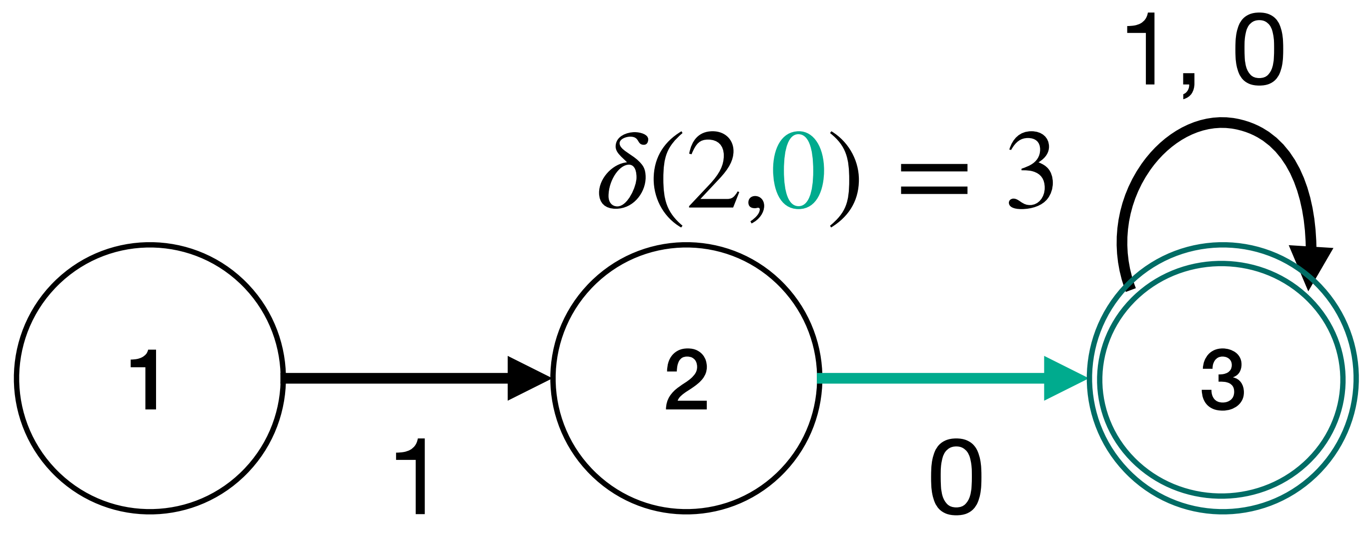


Deterministic Finite Automaton (DFA)

✦ states $\{1,2,\ldots,Q\}$

- initial state 1
- accepting state q_{acc}

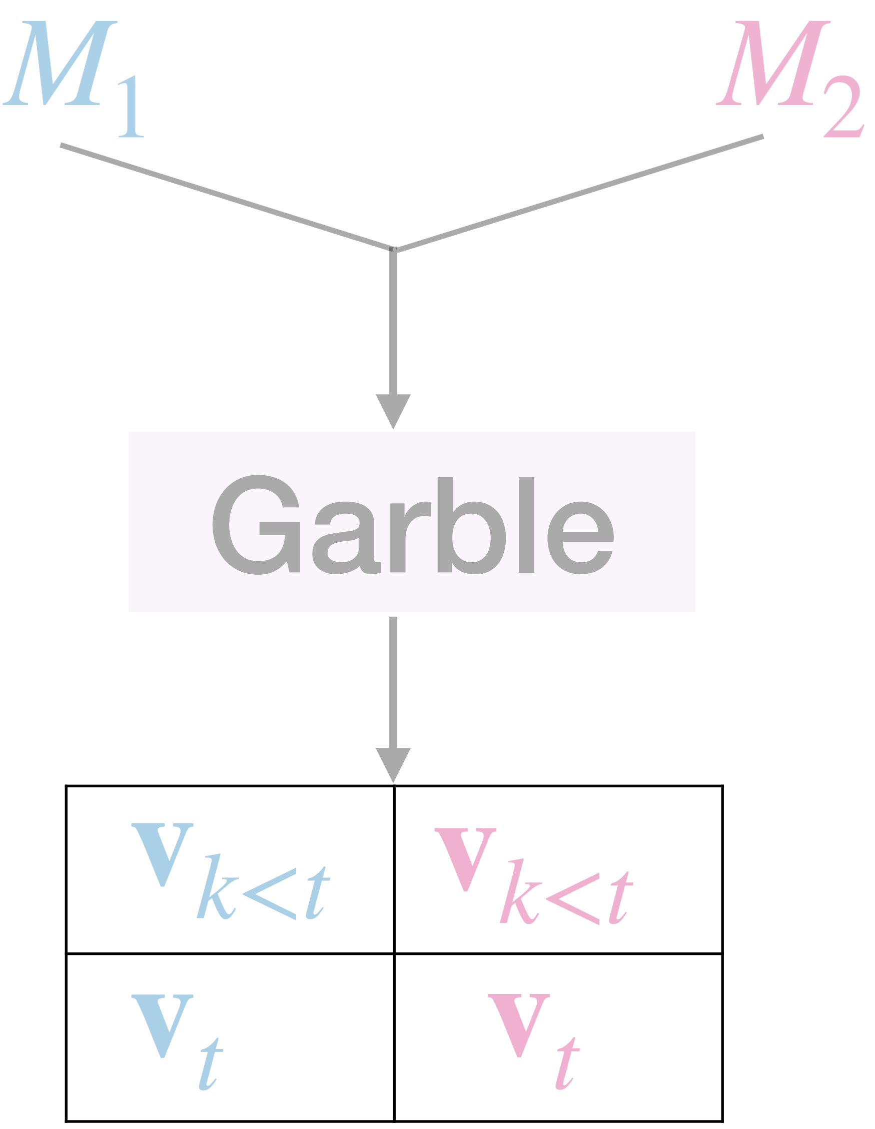
✦ transition function δ



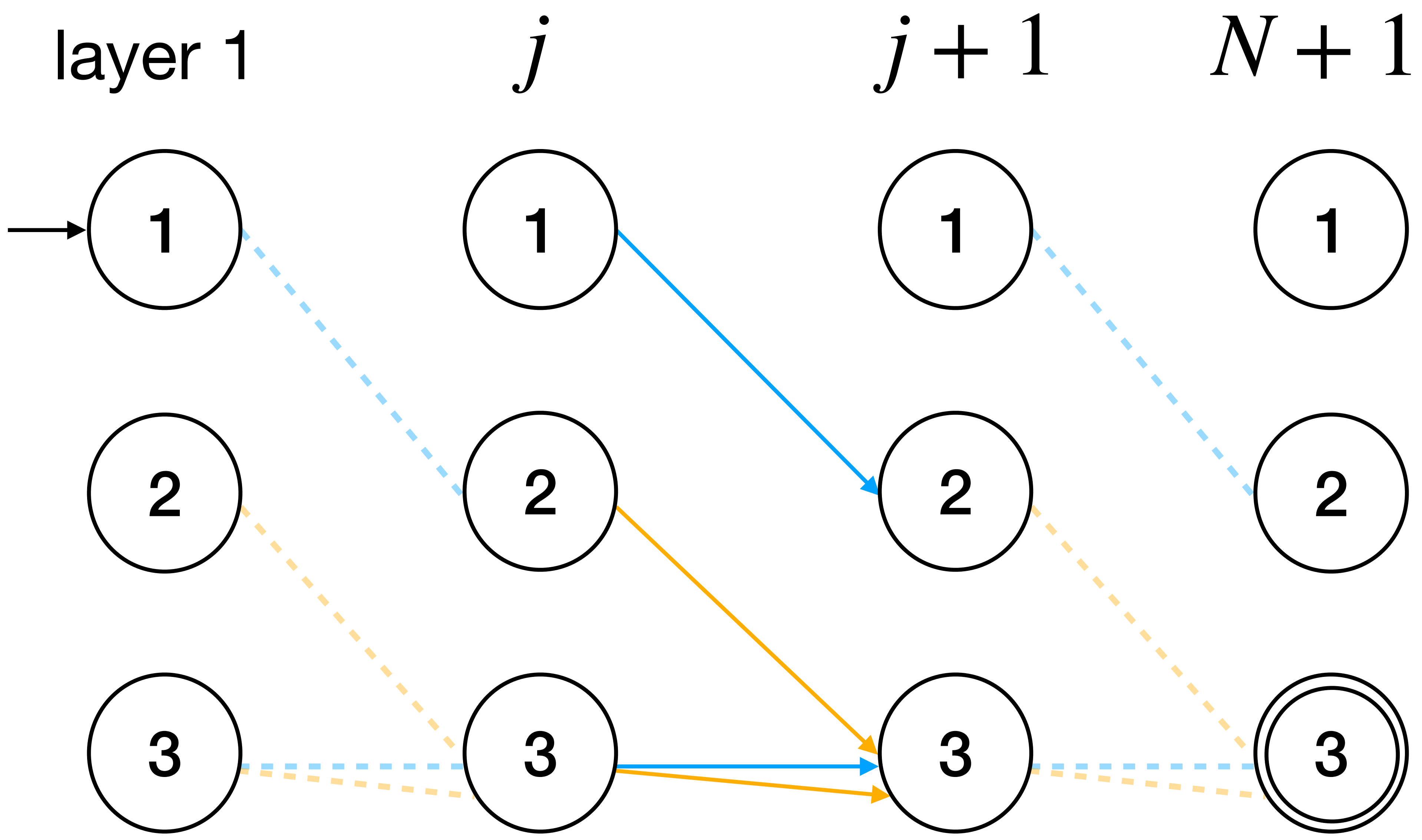
$$M(1001\cdots) = 1$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

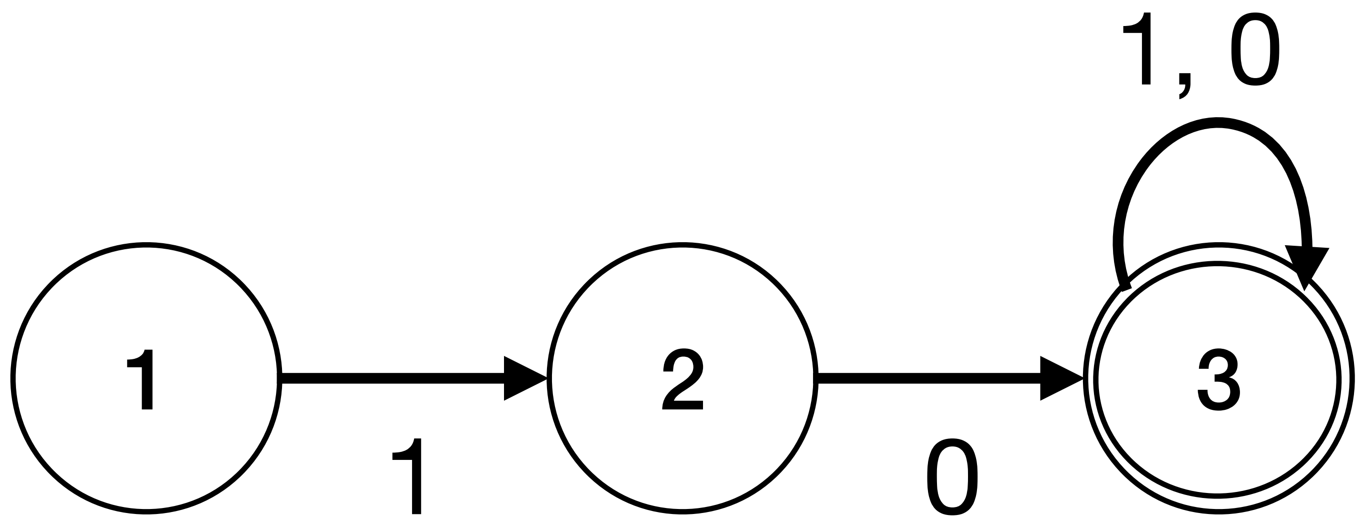
function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



DFA as Branching Program

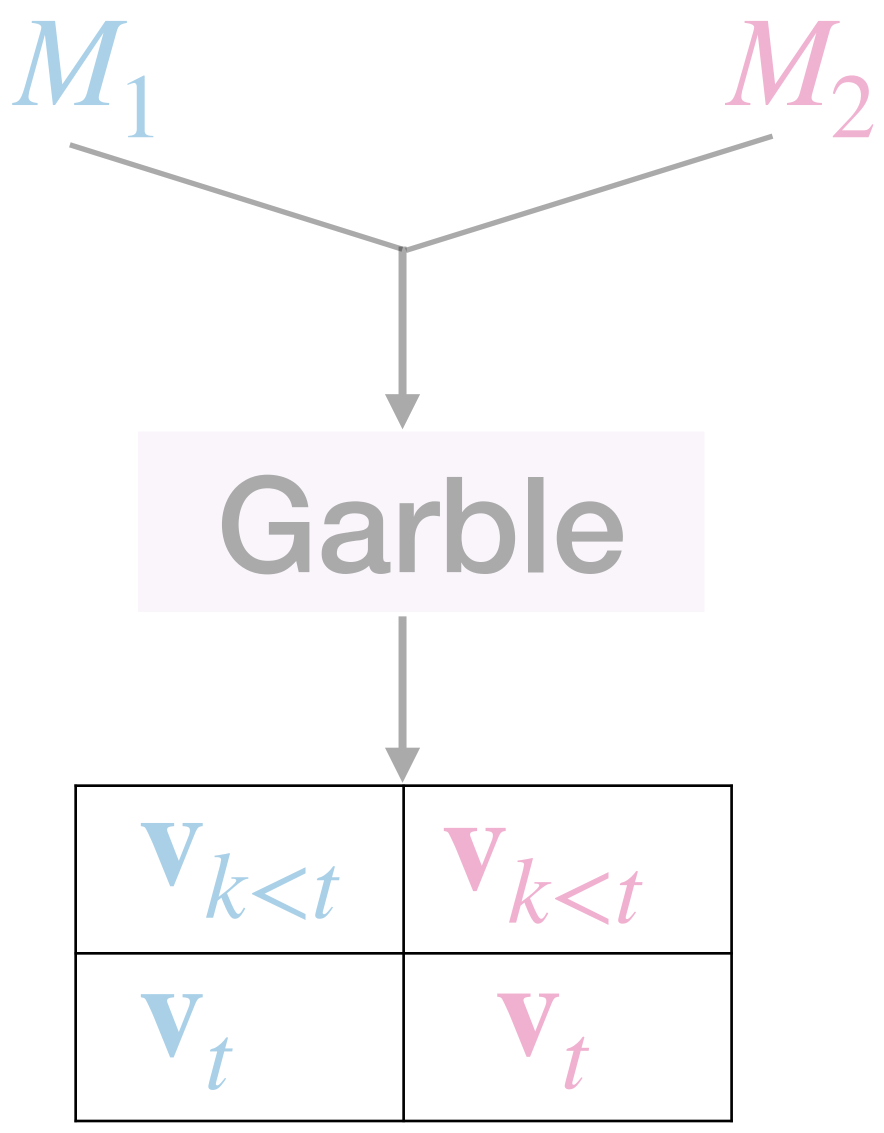


$x_j = 0$ or 1



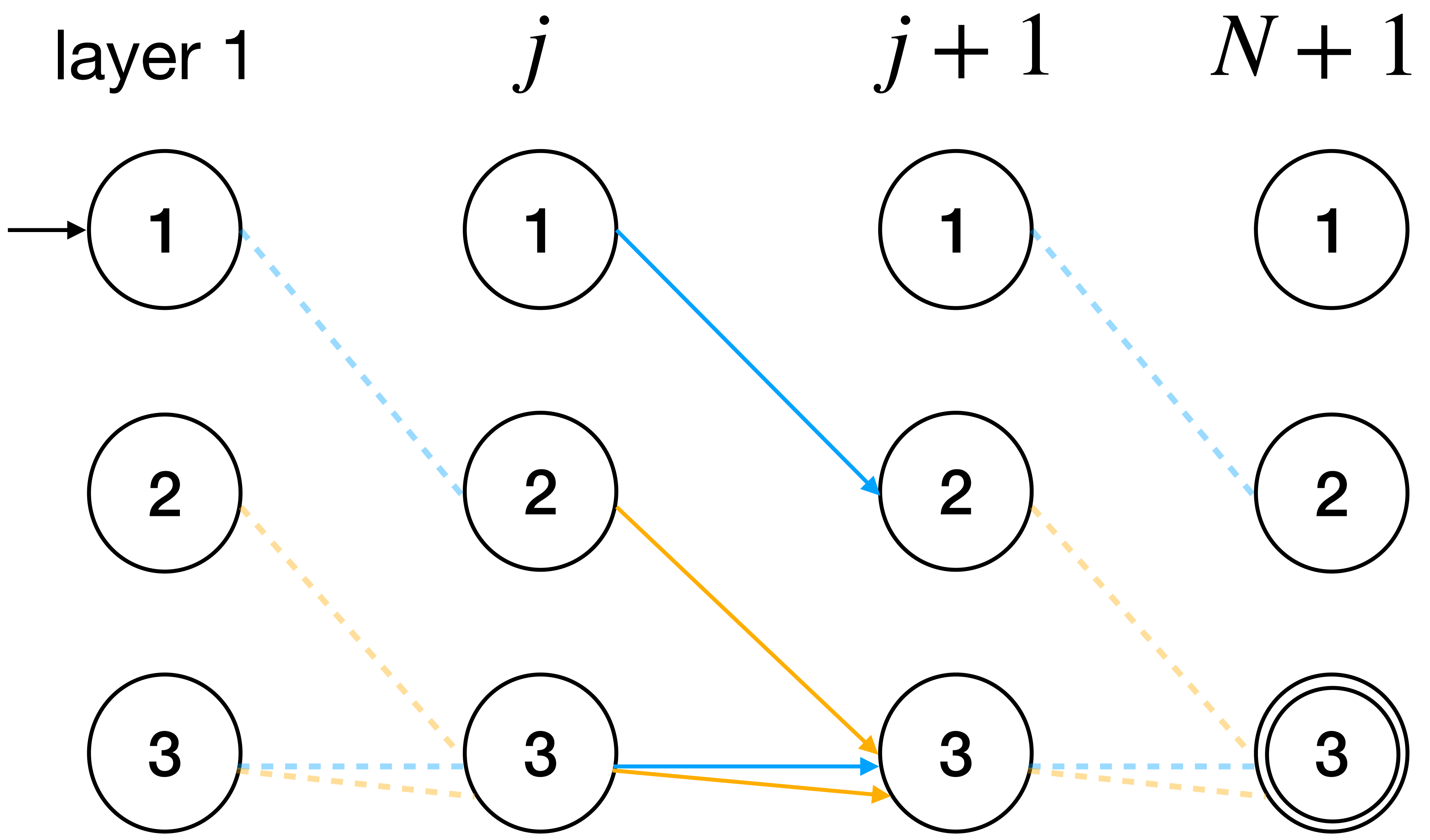
Our Idea for FE-UAWS for TMs (DFA for concreteness)

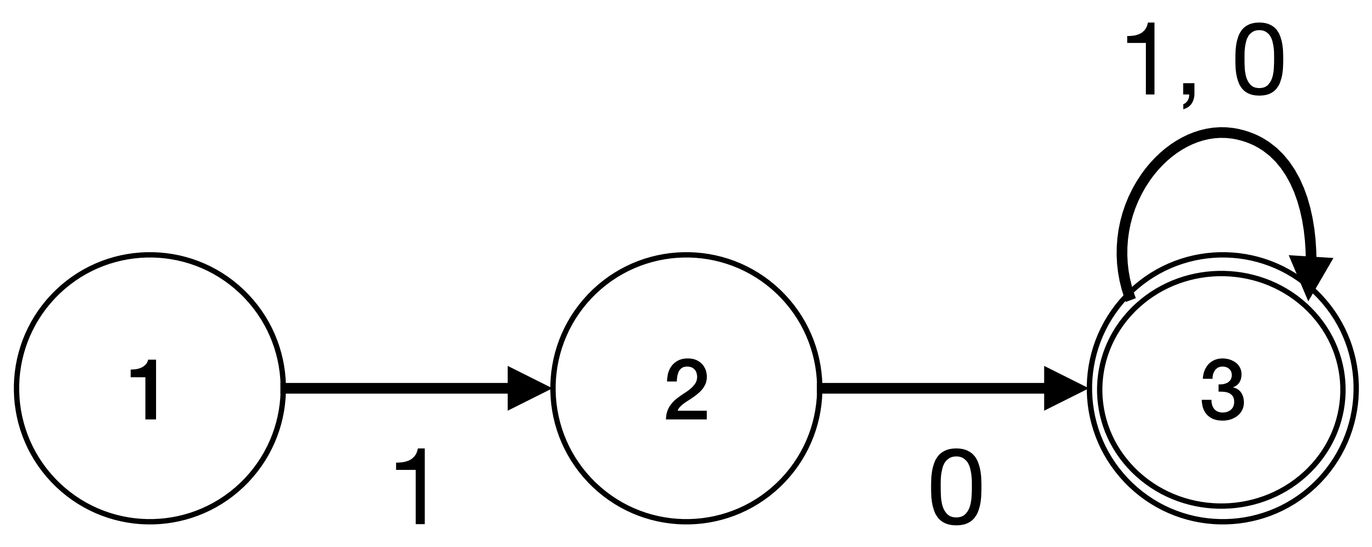
function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

DFA as Matrix Multiplication



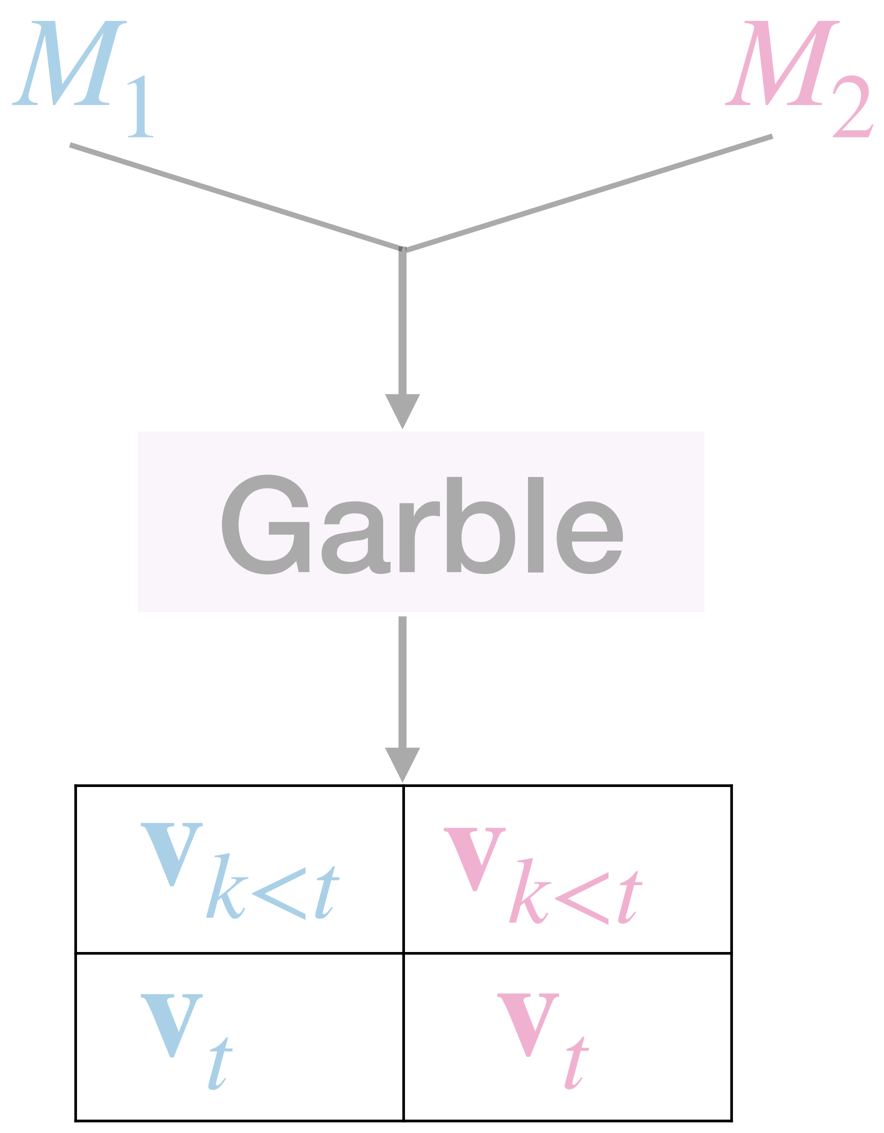


$\mathbf{M}_0$ or $\mathbf{M}_1$

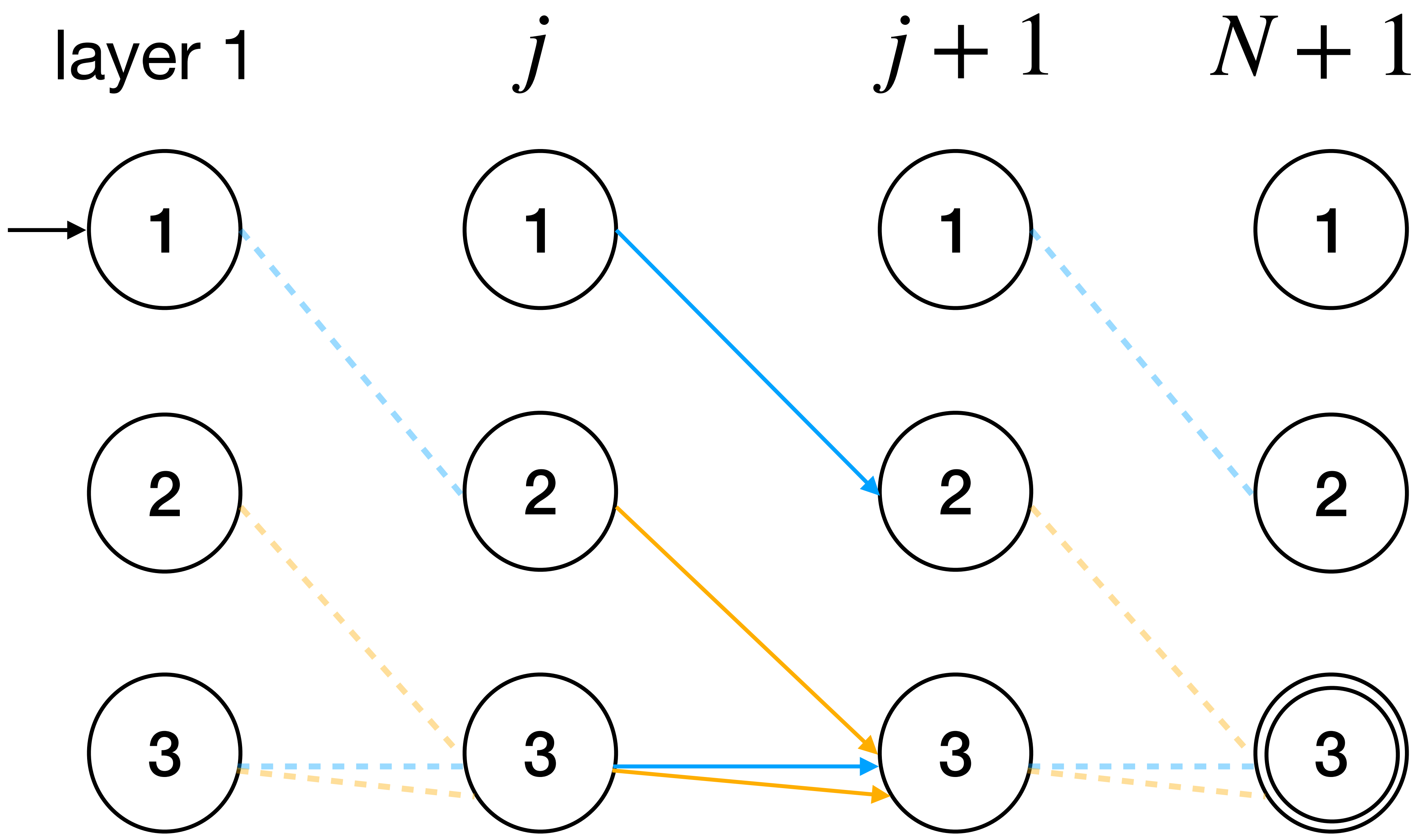
$$\mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N}$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

