Secure Multi-Party Quantum Computation with a Dishonest Majority

Yfke Dulek, Alex Grilo, Stacey Jeffery,
Christian Majenz, Christian Schaffner

arXiv: 1909.13770 , @EuroCrypt 2020

seminar talk @Simons, Tuesday, March 17, 2020

Secure Multi-Party Quantum Computation with a Dishonest Majority

Yfke Dulek, Alex Grilo, Stacey Jeffery, Christian Majenz, Christian Schaffner

CWI

QuSoft
Research Center for Quantum Software

ArXiv: 1909.13770

Secure Multi-Party Quantum Computation with a Dishonest Majority

Thanks!

Yfke Dulek, Alex Grilo, Stacey Jeffery,
Christian Majenz, Christian Schaffner

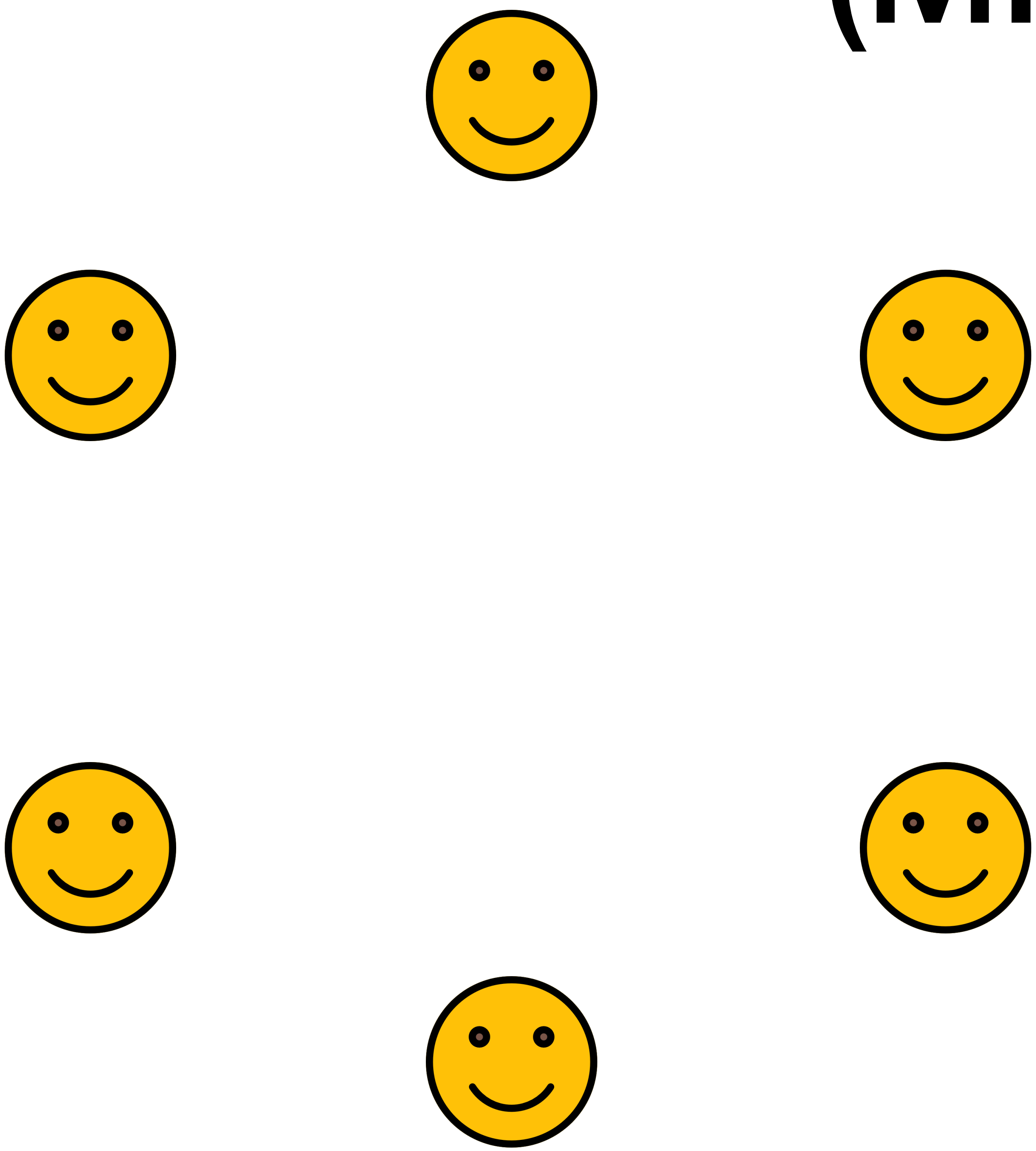
CWI

QuSoft
Research Center for Quantum Software

ArXiv: 1909.13770

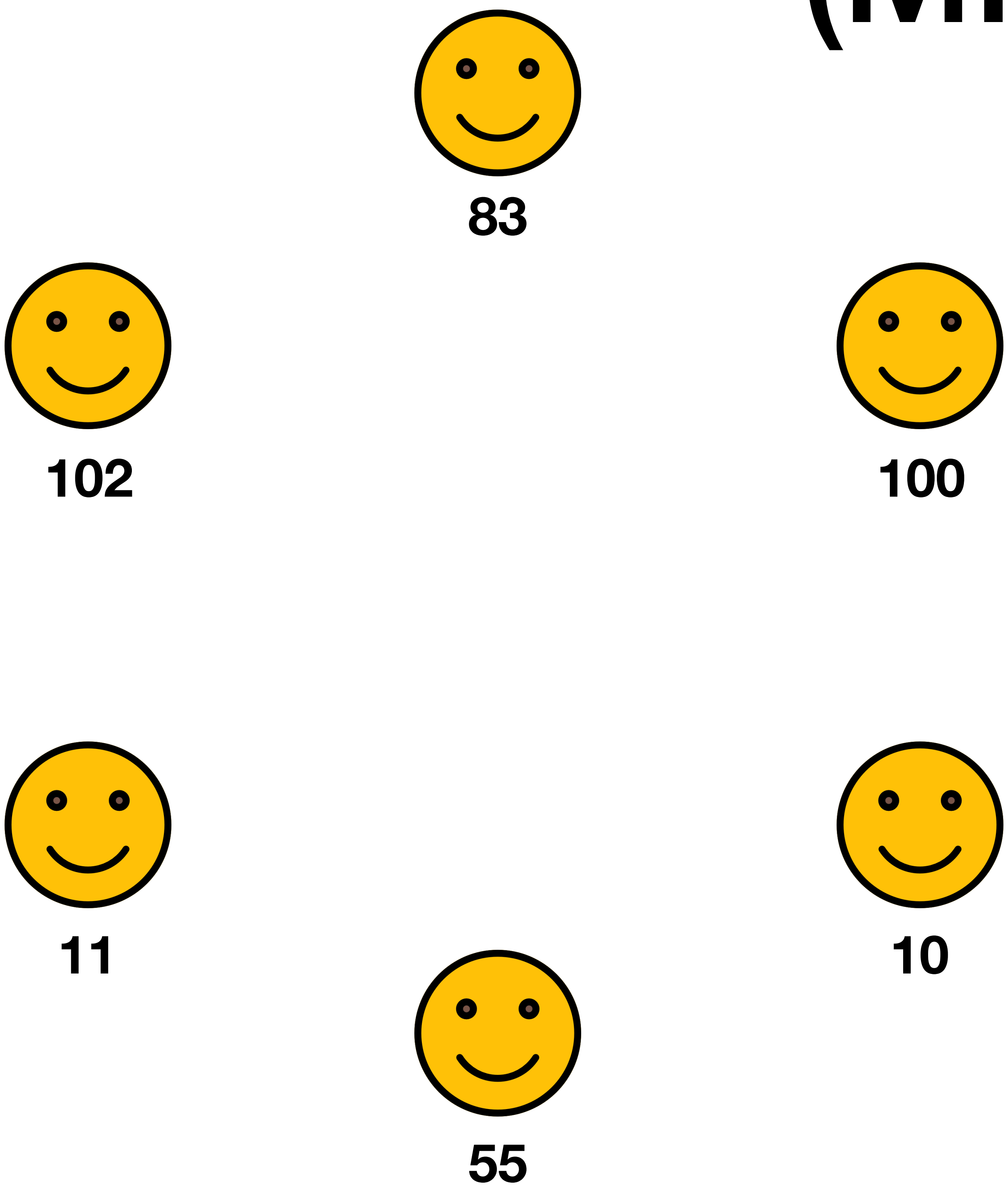
Multi-party computation (MPC)

Multi-party computation (MPC)



Icons by Pixel perfect.

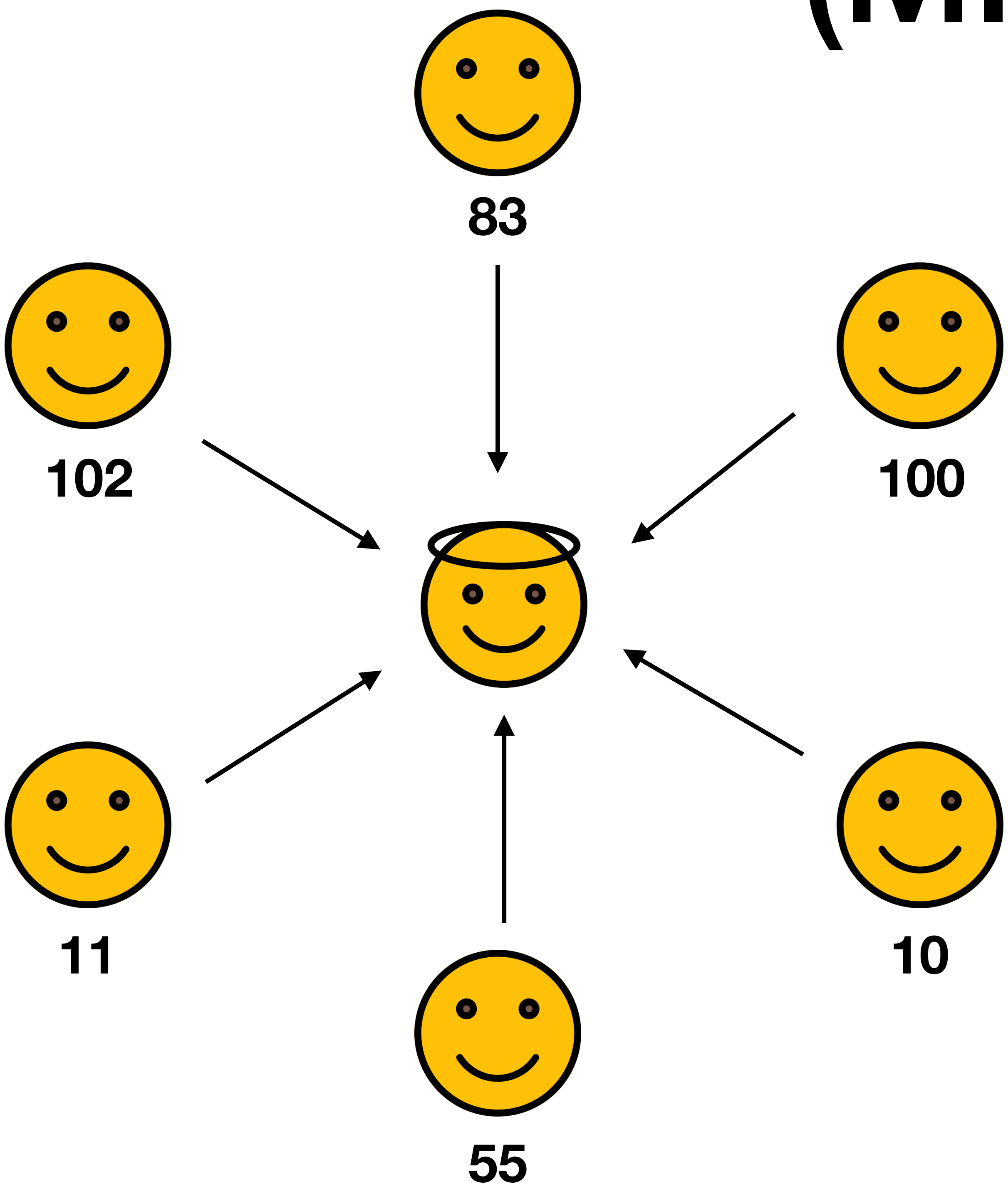
Multi-party computation (MPC)



Input (player i): x_i

Icons by Pixel perfect.

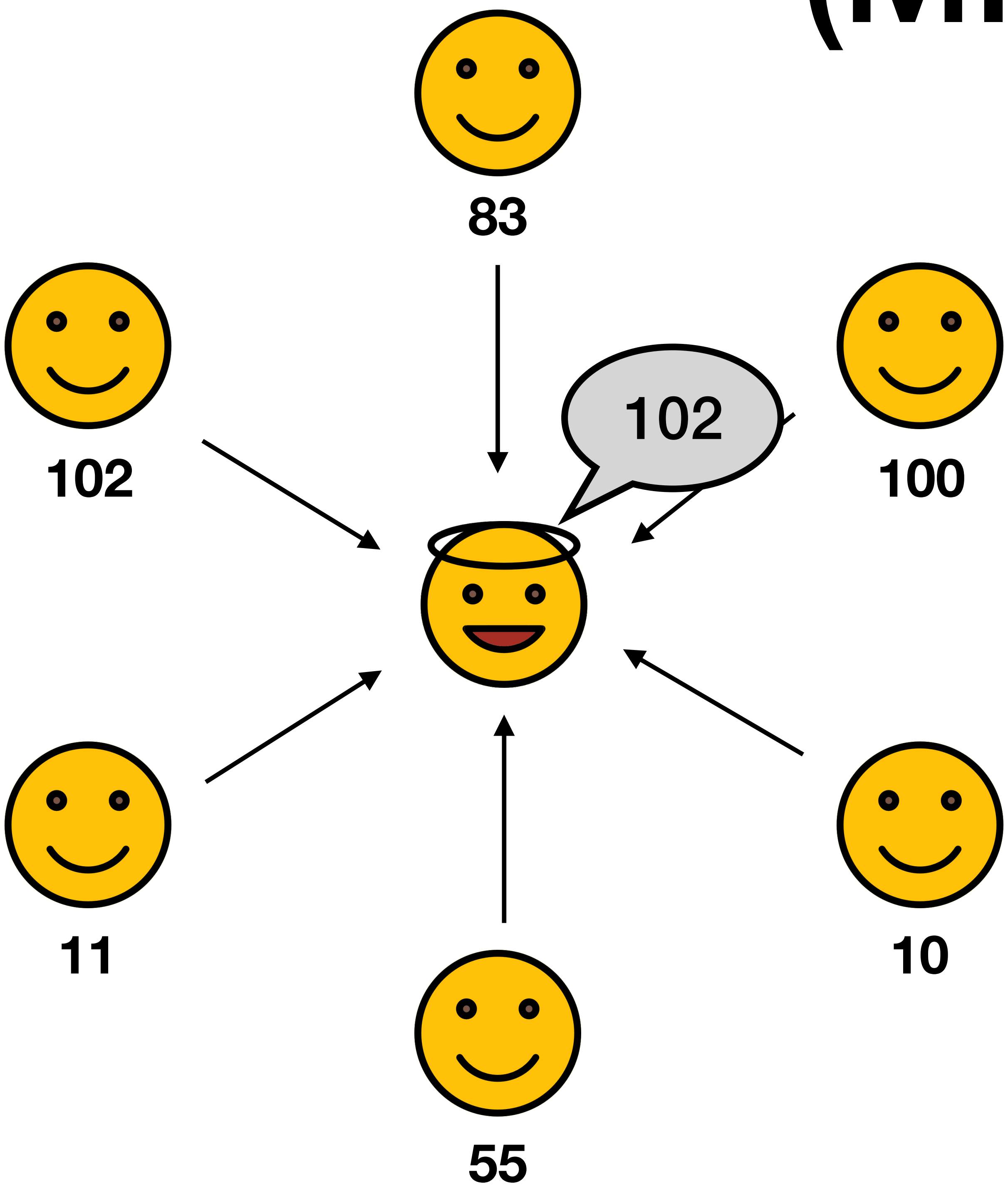
Multi-party computation (MPC)



Input (player i): x_i

Icons by Pixel perfect.

Multi-party computation (MPC)

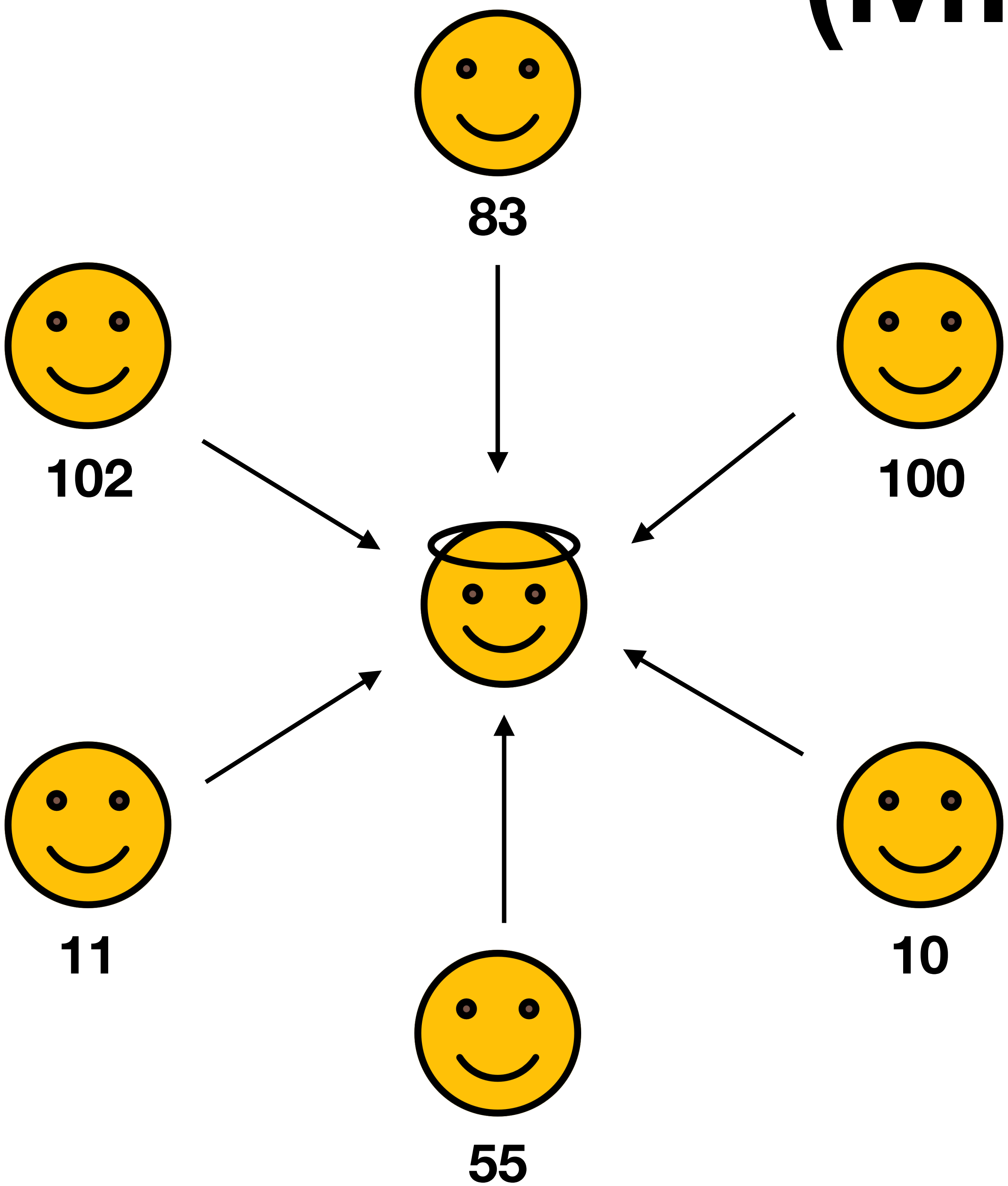


Input (player i): x_i

Output: $f(x_1, \ldots, x_k)$

Icons by Pixel perfect.

Multi-party computation (MPC)

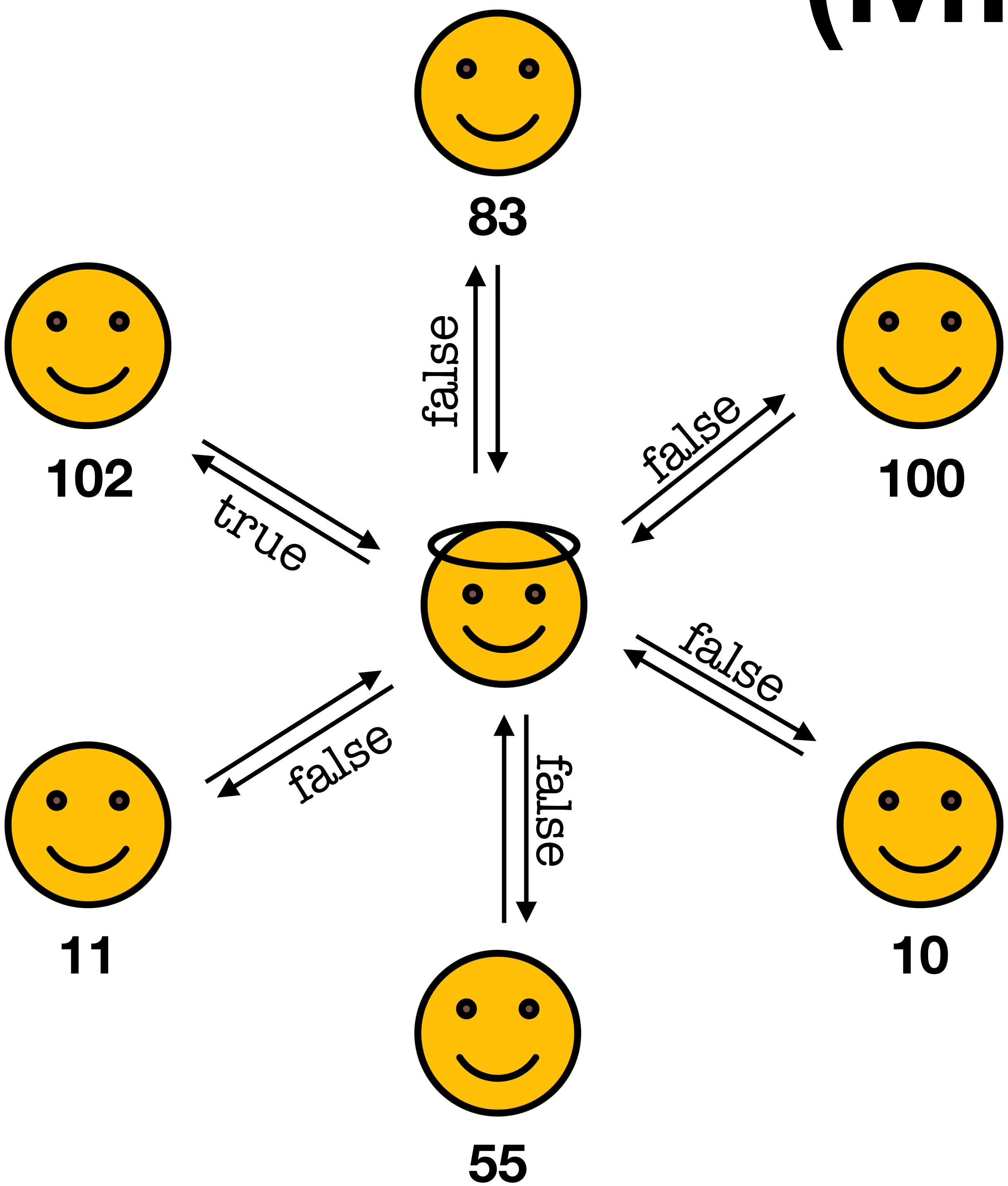


Input (player i): x_i

Output: $f(x_1, \ldots, x_k)$

Icons by Pixel perfect.

Multi-party computation (MPC)



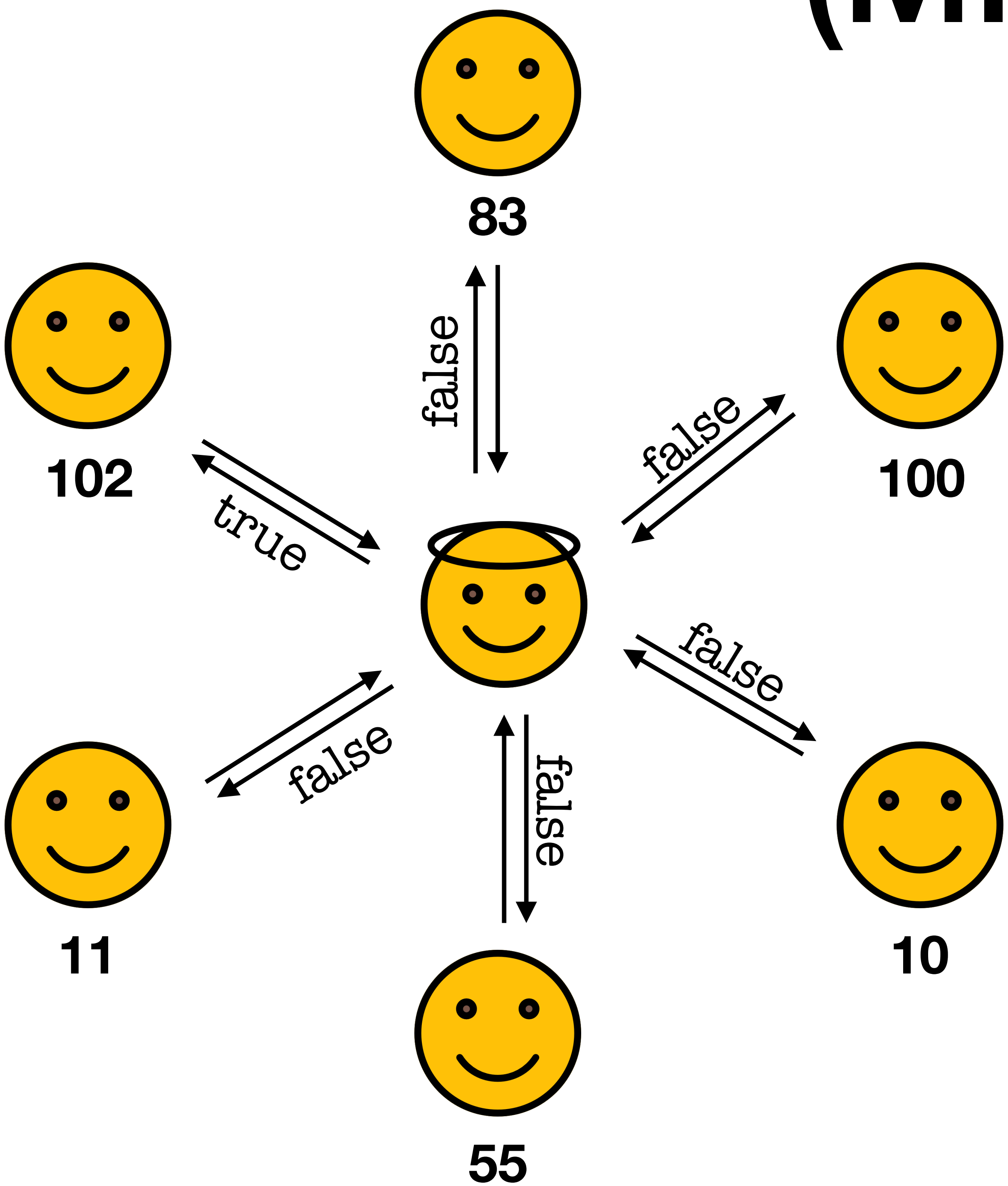
Input (player i): x_i

~~Output: $f(x_1, \ldots, x_k)$~~

Output (player i): $f_i(x_1, \ldots, x_k)$

Icons by Pixel perfect.

Multi-party computation (MPC)



Input (player i): x_i

~~Output: $f(x_1, \ldots, x_k)$~~

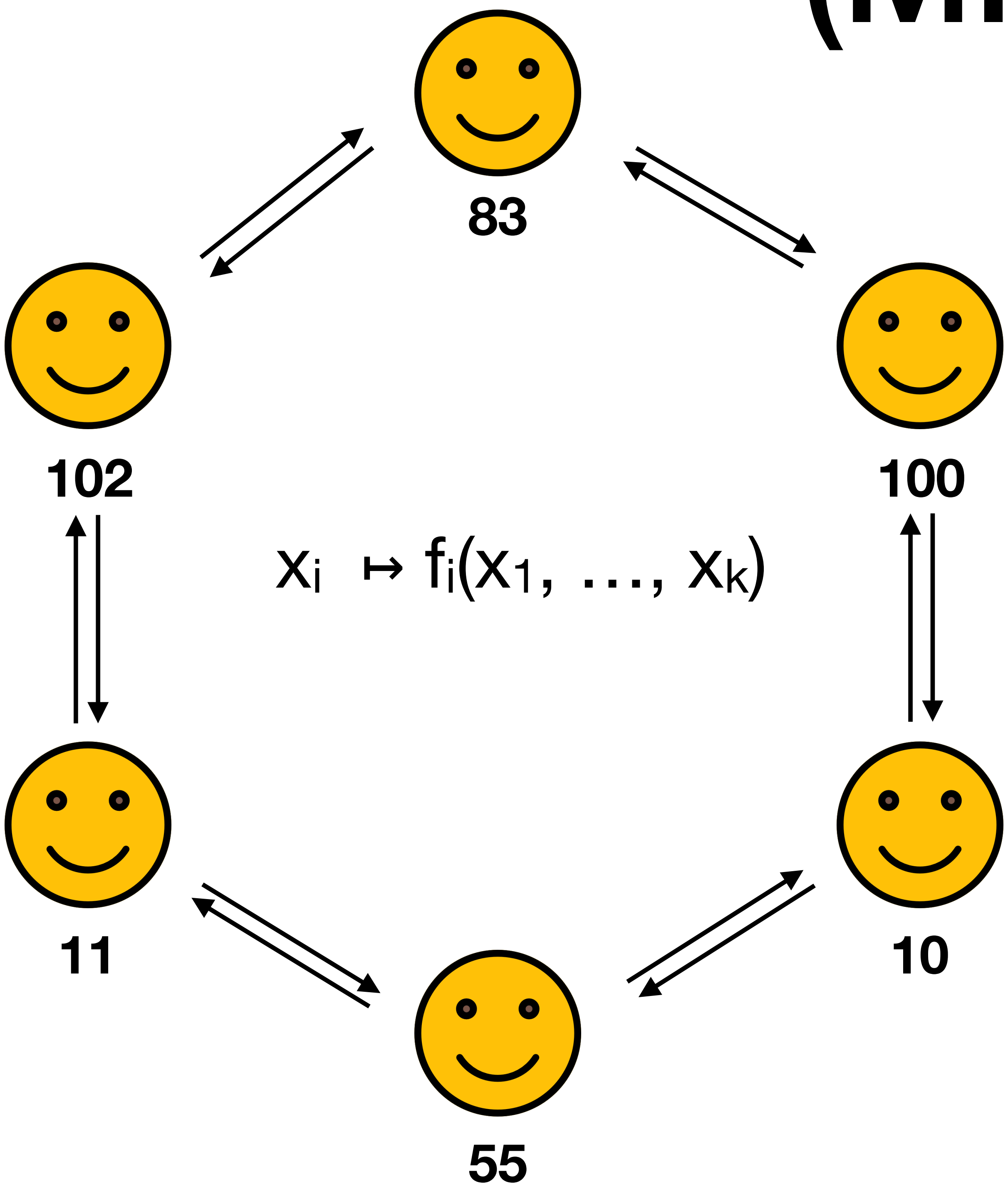
Output (player i): $f_i(x_1, \ldots, x_k)$

This is the **ideal** situation.

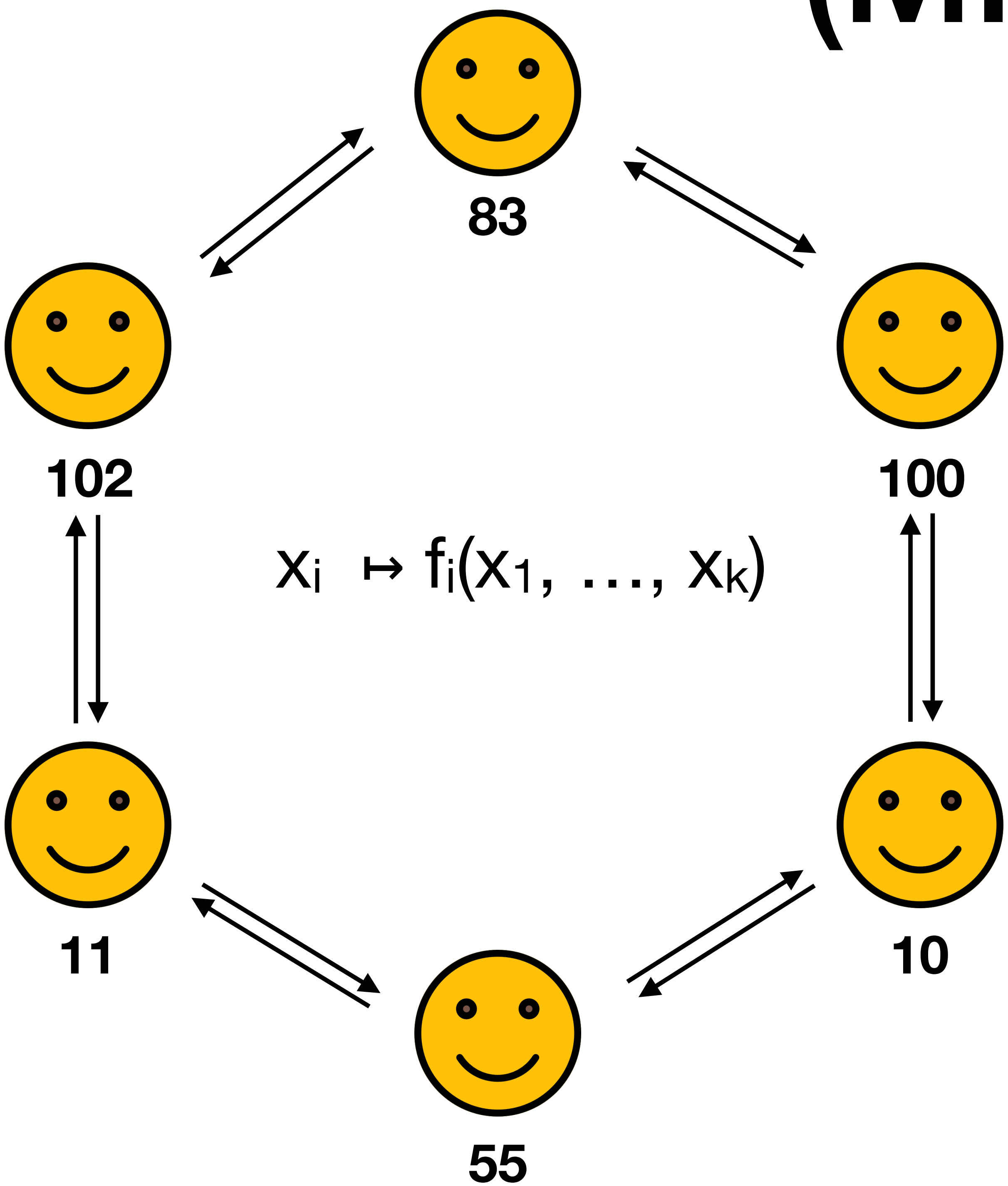
What if there is no  ?

