Funded by the European Union

European Research Council
Established by the European Commission

Designated-Verifier SNARGs with One Group Element

Gal Arnon

Jesko Dujmovic

Yuval Ishai

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

אוניברסיטת בר-אילן
Bar-Ilan University

CISPA

UNIVERSITÄT DES SAARLANDES

TECHNION
Israel Institute of Technology

aws

How Small Can Arguments Be?

... in the GGM

$|G|$ = group element, τ = statistical, λ = computational

How Small Can Arguments Be?

... in the GGM

	Size	Structure	priv coin/ pub coin	
Ours	1|G| + O(τ) bits	non-pairing group	priv	

|G| = group element, τ = statistical, λ = computational

How Small Can Arguments Be?

... in the GGM

	Size	Structure	priv coin/ pub coin	
Ours	1|G| + O(τ) bits	non-pairing group	priv	
[Kilian92]	Ω(τ)|G|	hash function	public	

|G| = group element, τ = statistical, λ = computational

How Small Can Arguments Be?

... in the GGM

	Size	Structure	priv coin/ pub coin	
Ours	1|G| + O(τ) bits	non-pairing group	priv	
[Kilian92]	Ω(τ)|G|	hash function	public	
[BIOW20]	O(1)|G|	non-pairing group	priv	1/poly soundness

$|G|$ = group element, τ = statistical, λ = computational

How Small Can Arguments Be?

... in the GGM

	Size	Structure	priv coin/ pub coin	
Ours	1|G| + O(τ) bits	non-pairing group	priv	
[Kilian92]	Ω(τ)|G|	hash function	public	
[BIOW20]	O(1)|G|	non-pairing group	priv	1/poly soundness
[BIOW20] amplified	ω(1)|G|	non-pairing group	priv	

|G| = group element, τ = statistical, λ = computational

How Small Can Arguments Be?

	Size	Structure	priv coin/ pub coin	
Ours	1|G| + O(τ) bits	non-pairing group	priv	
[Kilian92]	Ω(τ)|G|	hash function	public	
[BIOW20]	O(1)|G|	non-pairing group	priv	1/poly soundness
[BIOW20] amplified	ω(1)|G|	non-pairing group	priv	

|G| = group element, τ = statistical, λ = computational

How Small Can Arguments Be?

	Size	Structure	priv coin/ pub coin	
Ours	1|G| + O(τ) bits	non-pairing group	priv	
[Kilian92]	Ω(τ)|G|	hash function	public	
[BIOW20]	O(1)|G|	non-pairing group	priv	1/poly soundness
[BIOW20] amplified	ω(1)|G|	non-pairing group	priv	
DV-[SW14]	τ bits	obfuscation	priv	

$|G|$ = group element, τ = statistical, λ = computational

How Small Can Arguments Be?

	Size	Structure	priv coin/ pub coin	
Ours	1|G| + O(τ) bits	non-pairing group	priv	
[Kilian92]	Ω(τ)|G|	hash function	public	
[BIOW20]	O(1)|G|	non-pairing group	priv	1/poly soundness
[BIOW20] amplified	ω(1)|G|	non-pairing group	priv	
DV-[SW14]	τ bits	obfuscation	priv	
[Gro16,Lip24,DMS24]	10λ bits	pairing group	public	

$|G|$ = group element, τ = statistical, λ = computational

Designated Verifier SNARGs

Designated Verifier SNARGs

crs/pk

Setup()

sk

Designated Verifier SNARGs

Completeness: If $(x, w) \in R_L$

crs/pk

Setup()

sk

Designated Verifier SNARGs

Completeness: If $(x, w) \in R_L$

P(x,w)

crs/pk

Setup()

sk

Designated Verifier SNARGs

Completeness: If $(x, w) \in R_L$

P(x,w)

crs/pk

Setup()

π

sk

Designated Verifier SNARGs

Completeness: If $(x, w) \in R_L$

P(x,w)

crs/pk

Setup()

π

sk

V(x,π)=1

Designated Verifier SNARGs

Completeness: If $(x, w) \in R_L$

P(x,w)

crs/pk

Setup()

sk

π

V(x,π)=1

Succinctness: $|\pi| \ll |w|$

Designated Verifier SNARGs

Soundness: If $x \notin L$

crs/pk

Designated Verifier SNARGs

Soundness: If $x \notin L$

P*(x)

crs/pk

π

Designated Verifier SNARGs

Soundness: If $x \notin L$

crs/pk

π

P*(x)

V(x,π)=1 with prob. $2^{-\tau}$

Designated Verifier SNARGs

Soundness: If $x \notin L$

crs/pk

π

P*(x)

Application:

prover provides AI/DB service

V(x,π)=1 with prob. $2^{-\tau}$

DV-SNARGs Construction

[BCIOP13]-recipe: Linear-Only Encryption + 1-query LPCP = DV-SNARG

DV-SNARGs Construction

[BCIOP13]-recipe: Linear-Only Encryption + 1-query LPCP = DV-SNARG

DV-SNARGs Construction

[BCIOP13]-recipe: Linear-Only Encryption + 1-query LPCP = DV-SNARG

DV-SNARGs Construction

[BCIOP13]-recipe: Linear-Only Encryption + 1-query LPCP = DV-SNARG

Query()

$\vec{q} \in \mathbb{F}_p^n$

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

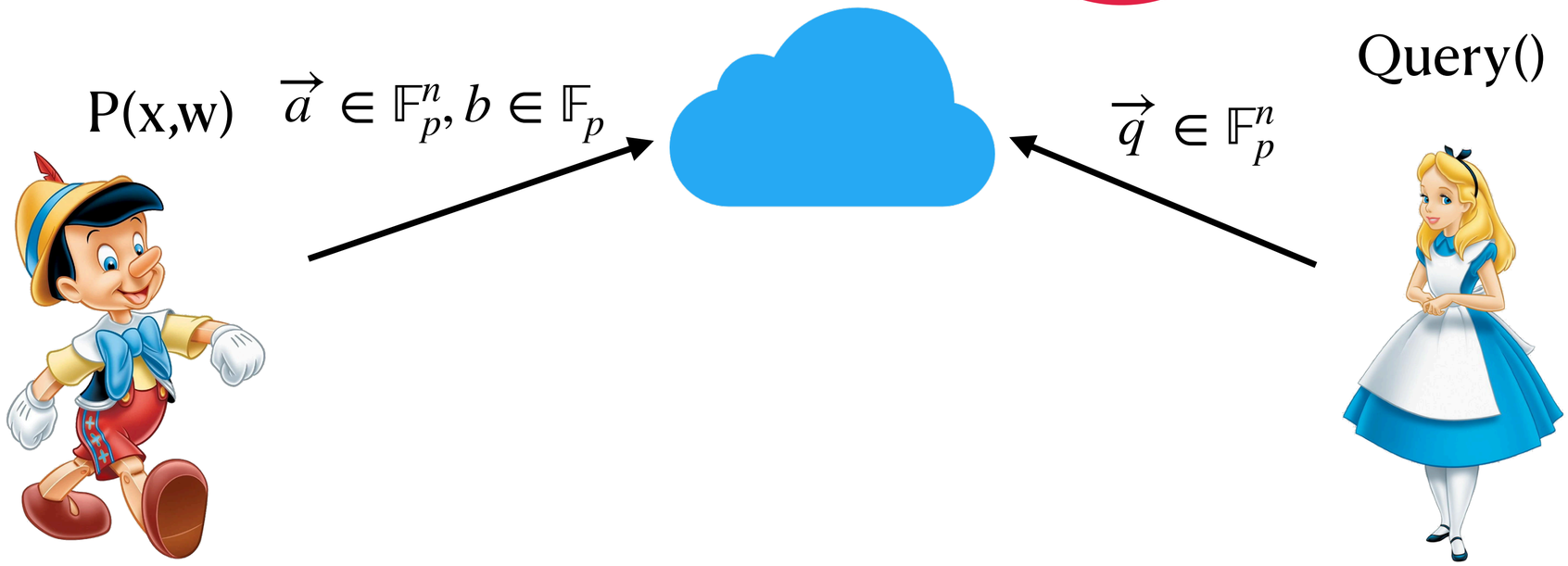
Query()

$\vec{q} \in \mathbb{F}_p^n$

Completeness: if $(x, w) \in R_L$

DV-SNARGs Construction

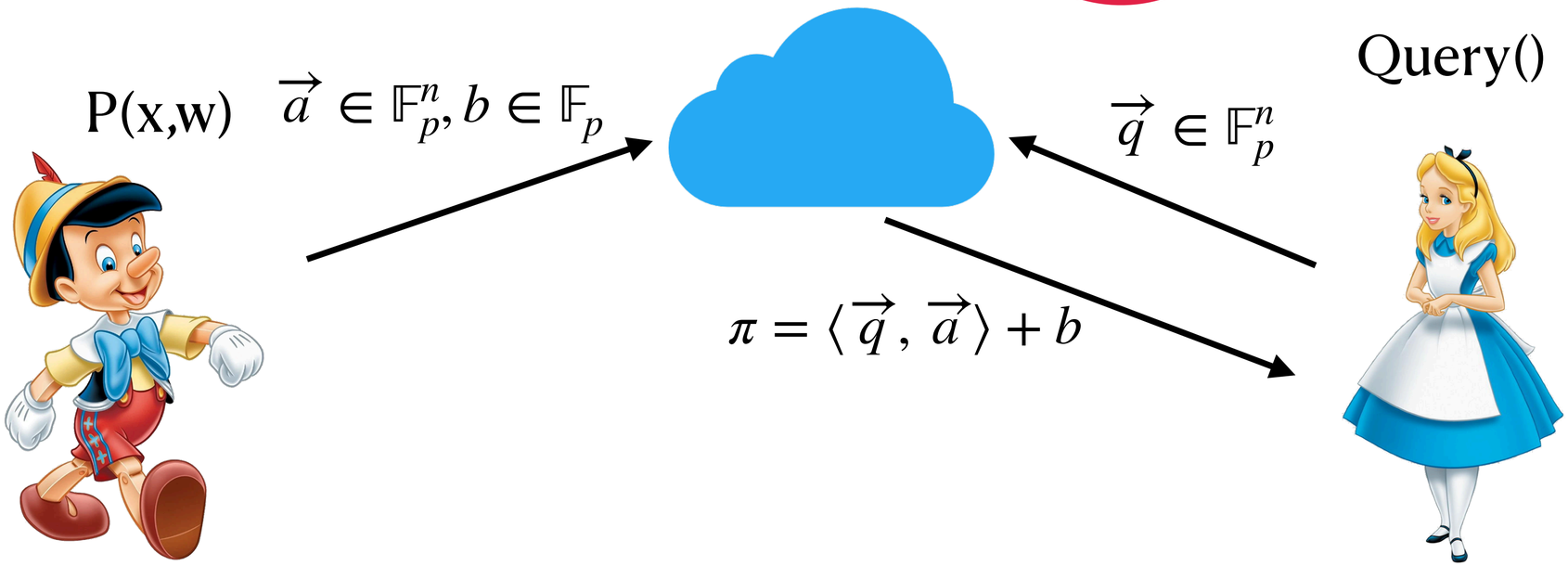
[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**