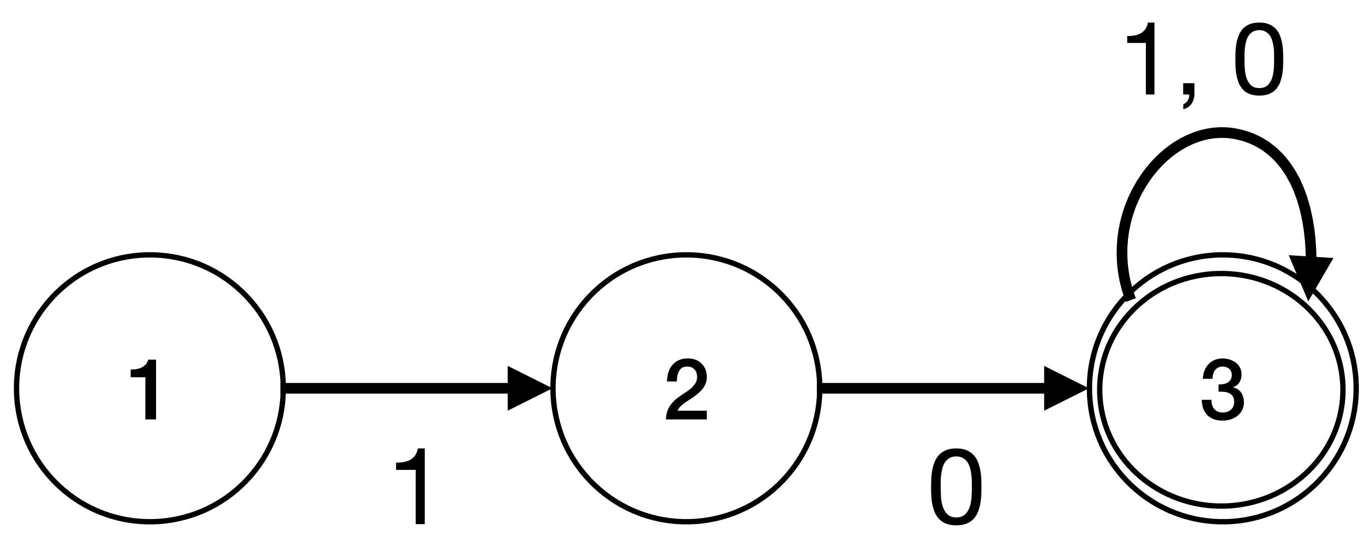
function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



DFA as Matrix Multiplication



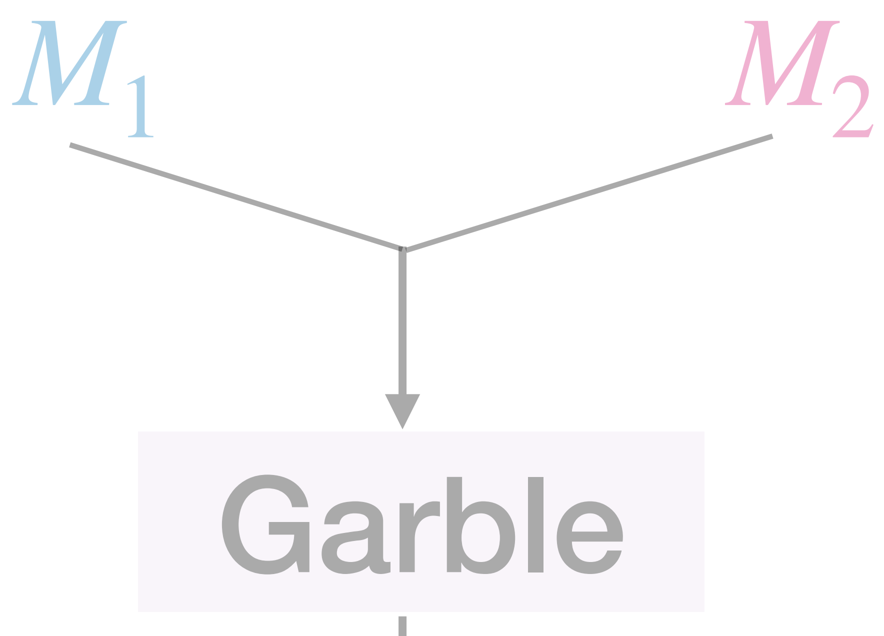
$\mathbf{M}_0$ or $\mathbf{M}_1$



$$\mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N} \mathbf{e}_{q_{\mathsf{acc}}} = 1 \text{ or } 0$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



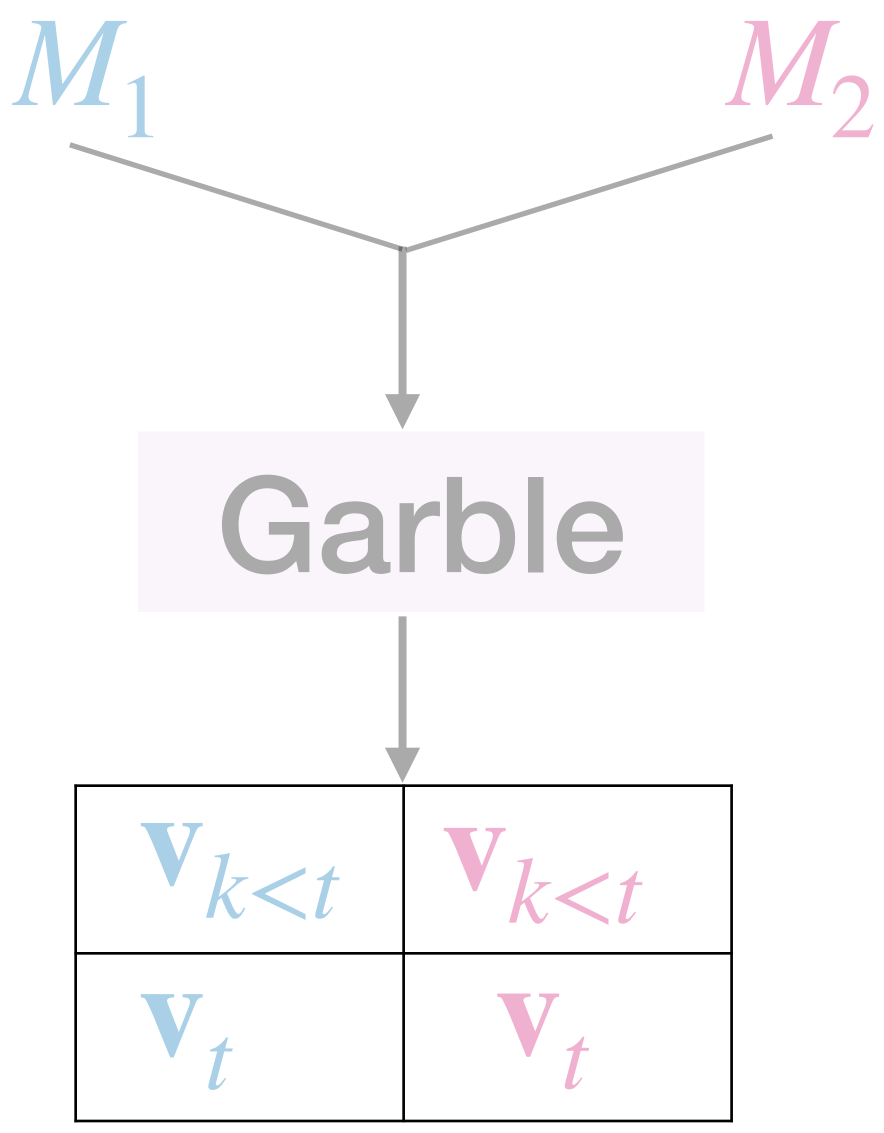
$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

Garble: $z \cdot \underbrace{\boxed{\mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N} \; \mathbf{e}_{q_{\mathsf{acc}}}}}_{\boxed{M(\mathbf{x})}} + \beta$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

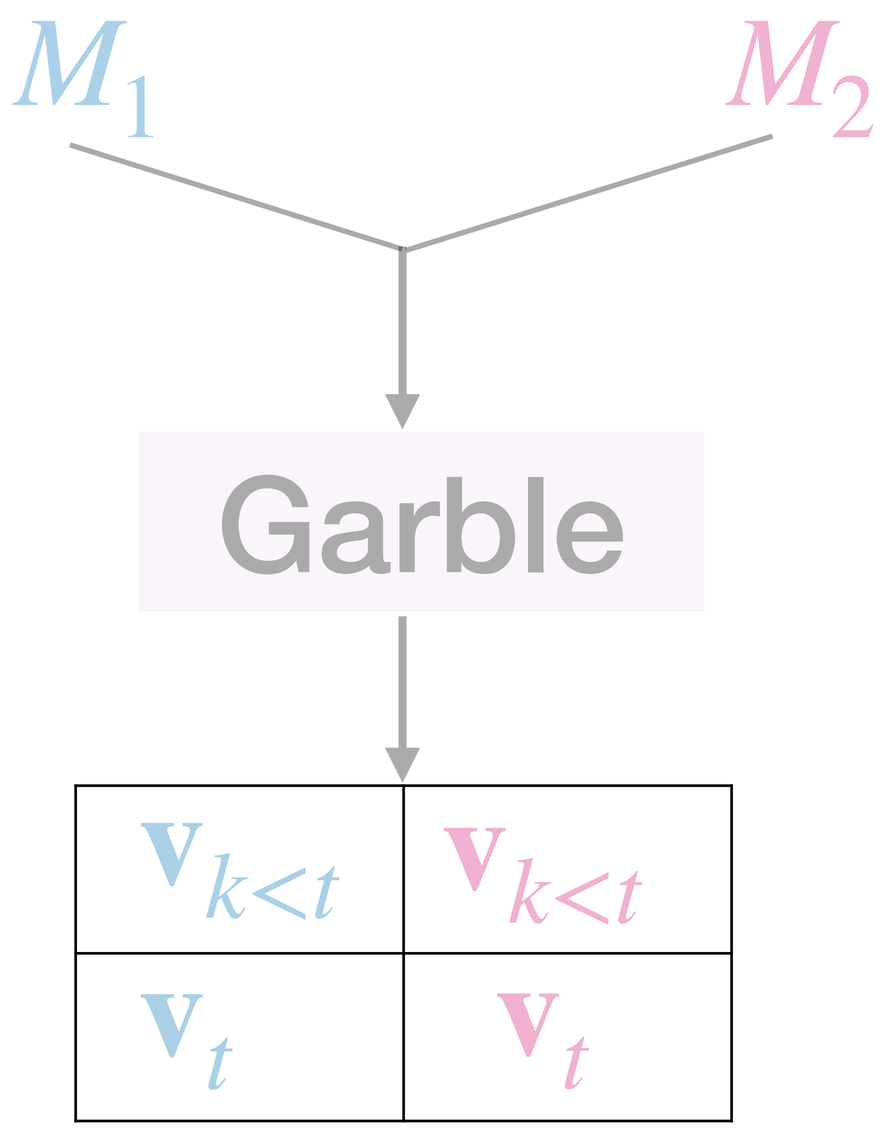
Garble: $z \cdot \underbrace{\boxed{\mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N}\ \mathbf{e}_{q_{\mathsf{acc}}}}}_{\boxed{M(\mathbf{x})}} + \beta$

$$z\, \mathbf{e}_1^\top \mathbf{M}_{x_1}\, \mathbf{e}_{q_{\mathsf{acc}}} + \beta$$

$$= \beta + \mathbf{e}_1^\top \mathbf{r}_0 + \mathbf{e}_1^\top(-\mathbf{r}_0 + \mathbf{M}_{x_1} \mathbf{r}_1) + \mathbf{e}_1^\top \mathbf{M}_{x_1} (-\mathbf{r}_1 + z\, \mathbf{e}_{q_{\mathsf{acc}}})$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\, M_1, M_2 \,)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

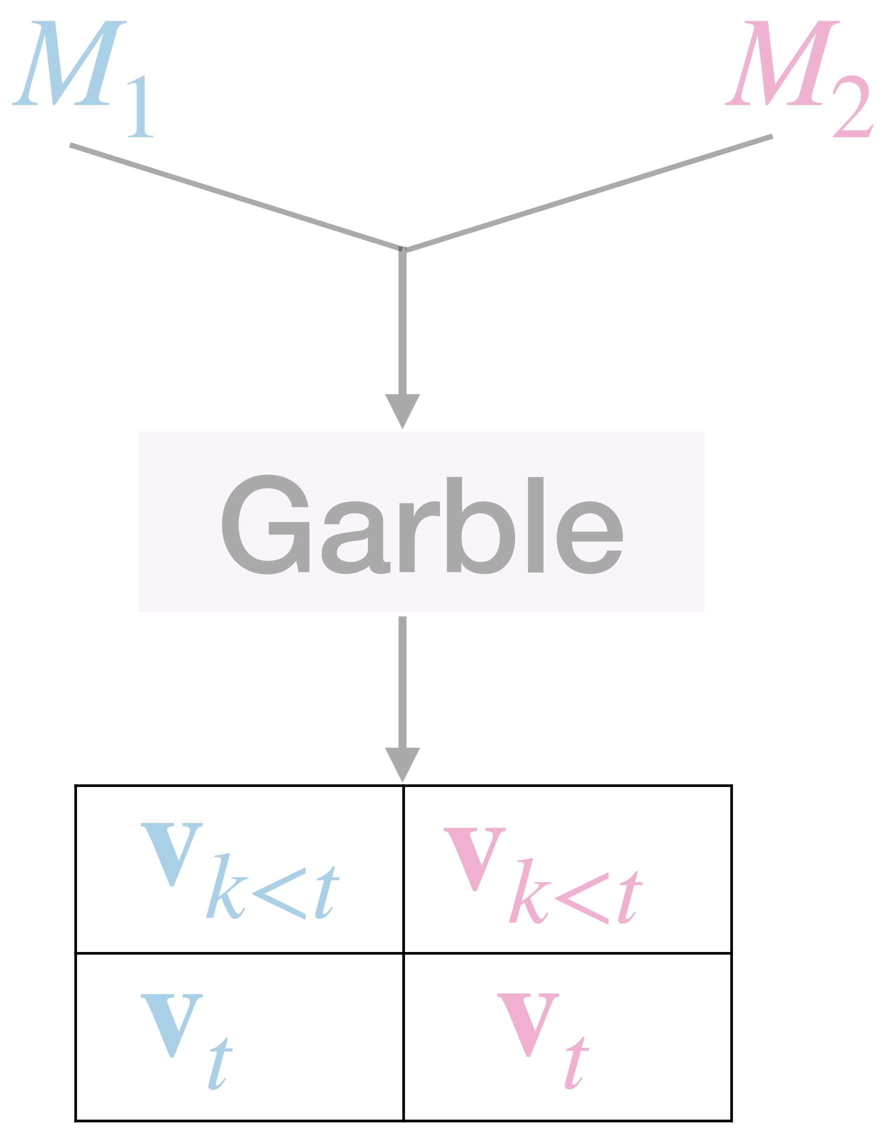
Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N} \; \mathbf{e}_{q_{\mathsf{acc}}} + \beta$

$$z \, \mathbf{e}_1^\top \mathbf{M}_{x_1} \, \mathbf{e}_{q_{\mathsf{acc}}} + \beta$$

$$= \underbrace{\beta + \mathbf{e}_1^\top \mathbf{r}_0}_{L_0(\mathbf{x})} \; + \mathbf{e}_1^\top \underbrace{(-\mathbf{r}_0 + \mathbf{M}_{x_1} \mathbf{r}_1)}_{L_1(\mathbf{x})} + \mathbf{e}_1^\top \mathbf{M}_{x_1} \underbrace{(-\mathbf{r}_1 + z \; \mathbf{e}_{q_{\mathsf{acc}}})}_{L_2(z)}$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

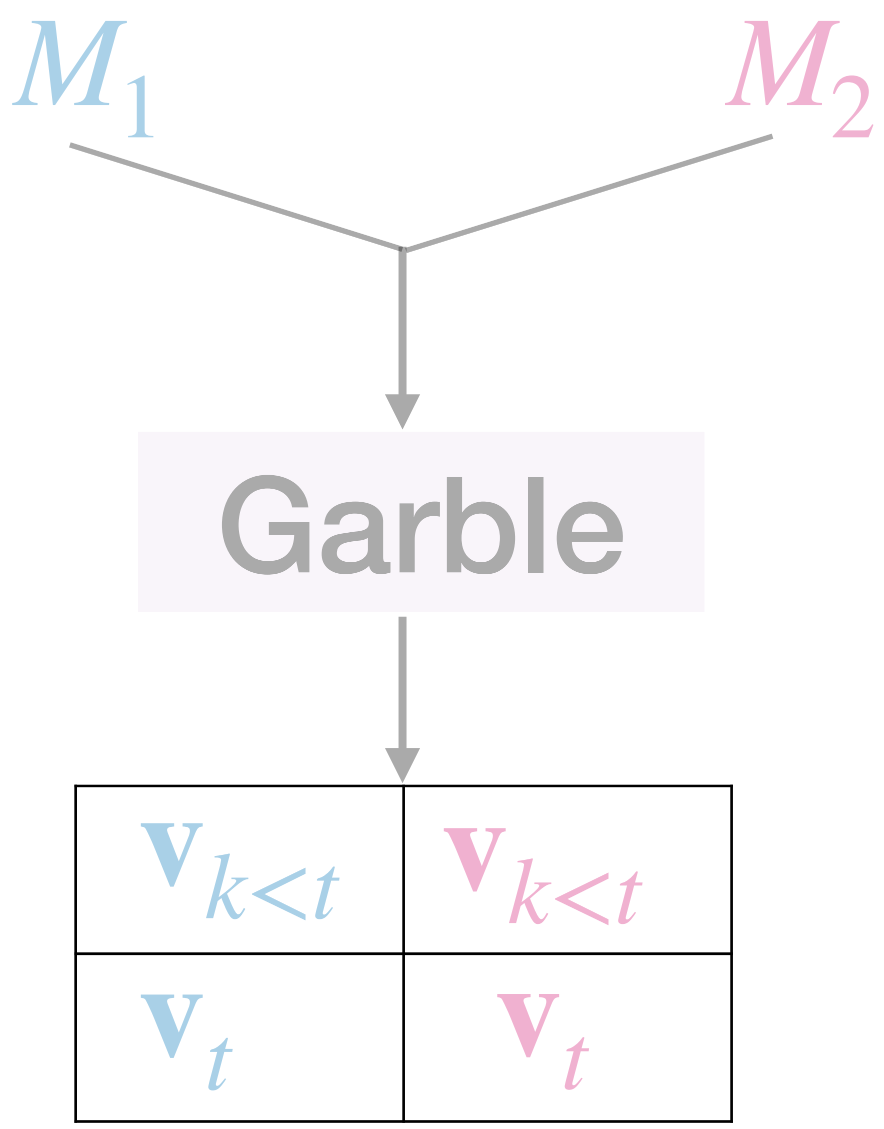
Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N} \ \mathbf{e}_{q_{\text{acc}}} + \beta$

$$z \, \mathbf{e}_1^\top \mathbf{M}_{x_1} \ \mathbf{M}_{x_2} \mathbf{e}_{q_{\text{acc}}} + \beta$$

$$= \underbrace{\beta + \mathbf{e}_1^\top \mathbf{r}_0}_{L_0(\mathbf{x})} \ \underbrace{+ \mathbf{e}_1^\top (-\mathbf{r}_0 + \mathbf{M}_{x_1} \mathbf{r}_1)}_{L_1(\mathbf{x})} \underbrace{+ \mathbf{e}_1^\top (-\mathbf{r}_1 + \mathbf{M}_{x_2} \mathbf{r}_2)}_{L_2(\mathbf{x})} + \mathbf{e}_1^\top \mathbf{M}_{x_2} \underbrace{(-\mathbf{r}_2 + z \ \mathbf{e}_{q_{\text{acc}}})}_{L_3(z)}$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$


M_1 M_2

Garble

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N}\ \mathbf{e}_{q_{\mathsf{acc}}} + \beta$

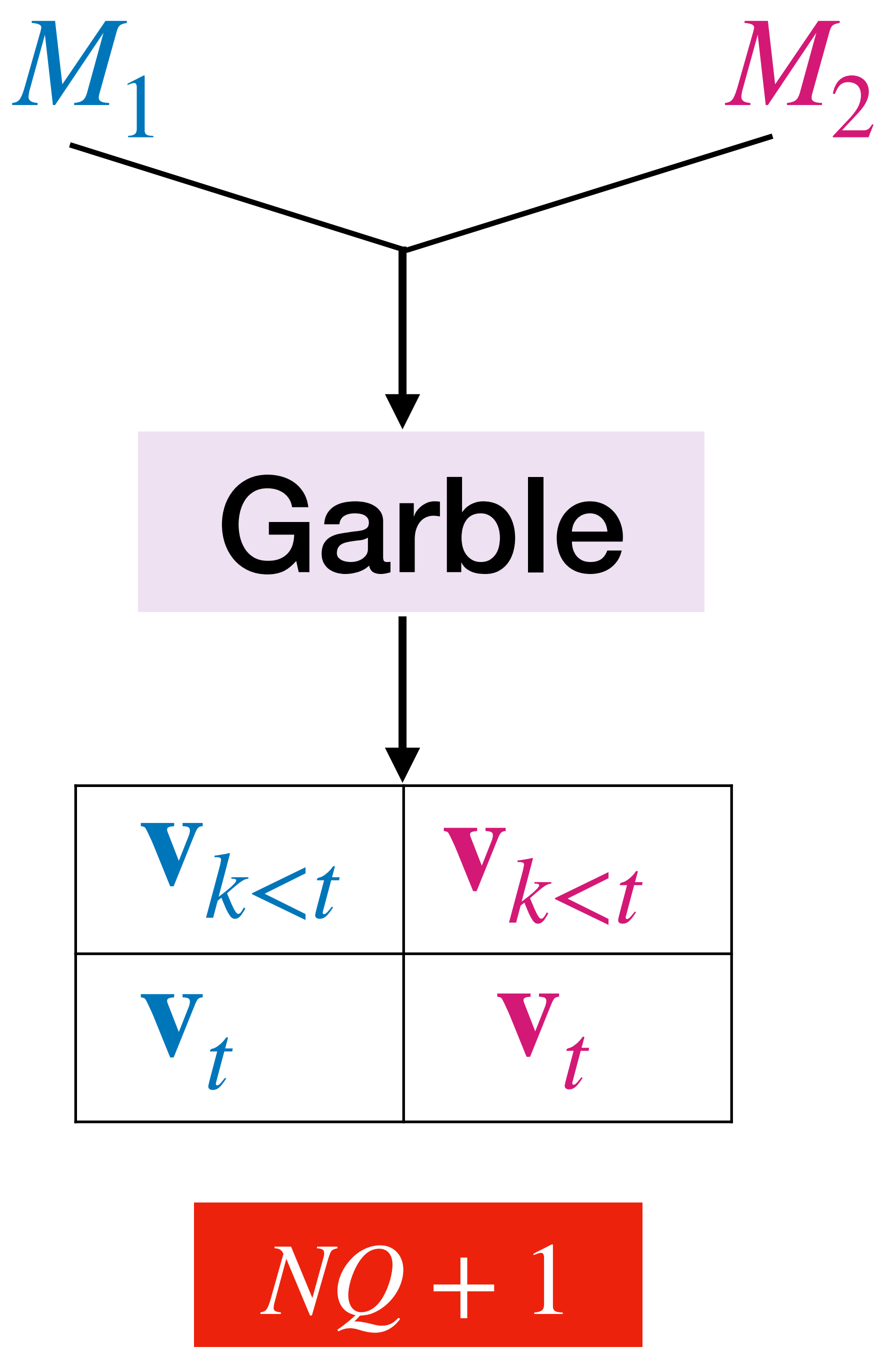
$$z\ \mathbf{e}_1^\top \mathbf{M}_{x_1}\ \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N} \mathbf{e}_{q_{\mathsf{acc}}} + \beta$$

$$= \underbrace{\beta + \mathbf{e}_1^\top \mathbf{r}_0}_{L_0(\mathbf{x})}\ + \underbrace{\mathbf{e}_1^\top(-\mathbf{r}_0 + \mathbf{M}_{x_1}\mathbf{r}_1)}_{L_1(\mathbf{x})}\ + \underbrace{\mathbf{e}_1^\top(-\mathbf{r}_1 + \mathbf{M}_{x_2}\mathbf{r}_2)}_{L_2(\mathbf{x})}\ + \cdots + \underbrace{\mathbf{e}_1^\top \mathbf{M}_{x_N}(-\mathbf{r}_N + z\ \mathbf{e}_{q_{\mathsf{acc}}})}_{L_{N+1}(z)}$$

$\boxed{\text{total size} = NQ + 1}$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N}\ \mathbf{e}_{q_{\text{acc}}} + \beta$

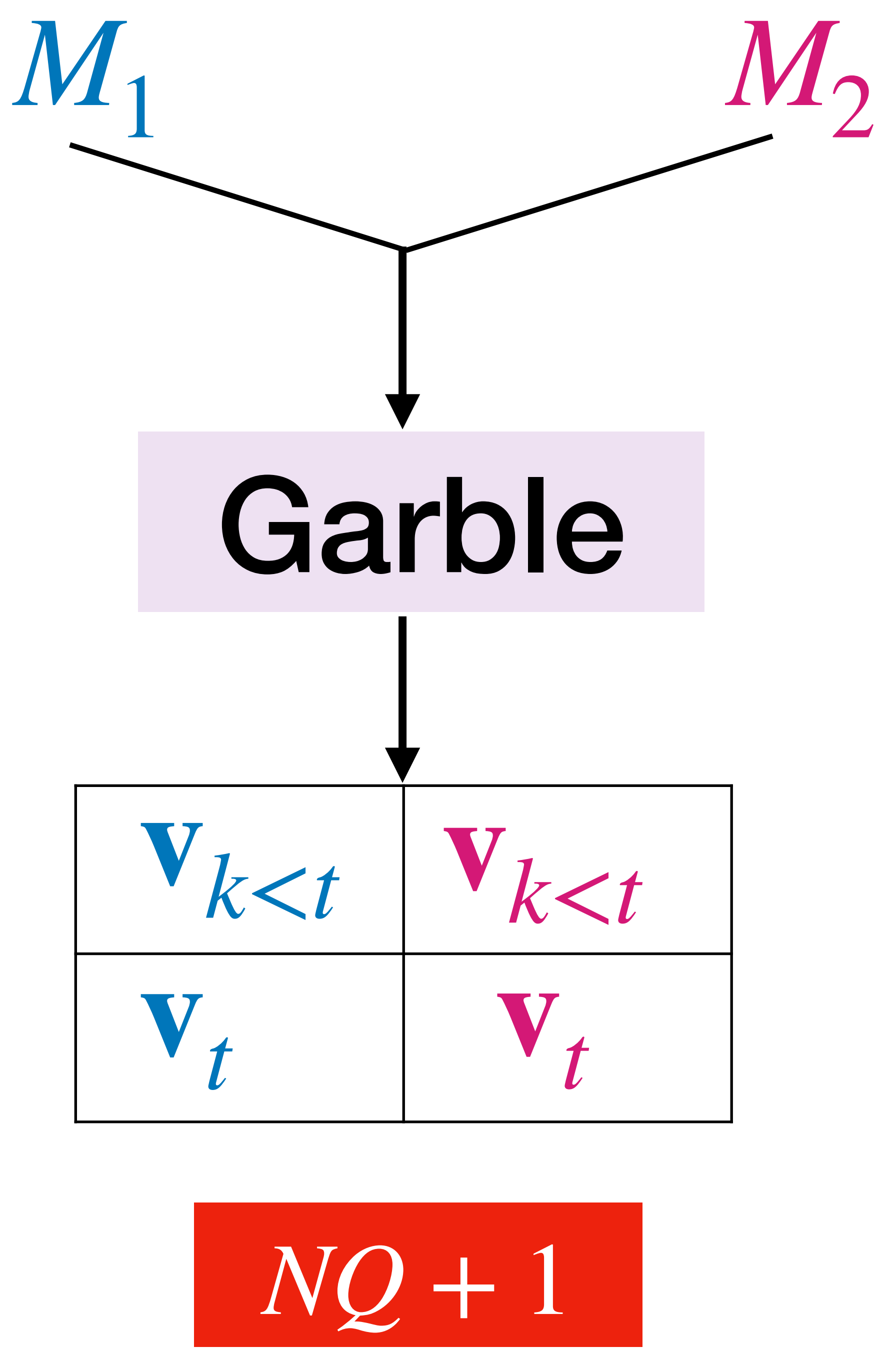
$$z\ \mathbf{e}_1^\top \mathbf{M}_{x_1}\ \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N} \mathbf{e}_{q_{\text{acc}}} + \beta$$

$$= \underbrace{\beta + \mathbf{e}_1^\top \mathbf{r}_0}_{L_0(\mathbf{x})}\ + \mathbf{e}_1^\top \underbrace{(-\mathbf{r}_0 + \mathbf{M}_{x_1} \mathbf{r}_1)}_{L_1(\mathbf{x})}\ + \mathbf{e}_1^\top \underbrace{(-\mathbf{r}_1 + \mathbf{M}_{x_2} \mathbf{r}_2)}_{L_2(\mathbf{x})}\ \ + \cdots + \mathbf{e}_1^\top \mathbf{M}_{x_N} \underbrace{(-\mathbf{r}_N + z\ \mathbf{e}_{q_{\text{acc}}})}_{L_{N+1}(z)}$$

total size = $NQ + 1$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

Garble

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

$NQ + 1$

AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

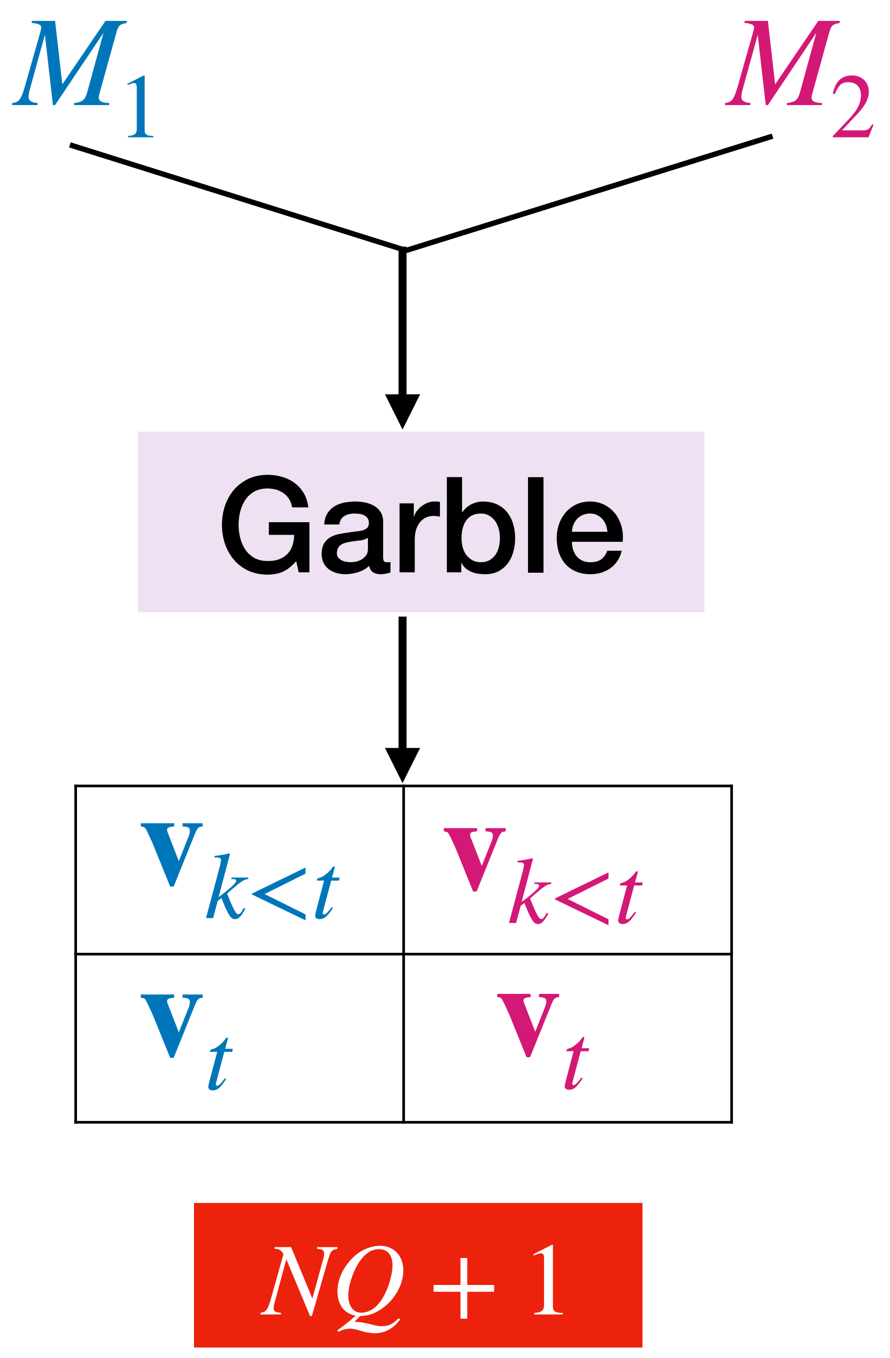
Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N}\ \mathbf{e}_{q_{\text{acc}}} + \beta$

$$L_{q,j}(\mathbf{x}) = -\mathbf{r}_{j-1}[q] + (\mathbf{M}_{x_j} \mathbf{r}_j)[q]$$

$$\beta + \mathbf{e}_1^\top \mathbf{r}_0\ + \mathbf{e}_1^\top(-\mathbf{r}_0 + \mathbf{M}_{x_1} \mathbf{r}_1)\ + \mathbf{e}_1^\top(-\mathbf{r}_1 + \mathbf{M}_{x_2} \mathbf{r}_2)\ \ + \cdots + \mathbf{e}_1^\top \mathbf{M}_{x_N} (-\mathbf{r}_N + z\ \mathbf{e}_{q_{\text{acc}}})$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

