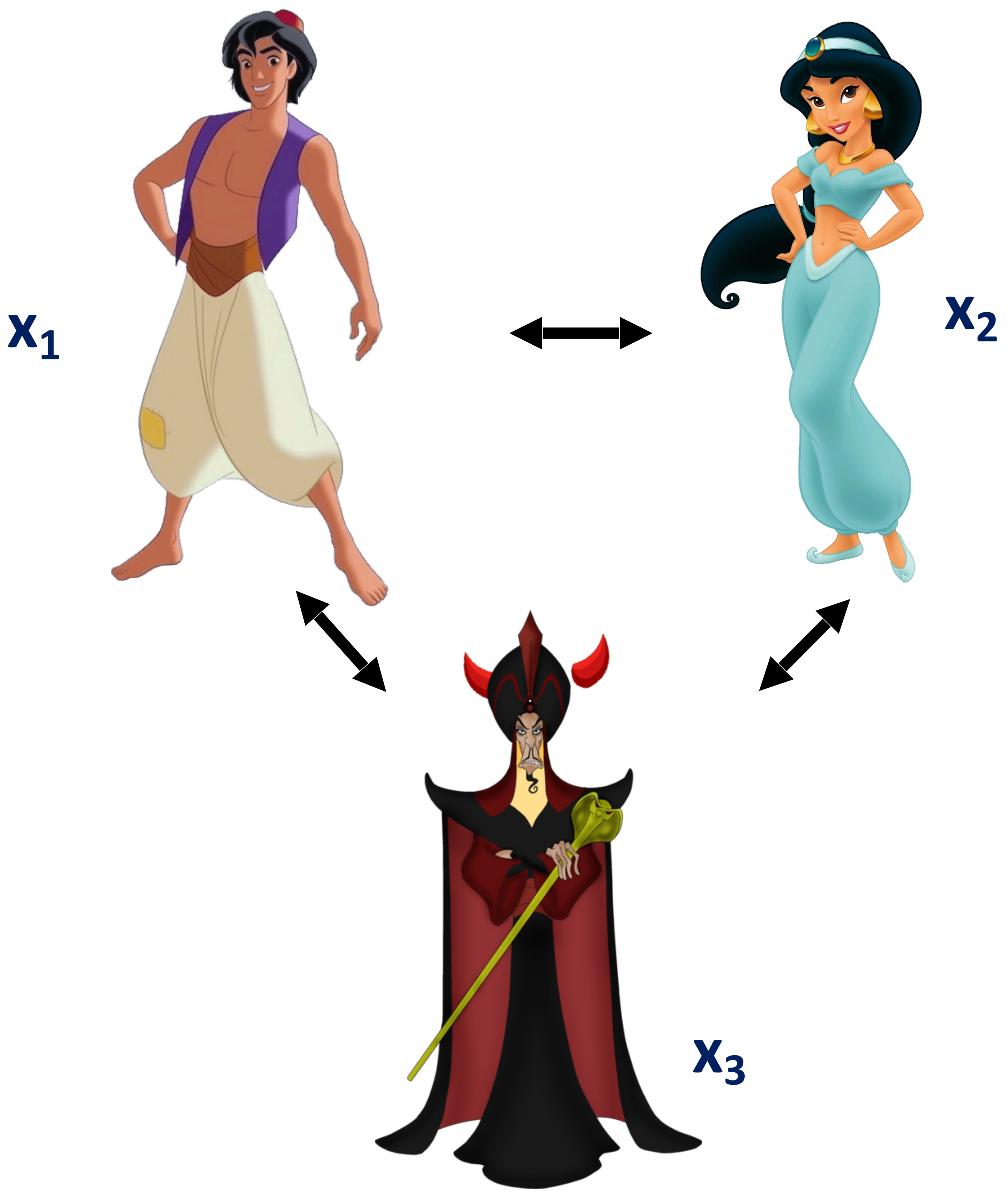
Fully-Secure MPC with Minimal Trust

Yuval Ishai, Arpita Patra, Sikhar Patranabis, **Divya Ravi**, Akshayaram Srinivasan

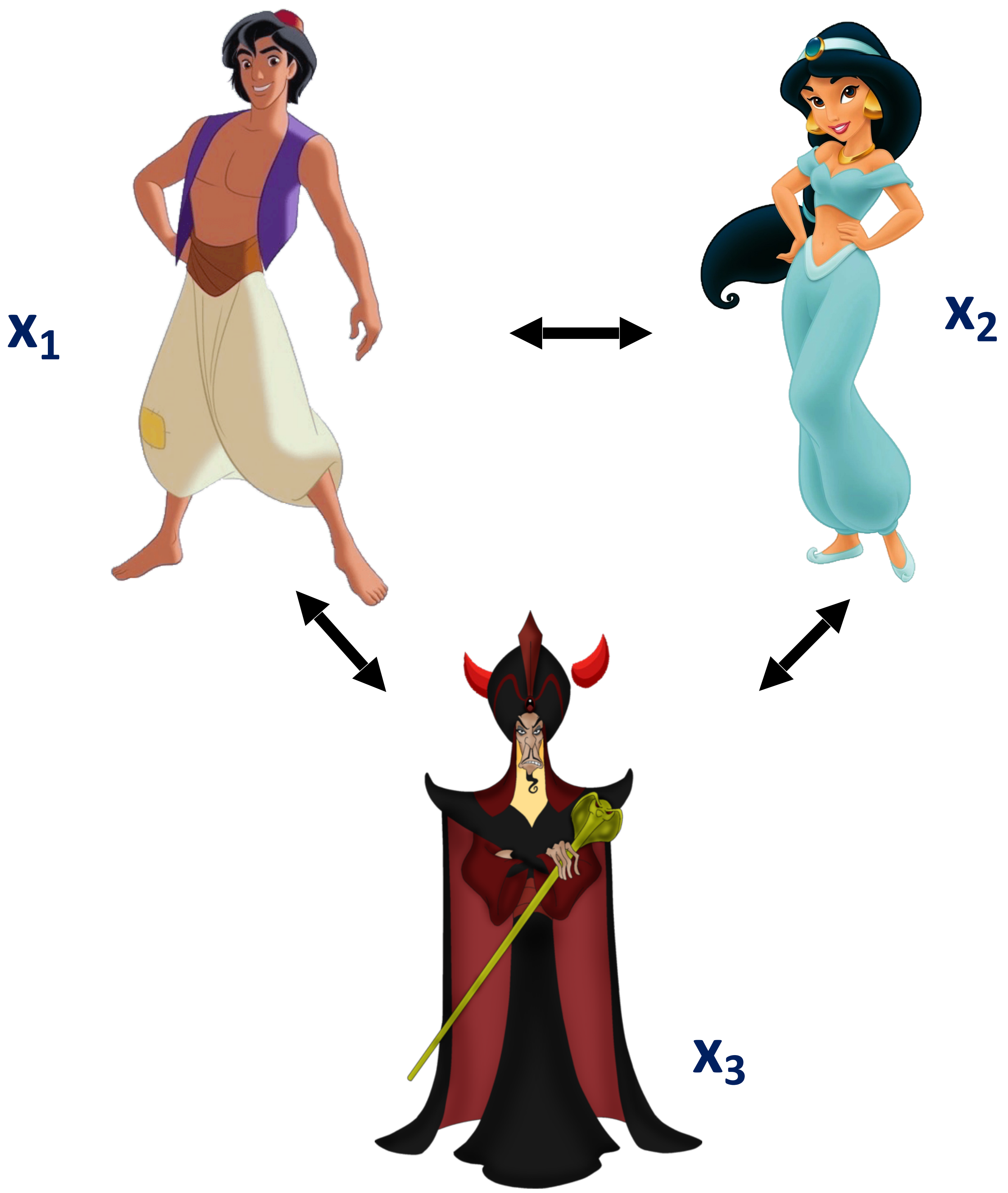
TCC 2022

Secure Multi-Party Computation



- Setup : n parties $\{P_1, P_2, \ldots., P_n\}$; t corrupt
- Input : P_i has input x_i
- Goal : Compute $\mathbf{f(x_1, x_2, x_3)}$

Secure Multi-Party Computation



- Setup : n parties $\{P_1, P_2, \ldots., P_n\}$; t corrupt
- Input : P_i has input x_i
- Goal : Compute $\mathbf{f(x_1, x_2, x_3)}$
- Properties:
  - Correctness : Protocol output = $f(x_1, x_2, x_3)$
  - Privacy : Nothing beyond function output revealed

Motivation

Motivation

Fairness and full-security (G.O.D)

Motivation

Fairness and full-security (G.O.D)

If adversary gets output, everyone does

Motivation

Fairness and full-security (G.O.D)

If adversary gets output, everyone does

Adversary cannot prevent honest parties from getting output

Motivation

Fairness and full-security (G.O.D) : impossible in dishonest majority **[Cleve86]**

If adversary gets output, everyone does

Adversary cannot prevent honest parties from getting output

Motivation

Fairness and full-security (G.O.D) : impossible in dishonest majority **[Cleve86]**

If adversary gets output, everyone does

Adversary cannot prevent honest parties from getting output

How to bypass this impossibility ?

Motivation

Fairness and full-security (G.O.D) : impossible in dishonest majority **[Cleve86]**

If adversary gets output, everyone does

Adversary cannot prevent honest parties from getting output

How to bypass this impossibility ?

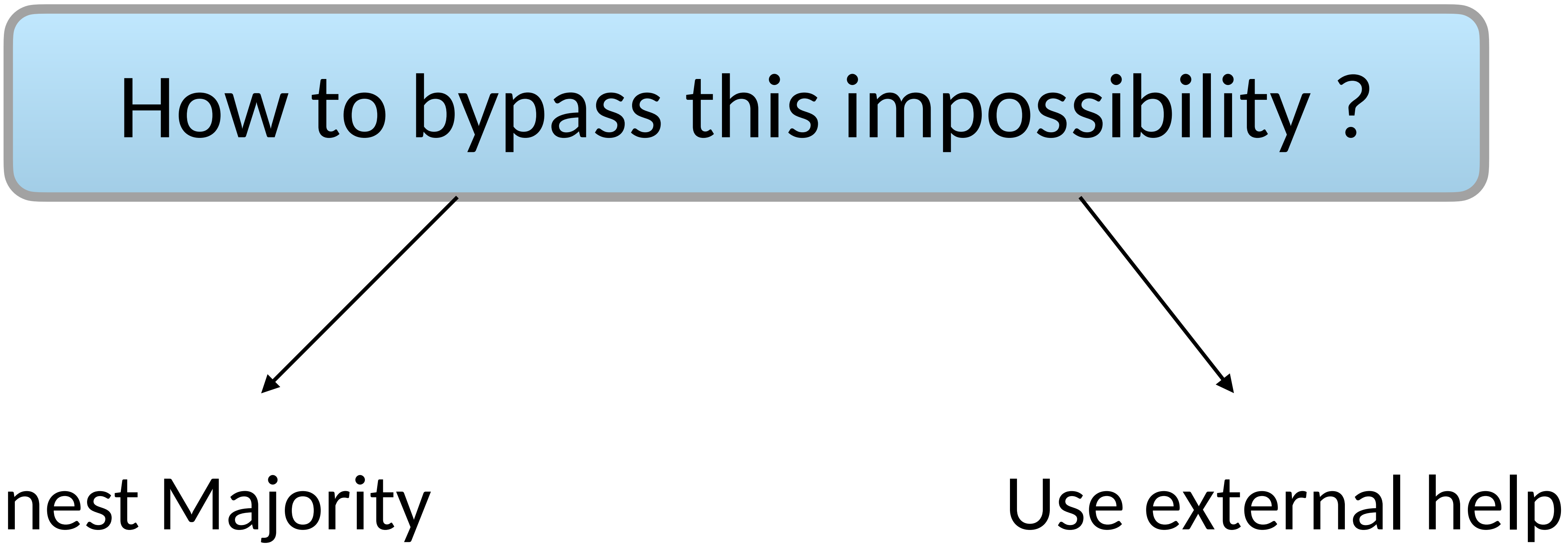
Honest Majority

Motivation

Fairness and full-security (G.O.D) : impossible in dishonest majority **[Cleve86]**

If adversary gets output, everyone does

Adversary cannot prevent honest parties from getting output



How to bypass this impossibility ?

Honest Majority

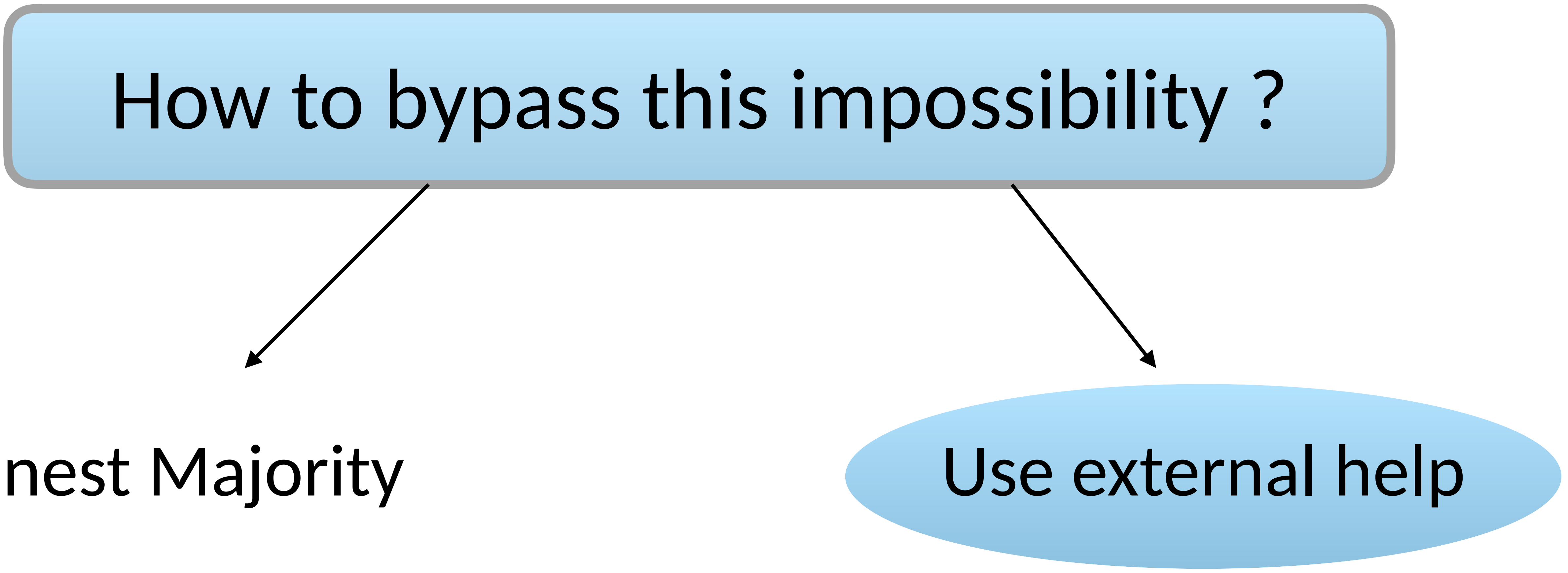
Use external help

Motivation

Fairness and full-security (G.O.D) : impossible in dishonest majority **[Cleve86]**

If adversary gets output, everyone does

Adversary cannot prevent honest parties from getting output



How to bypass this impossibility ?

