Witness Encryption for Succinct Functional Commitments and Applications

Matteo Campanelli
Matter Labs

Dario Fiore
IMDEA Software Institute
Madrid, Spain

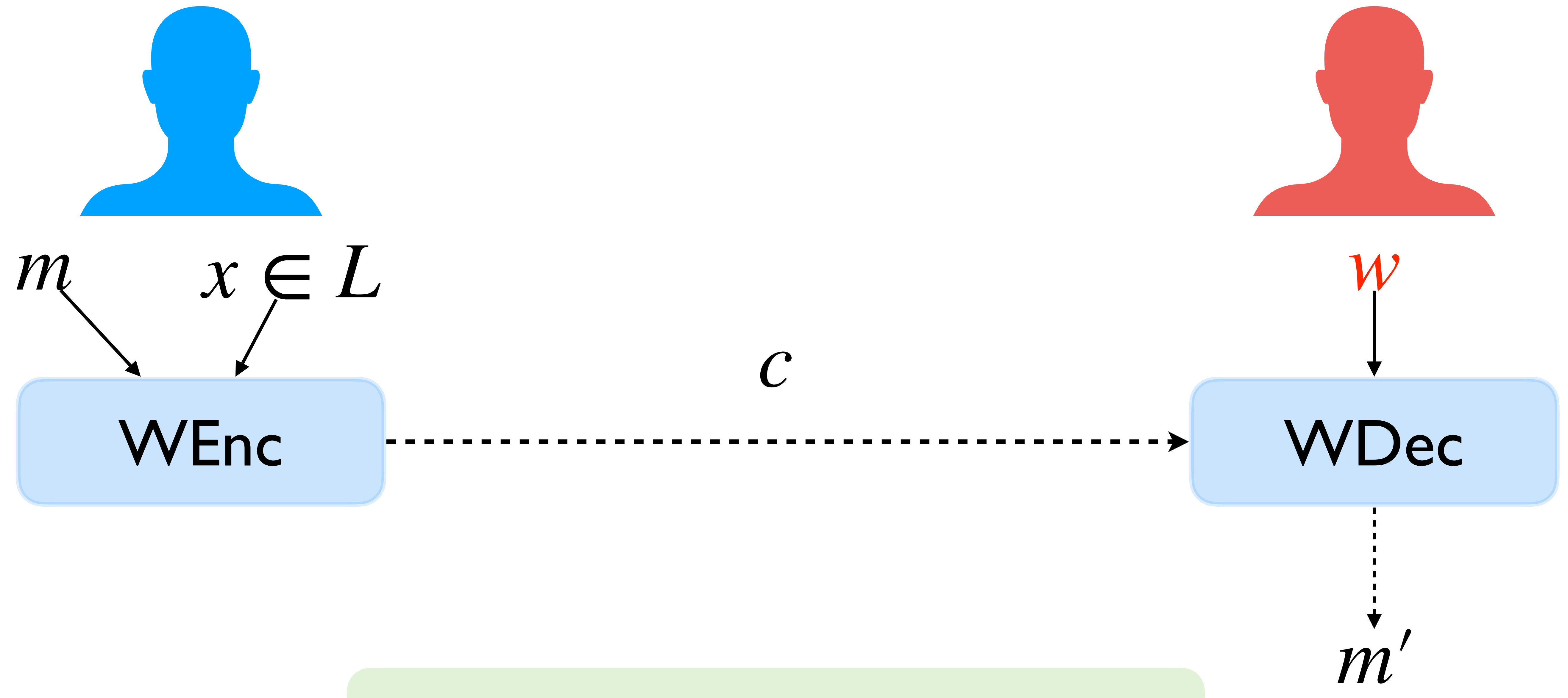
Hamidreza Khoshakhlagh
Concordium
Aarhus, Denmark

PKC 2024 — April 15, 2024

European Research Council
Established by the European Commission

Witness Encryption (WE) [GGSW13]

Main idea: encrypt a message w.r.t. NP statement **x** so that it can be decrypted by who holds a witness of **x**