Garble

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

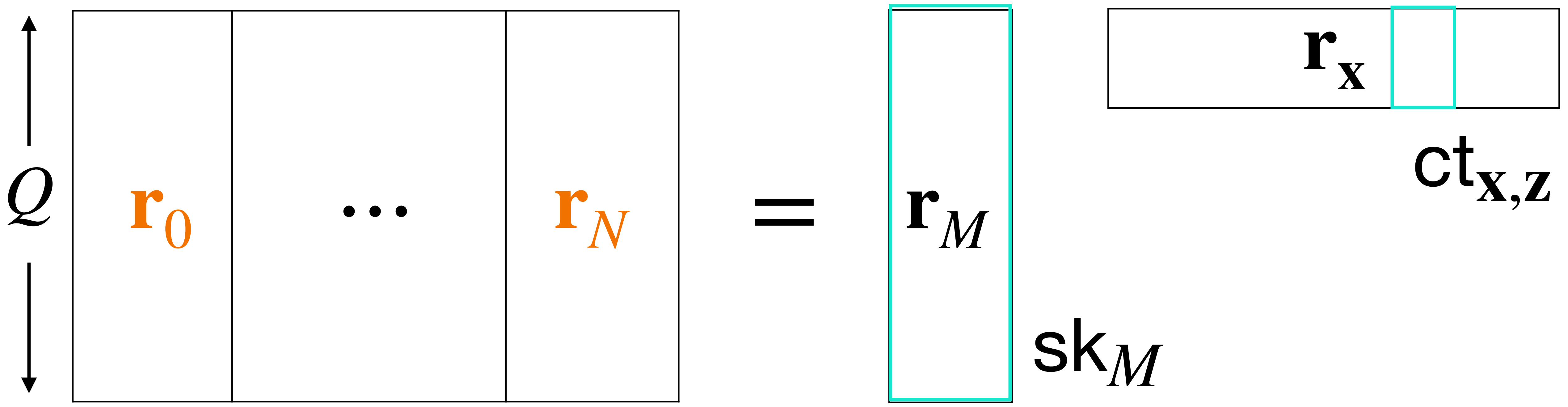
$NQ + 1$

AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N} \; \mathbf{e}_{q\text{acc}} + \beta$

$$L_{q,j}(\mathbf{x}) = -\mathbf{r}_{j-1}[q] + (\mathbf{M}_{x_j} \mathbf{r}_j)[q]$$

$$\beta + \mathbf{e}_1^\top \mathbf{r}_0 \; + \mathbf{e}_1^\top(-\mathbf{r}_0 + \mathbf{M}_{x_1} \mathbf{r}_1) \; + \mathbf{e}_1^\top(-\mathbf{r}_1 + \mathbf{M}_{x_2} \mathbf{r}_2) \; + \cdots + \mathbf{e}_1^\top \mathbf{M}_{x_N} (-\mathbf{r}_N + z \; \mathbf{e}_{q\text{acc}})$$



Q $\mathbf{r}_0$ $\cdots$ $\mathbf{r}_N$ $=$ $\mathbf{r}_M$

sk_M

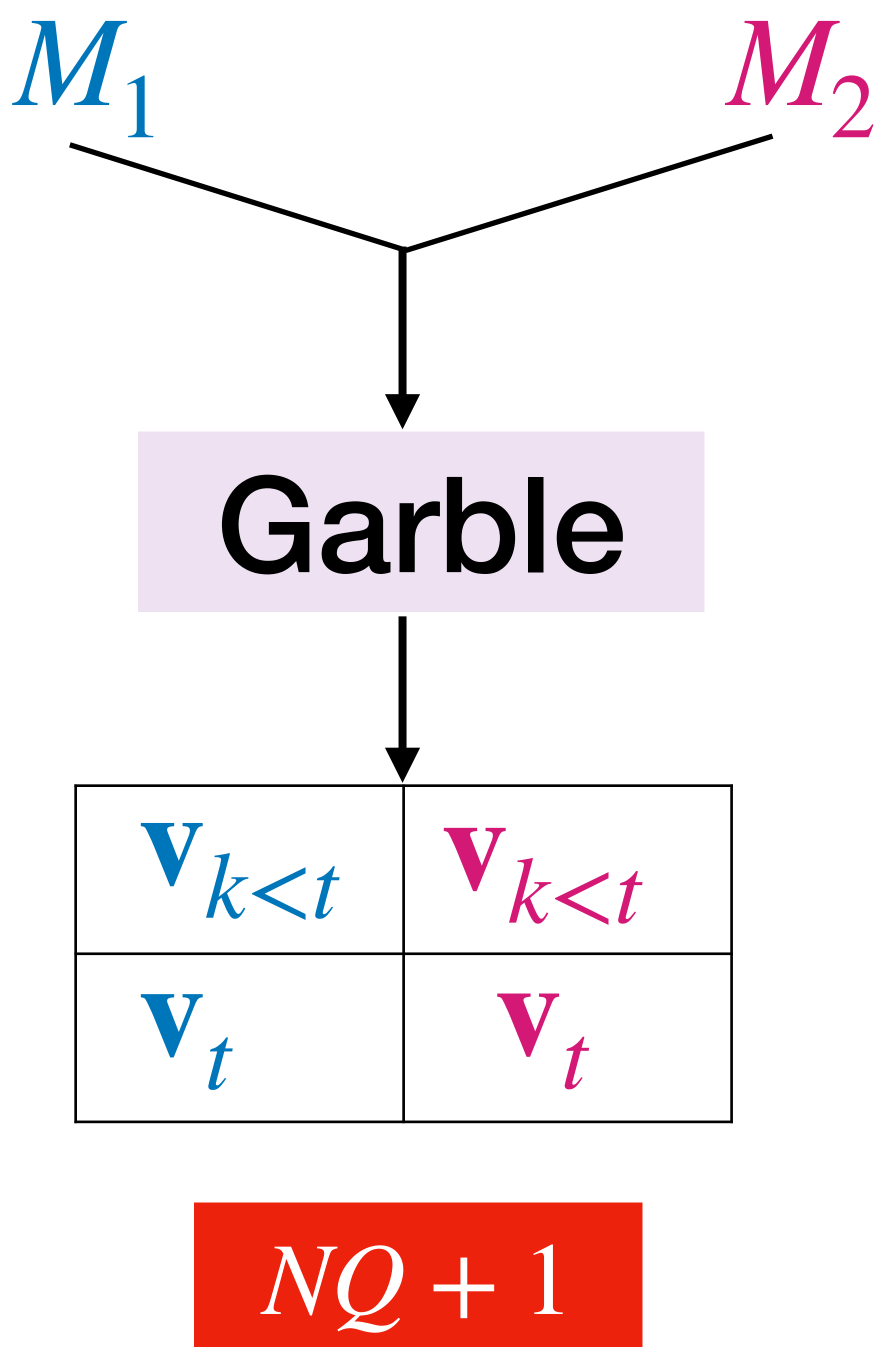
$\mathbf{r}_\mathbf{x}$

$\mathsf{ct}_{\mathbf{x},\mathbf{z}}$

$$\mathbf{r}_j = \mathbf{r}_\mathbf{x}[j] \cdot \mathbf{r}_M$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

Garble

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

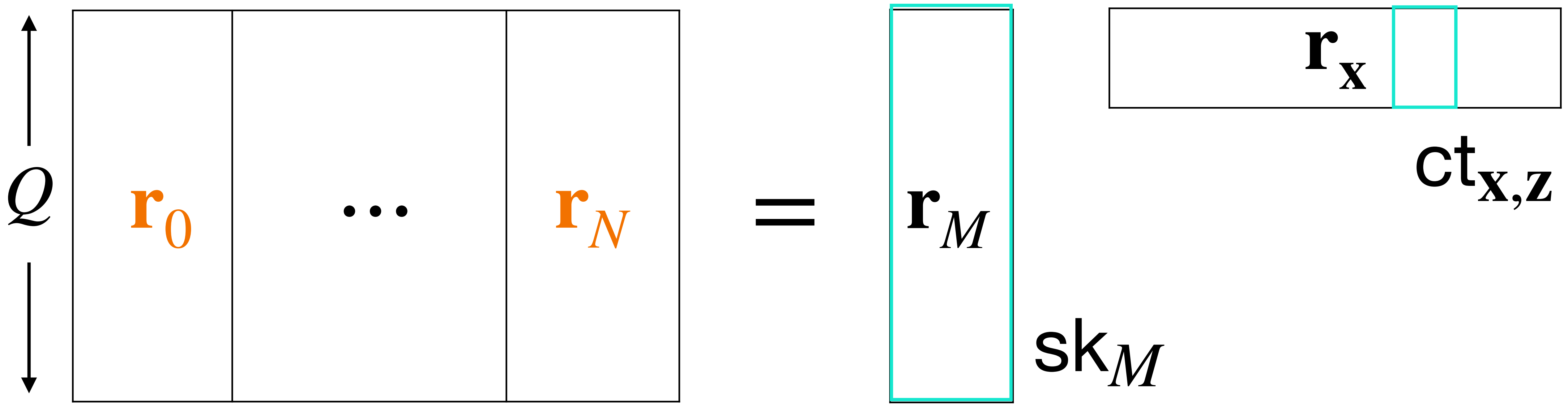
$NQ+1$

AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N}\ \mathbf{e}_{q\text{acc}} + \beta$

$$\begin{aligned} L_{q,j}(\mathbf{x}) &= -\ \mathbf{r}_{j-1}[q] + (\mathbf{M}_{x_j}\mathbf{r}_j)[q] \\ &= (\mathbf{r}_{\mathbf{x}}[j-1], \mathbf{r}_{\mathbf{x}}[j]) \cdot (-\mathbf{r}_M[q], (\mathbf{M}_{x_j}\mathbf{r}_M)[q]) \end{aligned}$$

$$\beta + \mathbf{e}_1^\top \mathbf{r}_0\ + \mathbf{e}_1^\top(-\mathbf{r}_0 + \mathbf{M}_{x_1}\mathbf{r}_1)\ + \mathbf{e}_1^\top(-\mathbf{r}_1 + \mathbf{M}_{x_2}\mathbf{r}_2)\ + \cdots + \mathbf{e}_1^\top \mathbf{M}_{x_N}(-\mathbf{r}_N + z\ \mathbf{e}_{q\text{acc}})$$



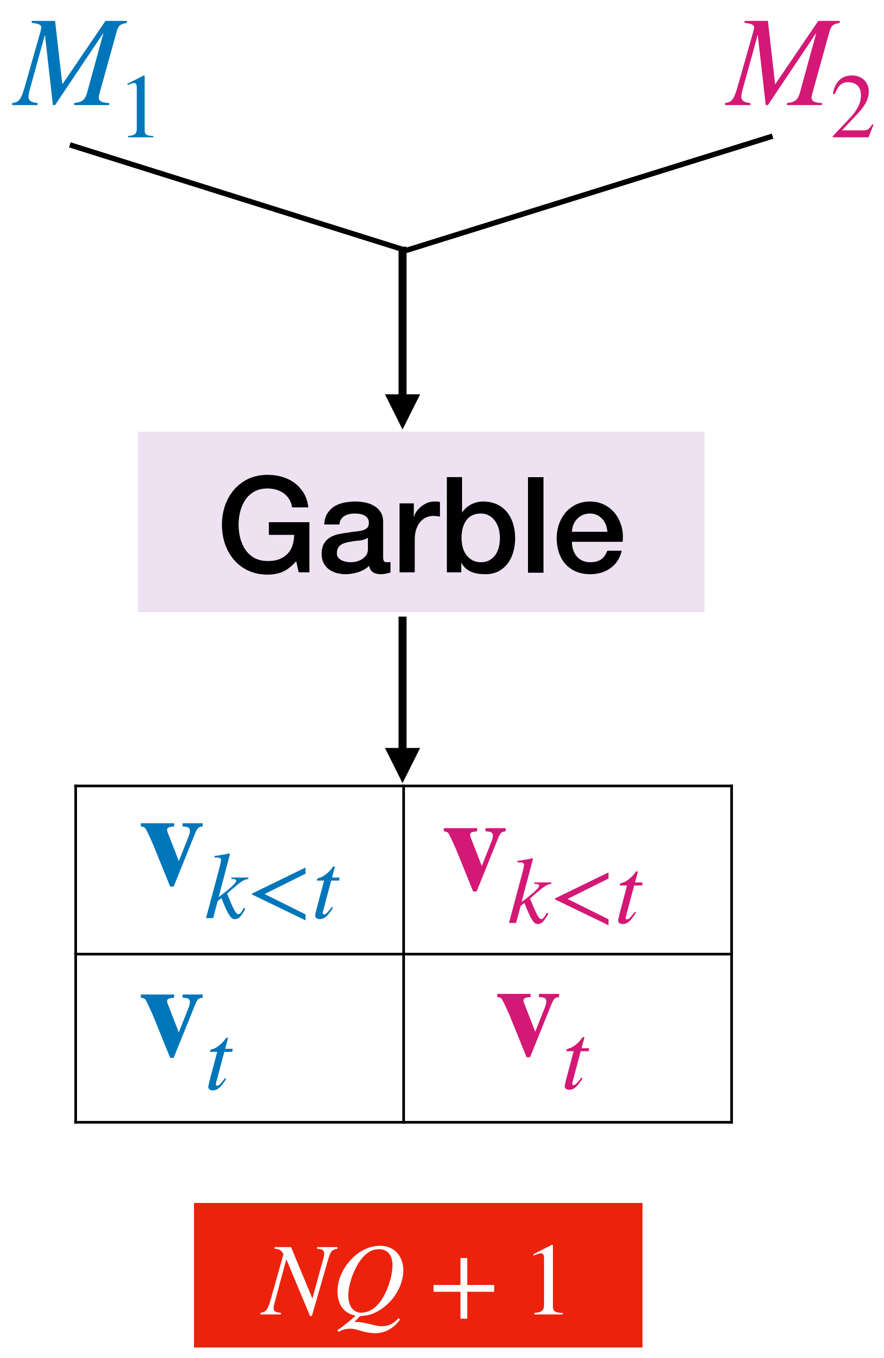
Q $\mathbf{r}_0$ $\cdots$ $\mathbf{r}_N$ $=$ $\mathbf{r}_M$ sk_M

$\mathbf{r}_{\mathbf{x}}$ $\text{ct}_{\mathbf{x},\mathbf{z}}$

$$\mathbf{r}_j = \mathbf{r}_{\mathbf{x}}[j] \cdot \mathbf{r}_M$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

Garble

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

$NQ + 1$

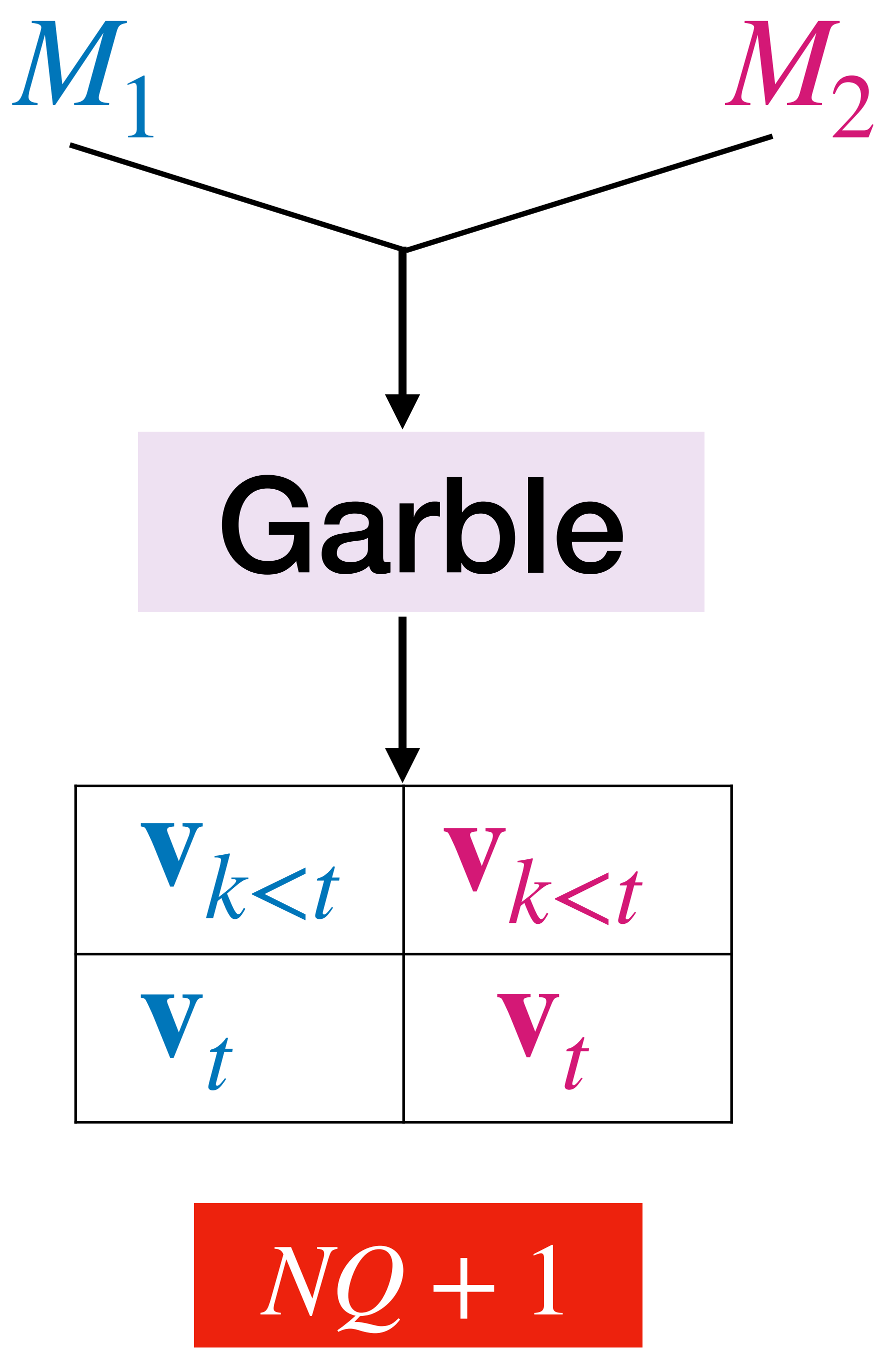
AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1}\mathbf{M}_{x_2}\cdots\mathbf{M}_{x_N}\ \mathbf{e}_{q_{\mathsf{acc}}} + \beta$

$$\begin{aligned} L_{q,j}(\mathbf{x}) &= -\,\mathbf{r}_{j-1}[q] + (\mathbf{M}_{x_j}\mathbf{r}_j)[q] \\ &= \underbrace{(\mathbf{r}_{\mathbf{x}}[j-1], \mathbf{r}_{\mathbf{x}}[j])}_{\text{Known to the } \textbf{Encrypter}} \cdot \underbrace{(-\mathbf{r}_M[q], (\mathbf{M}_{x_j}\mathbf{r}_M)[q])}_{\text{Known to the } \textbf{Key Generator}} \end{aligned}$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

Garble

$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

$NQ + 1$

AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

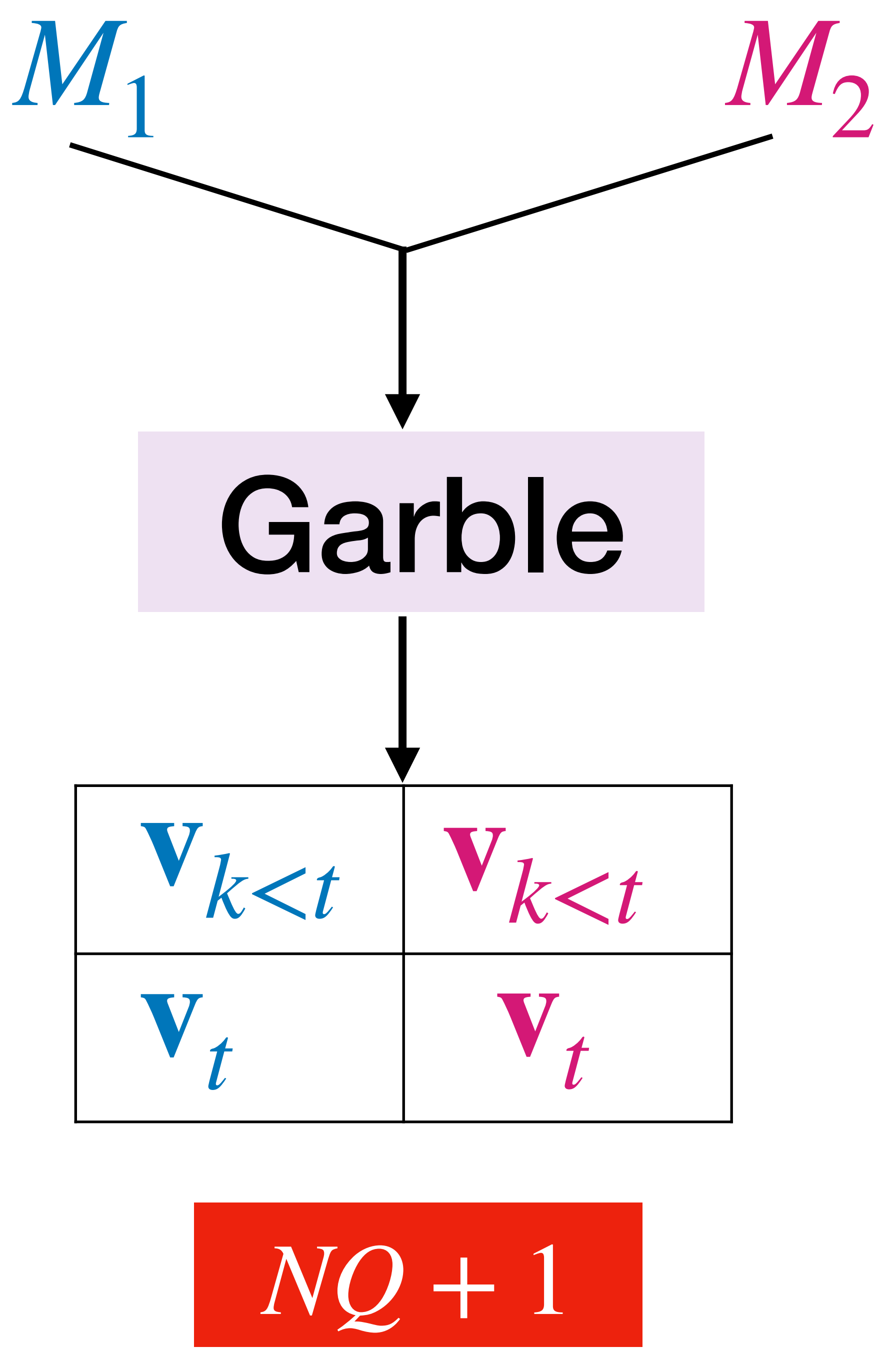
Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N} \ \mathbf{e}_{q_{\mathsf{acc}}} + \beta$

$$L_{q,j}(\mathbf{x}) = -\mathbf{r}_{j-1}[q] + (\mathbf{M}_{x_j}\mathbf{r}_j)[q] \ \boxed{= \mathbf{u}_j \cdot \mathbf{v}_q}$$
$$= \underbrace{(\mathbf{r}_{\mathbf{x}}[j-1], \mathbf{r}_{\mathbf{x}}[j])}_{\text{Known to the } \textbf{Encrypter}} \cdot \underbrace{(-\mathbf{r}_M[q], (\mathbf{M}_{x_j}\mathbf{r}_M)[q])}_{\text{Known to the } \textbf{Key Generator}}$$

$$\mathbf{v}_q = (\ -\mathbf{r}_M[q], \quad (\mathbf{M}_{x_j}\mathbf{r}_M)[q] \ )$$
$$\mathbf{u}_j = (\ \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N}\ \mathbf{e}_{q_{\text{acc}}} + \beta$

$$L_0(\mathbf{x}) = \beta + \mathbf{e}_1^\top \mathbf{r}_0 \qquad = \mathbf{u}_0 \cdot \mathbf{v}_0$$