Honest Majority

Use external help

Motivation

Fairness and full-security (G.O.D) : impossible in dishonest majority **[Cleve86]**

If adversary gets output, everyone does

Adversary cannot prevent honest parties from getting output

How to bypass this impossibility ?

Honest Majority

Use external help

Modelling the Trusted Party (TP)

Modelling the Trusted Party (TP)

What does the TP do?

Modelling the Trusted Party (TP)

What does the TP do?

Rule number two, I can't make
anybody fall in love with anybody else.

Modelling the Trusted Party (TP)

What does the TP do?

Rule number two, I can't make
anybody fall in love with anybody else.

How many times do you use the TP?

Modelling the Trusted Party (TP)

What does the TP do?

Rule number two, I can't make
anybody fall in love with anybody else.

How many times do you use the TP?

Modelling the Trusted Party (TP)

What does the TP do?

Rule number two, I can't make
anybody fall in love with anybody else.

How many times do you use the TP?

Trivial Solution

Modelling the Trusted Party (TP)

What does the TP do?

Rule number two, I can't make anybody fall in love with anybody else.

How many times do you use the TP?

Trivial Solution Compute the function directly

Modelling the Trusted Party (TP)

What does the TP do?

Rule number two, I can't make
anybody fall in love with anybody else.

How many times do you use the TP?

Trivial Solution

Compute the function directly

One

Modelling the Trusted Party (TP)

What does the TP do?

Rule number two, I can't make anybody fall in love with anybody else.

How many times do you use the TP?

Trivial Solution

Compute the function directly

✓ One

Modelling the Trusted Party (TP)

What does the TP do?

Rule number two, I can't make anybody fall in love with anybody else.

How many times do you use the TP?

Trivial Solution

✗ Compute the function directly

✓ One

Modelling the Trusted Party (TP)

Rule number two, I can't make
anybody fall in love with anybody else.

	What does the TP do?	How many times do you use the TP?
Trivial Solution	✗ Compute the function directly	✓ One
[IOS12]	“Small” (Independent of function)	

Modelling the Trusted Party (TP)

Rule number two, I can't make
anybody fall in love with anybody else.

	What does the TP do?	How many times do you use the TP?
Trivial Solution	✗ Compute the function directly	✓ One
[IOS12]	✓ "Small" (Independent of function)	

Modelling the Trusted Party (TP)

Rule number two, I can't make anybody fall in love with anybody else.

	What does the TP do?	How many times do you use the TP?
Trivial Solution	✗ Compute the function directly	✓ One
[IOS12]	✓ “Small” (Independent of function)	✗ n (number of parties)

Modelling the Trusted Party (TP)

Rule number two, I can't make
anybody fall in love with anybody else.

	What does the TP do?	How many times do you use the TP?
Trivial Solution	✗ Compute the function directly	✓ One
[IOS12]	✓ "Small" (Independent of function)	✗ n (number of parties)
[IOS12]		✓ One

Modelling the Trusted Party (TP)

Rule number two, I can't make
anybody fall in love with anybody else.

	What does the TP do?	How many times do you use the TP?
Trivial Solution	✗ Compute the function directly	✓ One
[IOS12]	✓ “Small” (Independent of function)	✗ n (number of parties)
[IOS12]	✗ Exponential in n	✓ One

Modelling the Trusted Party (TP)

Rule number two, I can't make
anybody fall in love with anybody else.

	What does the TP do?	How many times do you use the TP?
Trivial Solution	✗ Compute the function directly	✓ One
[IOS12]	✓ “Small” (Independent of function)	✗ n (number of parties)
[IOS12]	✗ Exponential in n	✓ One
GOAL	✓ **Small** poly(n, λ)	✓ **One**

Modelling the Trusted Party (TP)

TP realized by
Cloud service provider
charging fees ;
Large-scale Honest Majority
MPC

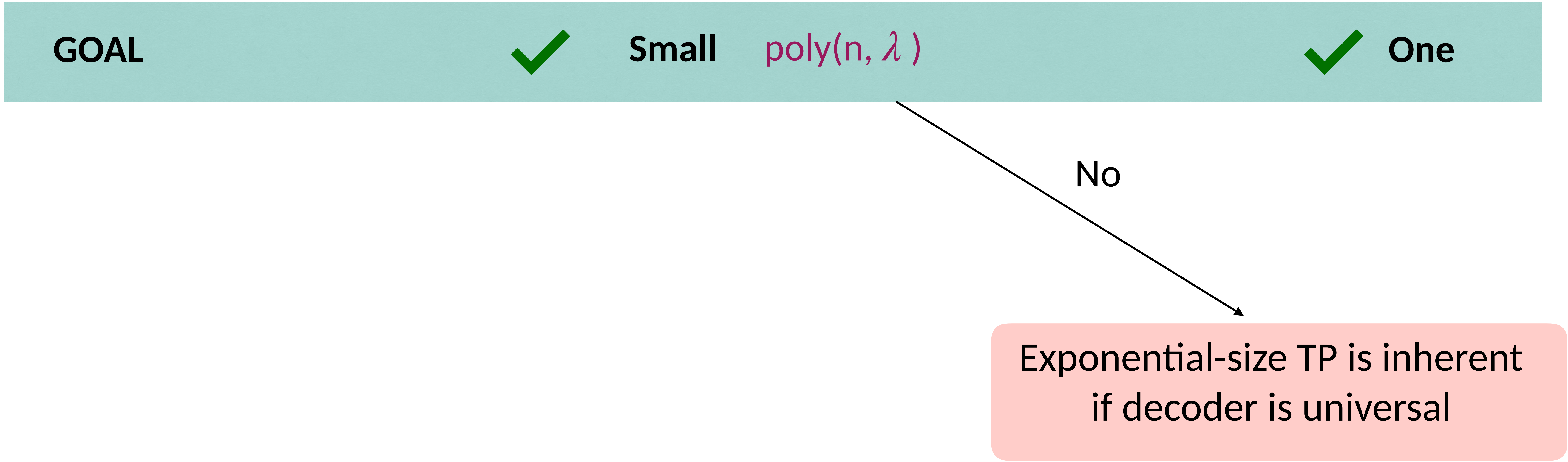
Rule number two, I can't make anybody fall in love with anybody else.

	What does the TP do?	How many times do you use the TP?
Trivial Solution	✗ Compute the function directly	✓ One
[IOS12]	✓ "Small" (Independent of function)	✗ n (number of parties)
[IOS12]	✗ Exponential in n	✓ One
GOAL	✓ **Small** poly(n, λ)	✓ **One**

GOAL ✓ **Small** poly(n, λ) ✓ **One**

GOAL ✓ **Small** poly(n, λ) ✓ **One**

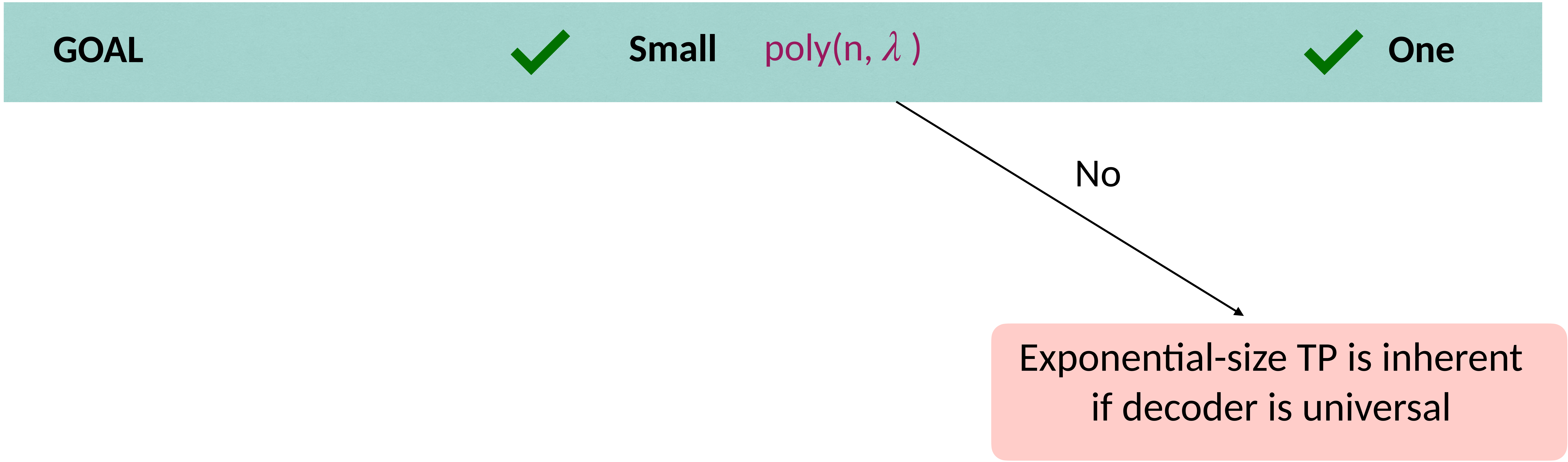
No



GOAL ✓ **Small** poly(n, λ) ✓ **One**

No

Exponential-size TP is inherent
if decoder is universal

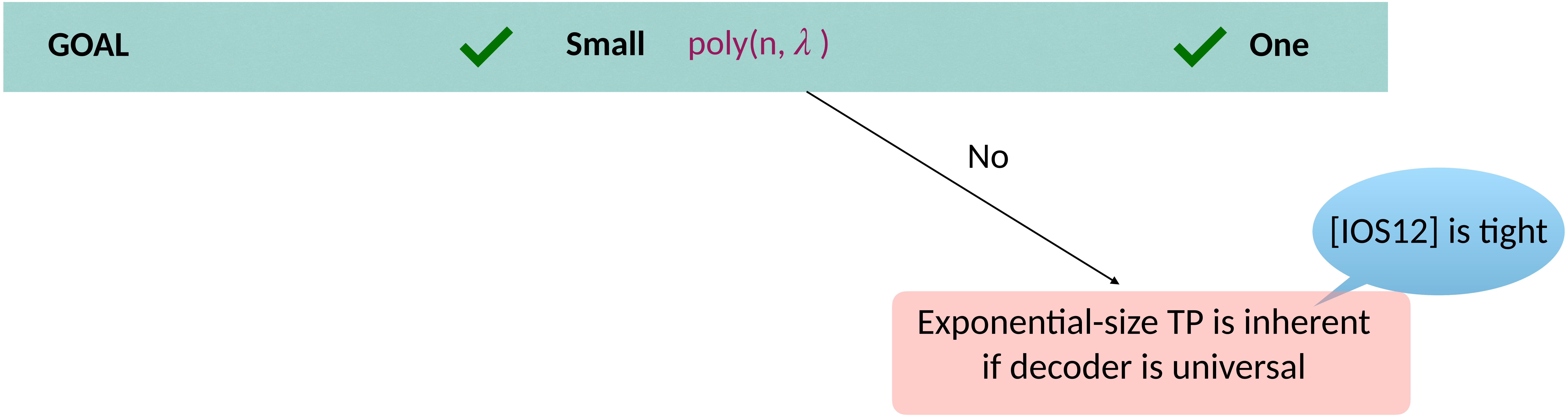


GOAL ✓ **Small** poly(n, λ) ✓ **One**

No

Exponential-size TP is inherent if decoder is universal

Universal Decoder: Function-Independent Computation done to derive output from the TP response



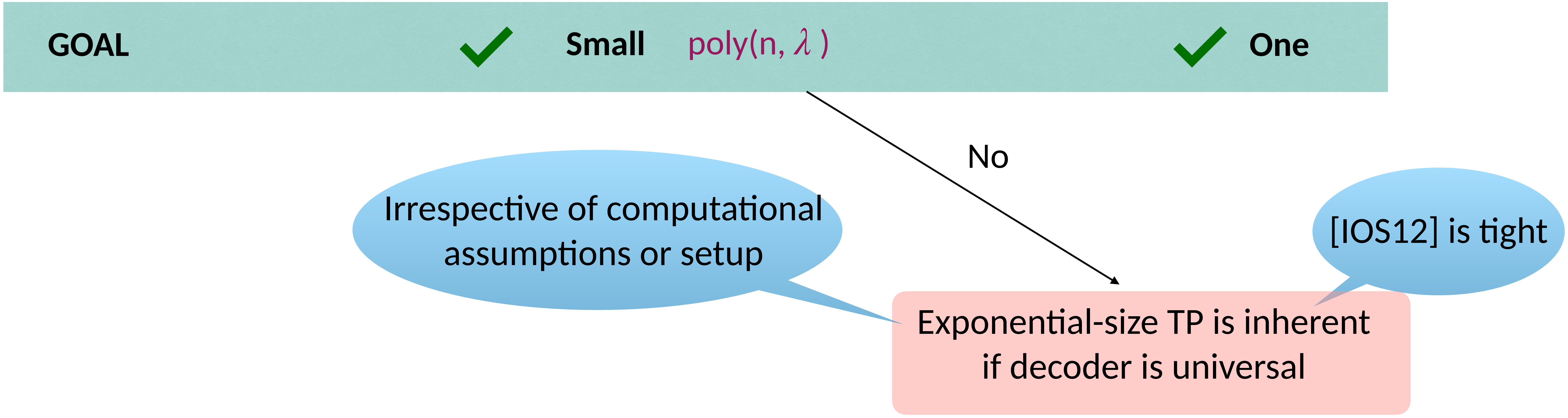
GOAL ✓ **Small** poly(n, λ) ✓ **One**

No

[IOS12] is tight

Exponential-size TP is inherent if decoder is universal

Universal Decoder: Function-Independent Computation done to derive output from the TP response



GOAL ✓ **Small** poly(n, λ) ✓ **One**

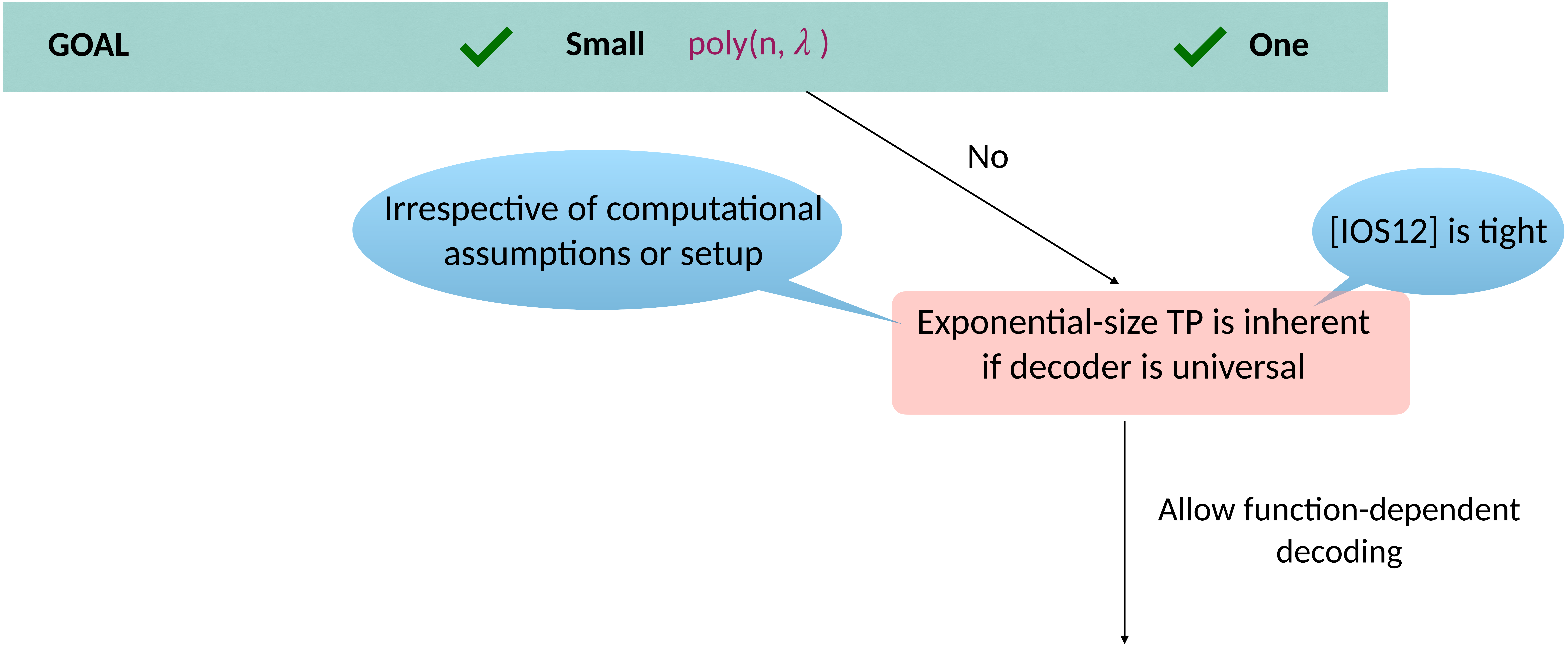
No

Exponential-size TP is inherent if decoder is universal

Irrespective of computational assumptions or setup

[IOS12] is tight

Universal Decoder: Function-Independent Computation done to derive output from the TP response