Correctness

If $(x, w) \in R_L, m' = m$

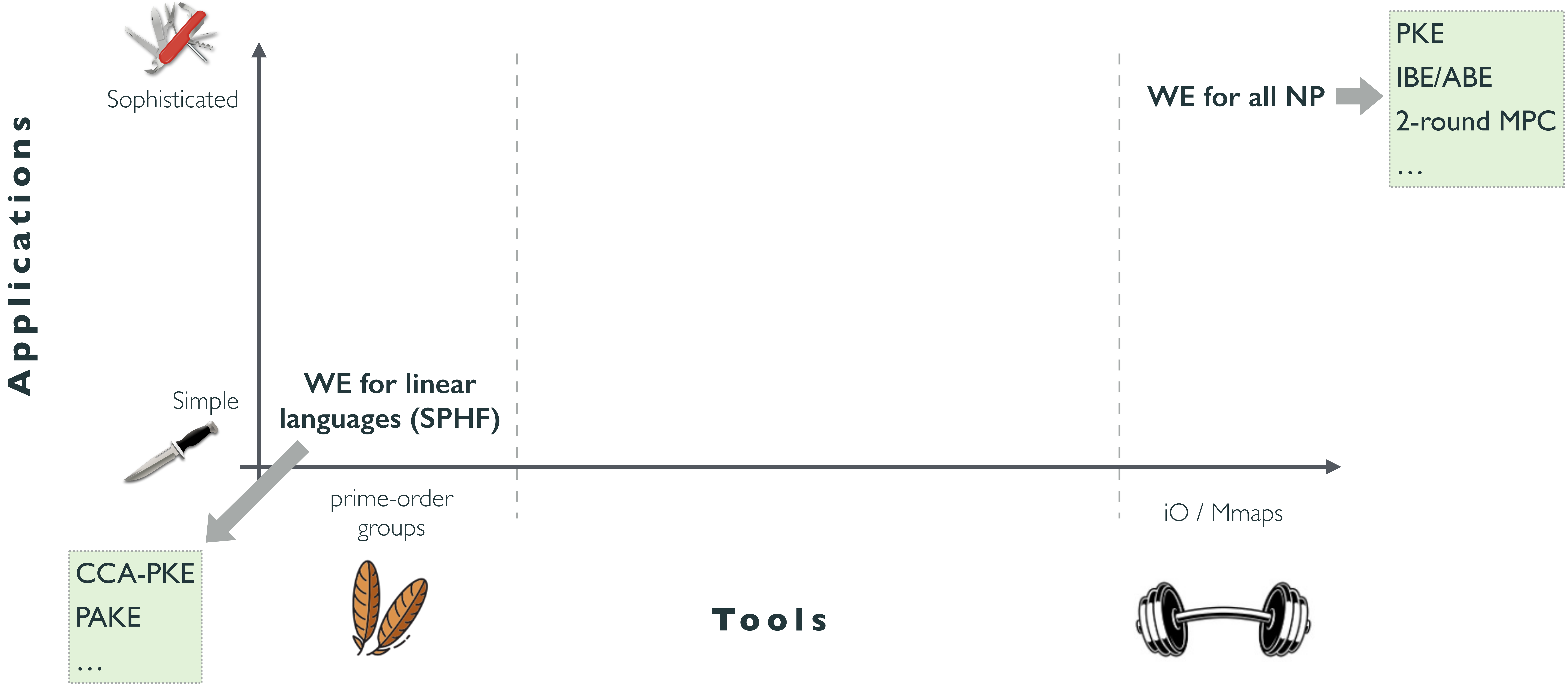
Security

If $x \notin R_L$ then c leaks no information on m

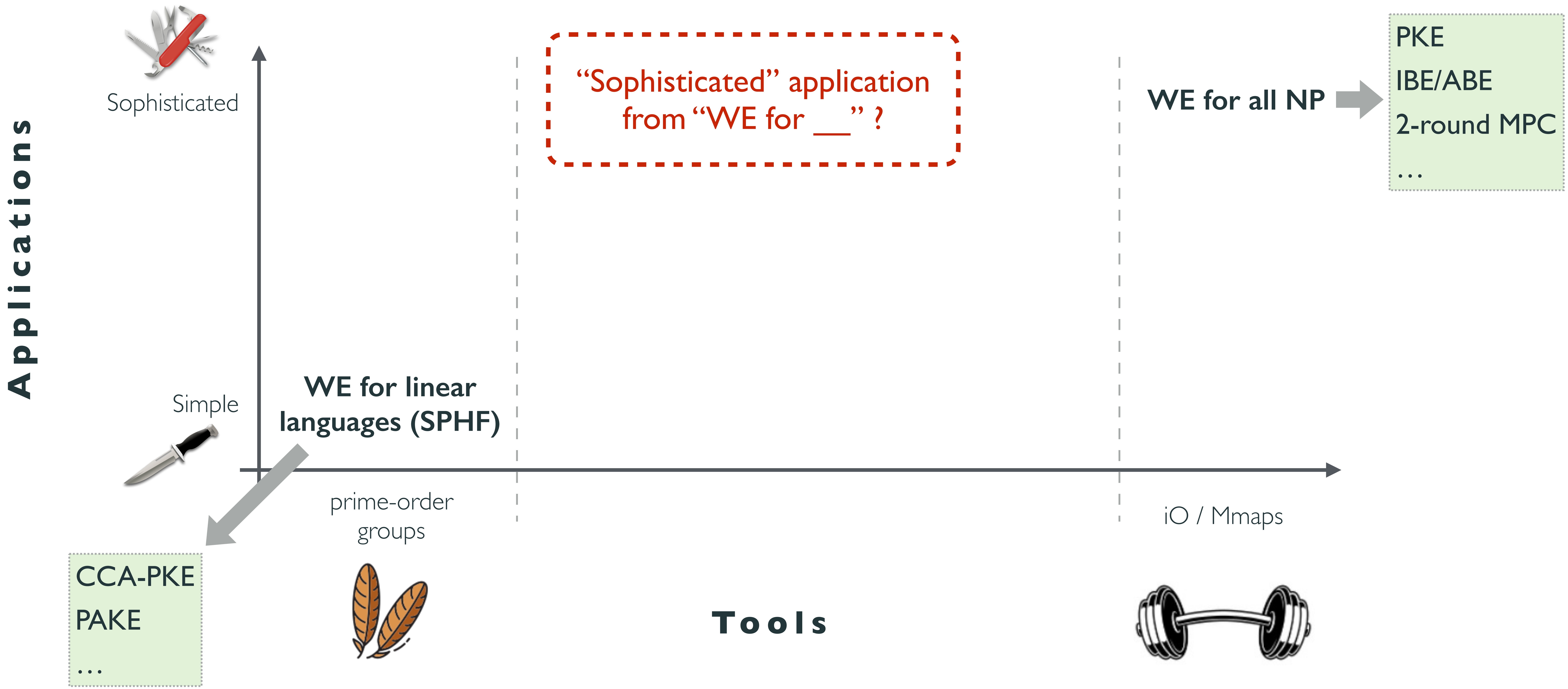
WE: constructions vs. applications

WE: constructions vs. applications

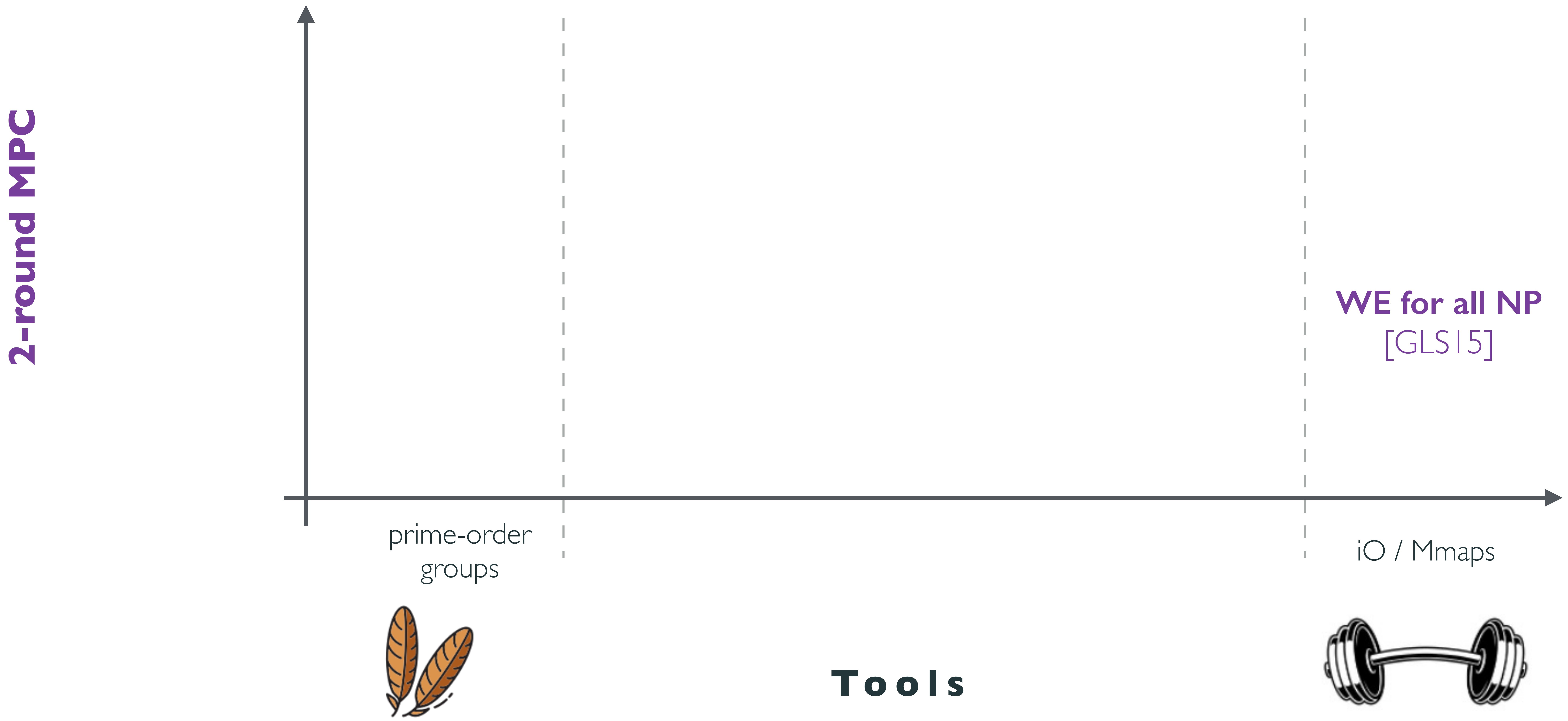
WE: constructions vs. applications



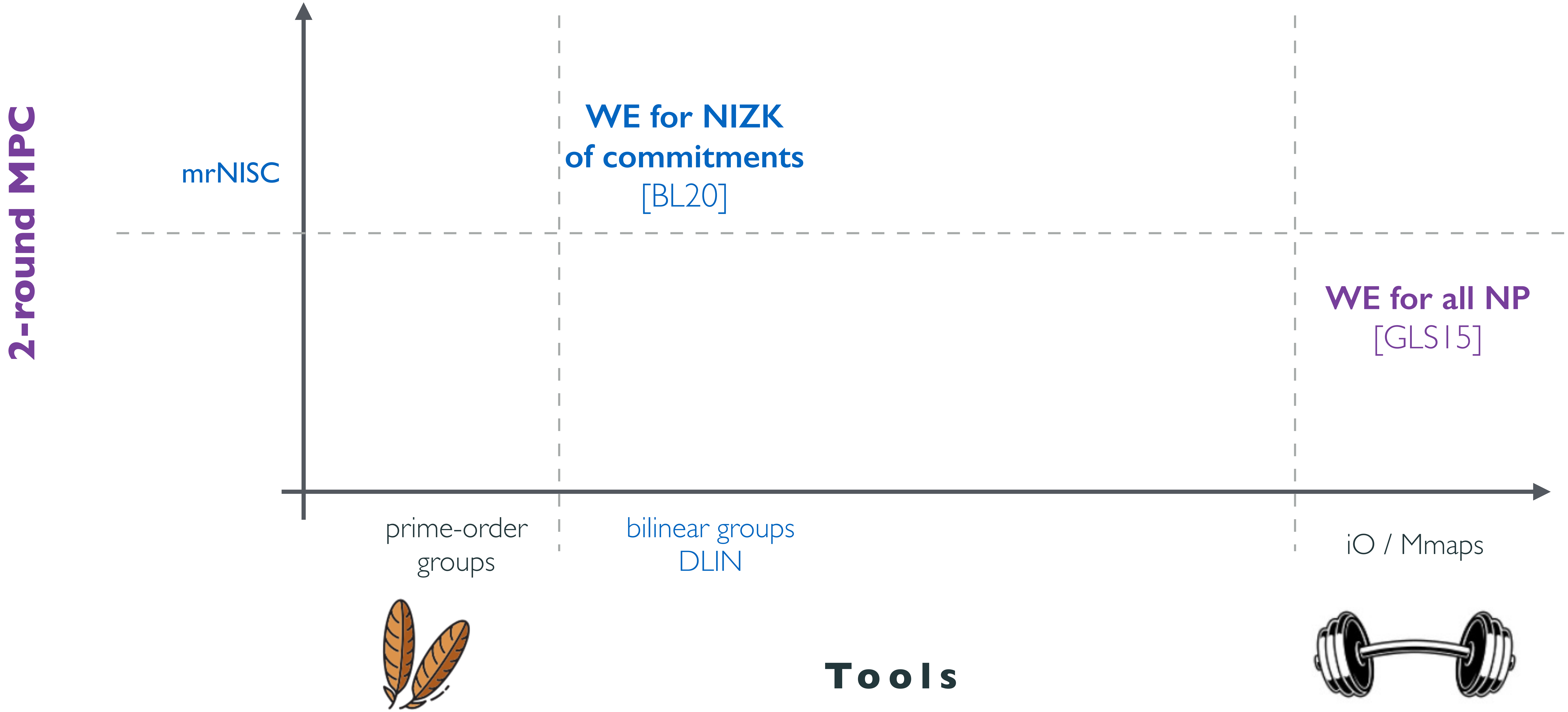
WE: constructions vs. applications



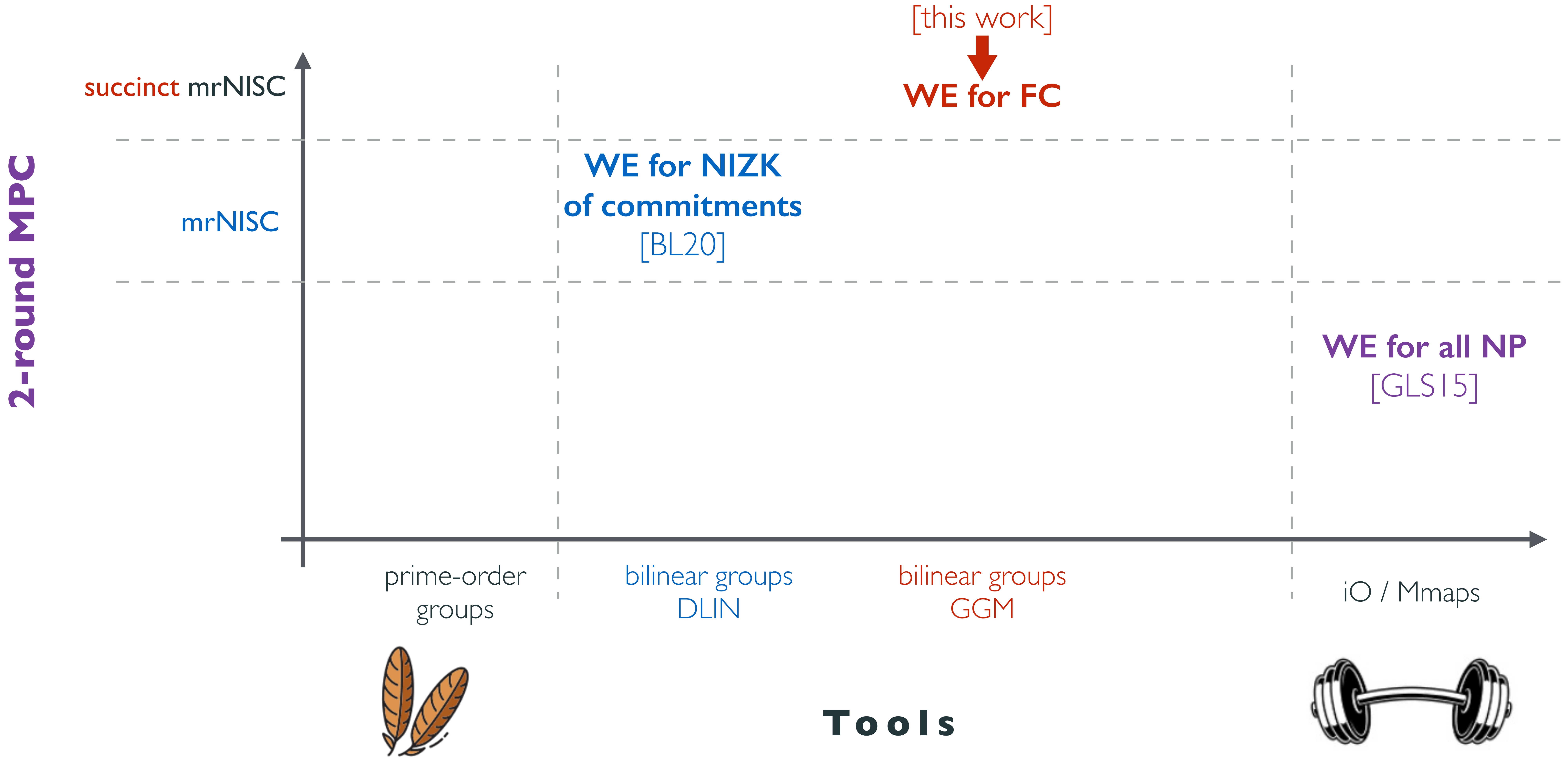
Our motivating application: 2-round MPC



Our motivating application: 2-round MPC



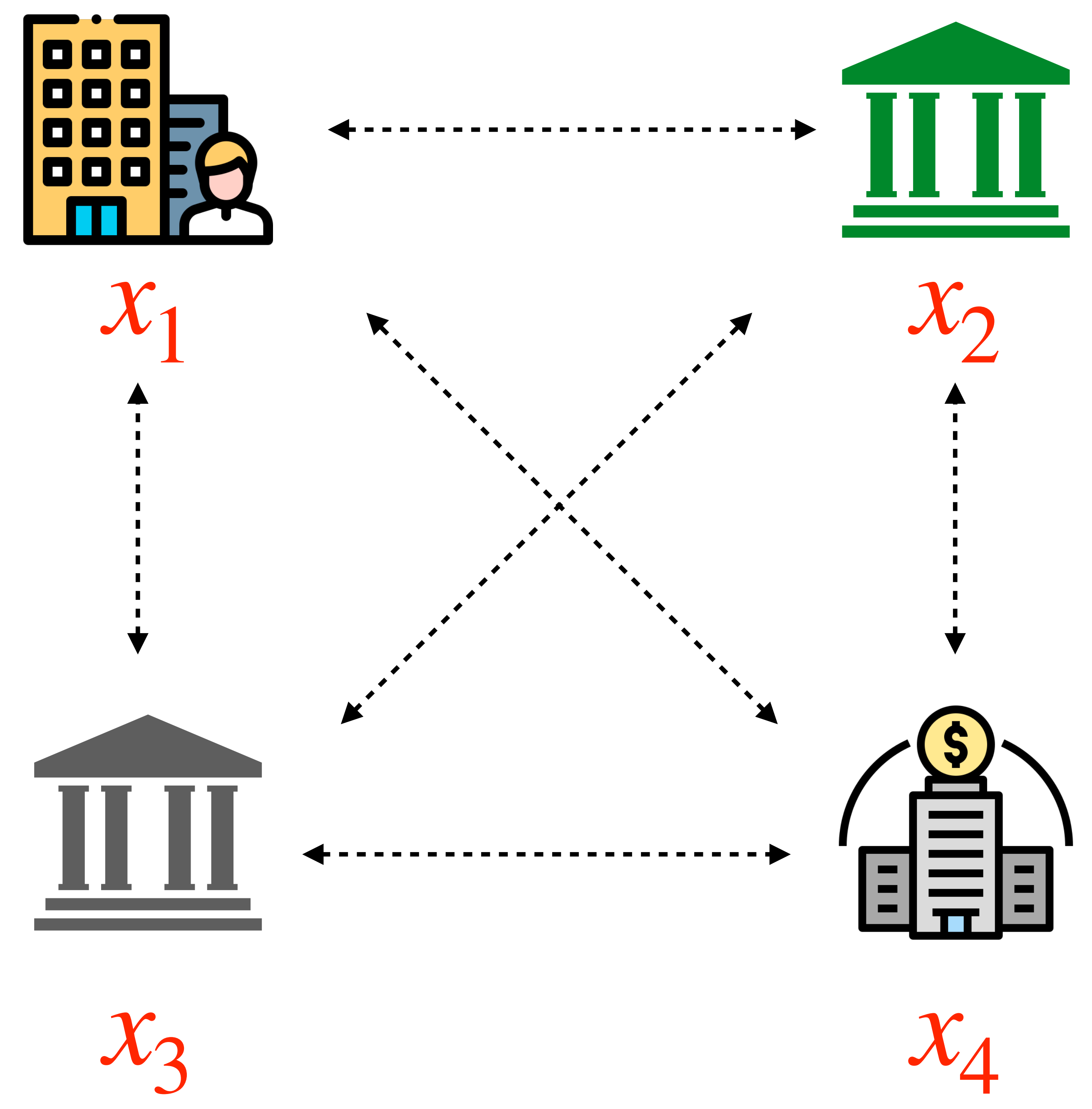