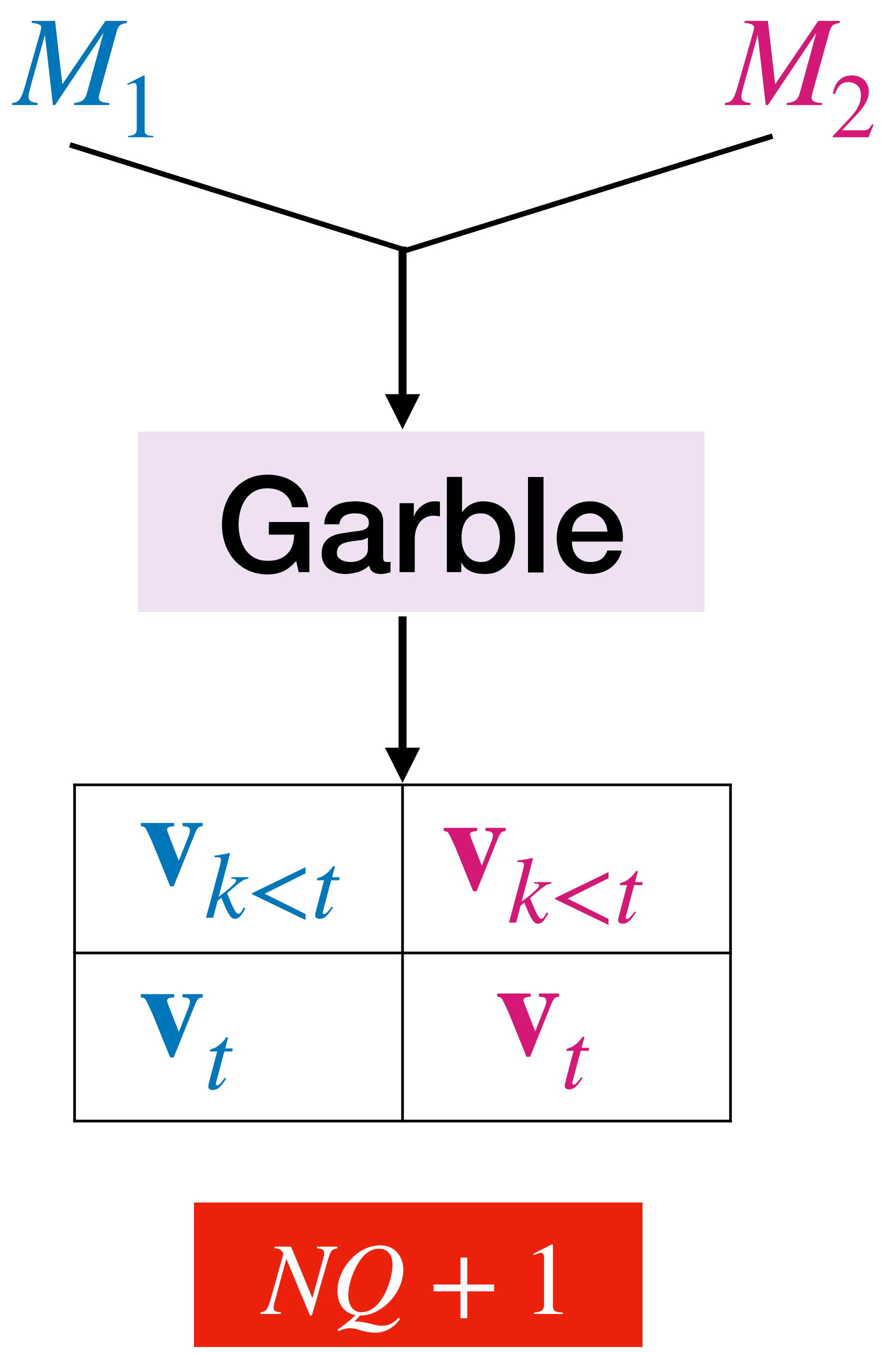
$$= (\mathbf{r}_{\mathbf{x}}[0],\ 1) \cdot (-\mathbf{r}_M[1],\ \beta)$$

$\mathbf{v}_0 = (\ -\mathbf{r}_M[1], \quad \beta\ )$ $\qquad$ $\mathbf{v}_q = (\ -\mathbf{r}_M[q], \quad (\mathbf{M}_{x_j}\mathbf{r}_M)[q]\ \ )$

$\mathbf{u}_0 = (\quad \mathbf{r}_{\mathbf{x}}[0], \qquad 1\ )$ $\qquad$ $\mathbf{u}_j = (\ \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \qquad)$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\, M_1, M_2 \,)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



$\mathbf{v}_{k<t}$	$\mathbf{v}_{k<t}$
$\mathbf{v}_t$	$\mathbf{v}_t$

AKGS for DFA ≡ AKGS for Matrix multiplication [LL20]

Garble: $z \cdot \mathbf{e}_1^\top \mathbf{M}_{x_1} \mathbf{M}_{x_2} \cdots \mathbf{M}_{x_N}\ \mathbf{e}_{q_{\mathsf{acc}}} + \beta$

$$L_{q,j}(z) = -\mathbf{r}_N[q] + z\ \mathbf{e}_{q_{\mathsf{acc}}}[q] \quad = \widetilde{\mathbf{u}}_j \cdot \widetilde{\mathbf{v}}_q$$

$$= (\mathbf{r}_{\mathbf{x}}[N], z) \cdot (-\mathbf{r}_M[q], \mathbf{e}_{q_{\mathsf{acc}}}[q])$$

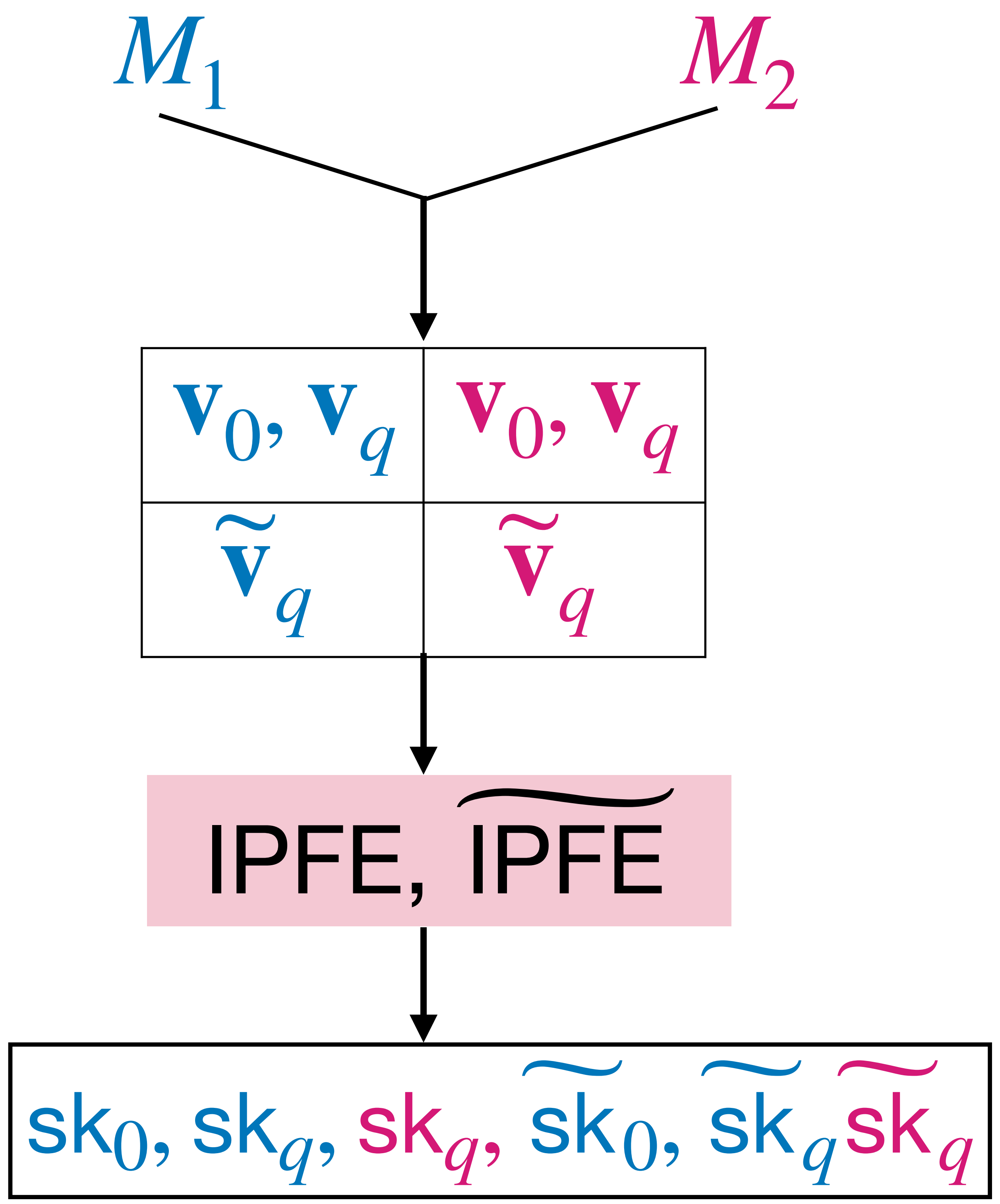
process of garbling is distributed between **key generator** and **encrypter**

$$\mathbf{v}_0 = (\ -\mathbf{r}_M[1], \quad \beta\ ) \qquad \mathbf{v}_q = (\ -\mathbf{r}_M[q], \quad (\mathbf{M}_{x_j}\mathbf{r}_M)[q]\ ) \qquad \widetilde{\mathbf{v}}_q = (\ -\mathbf{r}_M[q], \quad \mathbf{e}_{q_{\mathsf{acc}}}[q]\ )$$

$$\mathbf{u}_0 = (\ \mathbf{r}_{\mathbf{x}}[0], \quad 1\ ) \qquad \mathbf{u}_j = (\ \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_j = (\ \mathbf{r}_{\mathbf{x}}[N], \quad z\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

$\mathbf{v}_0, \mathbf{v}_q$	$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$	$\widetilde{\mathbf{v}}_q$

IPFE, $\widetilde{\text{IPFE}}$

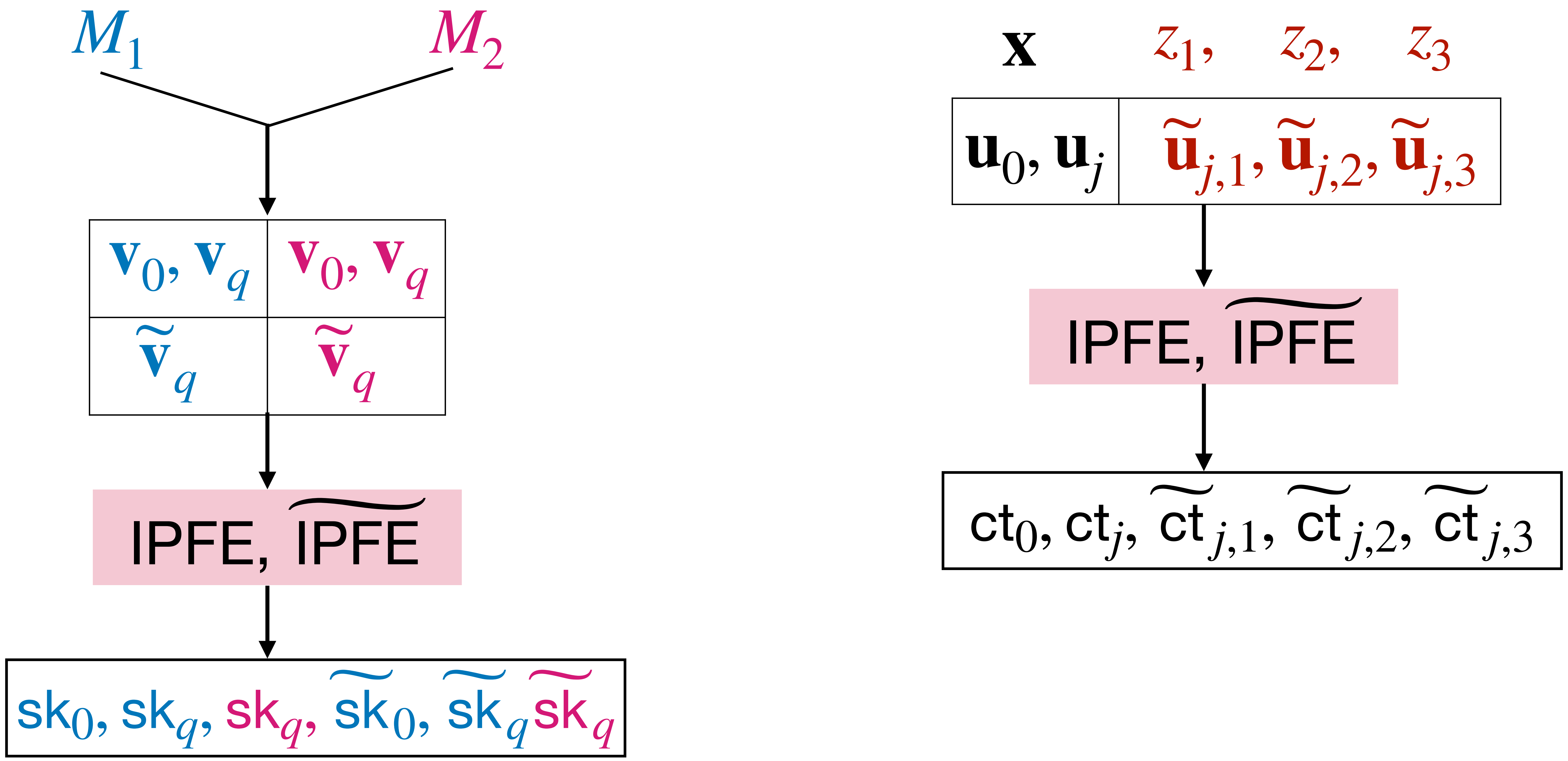
$\mathsf{sk}_0, \mathsf{sk}_q, \mathsf{sk}_q, \widetilde{\mathsf{sk}}_0, \widetilde{\mathsf{sk}}_q \widetilde{\mathsf{sk}}_q$

$$\mathbf{v}_{0,k} = (-\mathbf{r}_{M_k}[1], \quad \beta_k) \qquad \mathbf{v}_{q,k} = (-\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]) \qquad \widetilde{\mathbf{v}}_{q,k} = (-\mathbf{r}_{M_k}[q], \quad \mathbf{e}_{q_{\mathsf{acc}},k}[q])$$

$$\mathbf{u}_0 = (\mathbf{r}_{\mathbf{x}}[0], \quad 1) \qquad \mathbf{u}_j = (\mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j]) \qquad \widetilde{\mathbf{u}}_{j,i} = (\mathbf{r}_{\mathbf{x}}[N], \quad z_i)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

$\mathbf{v}_0, \mathbf{v}_q$	$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$	$\widetilde{\mathbf{v}}_q$

IPFE, $\widetilde{\text{IPFE}}$

$\text{sk}_0, \text{sk}_q, \text{sk}_q, \widetilde{\text{sk}}_0, \widetilde{\text{sk}}_q \widetilde{\text{sk}}_q$

$\mathbf{x}$ $z_1,$ $z_2,$ z_3

$\mathbf{u}_0, \mathbf{u}_j$	$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$

IPFE, $\widetilde{\text{IPFE}}$

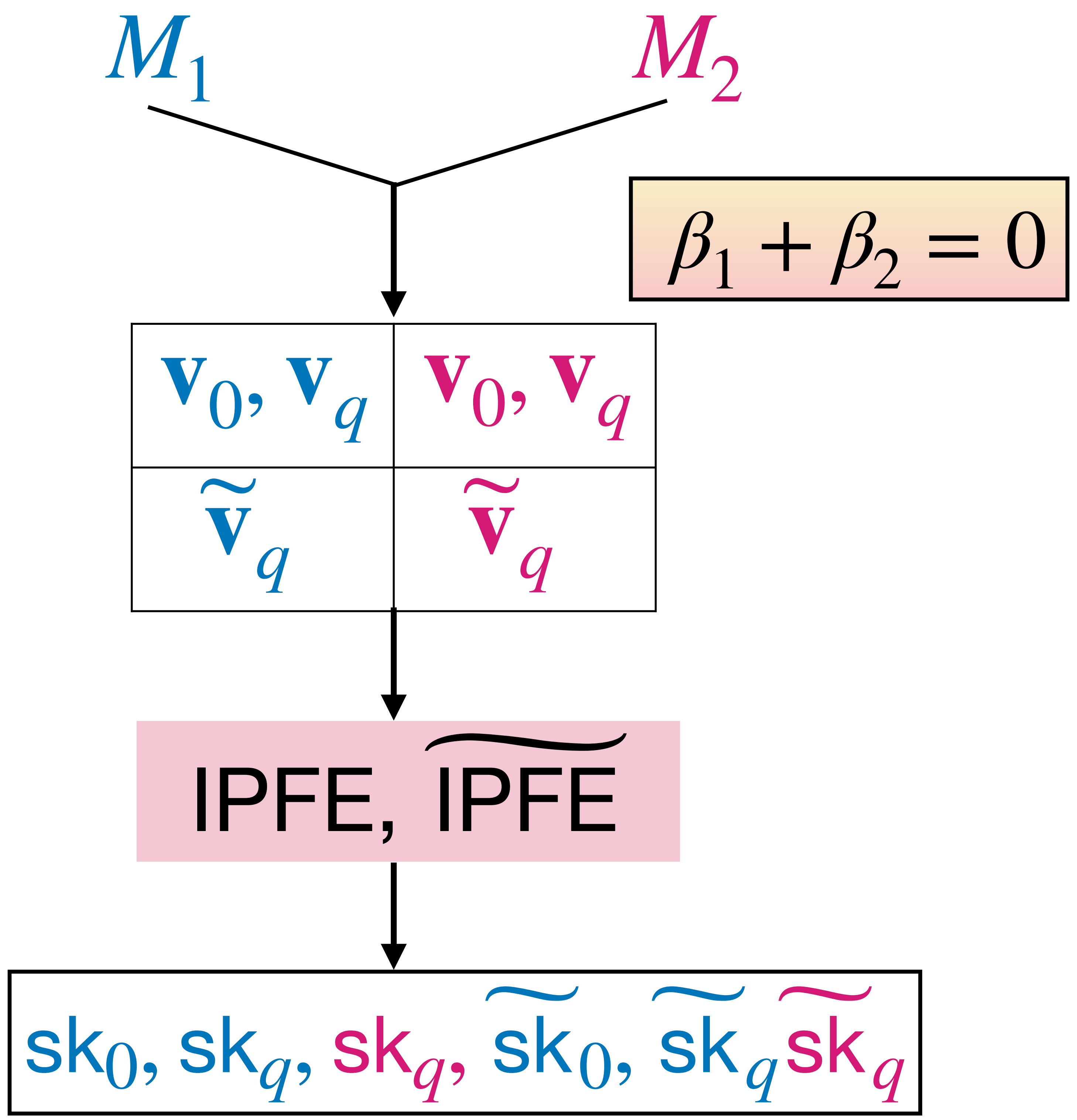
$\text{ct}_0, \text{ct}_j, \widetilde{\text{ct}}_{j,1}, \widetilde{\text{ct}}_{j,2}, \widetilde{\text{ct}}_{j,3}$

$\mathbf{v}_{0,k} = (\ -\mathbf{r}_{M_k}[1], \quad \beta_k\ )$ $\quad \mathbf{v}_{q,k} = (\ -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ )$ $\quad \widetilde{\mathbf{v}}_{q,k} = (\ -\mathbf{r}_{M_k}[q], \quad \mathbf{e}_{q_{\text{acc}},k}[q]\ )$

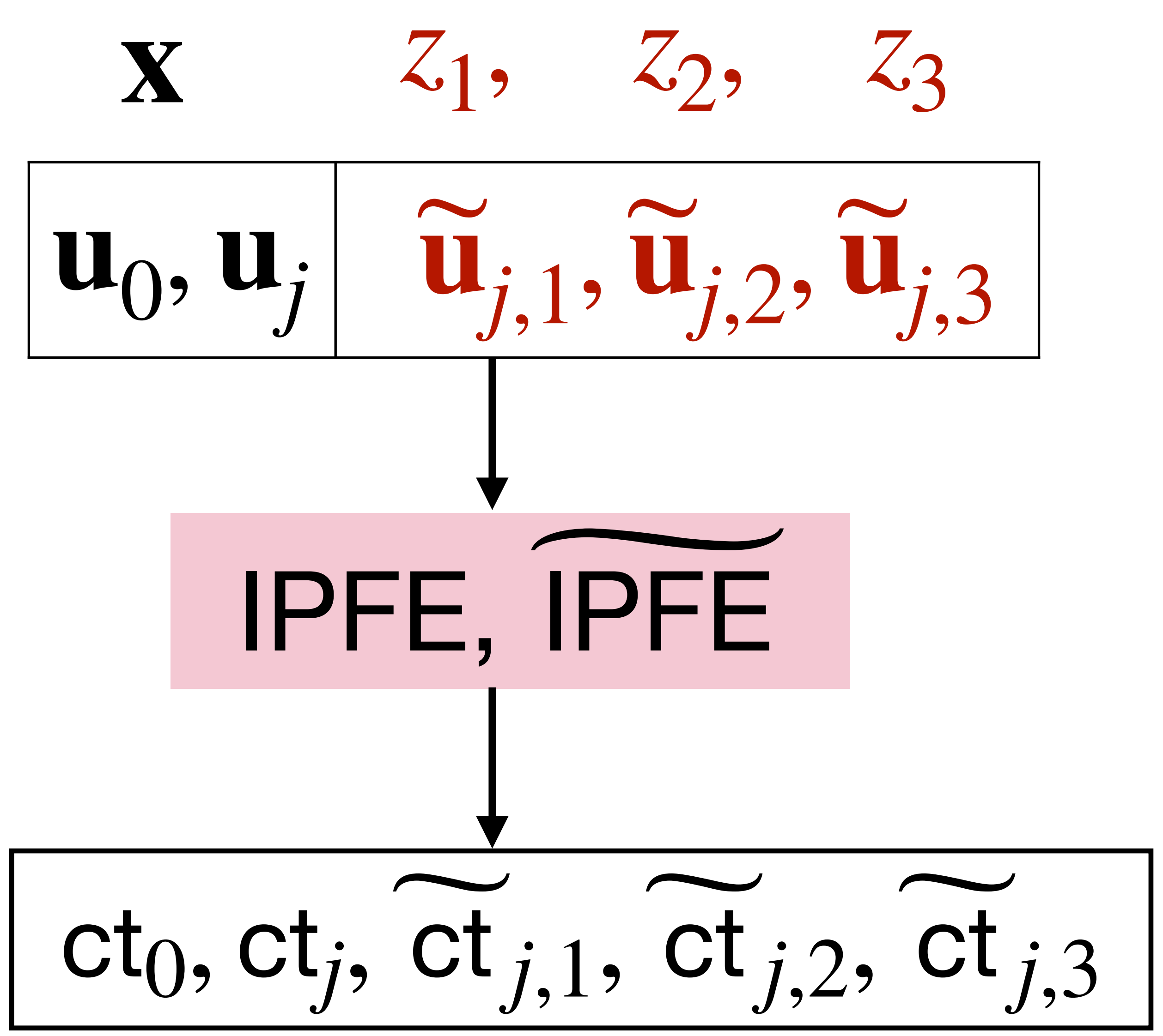
$\mathbf{u}_0 = (\ \mathbf{r}_\mathbf{x}[0], \quad 1\ )$ $\quad \mathbf{u}_j = (\ \mathbf{r}_\mathbf{x}[j-1], \quad \mathbf{r}_\mathbf{x}[j]\ )$ $\quad \widetilde{\mathbf{u}}_{j,i} = (\ \mathbf{r}_\mathbf{x}[N], \quad z_i\ )$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

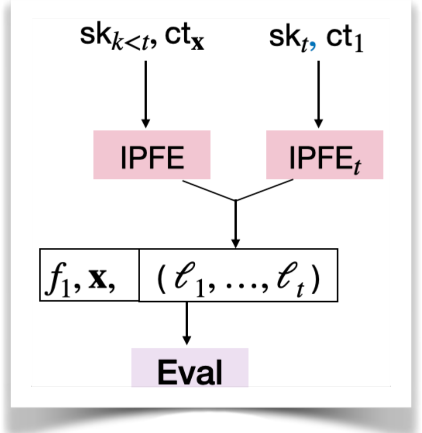
function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



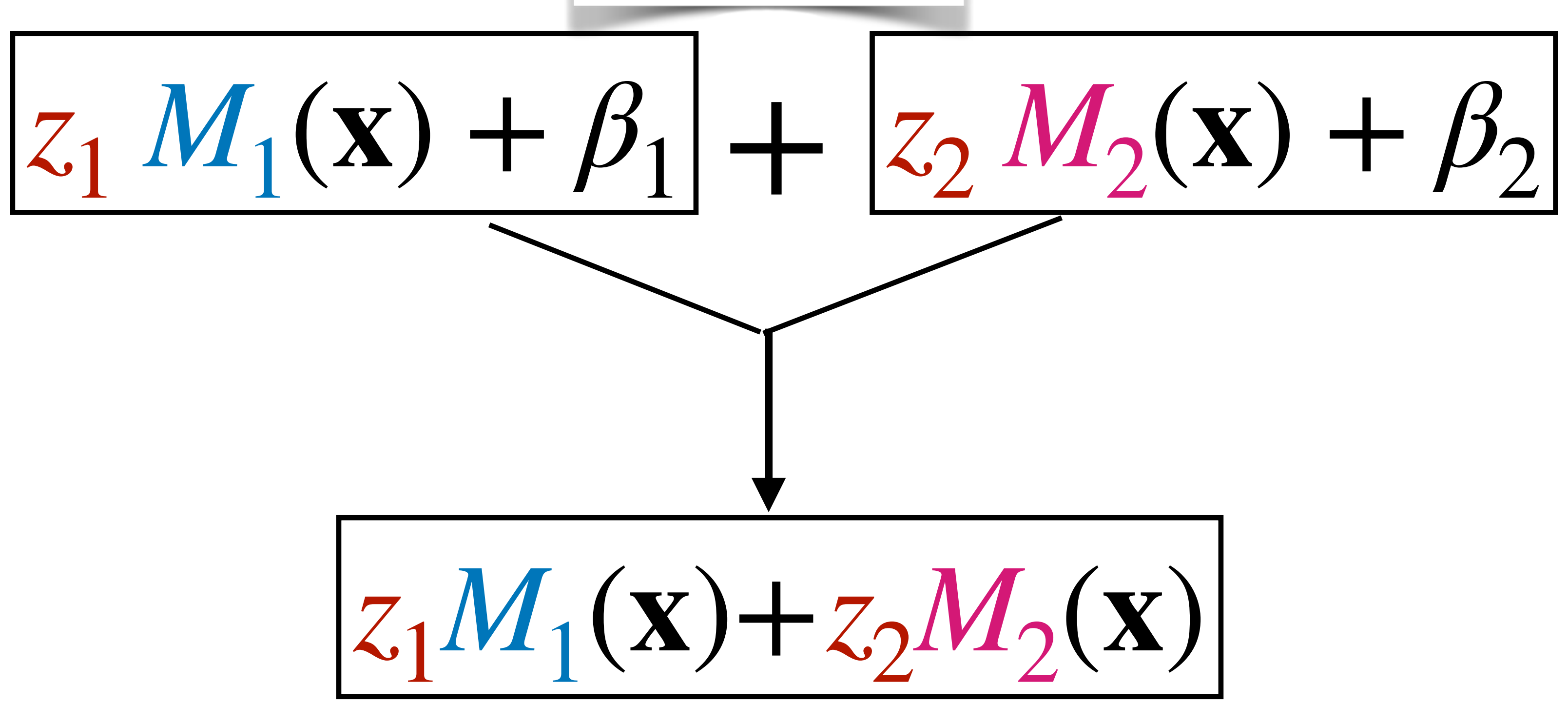
M_1
M_2
$\beta_1 + \beta_2 = 0$
$\mathbf{v}_0, \mathbf{v}_q$
$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$
$\widetilde{\mathbf{v}}_q$
IPFE, $\widetilde{\text{IPFE}}$
$\text{sk}_0, \text{sk}_q, \text{sk}_q, \widetilde{\text{sk}}_0, \widetilde{\text{sk}}_q \widetilde{\text{sk}}_q$



$\mathbf{x}$
$z_1,$
$z_2,$
z_3
$\mathbf{u}_0, \mathbf{u}_j$
$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$
IPFE, $\widetilde{\text{IPFE}}$
$\text{ct}_0, \text{ct}_j, \widetilde{\text{ct}}_{j,1}, \widetilde{\text{ct}}_{j,2}, \widetilde{\text{ct}}_{j,3}$



$\text{sk}_{k<t}, \text{ct}_{\mathbf{x}}$
sk_t, ct_1
IPFE
IPFE_t
$f_1, \mathbf{x},$
$(\ell_1, \ldots, \ell_t)$
Eval