Completeness: if $(x, w) \in R_L$

DV-SNARGs Construction

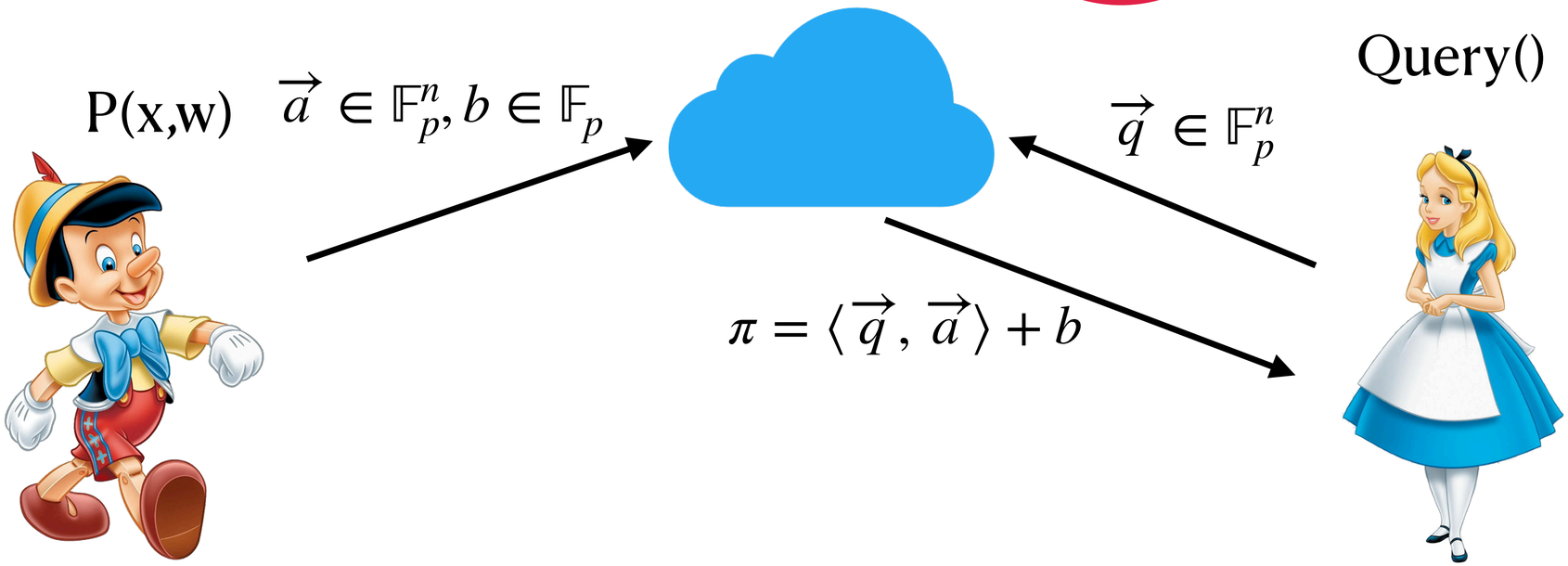
[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**



Completeness: if $(x, w) \in R_L$

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**



Completeness: if $(x, w) \in R_L$ V(x, π)=1

DV-SNARGs Construction

[BCIOP13]-recipe: Linear-Only Encryption + 1-query LPCP = DV-SNARG

P*(x)

DV-SNARGs Construction

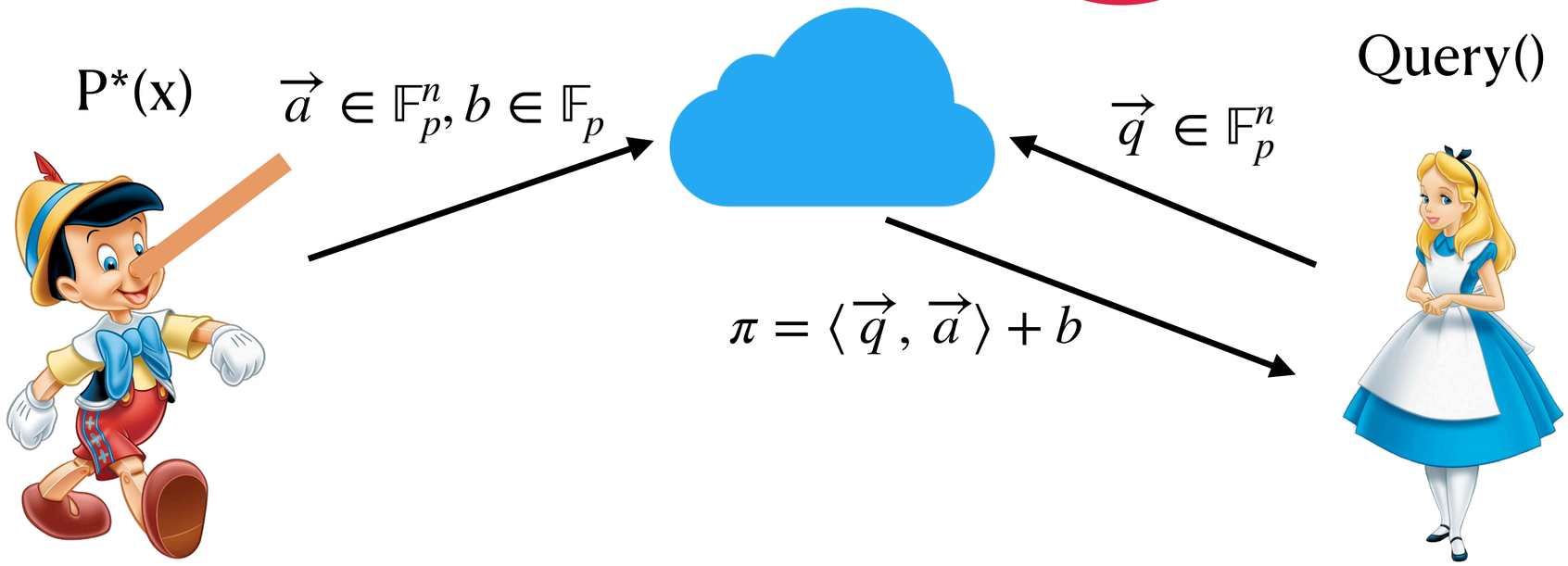
[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

P*(x)

Soundness: if $x \notin L$

DV-SNARGs Construction

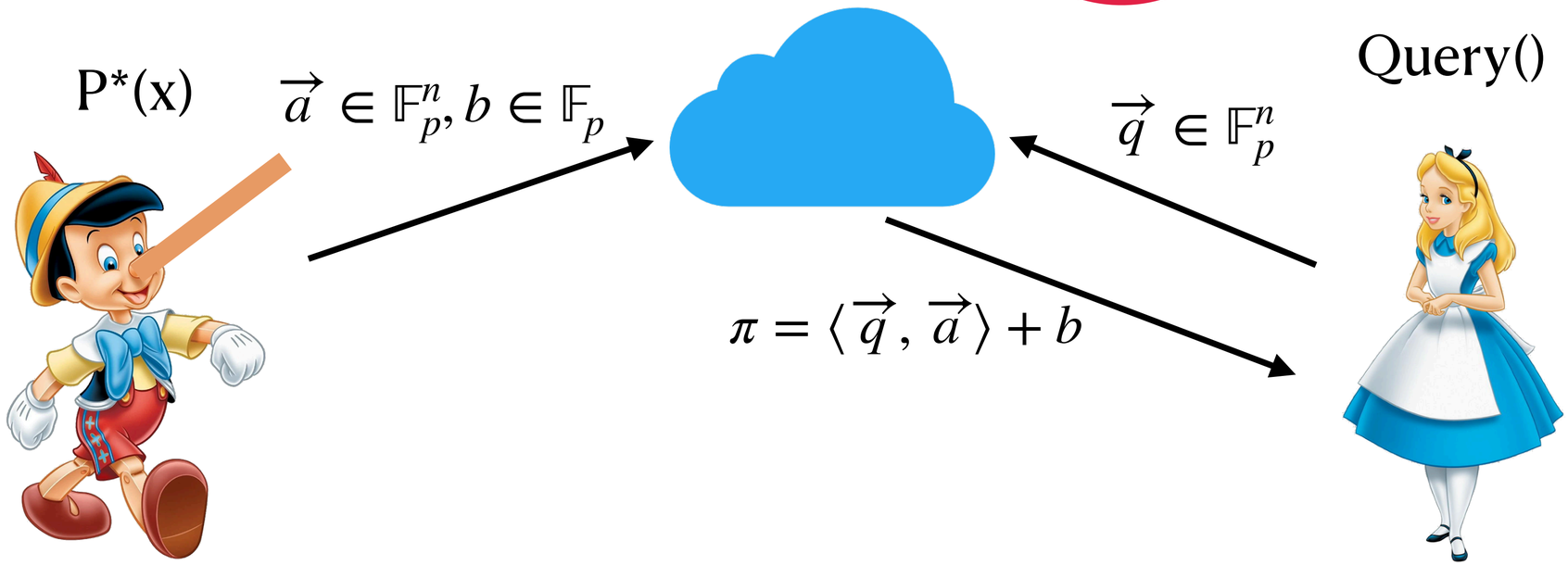
[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**



Soundness: if $x \notin L$

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**



Soundness: if $x \notin L$ V(x, π)=1 with prob. $2^{-\tau}$

DV-SNARGs Construction

[BCIOP13]-recipe: Linear-Only Encryption + 1-query LPCP = DV-SNARG

DV-SNARGs Construction

[BCIOP13]-recipe: Linear-Only Encryption + 1-query LPCP = DV-SNARG

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

(Keygen, Enc, Dec) is PKE

- Semantic Security
- Linear Homomorphism

  $\mathsf{Dec}(\mathsf{Enc}(d) \odot \mathsf{Enc}(e)) = d + e$

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

(Keygen, Enc, Dec) is PKE

- Semantic Security
- Linear Homomorphism

  $\mathsf{Dec}(\mathsf{Enc}(d) \odot \mathsf{Enc}(e)) = d + e$

$\mathsf{Enc}(q_1), \ldots, \mathsf{Enc}(q_n)$

ct

DV-SNARGs Construction

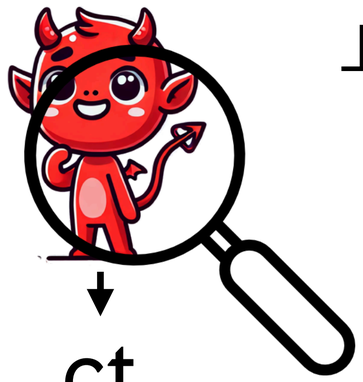
[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