Our motivating application: 2-round MPC



Recap of MPC

Goals

- Preserve privacy of parties' inputs
- Guarantee correctness of computation



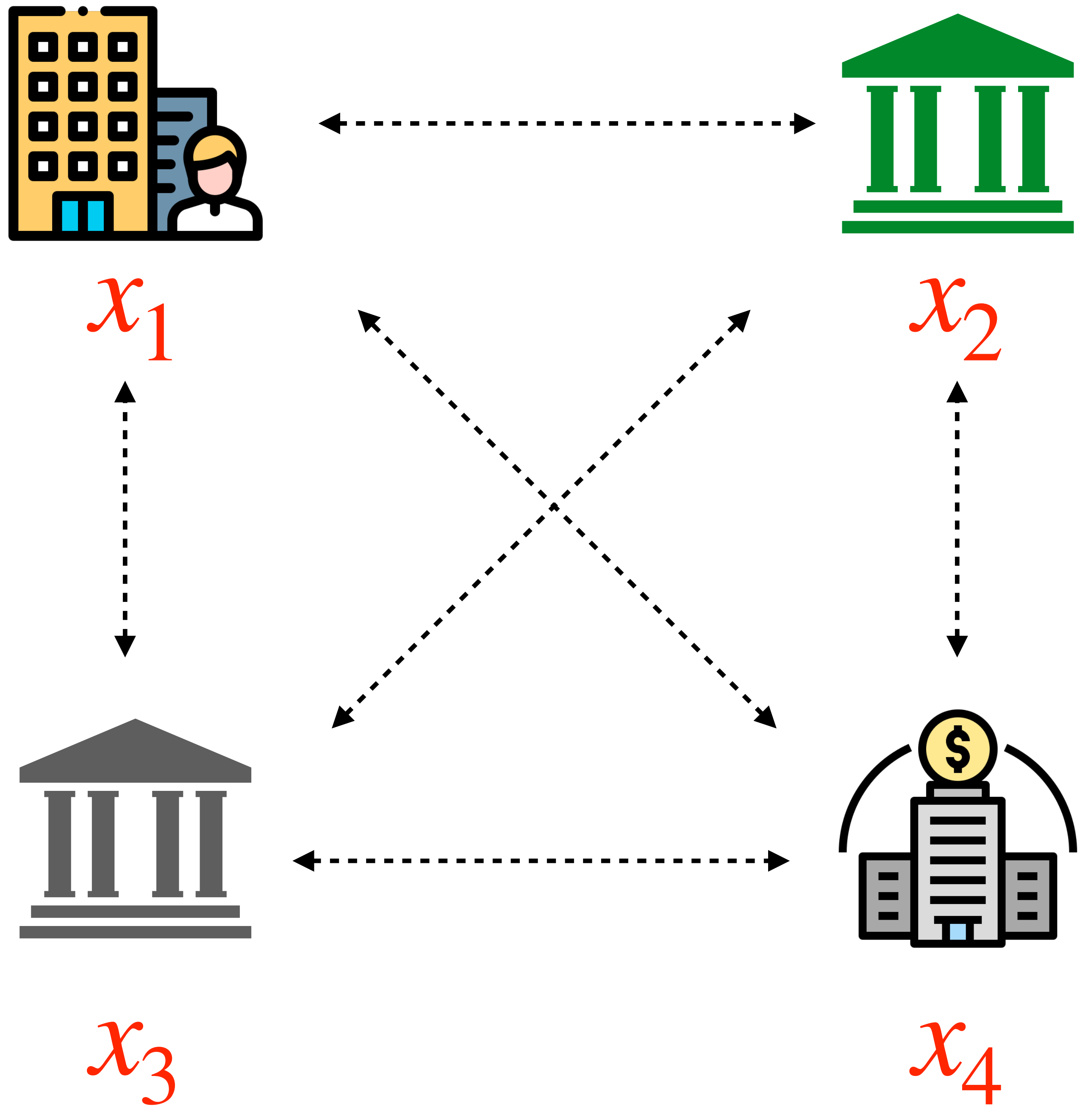
$$F(x_1, x_2, x_3, x_4) = y = (y_1, y_2, y_3, y_4)$$

Recap of MPC

Goals

- Preserve privacy of parties' inputs
- Guarantee correctness of computation

Round complexity can be high!

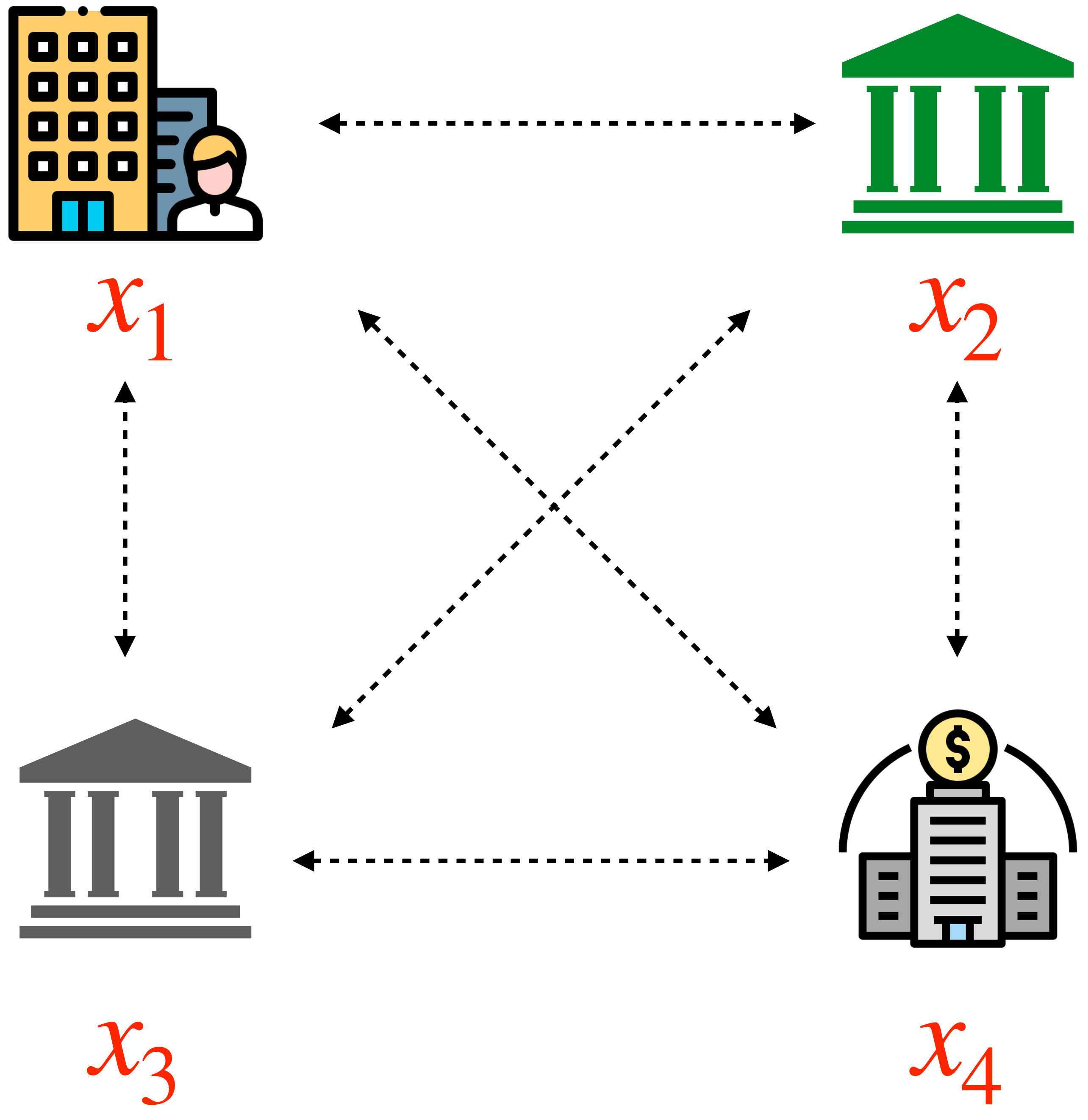


$$F(x_1, x_2, x_3, x_4) = y = (y_1, y_2, y_3, y_4)$$

2-round MPC

Round-collapsing (n-round) → (2-round)

using iO [GGHR14] — using WE for all NP [GLS15]