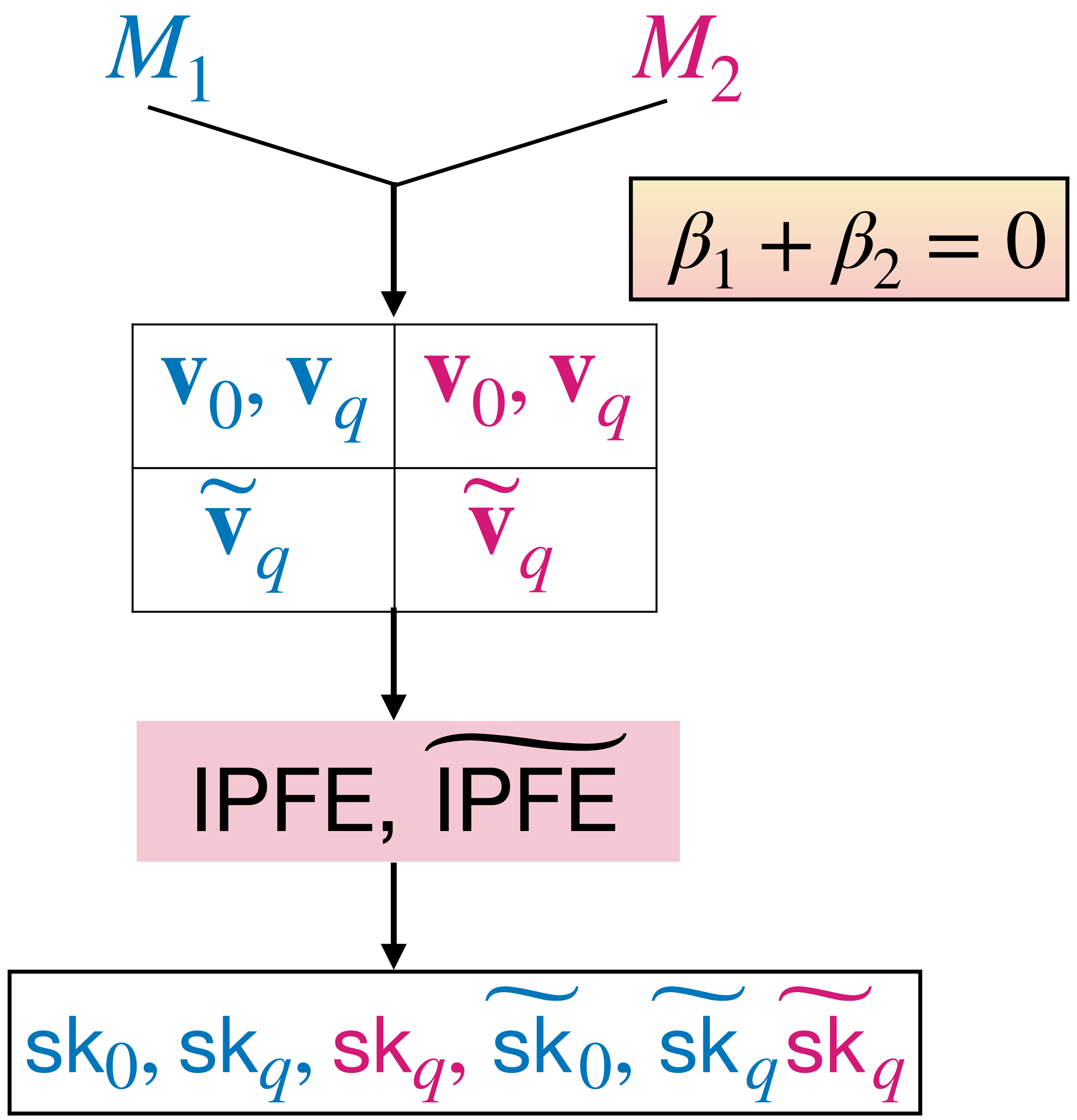
$z_1\ M_1(\mathbf{x}) + \beta_1$
$+$
$z_2\ M_2(\mathbf{x}) + \beta_2$
$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$

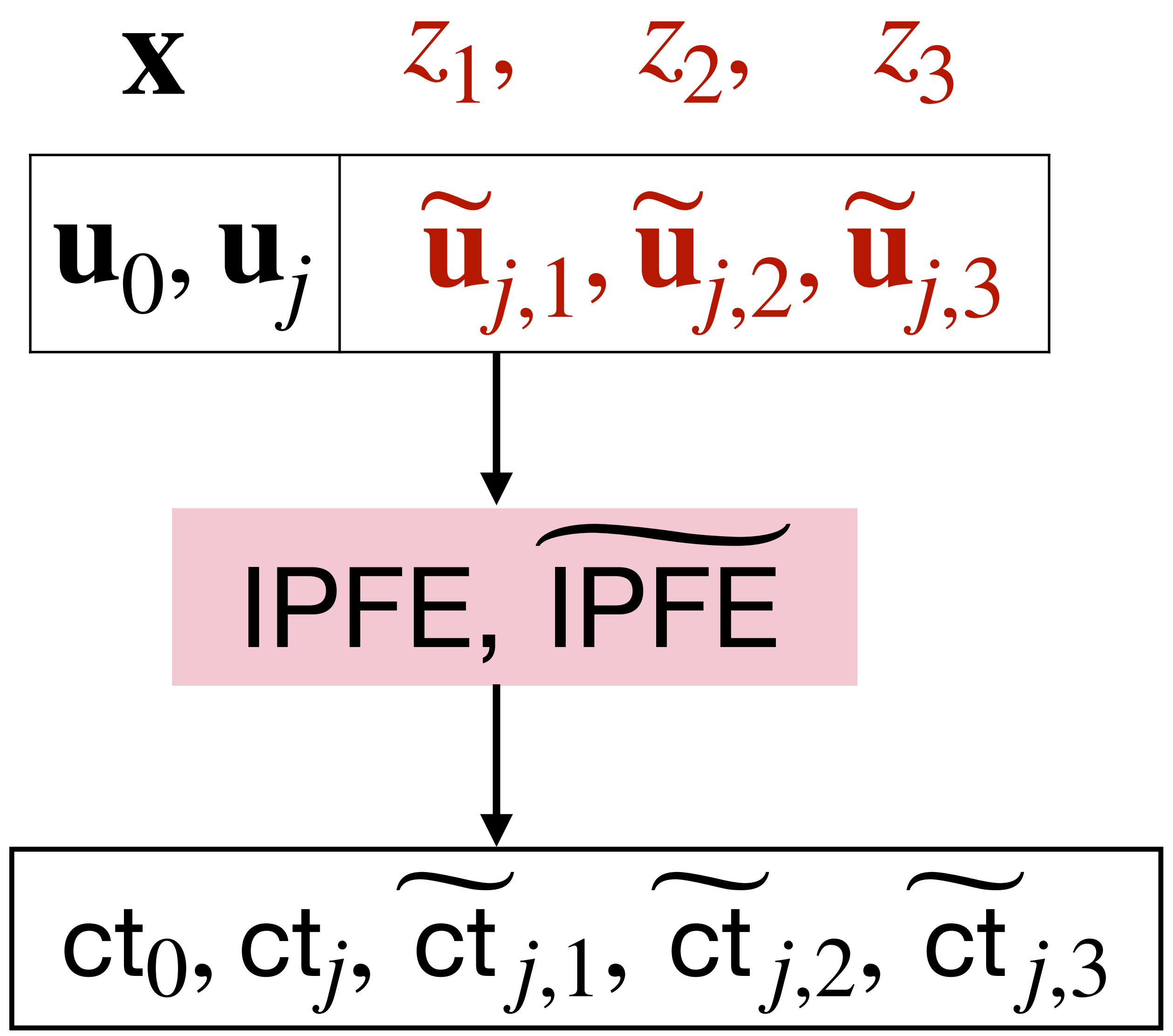
$$\mathbf{v}_{0,k} = (\ -\mathbf{r}_{M_k}[1],\quad \beta_k\ ) \qquad \mathbf{v}_{q,k} = (\ -\mathbf{r}_{M_k}[q],\quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ -\mathbf{r}_{M_k}[q],\quad \mathbf{e}_{q_{\text{acc}},k}[q]\ )$$

$$\mathbf{u}_0 = (\ \mathbf{r}_{\mathbf{x}}[0],\quad 1\ ) \qquad \mathbf{u}_j = (\ \mathbf{r}_{\mathbf{x}}[j-1],\quad \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \mathbf{r}_{\mathbf{x}}[N],\quad z_i\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\, M_1, M_2\,)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$


M_1
M_2
$\beta_1 + \beta_2 = 0$
$\mathbf{v}_0, \mathbf{v}_q$
$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$
$\widetilde{\mathbf{v}}_q$
IPFE, $\widetilde{\text{IPFE}}$
$\text{sk}_0, \text{sk}_q, \text{sk}_q, \widetilde{\text{sk}}_0, \widetilde{\text{sk}}_q \widetilde{\text{sk}}_q$


$\mathbf{x}$
$z_1,\quad z_2,\quad z_3$
$\mathbf{u}_0, \mathbf{u}_j$
$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$
IPFE, $\widetilde{\text{IPFE}}$
$\text{ct}_0, \text{ct}_j, \widetilde{\text{ct}}_{j,1}, \widetilde{\text{ct}}_{j,2}, \widetilde{\text{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

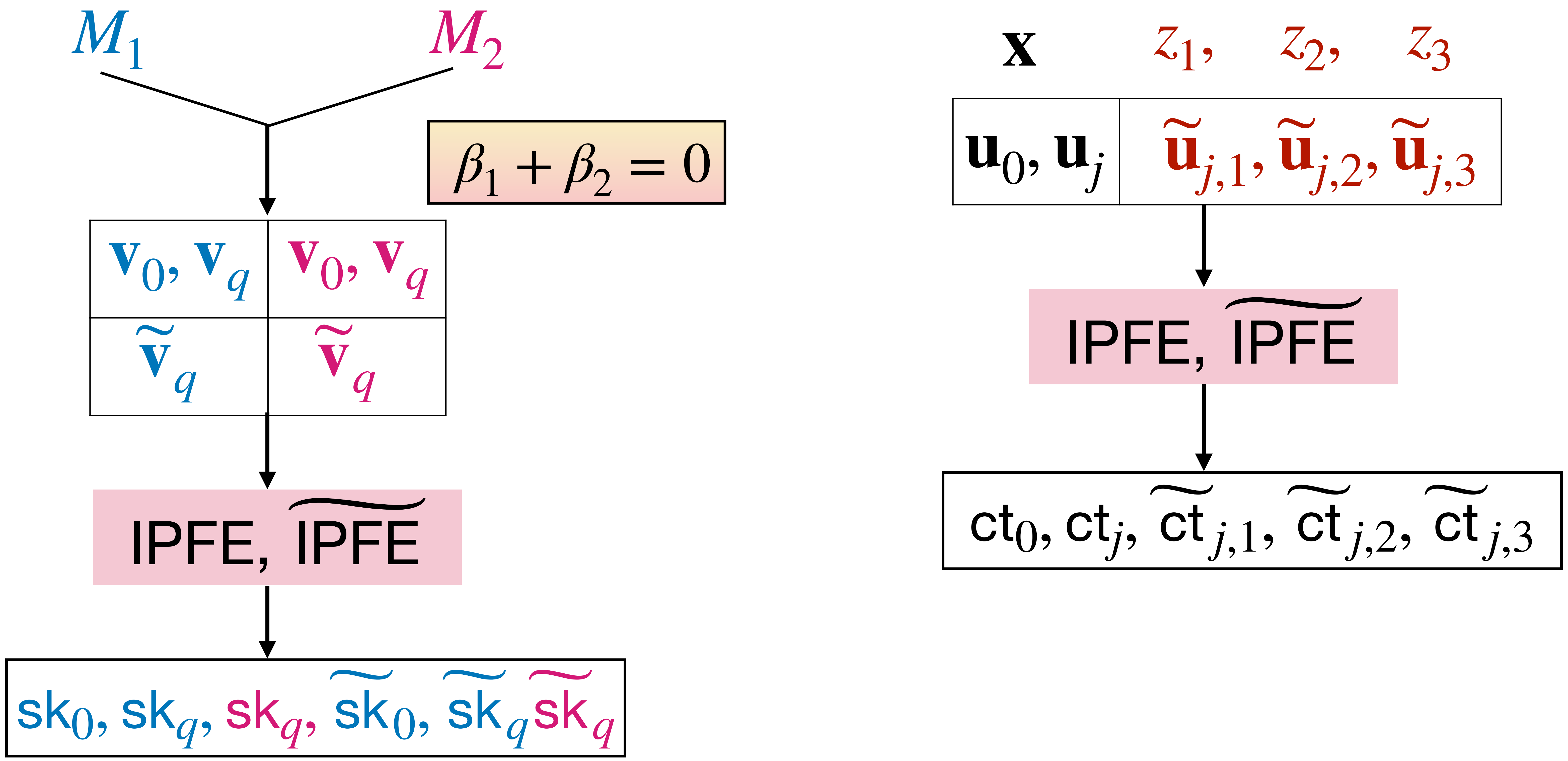
$$\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$$

$$\mathbf{v}_{0,k} = (\ -\mathbf{r}_{M_k}[1], \quad \beta_k\ ) \qquad \mathbf{v}_{q,k} = (\ -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ -\mathbf{r}_{M_k}[q], \quad \mathbf{e}_{q_{\mathsf{acc}},k}[q]\ )$$

$$\mathbf{u}_0 = (\ \mathbf{r}_{\mathbf{x}}[0], \quad 1\ ) \qquad \mathbf{u}_j = (\ \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \mathbf{r}_{\mathbf{x}}[N], \quad z_i\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1
M_2
$\beta_1 + \beta_2 = 0$
$\mathbf{v}_0, \mathbf{v}_q$
$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$
$\widetilde{\mathbf{v}}_q$
IPFE, $\widetilde{\text{IPFE}}$
$\text{sk}_0, \text{sk}_q, \text{sk}_q, \widetilde{\text{sk}}_0, \widetilde{\text{sk}}_q \widetilde{\text{sk}}_q$
$\mathbf{x}$
$z_1, \quad z_2, \quad z_3$
$\mathbf{u}_0, \mathbf{u}_j$
$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$
IPFE, $\widetilde{\text{IPFE}}$
$\text{ct}_0, \text{ct}_j, \widetilde{\text{ct}}_{j,1}, \widetilde{\text{ct}}_{j,2}, \widetilde{\text{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

$$\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$$

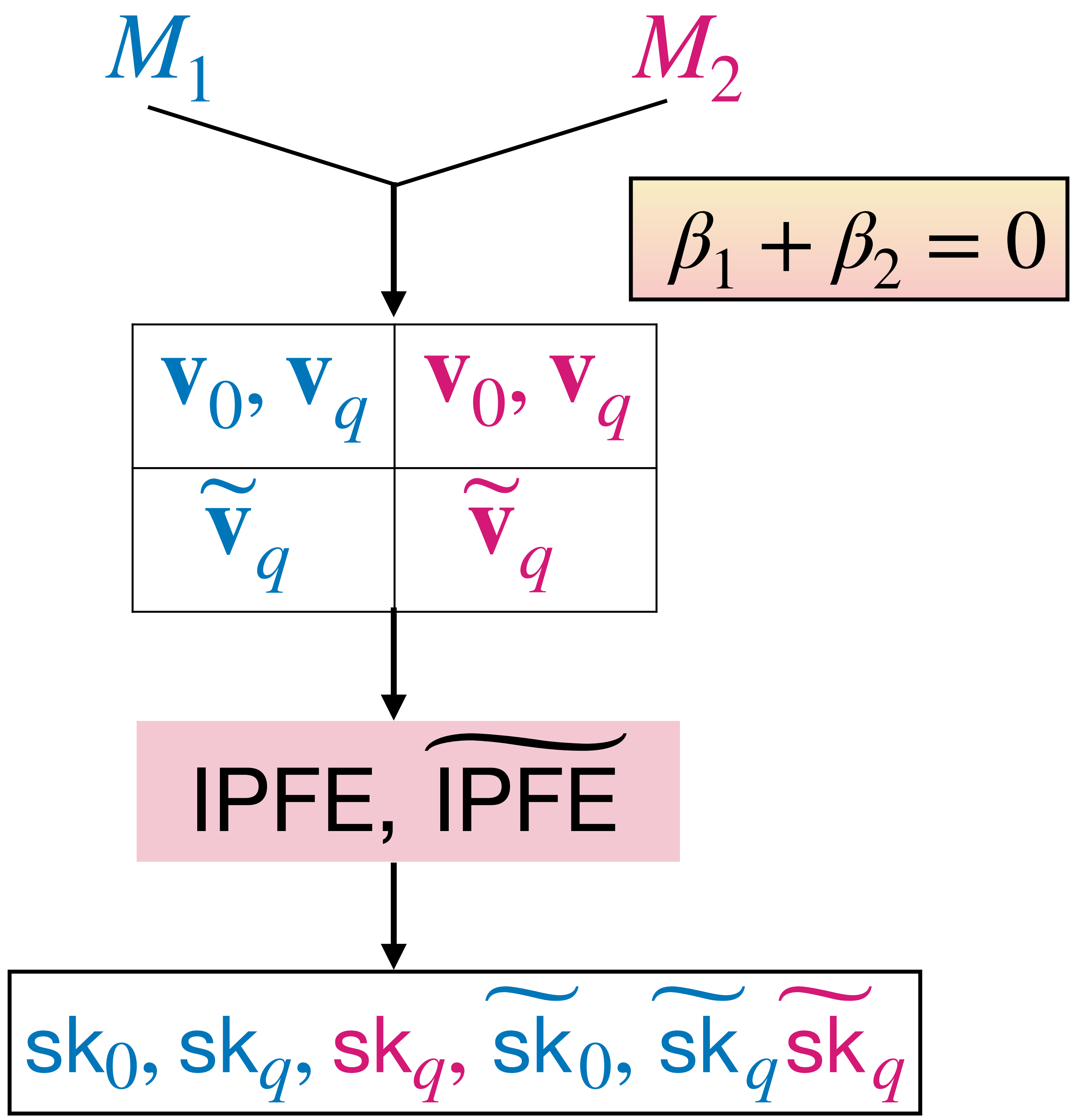
$$\mathbf{v}_{0,k} = (\ -\mathbf{r}_{M_k}[1], \quad \beta_k \ ) \qquad \mathbf{v}_{q,k} = (\ -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ -\mathbf{r}_{M_k}[q], \quad \mathbf{e}_{q_{\text{acc}},k}[q] \ )$$

$$\mathbf{u}_0 = (\ \mathbf{r}_{\mathbf{x}}[0], \quad 1 \ ) \qquad \mathbf{u}_j = (\ \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \mathbf{r}_{\mathbf{x}}[N], \quad z_i \ )$$

not enough space

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\, M_1, M_2 \,)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

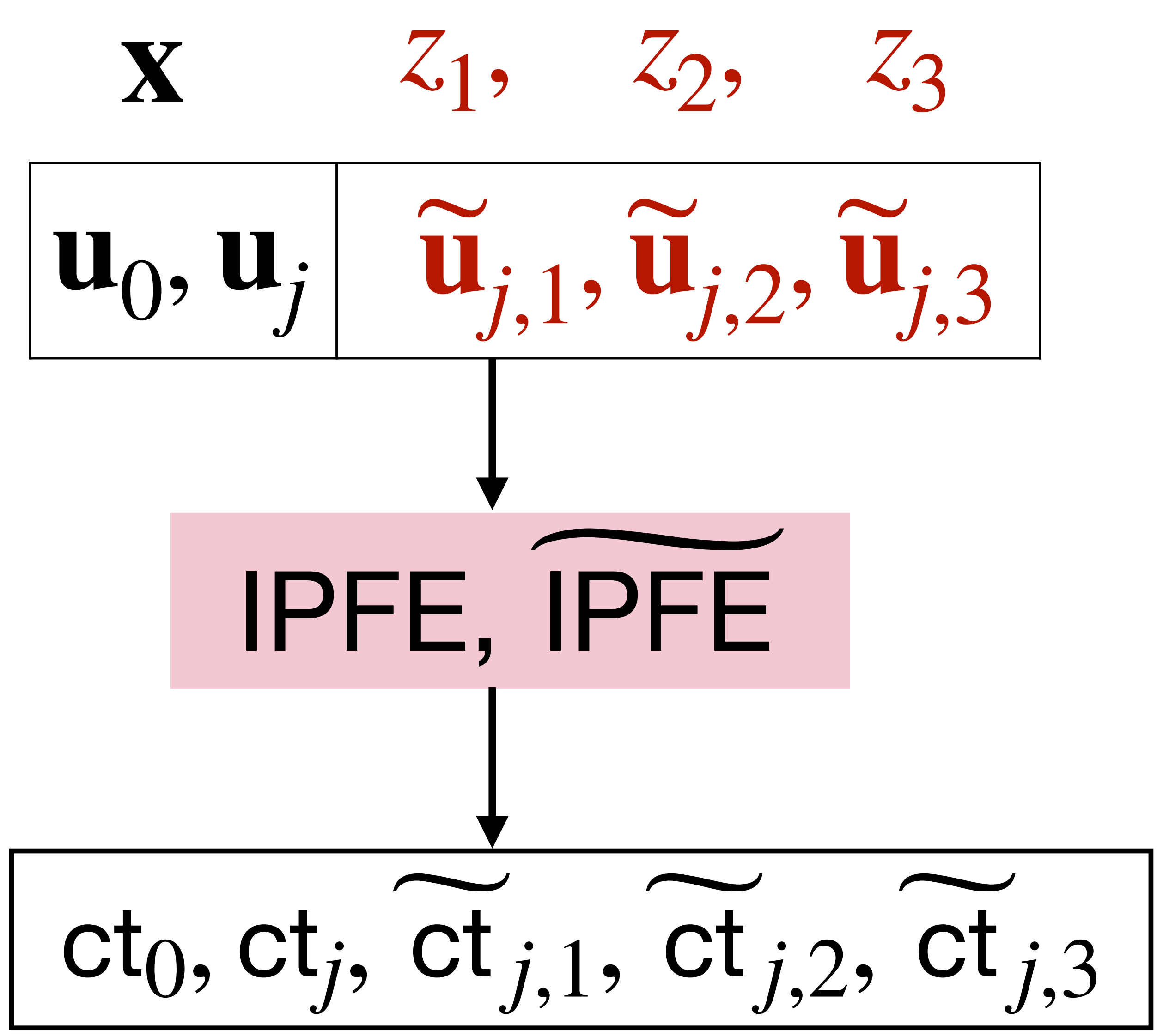
$\beta_1 + \beta_2 = 0$

$\mathbf{v}_0, \mathbf{v}_q$ $\mathbf{v}_0, \mathbf{v}_q$

$\widetilde{\mathbf{v}}_q$ $\widetilde{\mathbf{v}}_q$

IPFE, $\widetilde{\text{IPFE}}$

$\mathsf{sk}_0, \mathsf{sk}_q, \mathsf{sk}_q, \widetilde{\mathsf{sk}}_0, \widetilde{\mathsf{sk}}_q \widetilde{\mathsf{sk}}_q$



$\mathbf{x}$ $z_1,$ $z_2,$ z_3

$\mathbf{u}_0, \mathbf{u}_j$ $\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$

IPFE, $\widetilde{\text{IPFE}}$

$\mathsf{ct}_0, \mathsf{ct}_j, \widetilde{\mathsf{ct}}_{j,1}, \widetilde{\mathsf{ct}}_{j,2}, \widetilde{\mathsf{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

2. use **Marginal Randomness** to randomize all *other* labels

   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$$\mathbf{v}_{0,k} = (\; -\mathbf{r}_{M_k}[1], \quad \beta_k \;) \qquad \mathbf{v}_{q,k} = (\; -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \;) \qquad \widetilde{\mathbf{v}}_{q,k} = (\; -\mathbf{r}_{M_k}[q], \quad \mathbf{e}_{q_{\text{acc}},k}[q] \;)$$

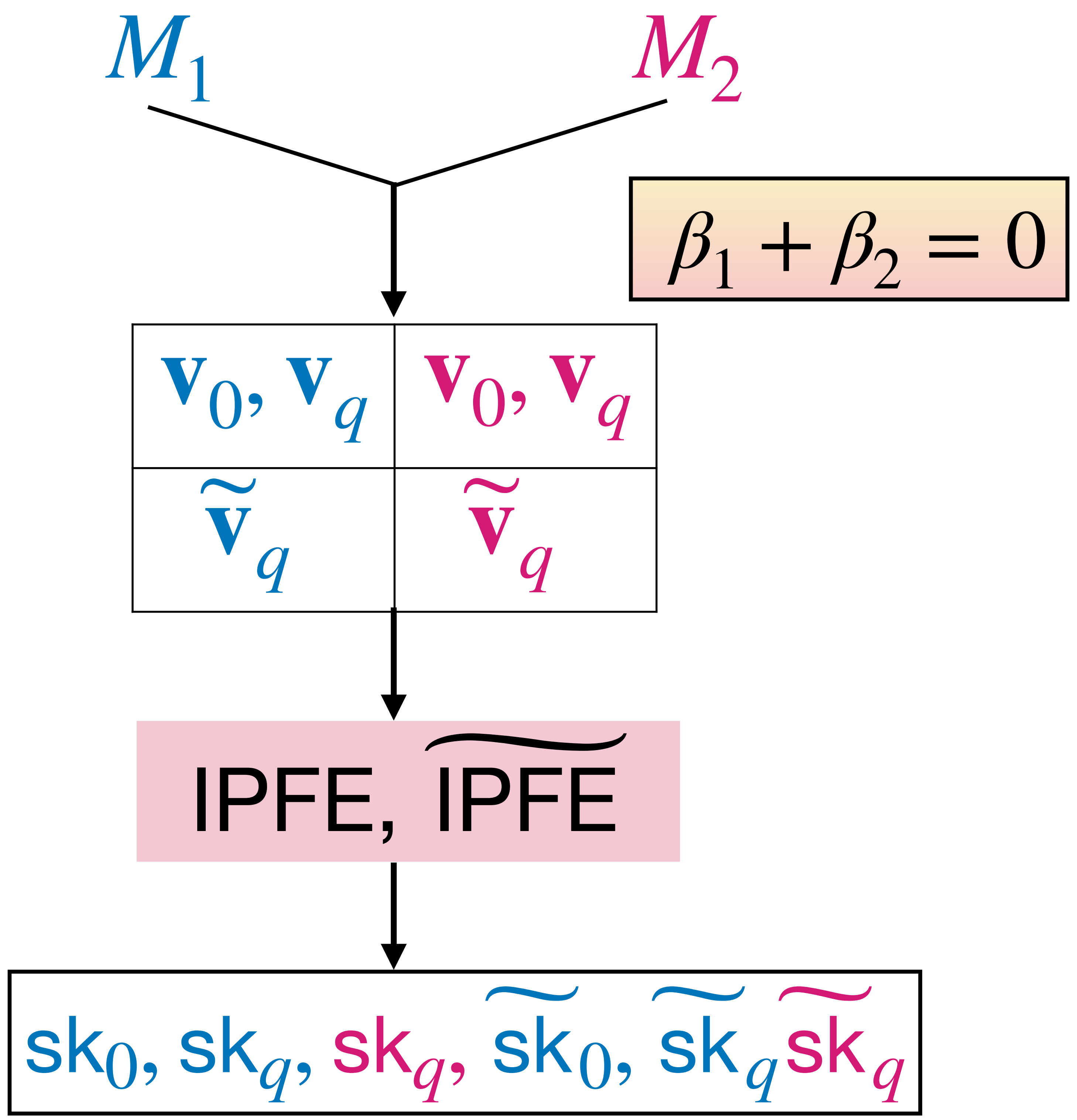
$$\mathbf{u}_0 = (\; \mathbf{r}_{\mathbf{x}}[0], \quad 1 \;) \qquad \mathbf{u}_j = (\; \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \;) \qquad \widetilde{\mathbf{u}}_{j,i} = (\; \mathbf{r}_{\mathbf{x}}[N], \quad z_i \;)$$

not enough space

not enough space

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1 M_2

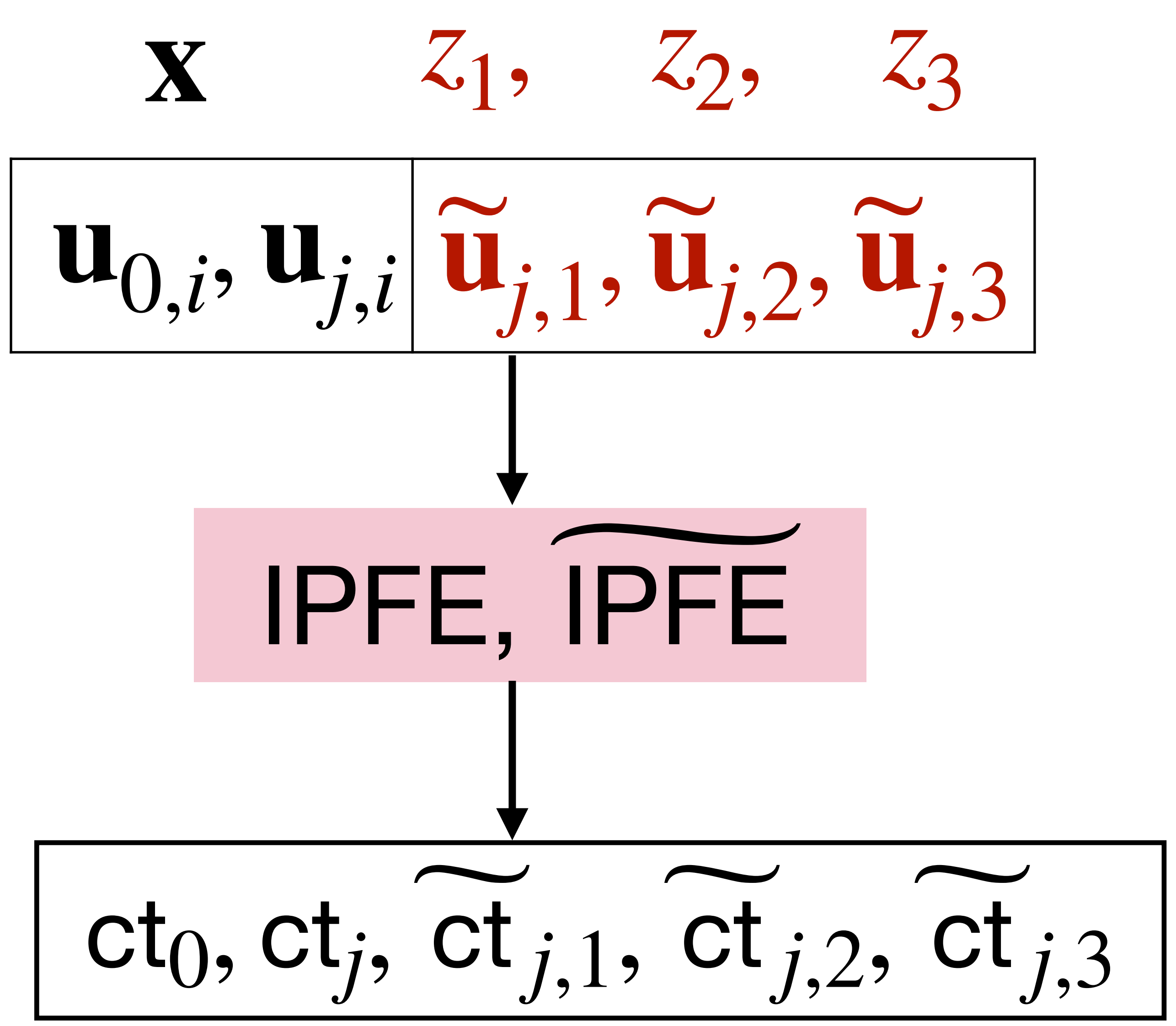
$\beta_1 + \beta_2 = 0$

$\mathbf{v}_0, \mathbf{v}_q$ | $\mathbf{v}_0, \mathbf{v}_q$

$\widetilde{\mathbf{v}}_q$ | $\widetilde{\mathbf{v}}_q$

IPFE, $\widetilde{\text{IPFE}}$

$\mathsf{sk}_0, \mathsf{sk}_q, \mathsf{sk}_q, \widetilde{\mathsf{sk}}_0, \widetilde{\mathsf{sk}}_q \widetilde{\mathsf{sk}}_q$