(Keygen, Enc, Dec) is PKE

- Semantic Security
- Linear Homomorphism

  $\mathsf{Dec}(\mathsf{Enc}(d) \odot \mathsf{Enc}(e)) = d + e$

$\mathsf{Enc}(q_1), \ldots, \mathsf{Enc}(q_n)$



ct

$\perp \leftarrow \mathsf{Dec}(\mathsf{ct})$ or

$\mathsf{Ext} \rightarrow (\vec{a}, b)$ and

$\langle \vec{q}, \vec{a} \rangle + b = \mathsf{Dec}(\mathsf{ct})$

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

P(x,w)

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

P(x,w)

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

P(x,w)

crs = Enc(q_1)…Enc(q_n)

$\vec{q} \leftarrow LPCP.Query$

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

P(x,w)

$(\vec{a}, b) \leftarrow LPCP\,.\,P(x, w)$

crs = Enc(q_1)…Enc(q_n)

$\vec{q} \leftarrow LPCP\,.\,Query$

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

P(x,w)

crs = $\mathsf{Enc}(q_1)\ldots\mathsf{Enc}(q_n)$

$\vec{q} \leftarrow LPCP\,.\,Query$

$(\vec{a}, b) \leftarrow LPCP\,.\,P(x, w)$

$$\pi = \mathsf{Enc}(b) \bigodot_{i\in[n]} a_i \mathsf{Enc}(q_i)$$

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

P(x,w)

crs = $\mathsf{Enc}(q_1)\ldots\mathsf{Enc}(q_n)$

$\vec{q} \leftarrow LPCP\,.\,Query$

$(\vec{a}, b) \leftarrow LPCP\,.\,P(x, w)$

$\pi = \mathsf{Enc}(b) \bigodot_{i\in[n]} a_i \mathsf{Enc}(q_i)$

V(x,π):

$LPCP\,.\,V(x, \mathsf{Dec}(\pi))$

↓

0/1

DV-SNARGs Construction

[BCIOP13]-recipe: **Linear-Only Encryption** + **1-query LPCP** = **DV-SNARG**

P(x,w)

crs = $\mathsf{Enc}(q_1)\ldots\mathsf{Enc}(q_n)$

$\vec{q} \leftarrow LPCP\,.\,Query$

$(\vec{a}, b) \leftarrow LPCP\,.\,P(x, w)$

$$\pi = \mathsf{Enc}(b) \bigodot_{i\in[n]} a_i \mathsf{Enc}(q_i)$$

$|\pi|$ = Size of one ciphertext

V(x,π):

$LPCP\,.\,V(x, \mathsf{Dec}(\pi))$

↓

0/1

Packed ElGamal

Packed ElGamal

Encrypts vector $\vec{m}$ of n small numbers

Packed ElGamal

$\overrightarrow{\mathsf{pk}} = g^{\overrightarrow{\mathsf{sk}}}$

Encrypts vector $\vec{m}$ of n small numbers

Packed ElGamal

$$\overrightarrow{\mathsf{pk}} = g^{\overrightarrow{\mathsf{sk}}}$$

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}})$$

Encrypts vector $\vec{m}$ of n small numbers

Packed ElGamal

$$\overrightarrow{\mathsf{pk}} = g^{\overrightarrow{\mathsf{sk}}}$$

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}})$$

$$(g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \odot (g^{r'}, \overrightarrow{\mathsf{pk}}^{r'} \cdot g^{\vec{m'}})$$

Encrypts vector $\vec{m}$ of n small numbers

Packed ElGamal

$$\overrightarrow{\mathsf{pk}} = g^{\overrightarrow{\mathsf{sk}}}$$

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}})$$

Encrypts vector $\vec{m}$ of n small numbers

$$(g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \odot (g^{r'}, \overrightarrow{\mathsf{pk}}^{r'} \cdot g^{\vec{m'}})$$

$$= (g^{r+r'}, \overrightarrow{\mathsf{pk}}^{r+r'} \cdot g^{\vec{m}+\vec{m}'})$$

Packed ElGamal

$$\overrightarrow{\mathsf{pk}} = g^{\overrightarrow{\mathsf{sk}}}$$

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}})$$

Encrypts vector $\vec{m}$ of n small numbers

$$(g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \odot (g^{r'}, \overrightarrow{\mathsf{pk}}^{r'} \cdot g^{\vec{m'}})$$

$$= (g^{r+r'}, \overrightarrow{\mathsf{pk}}^{r+r'} \cdot g^{\vec{m}+\vec{m}'})$$

Decrypt component-wise:

$$\mathsf{DLog}(\mathsf{ct}_i/(\mathsf{ct}_0)^{\mathsf{sk}_i}) =$$

$$\mathsf{DLog}(\mathsf{pk}_i^r \cdot g^{m_i}/(g^r)^{\mathsf{sk}_i}) = m_i$$

Packed ElGamal

$$\overrightarrow{\mathsf{pk}} = g^{\overrightarrow{\mathsf{sk}}}$$

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}})$$

Encrypts vector $\vec{m}$ of n small numbers

Ciphertext n+1 group elements

$$(g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \odot (g^{r'}, \overrightarrow{\mathsf{pk}}^{r'} \cdot g^{\vec{m'}})$$

$$= (g^{r+r'}, \overrightarrow{\mathsf{pk}}^{r+r'} \cdot g^{\vec{m}+\vec{m}'})$$

Decrypt component-wise:

$$\mathsf{DLog}(\mathsf{ct}_i/(\mathsf{ct}_0)^{\mathsf{sk}_i}) =$$

$$\mathsf{DLog}(\mathsf{pk}_i^r \cdot g^{m_i}/(g^r)^{\mathsf{sk}_i}) = m_i$$

Packed ElGamal

$$\overrightarrow{\mathsf{pk}} = g^{\overrightarrow{\mathsf{sk}}}$$

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}})$$

$$(g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \odot (g^{r'}, \overrightarrow{\mathsf{pk}}^{r'} \cdot g^{\vec{m'}})$$

$$= (g^{r+r'}, \overrightarrow{\mathsf{pk}}^{r+r'} \cdot g^{\vec{m}+\vec{m}'})$$

Decrypt component-wise:

$$\mathsf{DLog}(\mathsf{ct}_i/(\mathsf{ct}_0)^{\mathsf{sk}_i}) =$$

$$\mathsf{DLog}(\mathsf{pk}_i^r \cdot g^{m_i}/(g^r)^{\mathsf{sk}_i}) = m_i$$

Encrypts vector $\vec{m}$ of n small numbers

Ciphertext n+1 group elements

Can be compressed [DGIMMO19]

Distributed Discrete Logarithm

[BGI17,DKK18]

g^a

g^{a+b}

Distributed Discrete Logarithm

[BGI17,DKK18]

g^a

g^{a+b}

$c \leftarrow \mathsf{DDLog}(g^a)$

Distributed Discrete Logarithm

[BGI17,DKK18]

g^a

g^{a+b}

$c \leftarrow \mathsf{DDLog}(g^a)$

$c - b = \mathsf{DDLog}(g^{a+b})$ with prob. $1/\mathrm{poly}(\lambda)$

Packed ElGamal cont.

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \text{ and } \vec{m} < p$$

$$\mathsf{Compress}(\mathsf{ct}) = (g^r, \mathsf{DDLog}(\overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \mod p)$$

Packed ElGamal cont.

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \text{ and } \vec{m} < p$$

$$\mathsf{Compress}(\mathsf{ct}) = (g^r, \mathsf{DDLog}(\overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \mod p)$$

$$\mathsf{DDLog}((g^r)^{\mathsf{sk}_i})\text{-}\mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

Packed ElGamal cont.

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \text{ and } \vec{m} < p$$

$$\mathsf{Compress}(\mathsf{ct}) = (g^r, \mathsf{DDLog}(\overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \mod p)$$

$$\mathsf{DDLog}((g^r)^{\mathsf{sk}_i})\text{-}\mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

$$= \mathsf{DDLog}(\mathsf{pk}_i^r) - \mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

Packed ElGamal cont.

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \text{ and } \vec{m} < p$$

$$\mathsf{Compress}(\mathsf{ct}) = (g^r, \mathsf{DDLog}(\overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \mod p)$$

$$\mathsf{DDLog}((g^r)^{\mathsf{sk}_i})\text{-}\mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

$$= \mathsf{DDLog}(\mathsf{pk}_i^r) - \mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

$$= c - (c - m) \mod p \text{ (with prob. } 1/\mathrm{poly}(\lambda))$$

Packed ElGamal cont.

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \text{ and } \vec{m} < p$$

$$\mathsf{Compress}(\mathsf{ct}) = (g^r, \mathsf{DDLog}(\overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \mod p)$$

$$\mathsf{DDLog}((g^r)^{\mathsf{sk}_i})\text{-}\mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

$$= \mathsf{DDLog}(\mathsf{pk}_i^r) - \mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

$$= c - (c - m) \mod p \text{ (with prob. } 1/\mathrm{poly}(\lambda))$$

$$= m$$

Packed ElGamal cont.

$$\mathsf{ct} = (g^r, \overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \text{ and } \vec{m} < p$$

$$\mathsf{Compress}(\mathsf{ct}) = (g^r, \mathsf{DDLog}(\overrightarrow{\mathsf{pk}}^r \cdot g^{\vec{m}}) \mod p)$$

Size: 1 group element+n·log(p)

$$\mathsf{DDLog}((g^r)^{\mathsf{sk}_i})\text{-}\mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

$$= \mathsf{DDLog}(\mathsf{pk}_i^r) - \mathsf{DDLog}(\mathsf{pk}_i^r \cdot g^{m_i}) \mod p$$

$$= c - (c - m) \mod p \text{ (with prob. } 1/\mathrm{poly}(\lambda))$$

$$= m$$

Still Linear Only?

Still Linear Only?

No!

Still Linear Only?