Icons by Pixel perfect.

Multi-party computation (MPC)

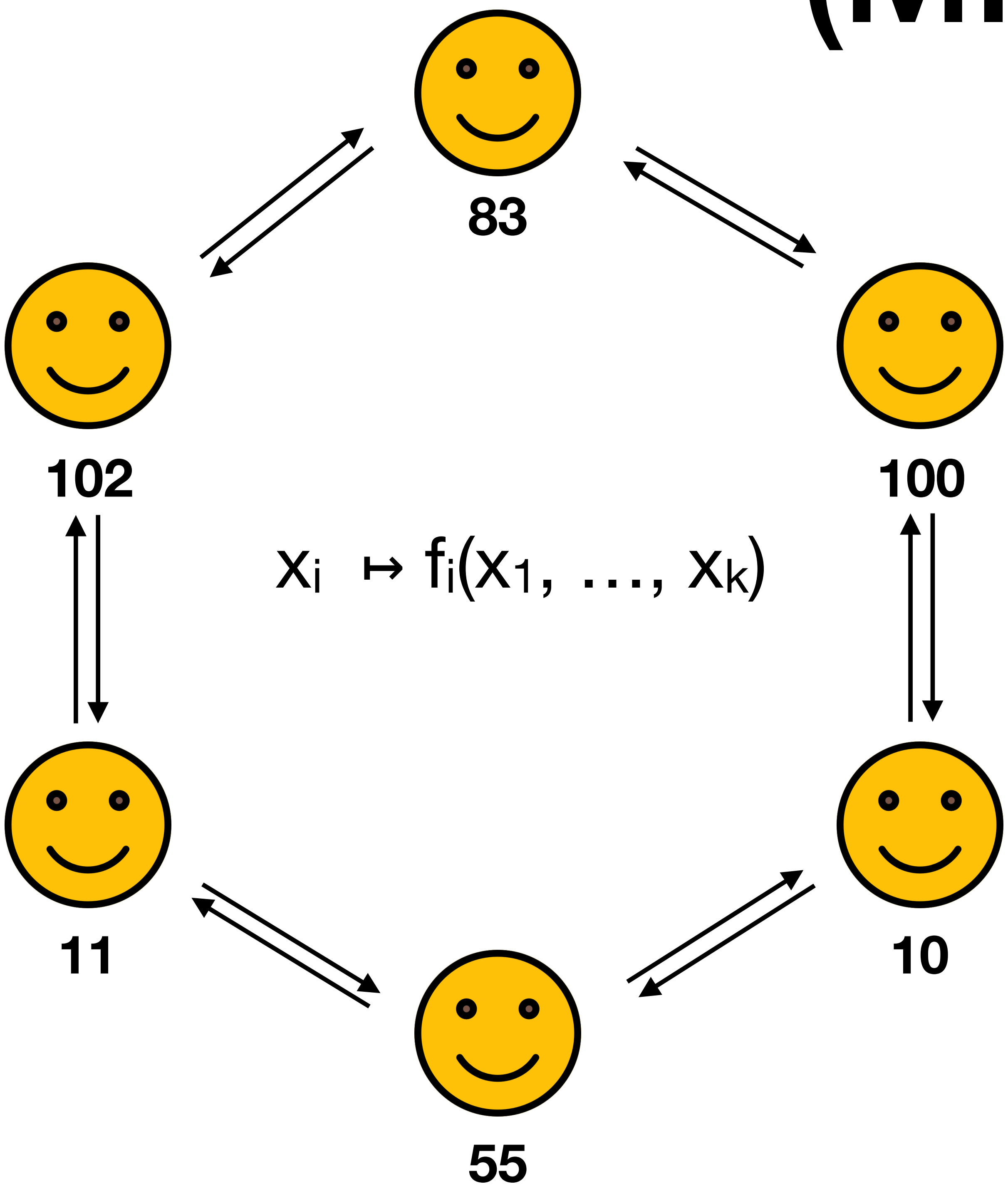


Multi-party computation (MPC)



We want:

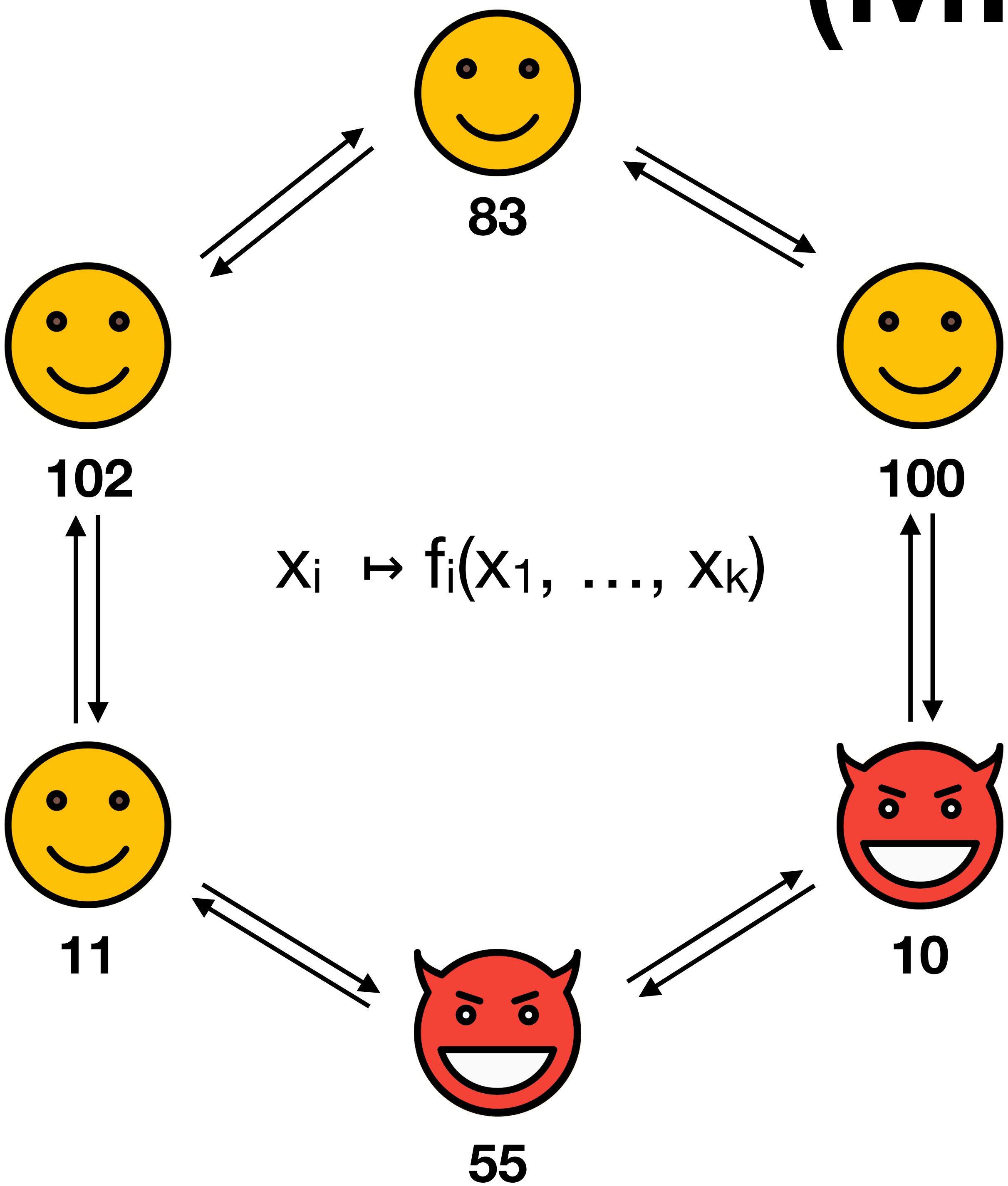
Multi-party computation (MPC)



We want:

- input privacy

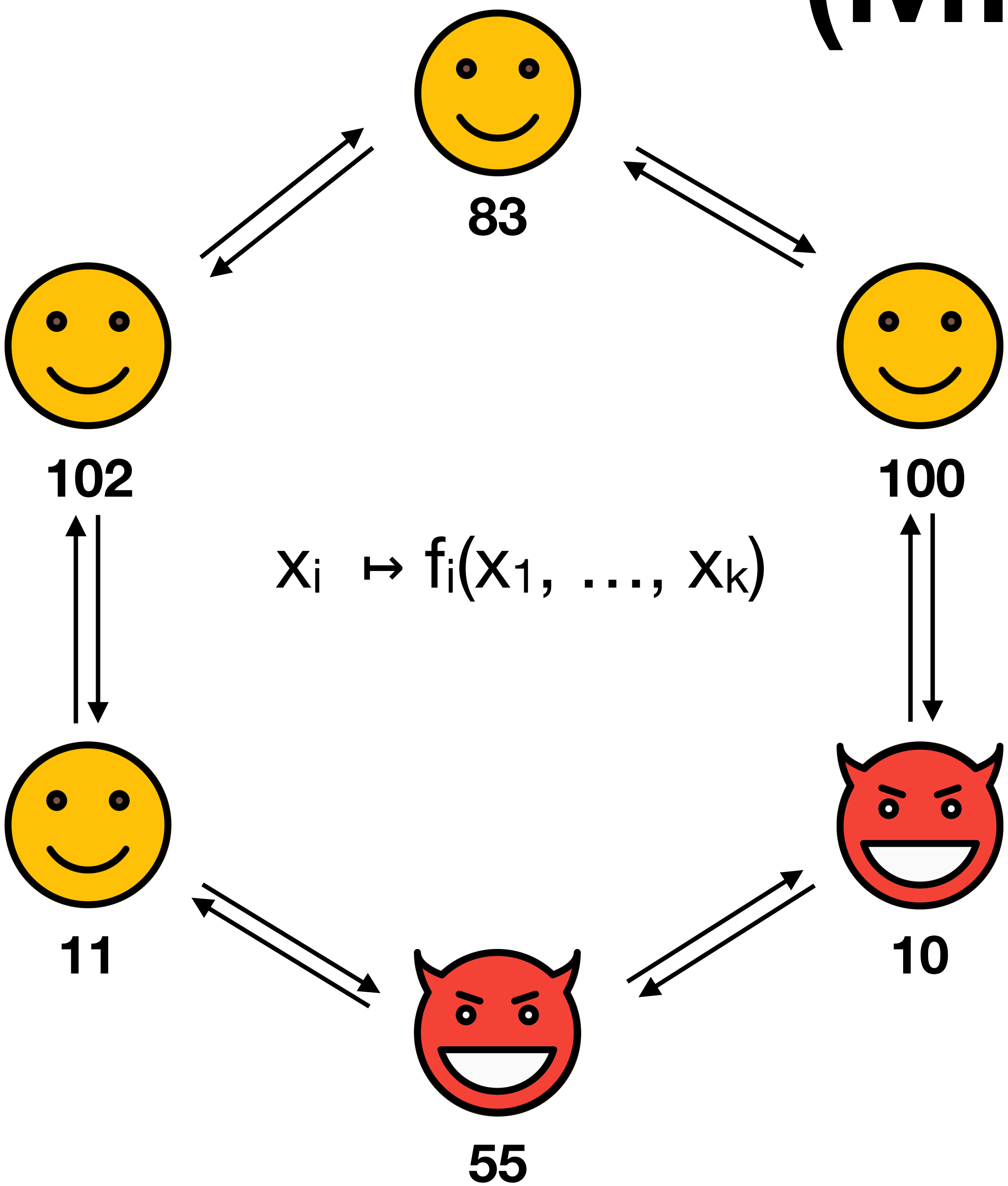
Multi-party computation (MPC)



We want:

- input privacy

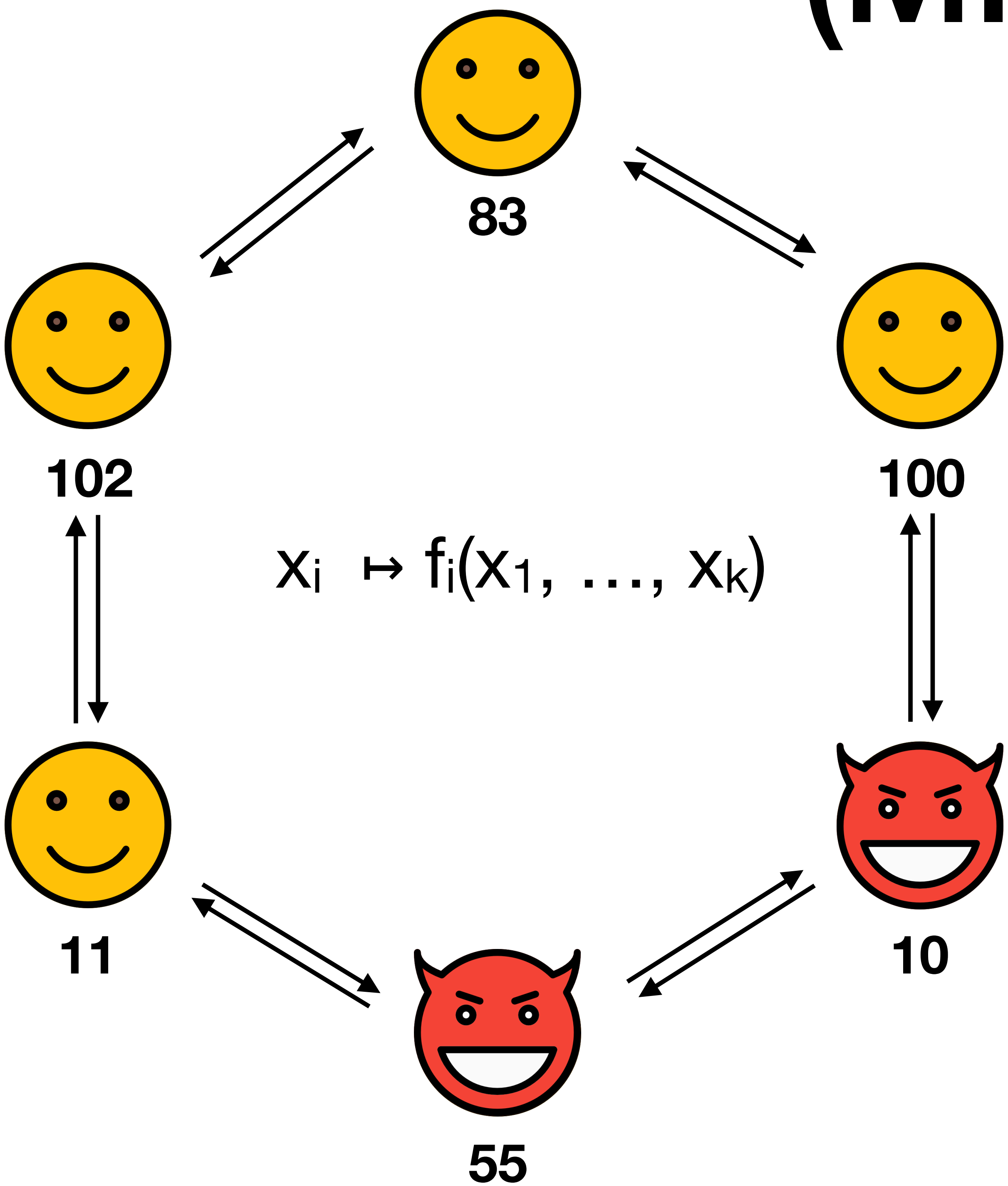
Multi-party computation (MPC)



We want:

- input privacy
- correctness

Multi-party computation (MPC)

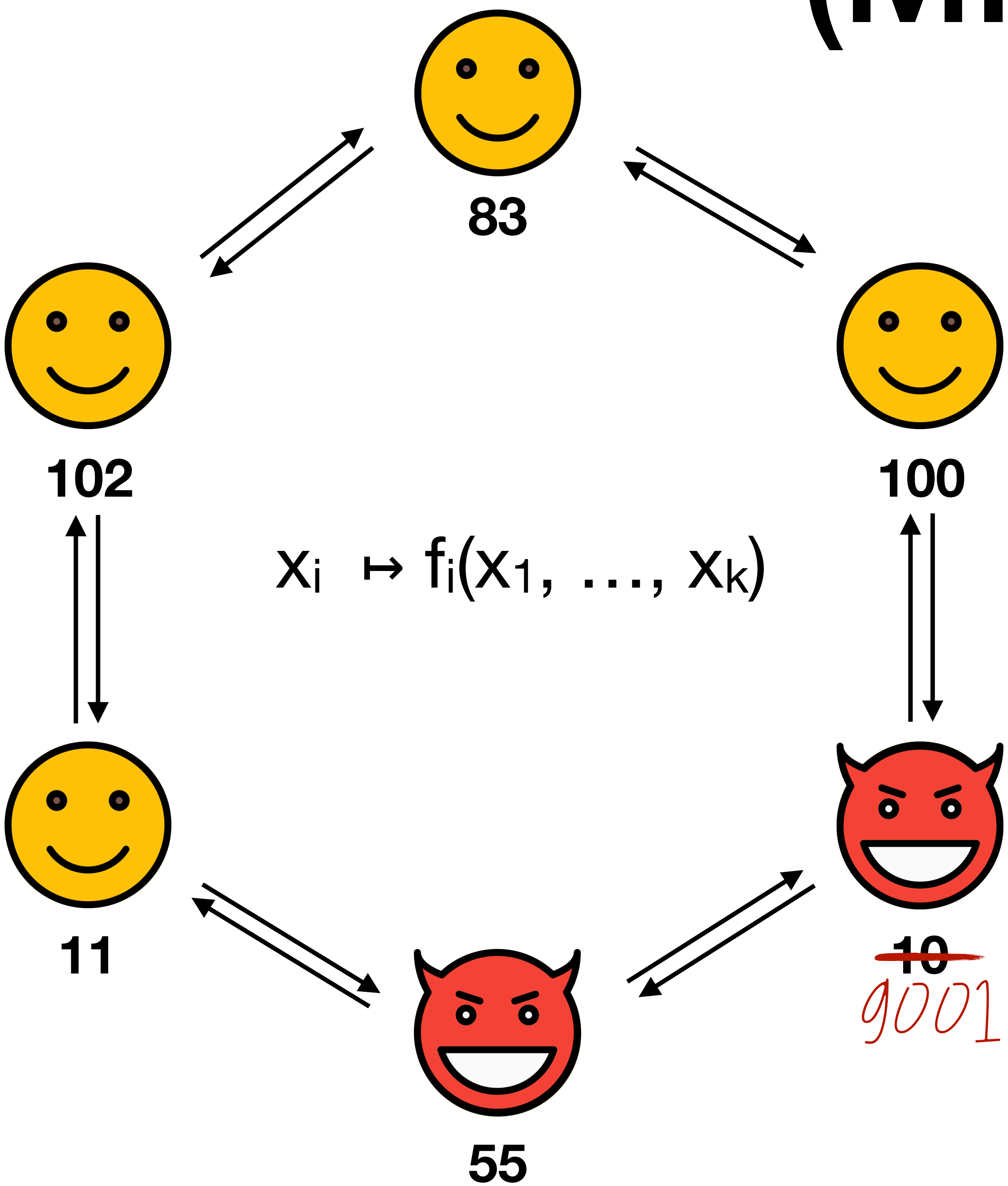


We want:

- input privacy
- correctness

We cannot prevent:

Multi-party computation (MPC)



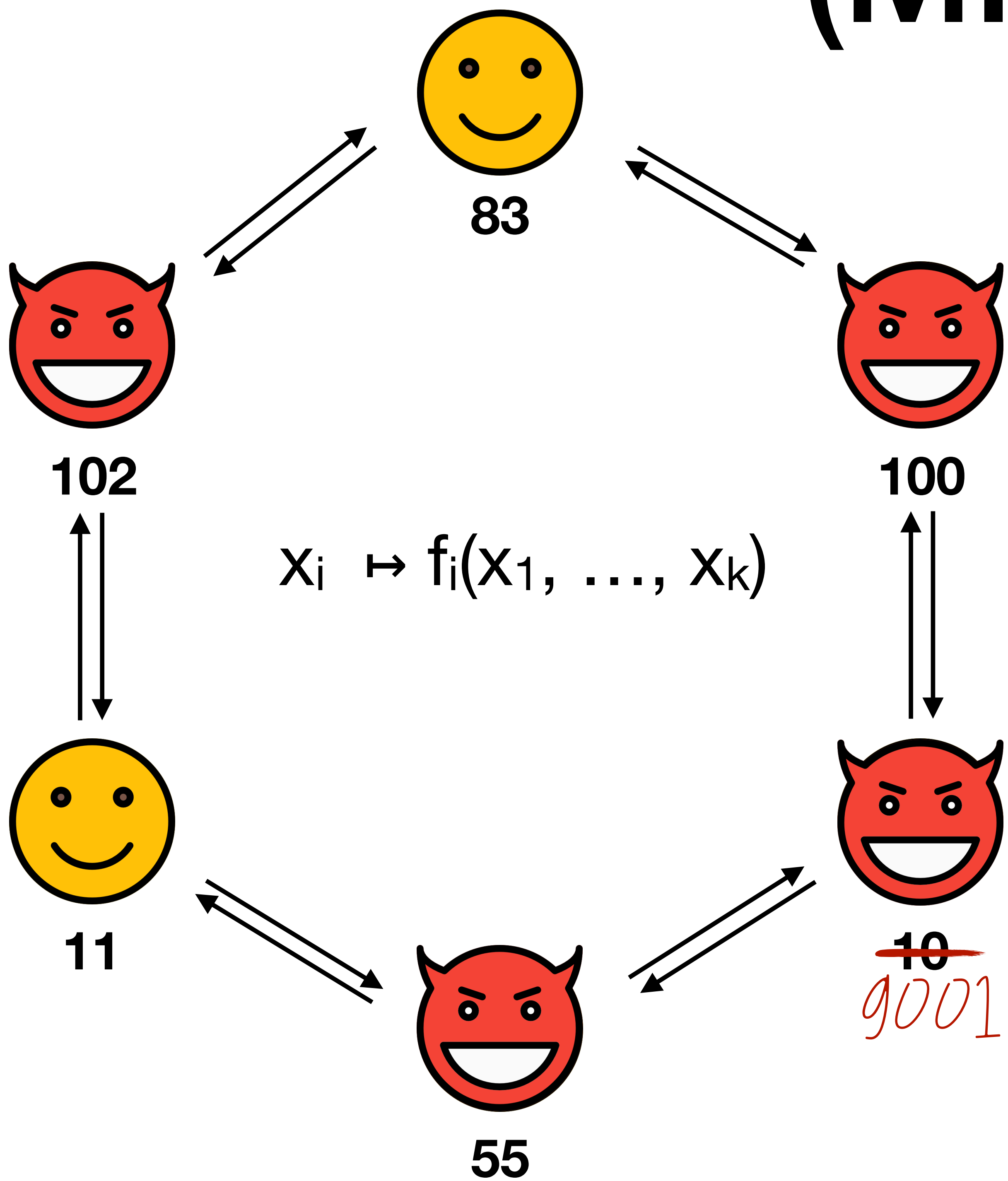
We want:

- input privacy
- correctness

We cannot prevent:

- lying about inputs

Multi-party computation (MPC)



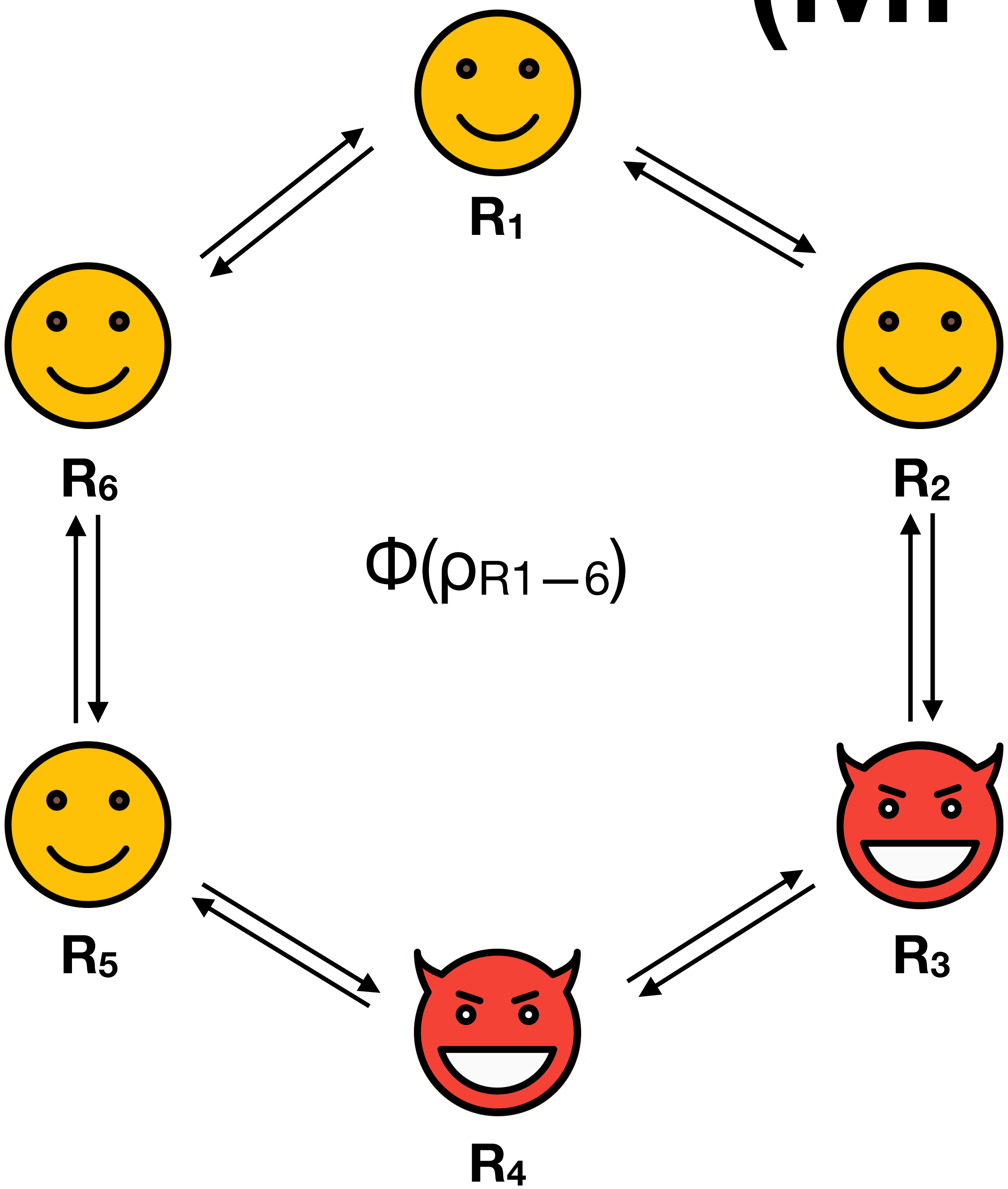
We want:

- input privacy
- correctness

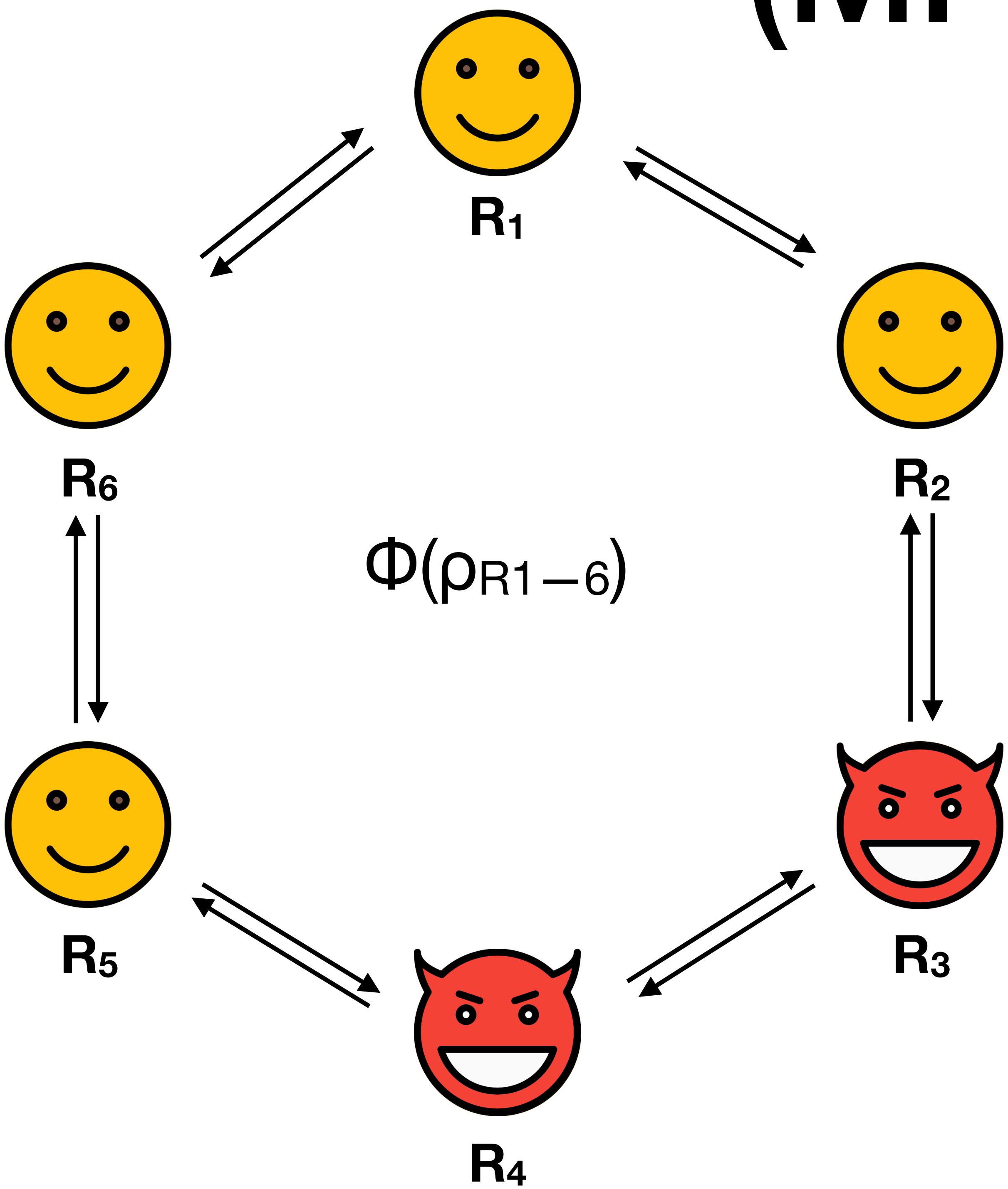
We cannot prevent:

- lying about inputs
- unfairness

Goal: Quantum MPC (MPQC)

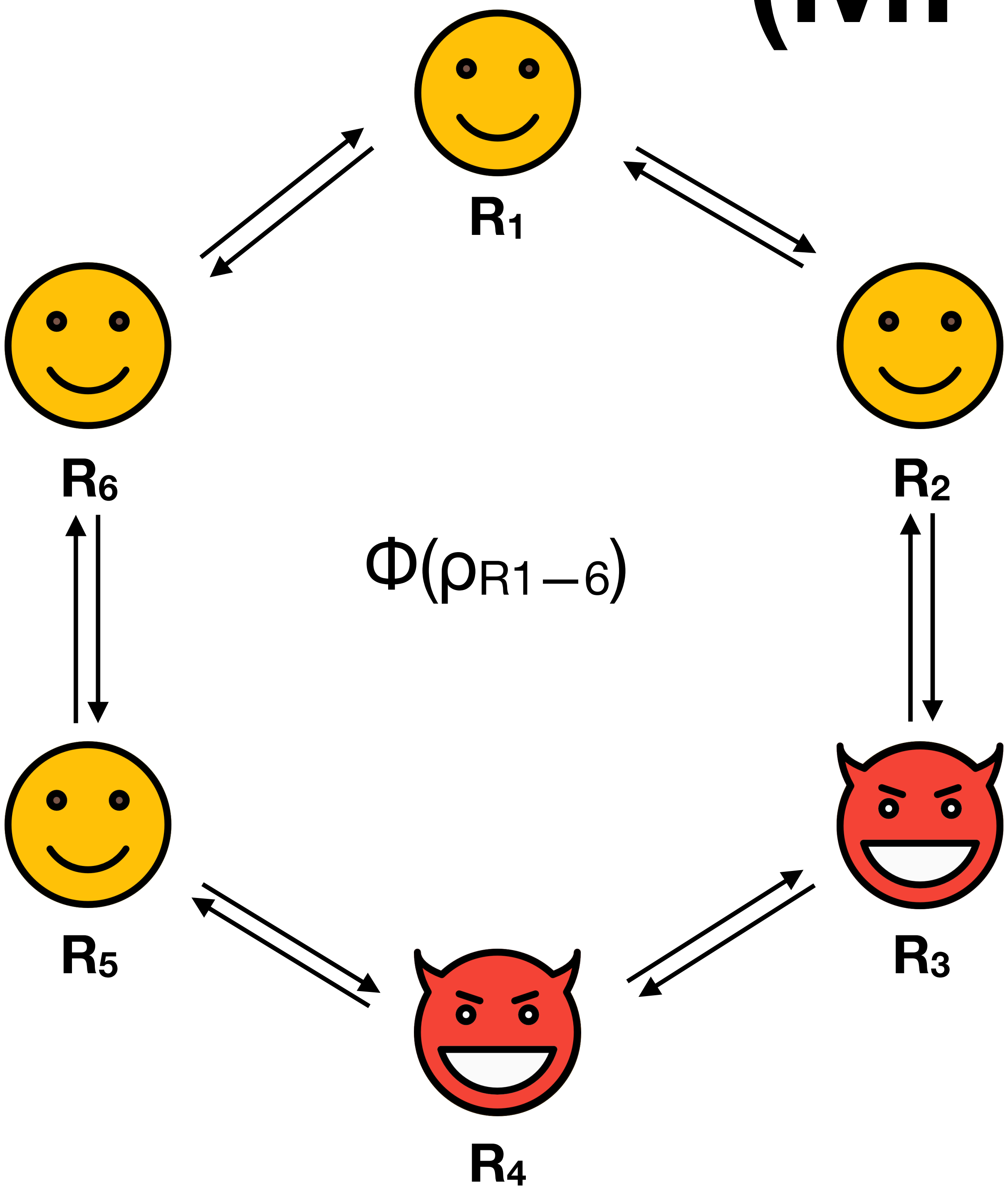


Goal: Quantum MPC (MPQC)