GOAL ✓ **Small** poly(n, λ) ✓ **One**

No

Irrespective of computational assumptions or setup

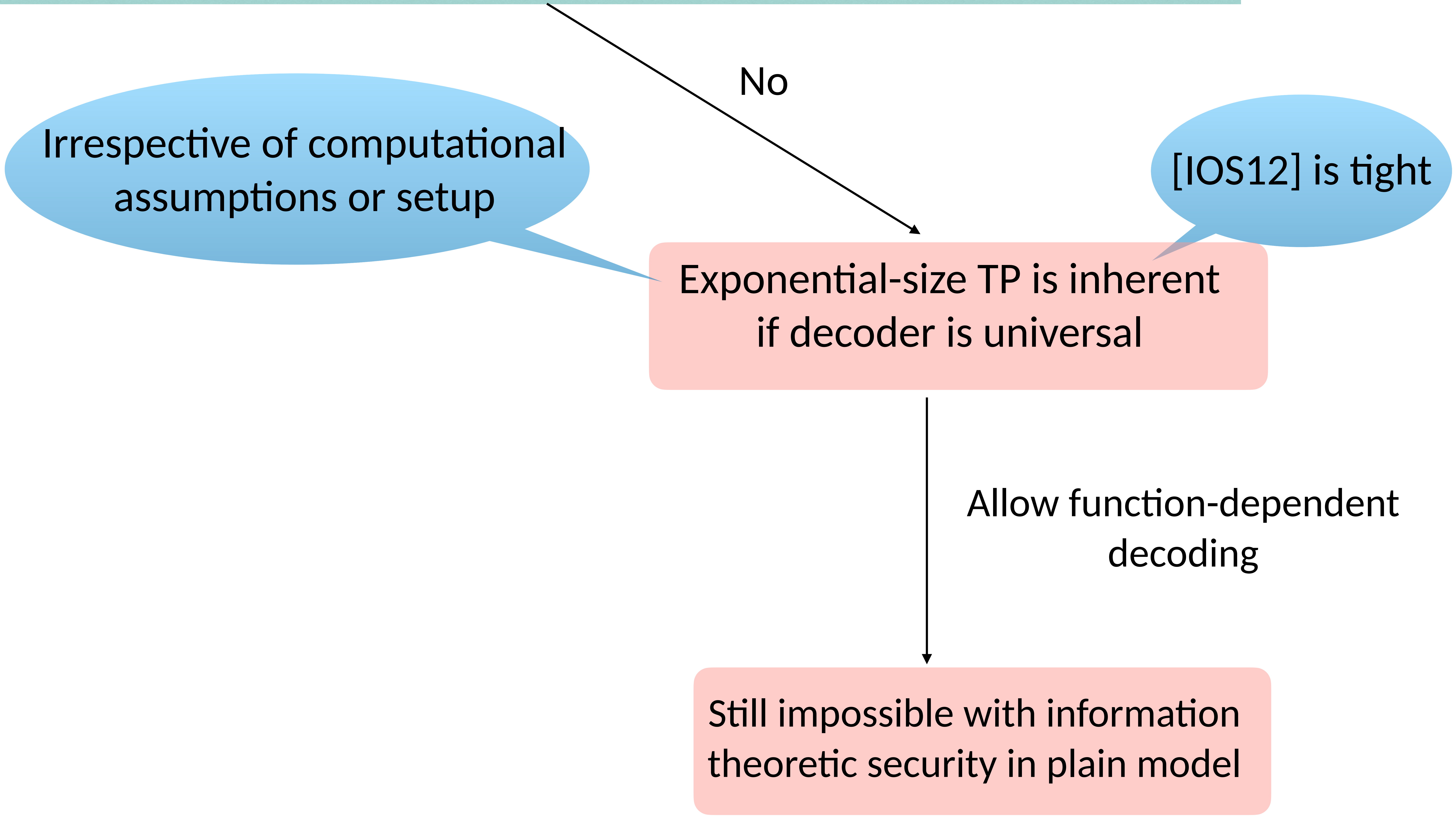
[IOS12] is tight

Exponential-size TP is inherent if decoder is universal

Allow function-dependent decoding

Universal Decoder: Function-Independent Computation done to derive output from the TP response

GOAL ✓ **Small** poly(n, λ) ✓ **One**



No

Irrespective of computational assumptions or setup

[IOS12] is tight

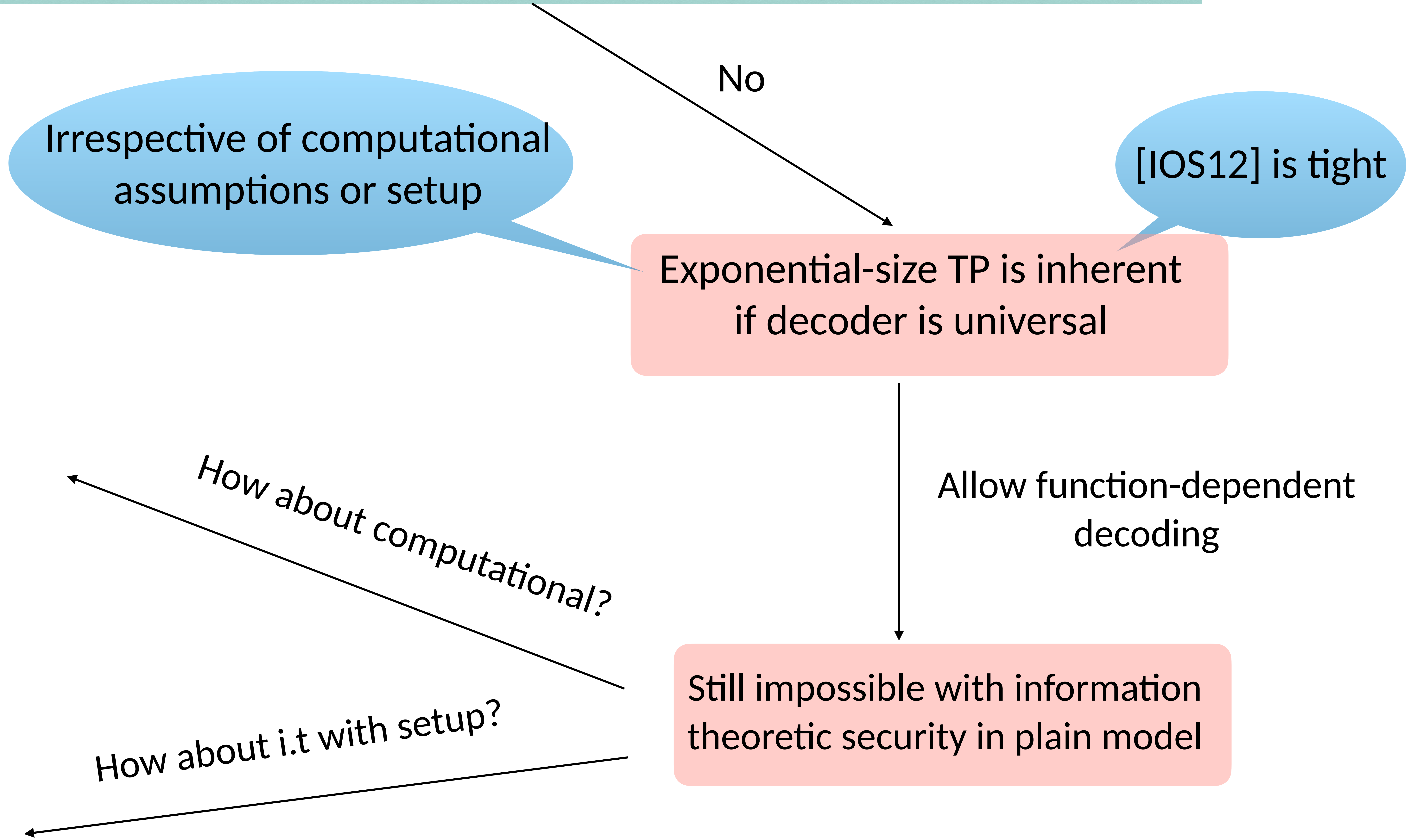
Exponential-size TP is inherent if decoder is universal

Allow function-dependent decoding

Still impossible with information theoretic security in plain model

Universal Decoder: Function-Independent Computation done to derive output from the TP response

GOAL ✓ **Small** poly(n, λ) ✓ **One**



No

Irrespective of computational assumptions or setup

[IOS12] is tight

Exponential-size TP is inherent if decoder is universal

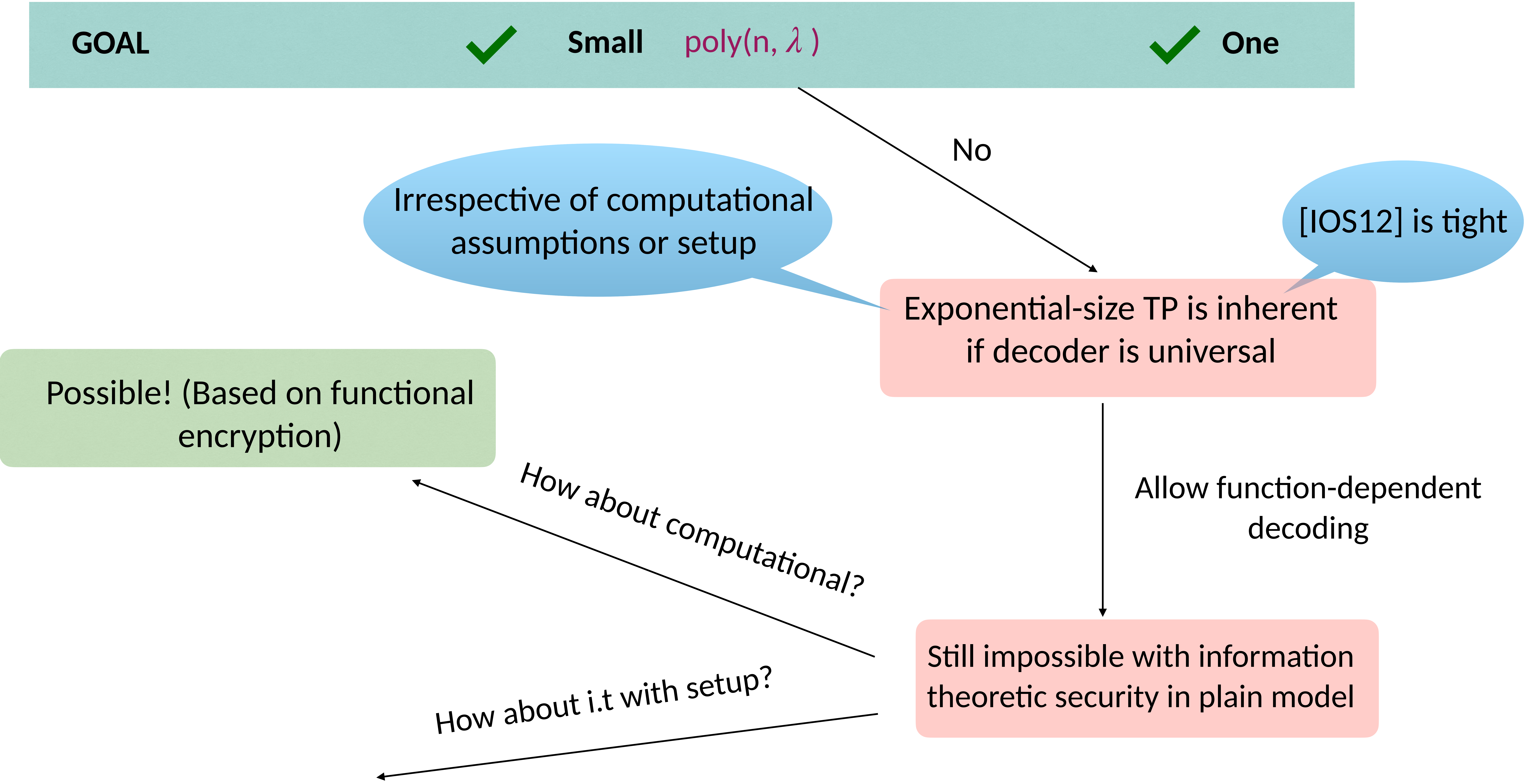
Allow function-dependent decoding

Still impossible with information theoretic security in plain model

How about computational?

How about i.t with setup?