No!

use decryption failure to

evaluate constant locality functions

Isolated Homomorphism

- Semantic Security
- Linear Homomorphism

$$\mathsf{Dec}(\mathsf{Compress}(\mathsf{Enc}(\vec{m}) \odot \mathsf{Enc}(\vec{m}'))) \approx \vec{m} + \vec{m}'$$

Isolated Homomorphism

- Semantic Security
- Linear Homomorphism

  $\mathsf{Dec}(\mathsf{Compress}(\mathsf{Enc}(\vec{m}) \odot \mathsf{Enc}(\vec{m}')))$

  $\approx \vec{m} + \vec{m}'$

$\mathsf{Enc}(q_{1,1}, \ldots, q_{1,n})$

$\vdots$

$\mathsf{Enc}(q_{m,1}, \ldots, q_{m,n})$

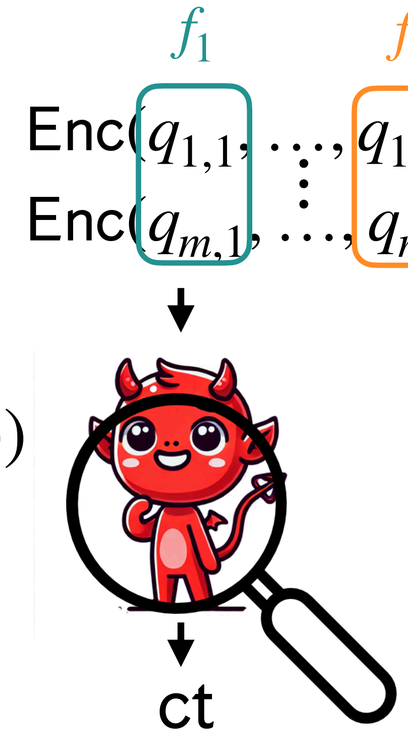
ct

Isolated Homomorphism

- Semantic Security
- Linear Homomorphism

  $\mathsf{Dec}(\mathsf{Compress}(\mathsf{Enc}(\vec{m}) \odot \mathsf{Enc}(\vec{m}')))$

  $\approx \vec{m} + \vec{m}'$



$\perp \leftarrow \mathsf{Dec}(\mathsf{ct})$ or

$\mathsf{Ext} \rightarrow (f_1, \ldots, f_n)$

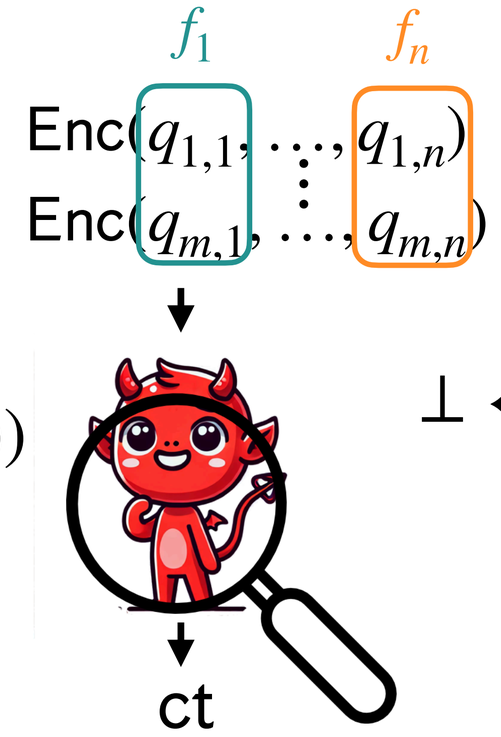
$f_1(q_{1,1} \ldots q_{m,1}), \ldots f_n(q_{n,1} \ldots q_{n,m}) = \mathsf{Dec}(\mathsf{ct})$

Isolated Homomorphism

- Semantic Security
- Linear Homomorphism

  $\mathsf{Dec}(\mathsf{Compress}(\mathsf{Enc}(\vec{m}) \odot \mathsf{Enc}(\vec{m}')))$

  $\approx \vec{m} + \vec{m}'$



$\perp \leftarrow \mathsf{Dec}(\mathsf{ct})$ or

$\mathsf{Ext} \rightarrow (f_1, \ldots, f_n)$

$f_1(q_{1,1} \ldots q_{m,1}), \ldots f_n(q_{n,1} \ldots q_{n,m}) = \mathsf{Dec}(\mathsf{ct})$

Packed ElGamal ✅

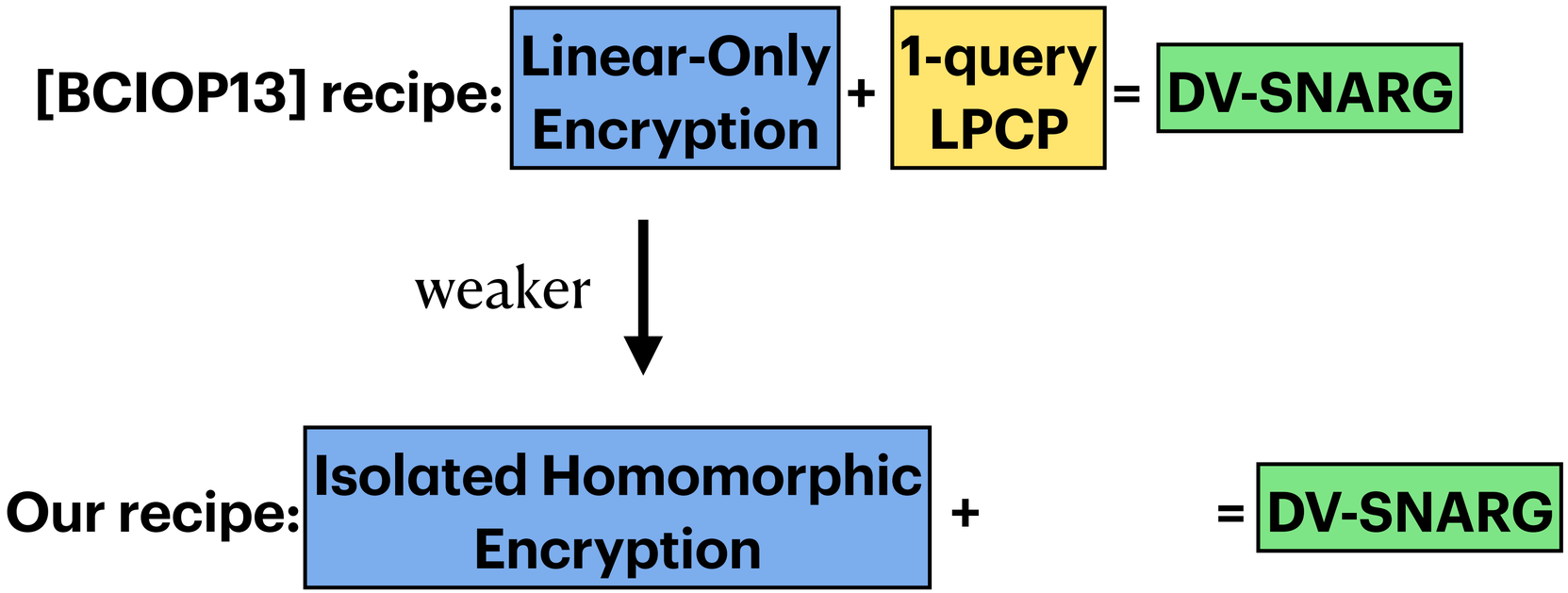
in the GGM

Recipes

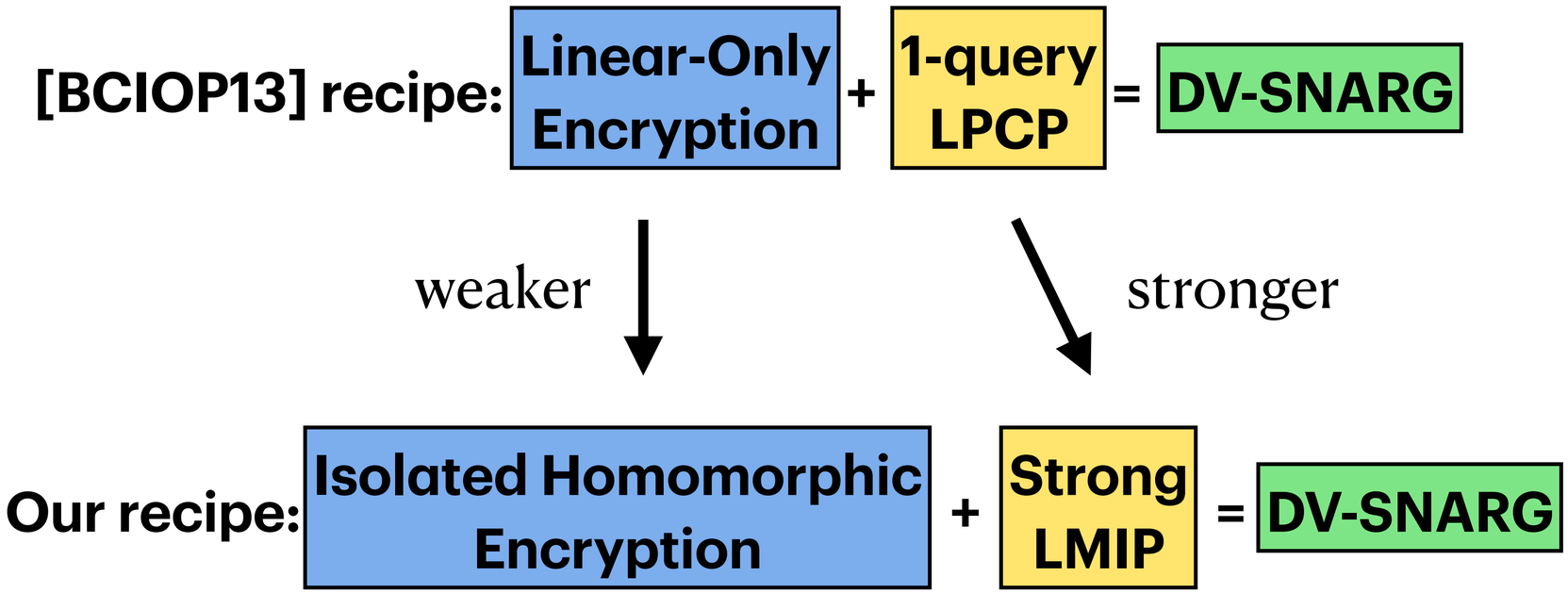
[BCIOP13] recipe: Linear-Only Encryption + 1-query LPCP = DV-SNARG

Our recipe: + = DV-SNARG

Recipes

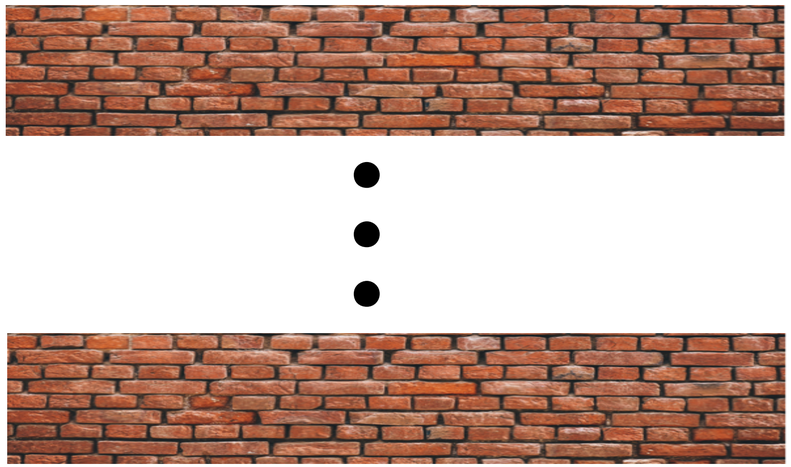

[BCIOP13] recipe:
Linear-Only Encryption
+
1-query LPCP
= DV-SNARG
weaker
Our recipe:
Isolated Homomorphic Encryption
+
= DV-SNARG

Recipes


[BCIOP13] recipe:
Linear-Only Encryption
+
1-query LPCP
=
DV-SNARG
weaker
stronger
Our recipe:
Isolated Homomorphic Encryption
+
Strong LMIP
=
DV-SNARG