$\mathbf{x}$ $z_1,$ $z_2,$ z_3

$\mathbf{u}_{0,i}, \mathbf{u}_{j,i}$ | $\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$

IPFE, $\widetilde{\text{IPFE}}$

$\mathsf{ct}_0, \mathsf{ct}_j, \widetilde{\mathsf{ct}}_{j,1}, \widetilde{\mathsf{ct}}_{j,2}, \widetilde{\mathsf{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

2. use **Marginal Randomness** to randomize all *other* labels

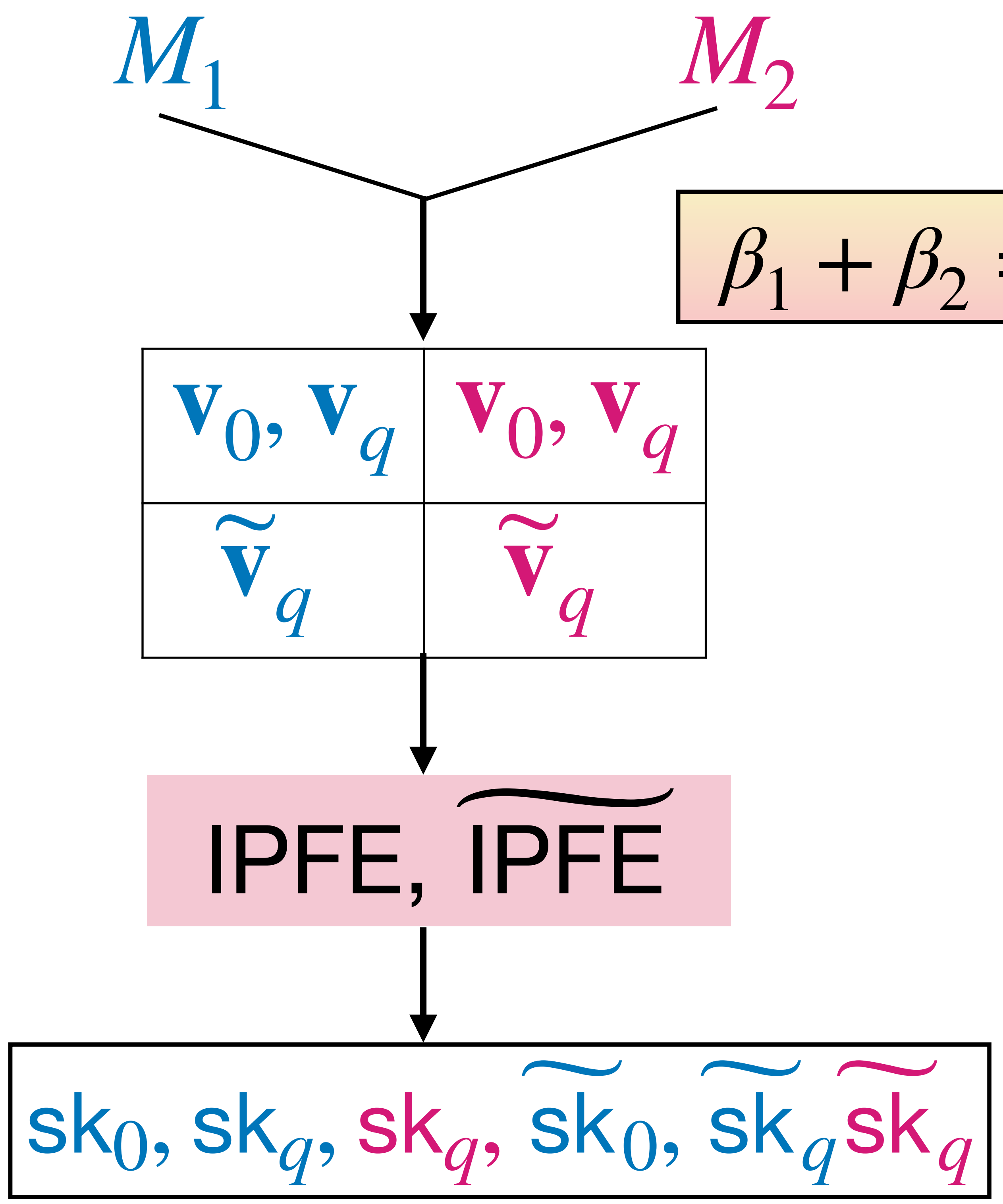
   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$$\mathbf{v}_{0,k} = (\ -\mathbf{r}_{M_k}[1], \quad \beta_k\ ) \qquad \mathbf{v}_{q,k} = (\ -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ -\mathbf{r}_{M_k}[q], \quad \mathbf{e}_{q_{\text{acc}},k}[q]\ )$$

$$\mathbf{u}_{0,i} = (\ \mathbf{r}_\mathbf{x}[0], \quad 1\ ) \qquad \mathbf{u}_{j,i} = (\ \mathbf{r}_\mathbf{x}[j-1], \quad \mathbf{r}_\mathbf{x}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \mathbf{r}_\mathbf{x}[N], \quad z_i\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



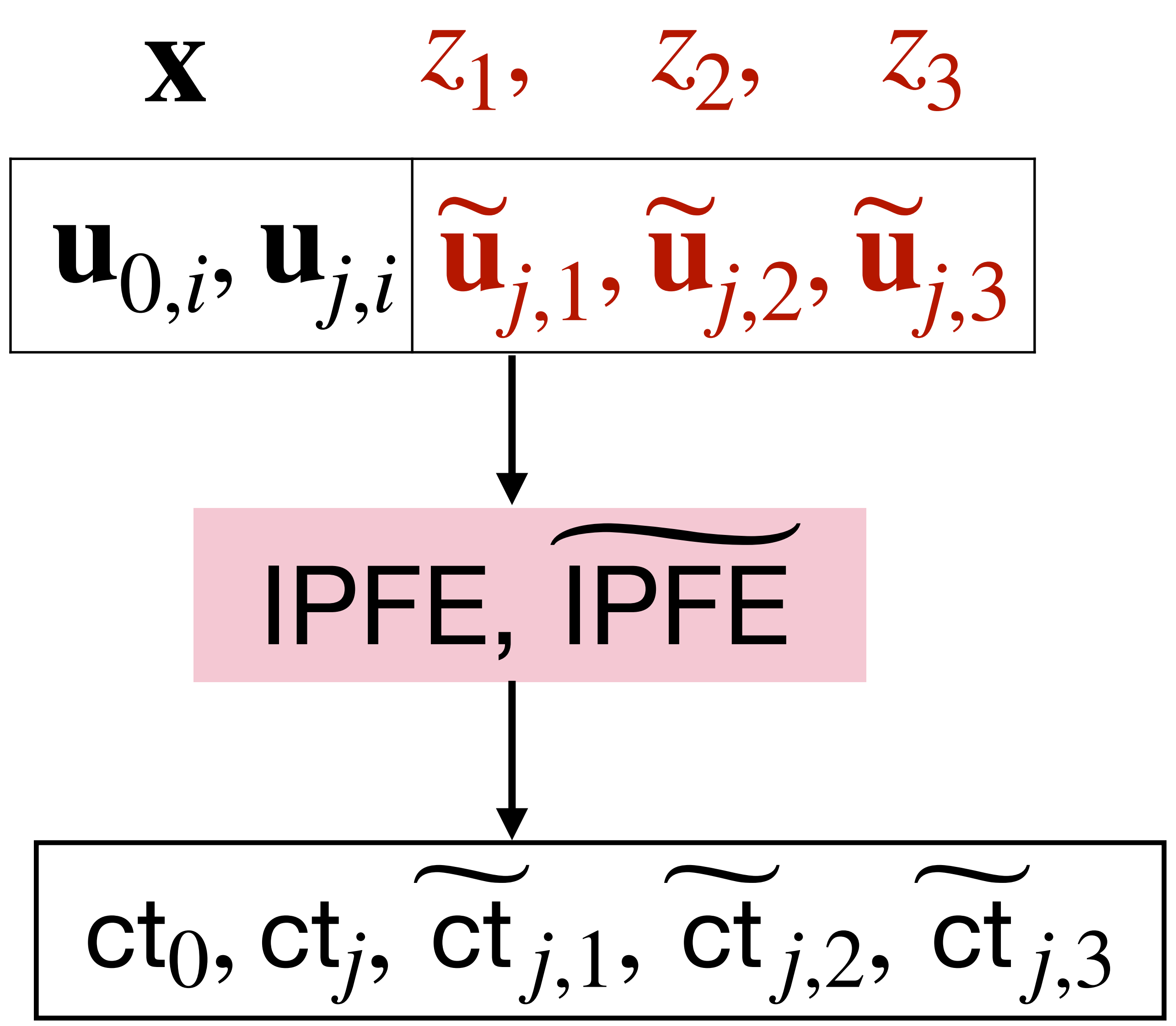
M_1 M_2

$\beta_1 + \beta_2 = 0$

$\mathbf{v}_0, \mathbf{v}_q$	$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$	$\widetilde{\mathbf{v}}_q$

IPFE, $\widetilde{\text{IPFE}}$

$\mathsf{sk}_0, \mathsf{sk}_q, \mathsf{sk}_q, \widetilde{\mathsf{sk}}_0, \widetilde{\mathsf{sk}}_q \widetilde{\mathsf{sk}}_q$



$\mathbf{x}$ $z_1,$ $z_2,$ z_3

$\mathbf{u}_{0,i}, \mathbf{u}_{j,i}$	$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$

IPFE, $\widetilde{\text{IPFE}}$

$\mathsf{ct}_0, \mathsf{ct}_j, \widetilde{\mathsf{ct}}_{j,1}, \widetilde{\mathsf{ct}}_{j,2}, \widetilde{\mathsf{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

2. use **Marginal Randomness** to randomize all *other* labels

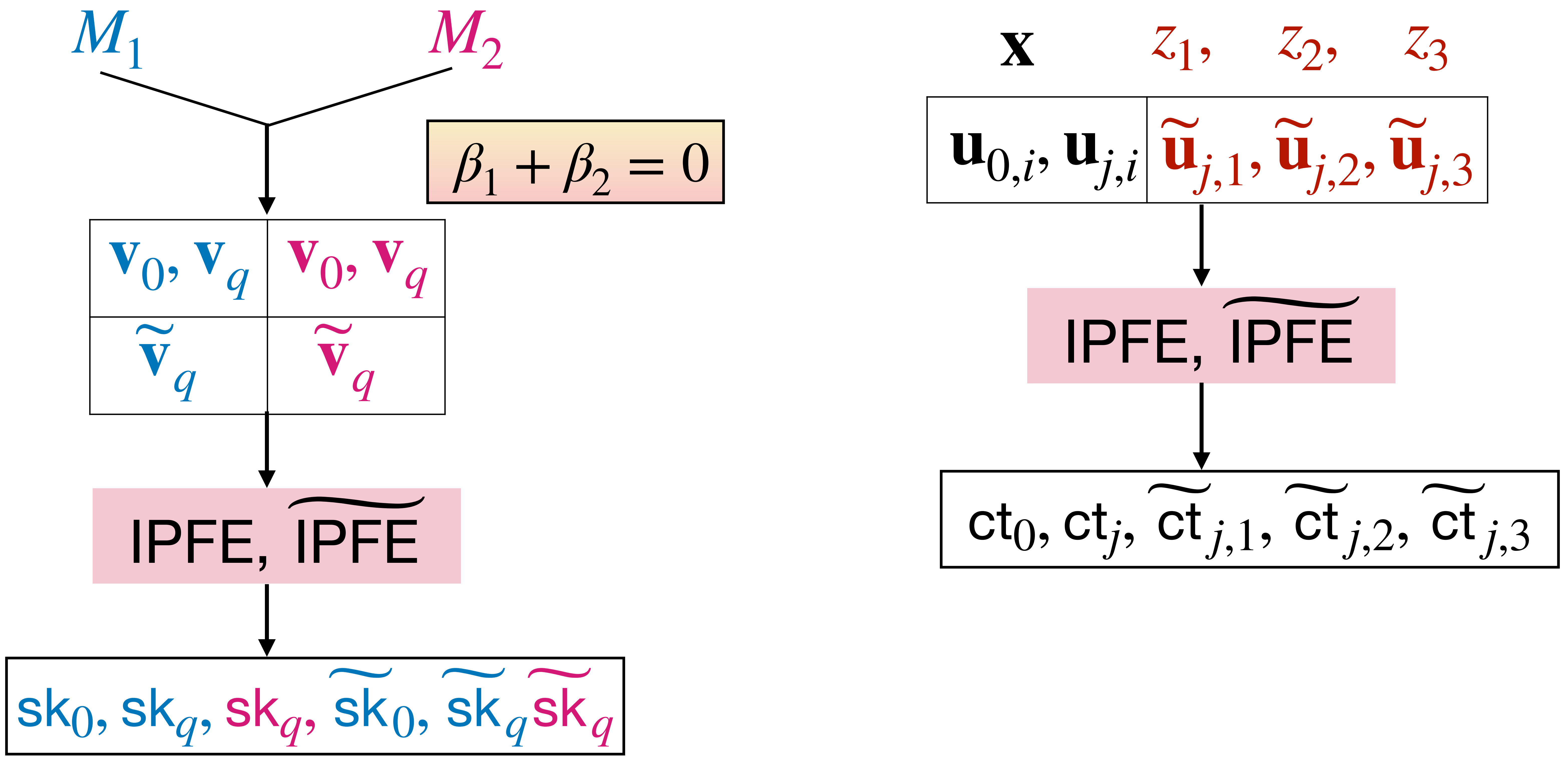
   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$$\mathbf{v}_{0,1} = (\ -\mathbf{r}_{M_1}[1], \quad \beta_1\ ) \qquad \mathbf{v}_{q,1} = (\ -\mathbf{r}_{M_1}[q], \quad (\mathbf{M}_{x_j,1}\mathbf{r}_{M_1})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,1} = (\ -\mathbf{r}_{M_1}[q], \quad \mathbf{e}_{q_{\mathsf{acc}},1}[q]\ )$$

$$\mathbf{u}_{0,2} = (\ \mathbf{r}_{\mathbf{x}}[0], \quad 1\ ) \qquad \mathbf{u}_{j,2} = (\ \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,2} = (\ \mathbf{r}_{\mathbf{x}}[N], \quad z_2\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})$



M_1
M_2
$\beta_1 + \beta_2 = 0$
$\mathbf{v}_0, \mathbf{v}_q$
$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$
$\widetilde{\mathbf{v}}_q$
IPFE, $\widetilde{\text{IPFE}}$
$\mathsf{sk}_0, \mathsf{sk}_q, \mathsf{sk}_q, \widetilde{\mathsf{sk}}_0, \widetilde{\mathsf{sk}}_q \widetilde{\mathsf{sk}}_q$
$\mathbf{x}$
$z_1,$
$z_2,$
z_3
$\mathbf{u}_{0,i}, \mathbf{u}_{j,i}$
$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$
IPFE, $\widetilde{\text{IPFE}}$
$\mathsf{ct}_0, \mathsf{ct}_j, \widetilde{\mathsf{ct}}_{j,1}, \widetilde{\mathsf{ct}}_{j,2}, \widetilde{\mathsf{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

2. use **Marginal Randomness** to randomize all *other* labels

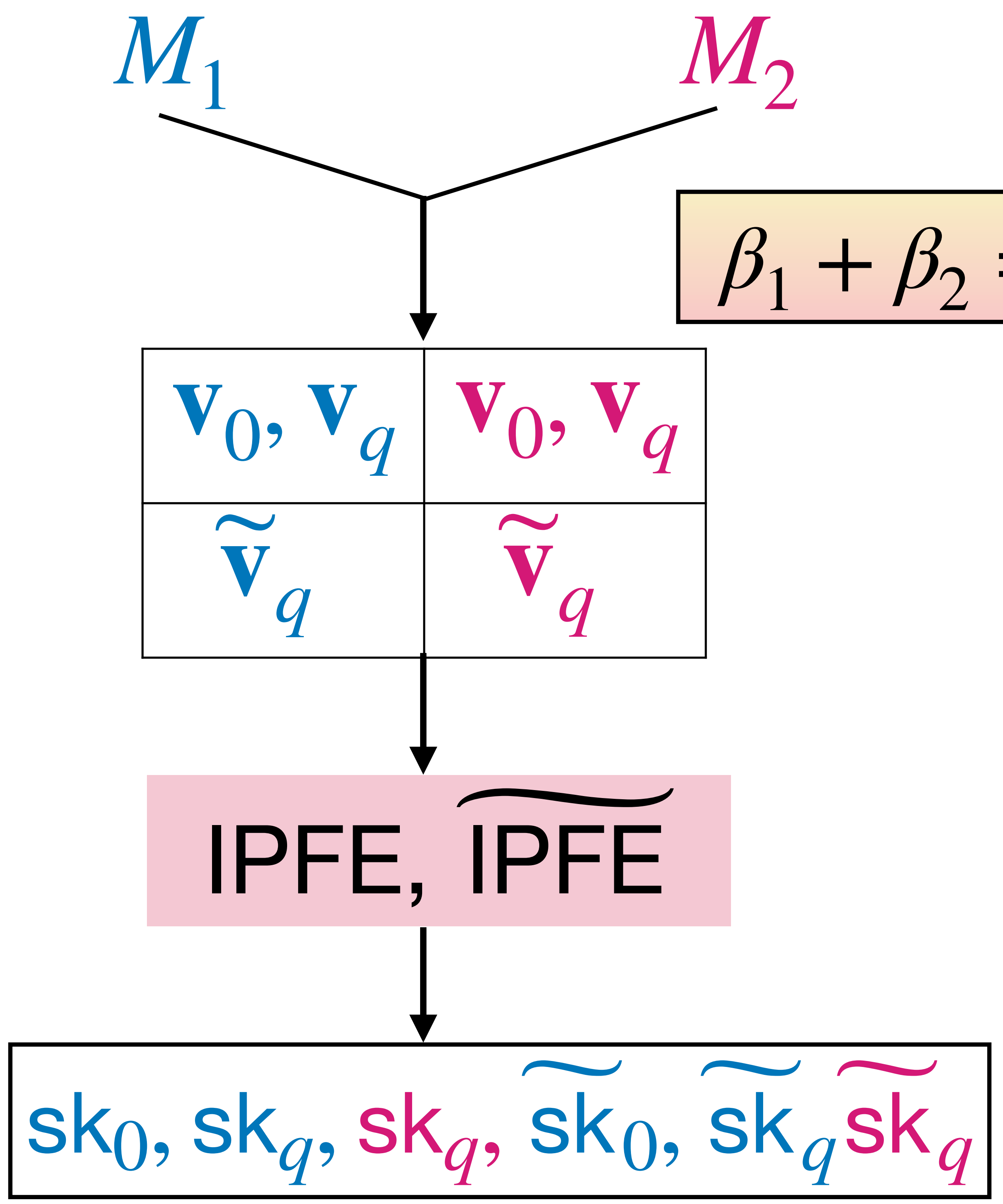
   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

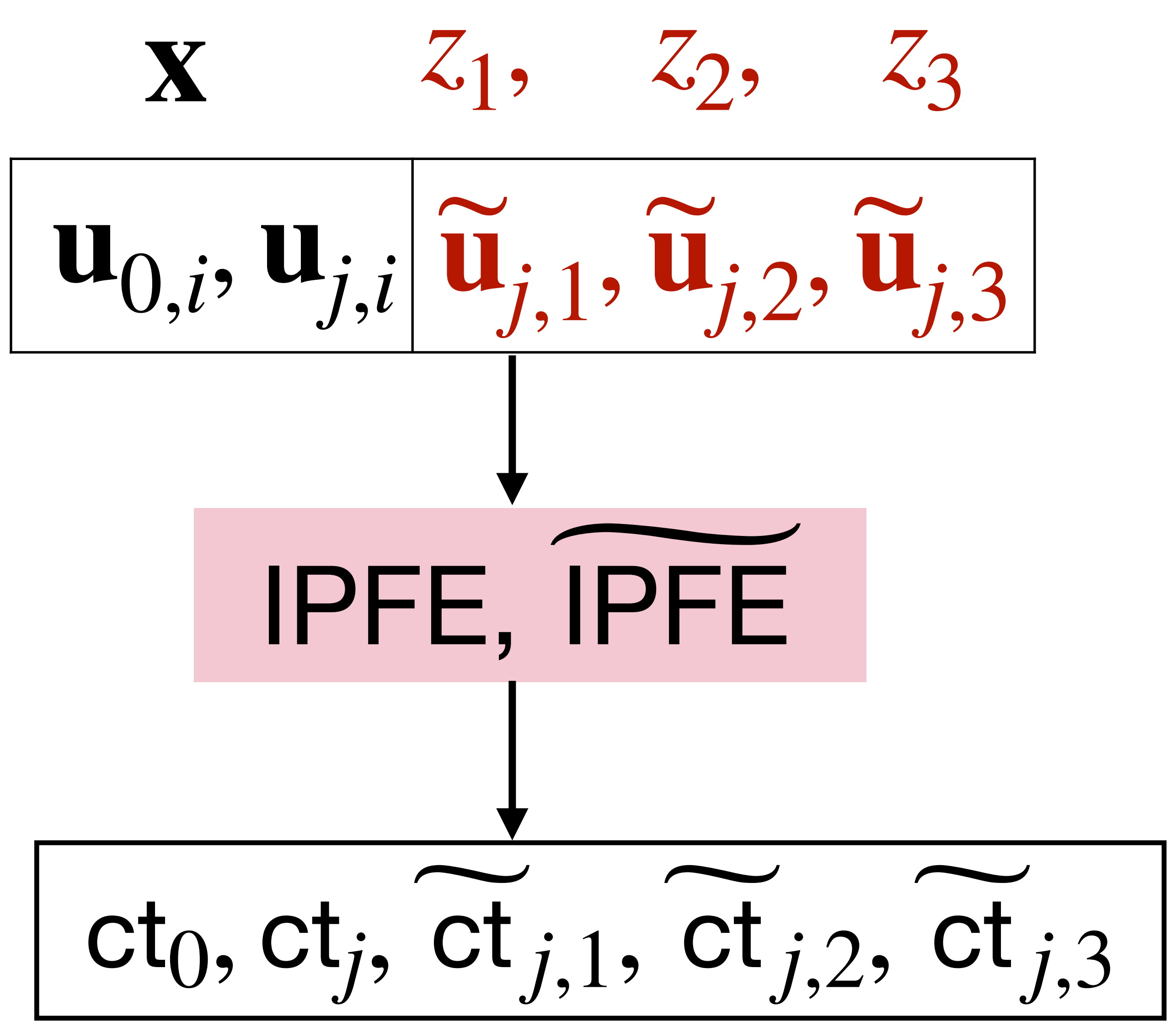
$$\mathbf{v}_{0,2} = (-\mathbf{r}_{M_2}[1], \quad \beta_2) \qquad \mathbf{v}_{q,2} = (-\mathbf{r}_{M_2}[q], \quad (\mathbf{M}_{x_j,2}\mathbf{r}_{M_2})[q]) \qquad \widetilde{\mathbf{v}}_{q,2} = (-\mathbf{r}_{M_2}[q], \quad \mathbf{e}_{q_{\mathsf{acc}},2}[q])$$

$$\mathbf{u}_{0,1} = (\mathbf{r}_\mathbf{x}[0], \quad 1) \qquad \mathbf{u}_{j,1} = (\mathbf{r}_\mathbf{x}[j-1], \quad \mathbf{r}_\mathbf{x}[j]) \qquad \widetilde{\mathbf{u}}_{j,1} = (\mathbf{r}_\mathbf{x}[N], \quad z_1)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_2(\mathbf{x}) + z_2 M_1(\mathbf{x})}$


M_1
M_2
$\beta_1 + \beta_2 = 0$
$\mathbf{v}_0, \mathbf{v}_q$
$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$
$\widetilde{\mathbf{v}}_q$
IPFE, $\widetilde{\text{IPFE}}$
$\mathsf{sk}_0, \mathsf{sk}_q, \mathsf{sk}_q, \widetilde{\mathsf{sk}}_0, \widetilde{\mathsf{sk}}_q \widetilde{\mathsf{sk}}_q$


$\mathbf{x}$
$z_1,$
$z_2,$
z_3
$\mathbf{u}_{0,i}, \mathbf{u}_{j,i}$
$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$
IPFE, $\widetilde{\text{IPFE}}$
$\mathsf{ct}_0, \mathsf{ct}_j, \widetilde{\mathsf{ct}}_{j,1}, \widetilde{\mathsf{ct}}_{j,2}, \widetilde{\mathsf{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \mathsf{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