Universal Decoder: Function-Independent Computation done to derive output from the TP response



GOAL ✓ Small poly(n, λ) ✓ One

No

Irrespective of computational assumptions or setup

[IOS12] is tight

Exponential-size TP is inherent if decoder is universal

Allow function-dependent decoding

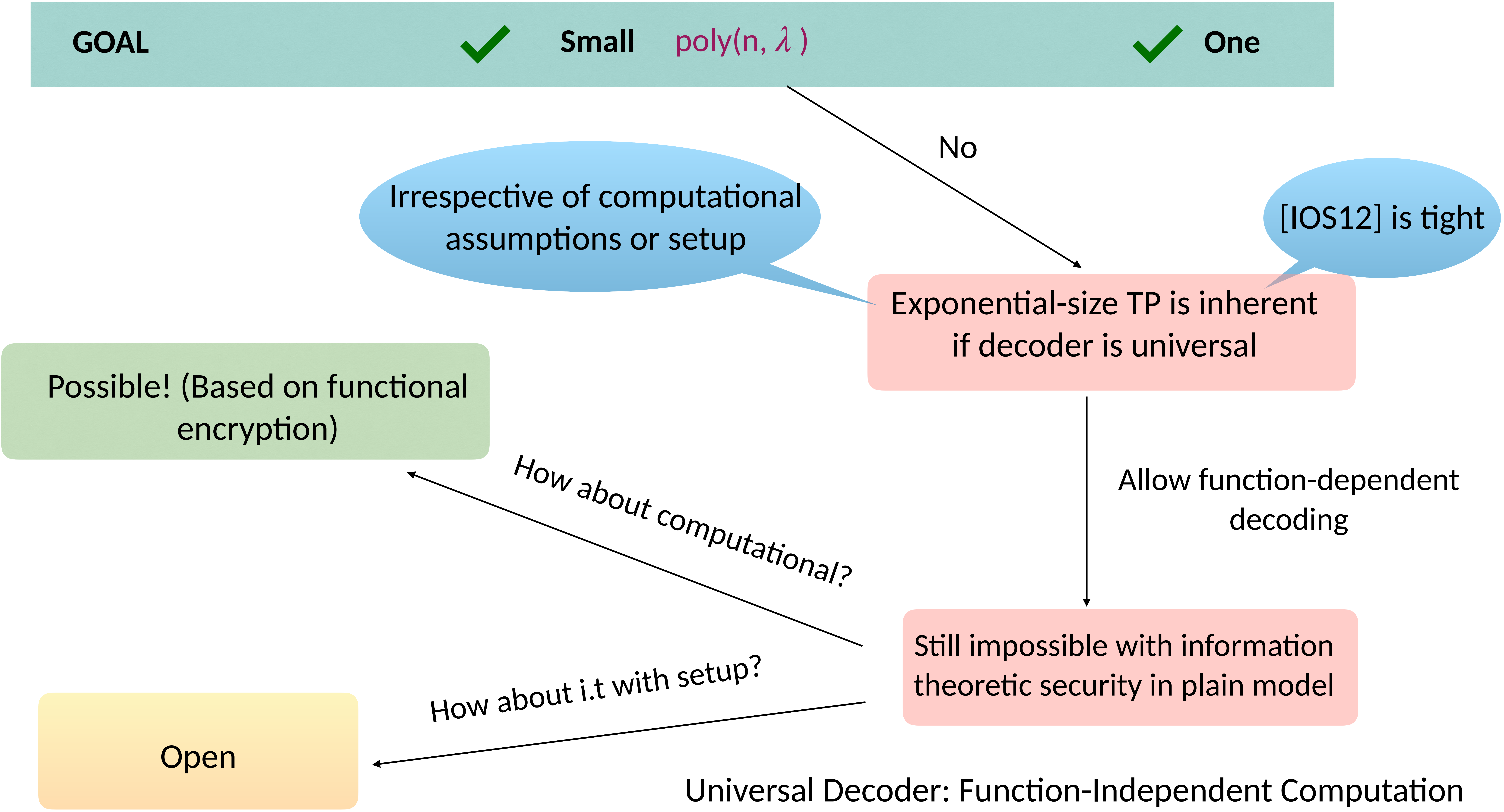
Still impossible with information theoretic security in plain model

How about computational?

Possible! (Based on functional encryption)

How about i.t with setup?

Universal Decoder: Function-Independent Computation done to derive output from the TP response



GOAL ✓ **Small** poly(n, λ) ✓ **One**

No

Irrespective of computational assumptions or setup

[IOS12] is tight

Exponential-size TP is inherent if decoder is universal

Allow function-dependent decoding

Still impossible with information theoretic security in plain model

How about computational?

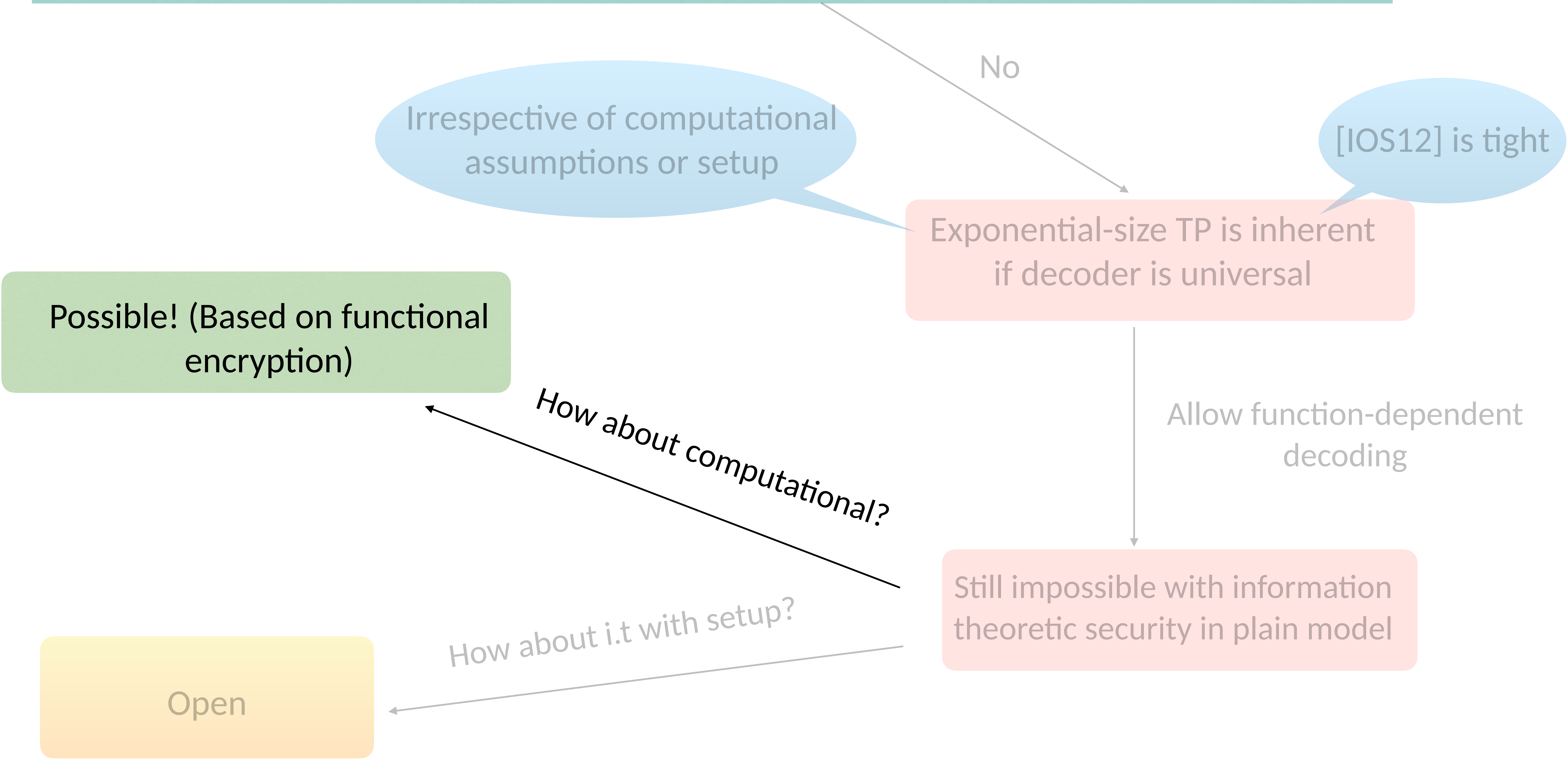
Possible! (Based on functional encryption)

How about i.t with setup?

Open

Universal Decoder: Function-Independent Computation done to derive output from the TP response

GOAL ✓ **Small** poly(n, λ) ✓ **One**



No

Irrespective of computational assumptions or setup

[IOS12] is tight

Exponential-size TP is inherent if decoder is universal

Possible! (Based on functional encryption)

Allow function-dependent decoding

How about computational?

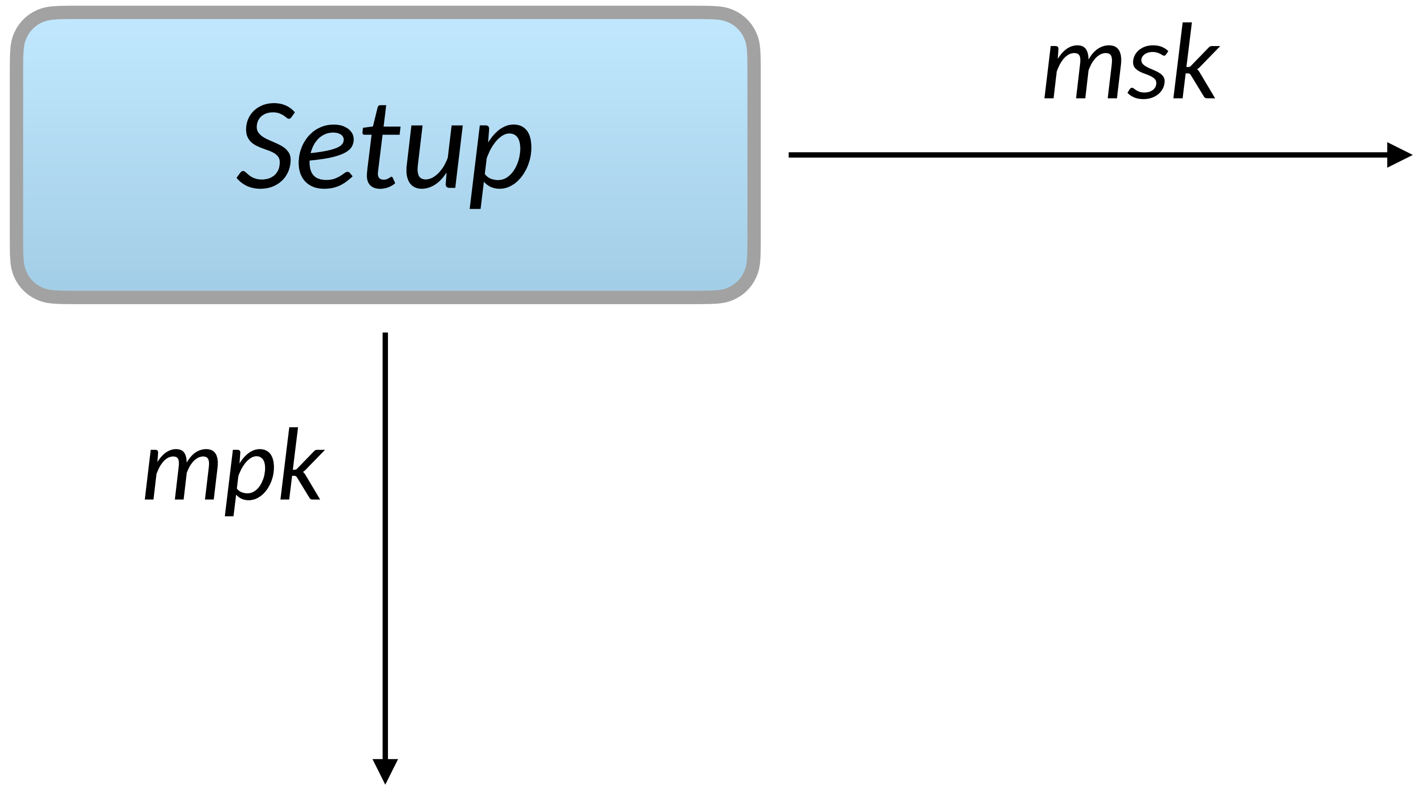
Still impossible with information theoretic security in plain model

How about i.t with setup?

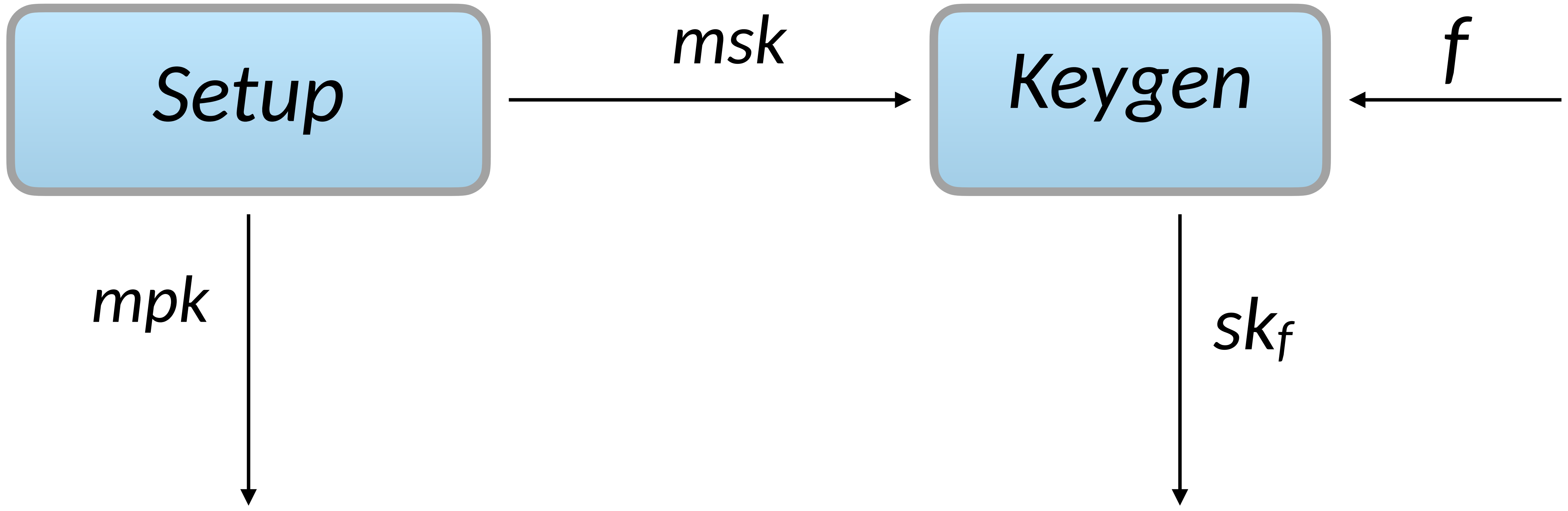
Open

Functional Encryption (FE)

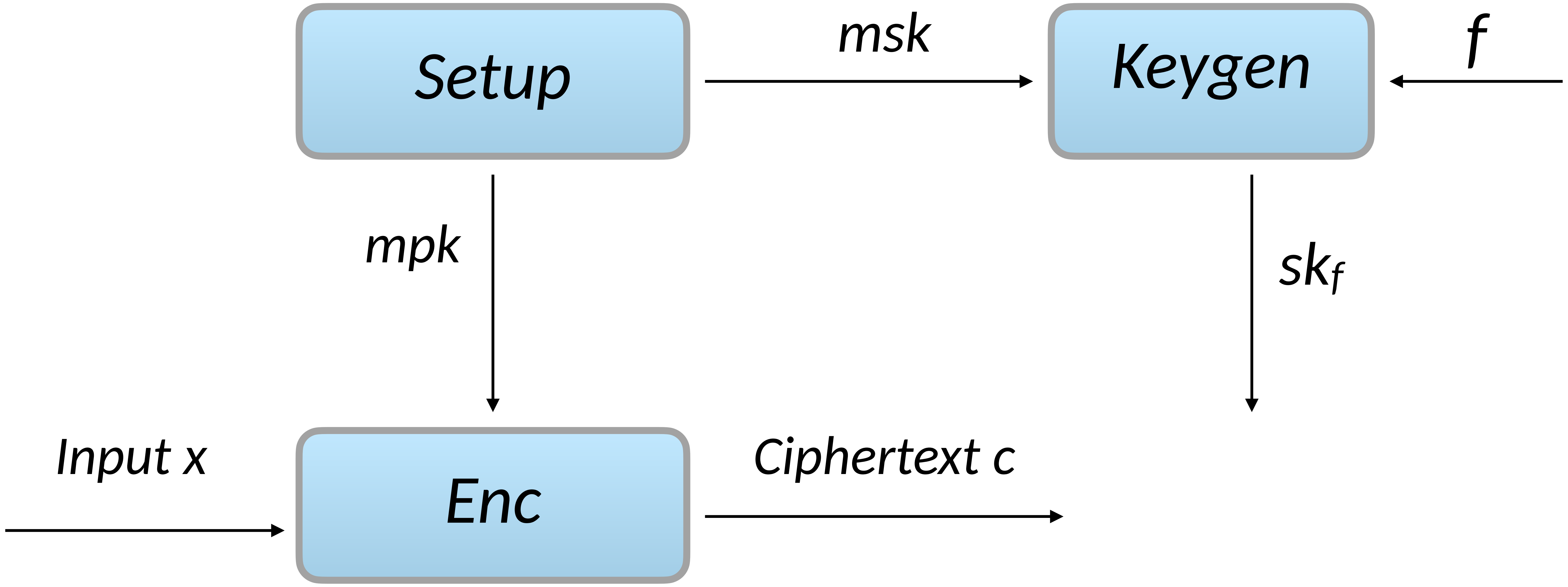
Functional Encryption (FE)