$$F(x_1, x_2, x_3, x_4) = y = (y_1, y_2, y_3, y_4)$$

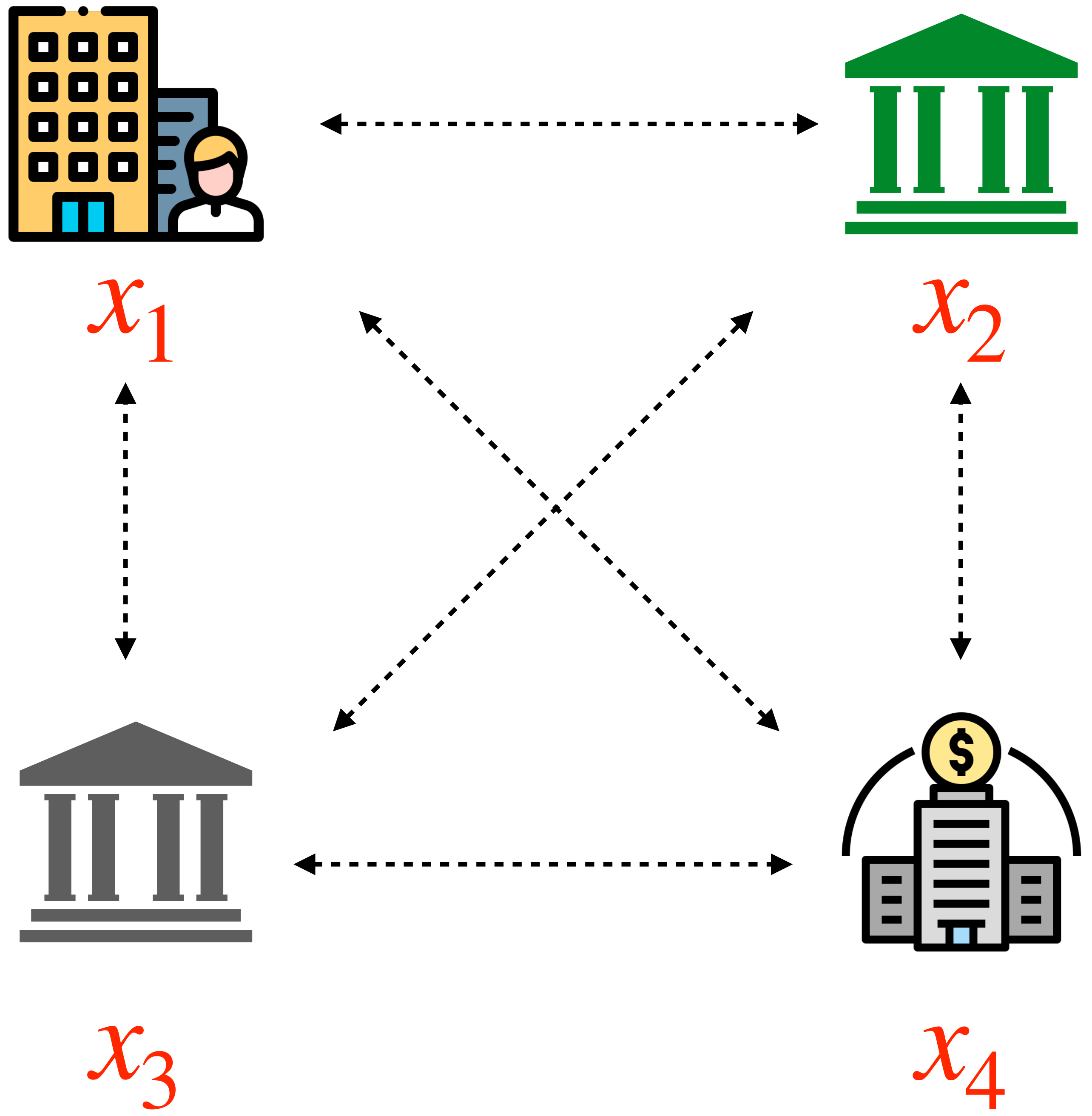
2-round MPC

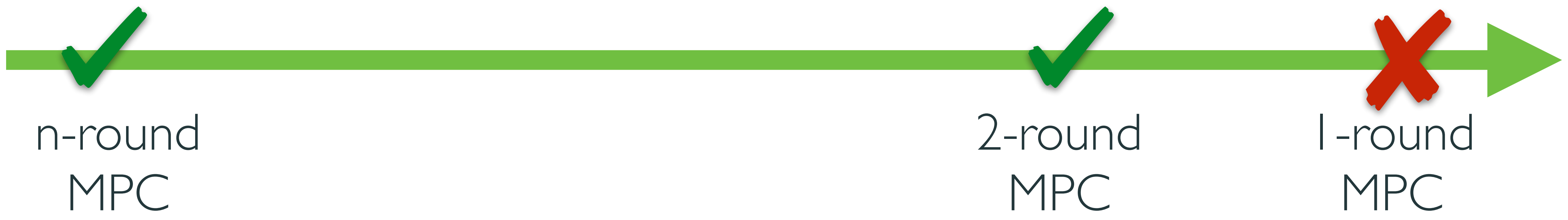
Round-collapsing (n-round) → (2-round)

using iO [GGHR14] — using WE for all NP [GLS15]

Can we reduce further?

Not really! Due to residual attacks





$$F(x_1, x_2, x_3, x_4) = y = (y_1, y_2, y_3, y_4)$$

2-round MPC

Round-collapsing (n-round) → (2-round)

using iO [GGHR14] — using WE for all NP [GLS15]

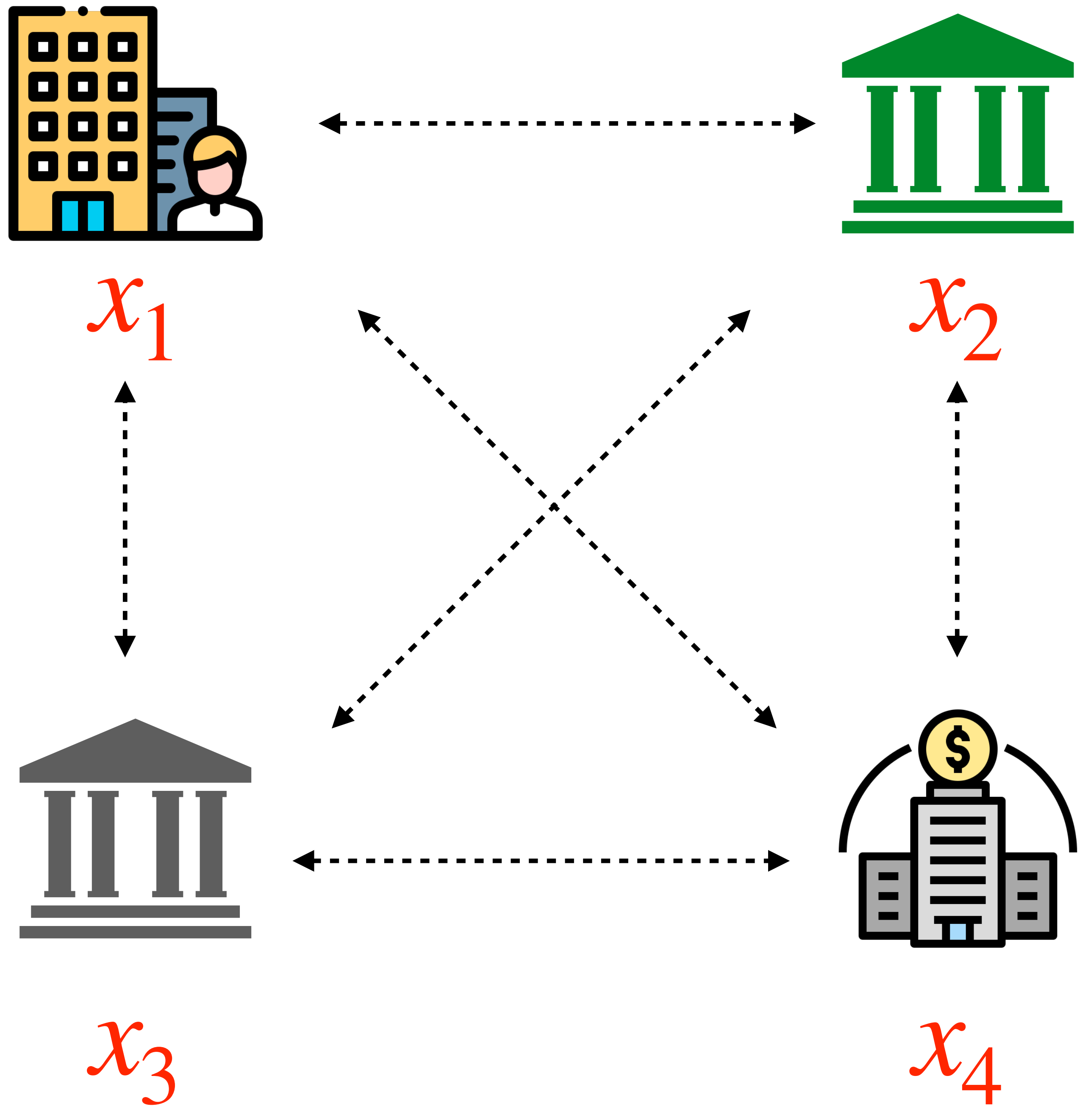
Can we reduce further?

Not really! Due to residual attacks

multiparty reusable Non-Interactive Secure Computation

(mrNISC)

2-round MPC with reusable 1st round [BL20]



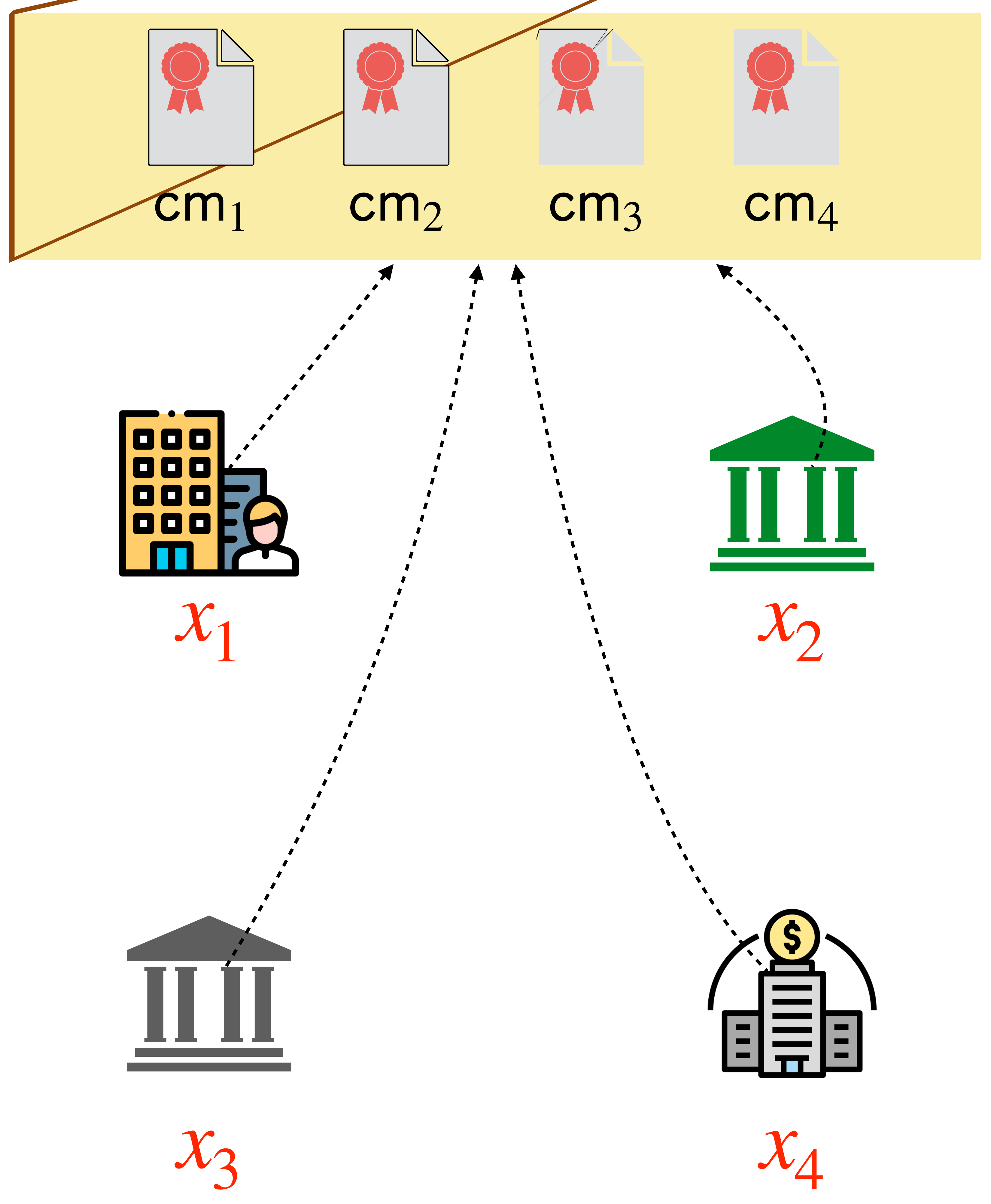
$$F(x_1, x_2, x_3, x_4) = y = (y_1, y_2, y_3, y_4)$$

mrNISC

[BL20]

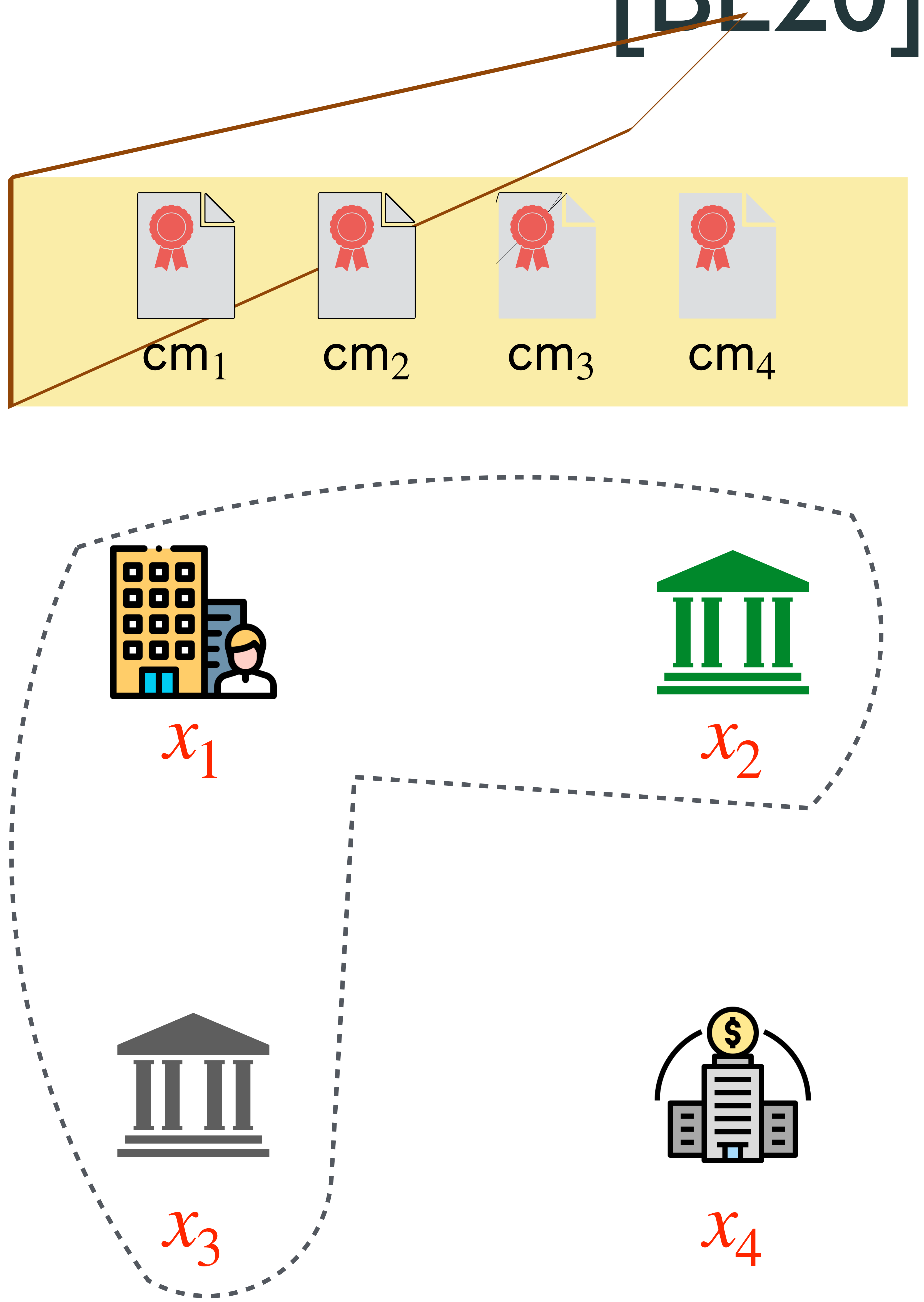
<u>**Round 1:**</u> commit to inputs x_i in a bulletin board

$$\mathsf{cm}_i = Com(x_i; r_i)$$



mrNISC

[BL20]