2. use **Marginal Randomness** to randomize all *other* labels

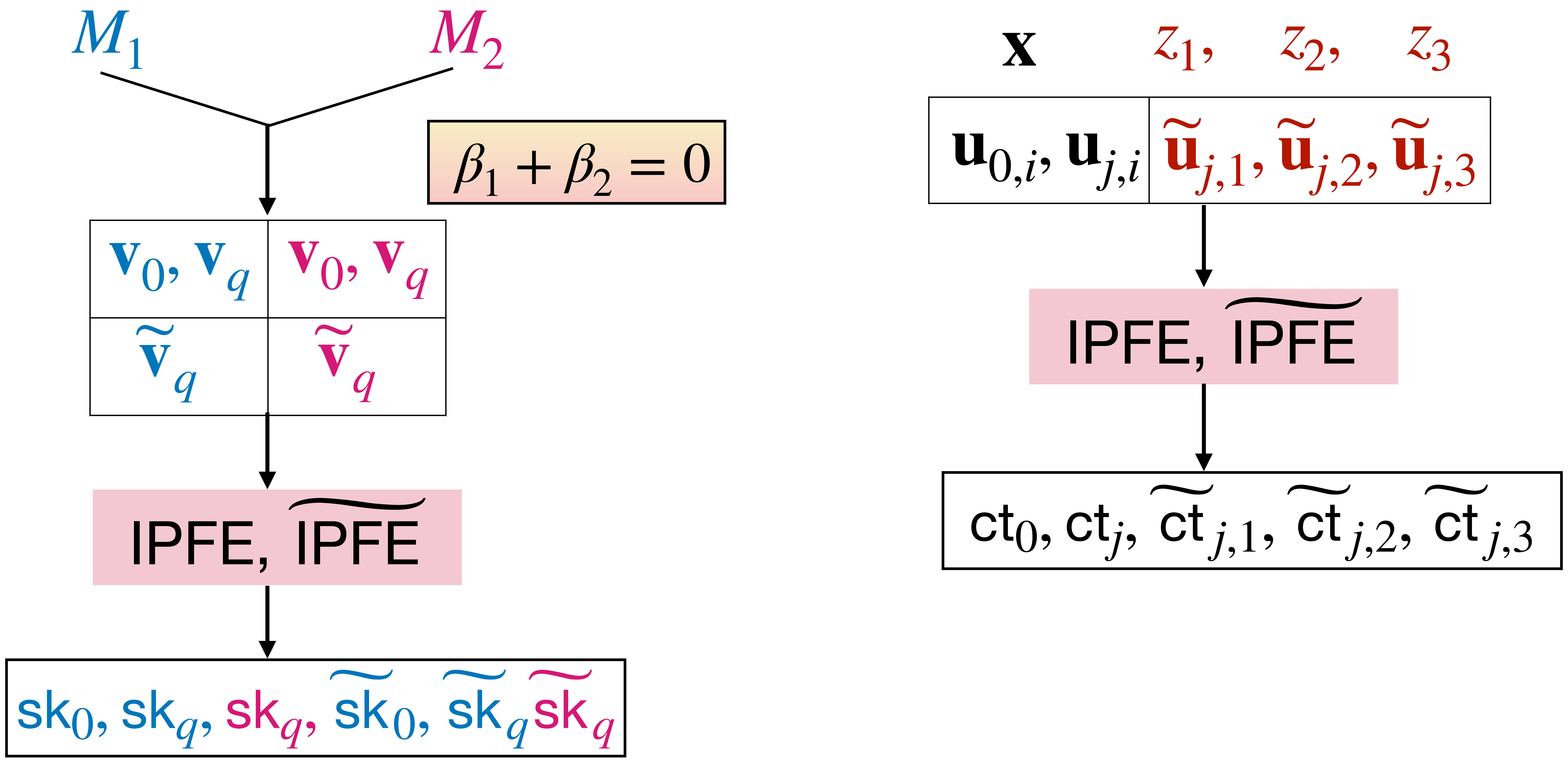
   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$$\mathbf{v}_{0,2} = (\; -\mathbf{r}_{M_2}[1], \quad \beta_2 \;) \qquad \mathbf{v}_{q,2} = (\; -\mathbf{r}_{M_2}[q], \quad (\mathbf{M}_{x_j,2}\mathbf{r}_{M_2})[q] \;) \qquad \widetilde{\mathbf{v}}_{q,2} = (\; -\mathbf{r}_{M_2}[q], \quad \mathbf{e}_{q_{\mathsf{acc}},2}[q] \;)$$

$$\mathbf{u}_{0,1} = (\; \mathbf{r}_{\mathbf{x}}[0], \quad 1 \;) \qquad \mathbf{u}_{j,1} = (\; \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \;) \qquad \widetilde{\mathbf{u}}_{j,1} = (\; \mathbf{r}_{\mathbf{x}}[N], \quad z_1 \;)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_2(\mathbf{x}) + z_2 M_1(\mathbf{x})}$


M_1
M_2
$\beta_1 + \beta_2 = 0$
$\mathbf{v}_0, \mathbf{v}_q$
$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$
$\widetilde{\mathbf{v}}_q$
IPFE, $\widetilde{\text{IPFE}}$
$\text{sk}_0, \text{sk}_q, \text{sk}_q, \widetilde{\text{sk}}_0, \widetilde{\text{sk}}_q \widetilde{\text{sk}}_q$
$\mathbf{x}$
$z_1,$
$z_2,$
z_3
$\mathbf{u}_{0,i}, \mathbf{u}_{j,i}$
$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$
IPFE, $\widetilde{\text{IPFE}}$
$\text{ct}_0, \text{ct}_j, \widetilde{\text{ct}}_{j,1}, \widetilde{\text{ct}}_{j,2}, \widetilde{\text{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

2. use **Marginal Randomness** to randomize all *other* labels

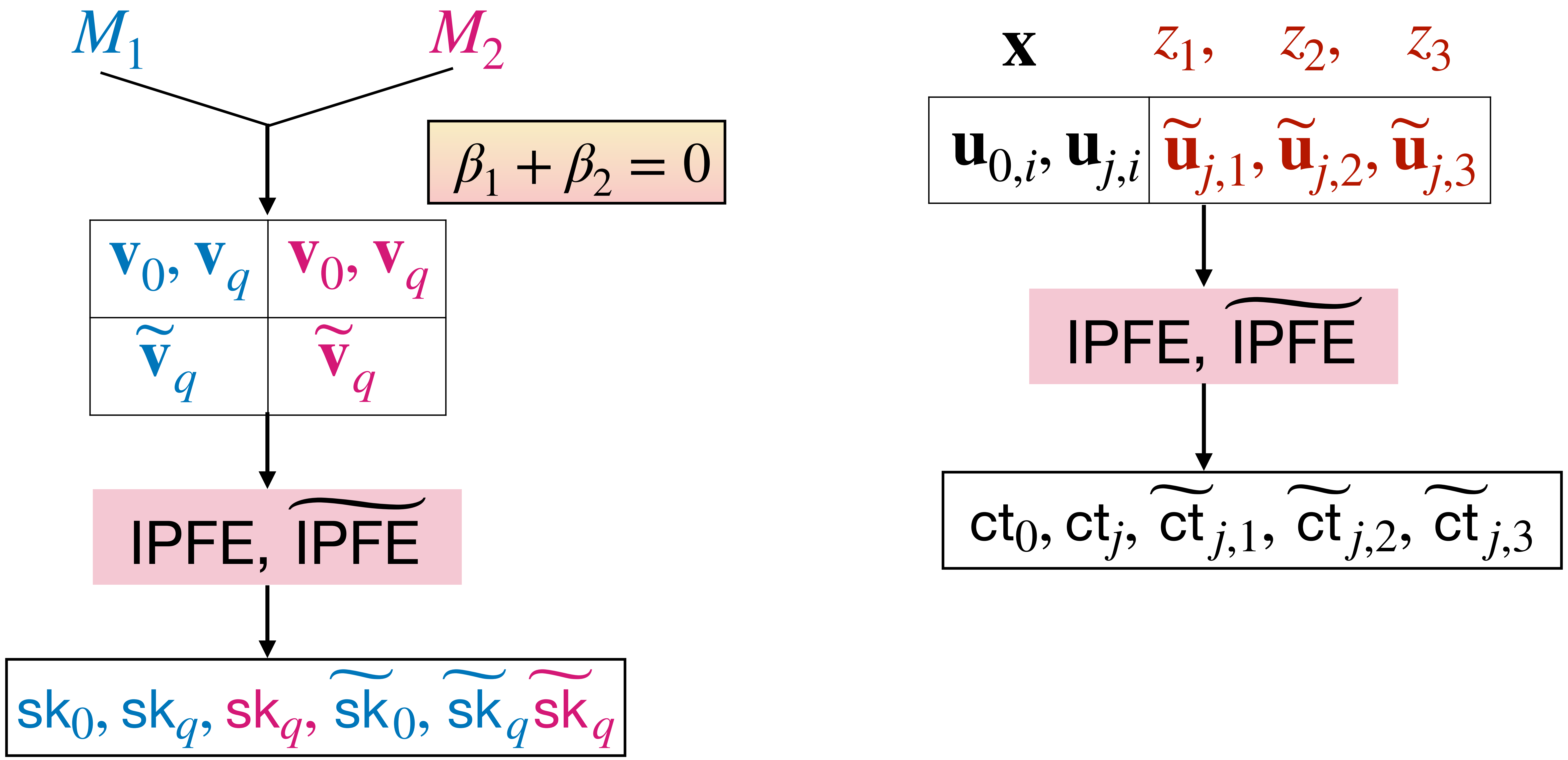
   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$$\mathbf{v}_{0,k} = (\ \pi_k(k,1),\ \ \cdots\ ) \qquad \mathbf{v}_{q,k} = (\ \pi_k(k,1),\ -\mathbf{r}_{M_k}[q],\ (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ \pi_k(k,1),\ \ \cdots\ )$$

$$\mathbf{u}_{0,i} = (\ \rho_i(-1,i),\ \ \cdots\ ) \qquad \mathbf{u}_{j,i} = (\ \rho_i(-1,i),\ \mathbf{r}_{\mathbf{x}}[j-1],\ \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \rho_i(-1,i),\ \ \cdots\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_2(\mathbf{x}) + z_2 M_1(\mathbf{x})}$


M_1
M_2
$\beta_1 + \beta_2 = 0$
$\mathbf{v}_0, \mathbf{v}_q$
$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$
$\widetilde{\mathbf{v}}_q$
IPFE, $\widetilde{\text{IPFE}}$
$\mathsf{sk}_0, \mathsf{sk}_q, \mathsf{sk}_q, \widetilde{\mathsf{sk}}_0, \widetilde{\mathsf{sk}}_q \widetilde{\mathsf{sk}}_q$
$\mathbf{x}$
$z_1,$
$z_2,$
z_3
$\mathbf{u}_{0,i}, \mathbf{u}_{j,i}$
$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$
IPFE, $\widetilde{\text{IPFE}}$
$\mathsf{ct}_0, \mathsf{ct}_j, \widetilde{\mathsf{ct}}_{j,1}, \widetilde{\mathsf{ct}}_{j,2}, \widetilde{\mathsf{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire
   $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

2. use **Marginal Randomness** to randomize all *other* labels

   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$$\mathbf{v}_{0,1} = (\ \pi_1(1,1),\quad \cdots\ )$$
$$\mathbf{u}_{0,2} = (\ \rho_2(-1,2),\quad \cdots\ )$$

❌

$$\mathbf{v}_{q,1} = (\ \pi_1(1,1),\ -\mathbf{r}_{M_1}[q],\ (\mathbf{M}_{x_j,1}\mathbf{r}_{M_1})[q]\ )$$
$$\mathbf{u}_{j,2} = (\ \rho_2(-1,2),\ \mathbf{r}_{\mathbf{x}}[j-1],\quad \mathbf{r}_{\mathbf{x}}[j]\ )$$

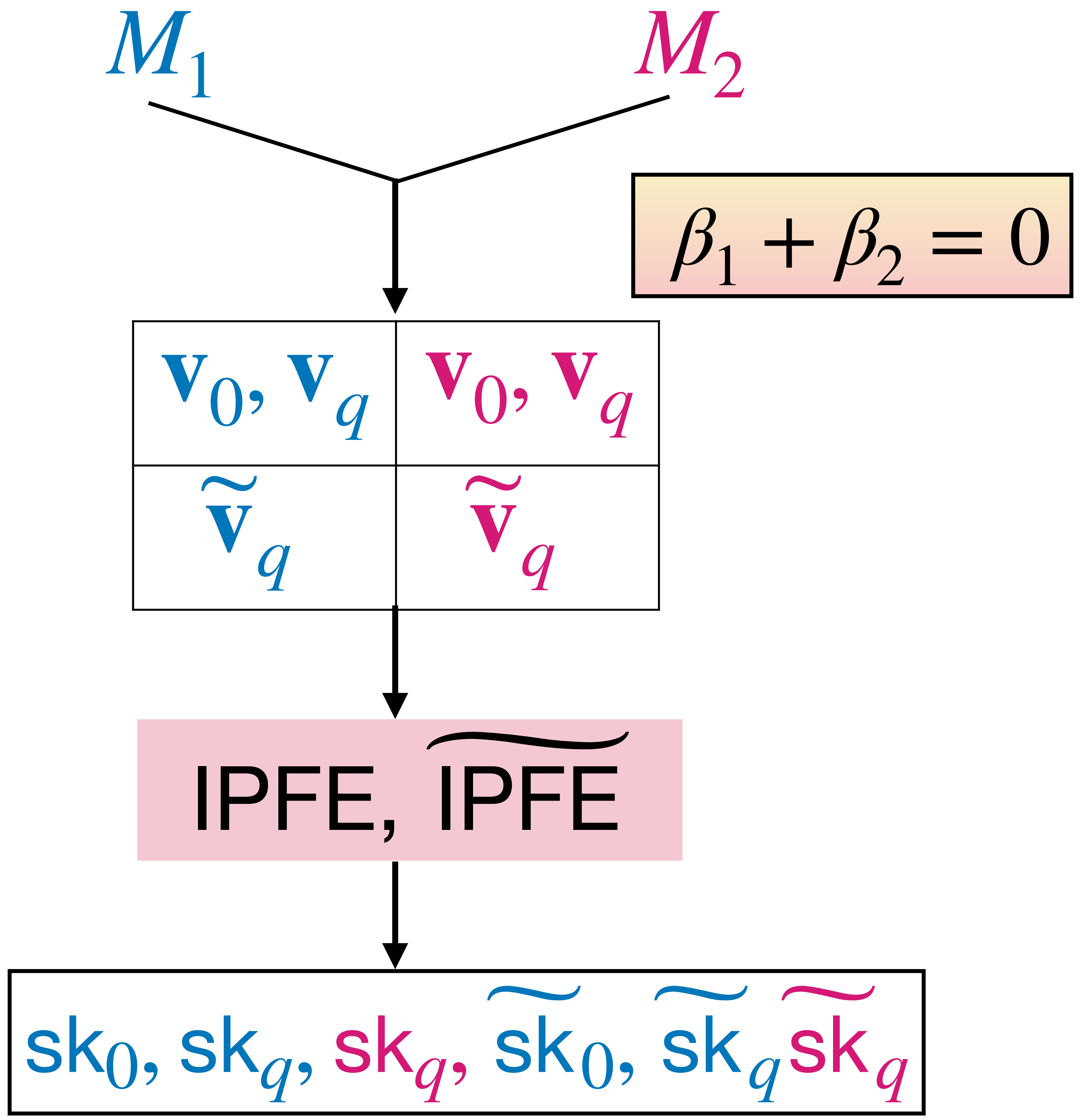
❌

$$\widetilde{\mathbf{v}}_{q,1} = (\ \pi_1(1,1),\quad \cdots\ )$$
$$\widetilde{\mathbf{u}}_{j,2} = (\ \rho_2(-1,2),\quad \cdots\ )$$

❌

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$



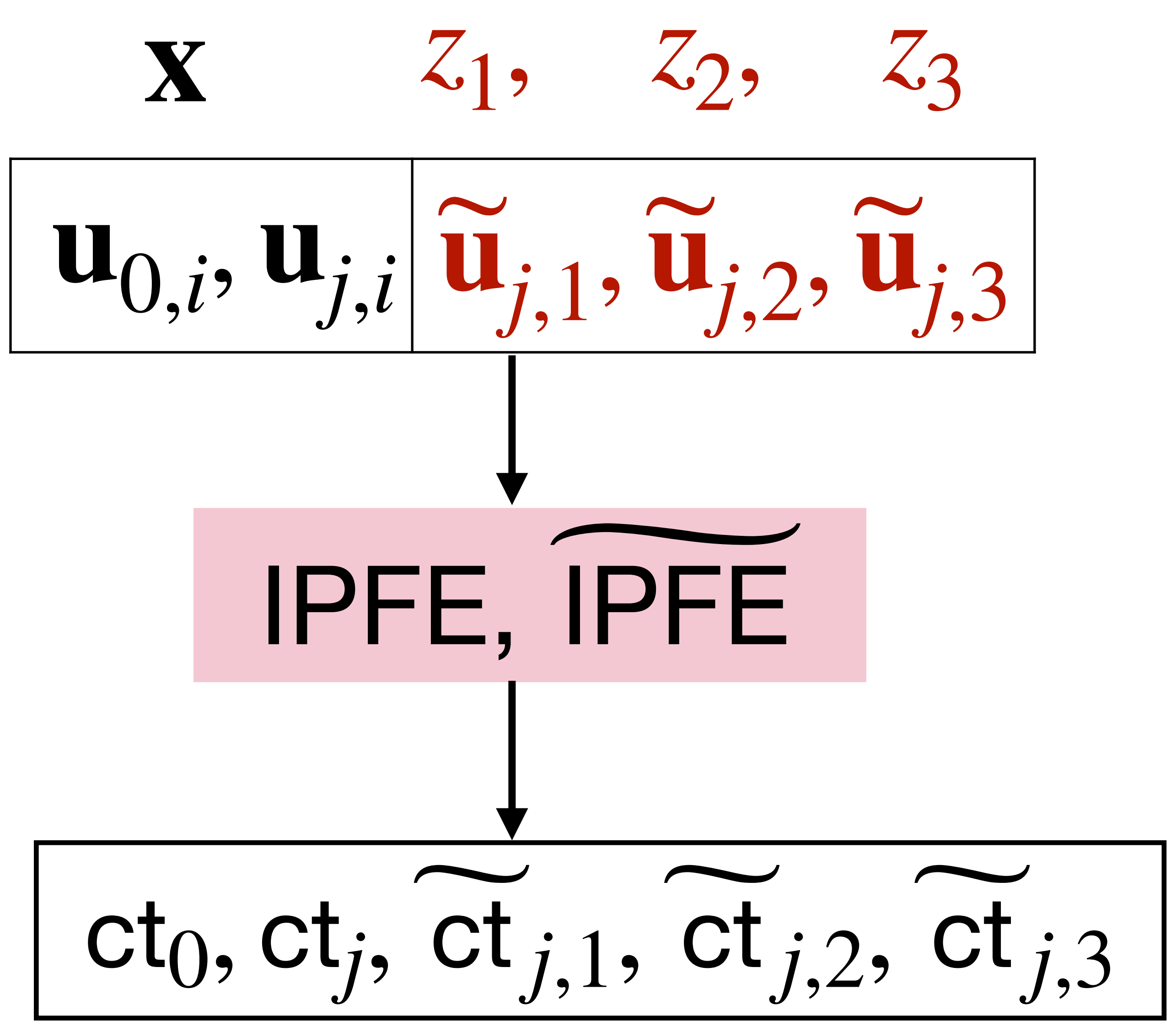
M_1 M_2

$\beta_1 + \beta_2 = 0$

$\mathbf{v}_0, \mathbf{v}_q$	$\mathbf{v}_0, \mathbf{v}_q$
$\widetilde{\mathbf{v}}_q$	$\widetilde{\mathbf{v}}_q$

IPFE, $\widetilde{\text{IPFE}}$

$\mathsf{sk}_0, \mathsf{sk}_q, \mathsf{sk}_q, \widetilde{\mathsf{sk}}_0, \widetilde{\mathsf{sk}}_q \widetilde{\mathsf{sk}}_q$



$\mathbf{x}$ $z_1,$ $z_2,$ z_3

$\mathbf{u}_{0,i}, \mathbf{u}_{j,i}$	$\widetilde{\mathbf{u}}_{j,1}, \widetilde{\mathbf{u}}_{j,2}, \widetilde{\mathbf{u}}_{j,3}$

IPFE, $\widetilde{\text{IPFE}}$

$\mathsf{ct}_0, \mathsf{ct}_j, \widetilde{\mathsf{ct}}_{j,1}, \widetilde{\mathsf{ct}}_{j,2}, \widetilde{\mathsf{ct}}_{j,3}$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \mathsf{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

2. use **Marginal Randomness** to randomize all *other* labels

   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$$\mathbf{v}_{0,k} = (\ \pi_k(k,1),\ \ \cdots\ ) \qquad \mathbf{v}_{q,k} = (\ \pi_k(k,1),\ -\mathbf{r}_{M_k}[q],\ (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ \pi_k(k,1),\ \ \cdots\ )$$

$$\mathbf{u}_{0,i} = (\ \rho_i(-1,i),\ \ \cdots\ ) \qquad \mathbf{u}_{j,i} = (\ \rho_i(-1,i),\ \mathbf{r}_{\mathbf{x}}[j-1],\ \ \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \rho_i(-1,i),\ \ \cdots\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_2}[1], \quad \beta_k \quad)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[0], \quad 1 \quad)$$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

$$\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$$

2. use **Marginal Randomness** to randomize all *other* labels

$$\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \cdots)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \cdots)$$

$$\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad)$$

$$\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \quad)$$

$$\widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \quad \cdots \quad)$$

$$\widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \quad \cdots \quad)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{0,k} = (\ \ \pi_k(k,1),\ \ \boxed{1,}\ \ \ 0\ \ )$$

FH-IPFE

$$\mathbf{u}_{0,i} = (\ \ \rho_i(-1,i),\ \ \boxed{\ell_{0,i},}\ \ \boxed{1}\ \ )$$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\ \ \pi_k(k,1),\ \ \ldots\ ) \qquad \mathbf{v}_{q,k} = (\ \ \pi_k(k,1),\ \ -\mathbf{r}_{M_k}[q],\ \ (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ \ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ \ \pi_k(k,1),\ \ \ldots\ \ )$$

$$\mathbf{u}_{0,i} = (\ \ \rho_i(-1,i),\ \ \ldots\ ) \qquad \mathbf{u}_{j,i} = (\ \ \rho_i(-1,i),\ \ \mathbf{r}_{\mathbf{x}}[j-1],\ \ \mathbf{r}_{\mathbf{x}}[j]\ \ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \ \rho_i(-1,i),\ \ \ldots\ \ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2, M_4)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{0,k} = (\ \pi_k(k,1), \ \boxed{1,} \ \ 0 \ )$$

$$\mathbf{u}_{0,i} = (\ \rho_i(-1,i), \ \boxed{\ell_{0,i},} \ \boxed{1} \ )$$

$$\boxed{\rho_i(-1,i) \cdot \pi_4(4,1) \neq 0}$$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\ \pi_k(k,1), \ \ldots \ ) \qquad \mathbf{v}_{q,k} = (\ \pi_k(k,1), \ -\mathbf{r}_{M_k}[q], \ (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ \pi_k(k,1), \ \ldots \ )$$

$$\mathbf{u}_{0,i} = (\ \rho_i(-1,i), \ \ldots \ ) \qquad \mathbf{u}_{j,i} = (\ \rho_i(-1,i), \ \mathbf{r}_{\mathbf{x}}[j-1], \ \mathbf{r}_{\mathbf{x}}[j] \ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \rho_i(-1,i), \ \ldots \ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2, M_4\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \boxed{1,} \quad 0 \quad)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \boxed{\ell_{0,i},} \quad \boxed{1} \quad)$$

$$\boxed{\rho_i(-1,i) \cdot \pi_4(4,1) \neq 0}$$

additional entropy from *index encoding*

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \dots\ ) \qquad \mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad) \qquad \widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \quad \dots \quad)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \dots\ ) \qquad \mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \quad) \qquad \widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \quad \dots \quad)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2, M_4)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \boxed{1,} \quad 0 \quad)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \boxed{\ell_{0,i},} \quad \boxed{1} \quad)$$

$$\boxed{\rho_i(-1,i) \cdot \pi_4(4,1) \neq 0} \Rightarrow \boxed{\beta_4 \leftarrow \$} \Rightarrow \boxed{\beta_1 + \beta_2 + \beta_4 \neq 0}$$

additional entropy from *index encoding*

along with FH-IPFE

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i \, M_i(\mathbf{x}) + \beta_i$ in the *first* label
   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$
2. use **Marginal Randomness** to randomize all *other* labels
   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \dots \;) \qquad \mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad) \qquad \widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \quad \dots \quad)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \cdots \;) \qquad \mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \quad) \qquad \widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \quad \cdots \quad)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\ M_1, M_2, M_4\ )$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$ ❌

$$\mathbf{v}_{0,k} = (\ \pi_k(k,1),\ \boxed{1,}\ \ 0\ )$$

$$\mathbf{u}_{0,i} = (\ \rho_i(-1,i),\ \boxed{\ell_{0,i},}\ \boxed{1}\ )$$

$$\boxed{\rho_i(-1,i) \cdot \pi_4(4,1) \neq 0} \Rightarrow \boxed{\beta_4 \leftarrow \$} \Rightarrow \boxed{\beta_1 + \beta_2 + \beta_4 \neq 0}$$

additional entropy from *index encoding*

along with FH-IPFE

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\ \pi_k(k,1),\ \ \ldots\ ) \qquad \mathbf{v}_{q,k} = (\ \pi_k(k,1),\ -\mathbf{r}_{M_k}[q],\ (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ \pi_k(k,1),\ \ \ldots\ )$$

$$\mathbf{u}_{0,i} = (\ \rho_i(-1,i),\ \ \ldots\ ) \qquad \mathbf{u}_{j,i} = (\ \rho_i(-1,i),\ \mathbf{r}_{\mathbf{x}}[j-1],\ \ \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \rho_i(-1,i),\ \ \ldots\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{q,k} = (\ \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ )$$

$$\mathbf{u}_{j,i} = (\ \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j]\ )$$

$$\xrightarrow{i = k} \ell_{j,i}$$

Steps: Simulation Security

1. use **RevSamp** to hardwire
   $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label
   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels
   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\ \pi_k(k,1), \quad \cdots\ ) \qquad \mathbf{v}_{q,k} = (\ \pi_k(k,1), \ -\mathbf{r}_{M_k}[q], \ (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ \pi_k(k,1), \quad \cdots\ )$$

$$\mathbf{u}_{0,i} = (\ \rho_i(-1,i), \quad \cdots\ ) \qquad \mathbf{u}_{j,i} = (\ \rho_i(-1,i), \ \mathbf{r}_{\mathbf{x}}[j-1], \ \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \rho_i(-1,i), \quad \cdots\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad [\![-\mathbf{r}_{M_k}[q]]\!]_2 \qquad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \;)$$

$$\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad [\![\mathbf{r}_{\mathbf{x}}[j-1]]\!]_1 \qquad \mathbf{r}_{\mathbf{x}}[j] \qquad)$$

$$\xrightarrow{i = k} \ell_{j,i}$$

Steps: Simulation Security

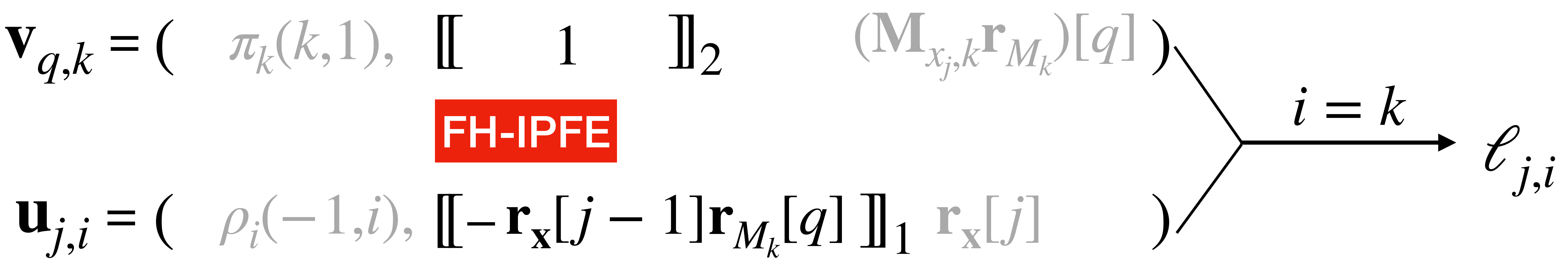
1. use **RevSamp** to hardwire
   $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label
   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$
2. use **Marginal Randomness** to randomize all *other* labels
   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \ldots \;) \qquad \mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad) \qquad \widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \qquad \ldots \qquad)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \ldots \;) \qquad \mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \qquad \mathbf{r}_{\mathbf{x}}[j] \qquad) \qquad \widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \qquad \ldots \qquad)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (\, M_1, M_2 \,)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$



$$\mathbf{v}_{q,k} = (\ \pi_k(k,1),\ [\![\ 1\ ]\!]_2 \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ )$$

FH-IPFE

$$\mathbf{u}_{j,i} = (\ \rho_i(-1,i),\ [\![-\mathbf{r}_{\mathbf{x}}[j-1]\mathbf{r}_{M_k}[q]\ ]\!]_1\ \ \mathbf{r}_{\mathbf{x}}[j] \quad)$$

$$\xrightarrow{i = k} \ell_{j,i}$$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i\, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

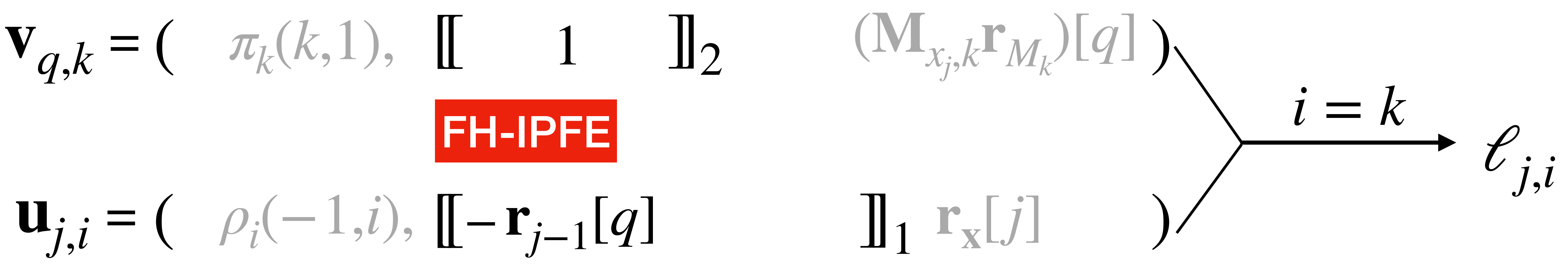
   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\ \pi_k(k,1),\ \ \ldots\ ) \qquad \mathbf{v}_{q,k} = (\ \pi_k(k,1),\ -\mathbf{r}_{M_k}[q],\ (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q]\ ) \qquad \widetilde{\mathbf{v}}_{q,k} = (\ \pi_k(k,1),\ \ \ldots\ )$$

$$\mathbf{u}_{0,i} = (\ \rho_i(-1,i),\ \ \ldots\ ) \qquad \mathbf{u}_{j,i} = (\ \rho_i(-1,i),\ \mathbf{r}_{\mathbf{x}}[j-1],\ \ \mathbf{r}_{\mathbf{x}}[j]\ ) \qquad \widetilde{\mathbf{u}}_{j,i} = (\ \rho_i(-1,i),\ \ \ldots\ )$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$