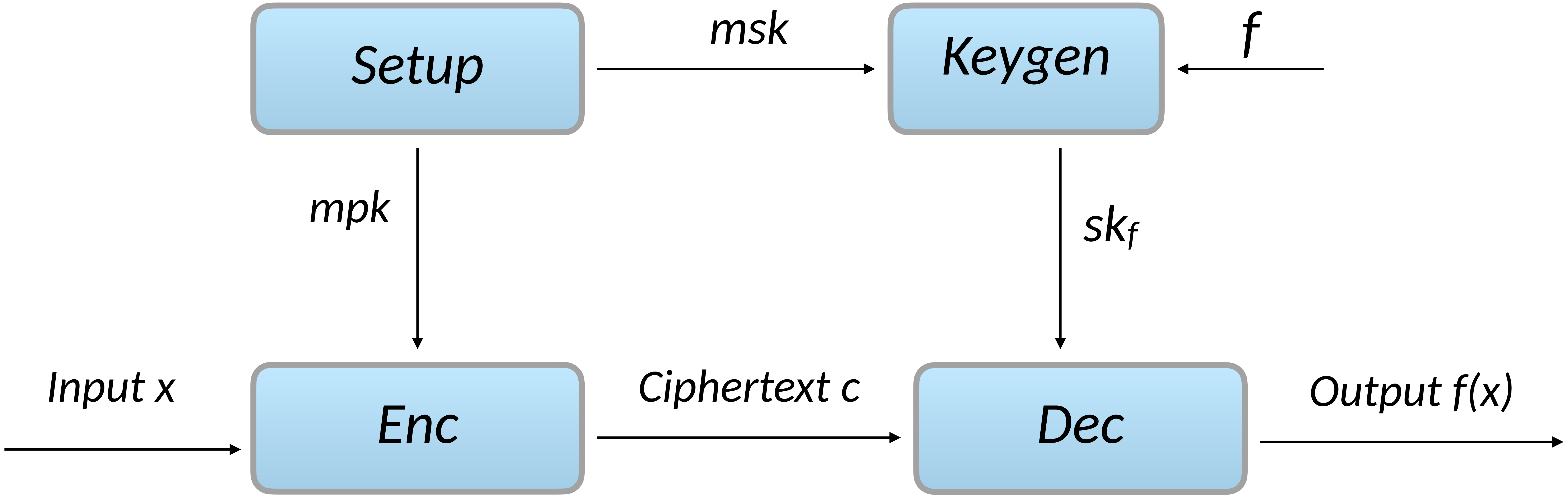
Functional Encryption (FE)



Functional Encryption (FE)



Functional Encryption (FE)



Fully-Secure MPC with 1 call to small TP

Fully-Secure MPC with 1 call to small TP

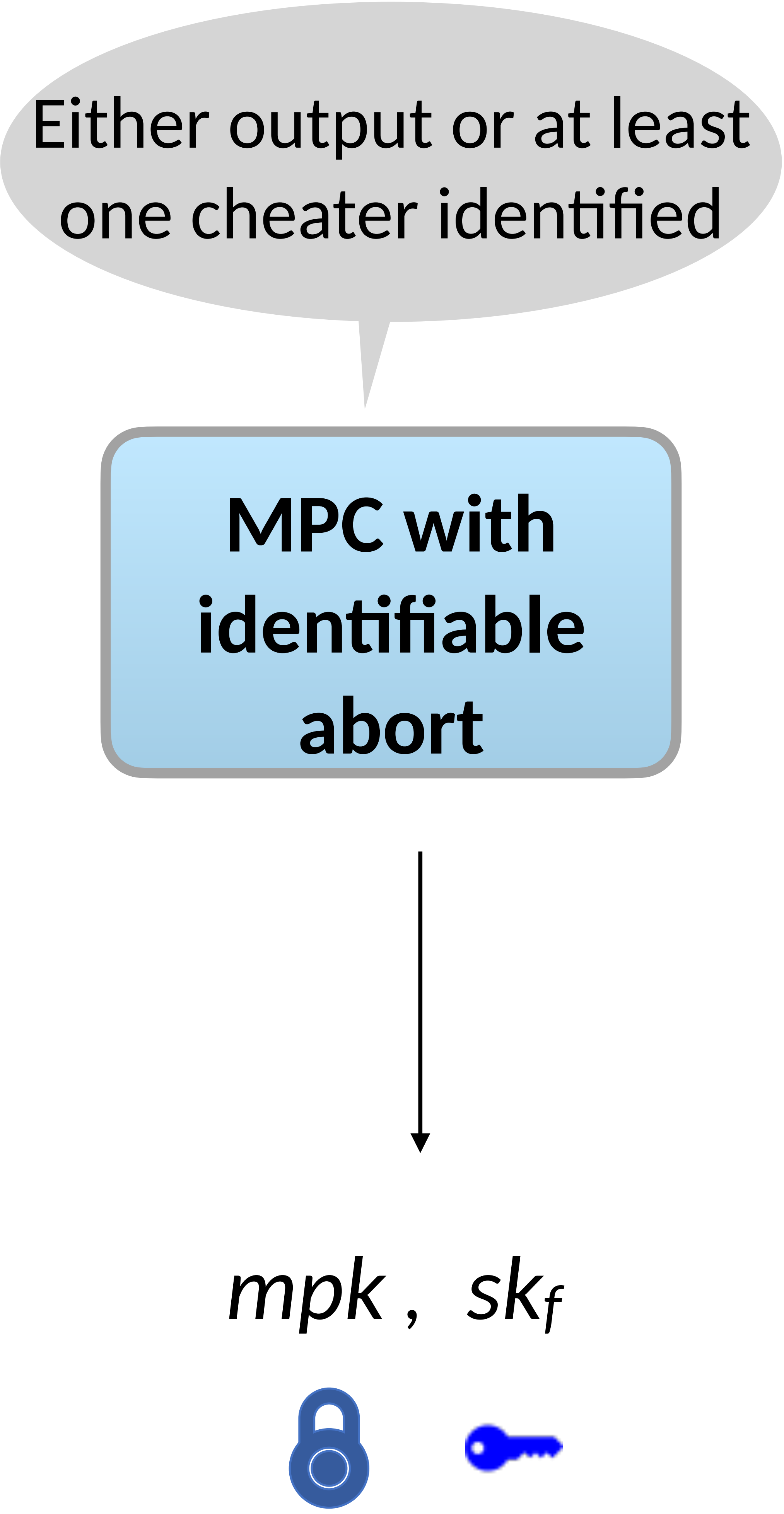
MPC with identifiable abort

Fully-Secure MPC with 1 call to small TP

Either output or at least one cheater identified

MPC with identifiable abort

Fully-Secure MPC with 1 call to small TP



Either output or at least one cheater identified

MPC with identifiable abort

mpk , sk_f

Fully-Secure MPC with 1 call to small TP

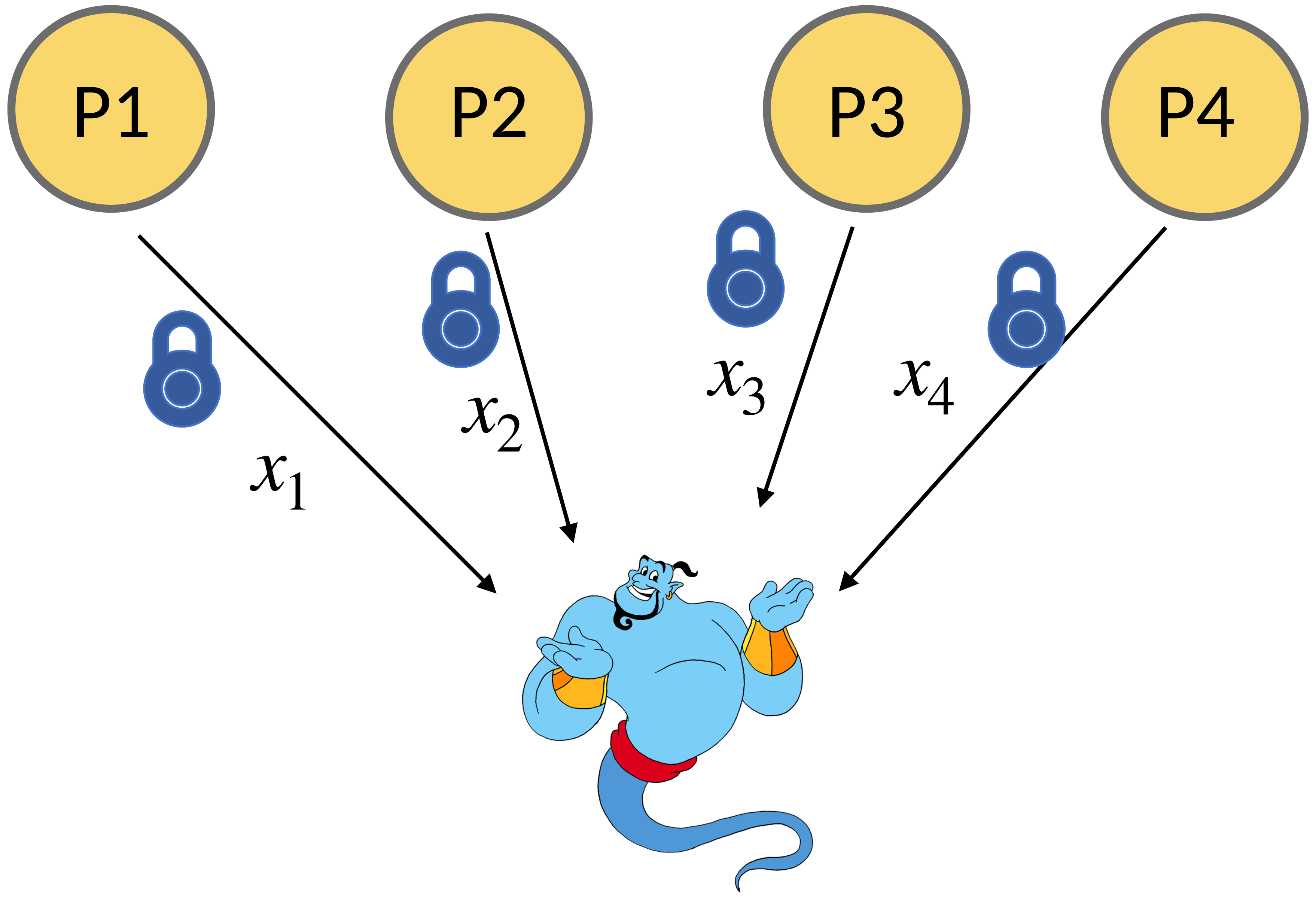
Either output or at least one cheater identified