Round 1: commit to inputs x_i in a bulletin board

$$\mathsf{cm}_i = Com(x_i; r_i)$$

Round 2: to compute $F(\{x_j\}_{j \in S})$ broadcast

$$\alpha_i = Encode(F, \{\mathsf{cm}_j\}_{j \in S}, (x_i; r_i)\ )$$

mrNISC

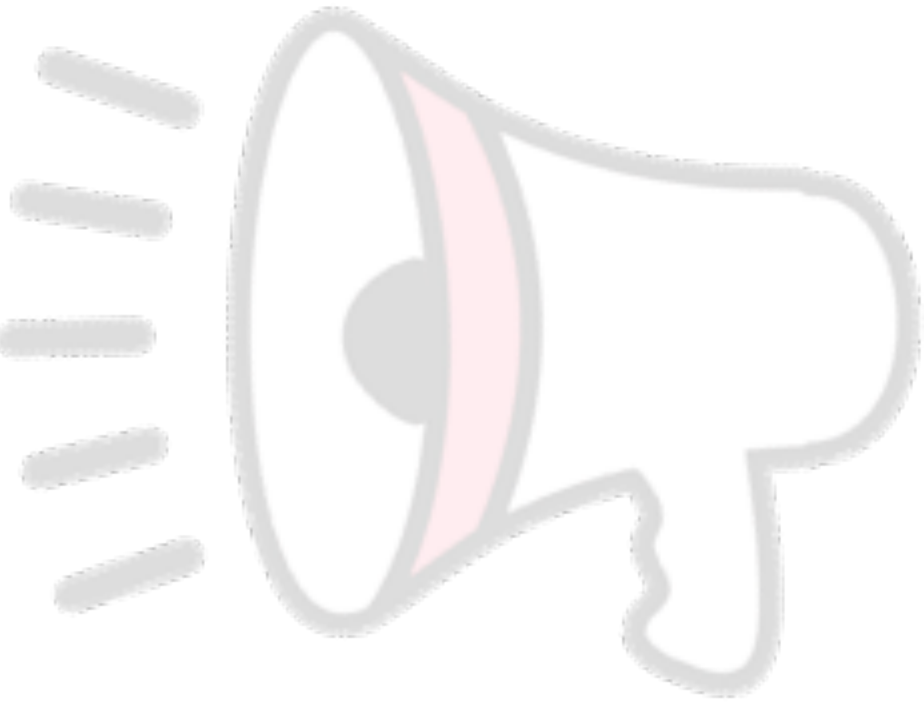
[BL20]

Round 1: commit to inputs x_i in a bulletin board

$$\mathsf{cm}_i = Com(x_i; r_i)$$

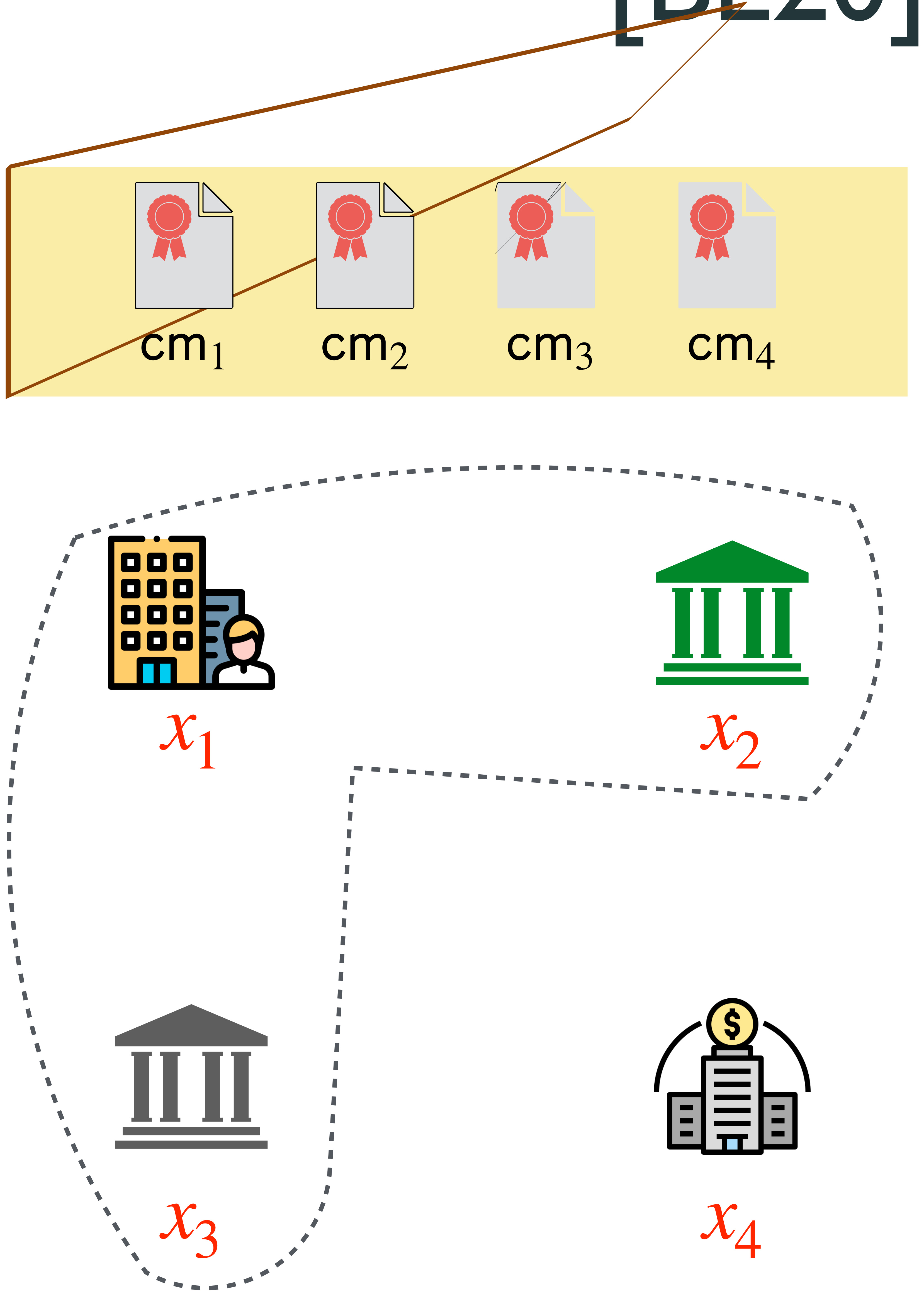
Round 2: to compute $F(\{x_j\}_{j\in S})$ broadcast