$\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad [\![\quad 1 \quad]\!]_2 \qquad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \ )$

FH-IPFE

$\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad [\![-\mathbf{r}_{j-1}[q] \qquad]\!]_1 \quad \mathbf{r}_{\mathbf{x}}[j] \qquad)$

$\xrightarrow{i = k} \ell_{j,i}$

$\mathbf{r}_{j-1} = \mathbf{r}_{\mathbf{x}}[j-1] \cdot \mathbf{r}_{M_k}$

Steps: Simulation Security

1. use **RevSamp** to hardwire
   $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label
   $\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$
2. use **Marginal Randomness**
   to randomize all *other* labels
   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \qquad \cdots)$

$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \qquad \cdots)$

$\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad)$

$\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \qquad \mathbf{r}_{\mathbf{x}}[j] \qquad)$

$\widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \qquad \cdots \quad)$

$\widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \qquad \cdots \quad)$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad [\![\quad 1 \quad]\!]_2 \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \;)$$

$$\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad [\![\; -\mathbf{s}_{j-1}[q] \leftarrow \$ \quad]\!]_1 \quad \mathbf{r}_{\mathbf{x}}[j] \quad)$$

$$\xrightarrow{i = k} \ell_{j,i}$$

$$\mathbf{r}_{j-1} = \mathbf{r}_{\mathbf{x}}[j-1] \cdot \mathbf{r}_{M_k} \xrightarrow{\text{DDH in } \mathbb{G}_1} \boxed{\mathbf{s}_{j-1} \leftarrow \$}$$

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i \, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \dots)$ $\qquad$ $\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad)$ $\qquad$ $\widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \quad \dots \quad)$

$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \dots)$ $\qquad$ $\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \quad)$ $\qquad$ $\widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \quad \dots \quad)$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad [\![\quad 1 \quad]\!]_2 \qquad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \ )$$

$$\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad [\![\ -\mathbf{s}_{j-1}[q] \leftarrow \$ \quad]\!]_1 \ \ \mathbf{r}_{\mathbf{x}}[j] \qquad)$$

$$\xrightarrow{i = k} \ell_{j,i}$$

$$\mathbf{r}_{j-1} = \mathbf{r}_{\mathbf{x}}[j-1] \cdot \mathbf{r}_{M_k} \xrightarrow{\text{DDH in } \mathbb{G}_1} \boxed{\mathbf{s}_{j-1} \leftarrow \$}$$ ✗

$\mathbf{r}_{M_k}$ may appear in *other* places of $\mathbf{v}_{q,k}$ for any k

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i \, M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \dots)$ $\qquad$ $\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad)$ $\qquad$ $\widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \quad \dots \quad)$

$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \dots)$ $\qquad$ $\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \quad)$ $\qquad$ $\widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \quad \dots \quad)$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{q,k} = (\pi_k(k,1), [\![1]\!]_2 \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q])$$

$$\mathbf{u}_{j,i} = (\rho_i(-1,i), [\![-\mathbf{s}_{j-1}[q] \leftarrow \$]\!]_1 \; \mathbf{r}_{\mathbf{x}}[j])$$

$$\xrightarrow{i = k} \ell_{j,i}$$

$$\mathbf{r}_{j-1} = \mathbf{r}_{\mathbf{x}}[j-1] \cdot \mathbf{r}_{M_k} \xrightarrow{\text{DDH in } \mathbb{G}_1} \boxed{\mathbf{s}_{j-1} \leftarrow \$}$$ ✗

$\mathbf{r}_{M_k}$ may appear in *other* places of $\mathbf{v}_{q,k}$ for any k

use *distributed randomness mechanism*

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\pi_k(k,1), \quad \dots)$$
$$\mathbf{u}_{0,i} = (\rho_i(-1,i), \quad \dots)$$

$$\mathbf{v}_{q,k} = (\pi_k(k,1), \; -\mathbf{r}_{M_k}[q], \; (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q])$$
$$\mathbf{u}_{j,i} = (\rho_i(-1,i), \; \mathbf{r}_{\mathbf{x}}[j-1], \; \mathbf{r}_{\mathbf{x}}[j])$$

$$\widetilde{\mathbf{v}}_{q,k} = (\pi_k(k,1), \quad \dots)$$
$$\widetilde{\mathbf{u}}_{j,i} = (\rho_i(-1,i), \quad \dots)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad \boxed{1,} \quad 0 \quad)$$

$$\mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \boxed{\ell_{j,i} \leftarrow \$,} \quad 0 \quad)$$

$$\xrightarrow{i = k} \ell_{j,i}$$

$$\mathbf{r}_{j-1} = \mathbf{r}_{\mathbf{x}}[j-1] \cdot \mathbf{r}_{M_k} \xrightarrow{\text{DDH in } \mathbb{G}_1} \boxed{\mathbf{s}_{j-1} \leftarrow \$}$$ ✓

$\mathbf{r}_{M_k}$ may appear in *other* places of $\mathbf{v}_{q,k}$ for any k

use *distributed randomness mechanism*

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\boxed{\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)}$

2. use **Marginal Randomness** to randomize all *other* labels

   $\boxed{\ell_{j,i} \leftarrow \$ \text{ for all } j > 1}$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \cdots \;) \qquad \mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad) \qquad \widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \quad \cdots \quad)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \cdots \;) \qquad \mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \quad) \qquad \widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \quad \cdots \quad)$$

Our Idea for FE-UAWS for TMs (DFA for concreteness)

function	input	output
$M = (M_1, M_2)$	$(\mathbf{x}, \mathbf{z} = (z_1, z_2, z_3))$	$\boxed{z_1 M_1(\mathbf{x}) + z_2 M_2(\mathbf{x})}$

$$\widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \quad \boxed{1,} \quad 0 \quad)$$

$$\widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \quad \boxed{\ell_{j,i} \leftarrow \$,} \quad 0 \quad)$$

$$\xrightarrow{i = k} \ell_{j,i}$$

$$\mathbf{r}_{j-1} = \mathbf{r}_{\mathbf{x}}[j-1] \cdot \mathbf{r}_{M_k} \xrightarrow{\text{DDH in } \mathbb{G}_1} \boxed{\mathbf{s}_{j-1} \leftarrow \$}$$ ✓

$\mathbf{r}_{M_k}$ may appear in *other* places of $\mathbf{v}_{q,k}$ for any k

use *distributed randomness mechanism*

Steps: Simulation Security

1. use **RevSamp** to hardwire $\mu_i = z_i M_i(\mathbf{x}) + \beta_i$ in the *first* label

   $\ell_{0,i} \leftarrow \text{RevSamp}(M_i, \mathbf{x}, \mu_i, \cdots)$

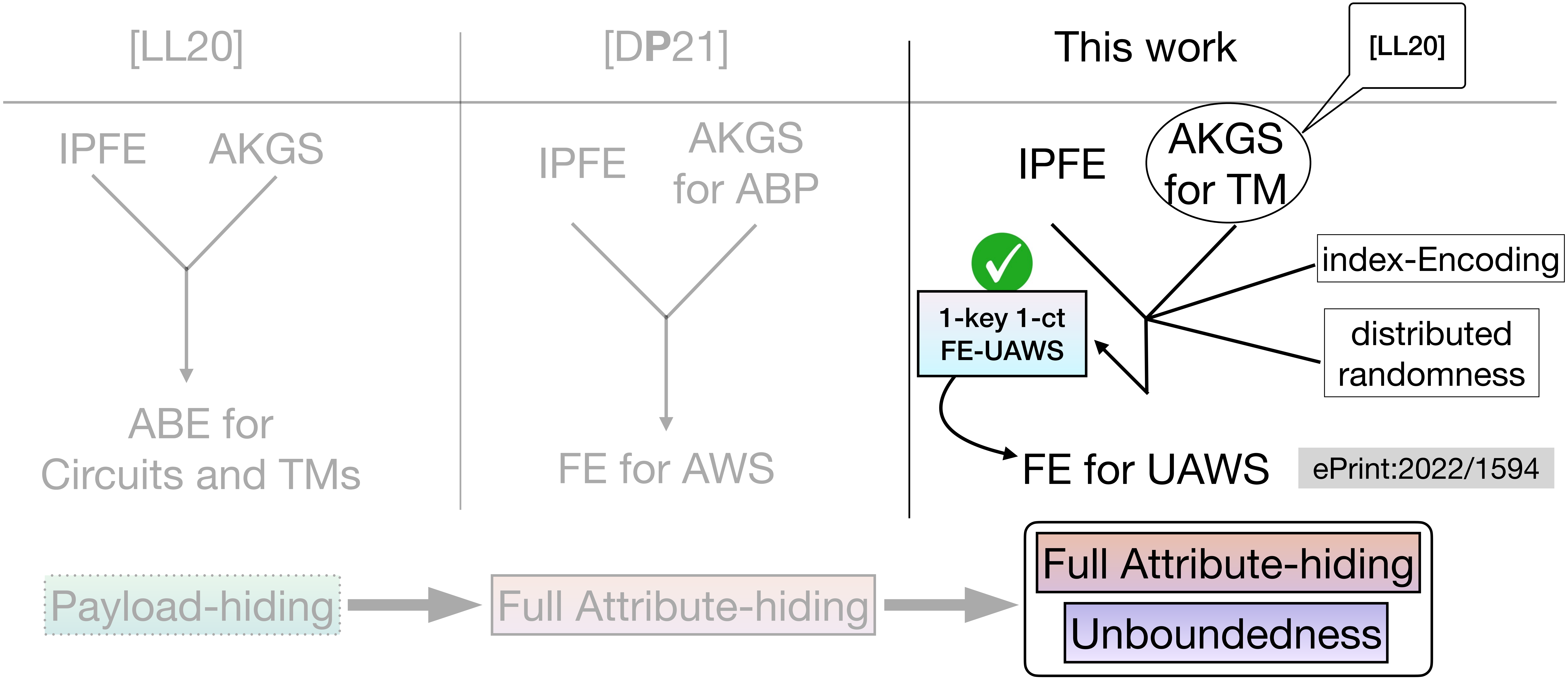
2. use **Marginal Randomness** to randomize all *other* labels

   $\ell_{j,i} \leftarrow \$$ for all $j > 1$

$$\mathbf{v}_{0,k} = (\quad \pi_k(k,1), \quad \ldots \;) \qquad \mathbf{v}_{q,k} = (\quad \pi_k(k,1), \quad -\mathbf{r}_{M_k}[q], \quad (\mathbf{M}_{x_j,k}\mathbf{r}_{M_k})[q] \quad) \qquad \widetilde{\mathbf{v}}_{q,k} = (\quad \pi_k(k,1), \quad \ldots \quad)$$

$$\mathbf{u}_{0,i} = (\quad \rho_i(-1,i), \quad \ldots \;) \qquad \mathbf{u}_{j,i} = (\quad \rho_i(-1,i), \quad \mathbf{r}_{\mathbf{x}}[j-1], \quad \mathbf{r}_{\mathbf{x}}[j] \quad) \qquad \widetilde{\mathbf{u}}_{j,i} = (\quad \rho_i(-1,i), \quad \ldots \quad)$$

Roadmap towards FE for UAWS



[LL20]

IPFE

AKGS

ABE for Circuits and TMs

[DP21]

IPFE

AKGS for ABP

FE for AWS

This work

[LL20]

IPFE

AKGS for TM

index-Encoding

distributed randomness

✓

1-key 1-ct FE-UAWS

FE for UAWS

ePrint:2022/1594

Payload-hiding

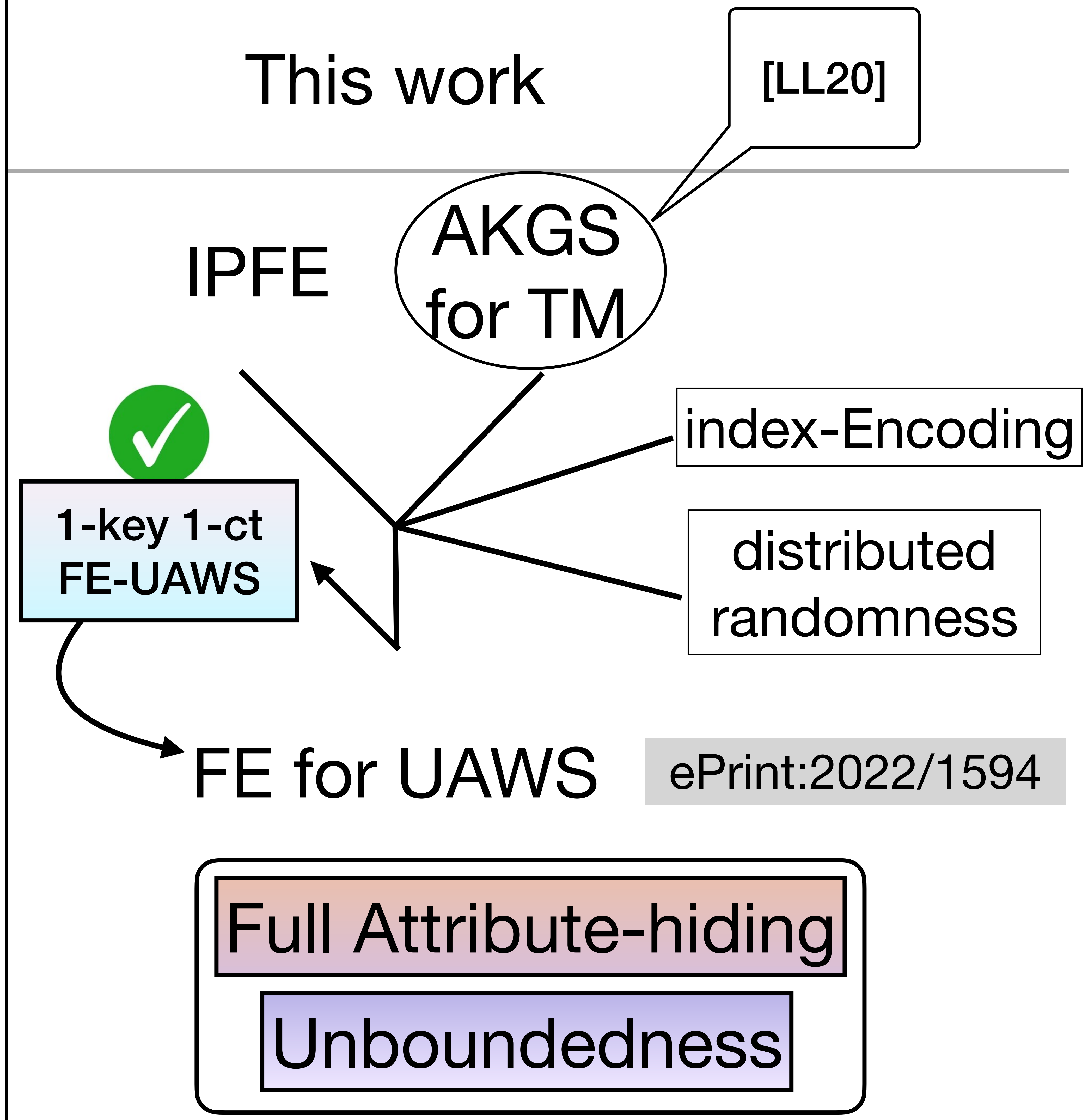
Full Attribute-hiding

Full Attribute-hiding

Unboundedness

Conclusion

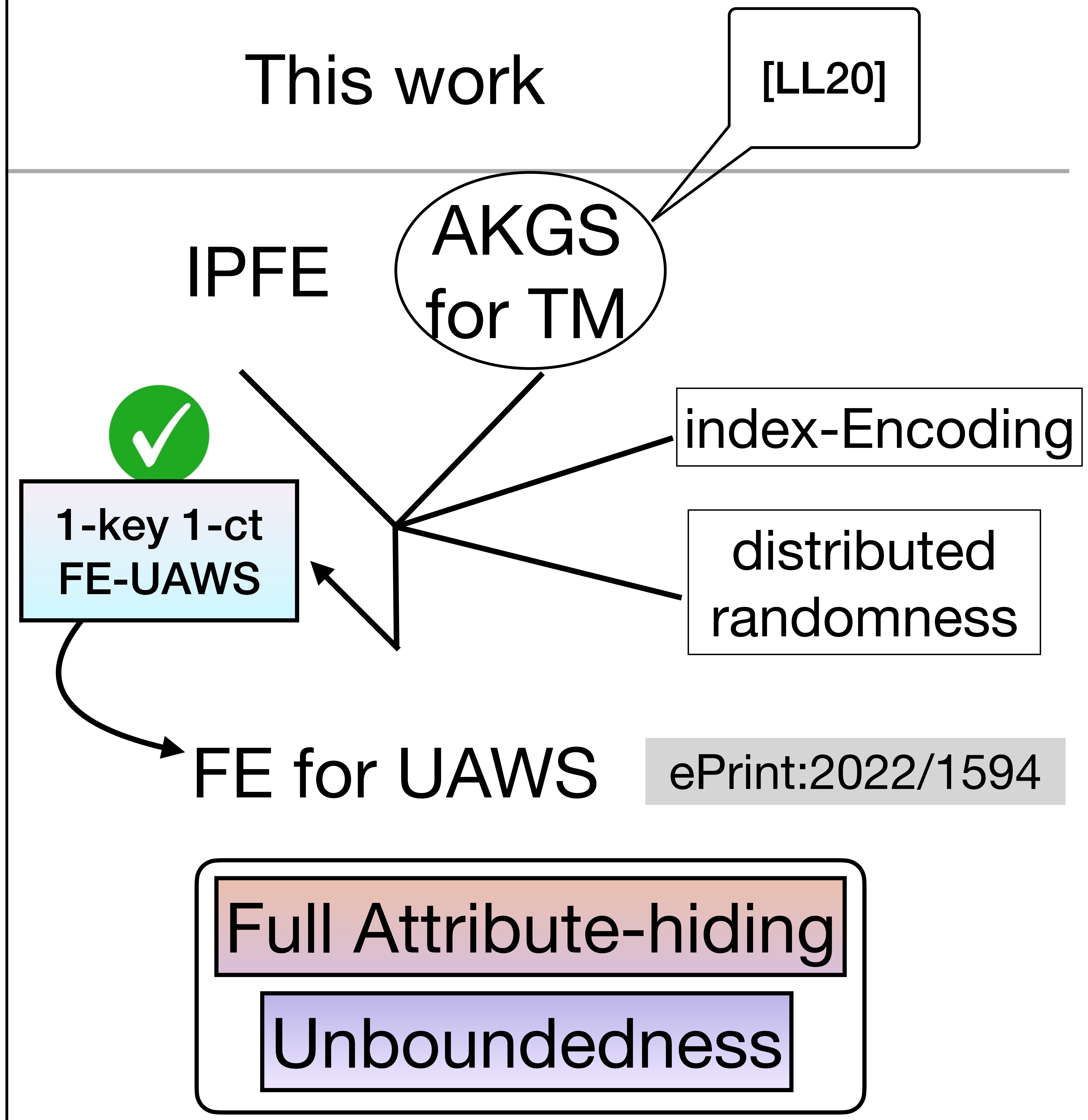
- Definition and Construction of FE for UAWS
  - ✦ input-specific $|\mathsf{ct}| = O(|\mathbf{x}|, |\mathbf{z}|)$
  - ✦ Compact ciphertext
  - ✦ Fully collusion-resistant AD-SIM
  - ✦ Standard assumption: SXDH



Conclusion

- Definition and Construction of FE for UAWS
  - ✦ input-specific $|\mathsf{ct}| = O(|\mathbf{x}|, |\mathbf{z}|)$
  - ✦ Compact ciphertext
  - ✦ Fully collusion-resistant AD-SIM
  - ✦ Standard assumption: SXDH

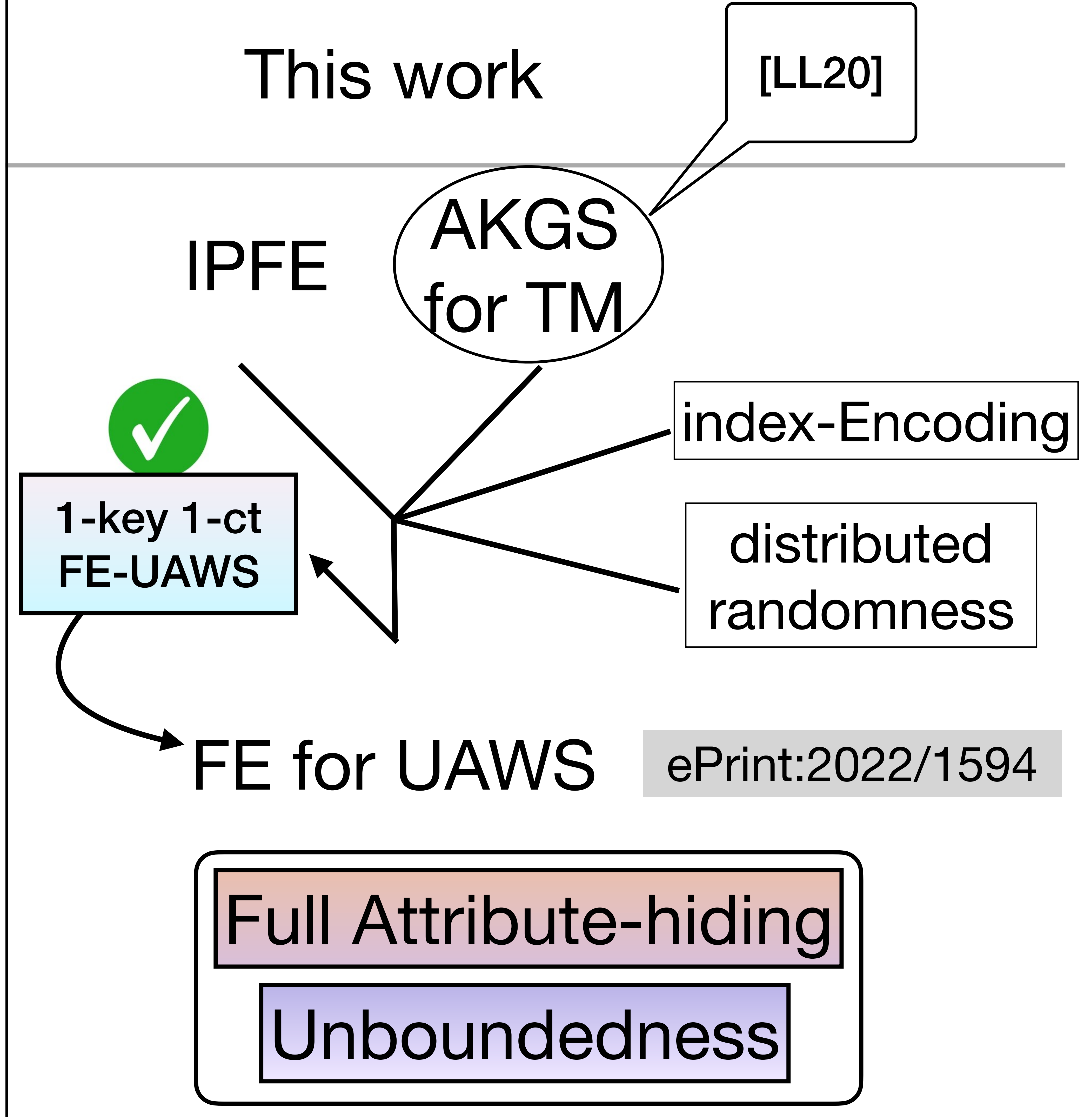
- Future research directions of FE for UAWS
  - ✦ succinctness: $|\mathsf{ct}| = O(|\mathbf{z}|)$



Conclusion

- Definition and Construction of FE for UAWS
  - ✦ input-specific $|\mathsf{ct}| = O(|\mathbf{x}|, |\mathbf{z}|)$
  - ✦ Compact ciphertext
  - ✦ Fully collusion-resistant AD-SIM
  - ✦ Standard assumption: SXDH

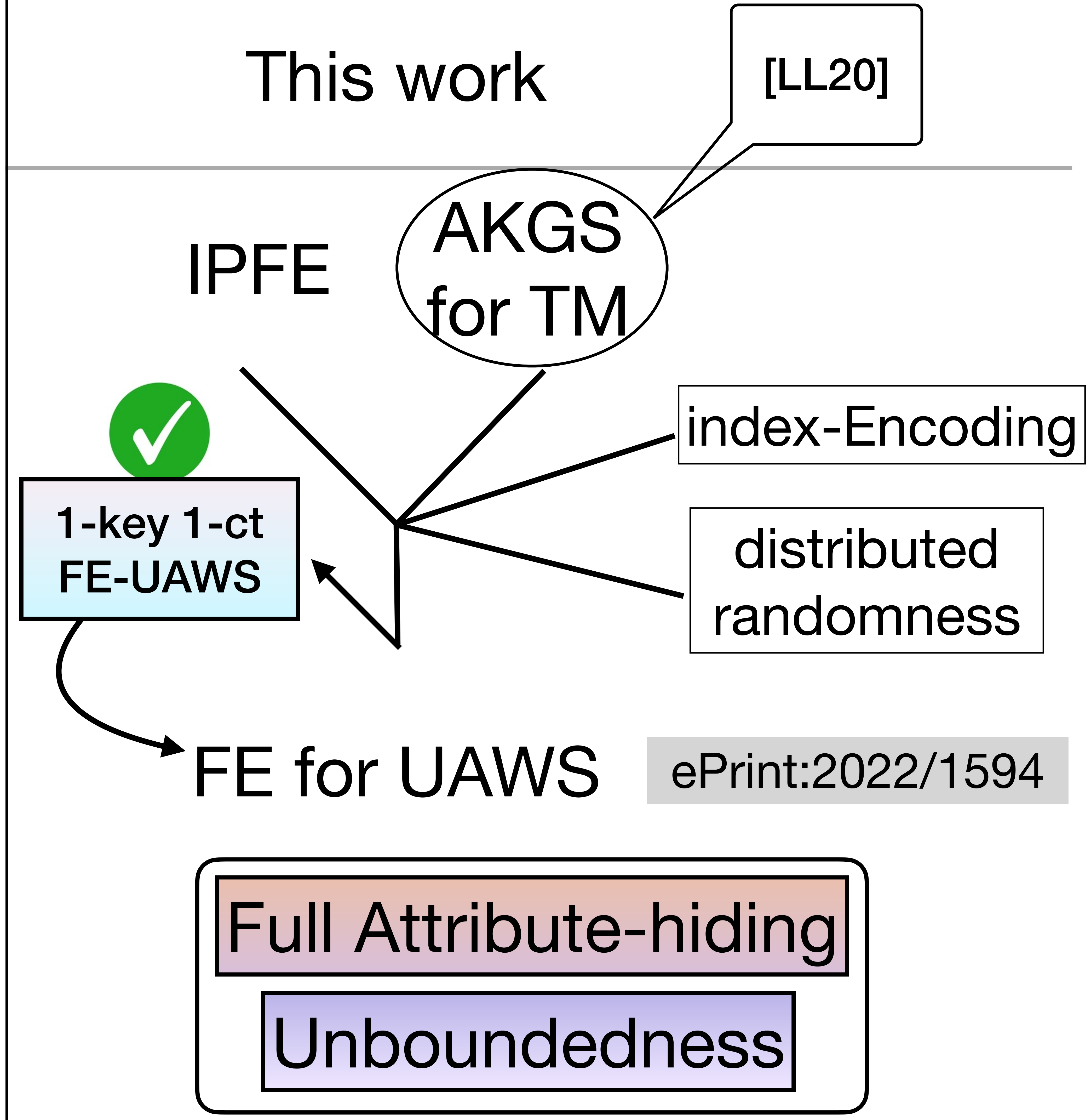
- Future research directions of FE for UAWS
  - ✦ succinctness: $|\mathsf{ct}| = O(|\mathbf{z}|)$
  - ✦ lattice-based assumption: LWE



Conclusion

- Definition and Construction of FE for UAWS
  - ✦ input-specific $|\mathrm{ct}| = O(|\mathbf{x}|, |\mathbf{z}|)$
  - ✦ Compact ciphertext
  - ✦ Fully collusion-resistant AD-SIM
  - ✦ Standard assumption: SXDH

- Future research directions of FE for UAWS
  - ✦ succinctness: $|\mathrm{ct}| = O(|\mathbf{z}|)$
  - ✦ lattice-based assumption: LWE



Thank You!