MPC with identifiable abort

↓

mpk , sk_f

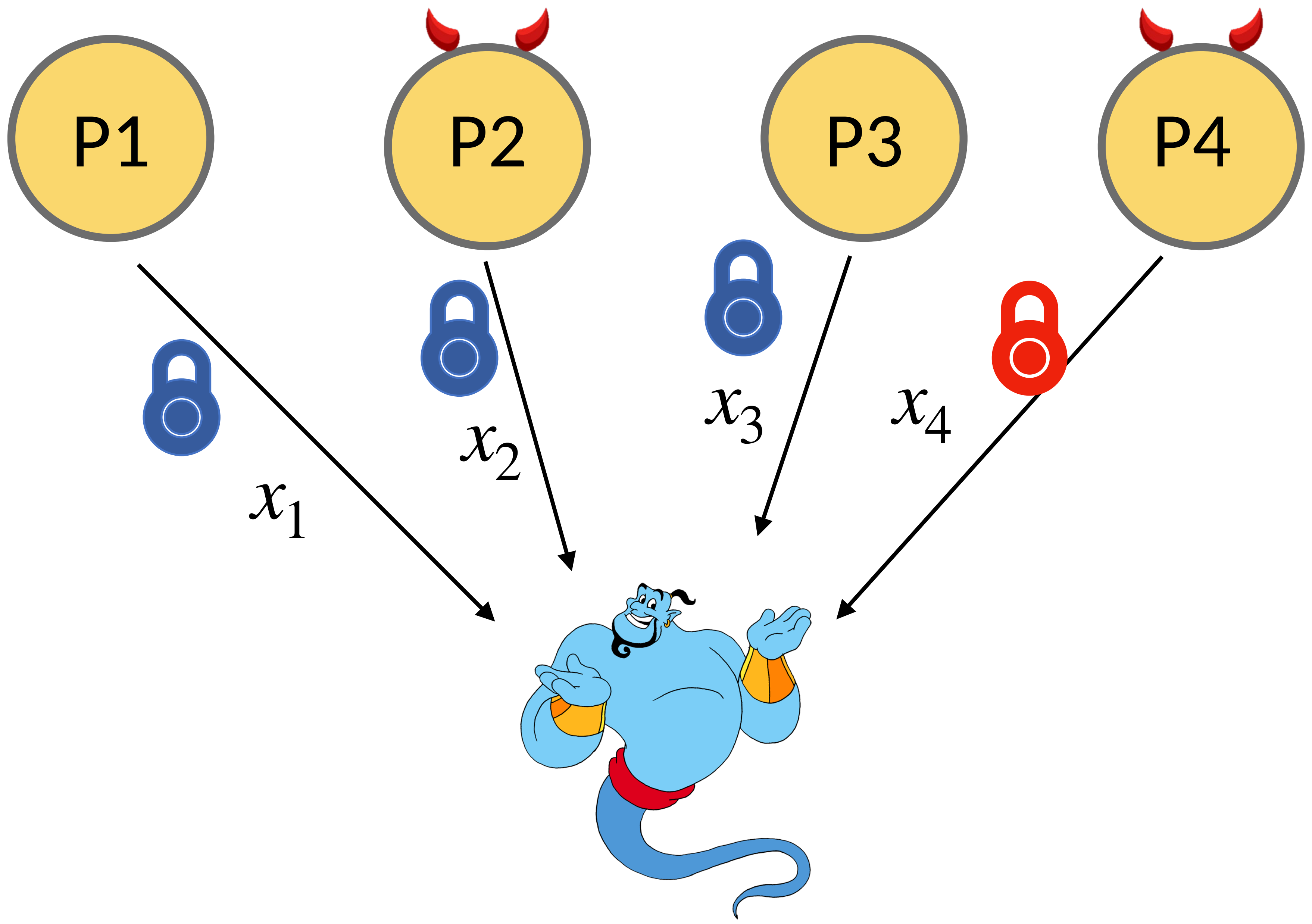
🔒 🔑



Fully-Secure MPC with 1 call to small TP

MPC with identifiable abort

mpk , sk_f

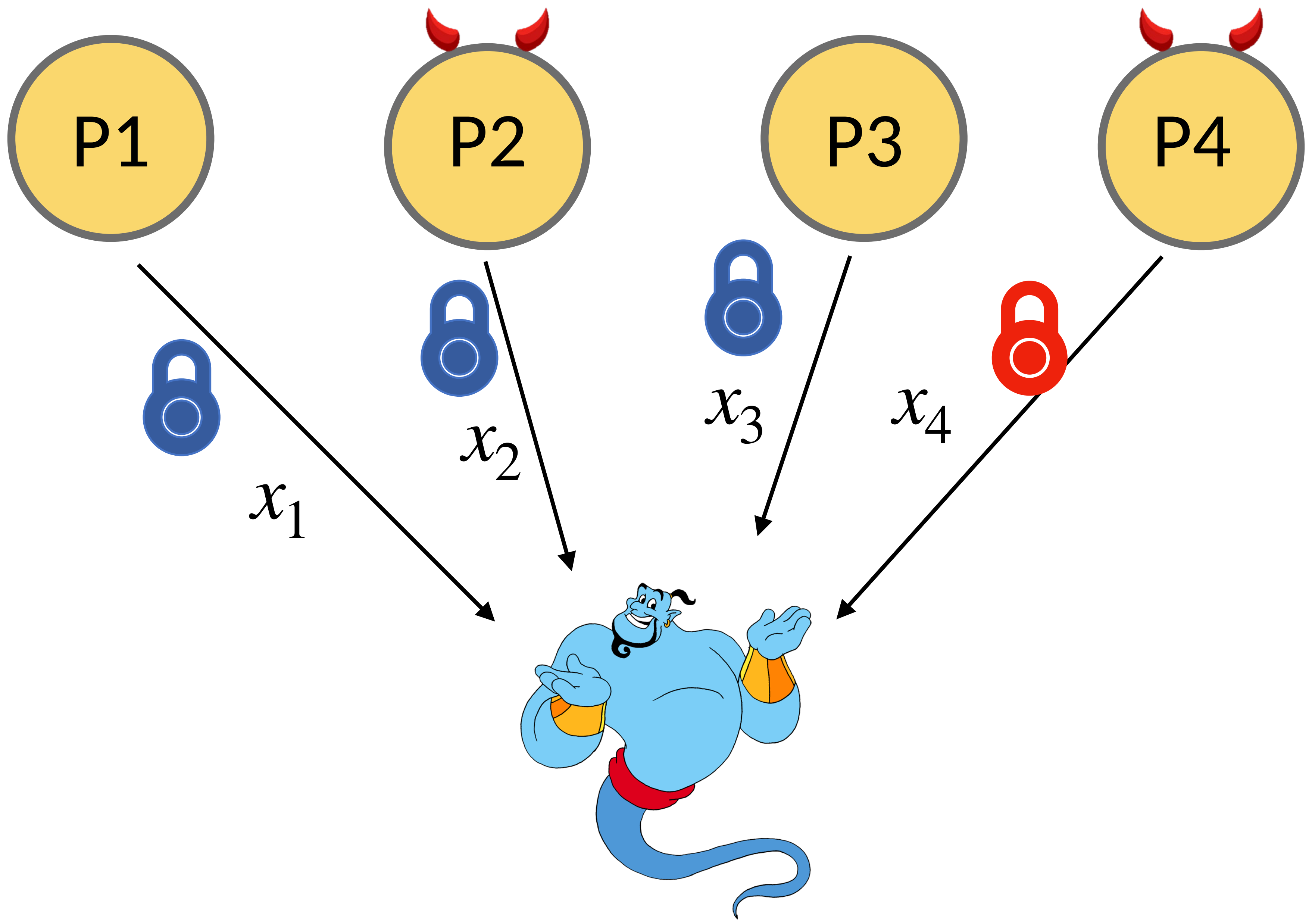


Fully-Secure MPC with 1 call to small TP

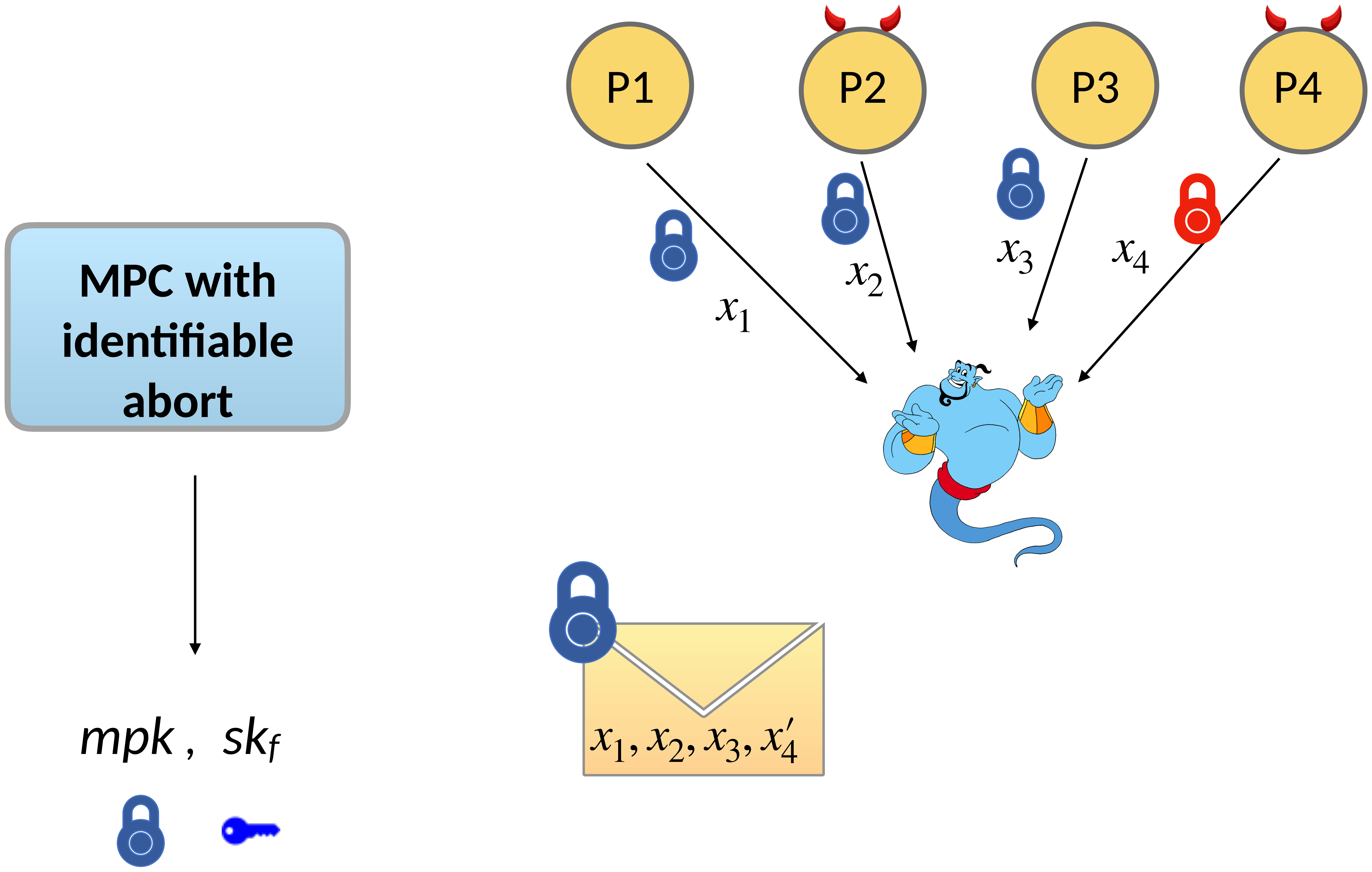
MPC with identifiable abort

↓

mpk , *sk$_f$*



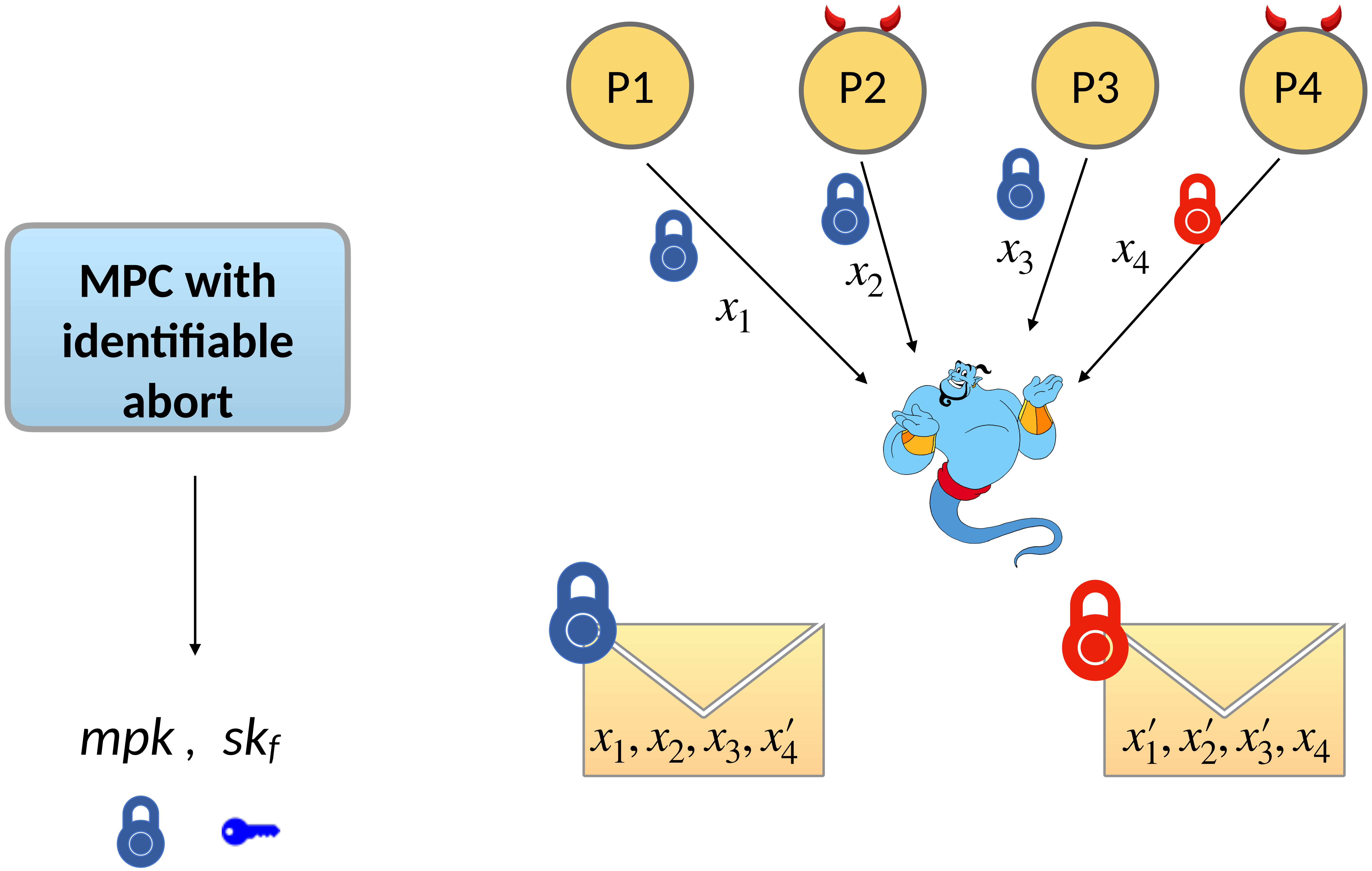
Fully-Secure MPC with 1 call to small TP



MPC with identifiable abort

mpk , *sk*$_f$

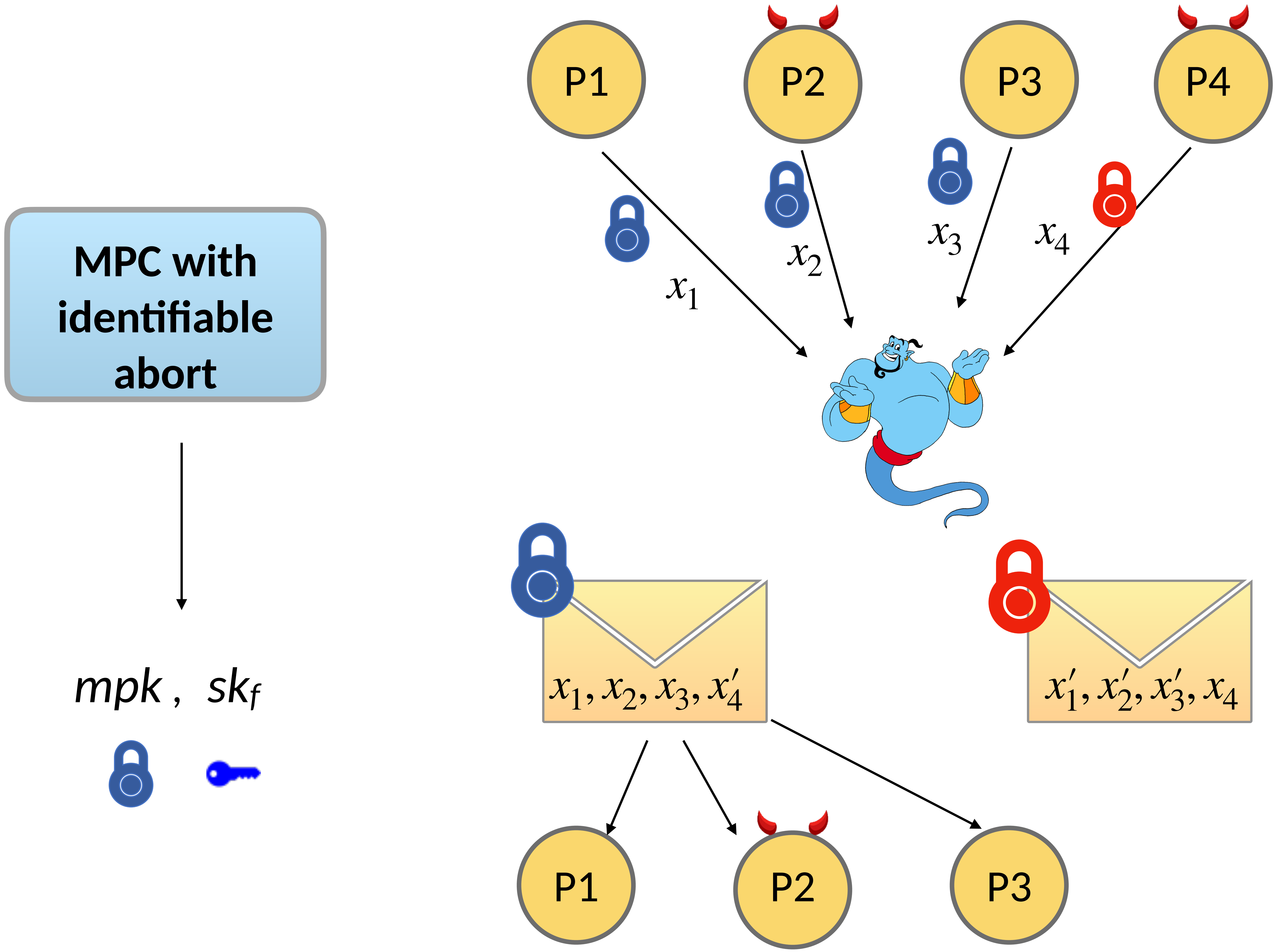
Fully-Secure MPC with 1 call to small TP