This talk: protocol for MPQC

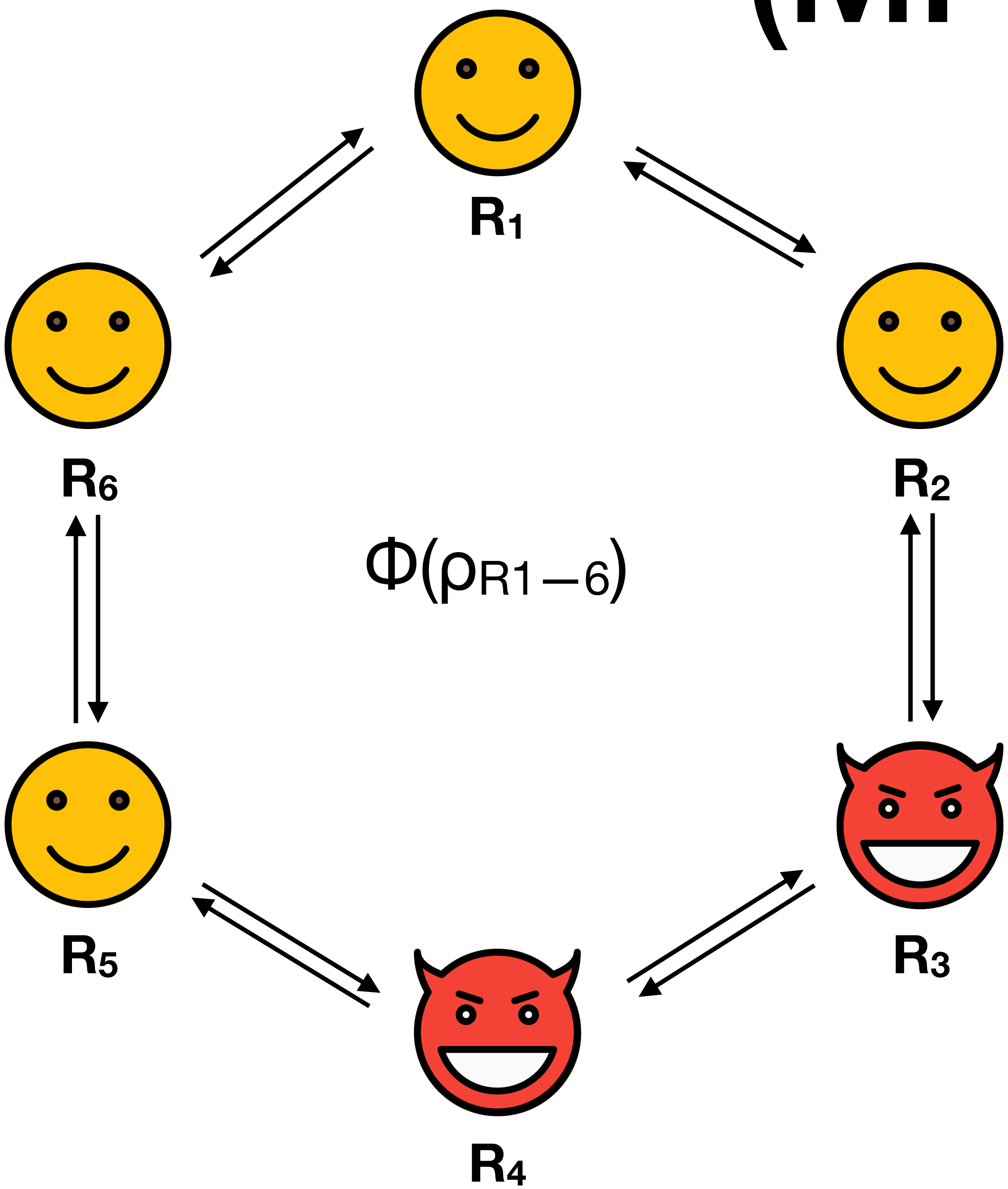
Goal: Quantum MPC (MPQC)



This talk: protocol for MPQC

- Up to k-1 😈

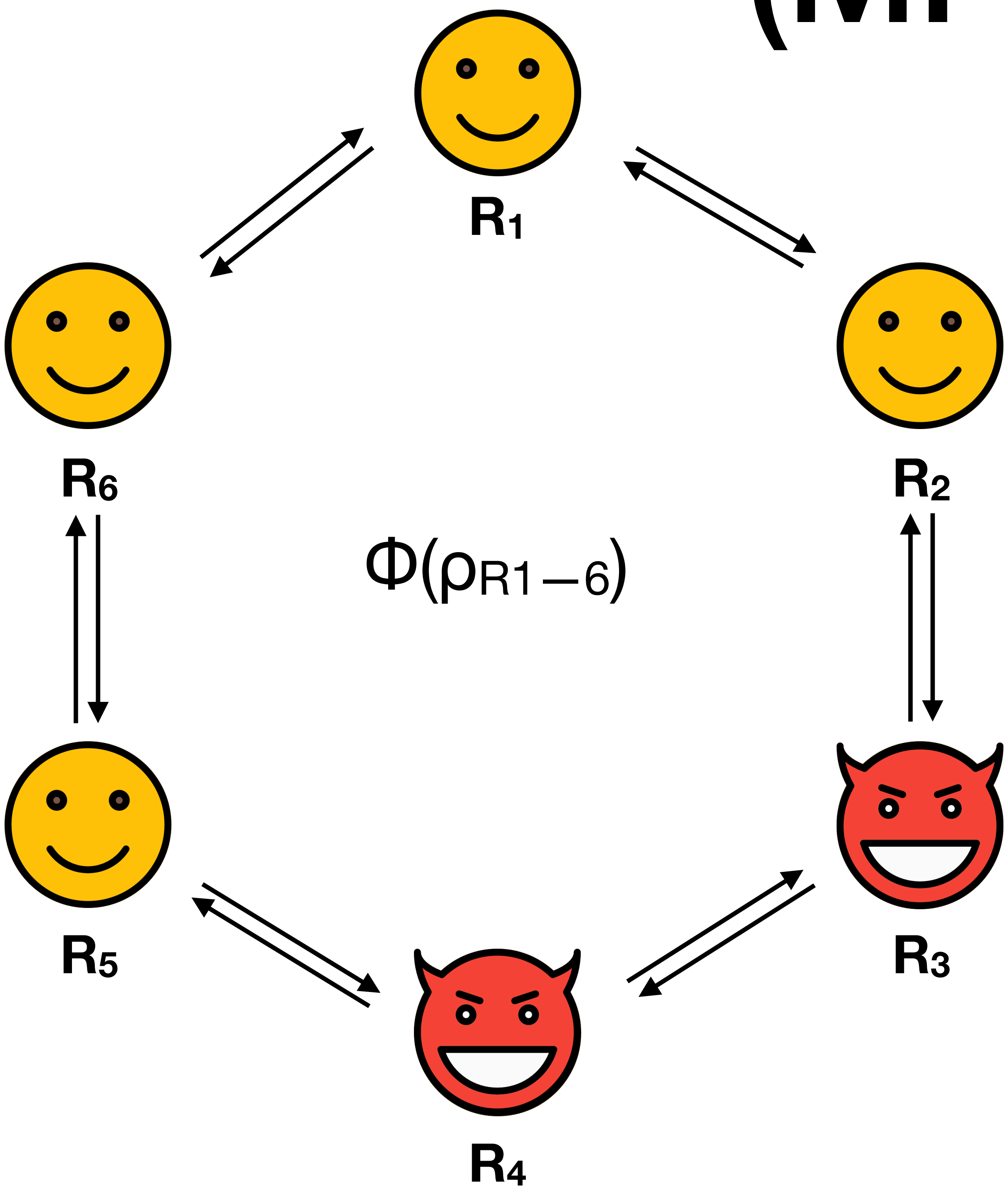
Goal: Quantum MPC (MPQC)



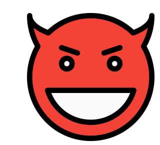
This talk: protocol for MPQC

- Up to k-1 
- Computationally secure

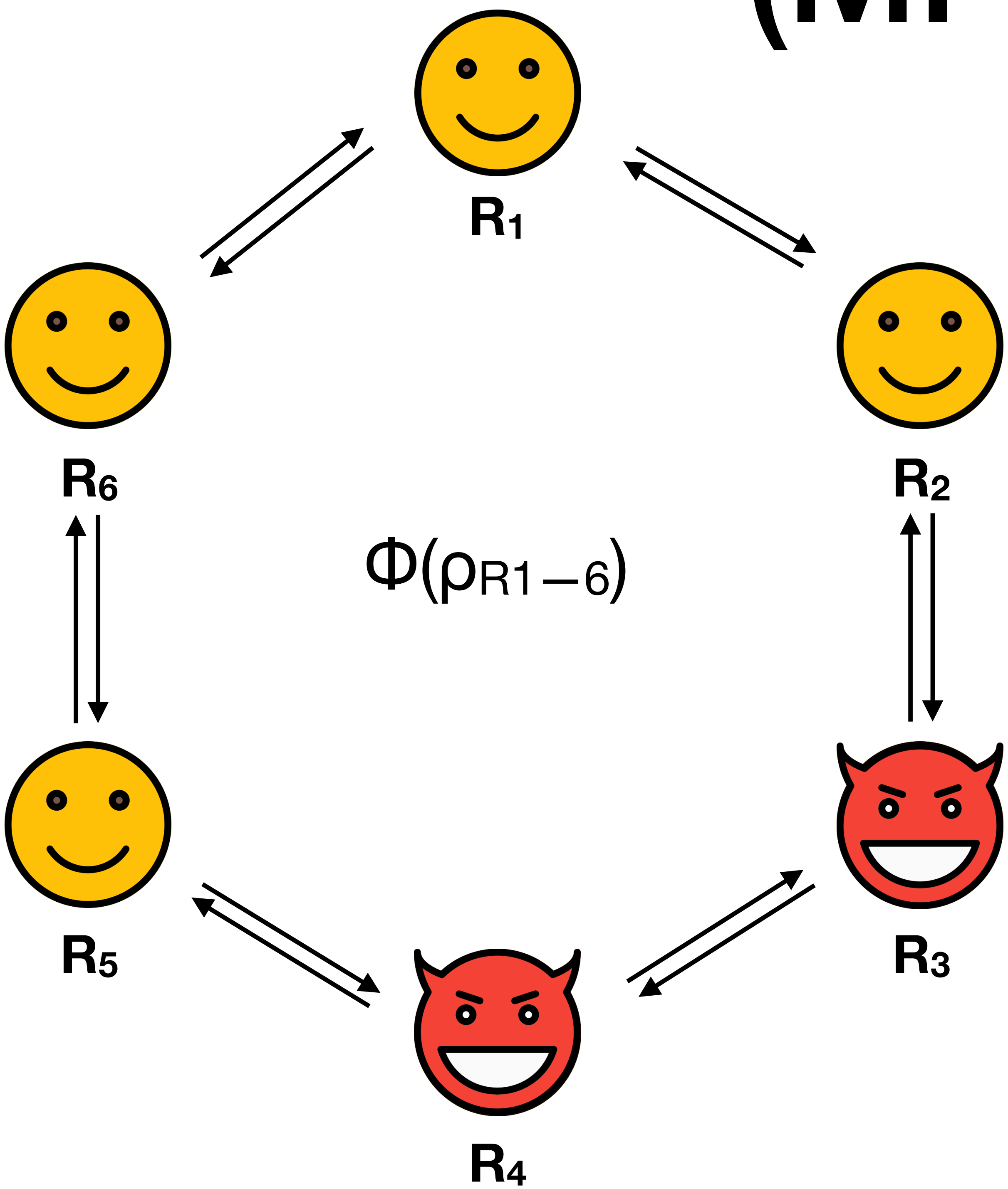
Goal: Quantum MPC (MPQC)



This talk: protocol for MPQC

- Up to k-1 
- Computationally secure
- gate-by-gate, using $O\big(k(d + \log(n))\big)$ quant rounds for d the {CNOT,T}-depth of the q computation

Goal: Quantum MPC (MPQC)

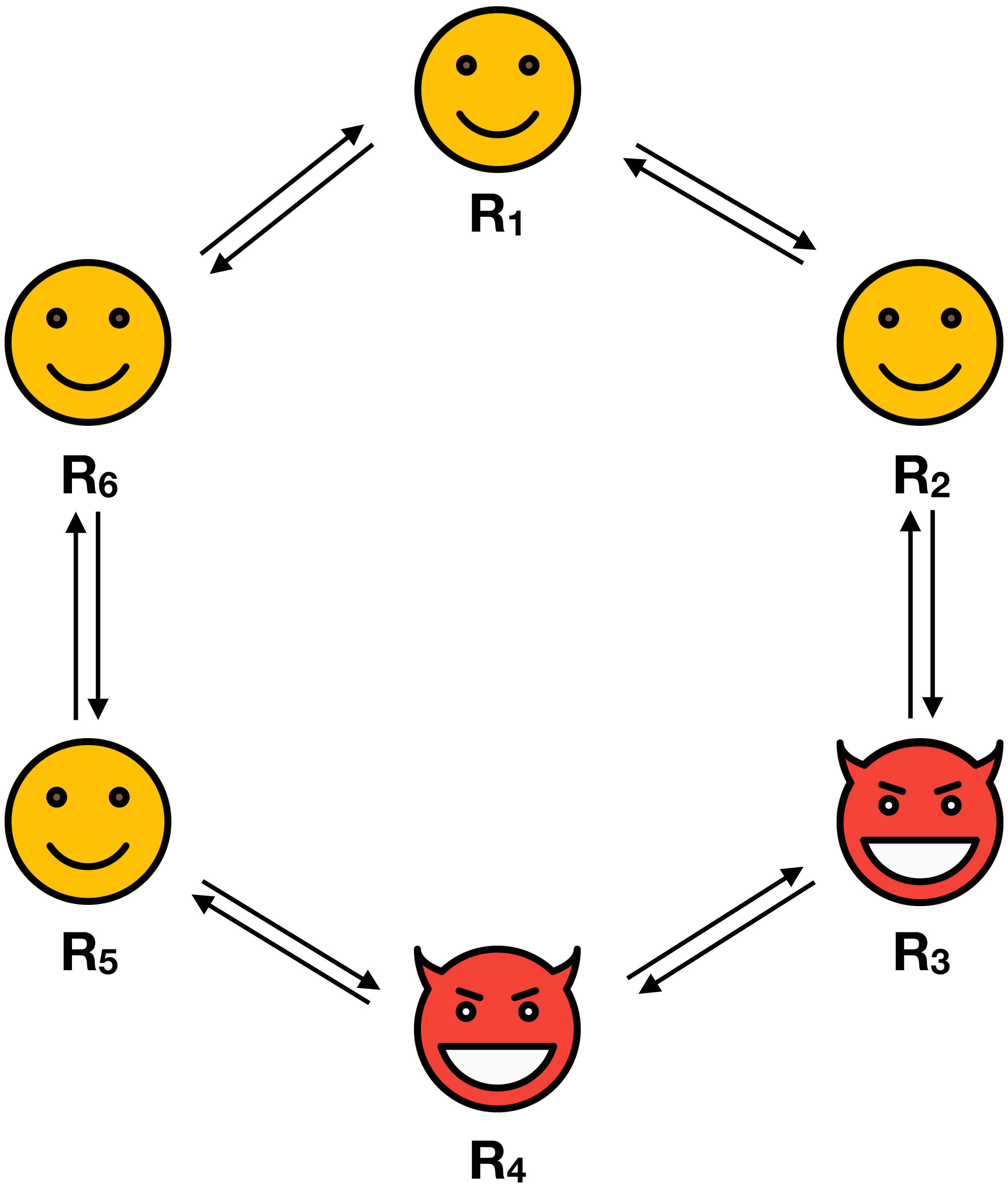


This talk: protocol for MPQC

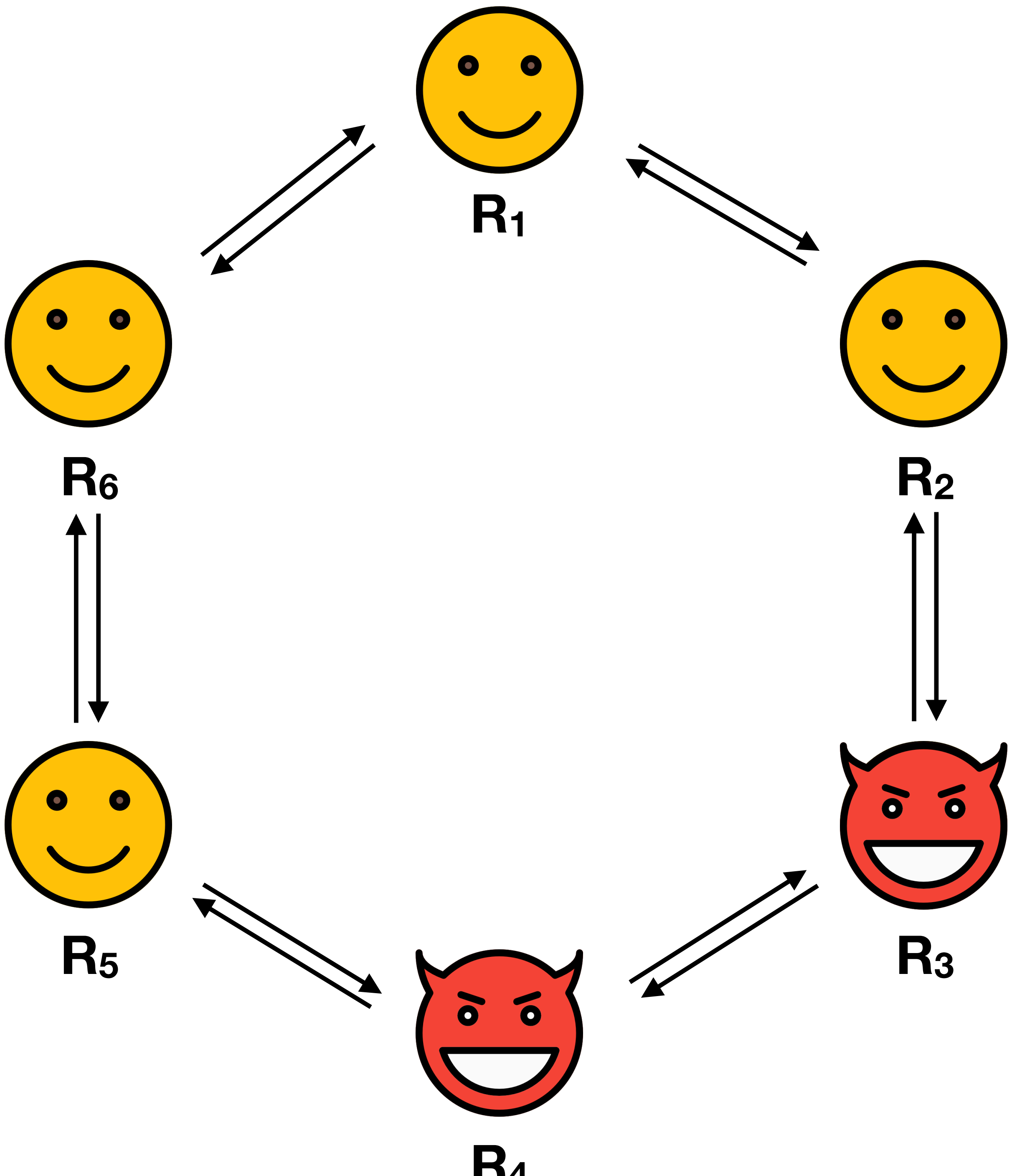
- Up to k-1 
- Computationally secure
- gate-by-gate, using $O\big(k(d+\log(n))\big)$ quant rounds for d the {CNOT,T}-depth of the q computation
- subroutine: classical MPC 

Previous Approaches

MPQC: two approaches



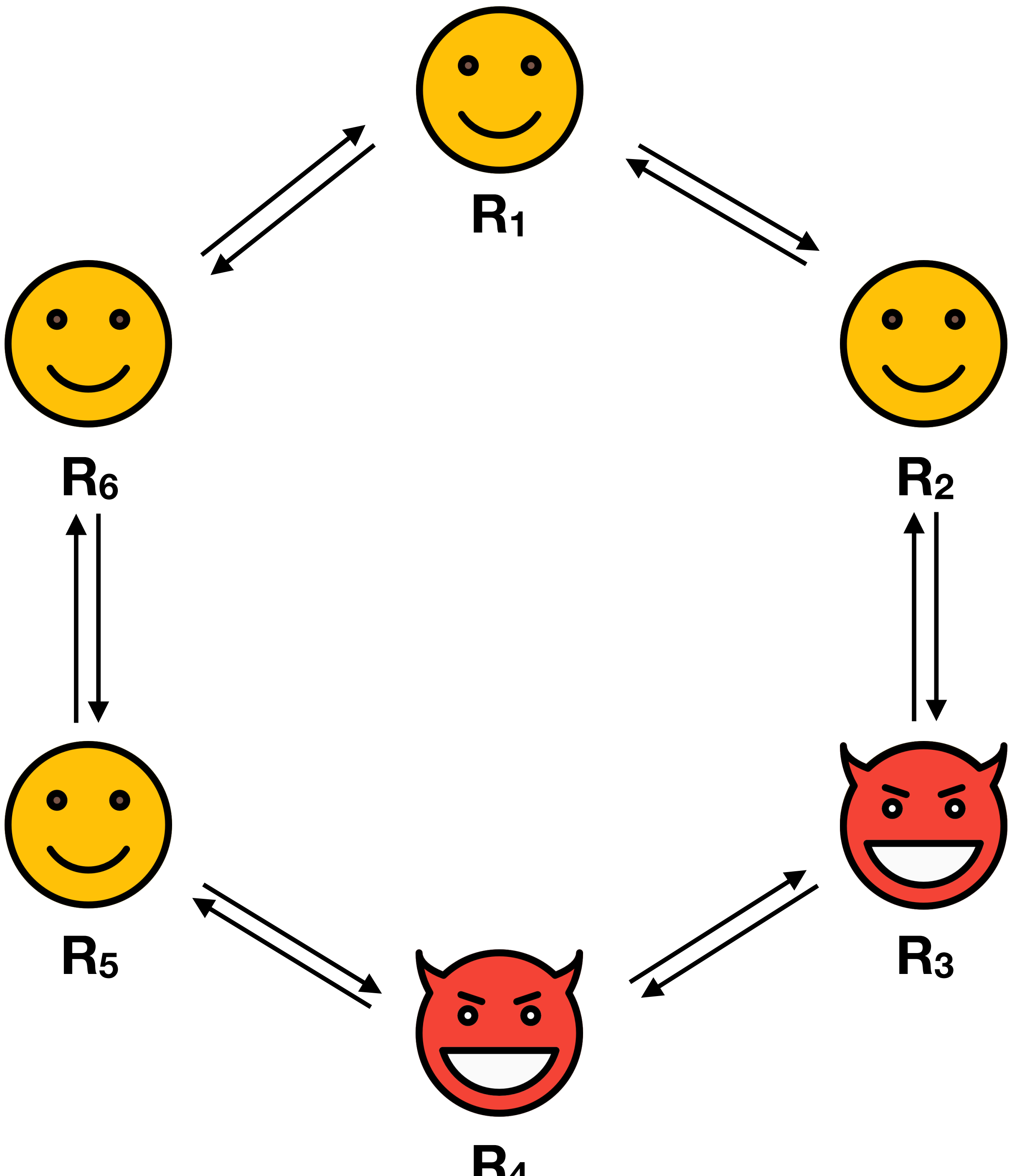
MPQC: two approaches



1. Secret sharing [CGS02, BCGHS06]

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)

MPQC: two approaches

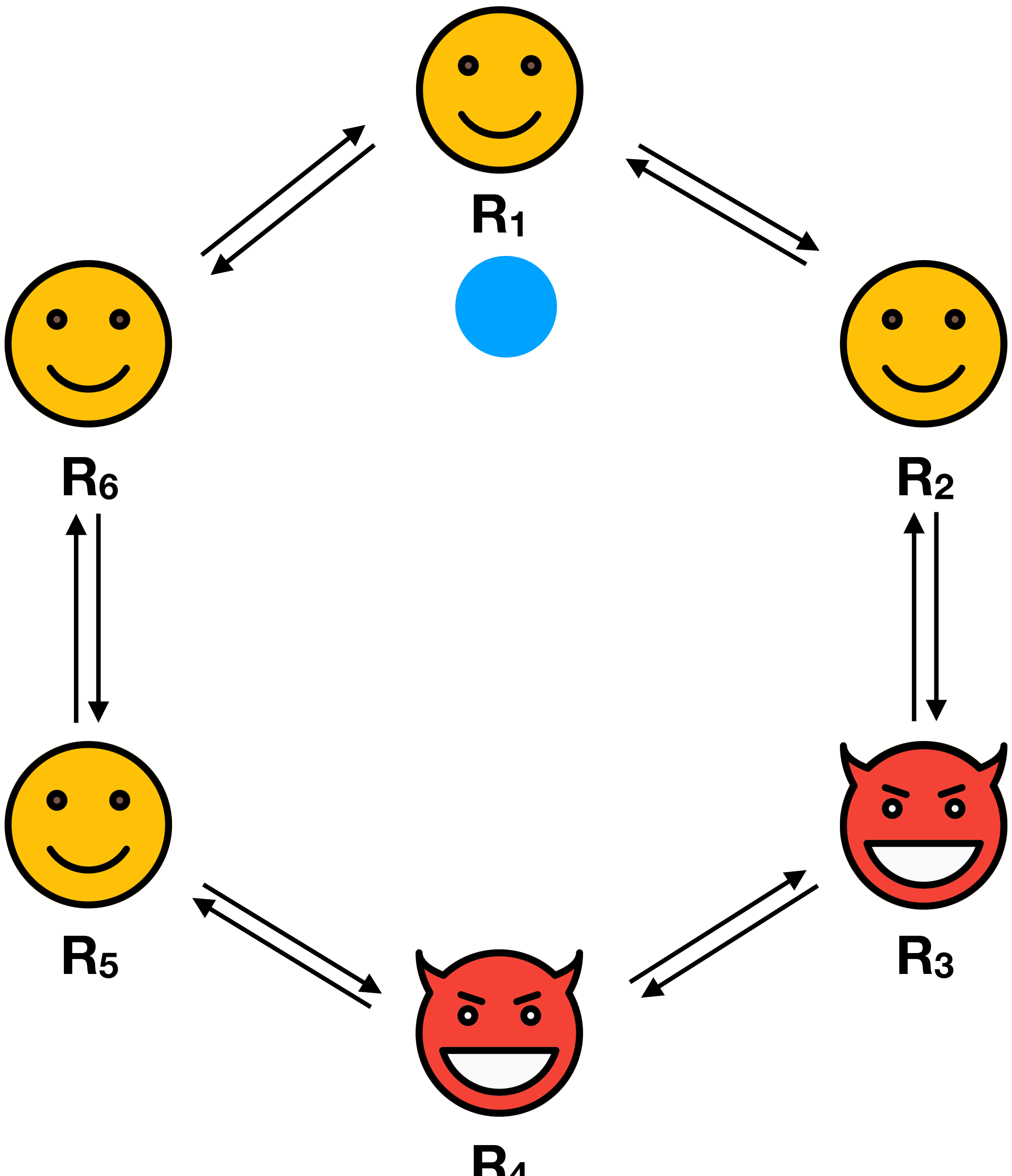


1. Secret sharing [CGS02, BCGHS06]

- distribute inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)

MPQC: two approaches

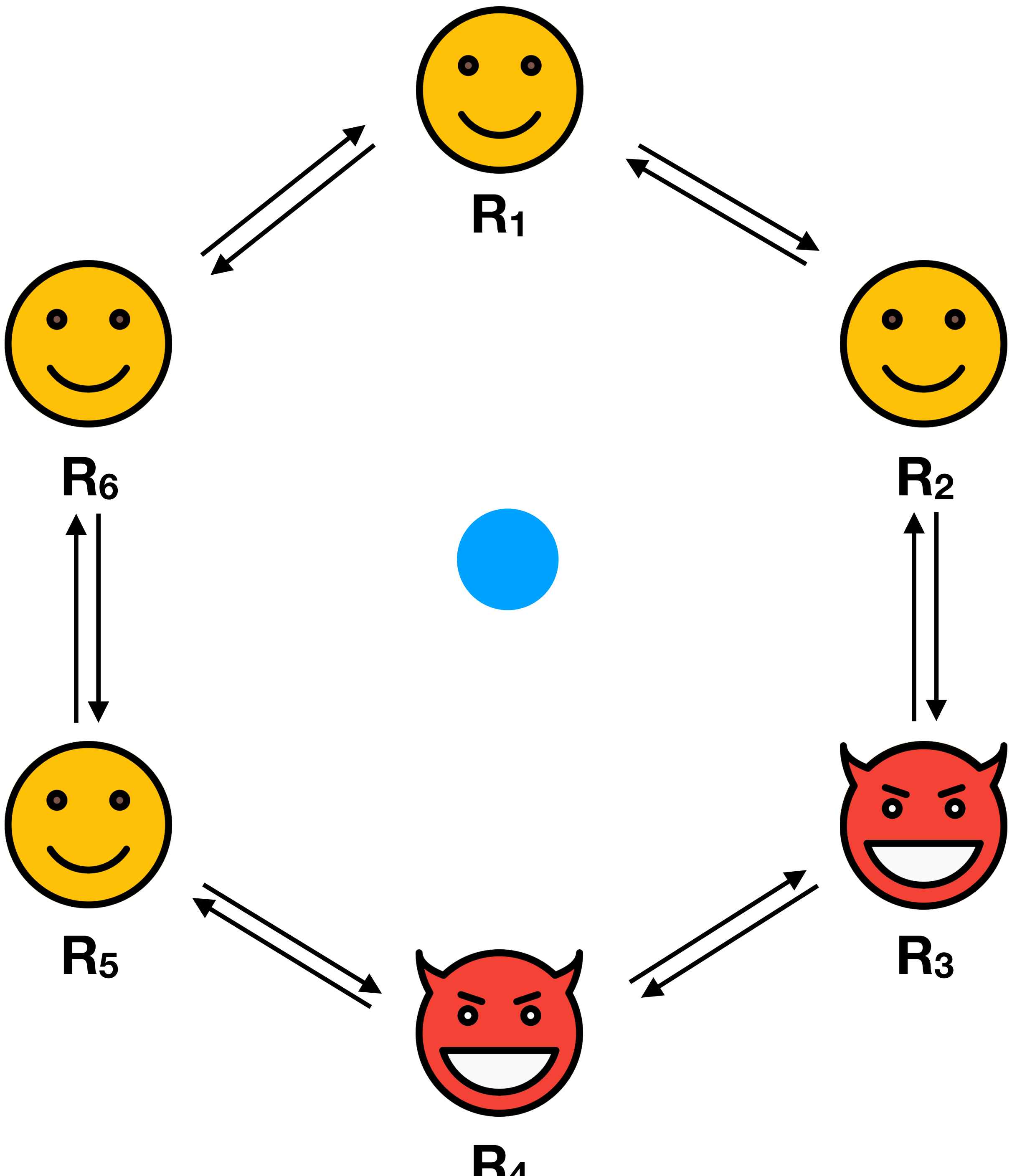


1. Secret sharing [CGS02, BCGHS06]

- distribute inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)