Strong Linear MIP

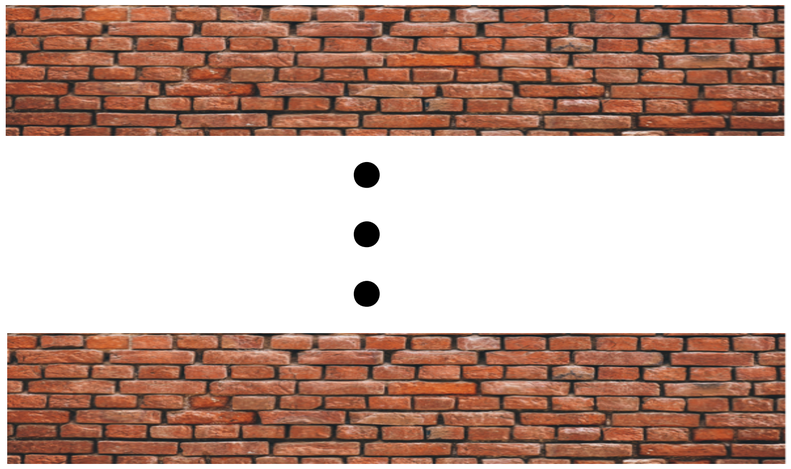
$P_1(x,w)$



$P_n(x,w)$

Strong Linear MIP

$P_1(x,w)$



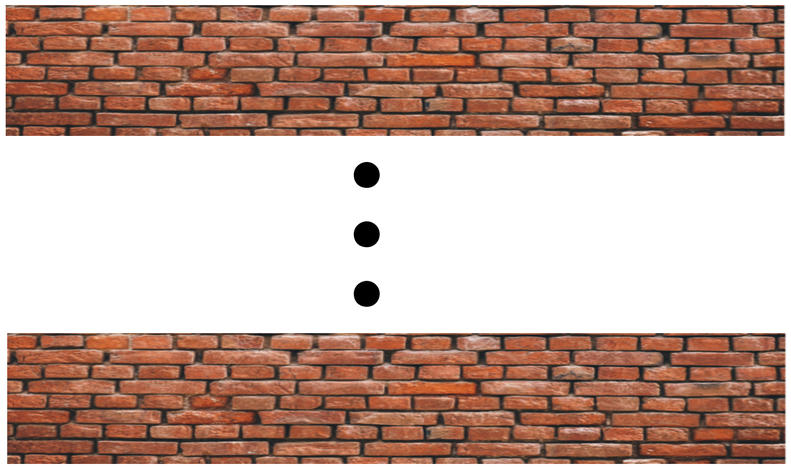
$P_n(x,w)$

$$\begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix}$$

Query()

Strong Linear MIP

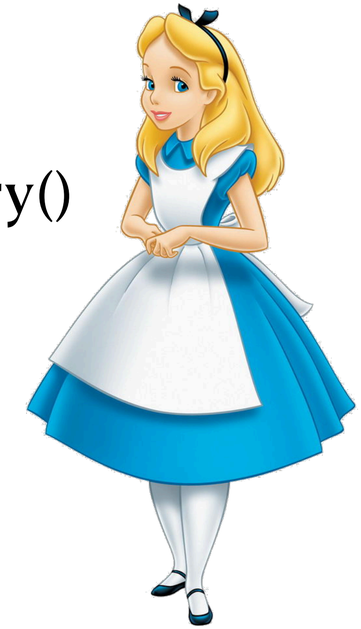
$P_1(x,w)$



$P_n(x,w)$

$$\begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix}$$

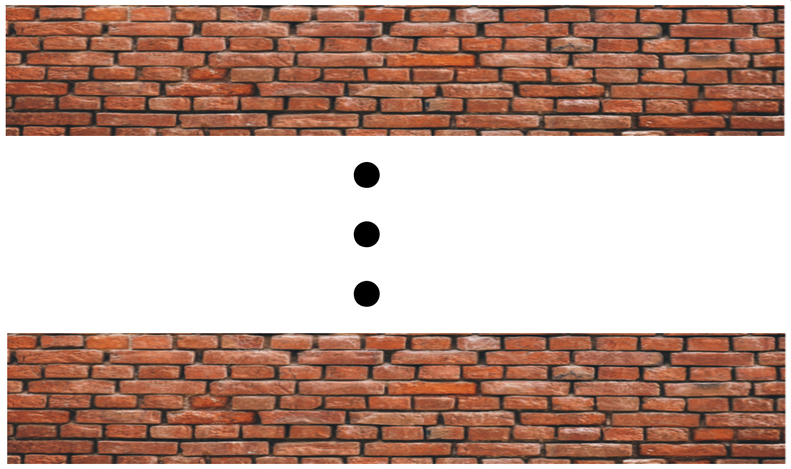
Query()



Completeness: if $(x, w) \in R_L$

Strong Linear MIP

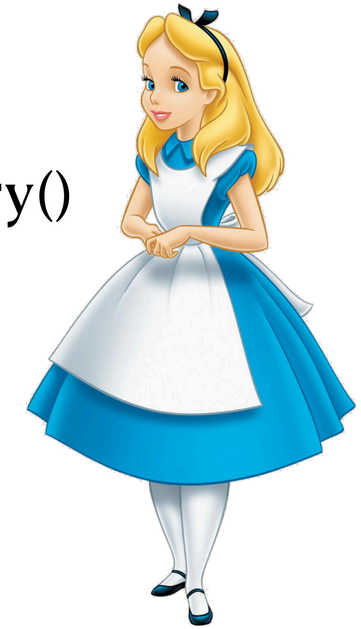
$P_1(x,w)$



$\vec{a} \in \mathbb{F}_p^n, \vec{b} \in \mathbb{F}_p^m$

$$\begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix}$$

Query()



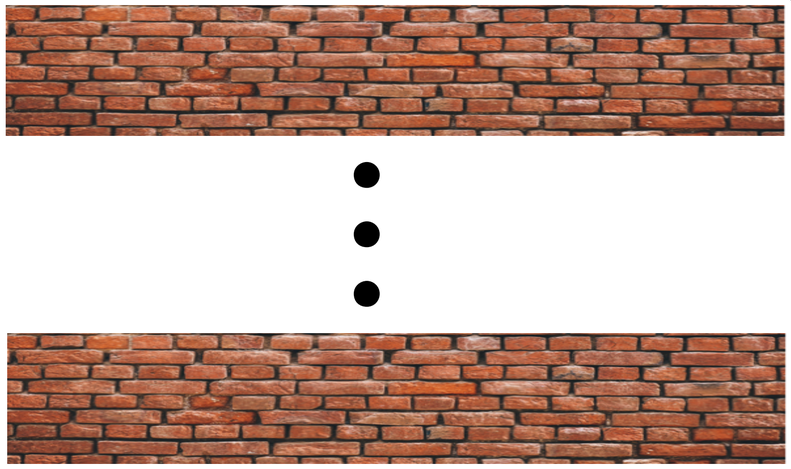
$\vec{a} \in \mathbb{F}_p^n, \vec{b} \in \mathbb{F}_p^m$

$P_n(x,w)$

Completeness: if $(x, w) \in R_L$

Strong Linear MIP

$P_1(x,w)$



$\vec{a} \in \mathbb{F}_p^n, \vec{b} \in \mathbb{F}_p^m$

$$\begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix}$$ Query()

$$\vec{a}^t \begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix} + b = \pi$$

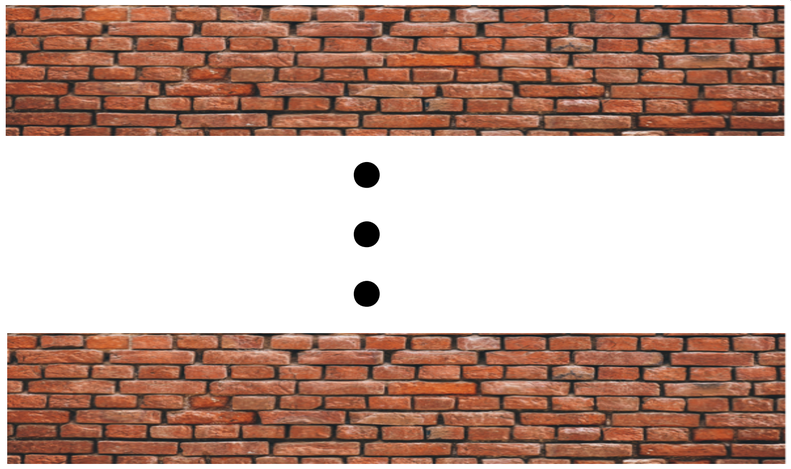
$\vec{a} \in \mathbb{F}_p^n, \vec{b} \in \mathbb{F}_p^m$

$P_n(x,w)$

Completeness: if $(x, w) \in R_L$

Strong Linear MIP

$P_1(x,w)$



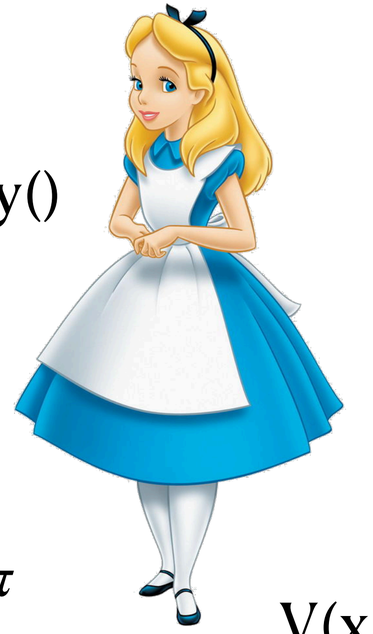
$P_n(x,w)$

$\vec{a} \in \mathbb{F}_p^n, \vec{b} \in \mathbb{F}_p^m$

$\vec{a} \in \mathbb{F}_p^n, \vec{b} \in \mathbb{F}_p^m$

$$\begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix}$$ Query()

$$\vec{a}^t \begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix} + b = \pi$$