MPC with identifiable abort

mpk , *sk*$_f$

Fully-Secure MPC with 1 call to small TP

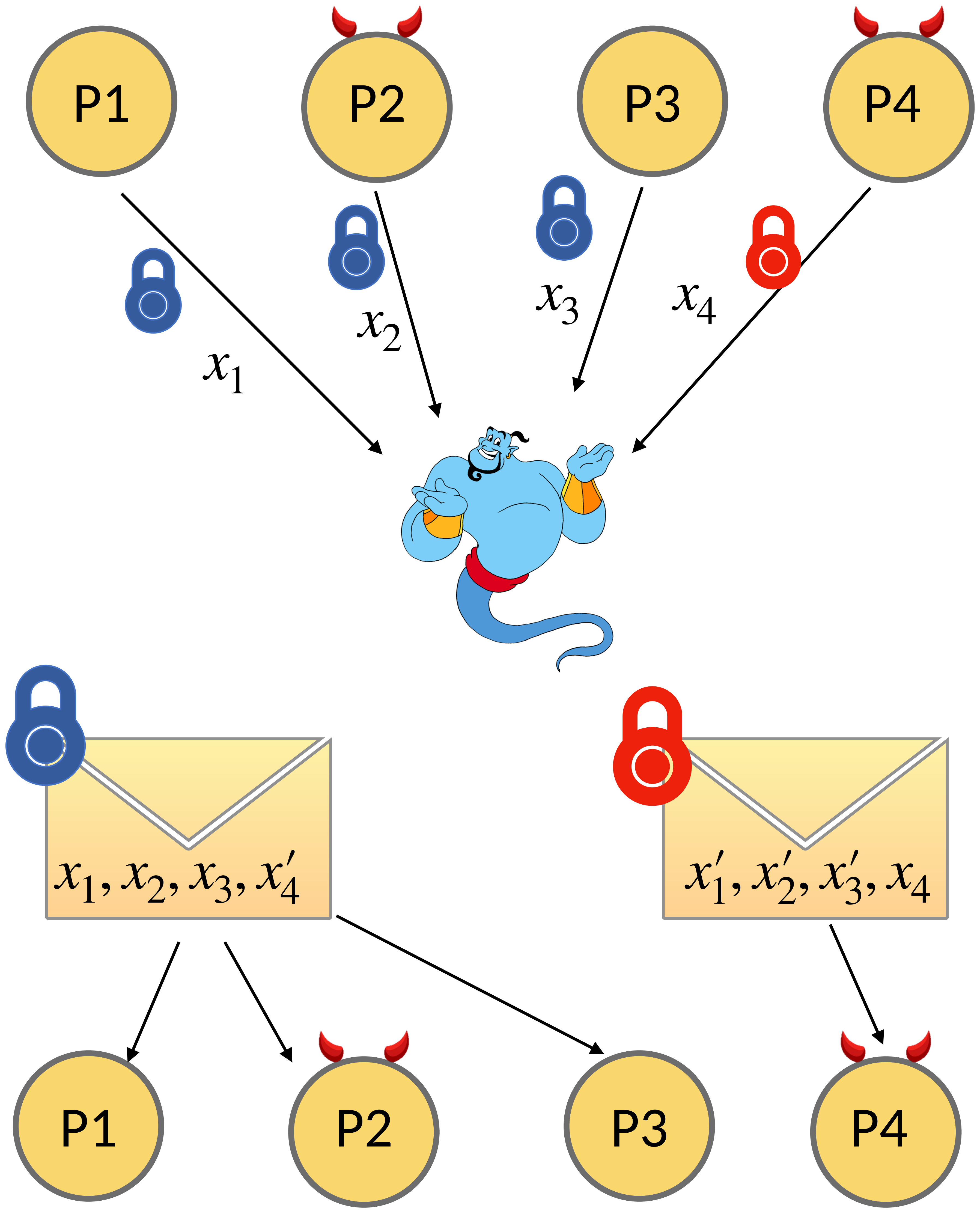


Fully-Secure MPC with 1 call to small TP

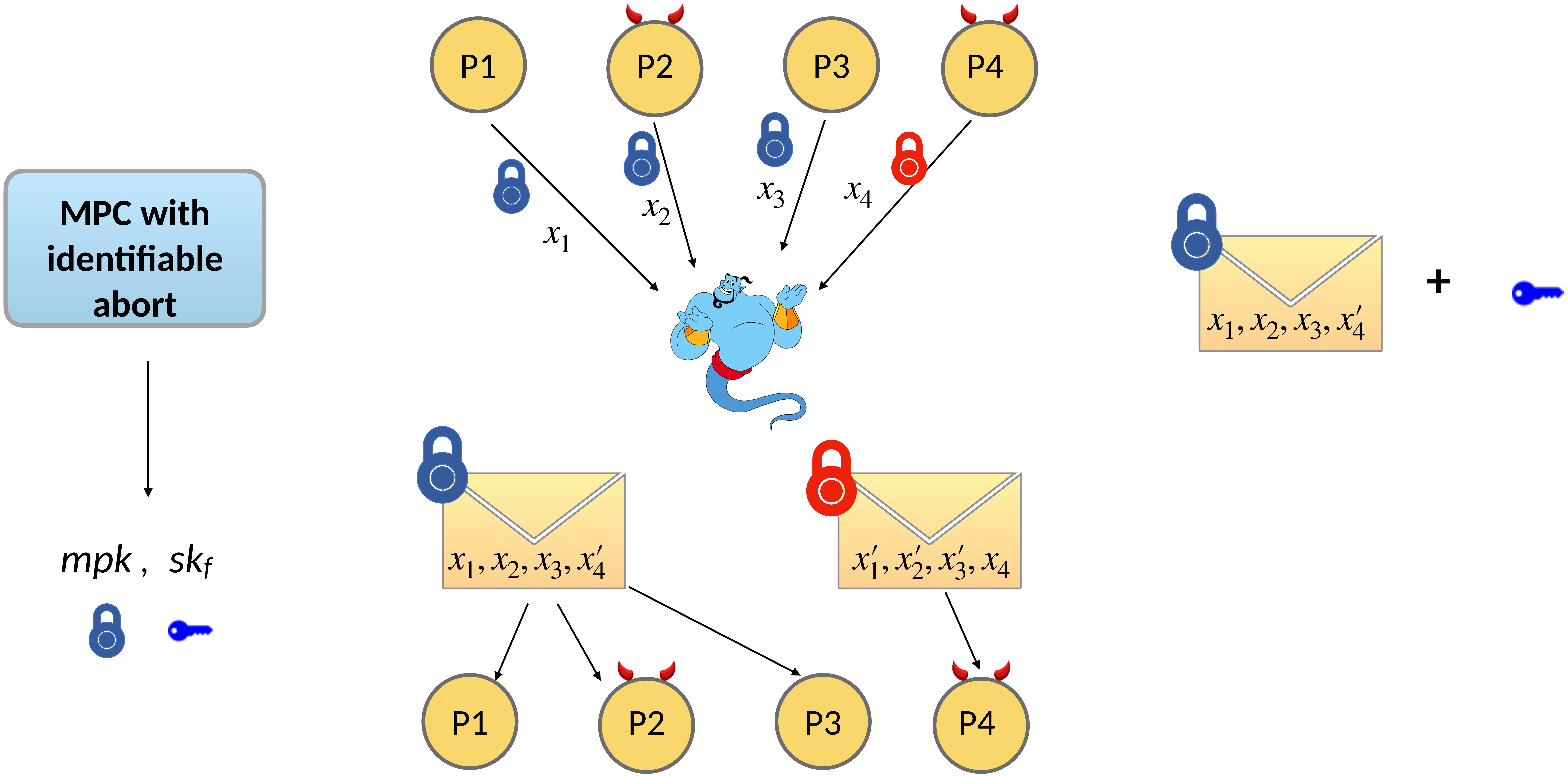
MPC with identifiable abort

↓

mpk , sk_f



Fully-Secure MPC with 1 call to small TP



MPC with identifiable abort

mpk , *sk*$_f$

P1 P2 P3 P4

x_1 x_2 x_3 x_4

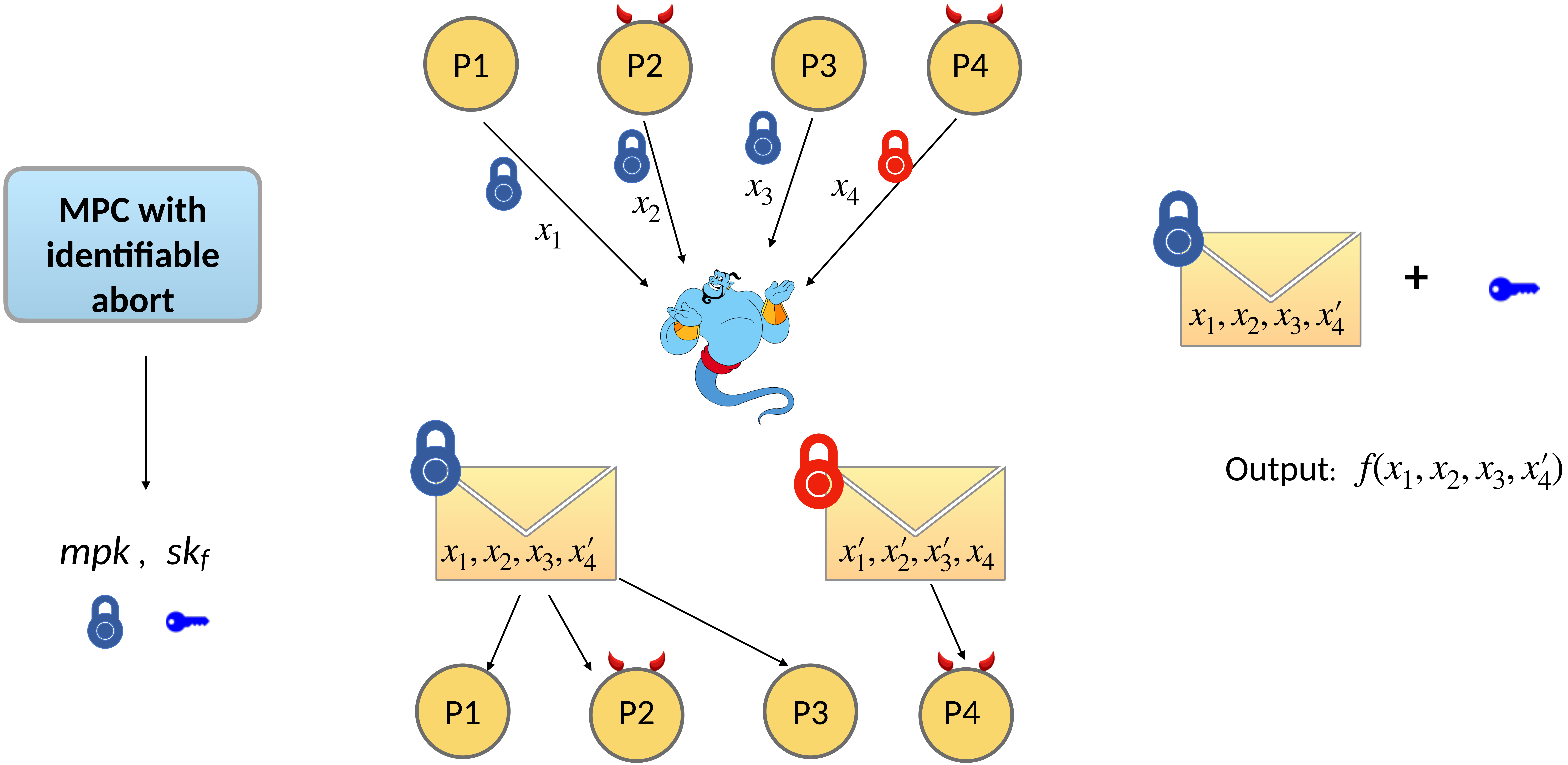
x_1, x_2, x_3, x_4'