MPQC: two approaches

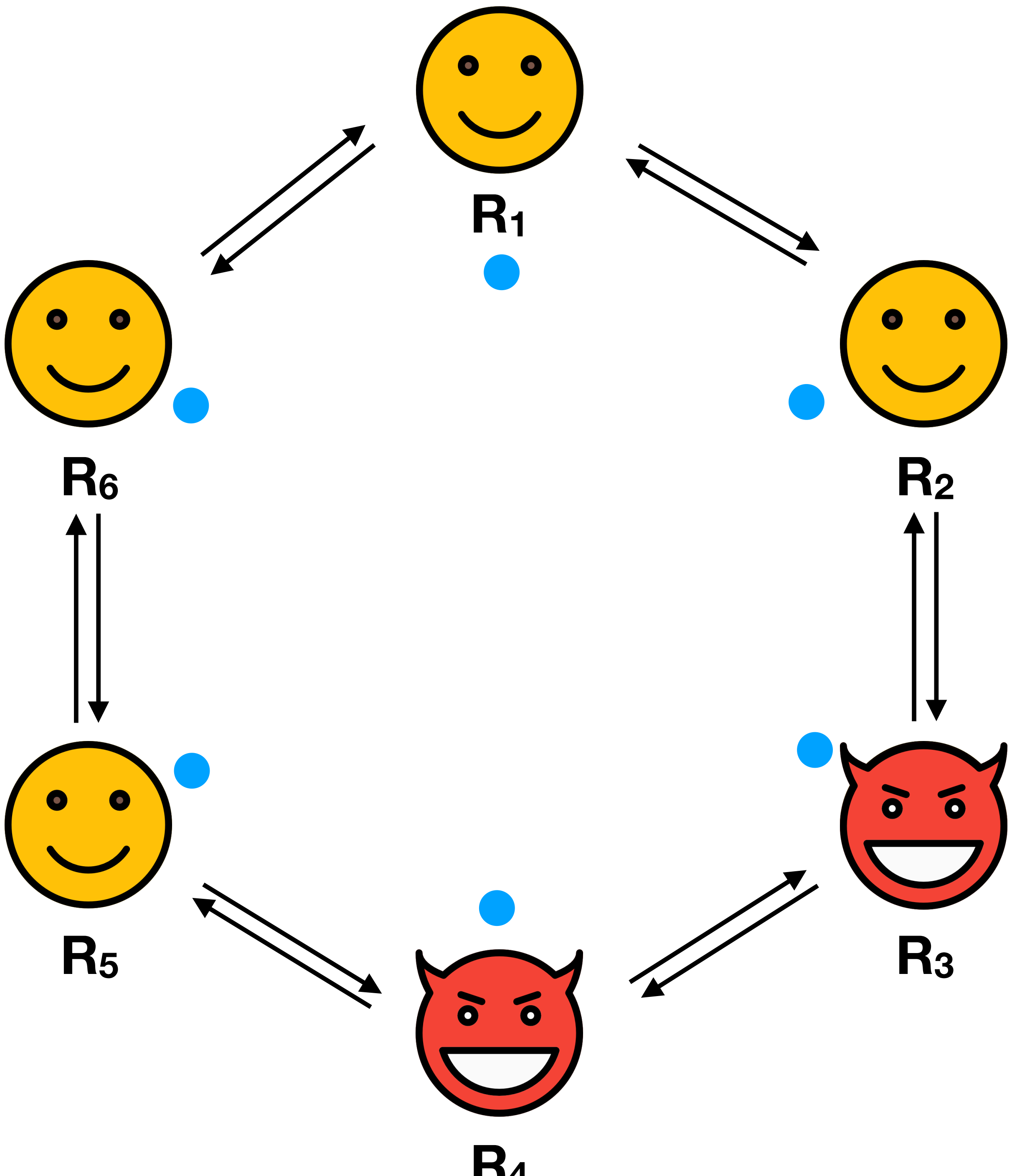


1. Secret sharing [CGS02, BCGHS06]

- distribute inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)

MPQC: two approaches

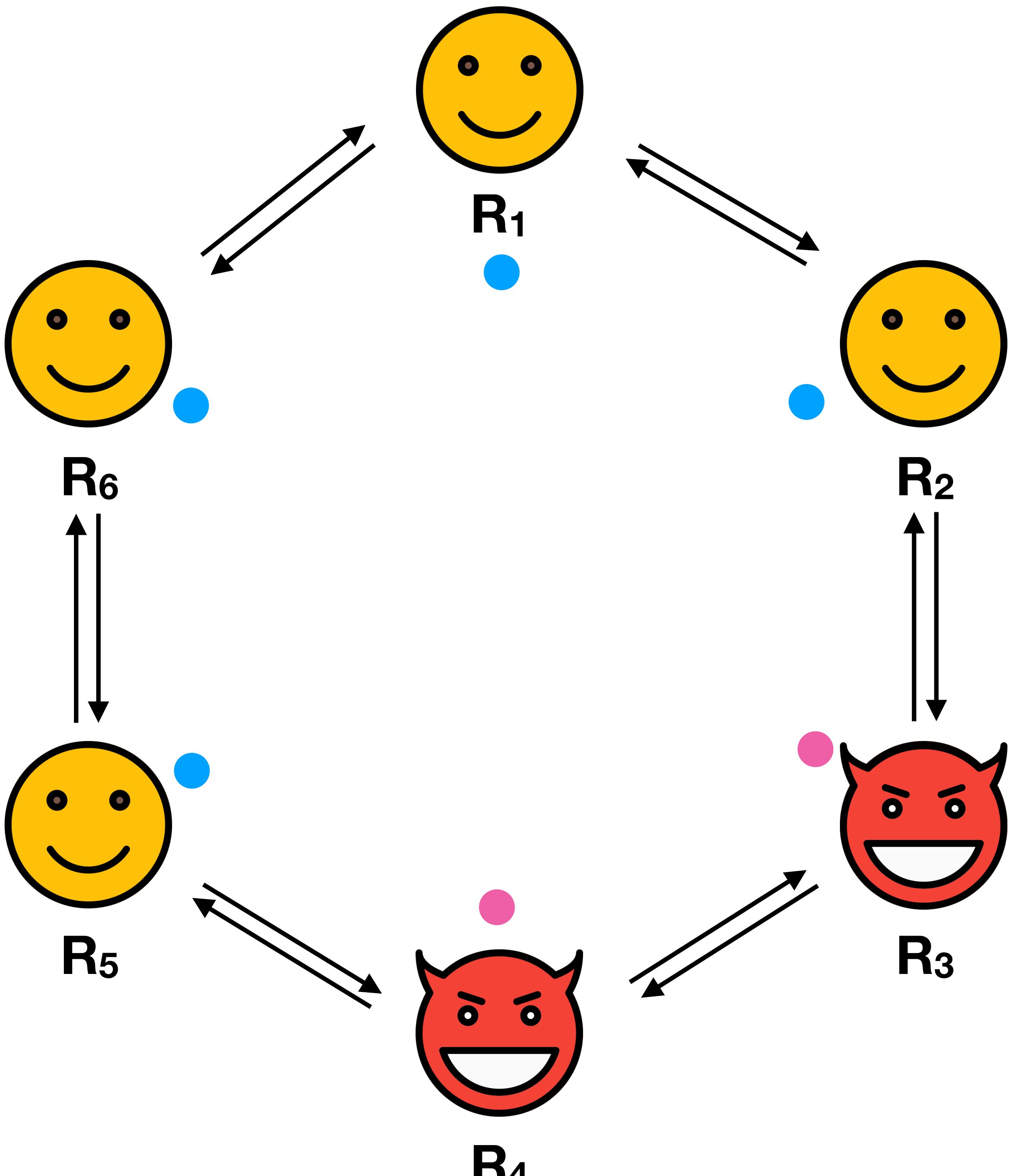


1. Secret sharing [CGS02, BCGHS06]

- distribute inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)

MPQC: two approaches

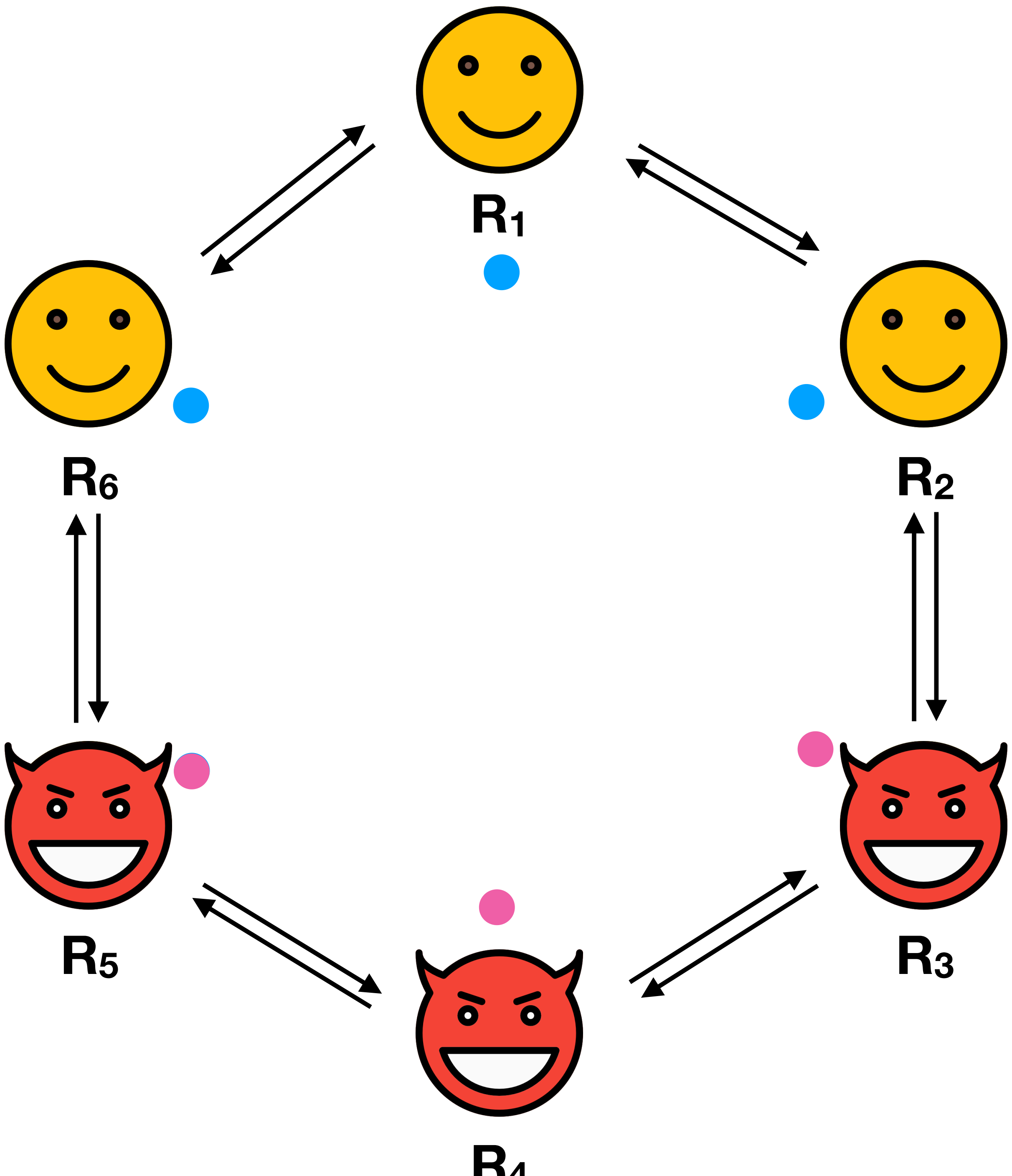


1. Secret sharing [CGS02, BCGHS06]

- distribute inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)

MPQC: two approaches

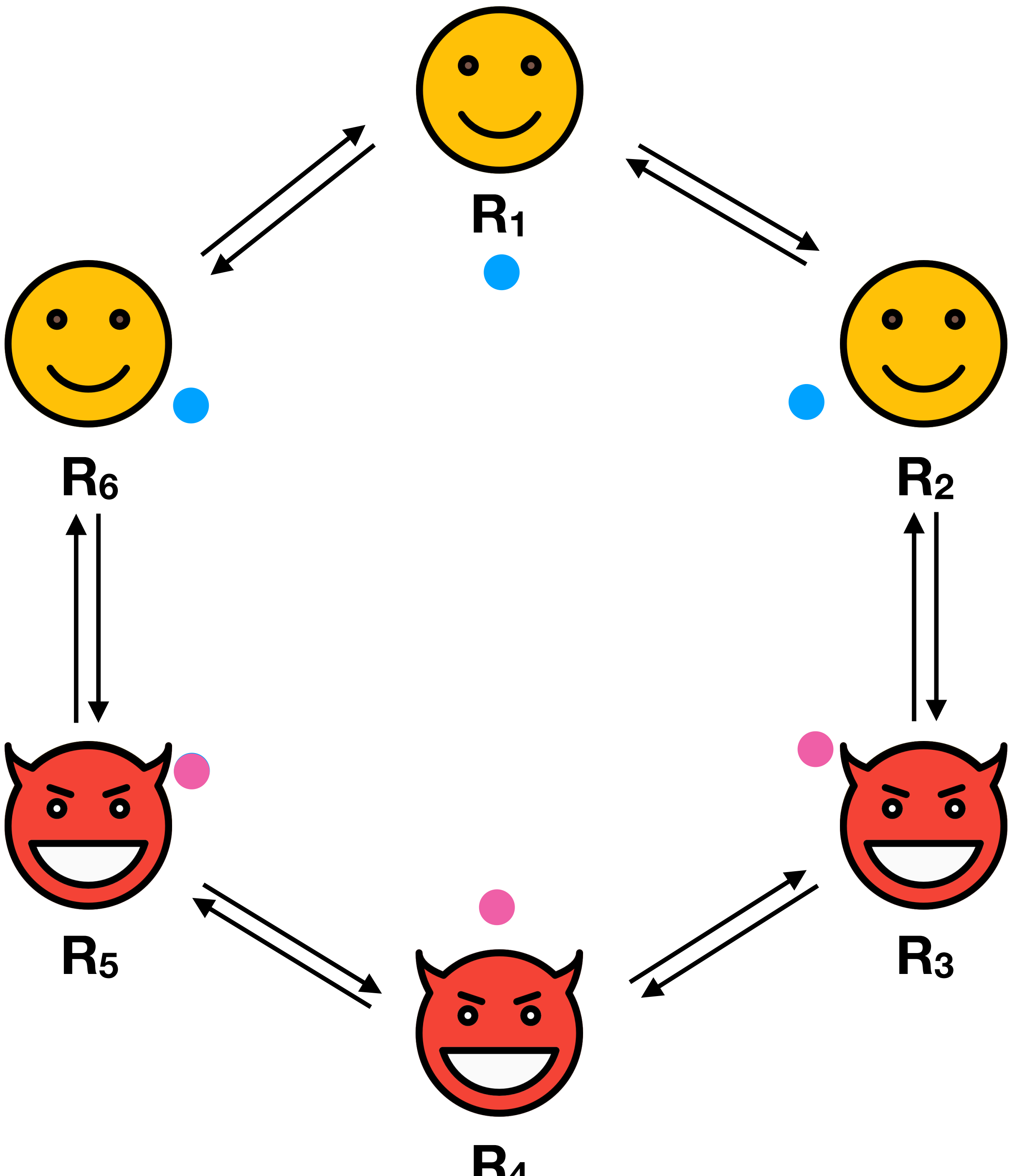


1. Secret sharing [CGS02, BCGHS06]

- distribute inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)

MPQC: two approaches

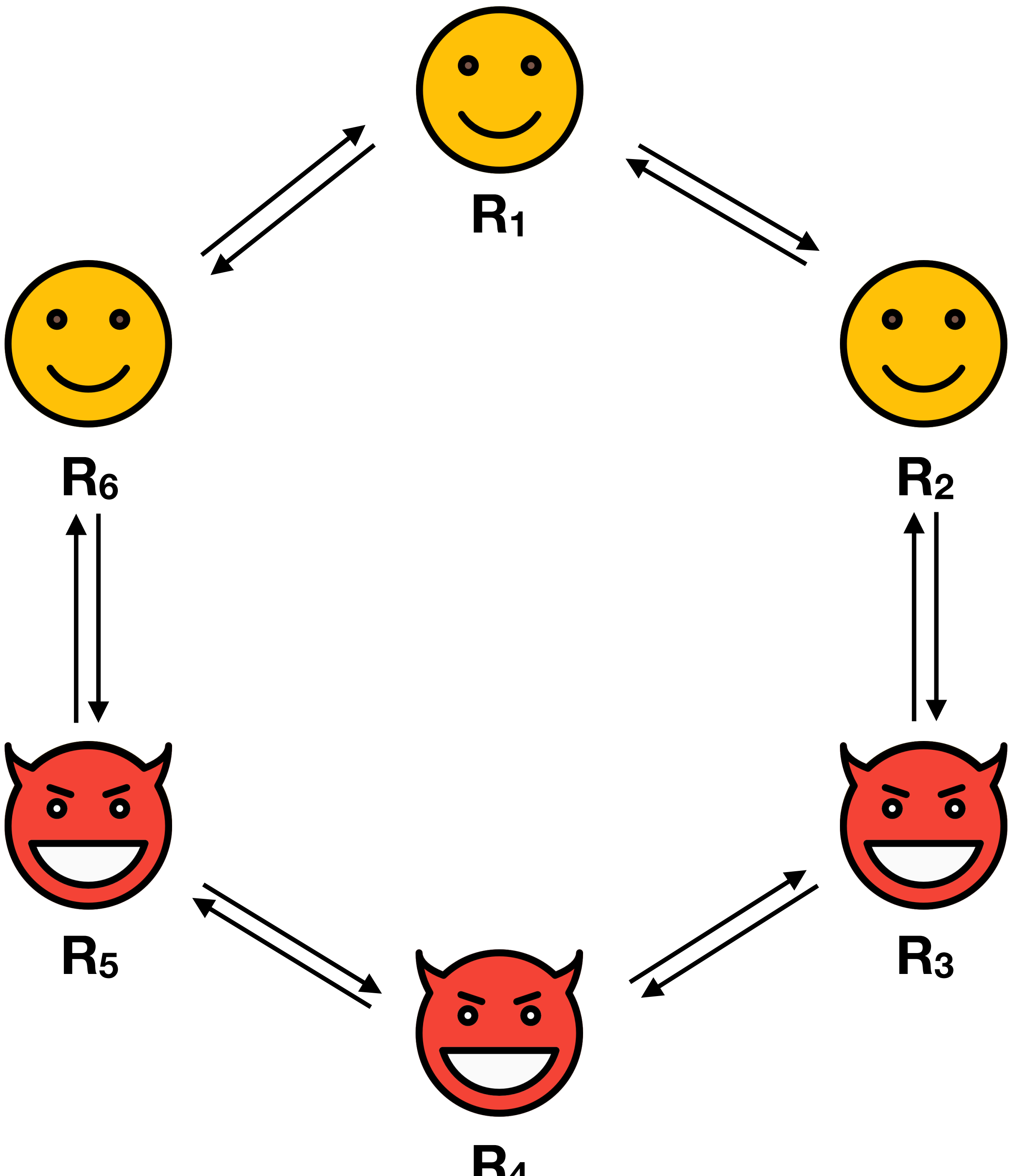


1. Secret sharing [CGS02, BCGHS06]

- distribute inputs
- up to <k/2 dishonest

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)

MPQC: two approaches



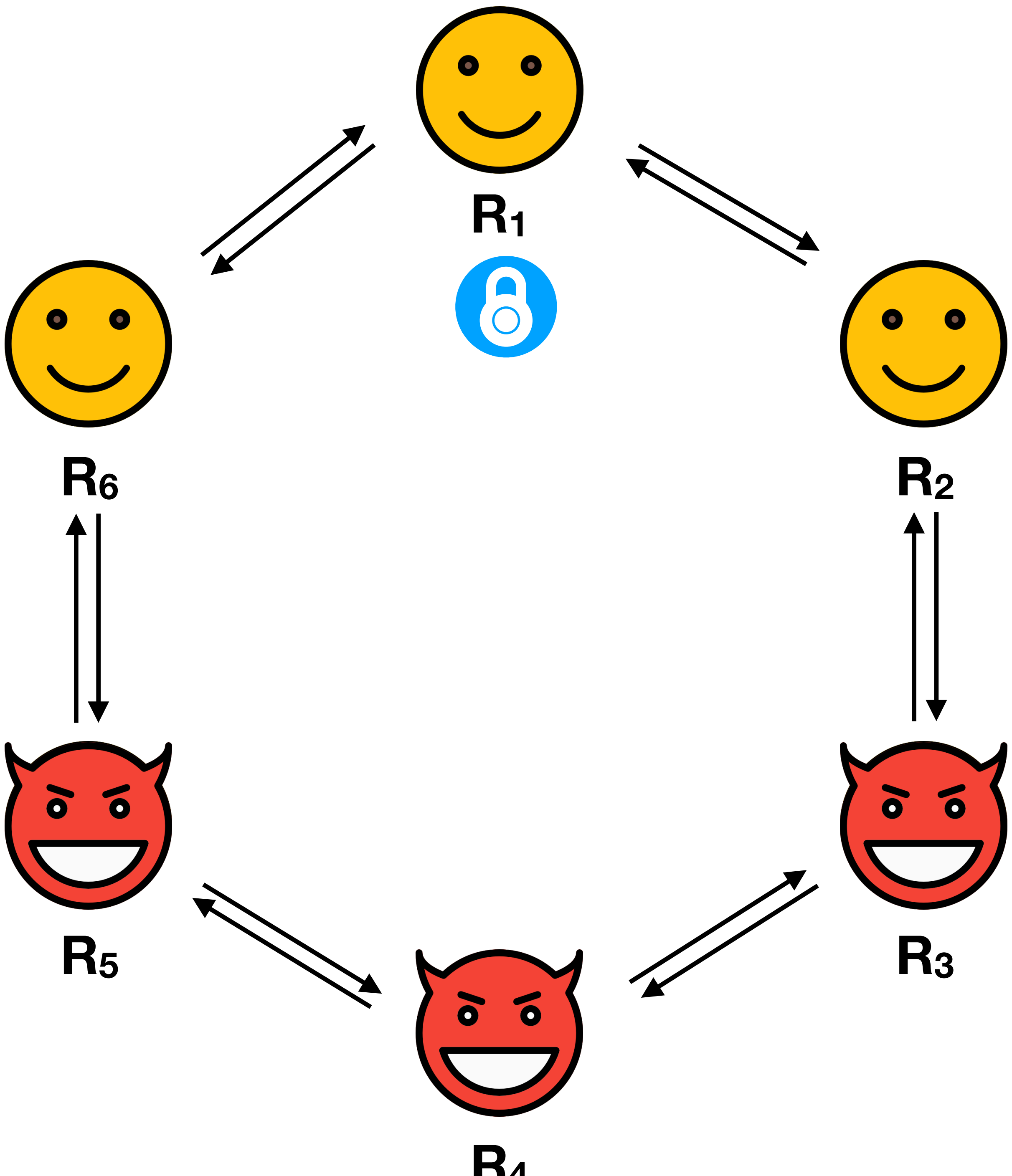
1. Secret sharing [CGS02, BCGHS06]

- distribute inputs
- up to <k/2 dishonest

2. Authentication [DNS12]

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)
[DNS12] Dupuis, Nielsen, and Salvail. Actively secure two-party evaluation of any quantum operation. (CRYPTO 2012)

MPQC: two approaches



1. Secret sharing [CGS02, BCGHS06]

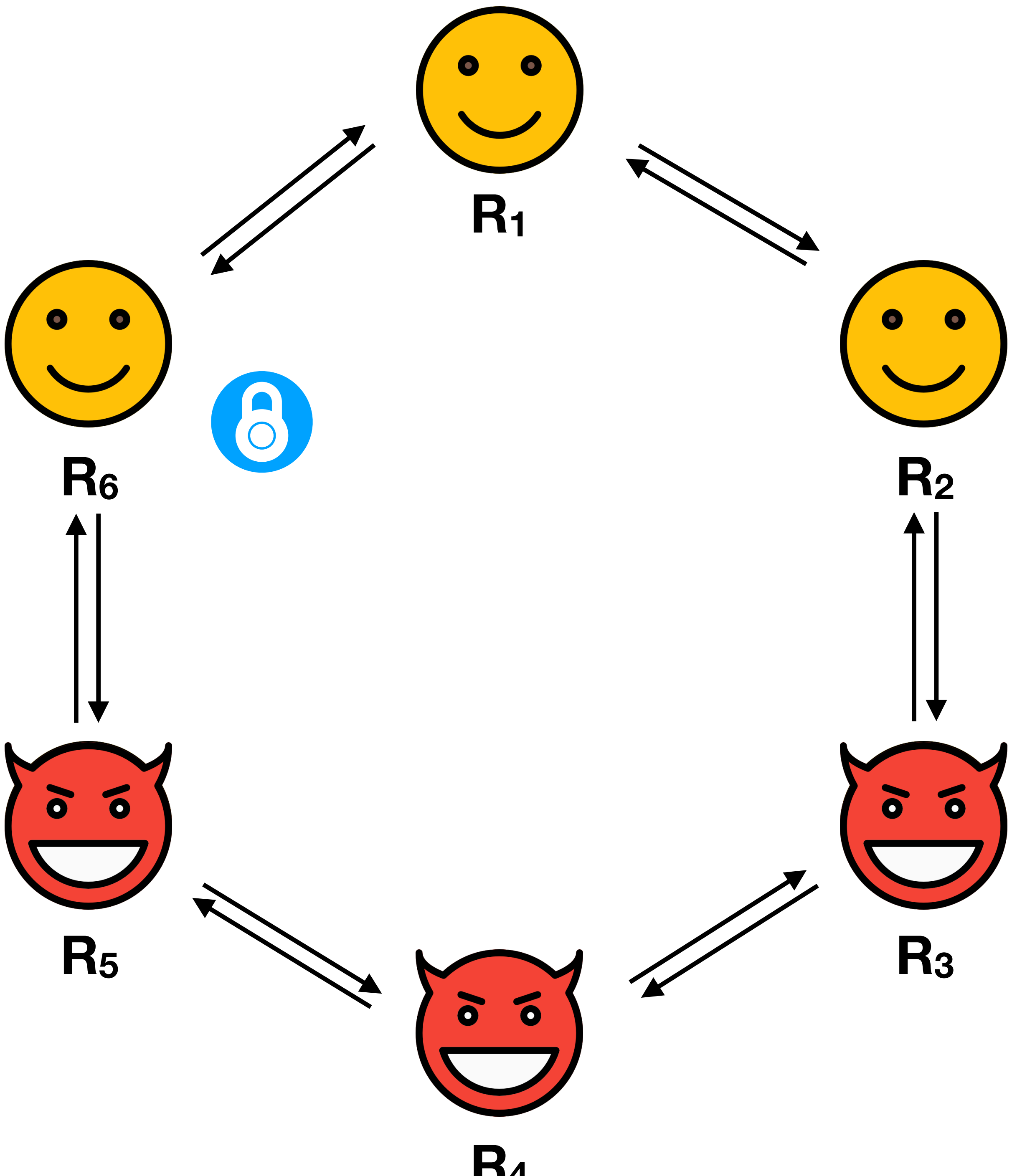
- distribute inputs
- up to <k/2 dishonest

2. Authentication [DNS12]

- protect inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)
[DNS12] Dupuis, Nielsen, and Salvail. Actively secure two-party evaluation of any quantum operation. (CRYPTO 2012)

MPQC: two approaches



1. Secret sharing [CGS02, BCGHS06]

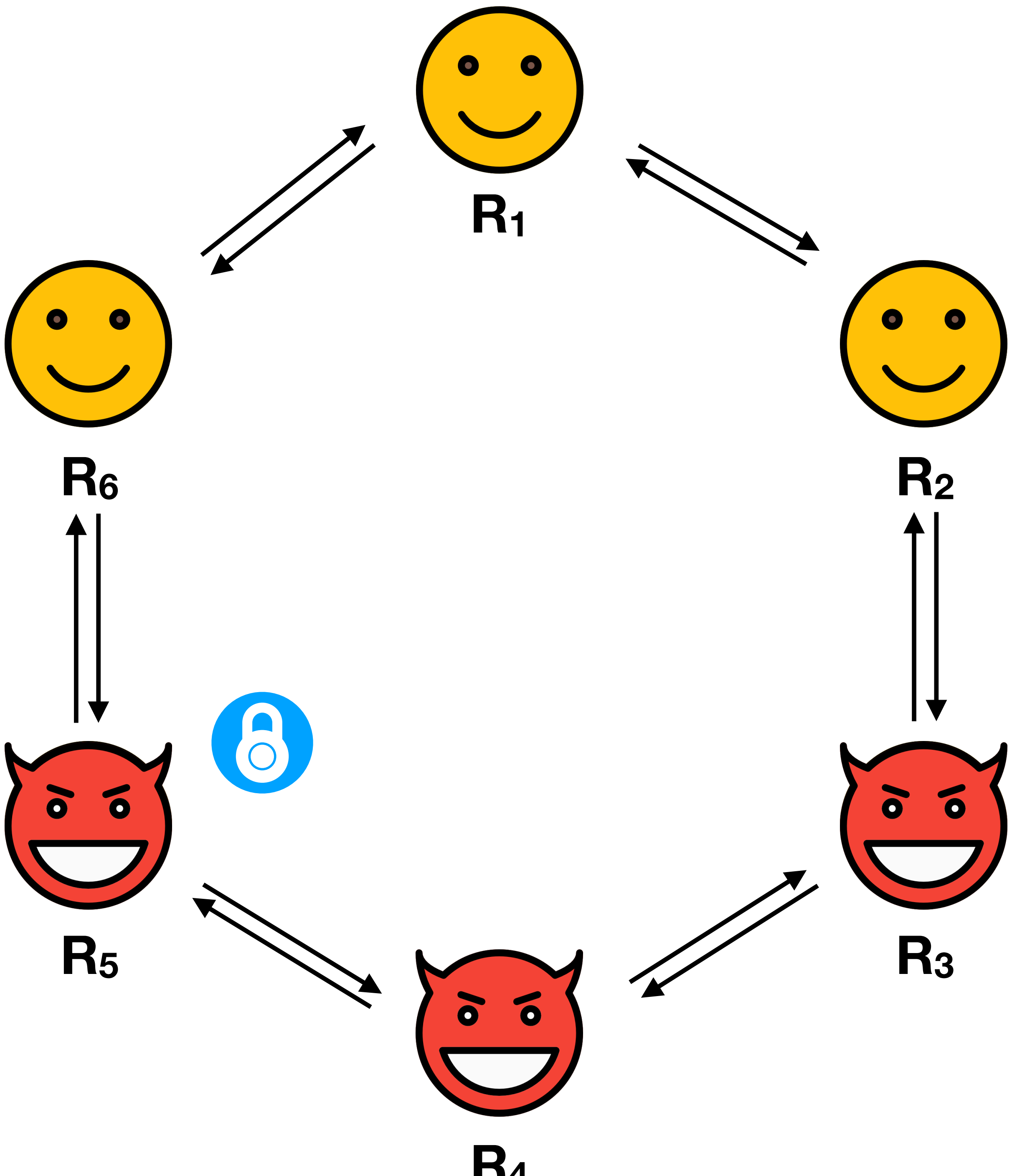
- distribute inputs
- up to <k/2 dishonest

2. Authentication [DNS12]

- protect inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)
[DNS12] Dupuis, Nielsen, and Salvail. Actively secure two-party evaluation of any quantum operation. (CRYPTO 2012)

MPQC: two approaches



1. Secret sharing [CGS02, BCGHS06]

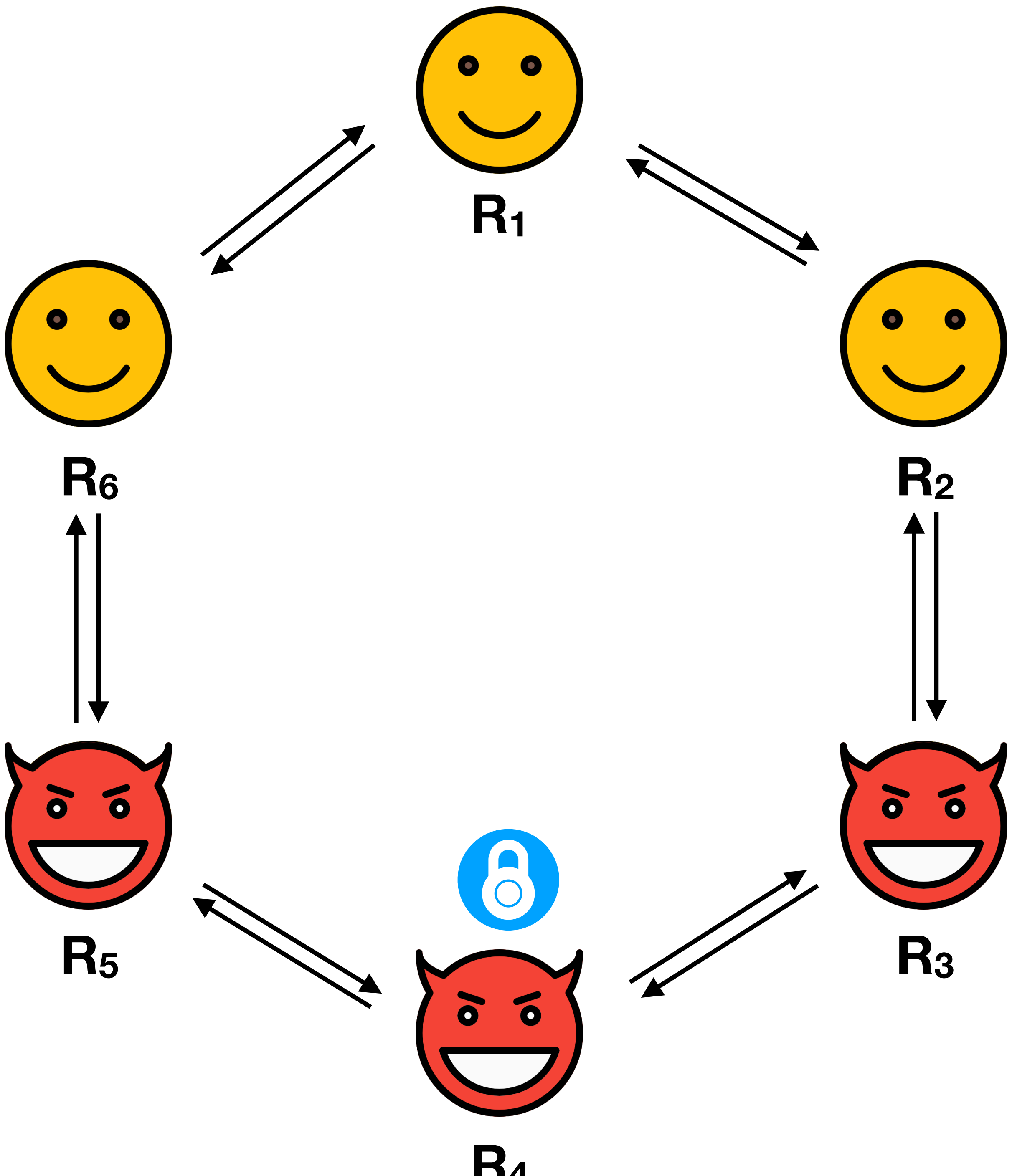
- distribute inputs
- up to <k/2 dishonest