α_1
α_2
α_3



$$\alpha_i = Encode(F, \{\mathsf{cm}_j\}_{j\in S}, (x_i; r_i)\ )$$

Output: locally compute $y = Eval(F, \{\mathsf{cm}_j, \alpha_j\}_{j\in S})$

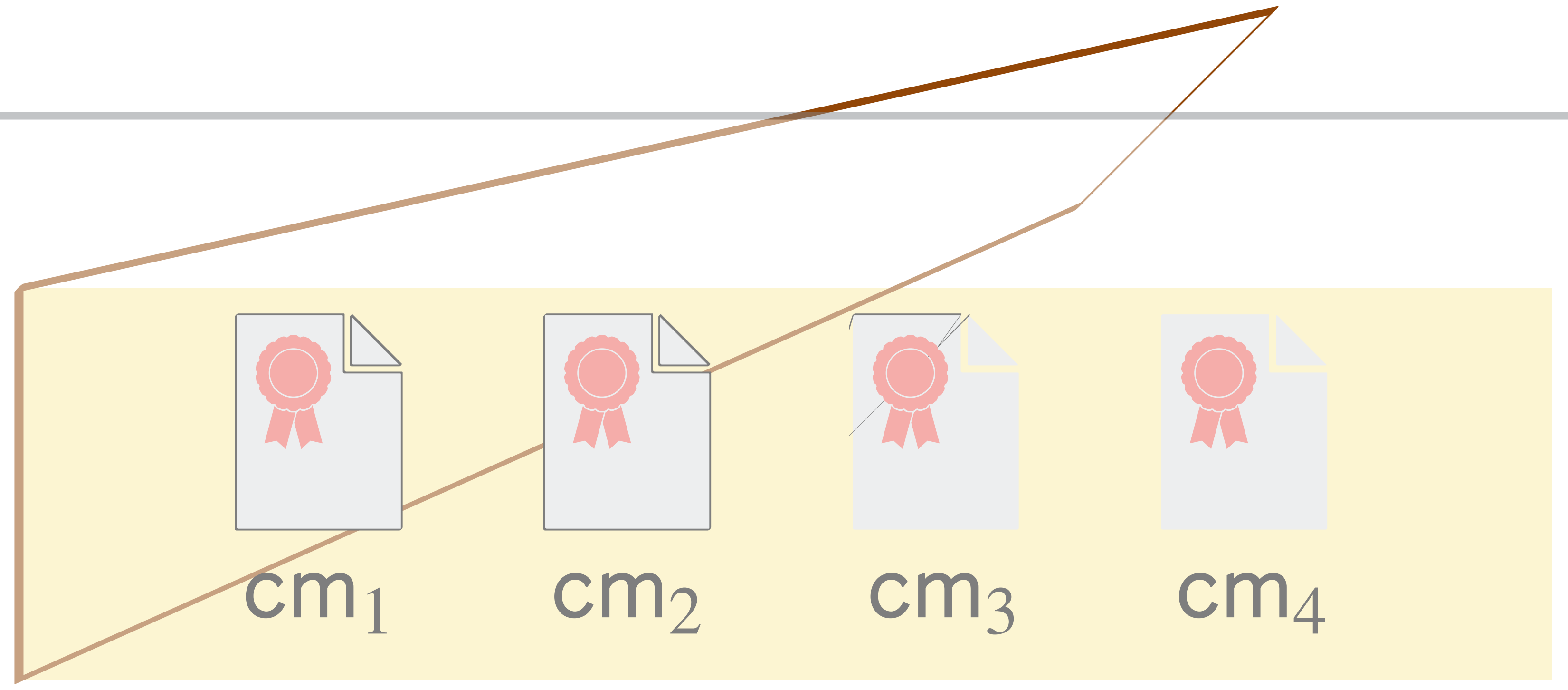


Reusability

Fixed

Round 1: commit to inputs x_i in a bulletin board

$$\mathsf{cm}_i = Com(x_i; r_i)$$

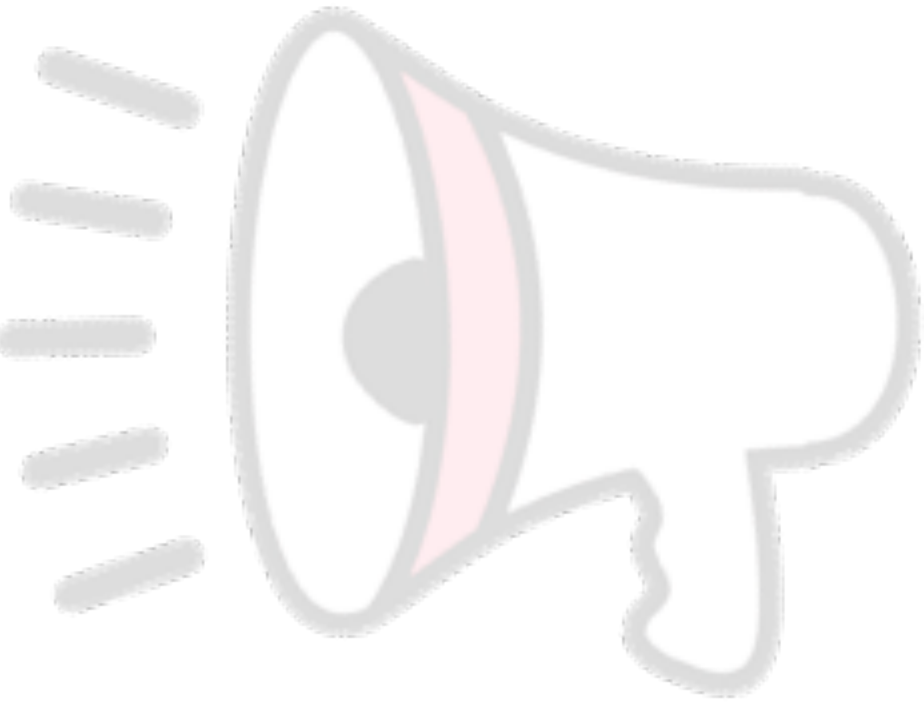


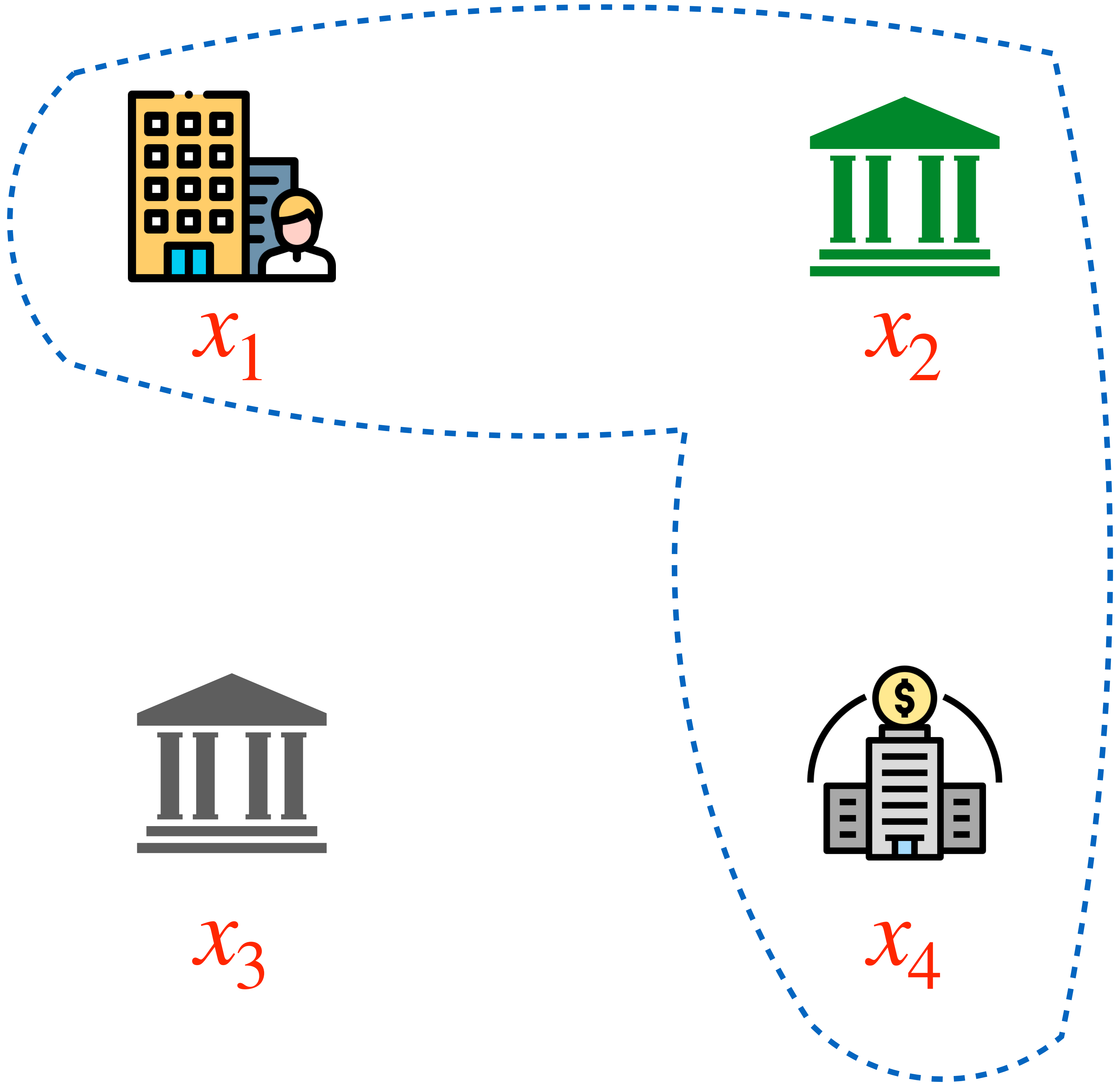
Round 2: to compute $F'(\{x_j\}_{j\in S'})$ broadcast

α'_1

α'_2

α'_4





$$\alpha'_i = Encode(F', \{\mathsf{cm}_j\}_{j\in S'}, (x_i; r_i)\ )$$

Output: locally compute $y' = Eval(F', \{\mathsf{cm}_j, \alpha'_j\}_{j\in S'})$

mrNISC construction of [BL20]

Use [GLS15] round-collapsing with a <u>weaker variant of WE</u>

WE for NIZK of Commitments (WE-NIZK)

a WE for $L = \left\{(\mathsf{cm}, G, y) : \exists\, \mathbf{x} \text{ and NIZK } \pi \text{ for "} y = G(\mathbf{x}) \wedge \mathsf{cm} = \mathsf{Com}(\mathbf{x})\text{"}\right\}$

mrNISC construction of [BL20]

Use [GLS15] round-collapsing with a weaker variant of WE

WE for NIZK of Commitments (WE-NIZK)

a WE for $L = \{(\mathsf{cm}, G, y) : \exists\, \mathbf{x}$ and NIZK π for "$y = G(\mathbf{x}) \wedge \mathsf{cm} = \mathsf{Com}(\mathbf{x})$"$\}$

[BL20] realized it from DLIN over bilinear groups

(Variant) Groth-Sahai coms&proofs w/linear verification + SPHF → WE-NIZK

mrNISC construction of [BL20]

Use [GLS15] round-collapsing with a weaker variant of WE

WE for NIZK of Commitments (WE-NIZK)

a WE for $L = \left\{(\mathsf{cm}, G, y) : \exists\, \mathbf{x} \text{ and NIZK } \pi \text{ for "} y = G(\mathbf{x}) \wedge \mathsf{cm} = \mathsf{Com}(\mathbf{x}) \text{"}\right\}$

[BL20] realized it from DLIN over bilinear groups

(Variant) Groth-Sahai coms&proofs w/linear verification + SPHF → WE-NIZK

Efficiency of [BL20] WE-NIZK ⬅ **Our focus**

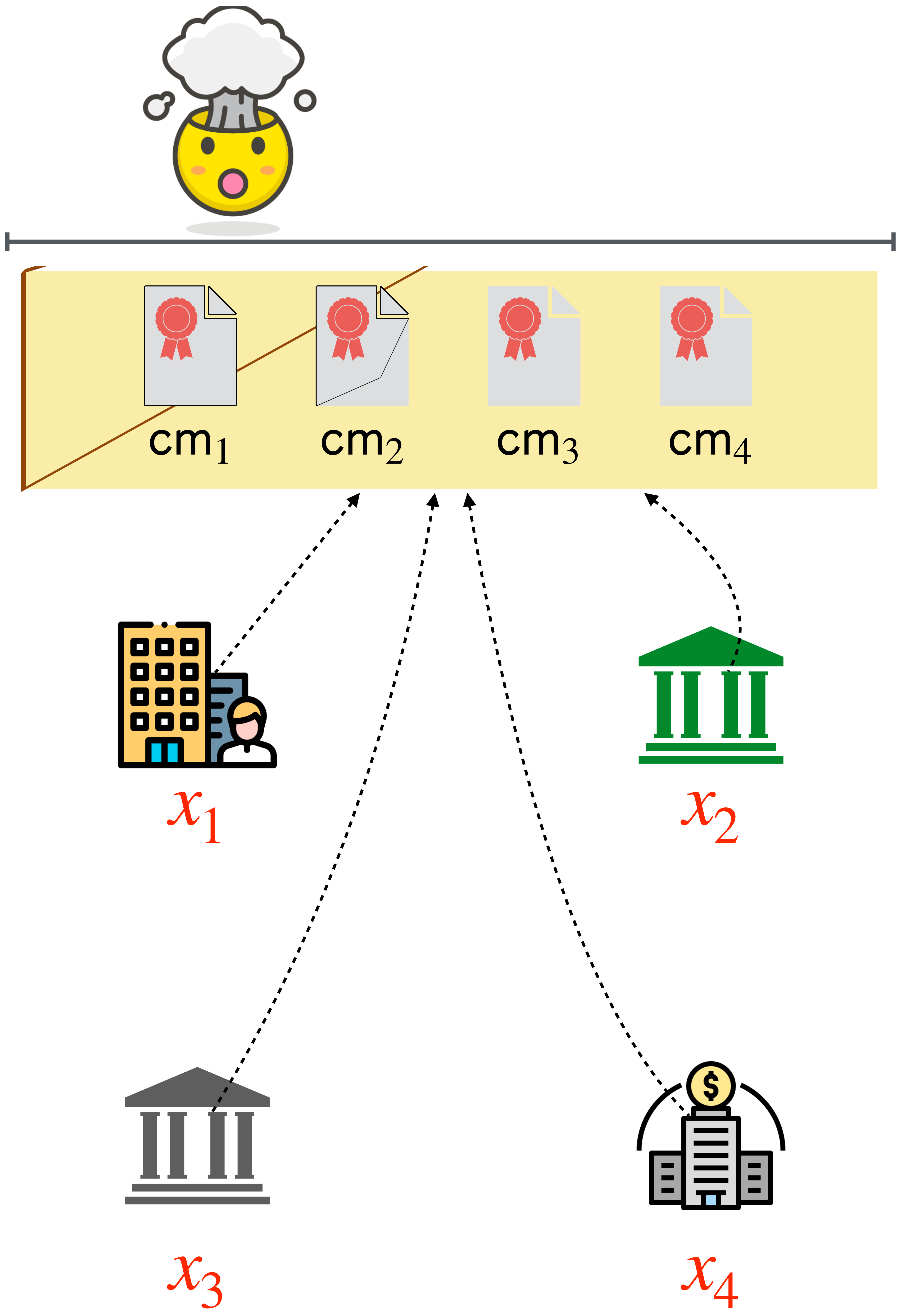
Requires statistically binding commitments ⇒ commitments are large $O(|\mathbf{x}|)$