x_1', x_2', x_3', x_4

P1 P2 P3 P4

x_1, x_2, x_3, x_4' +

Fully-Secure MPC with 1 call to small TP

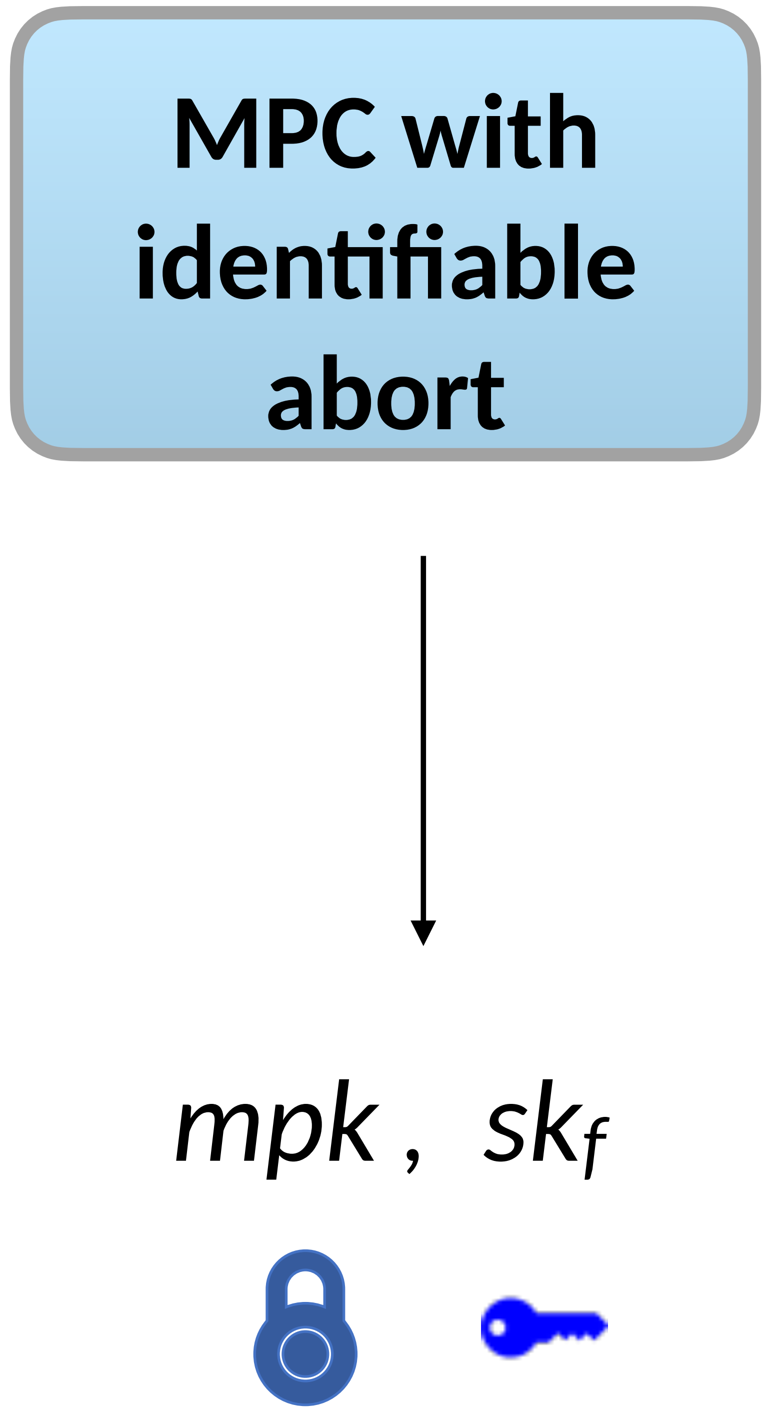


MPC with identifiable abort

mpk , sk_f

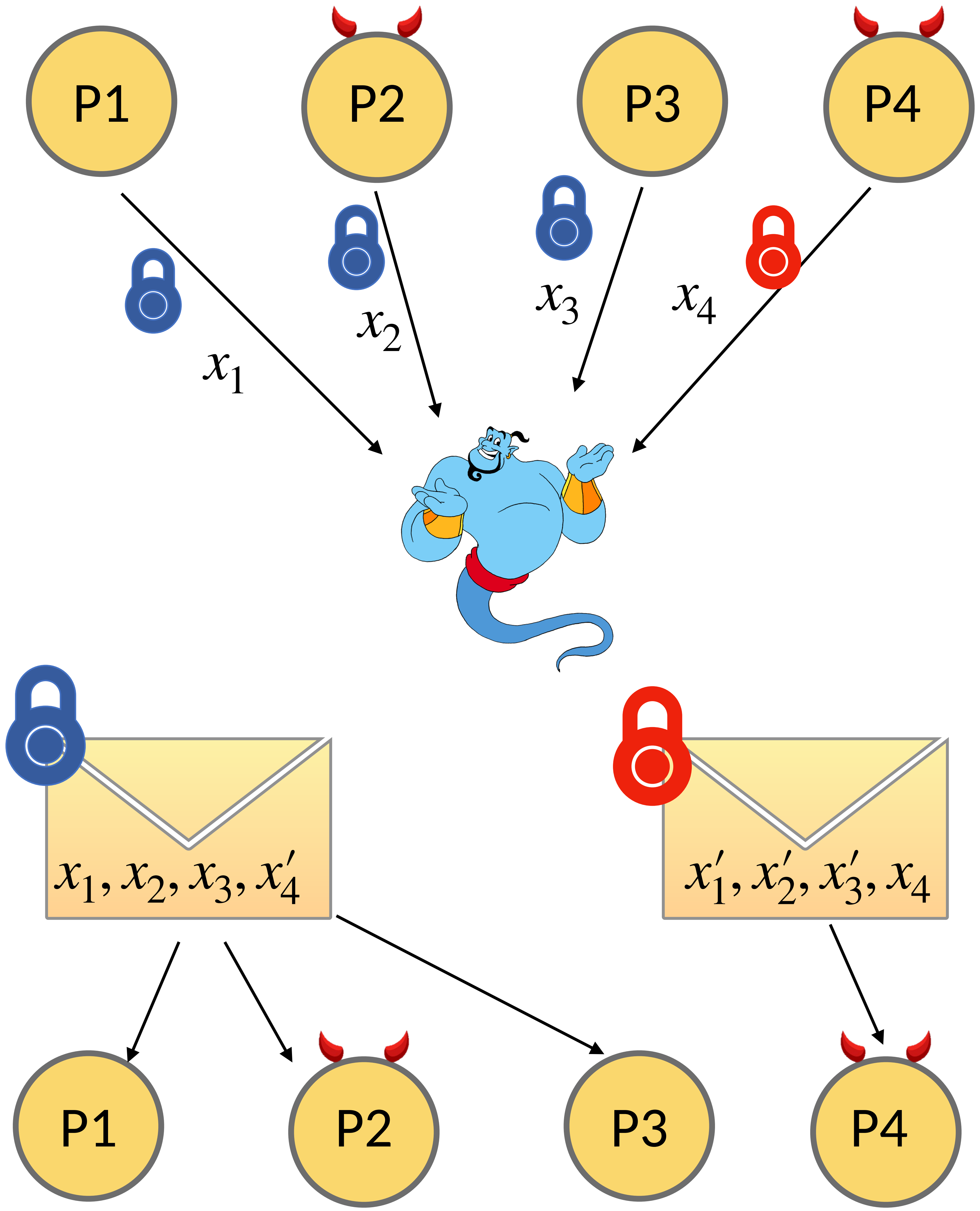
Output: $f(x_1, x_2, x_3, x_4')$

Fully-Secure MPC with 1 call to small TP



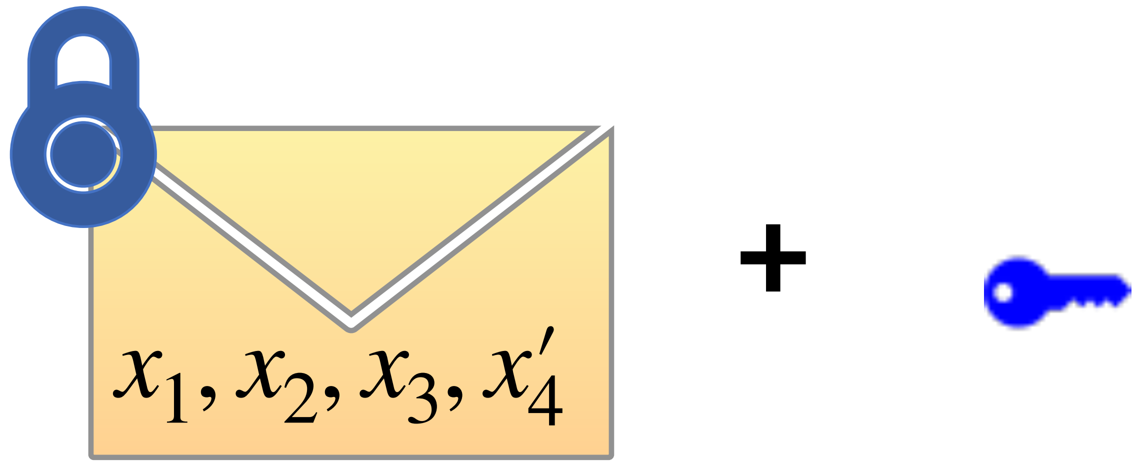
MPC with identifiable abort

mpk , sk_f



Size: poly(n, λ, d) (sub-exp LWE)

Size: poly(n, λ) (iO)



Output: $f(x_1, x_2, x_3, x_4')$

Semi-Honest TP ?

Semi-Honest TP ?

Dishonest Majority of active corruptions

AND

Semi-honest TP

Semi-Honest TP ?

Colluding Model

Dishonest Majority of active corruptions
AND
Semi-honest TP

Semi-Honest TP ?

Colluding Model

Dishonest Majority of active corruptions
AND
Semi-honest TP

Fairness impossible!

Semi-Honest TP ?

Colluding Model

Dishonest Majority of active corruptions
AND
Semi-honest TP

Fairness impossible!

Non- Colluding Model

Dishonest Majority of active corruptions
OR
Semi-honest TP

Semi-Honest TP ?

Colluding Model

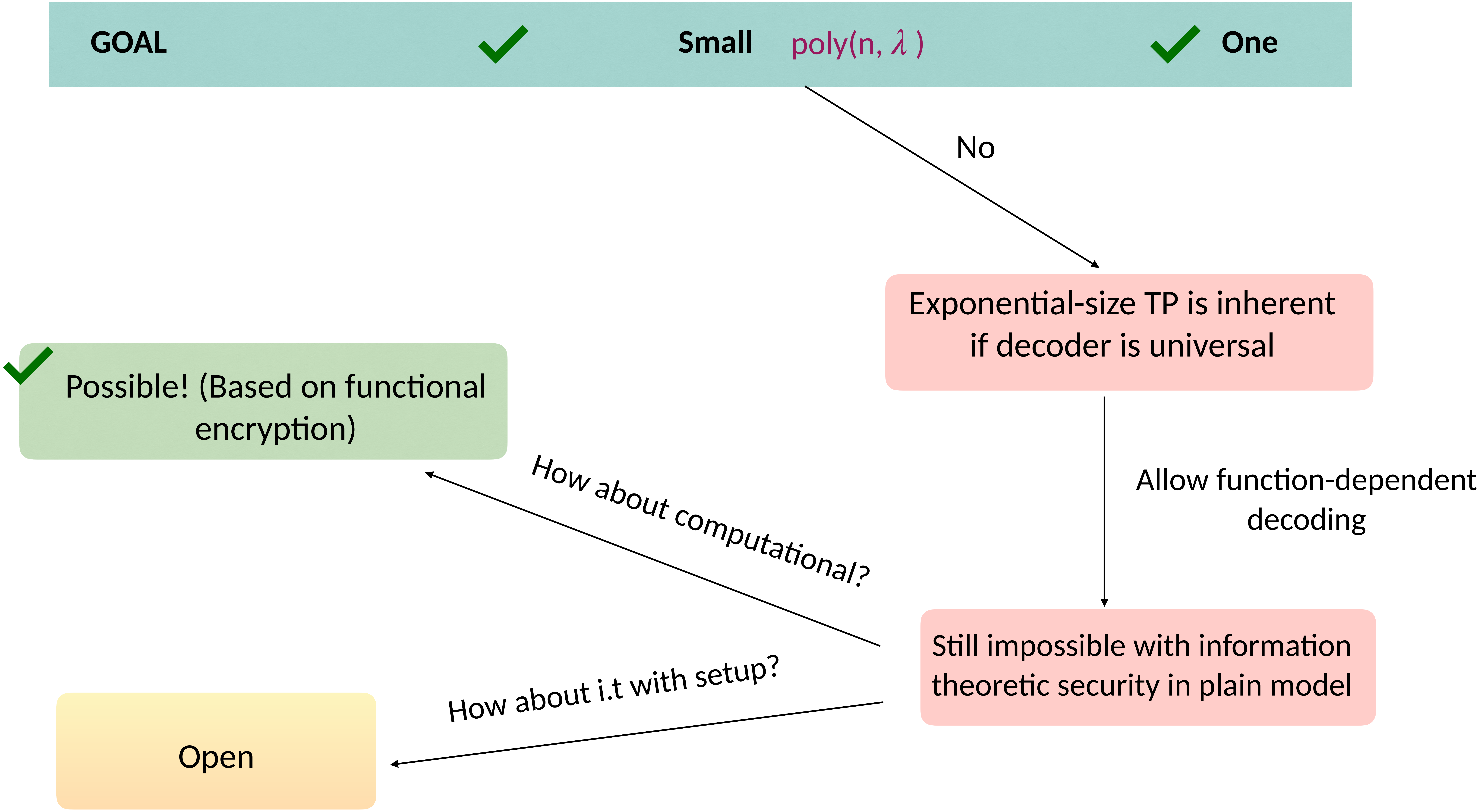
Dishonest Majority of active corruptions
AND
Semi-honest TP

Fairness impossible!

Non- Colluding Model

Dishonest Majority of active corruptions
OR
Semi-honest TP

FE-Based construction works! (With tweaks)



GOAL ✓ **Small** poly(n, λ) ✓ **One**

No

Exponential-size TP is inherent if decoder is universal

Allow function-dependent decoding

Still impossible with information theoretic security in plain model

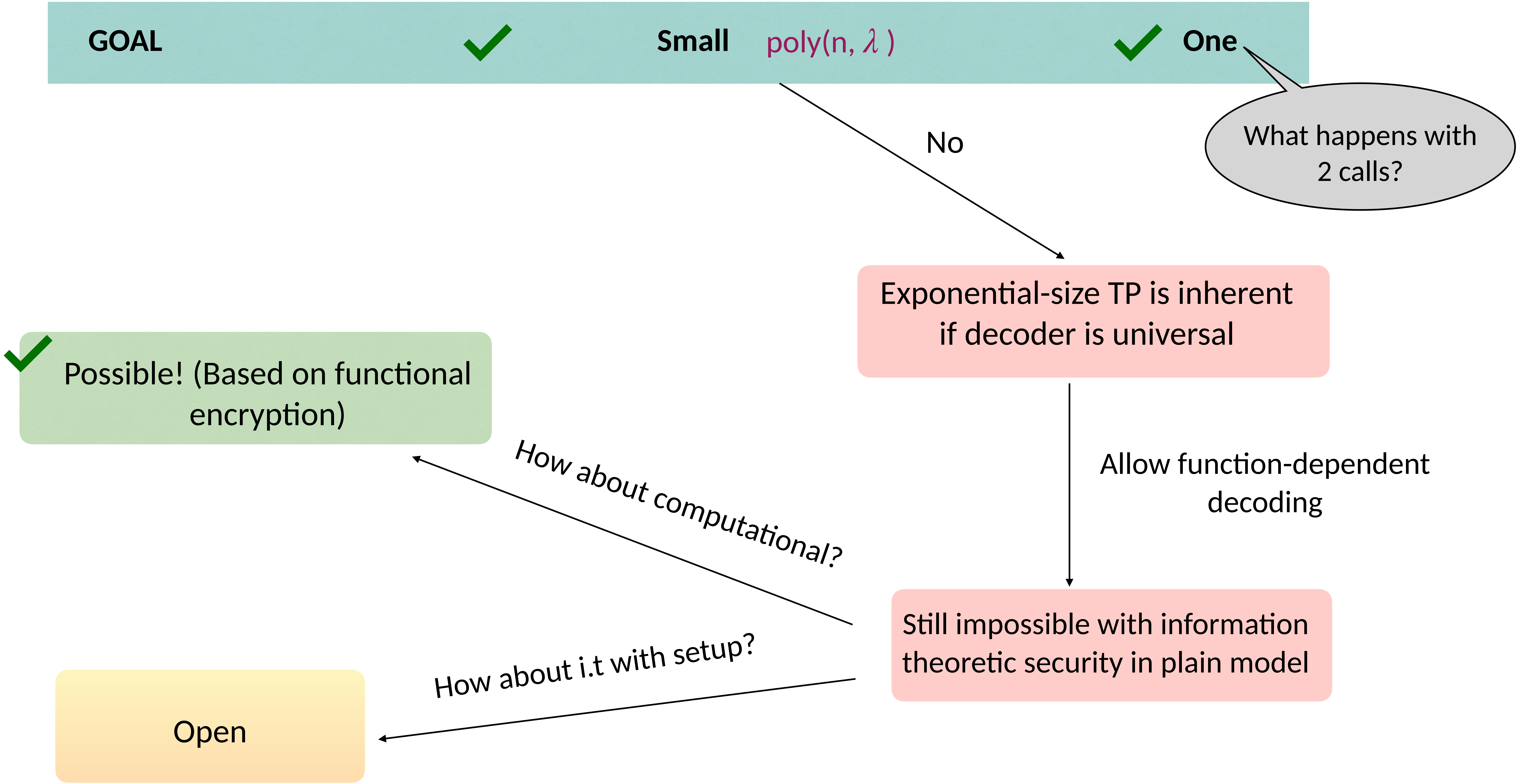
How about computational?

✓ Possible! (Based on functional encryption)

How about i.t with setup?

Open

Thank you :)



GOAL ✓ Small $\text{poly}(n, \lambda)$ ✓ One

What happens with 2 calls?

No

Exponential-size TP is inherent if decoder is universal

Allow function-dependent decoding

Still impossible with information theoretic security in plain model

How about computational?

✓ Possible! (Based on functional encryption)

How about i.t with setup?

Open

Thank you :)