2. Authentication [DNS12]

- protect inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)
[DNS12] Dupuis, Nielsen, and Salvail. Actively secure two-party evaluation of any quantum operation. (CRYPTO 2012)

MPQC: two approaches



1. Secret sharing [CGS02, BCGHS06]

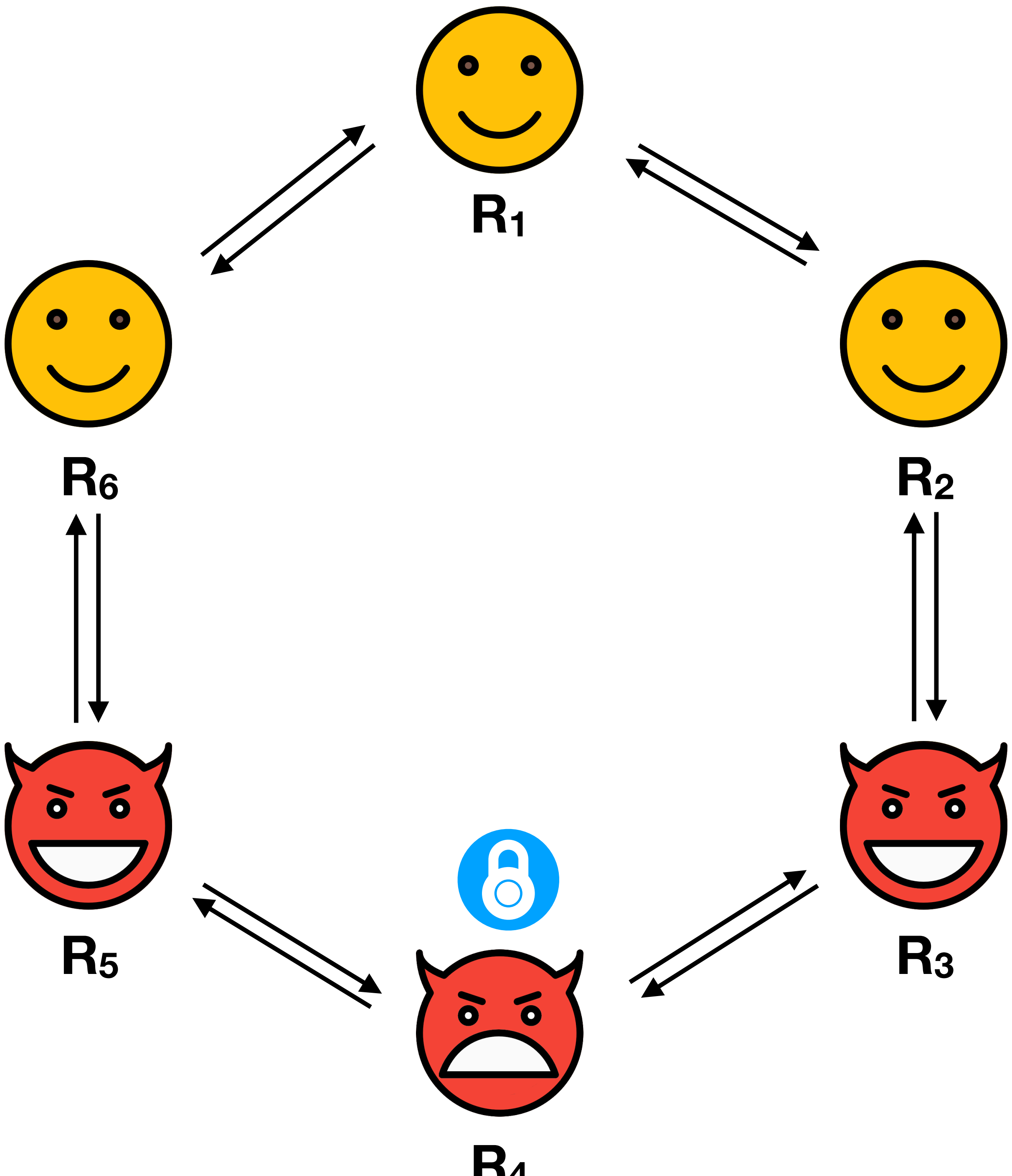
- distribute inputs
- up to <k/2 dishonest

2. Authentication [DNS12]

- protect inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)
[DNS12] Dupuis, Nielsen, and Salvail. Actively secure two-party evaluation of any quantum operation. (CRYPTO 2012)

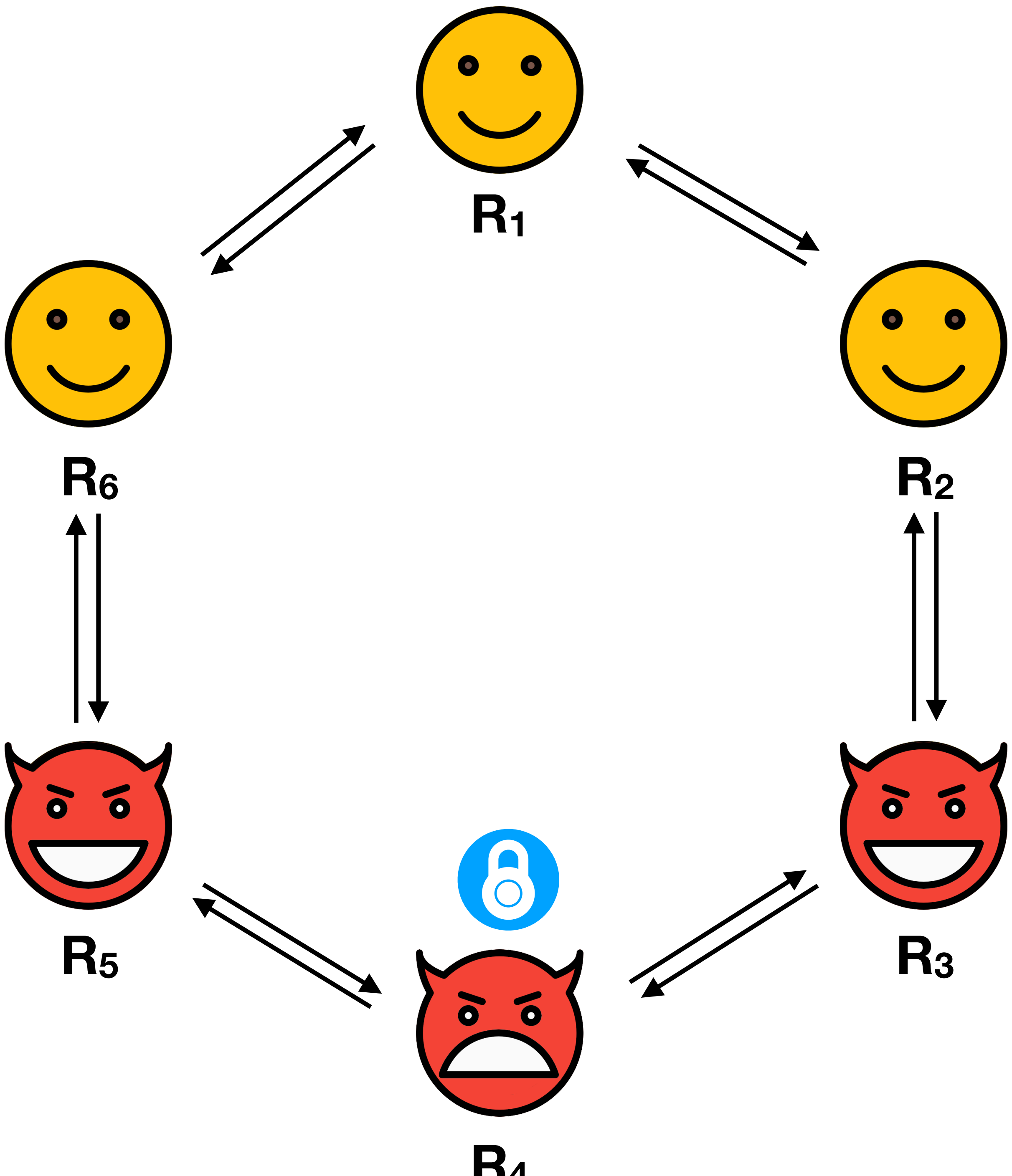
MPQC: two approaches



1. Secret sharing [CGS02, BCGHS06]
   - distribute inputs
   - up to <k/2 dishonest
2. Authentication [DNS12]
   - protect inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)
[DNS12] Dupuis, Nielsen, and Salvail. Actively secure two-party evaluation of any quantum operation. (CRYPTO 2012)

MPQC: two approaches



1. Secret sharing [CGS02, BCGHS06]

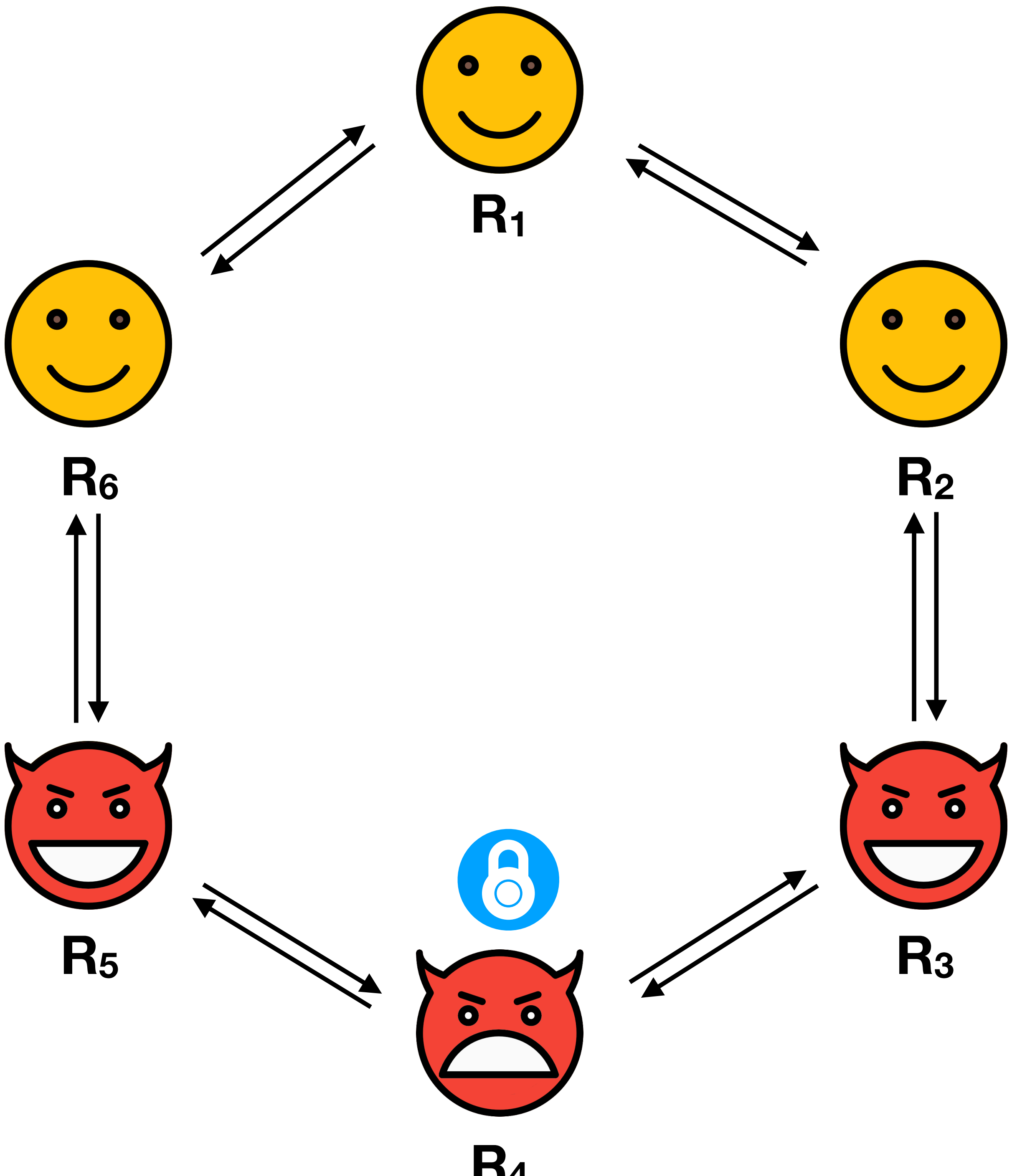
- distribute inputs
- up to <k/2 dishonest

2. Authentication [DNS12]

- protect inputs

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)
[DNS12] Dupuis, Nielsen, and Salvail. Actively secure two-party evaluation of any quantum operation. (CRYPTO 2012)

MPQC: two approaches



1. Secret sharing [CGS02, BCGHS06]
   - distribute inputs
   - up to <k/2 dishonest
2. Authentication [DNS12]
   - protect inputs
   - hope: up to k-1 dishonest

[CGS02] Crépeau, Gottesman, and Smith. Secure multi-party quantum computation. (STOC 2002)
[BCGHS06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith. (FOCS 2006)
[DNS12] Dupuis, Nielsen, and Salvail. Actively secure two-party evaluation of any quantum operation. (CRYPTO 2012)

Introduction

Authentication

Computation

Magic-state generation

Summary

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Key: $C \in_R \mathrm{Clifford}_{n+1}$

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

SUBGROUP OF UNITARIES

Key: $C \in_R \mathrm{Clifford}_{n+1}$

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Key: $C \in_R \text{Clifford}_{n+1}$

SUBGROUP OF UNITARIES GENERATED BY H, $\sqrt{Z}$, CNOT

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Key: $C \in_R \text{Clifford}_{n+1}$

SUBGROUP OF UNITARIES
GENERATED BY H, √ Z, CNOT
LOOKS "RANDOM"

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Key: $C \in_R \mathrm{Clifford}_{n+1}$

SUBGROUP OF UNITARIES
GENERATED BY H, $\sqrt{Z}$, CNOT
LOOKS "RANDOM"

Encoding: $|\psi\rangle \mapsto C\left(|\psi\rangle \otimes |0\rangle^{\otimes n}\right)$

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Key: $C \in_R \text{Clifford}_{n+1}$

SUBGROUP OF UNITARIES GENERATED BY H, $\sqrt{Z}$, CNOT
LOOKS "RANDOM"

Encoding: $|\psi\rangle \mapsto C\left(|\psi\rangle \otimes |0\rangle^{\otimes n}\right)$

TRAPS

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Key: $C \in_R \text{Clifford}_{n+1}$

SUBGROUP OF UNITARIES GENERATED BY H, $\sqrt{Z}$, CNOT
LOOKS "RANDOM"

Encoding: $|\psi\rangle \mapsto C\left(|\psi\rangle \otimes |0\rangle^{\otimes n}\right)$

TRAPS

Decoding: apply $C^\dagger$, measure traps

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Key: $C \in_R \text{Clifford}_{n+1}$

SUBGROUP OF UNITARIES GENERATED BY H, $\sqrt{Z}$, CNOT
LOOKS "RANDOM"

Encoding: $|\psi\rangle \mapsto C\left(|\psi\rangle \otimes |0\rangle^{\otimes n}\right)$

TRAPS

Decoding: apply $C^\dagger$, measure traps

Theorem (informal): for any A on $n+1$ qubits, the probability that A changes $|\psi\rangle$, but is not detected at decoding is very small (2^{-n}).

Clifford code

Remember Yfke's tutorial: https://www.youtube.com/watch?v=yEjh8qJQqsM

Key: $C \in_R \text{Clifford}_{n+1}$

SUBGROUP OF UNITARIES GENERATED BY H, $\sqrt{Z}$, CNOT
LOOKS "RANDOM"

Encoding: $|\psi\rangle \mapsto C\left(|\psi\rangle \otimes |0\rangle^{\otimes n}\right)$

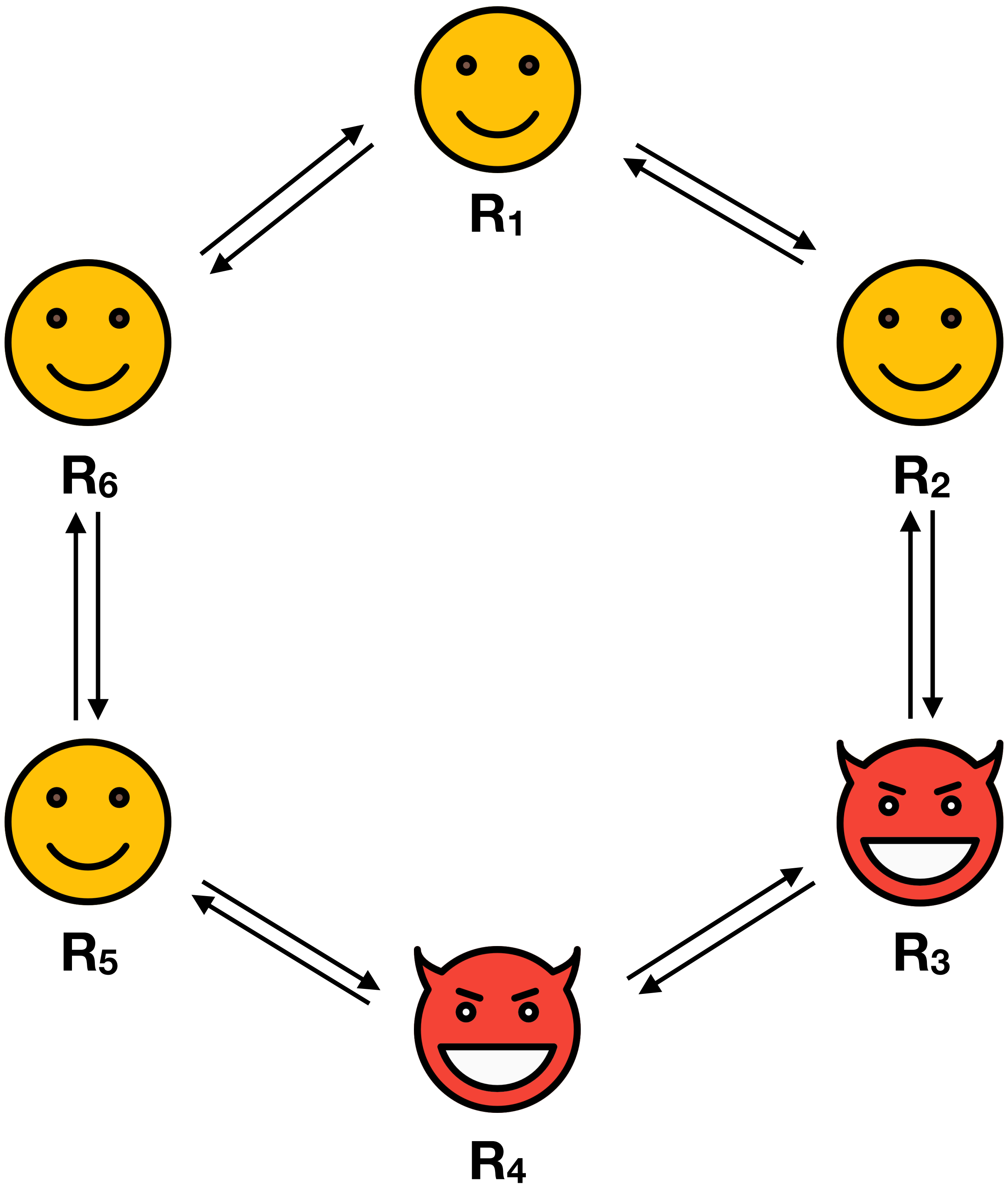
TRAPS

Decoding: apply $C^\dagger$, measure traps

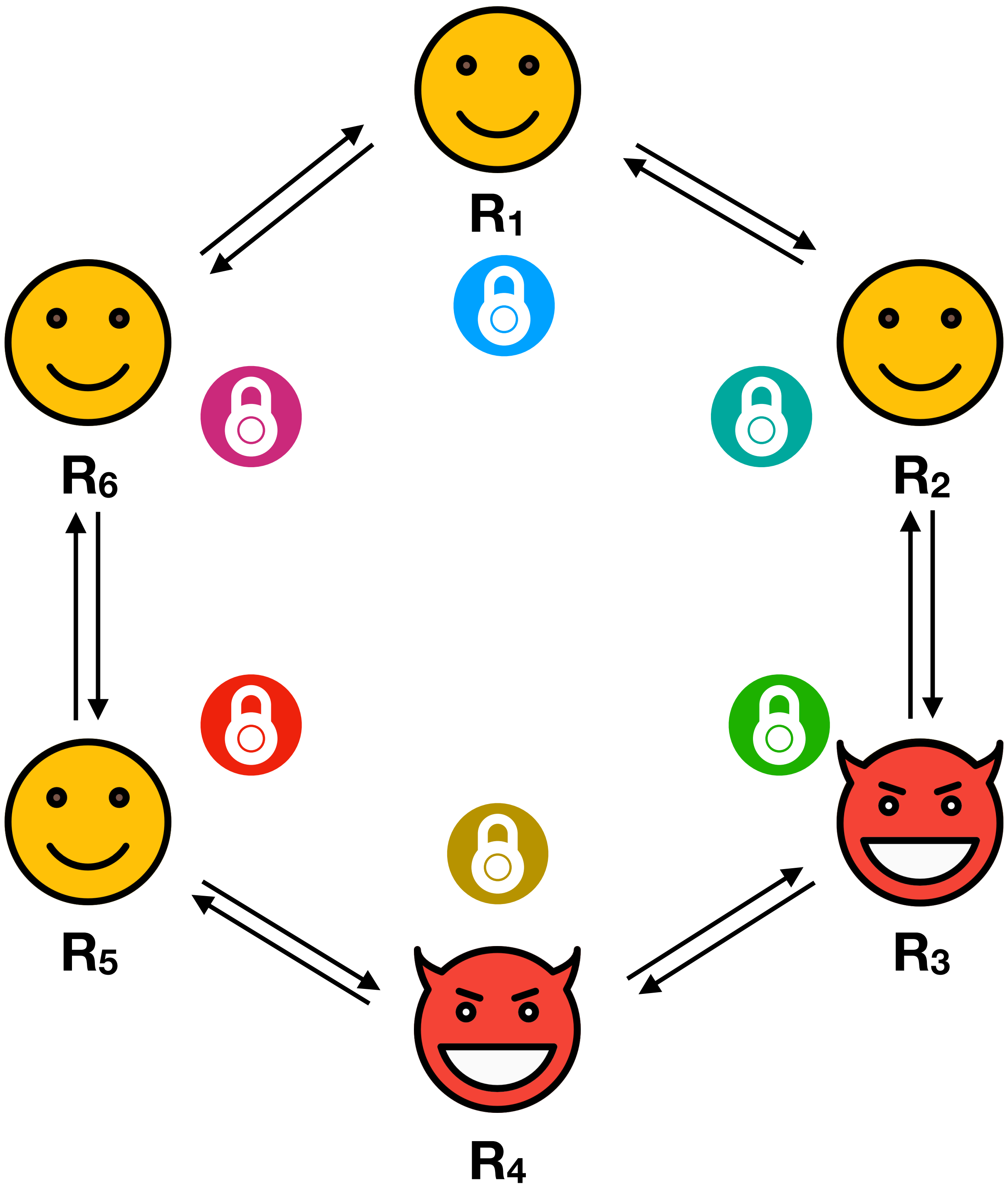
Theorem (informal): for any A on $n+1$ qubits, the probability that A changes $|\psi\rangle$, but is not detected at decoding is very small (2^{-n}).

Bonus: the Clifford code also provides privacy.

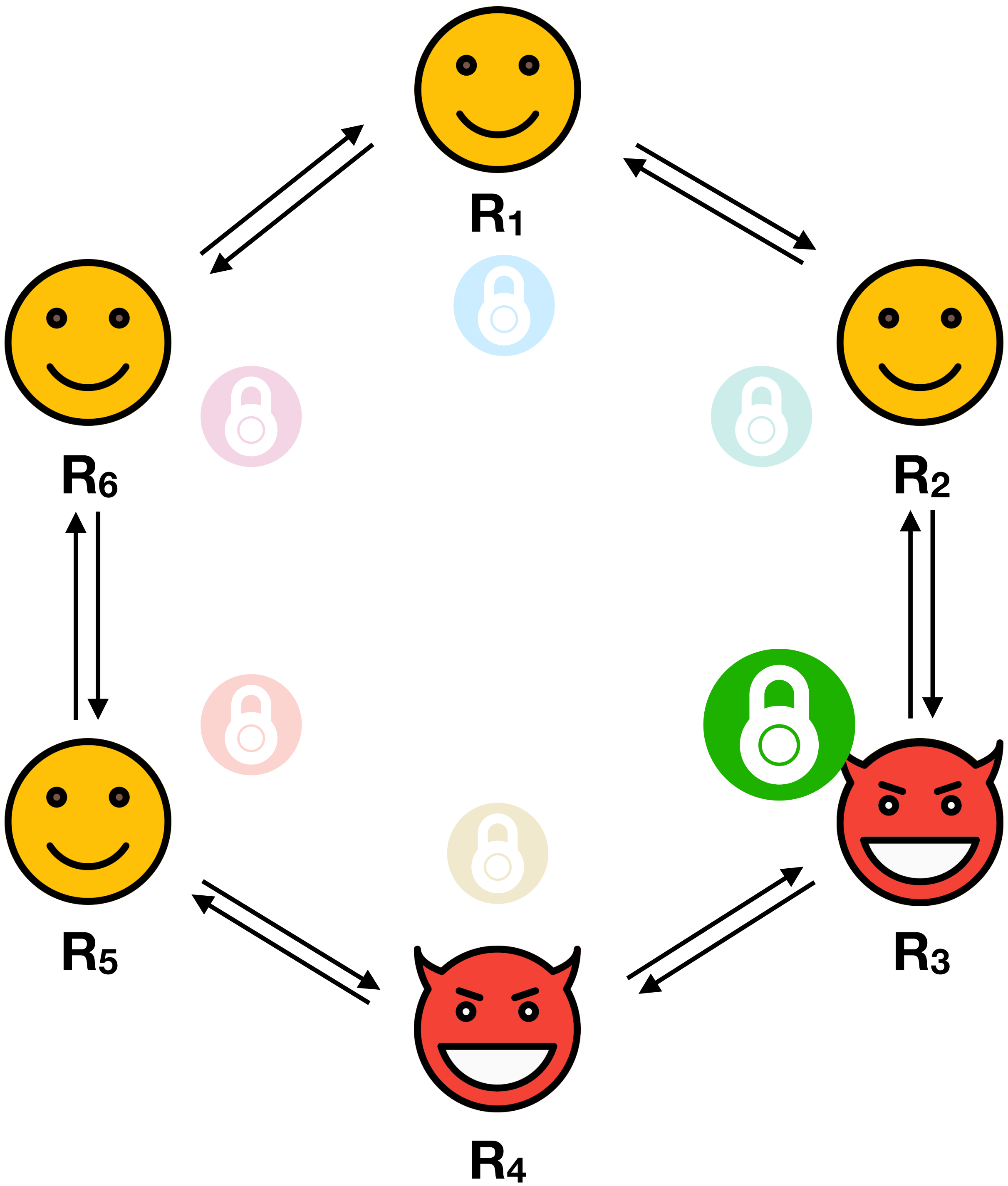
Clifford code in MPQC



Clifford code in MPQC

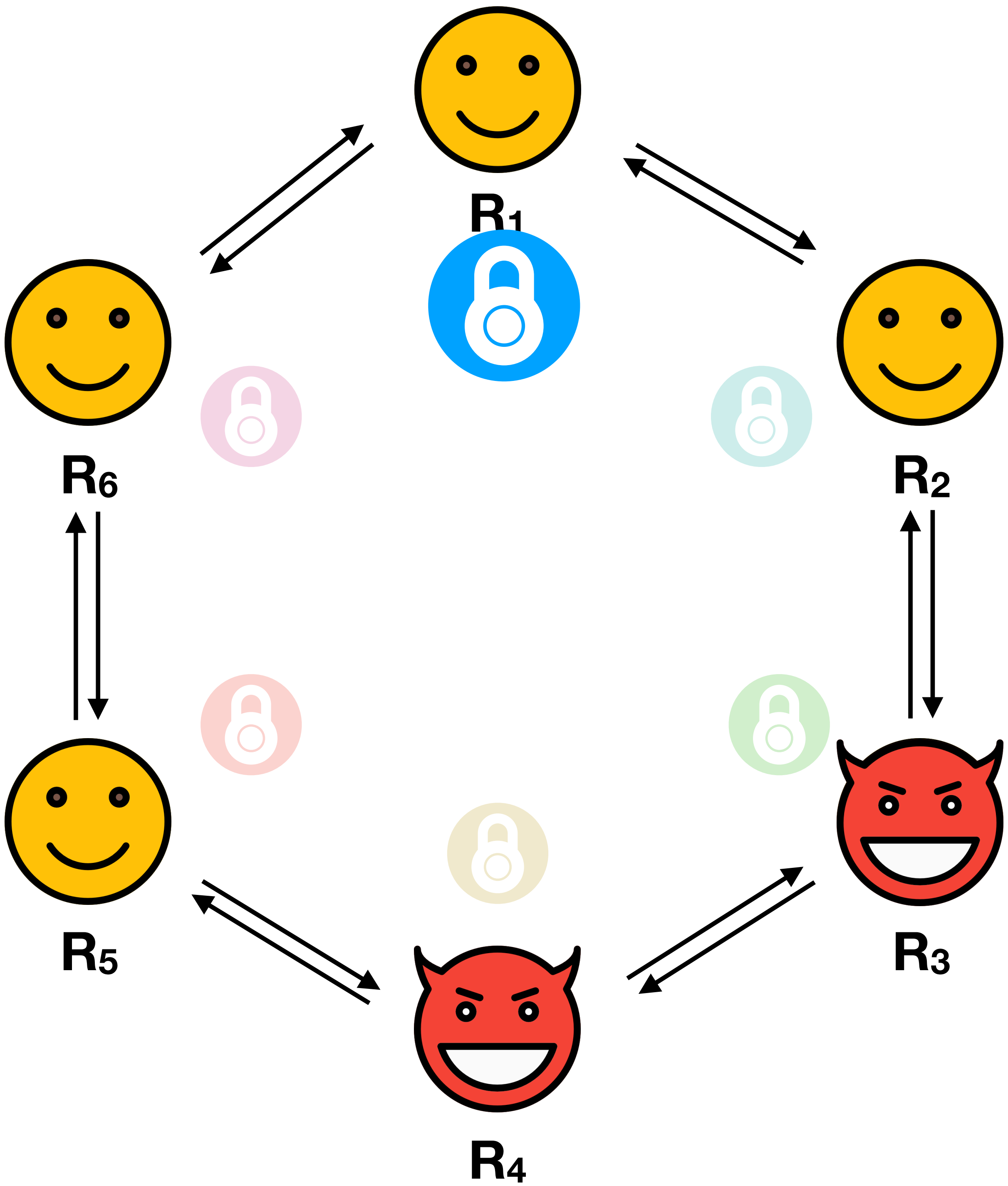


Clifford code in MPQC



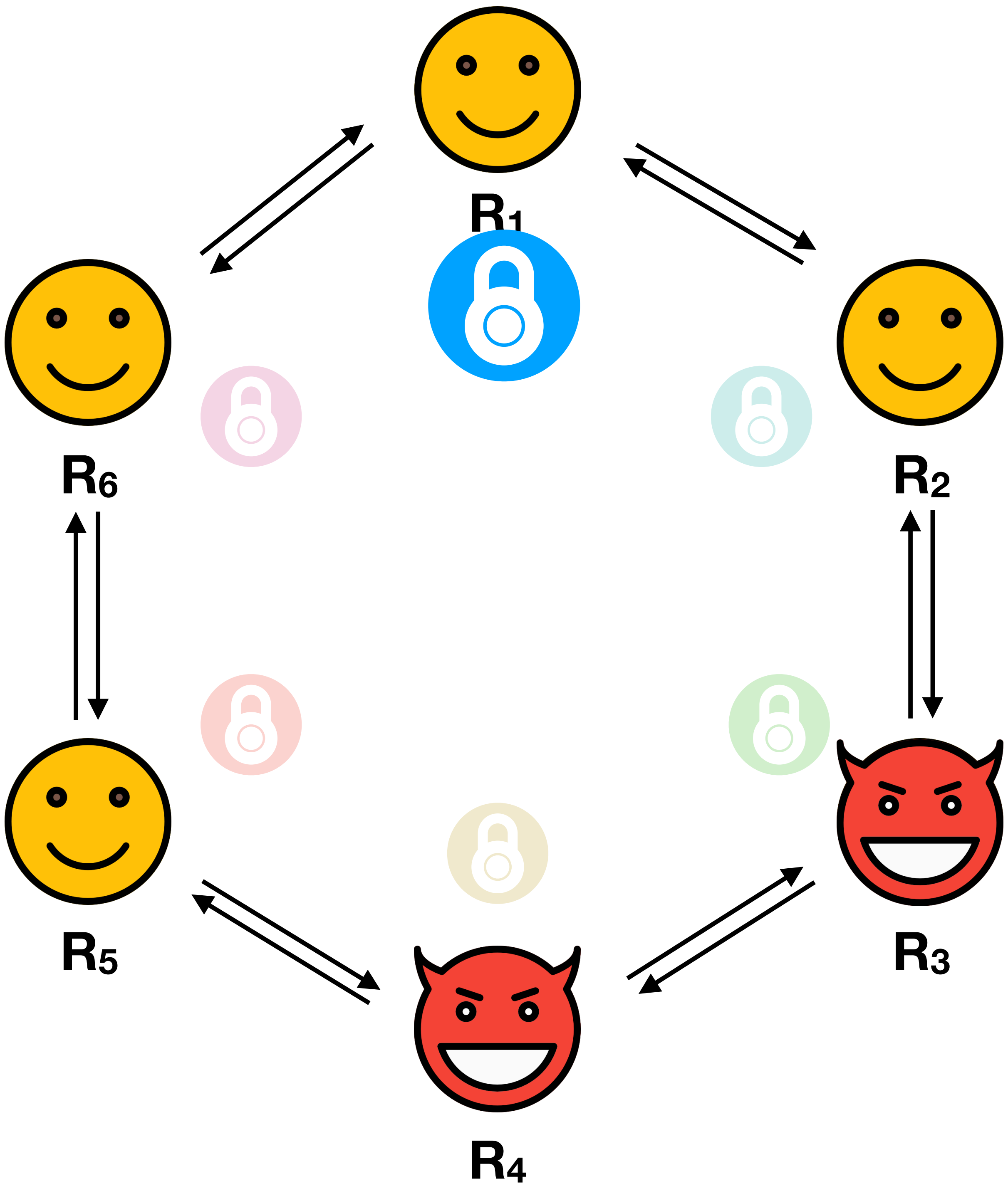
- What if the encoding player is dishonest?