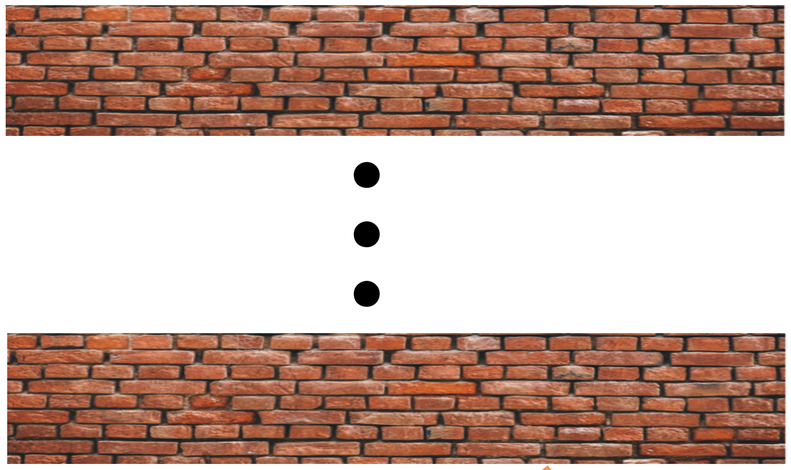


V(x, π)=1

Completeness: if $(x, w) \in R_L$

Strong Linear MIP

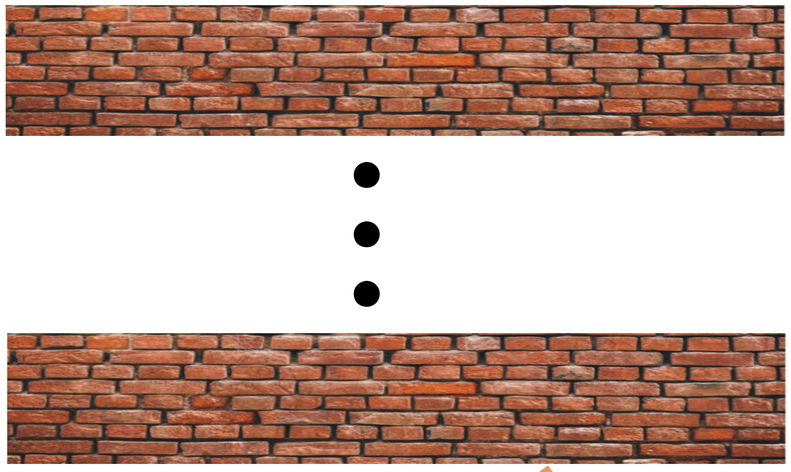
$P^*_1(x)$



$P^*_n(x)$

Strong Linear MIP

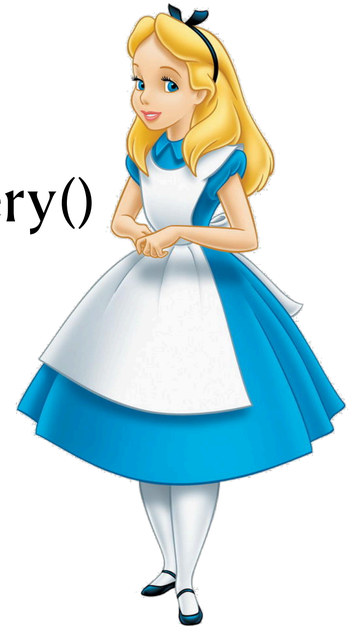
$P^*_1(x)$



$P^*_n(x)$

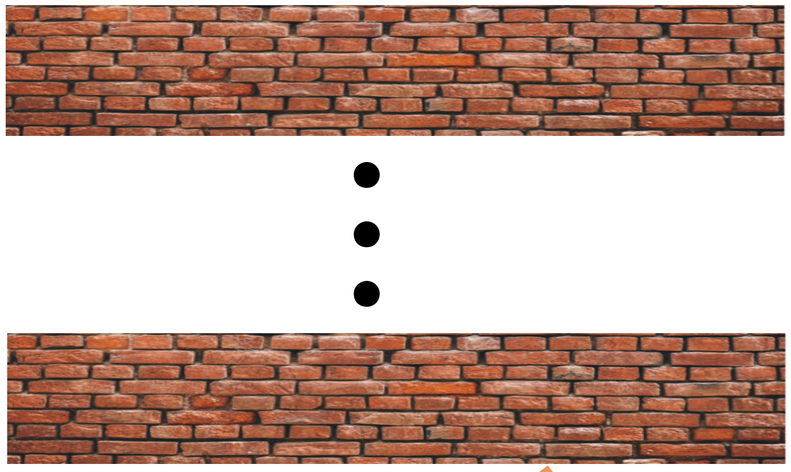
$$\begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix}$$

Query()



Strong Linear MIP

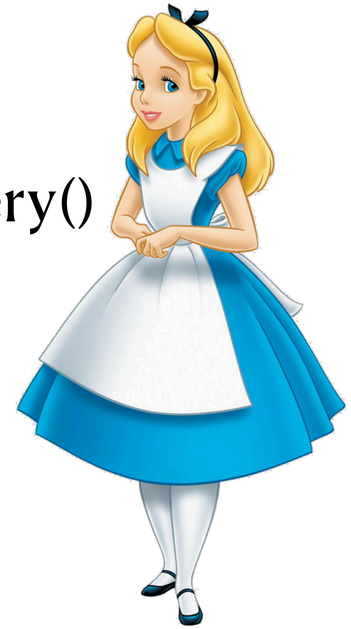
$P^*_1(x)$



$P^*_n(x)$

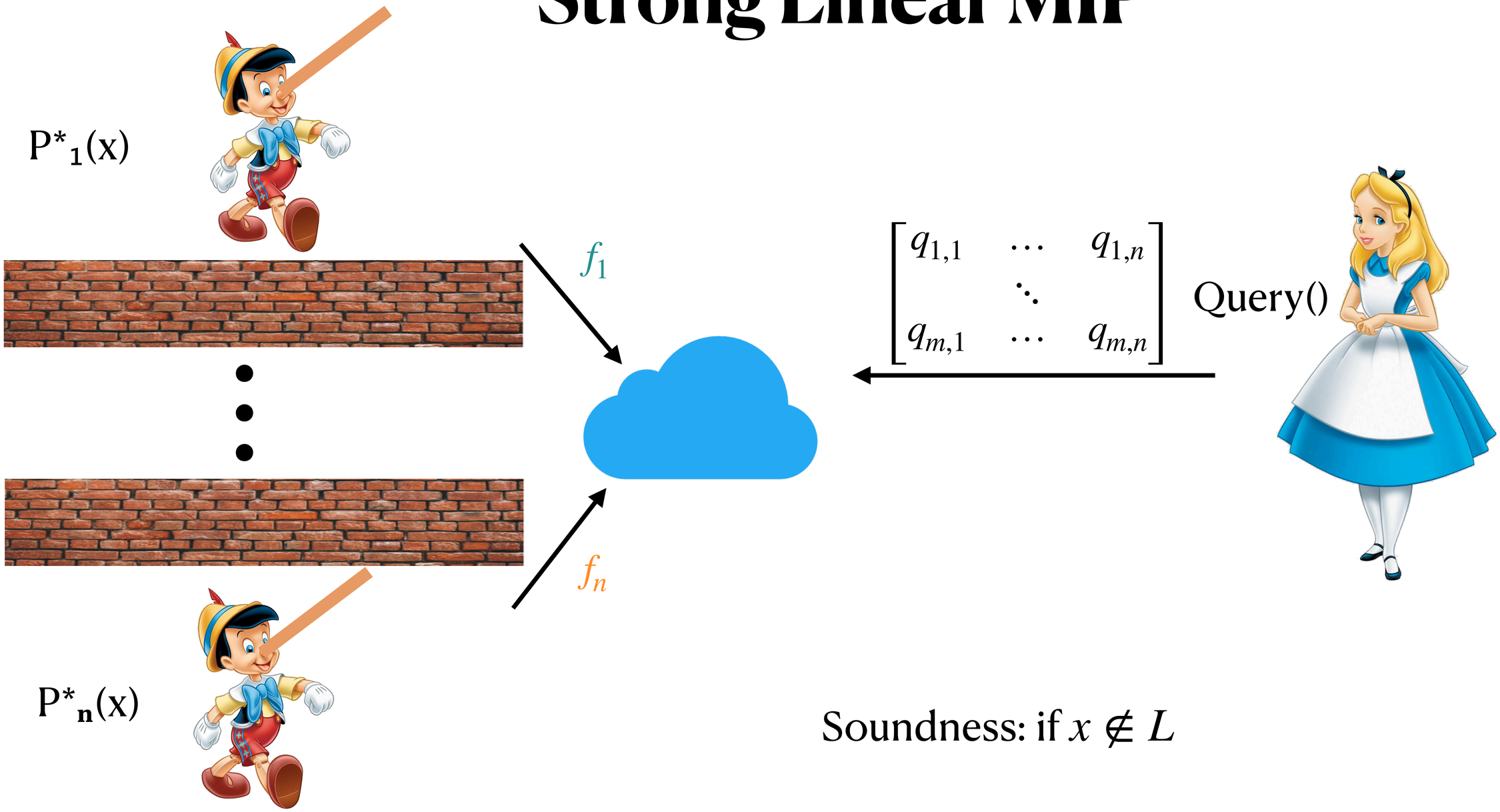
$$\begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix}$$

Query()