Requires statistically sound NIZKs ⇒ WE decryption time $O(|\mathbf{x}|)$

Impact of WE-NIZK in mrNISC

Bulletin board grows with data size…

$$|BB| = \sum_i |\mathsf{cm}_i| \geq \sum_i |\mathbf{x}_i|$$

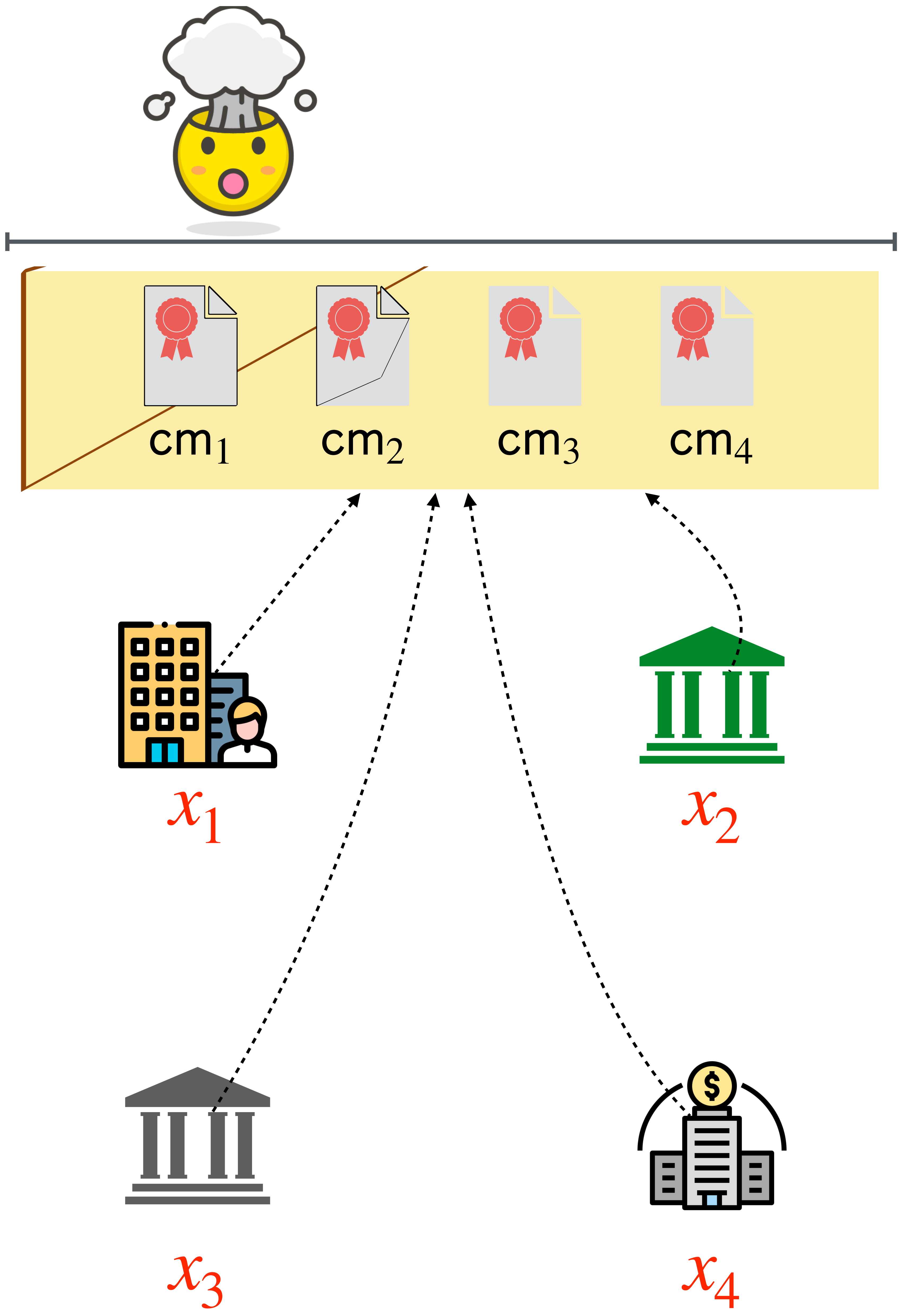


Impact of WE-NIZK in mrNISC

Bulletin board grows with data size…

$$|BB| = \sum_i |\mathsf{cm}_i| \geq \sum_i |\mathbf{x}_i|$$

Can we have a succinct Round 1 (and BB)?



Impact of WE-NIZK in mrNISC

Bulletin board grows with data size…

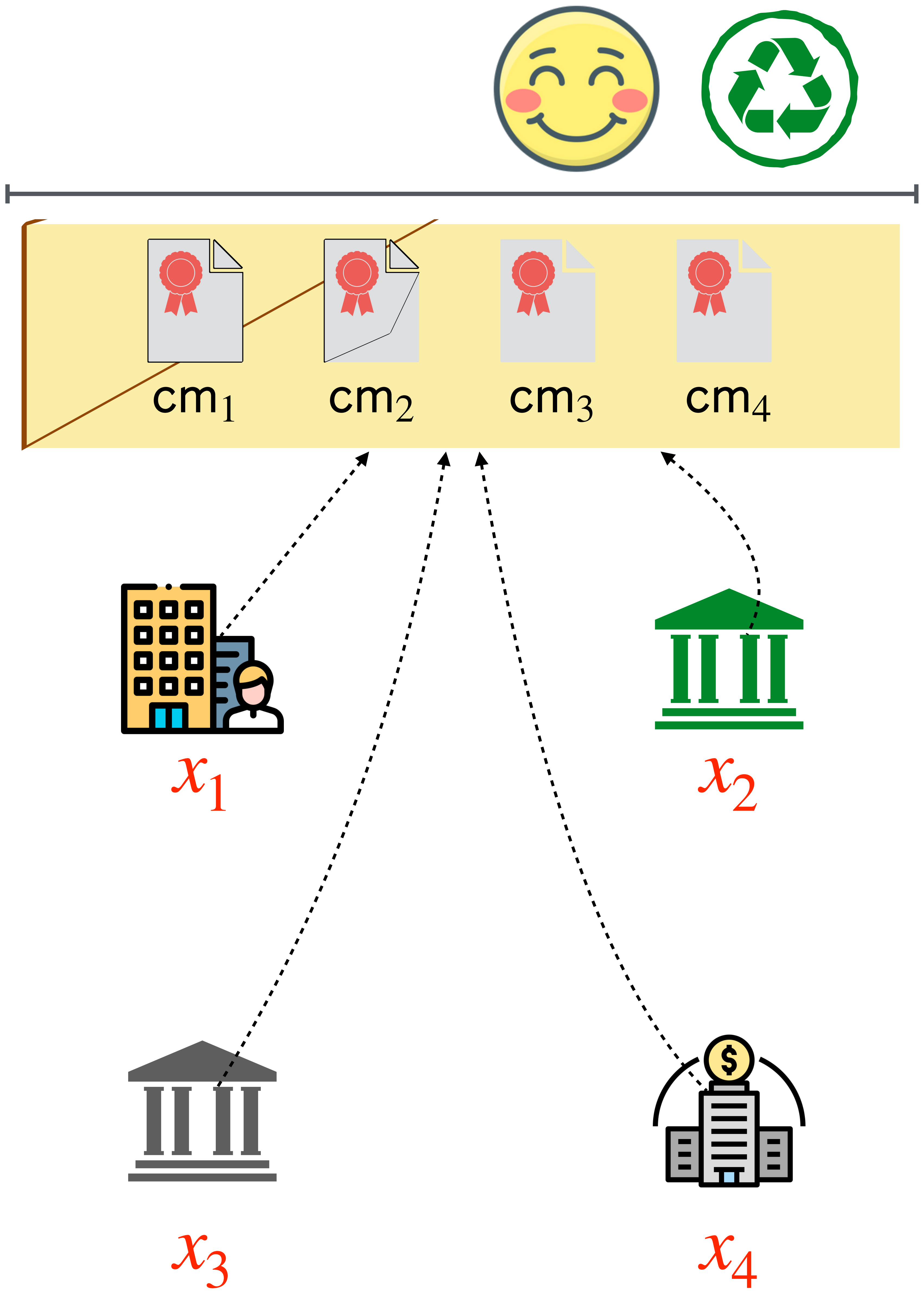
$$|BB| = \sum_i |\mathsf{cm}_i| \geq \sum_i |\mathbf{x}_i|$$

Can we have a succinct Round 1 (and BB)?

Our solution

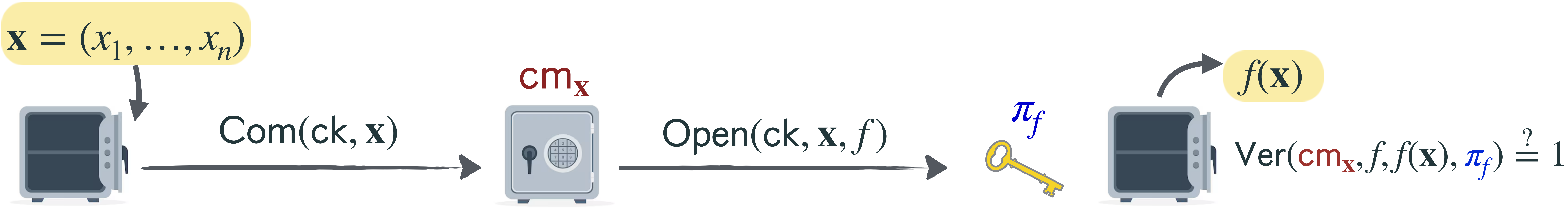
WE-FC: WE for *succinct* Functional Commitments

$$|BB| = \sum_i |\mathsf{cm}_i| = n \cdot p(\lambda)$$



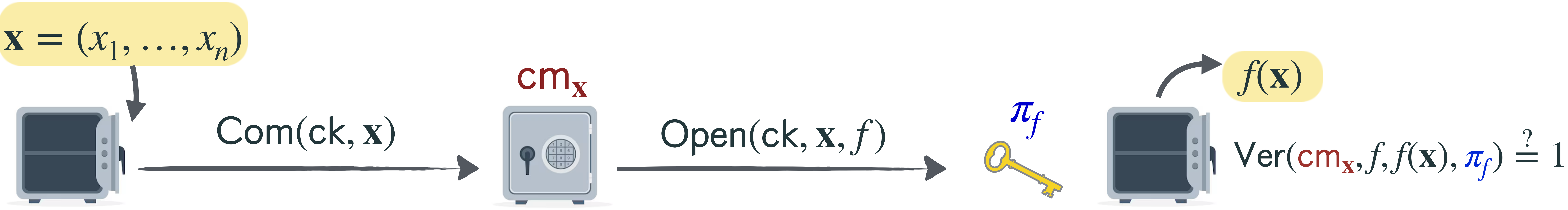
Functional Commitments

[LRY16]



Functional Commitments

[LRY16]



Commitments and openings are "short"

 Short commitments $|\mathsf{cm}_{\mathbf{x}}| \leq p(\lambda, \log|\mathbf{x}|)$

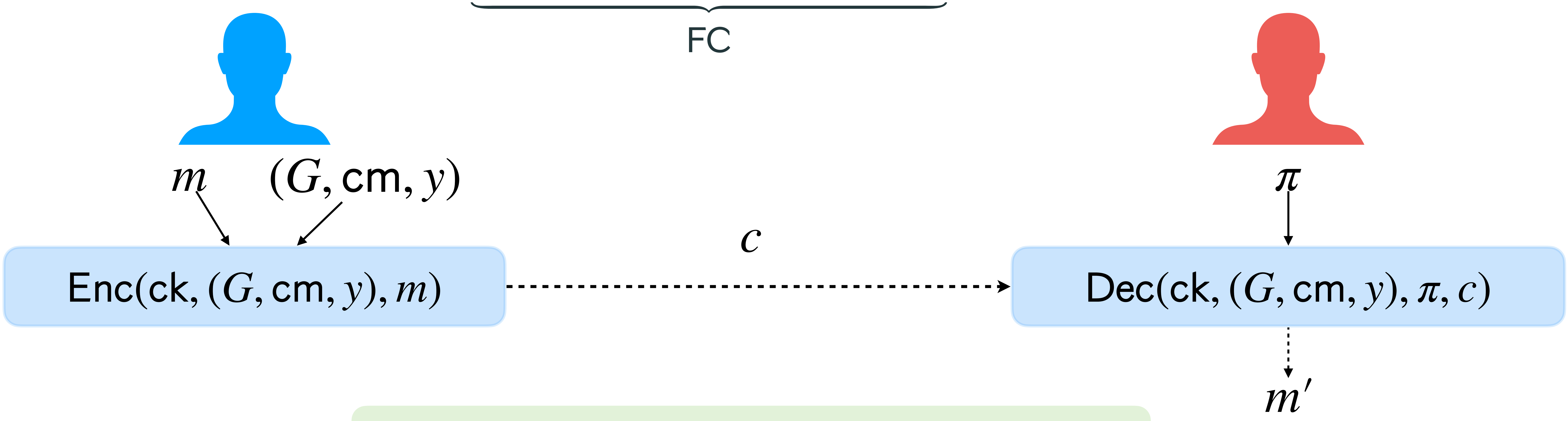
 Short openings: $|\pi_f| \leq p(\lambda, \log|\mathbf{x}|)$