Clifford code in MPQC



- What if the encoding player is dishonest?
- How to do computation? Data is unalterable!

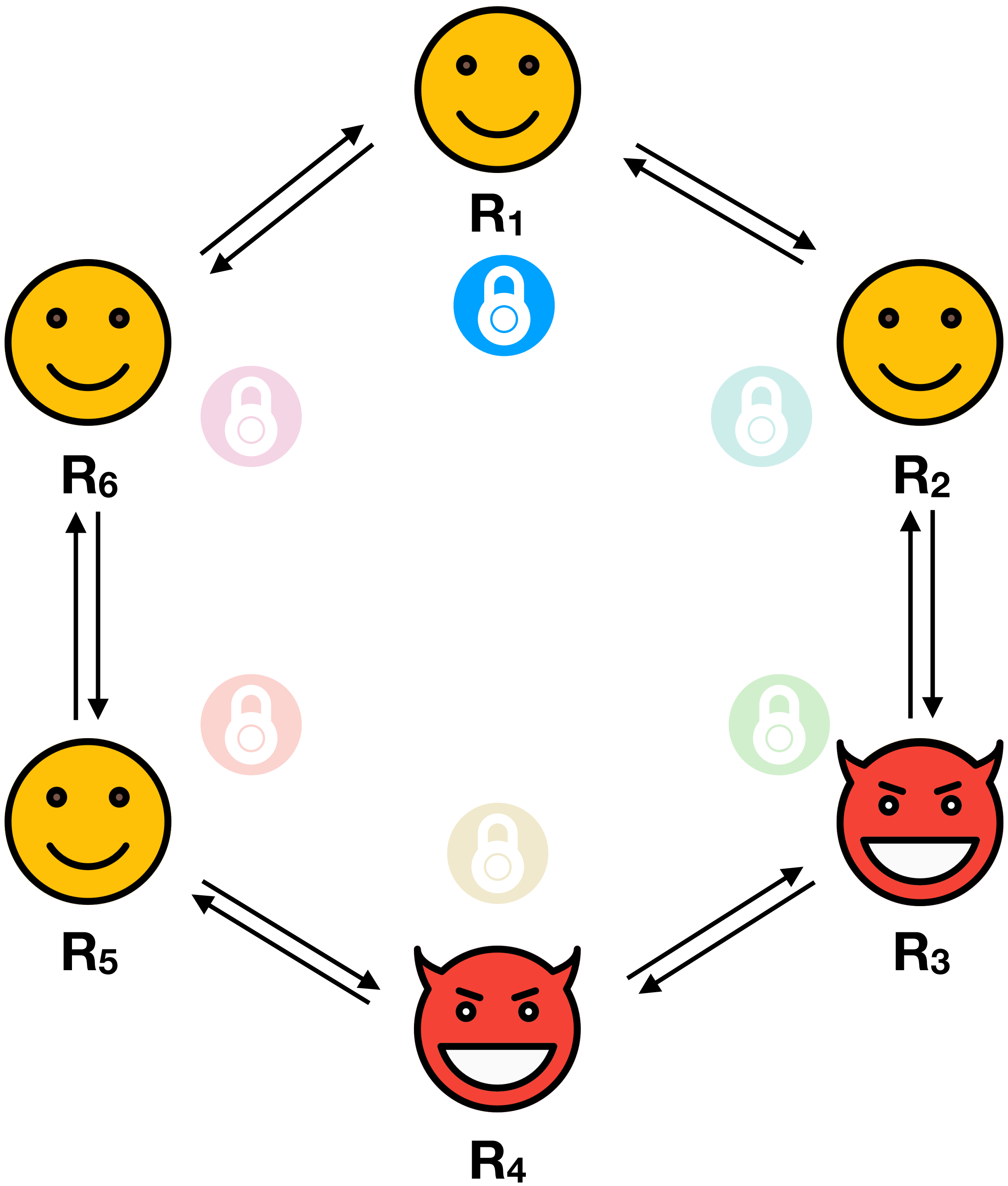
Clifford code in MPQC



- What if the encoding player is dishonest?
- How to do computation? Data is unalterable!

Answers: use classical multi-party computation! 

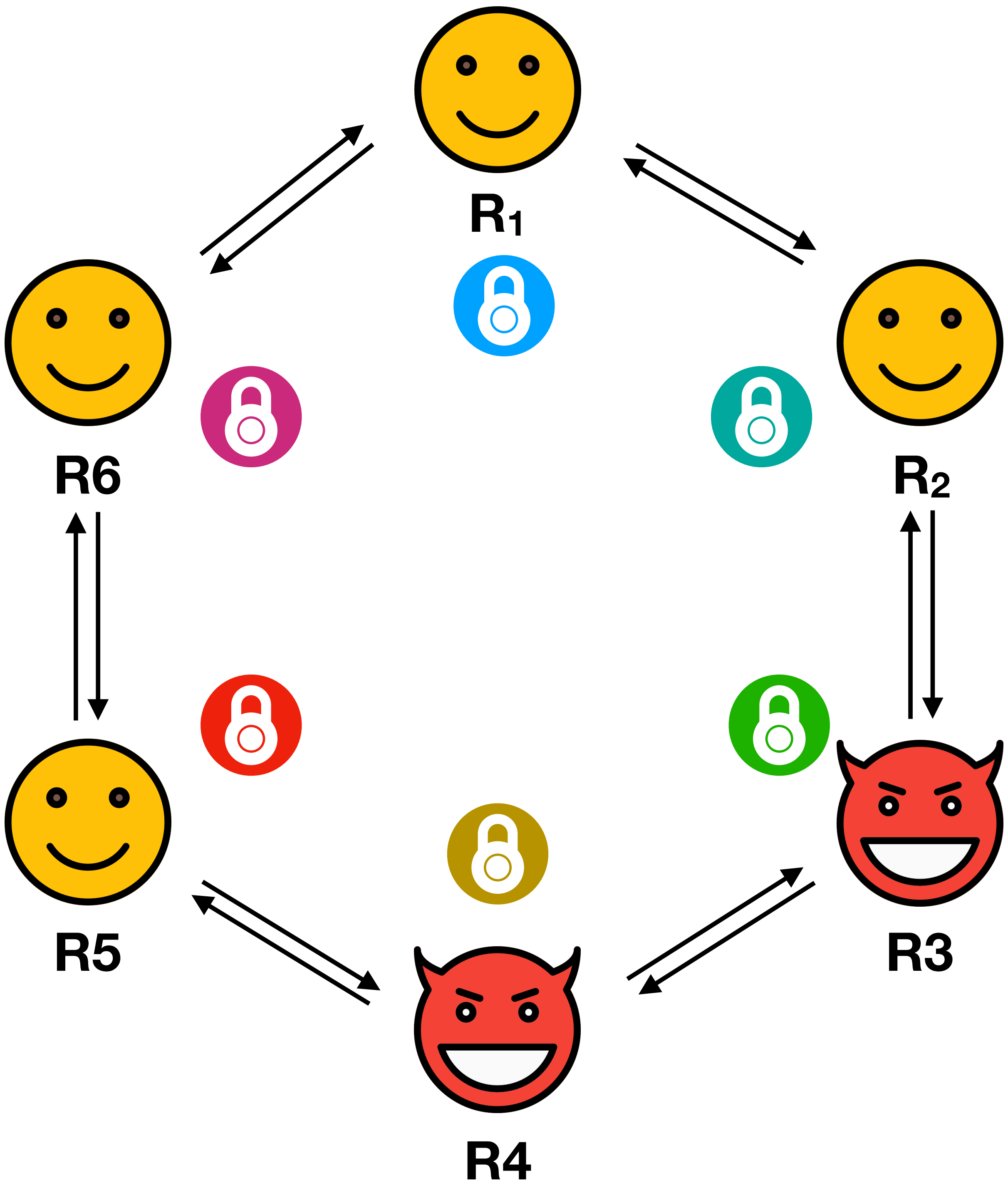
Clifford code in MPQC



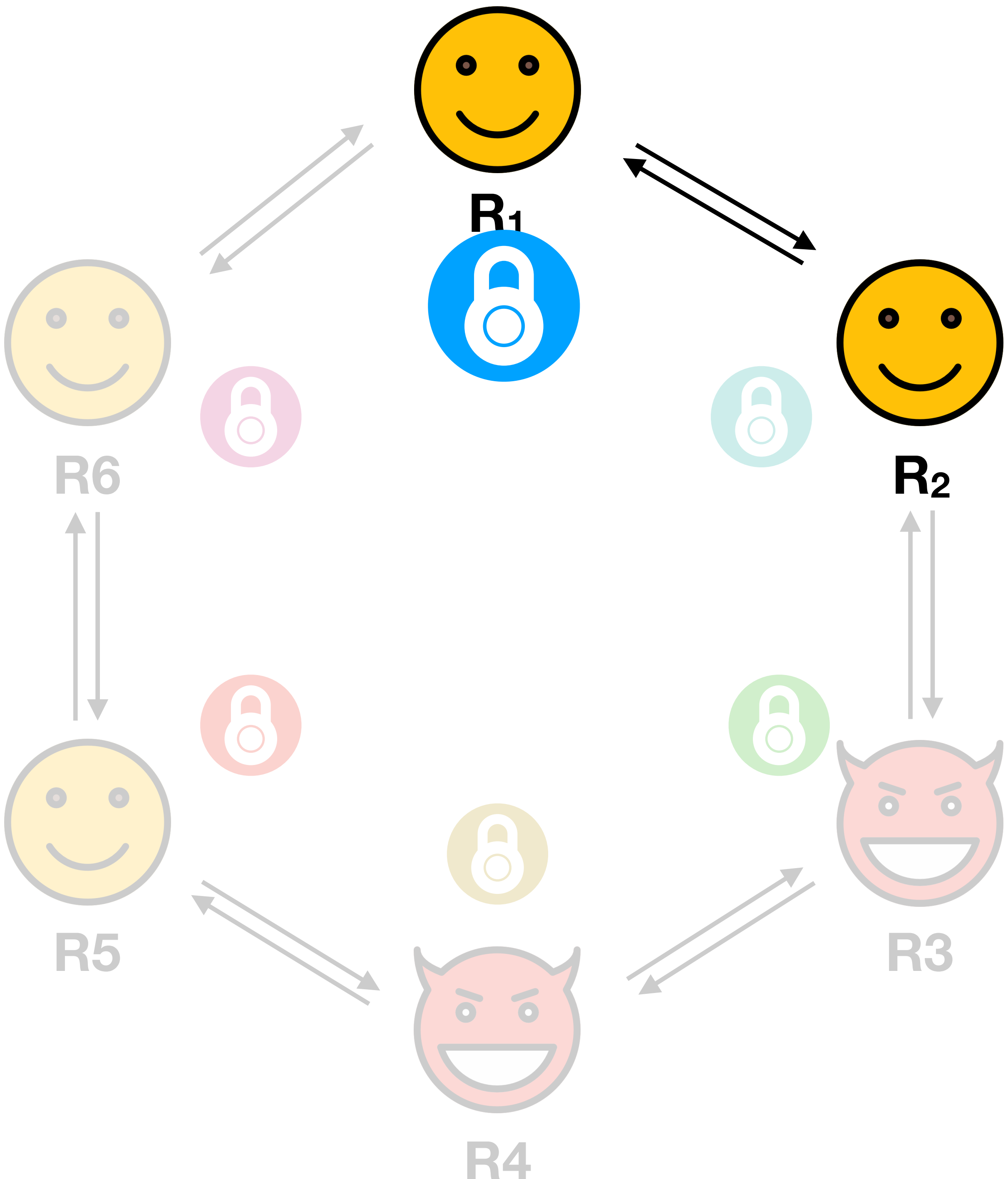
- What if the encoding player is dishonest?
- How to do computation? Data is unalterable!

Answers: use classical multi-party computation! 😇

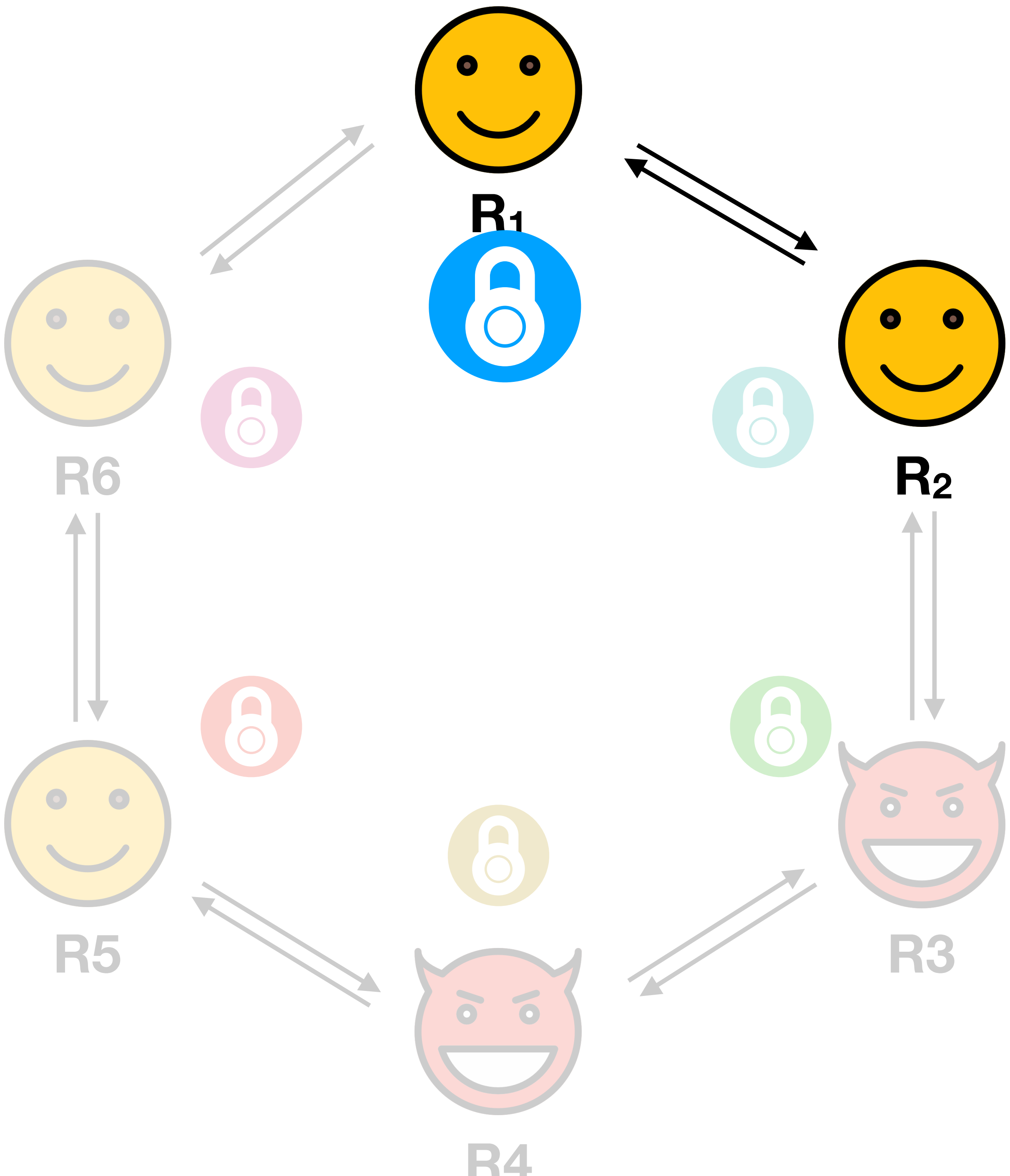
First idea: stack encodings



First idea: stack encodings

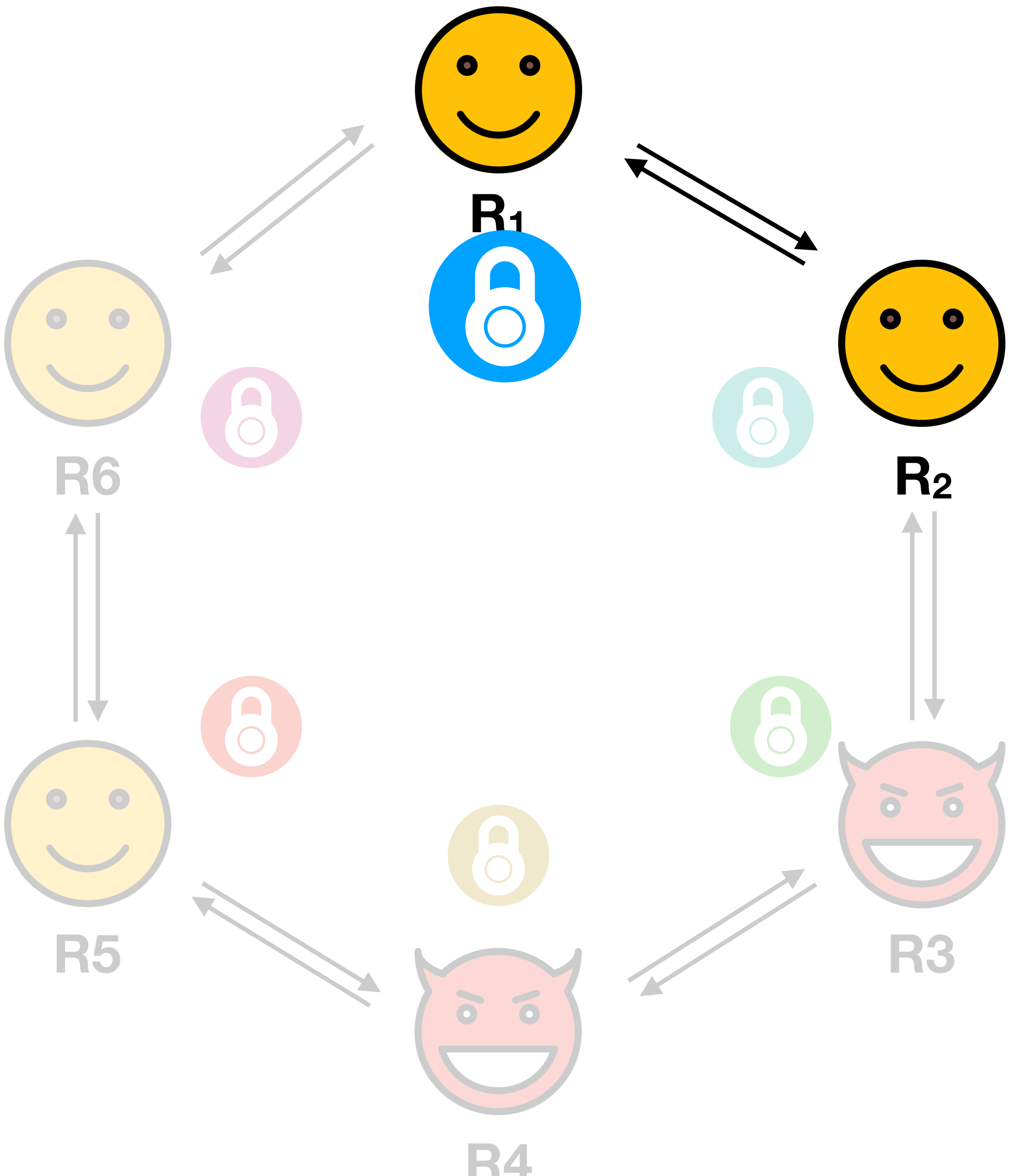


First idea: stack encodings



From [DNS12]:

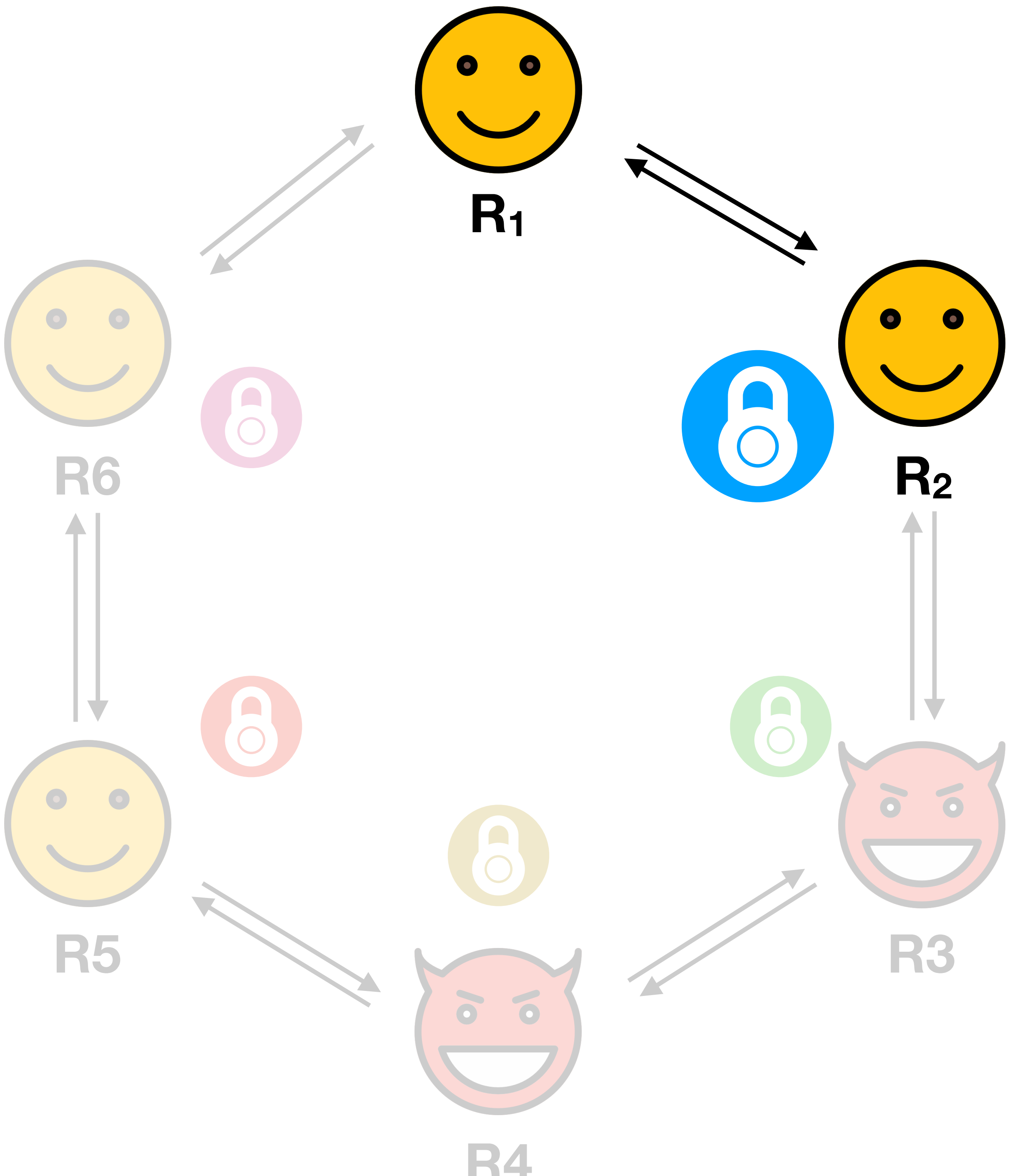
First idea: stack encodings



From [DNS12]:

$$C_1(|\psi\rangle \otimes |0^n\rangle)$$

First idea: stack encodings



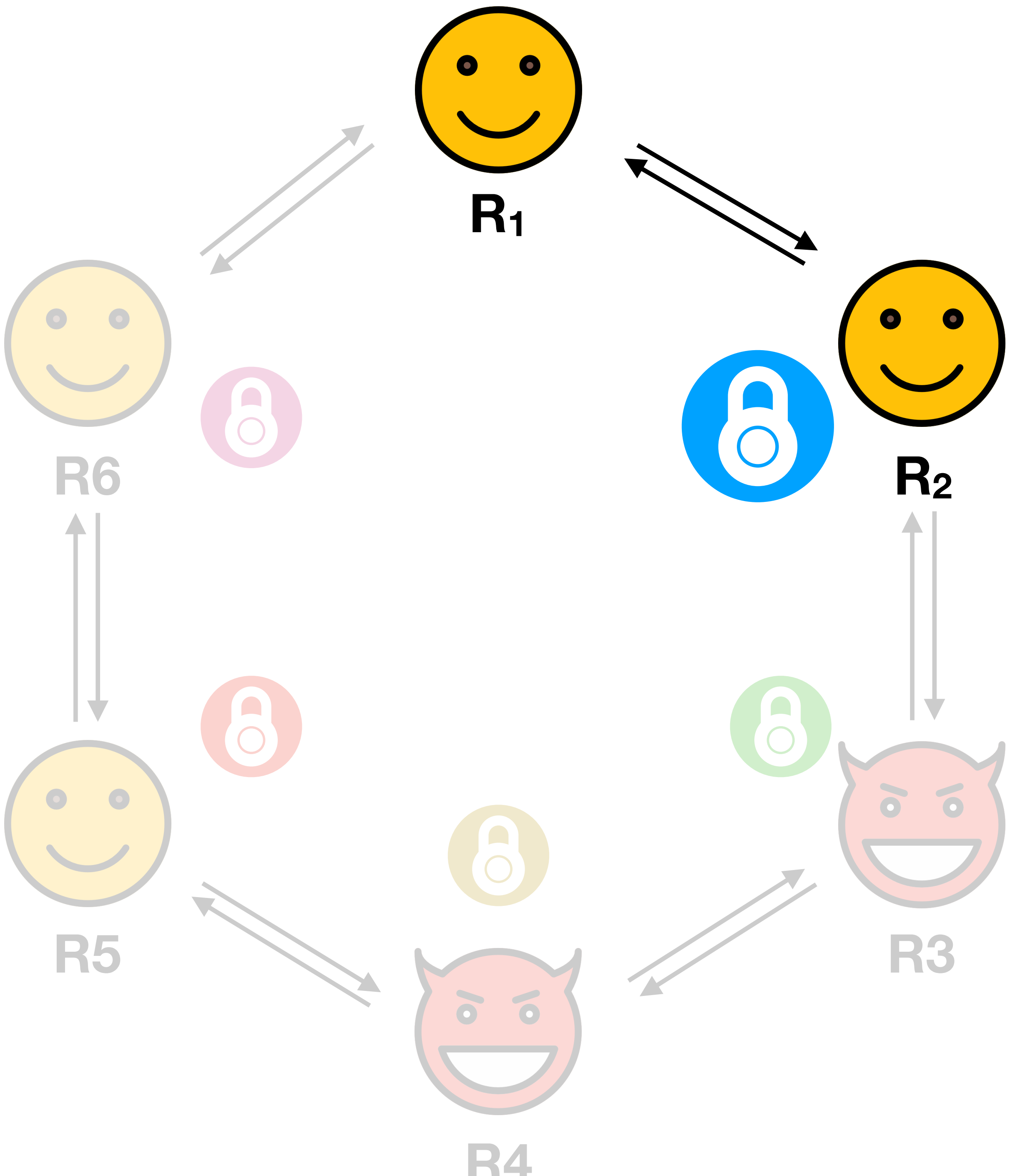
From [DNS12]:

$$C_1(|\psi\rangle \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle)$$

First idea: stack encodings



From [DNS12]:

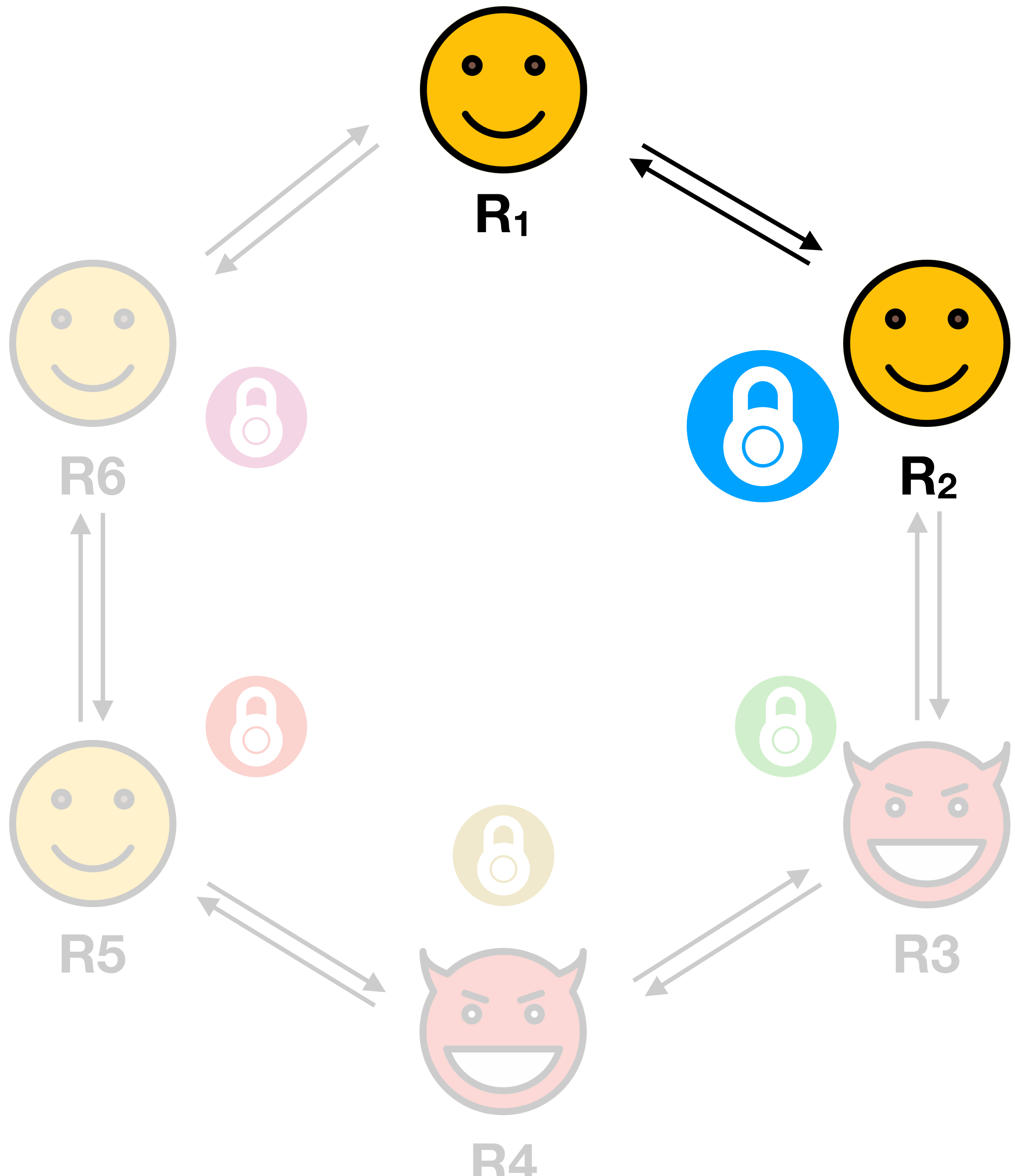
$$C_1(|\psi\rangle \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle)$$