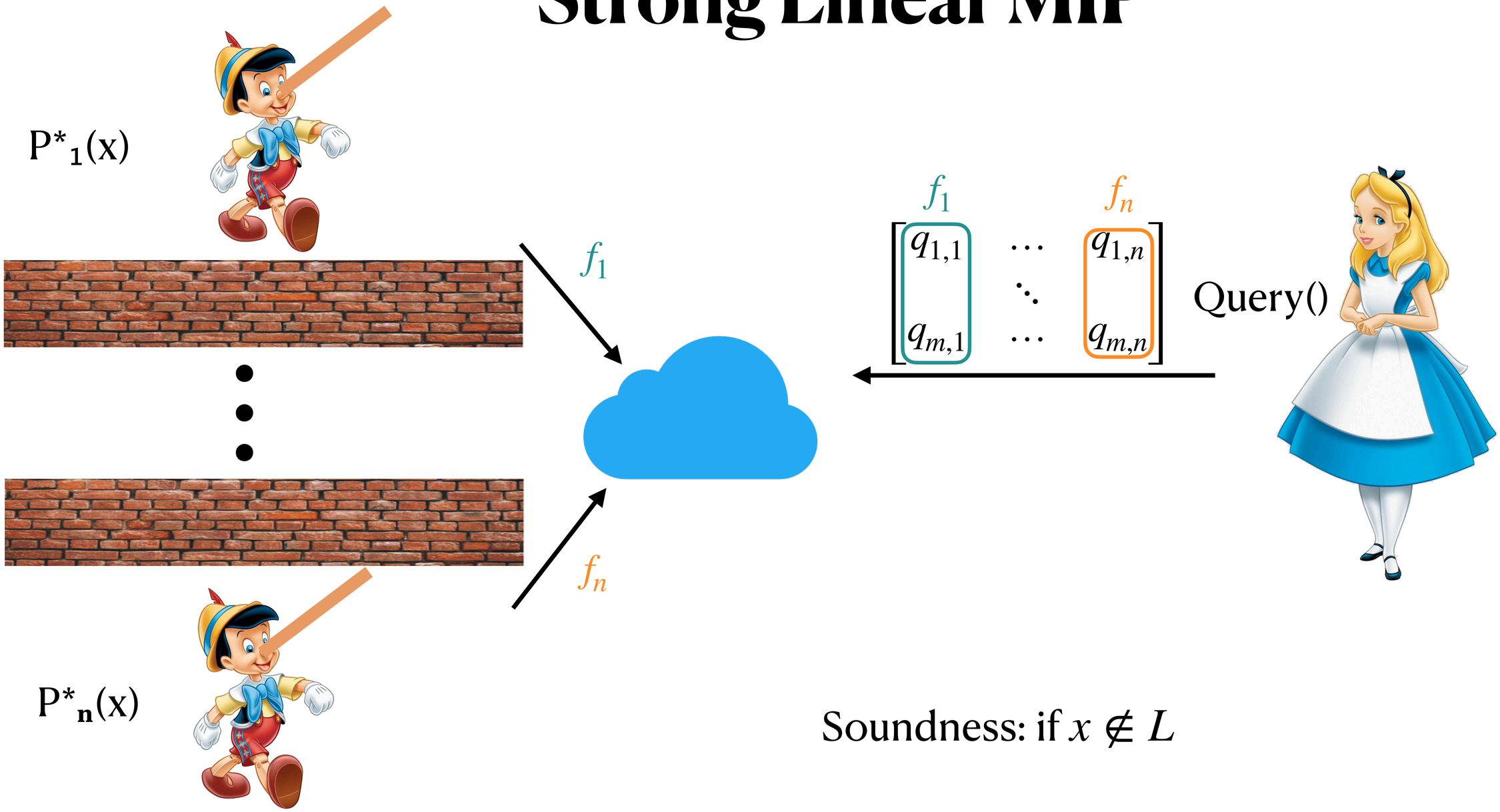


Soundness: if $x \notin L$

Strong Linear MIP

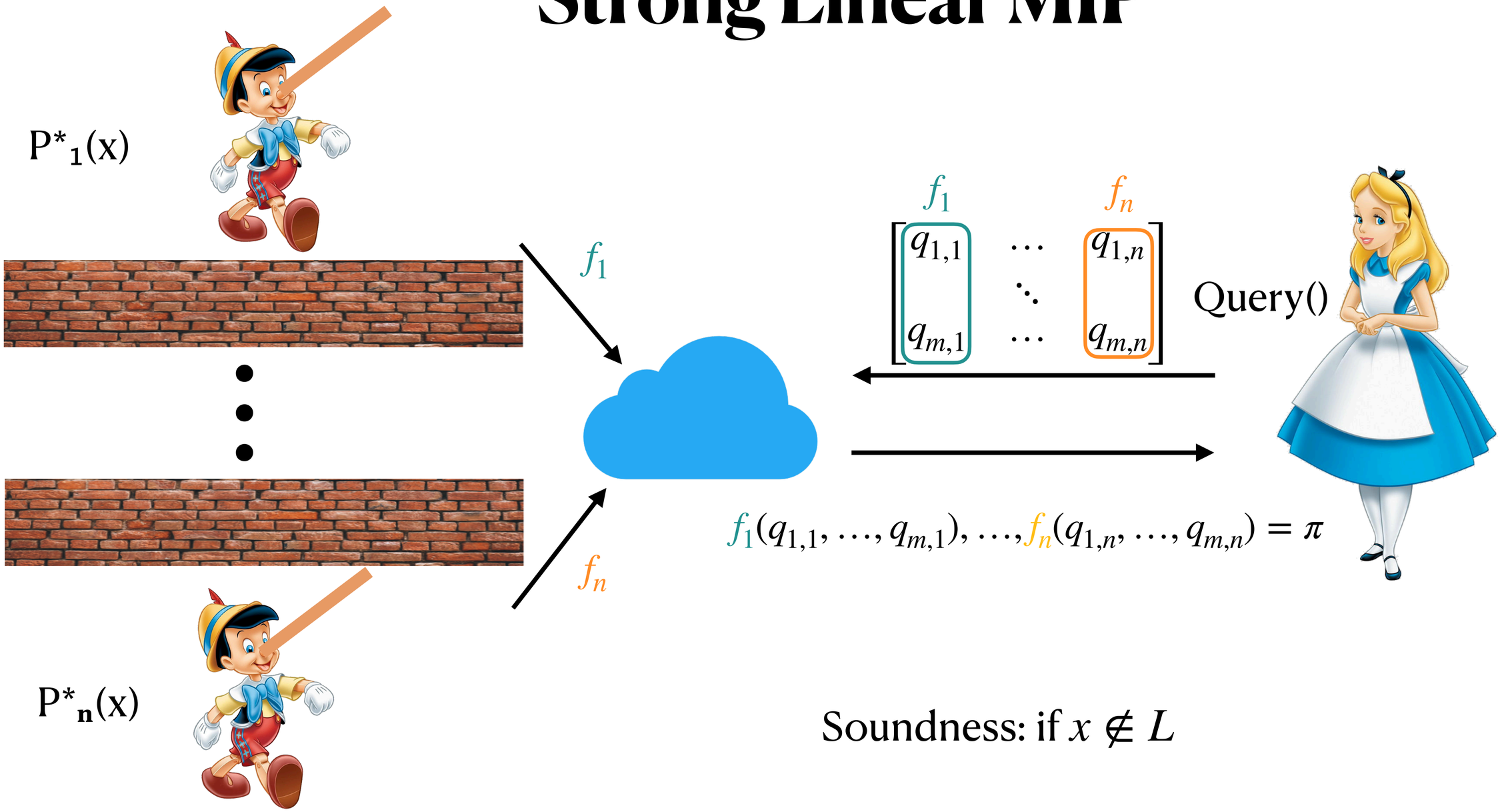


Strong Linear MIP



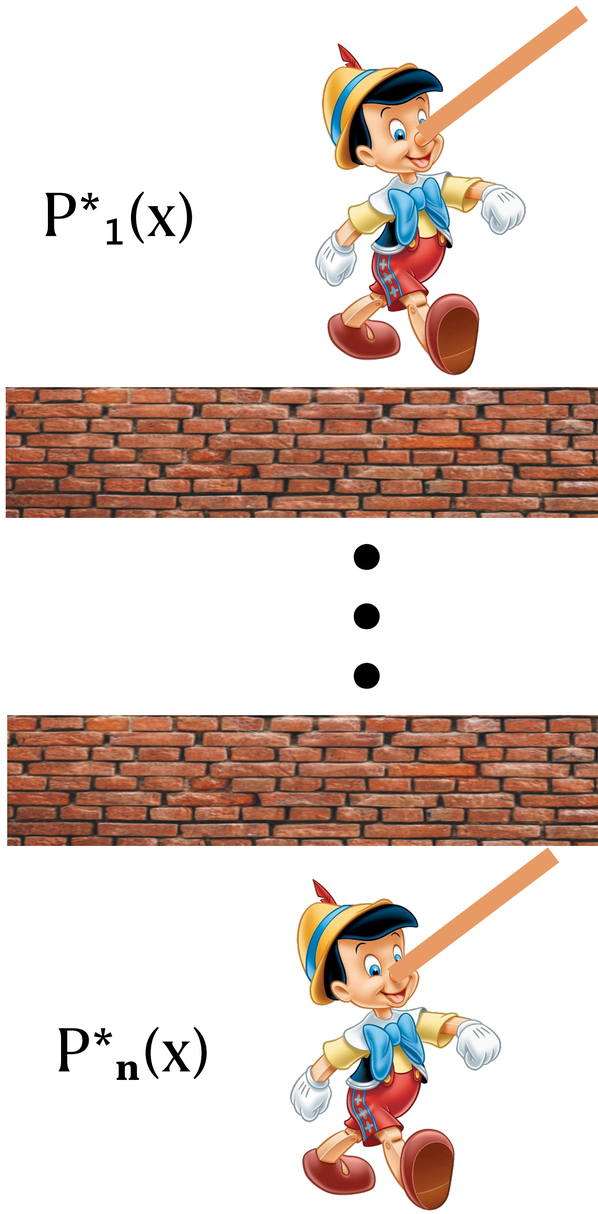
Soundness: if $x \notin L$

Strong Linear MIP



Soundness: if $x \notin L$

Strong Linear MIP



f_1

f_n

$$\begin{matrix} f_1 & & f_n \\ \begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ & \ddots & \\ q_{m,1} & \cdots & q_{m,n} \end{bmatrix} & & \end{matrix}$$

Query()

$f_1(q_{1,1}, \ldots, q_{m,1}), \ldots, f_n(q_{1,n}, \ldots, q_{m,n}) = \pi$

V(x, π)=1

with prob. $2^{-\tau}$

Soundness: if $x \notin L$

DV-SNARGs from Linear-Only Encryption

P(x,w)

DV-SNARGs from Linear-Only Encryption

P(x,w)

$\vec{q_1}, \ldots \vec{q_n} \leftarrow LMIP\,.\,Query$

DV-SNARGs from Linear-Only Encryption

P(x,w)

$\mathsf{Enc}(\vec{q_1})$
$\vdots$
$\mathsf{Enc}(\vec{q_n})$

$\vec{q_1}, \ldots \vec{q_n} \leftarrow LMIP\,.\,Query$

DV-SNARGs from Linear-Only Encryption

P(x,w)

$(\vec{a}, b) \leftarrow LMIP.P(x, w)$

$\mathsf{Enc}(\vec{q_1})$
$\vdots$
$\mathsf{Enc}(\vec{q_n})$

$\vec{q_1}, \ldots \vec{q_n} \leftarrow LMIP.Query$

DV-SNARGs from Linear-Only Encryption

P(x,w)

$(\vec{a}, b) \leftarrow LMIP.P(x, w)$

$\pi = \mathsf{Compress}(\mathsf{Enc}(b) \bigodot_{i \in [n]} a_i \mathsf{Enc}(\vec{q}_i))$

$\mathsf{Enc}(\vec{q}_1)$
$\vdots$
$\mathsf{Enc}(\vec{q}_n)$

$\vec{q}_1, \ldots \vec{q}_n \leftarrow LMIP.Query$

DV-SNARGs from Linear-Only Encryption

$\mathsf{Enc}(\vec{q_1})$
$\vdots$
$\mathsf{Enc}(\vec{q_n})$

P(x,w)

$\vec{q_1}, \ldots \vec{q_n} \leftarrow LMIP.Query$

$(\vec{a}, b) \leftarrow LMIP.P(x, w)$

$\pi = \mathsf{Compress}(\mathsf{Enc}(b) \bigodot_{i \in [n]} a_i \mathsf{Enc}(\vec{q_i}))$

V(x,π):

$LMIP.V(x, \mathsf{Dec}(\pi))$

↓

0/1

DV-SNARGs from Linear-Only Encryption

P(x,w)

$(\vec{a}, b) \leftarrow LMIP.P(x, w)$

$\pi = \mathsf{Compress}(\mathsf{Enc}(b) \bigodot_{i \in [n]} a_i \mathsf{Enc}(\vec{q}_i))$

$\mathsf{Enc}(\vec{q}_1)$
$\vdots$
$\mathsf{Enc}(\vec{q}_n)$

$\vec{q}_1, \ldots \vec{q}_n \leftarrow LMIP.Query$

- Completeness: ✅ if decryption correct

V(x,π):

$LMIP.V(x, \mathsf{Dec}(\pi))$

↓

0/1

DV-SNARGs from Linear-Only Encryption

$\mathsf{Enc}(\vec{q_1})$
$\vdots$
$\mathsf{Enc}(\vec{q_n})$

P(x,w)