Security (Evaluation binding): hard to open $\mathsf{cm}_{\mathbf{x}}$ to two different outputs for the same f

WE for FCs

Main idea: encrypt a message w.r.t. who holds an FC opening to $G(\mathbf{x})$

$$\underbrace{(\mathsf{Setup}, \mathsf{Com}, \mathsf{Open}, \mathsf{Ver}}_{\mathsf{FC}}, \mathsf{Enc}, \mathsf{Dec})$$



Correctness

If $\mathsf{Ver}(\mathsf{ck}, G, \mathsf{cm}, y, \pi) = 1$ then $m' = m$

Security

If $\mathsf{cm} = \mathsf{Com}(\mathbf{x}) \wedge y \neq G(\mathbf{x})$ then c leaks no information on m

Our Contributions

- **Definition of WE-FC** compared to [BL20] we deal with computational binding/soundness

Our Contributions

- **Definition of WE-FC** compared to [BL20] we deal with computational binding/soundness
- **Generic construction of WE-FC:** FC with linear verification + EPHF (new notion)

Our Contributions

- **Definition of WE-FC** compared to [BL20] we deal with computational binding/soundness
- **Generic construction of WE-FC:** FC with linear verification + EPHF (new notion)
- **EPHF construction** under discrete log in the AGM

Our Contributions

- **Definition of WE-FC** compared to [BL20] we deal with computational binding/soundness
- **Generic construction of WE-FC:** FC with linear verification + EPHF (new notion)
- **EPHF construction** under discrete log in the AGM
- **New FC for NC1 with linear verification** under QP-BDHE (falsifiable) assumption

Verification				
quadratic Ver			**CFT22**	**BCFL23**
linear Ver	**LRY16, LM19**	**LP20**	**Ours**	
	linear maps	semi-sparse poly	NC1	all circuits

Functions

Our Contributions

- **Definition of WE-FC** compared to [BL20] we deal with computational binding/soundness
- **Generic construction of WE-FC:** FC with linear verification + EPHF (new notion)
- **EPHF construction** under discrete log in the AGM
- **New FC for NC1 with linear verification** under QP-BDHE (falsifiable) assumption

Verification				
quadratic Ver			CFT22	BCFL23
linear Ver	LRY16, LM19	LP20	**Ours**	
	linear maps	semi-sparse poly	NC1	all circuits
				Functions

- **Applications to *succinct* mrNISC, targeted broadcast, contingent payments**

Our Contributions

- **Definition of WE-FC** compared to [BL20] we deal with computational binding/soundness
- **Generic construction of WE-FC:** FC with linear verification + EPHF (new notion)

This talk

- **EPHF construction** under discrete log in the AGM
- **New FC for NC1 with linear verification** under QP-BDHE (falsifiable) assumption

Verification				
quadratic Ver			CFT22	BCFL23
linear Ver	LRY16, LM19	LP20	Ours	
	linear maps	semi-sparse poly	NC1	all circuits
				Functions

- **Applications to *succinct* mrNISC, targeted broadcast, contingent payments**

Our Generic Construction

FCs w/ linear verification
in bilinear groups

$\mathsf{Ver}(\mathsf{ck}, G, \mathsf{cm}, y, \vec{\pi})$:

$$[\mathbf{\Theta}]_T \stackrel{?}{=} [\mathbf{M}]_1 \cdot \vec{\pi}$$

Our Generic Construction

FCs w/ linear verification
in bilinear groups

+

EPHF for linear eq.
Extractable Projective
Hash Functions

$\mathsf{Ver}(\mathsf{ck}, G, \mathsf{cm}, y, \vec{\pi})$:

$$[\boldsymbol{\Theta}]_T \stackrel{?}{=} [\mathbf{M}]_1 \cdot \vec{\pi}$$

Our Generic Construction

FCs w/ linear verification
in bilinear groups

+

EPHF for linear eq.
Extractable Projective
Hash Functions

$\mathsf{Ver}(\mathsf{ck}, G, \mathsf{cm}, y, \vec{\pi})$:

$$[\boldsymbol{\Theta}]_T \stackrel{?}{=} [\mathbf{M}]_1 \cdot \vec{\pi}$$

$\mathsf{ProjKG}\left(\mathsf{stmt} = ([\boldsymbol{\Theta}]_T, [\mathbf{M}]_1)\right)$

hk

hp

$\mathsf{Hash}(\mathsf{hk}, \mathsf{stmt})$

$\mathsf{ProjHash}(\mathsf{hp}, \mathsf{stmt}, \vec{\pi})$

Our Generic Construction

FCs w/ linear verification
in bilinear groups

+

EPHF for linear eq.
Extractable Projective
Hash Functions

$\mathsf{Ver}(\mathsf{ck}, G, \mathsf{cm}, y, \vec{\pi})$:

$$[\boldsymbol{\Theta}]_T \stackrel{?}{=} [\mathbf{M}]_1 \cdot \vec{\pi}$$

$\mathsf{ProjKG}\left(\mathsf{stmt} = ([\boldsymbol{\Theta}]_T, [\mathbf{M}]_1)\right)$

hk

hp

Correctness

If $[\boldsymbol{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

$\mathsf{Hash}(\mathsf{hk}, \mathsf{stmt})$ = $\mathsf{ProjHash}(\mathsf{hp}, \mathsf{stmt}, \vec{\pi})$

Our Generic Construction

FCs w/ linear verification in bilinear groups

+

EPHF for linear eq.
Extractable Projective Hash Functions

$\mathsf{Ver}(\mathsf{ck}, G, \mathsf{cm}, y, \vec{\pi})$:

$$[\boldsymbol{\Theta}]_T \stackrel{?}{=} [\mathbf{M}]_1 \cdot \vec{\pi}$$

$\mathsf{ProjKG}\left(\mathsf{stmt} = ([\boldsymbol{\Theta}]_T, [\mathbf{M}]_1)\right)$

hk

hp

Correctness

If $[\boldsymbol{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

$\mathsf{Hash}(\mathsf{hk}, \mathsf{stmt})$ = $\mathsf{ProjHash}(\mathsf{hp}, \mathsf{stmt}, \vec{\pi})$

Knowledge Smoothness

$\forall$PPT $\mathscr{A}(\mathsf{hp})$ producing (stmt, H) s.t. $H = \mathsf{Hash}(\mathsf{hk}, \mathsf{stmt})$

$\exists \mathscr{E}(\mathsf{hp}) \rightarrow \vec{\pi}$ s.t. $[\boldsymbol{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

Our Generic Construction

FCs w/ linear verification in bilinear groups + **EPHF for linear eq.** Extractable Projective Hash Functions → **WE-FC** for bits

Ver(ck, G, cm, y, $\vec{\pi}$) :

$$[\boldsymbol{\Theta}]_T \stackrel{?}{=} [\mathbf{M}]_1 \cdot \vec{\pi}$$

ProjKG $\left(\text{stmt} = ([\boldsymbol{\Theta}]_T, [\mathbf{M}]_1)\right)$

hk hp

Correctness

If $[\boldsymbol{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

Hash(hk, stmt) = ProjHash(hp, stmt, $\vec{\pi}$)

Knowledge Smoothness

$\forall$PPT $\mathscr{A}$(hp) producing (stmt, H) s.t. H = Hash(hk, stmt)

$\exists \mathscr{E}$(hp) $\rightarrow \vec{\pi}$ s.t. $[\boldsymbol{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

Our Generic Construction

FCs w/ linear verification in bilinear groups + **EPHF for linear eq.** Extractable Projective Hash Functions → **WE-FC** for bits

$\mathsf{Ver}(\mathsf{ck}, G, \mathsf{cm}, y, \vec{\pi})$:

$$[\boldsymbol{\Theta}]_T \stackrel{?}{=} [\mathbf{M}]_1 \cdot \vec{\pi}$$

$\mathsf{ProjKG}\left(\mathsf{stmt} = ([\boldsymbol{\Theta}]_T, [\mathbf{M}]_1)\right)$

hk   hp

Correctness

If $[\boldsymbol{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

$\mathsf{Hash}(\mathsf{hk}, \mathsf{stmt})$ = $\mathsf{ProjHash}(\mathsf{hp}, \mathsf{stmt}, \vec{\pi})$

$\mathsf{Enc}(\mathsf{ck}, (G, \mathsf{cm}, y), m)$

// get $[\boldsymbol{\Theta}]_{\mathbf{T}}, [\mathbf{M}]_{\mathbf{1}}$ from (G, cm, y)

$\mathsf{hk}, \mathsf{hp} \leftarrow \mathsf{ProjKG}\left([\boldsymbol{\Theta}]_T, [\mathbf{M}]_1\right)$

$H \leftarrow \mathsf{Hash}(\mathsf{hk}, [\boldsymbol{\Theta}]_{\mathrm{T}}, [\mathbf{M}]_1)$

$r \xleftarrow{\$} \{0,1\}^{|H|}$

Return $c = (\mathsf{hp}, r, \hat{c} = \langle H, r\rangle \oplus m)$

Knowledge Smoothness

$\forall$PPT $\mathscr{A}(\mathsf{hp})$ producing (stmt, H) s.t. $H = \mathsf{Hash}(\mathsf{hk}, \mathsf{stmt})$

$\exists \mathscr{E}(\mathsf{hp}) \rightarrow \vec{\pi}$ s.t. $[\boldsymbol{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

Our Generic Construction

FCs w/ linear verification in bilinear groups

+

EPHF for linear eq.
Extractable Projective Hash Functions

→

WE-FC
for bits

$\mathsf{Ver}(\mathsf{ck}, G, \mathsf{cm}, y, \vec{\pi})$:

$$[\mathbf{\Theta}]_T \stackrel{?}{=} [\mathbf{M}]_1 \cdot \vec{\pi}$$

$\mathsf{ProjKG}\left(\mathsf{stmt} = ([\mathbf{\Theta}]_T, [\mathbf{M}]_1)\right)$

hk   hp

Correctness
If $[\mathbf{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

$\mathsf{Hash}(\mathsf{hk}, \mathsf{stmt})$ = $\mathsf{ProjHash}(\mathsf{hp}, \mathsf{stmt}, \vec{\pi})$

Knowledge Smoothness

$\forall \mathrm{PPT}\ \mathscr{A}(\mathsf{hp})$ producing (stmt, H) s.t. $H = \mathsf{Hash}(\mathsf{hk}, \mathsf{stmt})$

$\exists \mathscr{E}(\mathsf{hp}) \rightarrow \vec{\pi}$ s.t. $[\mathbf{\Theta}]_T = [\mathbf{M}]_1 \cdot \vec{\pi}$

$\mathsf{Enc}(\mathsf{ck}, (G, \mathsf{cm}, y), m)$

// get $[\mathbf{\Theta}]_{\mathbf{T}}, [\mathbf{M}]_{\mathbf{1}}$ from (G, cm, y)

$\mathsf{hk}, \mathsf{hp} \leftarrow \mathsf{ProjKG}\left([\mathbf{\Theta}]_T, [\mathbf{M}]_1\right)$

$H \leftarrow \mathsf{Hash}(\mathsf{hk}, [\mathbf{\Theta}]_{\mathrm{T}}, [\mathbf{M}]_1)$

$r \xleftarrow{\$} \{0,1\}^{|H|}$

Return $c = (\mathsf{hp}, r, \hat{c} = \langle H, r \rangle \oplus m)$

$\mathsf{Dec}(\mathsf{ck}, (G, \mathsf{cm}, y), c, \vec{\pi})$

$H \leftarrow \mathsf{ProjHash}(\mathsf{hp}, [\mathbf{\Theta}]_{\mathrm{T}}, [\mathbf{M}]_1, \vec{\pi})$

$m' \leftarrow \langle H, r \rangle \oplus \hat{c}$

Conclusion and open problems

New WE notion: realization from simple tools + applications (w/succinctness)

Open problems:

- **Avoiding Goldreich-Levin technique** $\implies$ efficiency + algebraic reduction
- **WE-FC for circuits**
- **Standard assumptions**
- **More applications** e.g., [FKdP23] use special case (WE for VC) to build RBE

Conclusion and open problems

New WE notion: realization from simple tools + applications (w/succinctness)

Open problems:

- **Avoiding Goldreich-Levin technique** $\Longrightarrow$ efficiency + algebraic reduction
- **WE-FC for circuits**
- **Standard assumptions**
- **More applications** e.g., [FKdP23] use special case (WE for VC) to build RBE

Thank you!

Questions?

ePrint
ia.cr/2022/1510