PLAYER 2'S TRAPS ACCESSIBLE

First idea: stack encodings



From [DNS12]:

$$C_1(|\psi\rangle \otimes |0^n\rangle)$$

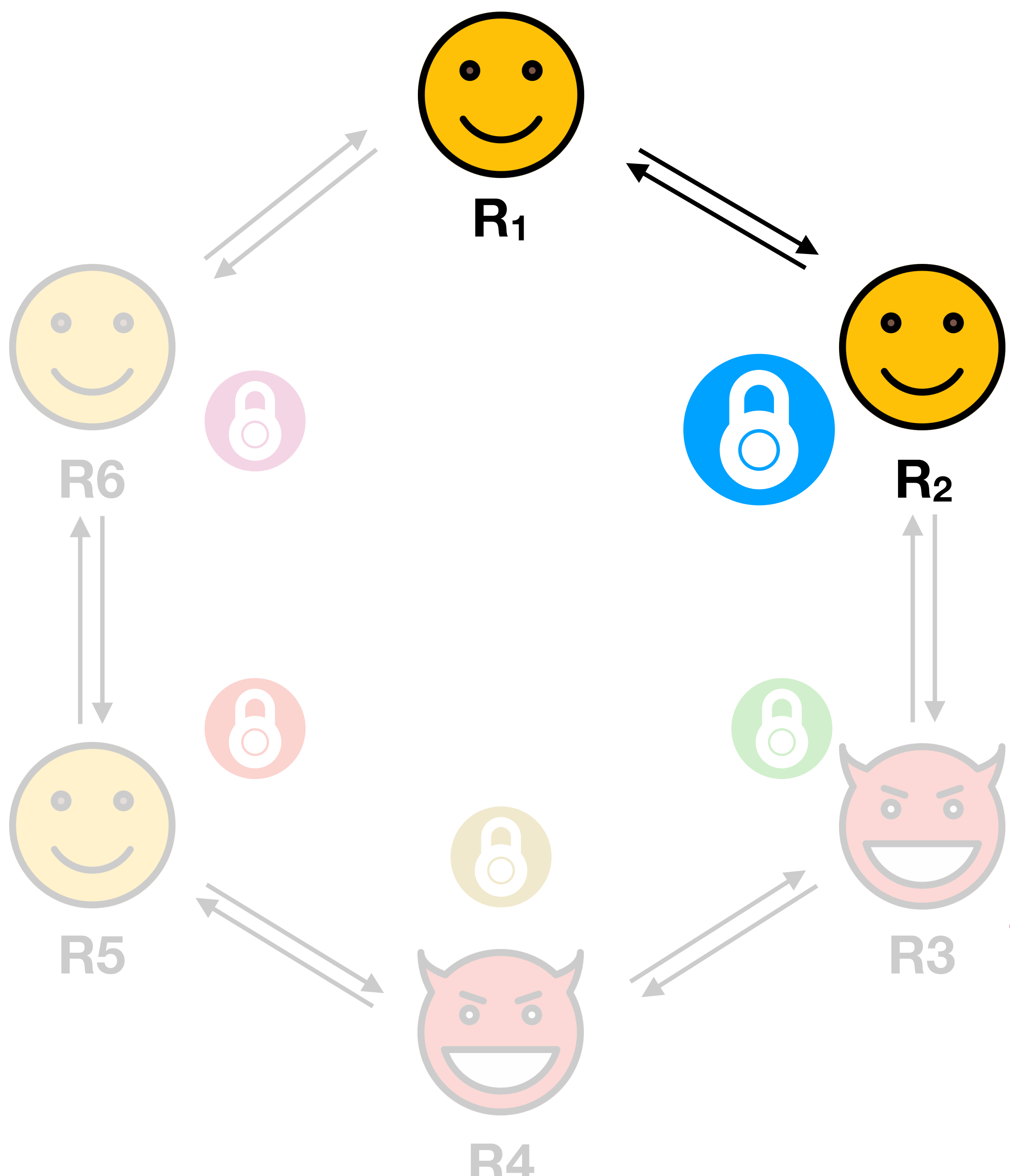
$$\downarrow$$

$$C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle)$$

PLAYER 1'S TRAPS INACCESSIBLE

PLAYER 2'S TRAPS ACCESSIBLE

First idea: stack encodings



From [DNS12]:

$$C_1(|\psi\rangle \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle)$$

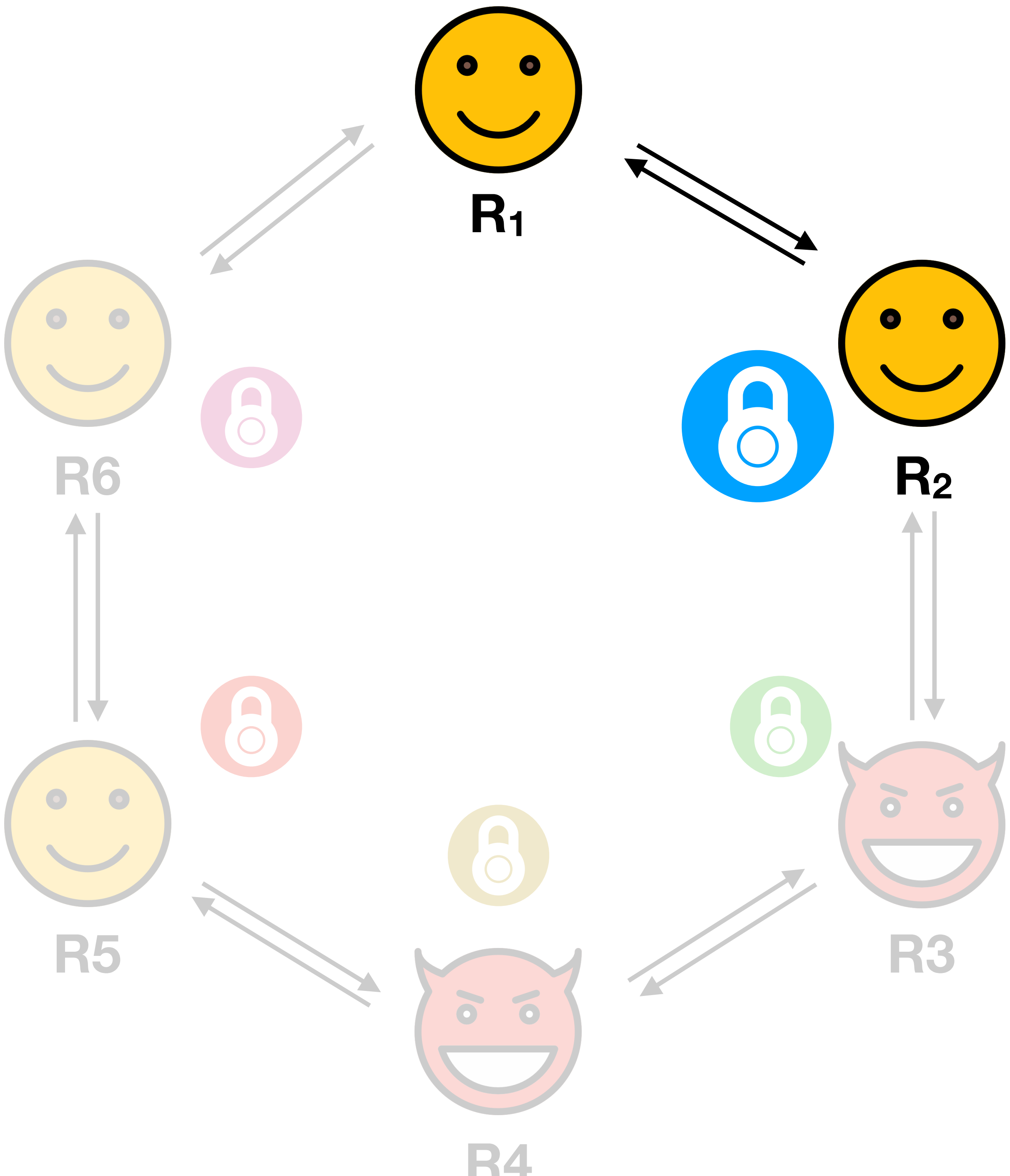
PLAYER 1'S TRAPS INACCESSIBLE

PLAYER 2'S TRAPS ACCESSIBLE

MAKE ACCESSIBLE WITH CLASSICAL MPC:

$$f(C_1, C_2) = (C_2' \otimes I^{\otimes n})(C_1^\dagger \otimes I^{\otimes n})C_2^\dagger$$

First idea: stack encodings



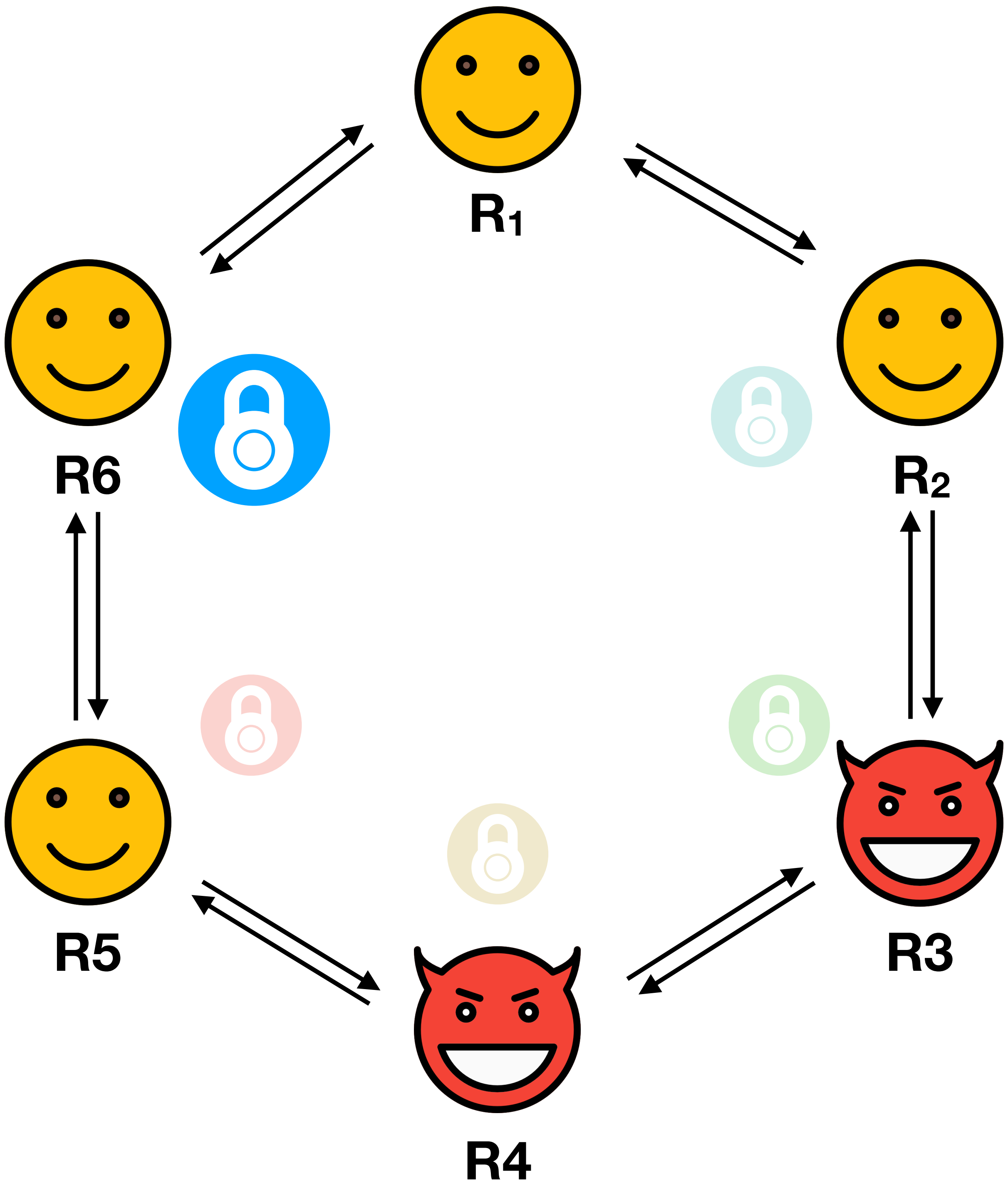
From [DNS12]:

$$C_1(|\psi\rangle \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle)$$

First idea: stack encodings



From [DNS12]:

$$C_1(|\psi\rangle \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_3(C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_4(C_3(C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle)$$

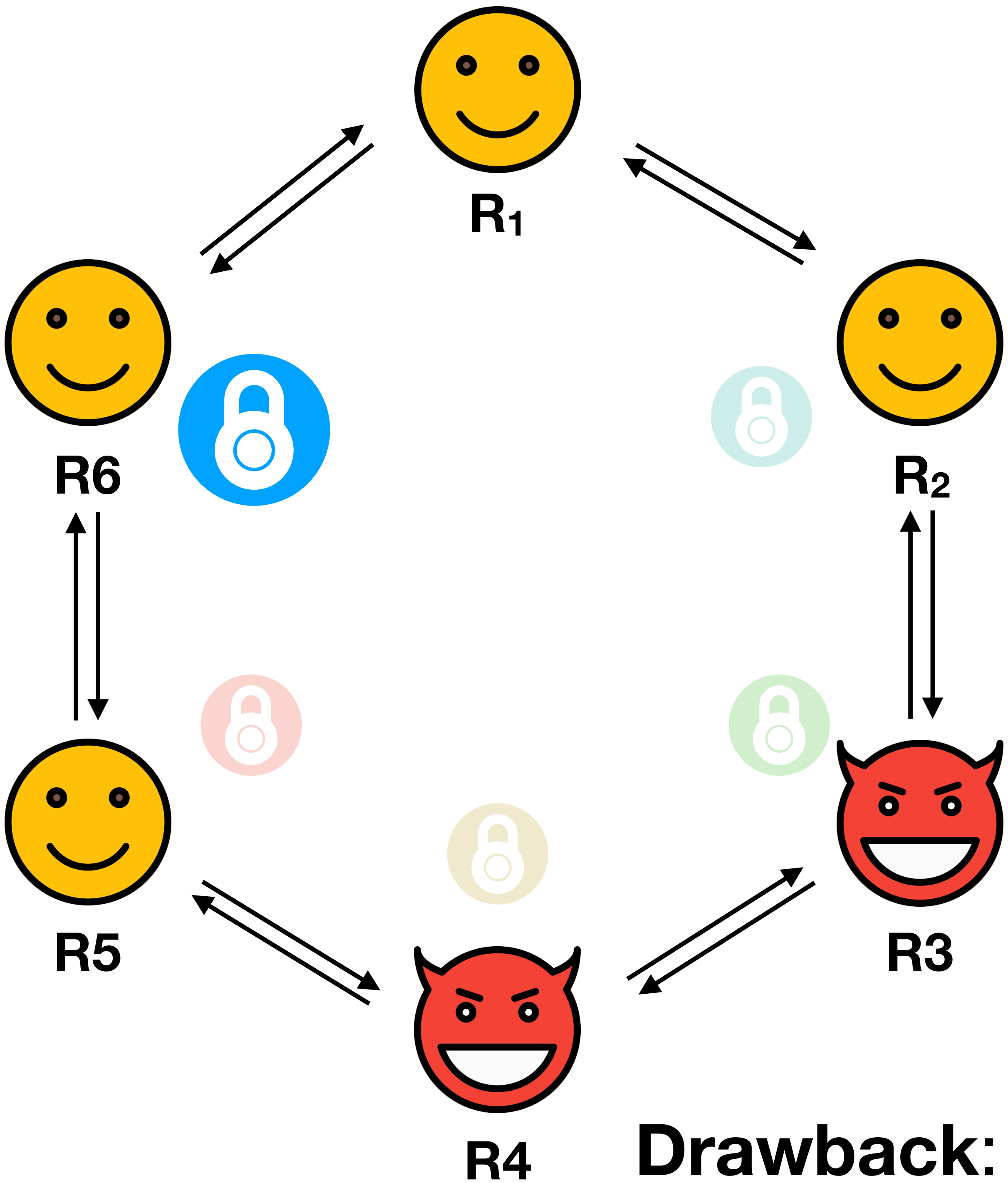
$$\downarrow$$

$$C_5(C_4(C_3(C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_6(C_5(C_4(C_3(C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle)$$

First idea: stack encodings



From [DNS12]:

$$C_1(|\psi\rangle \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_3(C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle)$$

$$\downarrow$$

$$C_4(C_3(C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle)$$

$$\downarrow$$

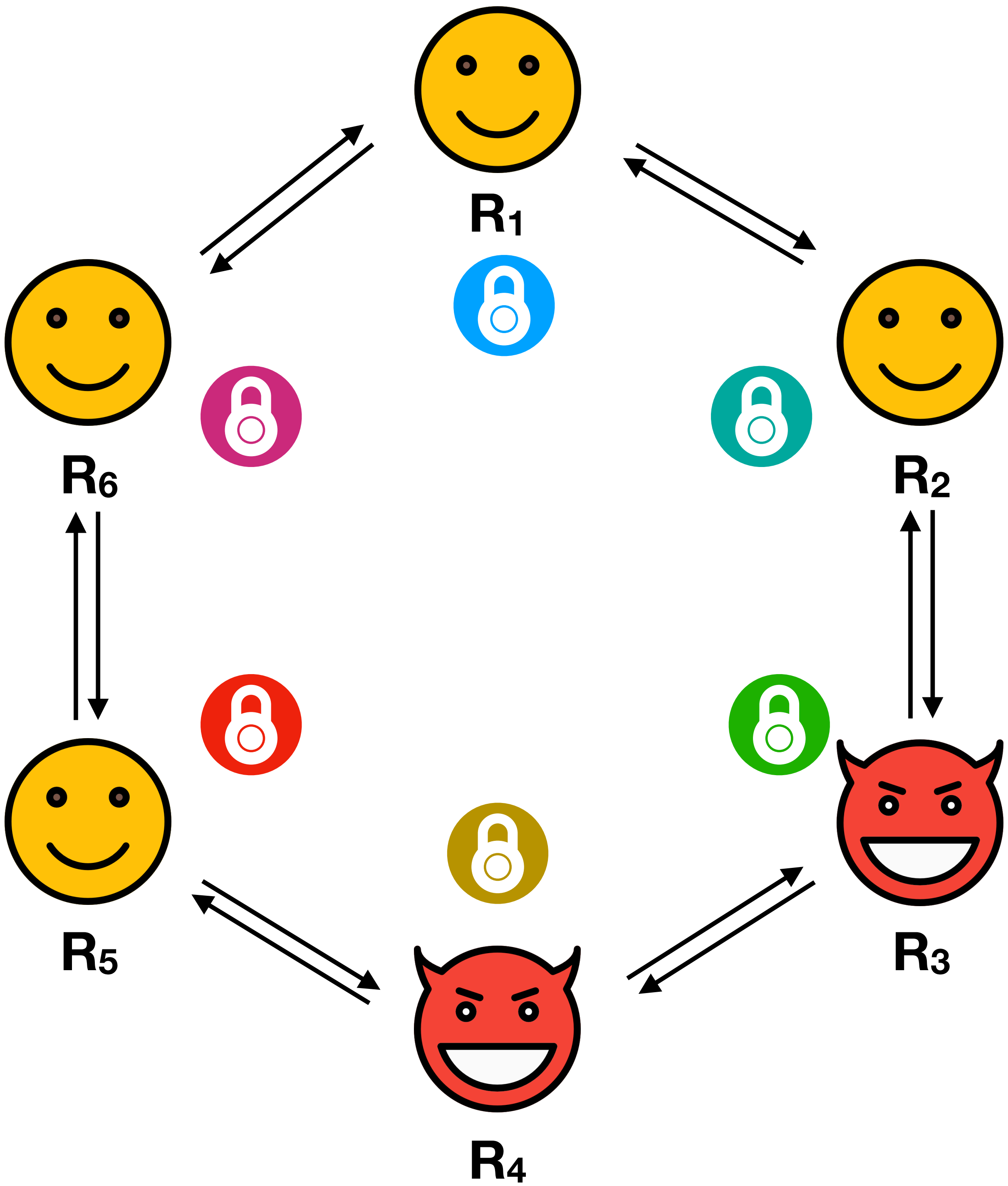
$$C_5(C_4(C_3(C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle)$$

$$\downarrow$$

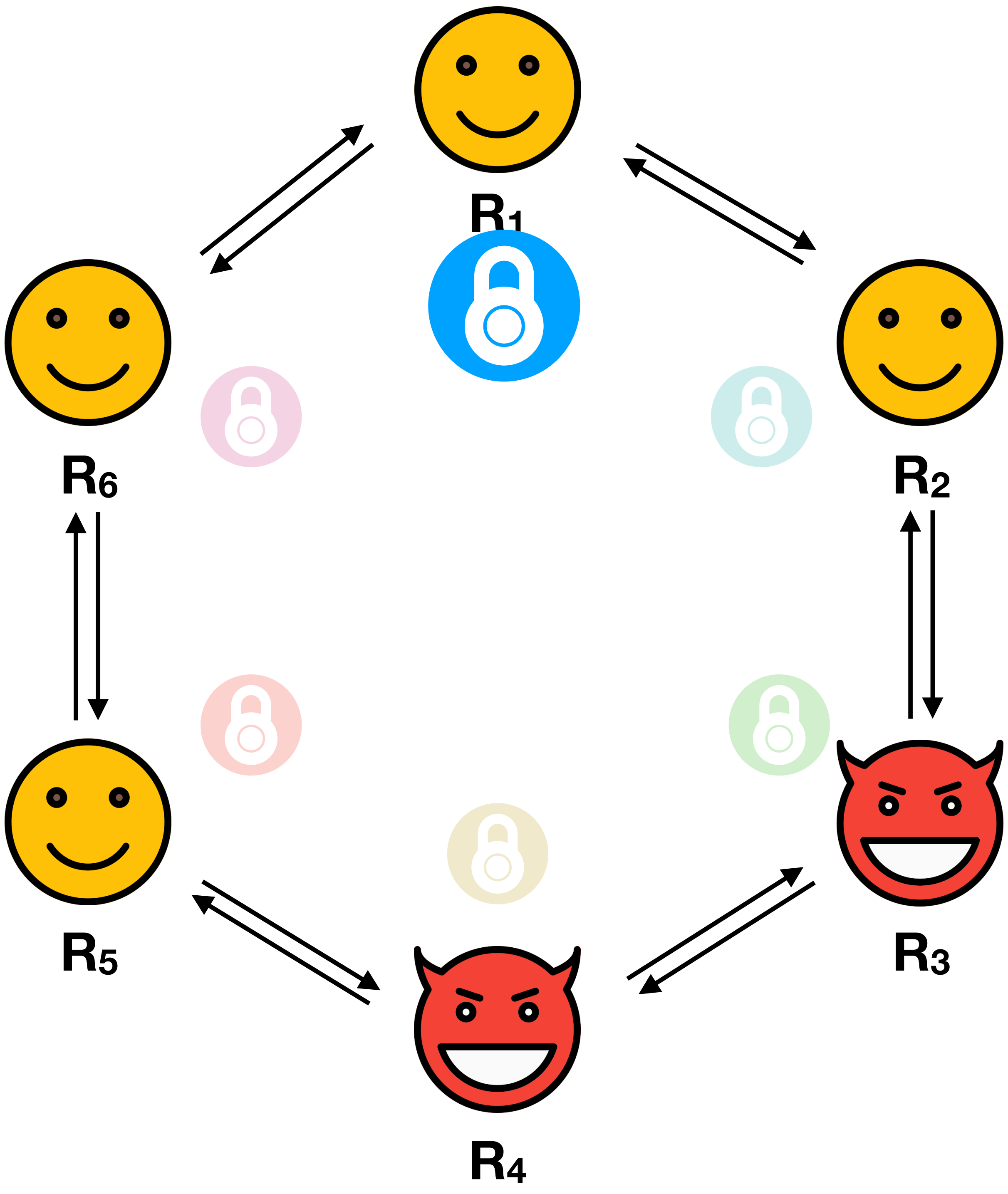
$$C_6(C_5(C_4(C_3(C_2(C_1(|\psi\rangle \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle) \otimes |0^n\rangle)$$

Drawback: very large ciphertexts ($nk + 1$)

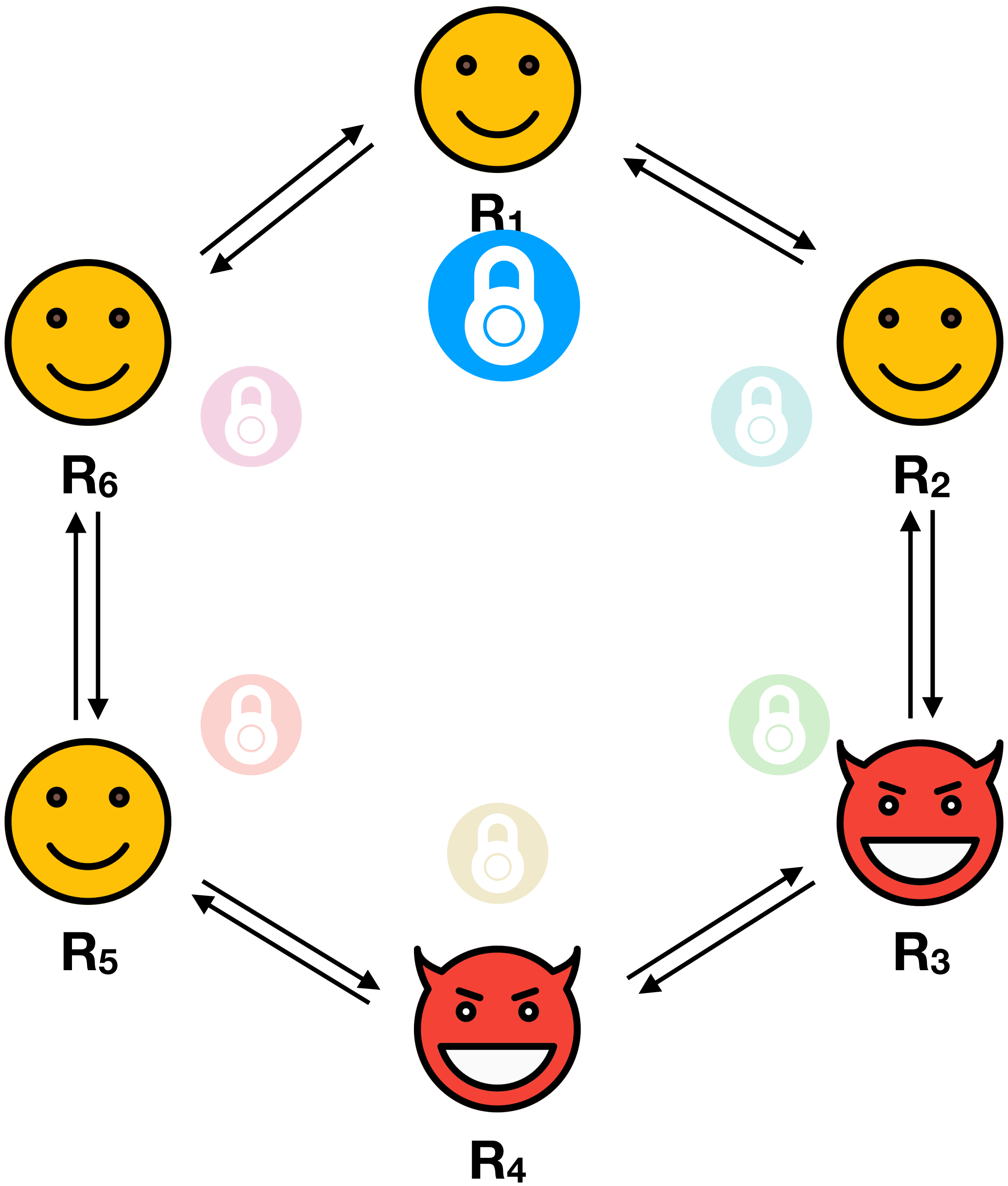
Public authentication test



Public authentication test

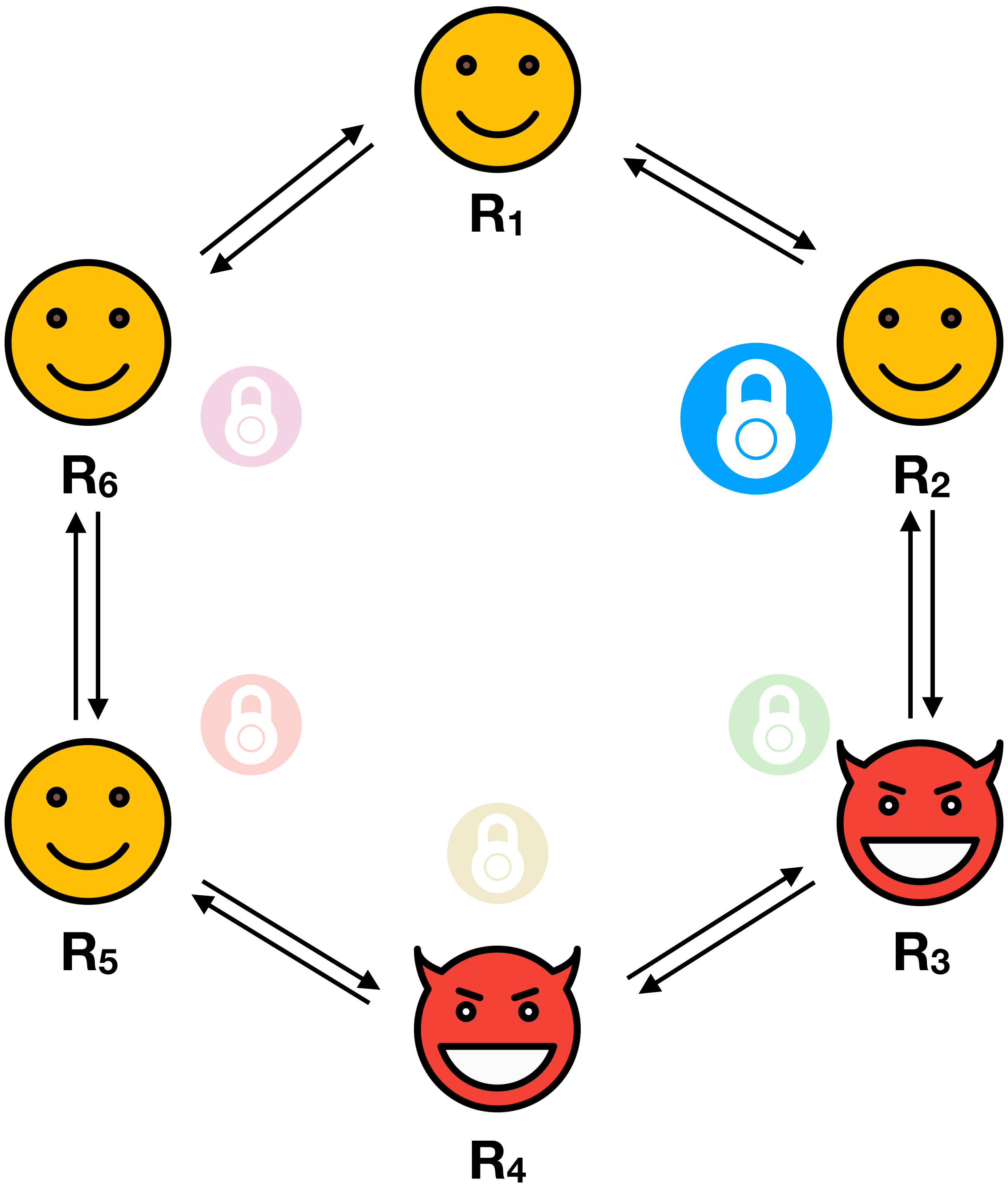


Public authentication test



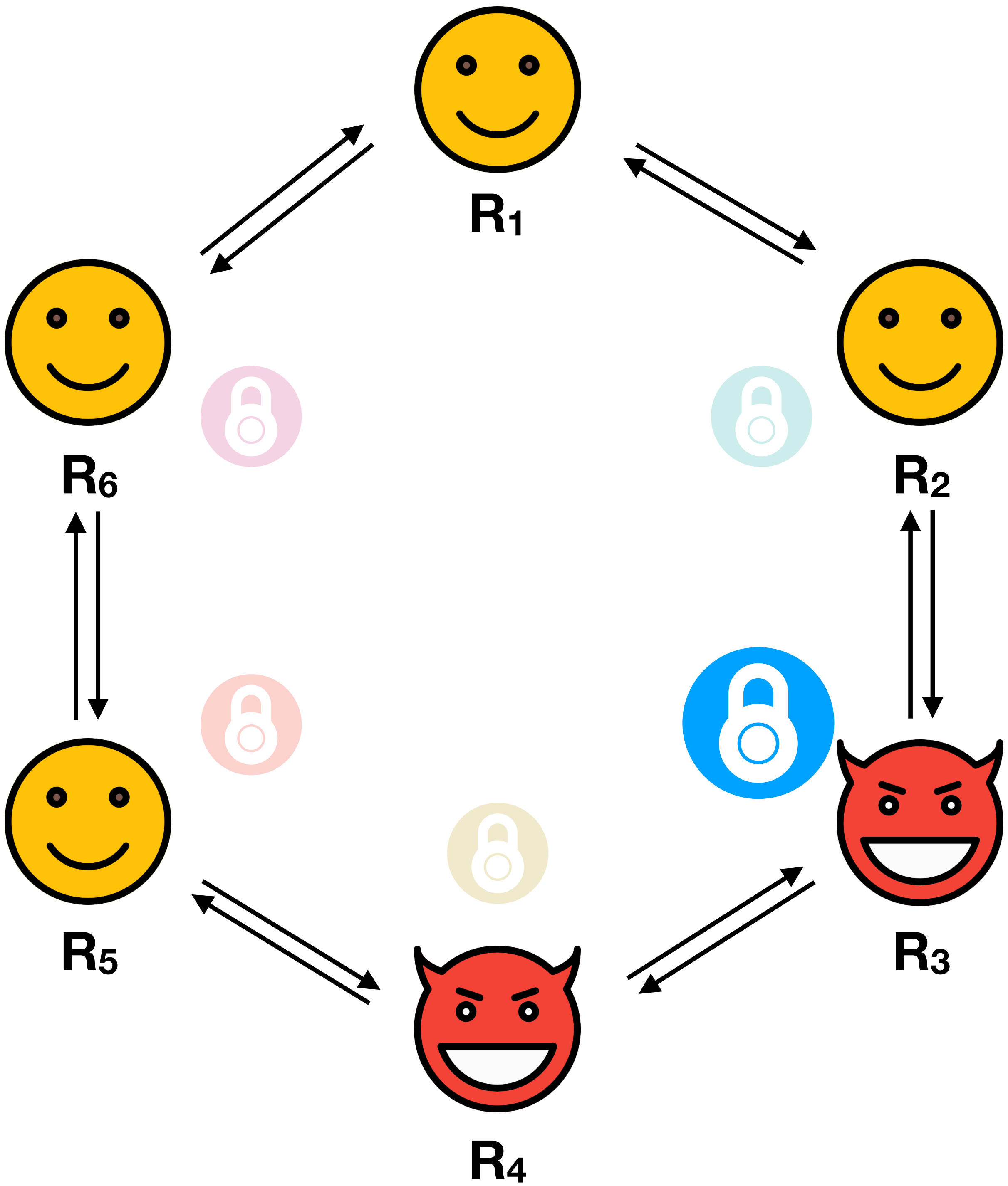
$$C_1(|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test