$\vec{q_1}, \ldots \vec{q_n} \leftarrow LMIP\,.\,Query$

$(\vec{a}, b) \leftarrow LMIP\,.\,P(x, w)$

$$\pi = \mathsf{Compress}(\mathsf{Enc}(b) \bigodot_{i \in [n]} a_i \mathsf{Enc}(\vec{q_i}))$$

- Completeness: ✅ if decryption correct
- Succinctness: $|\pi|$ = Size of one ciphertext
  $= 256 + |\text{LMIP response}|$
  $= 263$

V(x,π):

$LMIP\,.\,V(x, \mathsf{Dec}(\pi))$

↓

0/1

DV-SNARGs from Linear-Only Encryption

$\mathsf{Enc}(\vec{q_1})$
$\vdots$
$\mathsf{Enc}(\vec{q_n})$

P(x,w)

$(\vec{a}, b) \leftarrow LMIP.P(x, w)$

$\vec{q_1}, \ldots \vec{q_n} \leftarrow LMIP.Query$

$\pi = \mathsf{Compress}(\mathsf{Enc}(b) \bigodot_{i \in [n]} a_i \mathsf{Enc}(\vec{q_i}))$

V(x,π):

$LMIP.V(x, \mathsf{Dec}(\pi))$

0/1

- Completeness: ✅ if decryption correct
- Succinctness: $|\pi|$ = Size of one ciphertext
  = 256 + |LMIP response|
  = 263
- Plausibly practical: ⚡P,V complexity linear in |w|

DV-SNARGs from Linear-Only Encryption

P*(x)

DV-SNARGs from Linear-Only Encryption

P*(x)

Soundness: $x \notin L$

DV-SNARGs from Linear-Only Encryption

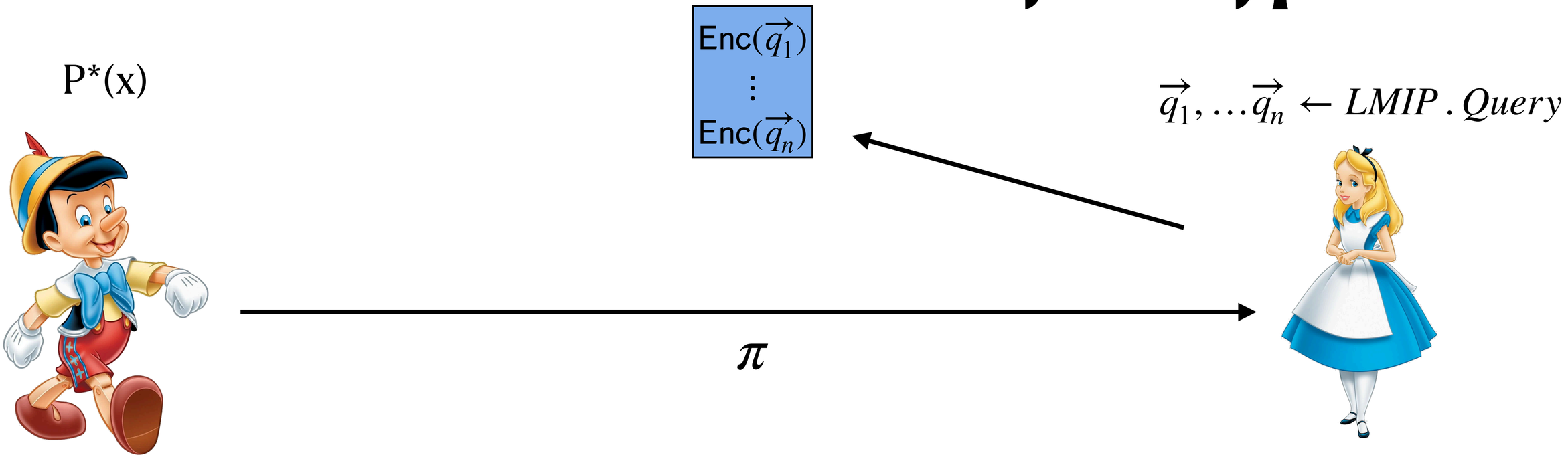
P*(x)

$\mathsf{Enc}(\vec{q_1})$
$\vdots$
$\mathsf{Enc}(\vec{q_n})$

$\vec{q_1}, \ldots \vec{q_n} \leftarrow LMIP\,.\,Query$

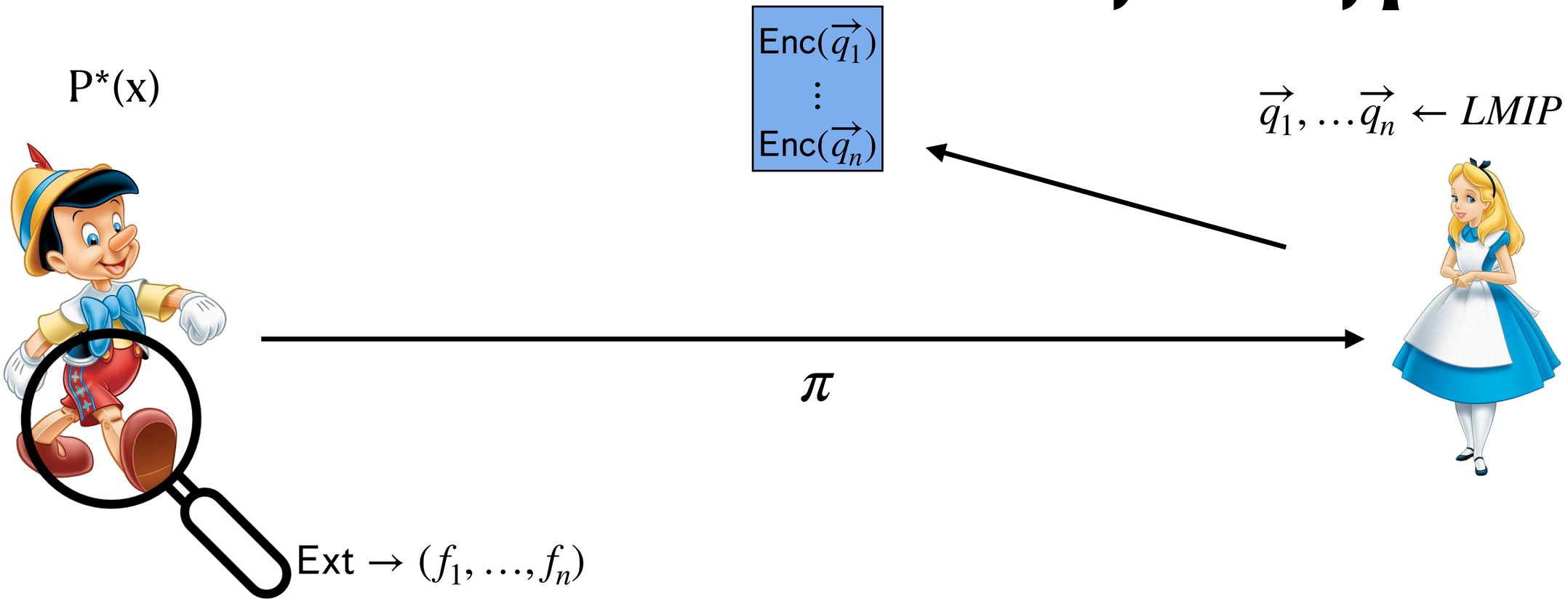
Soundness: $x \notin L$

DV-SNARGs from Linear-Only Encryption



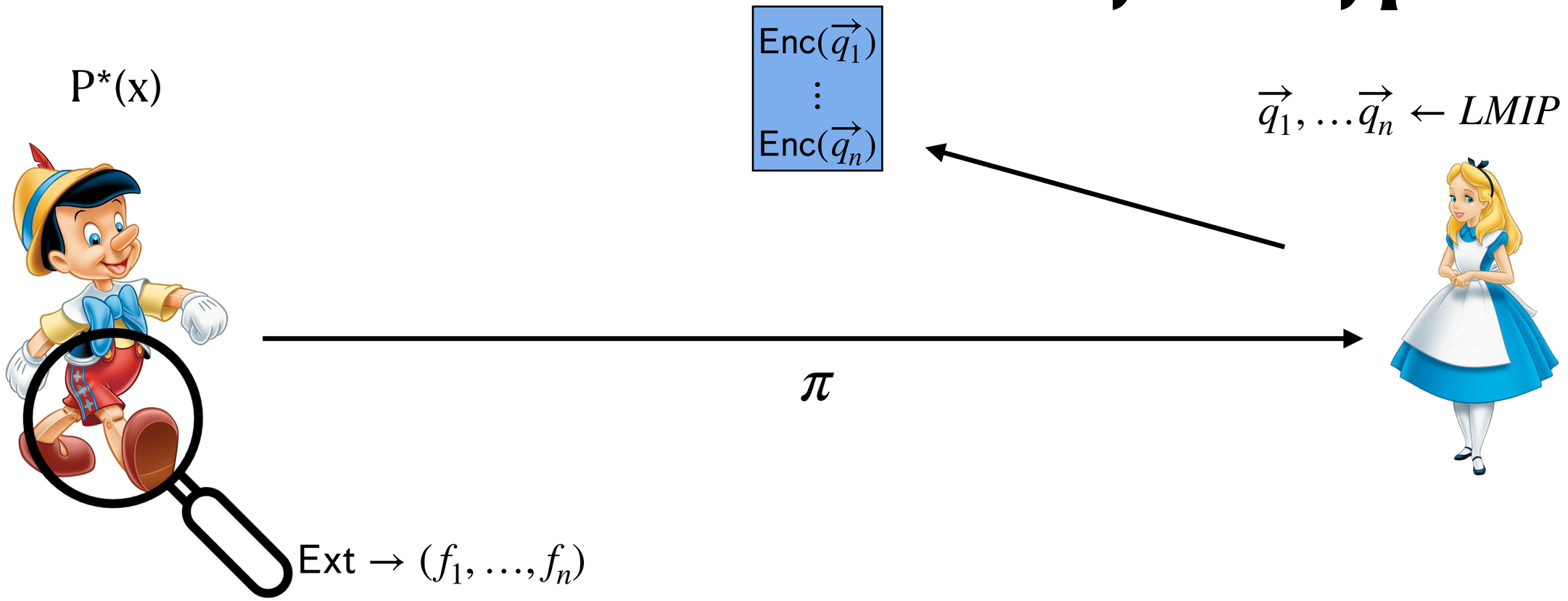
Soundness: $x \notin L$

DV-SNARGs from Linear-Only Encryption



Soundness: $x \notin L$

DV-SNARGs from Linear-Only Encryption



Soundness: $x \notin L$

reduce to LMIP soundness with proof $f_1, \ldots, f_n$

Open Problems

- Further reduce size - Challenge: ~1|G|+τ
- Make this strongly reusable
- Improve P, V complexity to $\sqrt[4]{|w|}$
- Use compressed ElGamal homomorphism

Take-Aways

There are very small group based DV-SNARGs

Isolated Homomorphic Encryption + **Strong LMIP** = **DV-SNARG**

Compressed ElGamal is not linear-only

Read the paper :)

https://ia.cr/2025/517