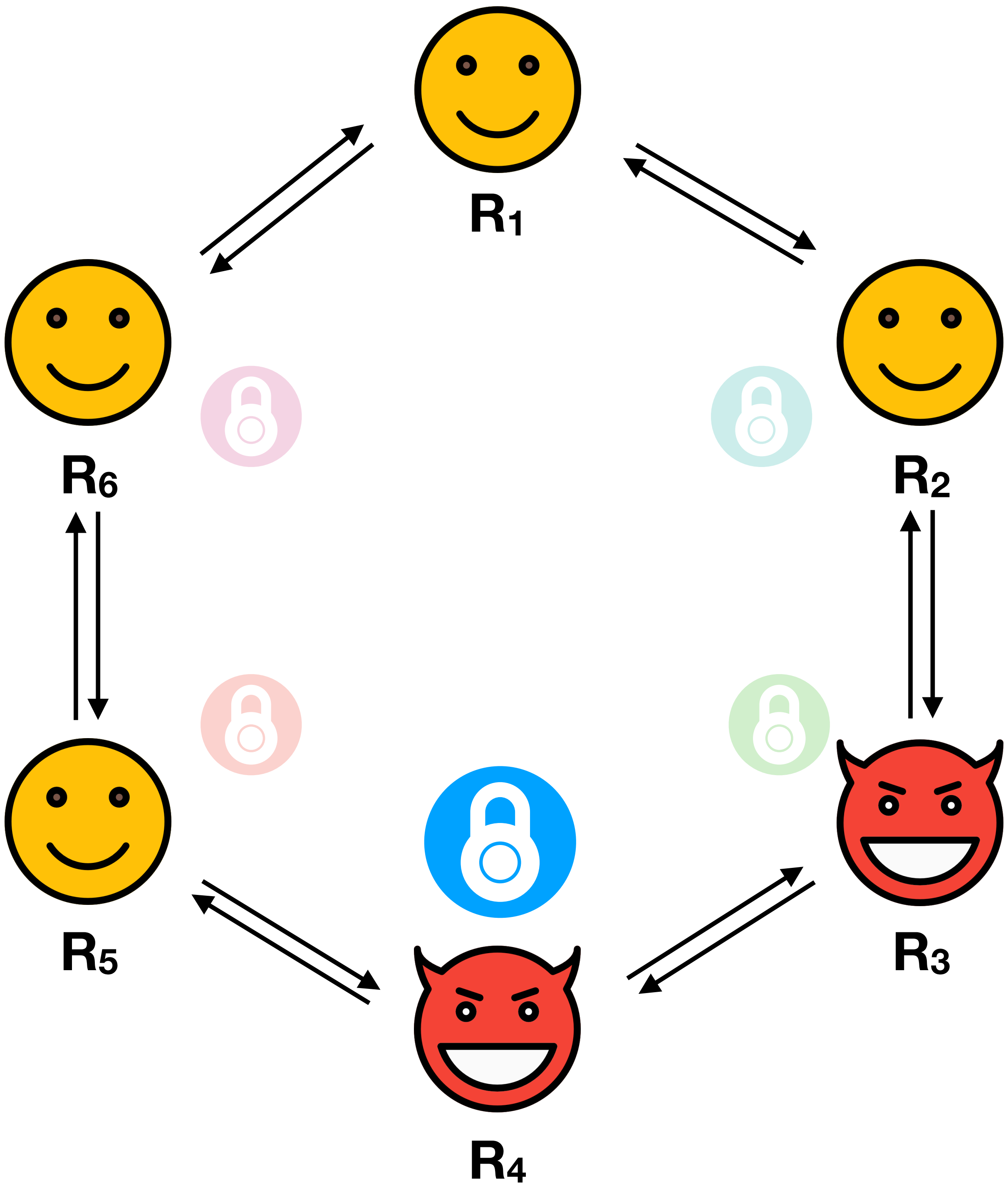
$$C_2 C_1 (|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test



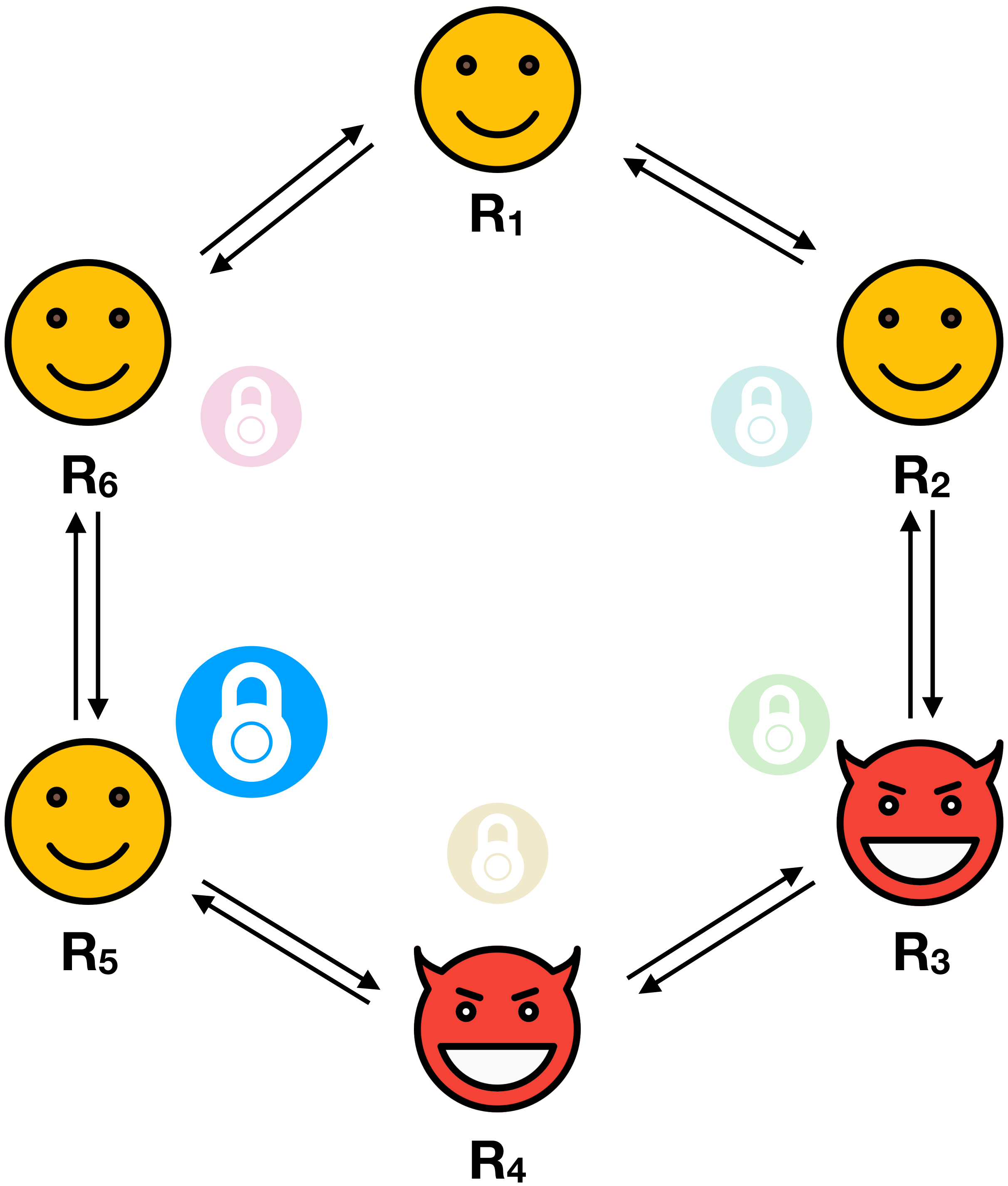
$$C_3 C_2 C_1 (|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test



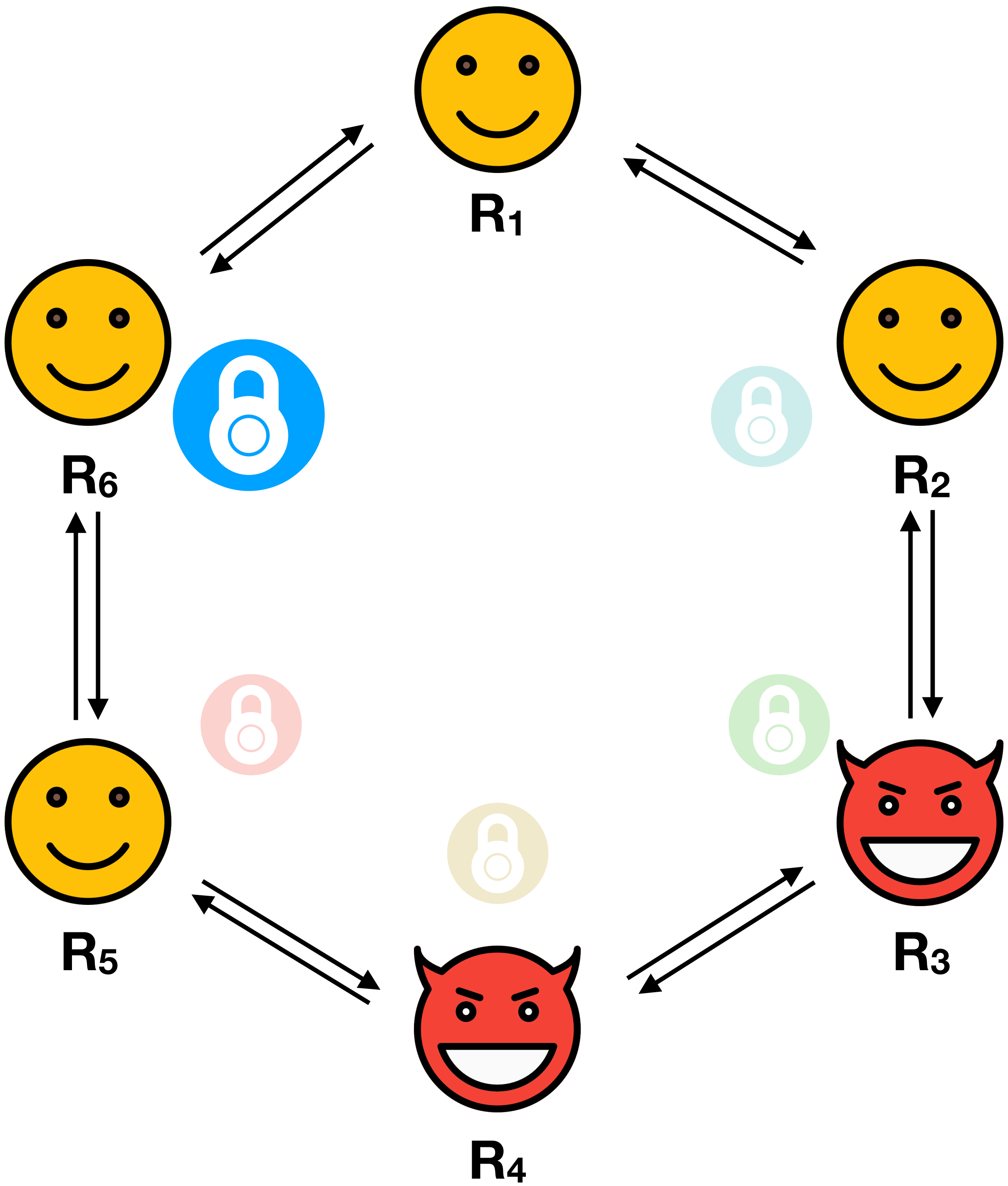
$$C_4C_3C_2C_1(|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test



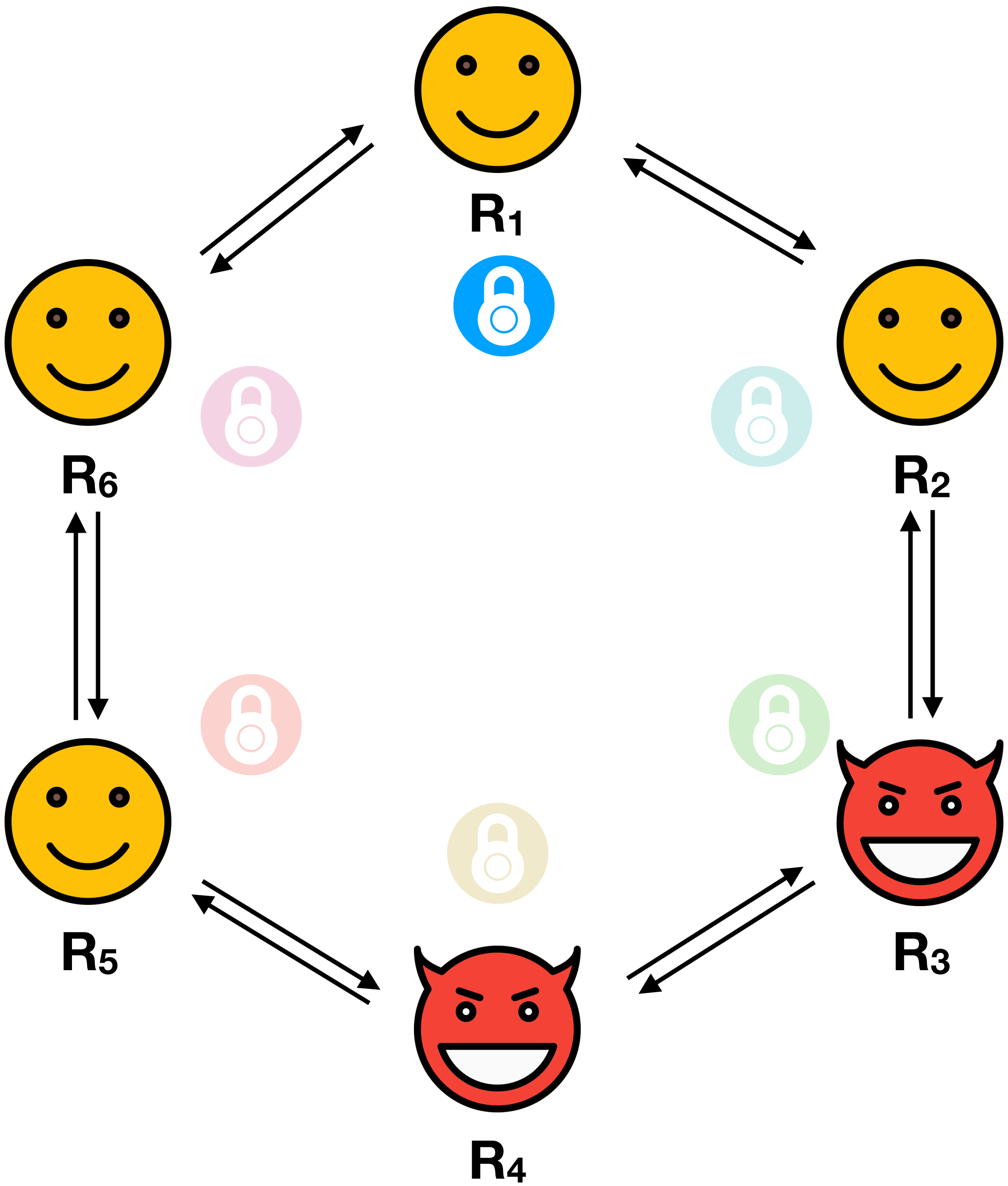
$$C_5C_4C_3C_2C_1(|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test



$$C_6C_5C_4C_3C_2C_1(|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test



$$C_6C_5C_4C_3C_2C_1(|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test

$$C_6 C_5 C_4 C_3 C_2 C_1 (|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test

$$\underbrace{C_6C_5C_4C_3C_2C_1}_{C}(|\psi\rangle \otimes |0^{2n}\rangle)$$

Public authentication test

$$\underbrace{C_6C_5C_4C_3C_2C_1}_{C}(|\psi\rangle \otimes |0^{2n}\rangle)$$

UNKNOWN TO ALL

Public authentication test

$$\underbrace{C_6 C_5 C_4 C_3 C_2 C_1}_{C}(|\psi\rangle \otimes |0^{2n}\rangle)$$

PLAYER I CREATED THESE

UNKNOWN TO ALL

Public authentication test

$$\underbrace{C_6C_5C_4C_3C_2C_1}_{C}(|\psi\rangle \otimes |0^{2n}\rangle)$$

PLAYER I CREATED THESE

UNKNOWN TO ALL

Using classical MPC: 

Public authentication test

$$\underbrace{C_6 C_5 C_4 C_3 C_2 C_1}_{C}(|\psi\rangle \otimes |0^{2n}\rangle)$$

PLAYER I CREATED THESE

UNKNOWN TO ALL

Using classical MPC: 

- Select $g \in_R GL(2n, \mathbb{F}_2)$. Note: $g(y) = 0^{2n}$ iff $y = 0^{2n}$

  Lemma: apply random g and measure n traps

  $\approx$ measure $2n$ traps

Public authentication test

$$\underbrace{C_6 C_5 C_4 C_3 C_2 C_1}_{C}(|\psi\rangle \otimes |0^{2n}\rangle)$$

PLAYER I CREATED THESE

UNKNOWN TO ALL

Using classical MPC: 😇

- Select $g \in_R GL(2n, \mathbb{F}_2)$. Note: $g(y) = 0^{2n}$ iff $y = 0^{2n}$
  Lemma: apply random g and measure n traps
  $\approx$ measure $2n$ traps
- Let player 1 apply $(C' \otimes X^r)(I \otimes g)C^\dagger$ for random C', r

Public authentication test

$$\underbrace{C_6C_5C_4C_3C_2C_1}_{C}(|\psi\rangle \otimes |0^{2n}\rangle)$$

PLAYER I CREATED THESE

UNKNOWN TO ALL

Using classical MPC: 😇

- Select $g \in_R GL(2n, \mathbb{F}_2)$. Note: $g(y) = 0^{2n}$ iff $y = 0^{2n}$
  Lemma: apply random g and measure n traps
  $\approx$ measure $2n$ traps
- Let player 1 apply $(C' \otimes X^r)(I \otimes g)C^\dagger$ for random C', r
- Let player 1 measure last n qubits (check if outcome is r)

Public authentication test

$$\underbrace{C_6C_5C_4C_3C_2C_1}_{C}(|\psi\rangle \otimes |0^{2n}\rangle)$$

UNKNOWN TO ALL (C)

PLAYER 1 CREATED THESE ($|0^{2n}\rangle$)

Using classical MPC: 😇

- Select $g \in_R GL(2n, \mathbb{F}_2)$. Note: $g(y) = 0^{2n}$ iff $y = 0^{2n}$
  Lemma: apply random g and measure n traps
  $\approx$ measure $2n$ traps
- Let player 1 apply $(C' \otimes X^r)(I \otimes g)C^\dagger$ for random C', r
- Let player 1 measure last n qubits (check if outcome is r)

Result: authenticated state $C'(|\psi\rangle \otimes |0^n\rangle)$

Public authentication test

Public authentication test

One player **performs** the test: applies Clifford, measures, …

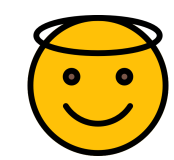
Public authentication test

One player **performs** the test: applies Clifford, measures, …

All players **verify** the test through classical MPC 

Public authentication test

One player **performs** the test: applies Clifford, measures, …

All players **verify** the test through classical MPC 

The test can be used:

- to test encodings (as in previous slide);

Public authentication test

One player **performs** the test: applies Clifford, measures, …

All players **verify** the test through classical MPC 😇

The test can be used:

- to test encodings (as in previous slide);
- to test whether a computation step was executed honestly

Introduction
Authentication
Computation
Magic-state generation
Summary

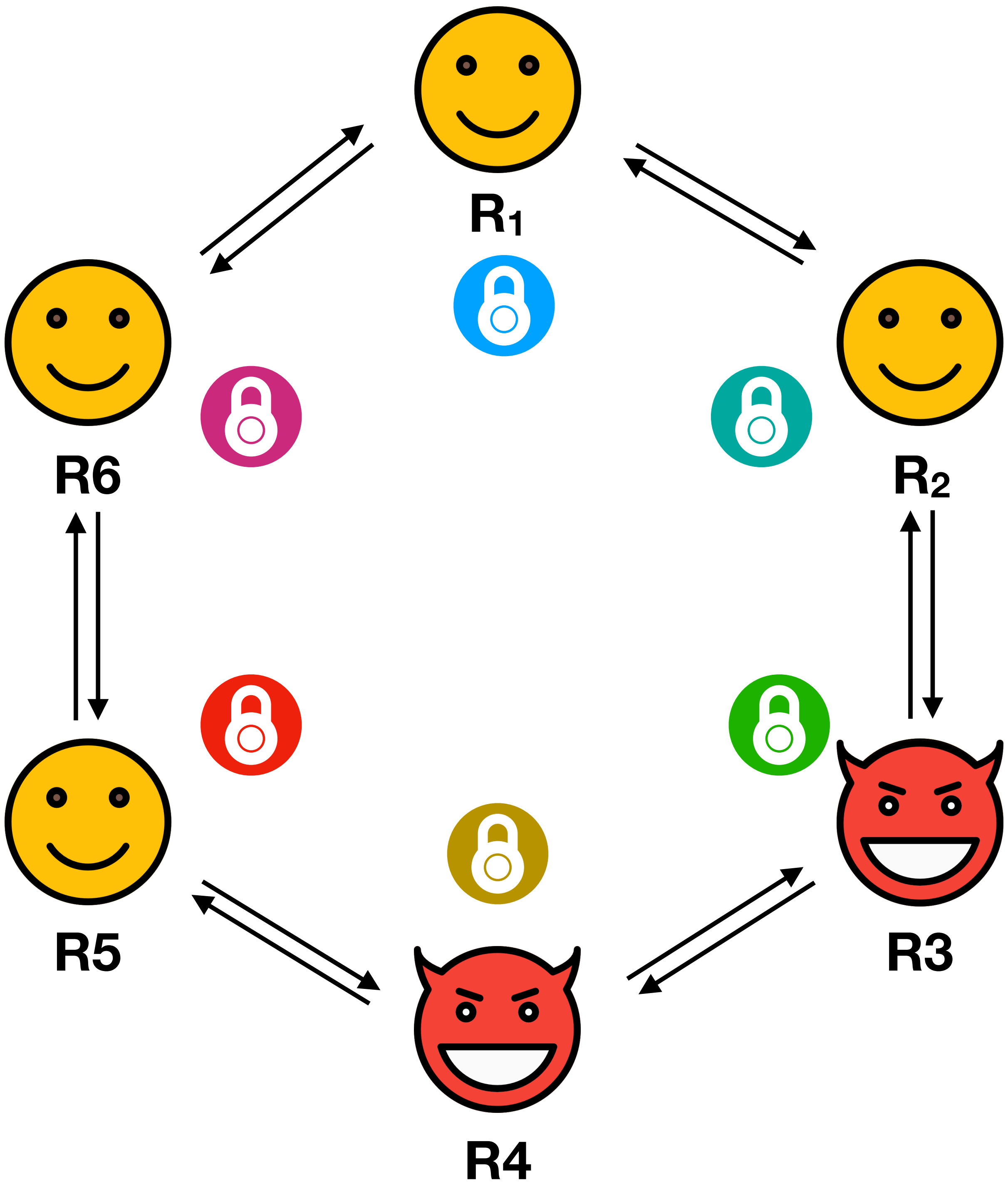
Computation

$|\psi\rangle$ $G|\psi\rangle$

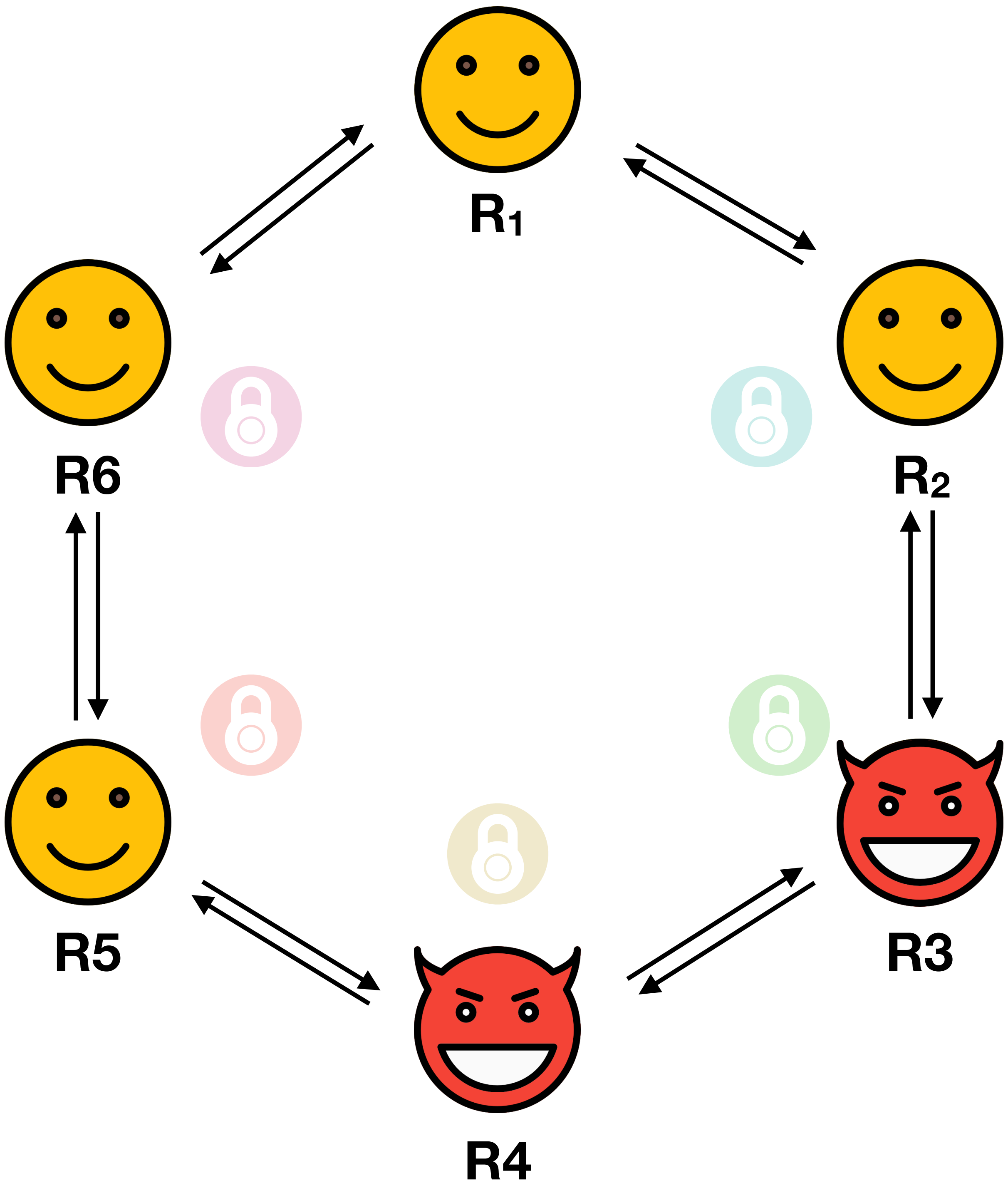
Protocols ($C(|\psi\rangle \otimes |0^n\rangle) \mapsto C'(G|\psi\rangle \otimes |0^n\rangle)$) for these G :

- 1-qubit Cliffords
- CNOT (2-qubit Clifford)
- T (non-Clifford)
- Computational-basis measurement

Single-qubit Cliffords

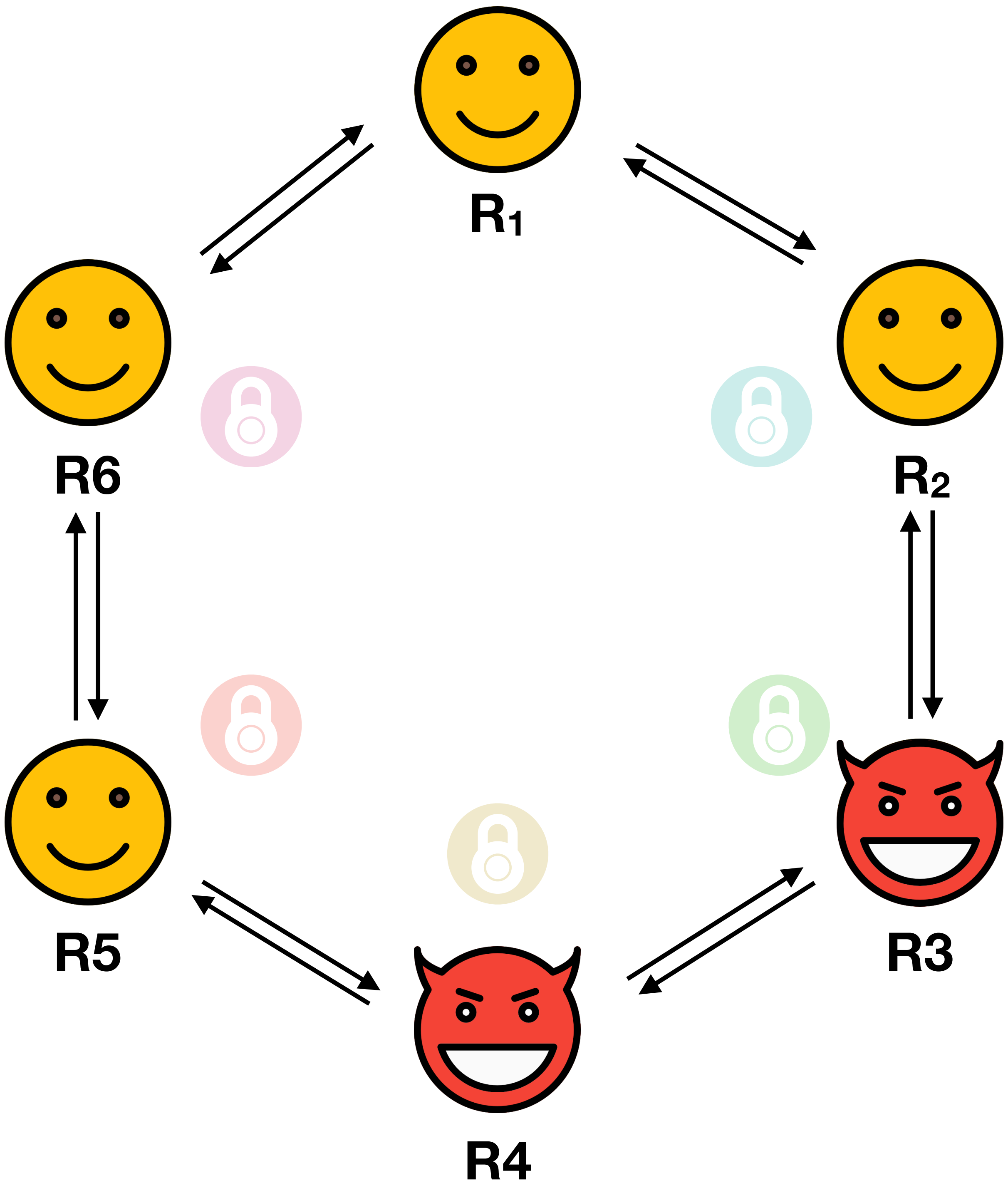


Single-qubit Cliffords



$$= C(|\psi\rangle \otimes |0\rangle^{\otimes n})$$

Single-qubit Cliffords

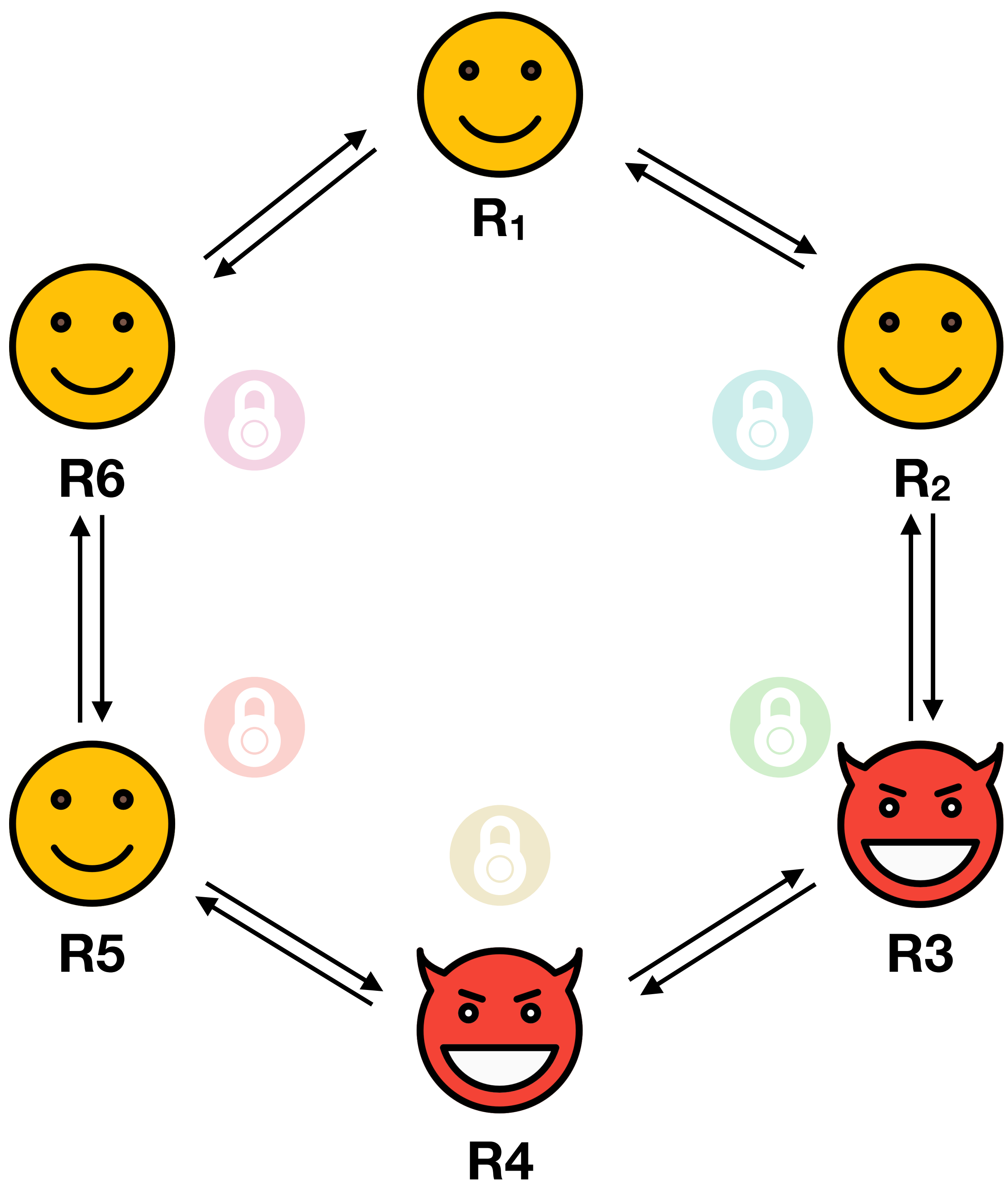


 $= C(|\psi\rangle \otimes |0\rangle^{\otimes n})$

Using classical MPC: update classical key 

$$C \mapsto C' := C(G^\dagger \otimes I^{\otimes n})$$

Single-qubit Cliffords



$$= C(|\psi\rangle \otimes |0\rangle^{\otimes n})$$

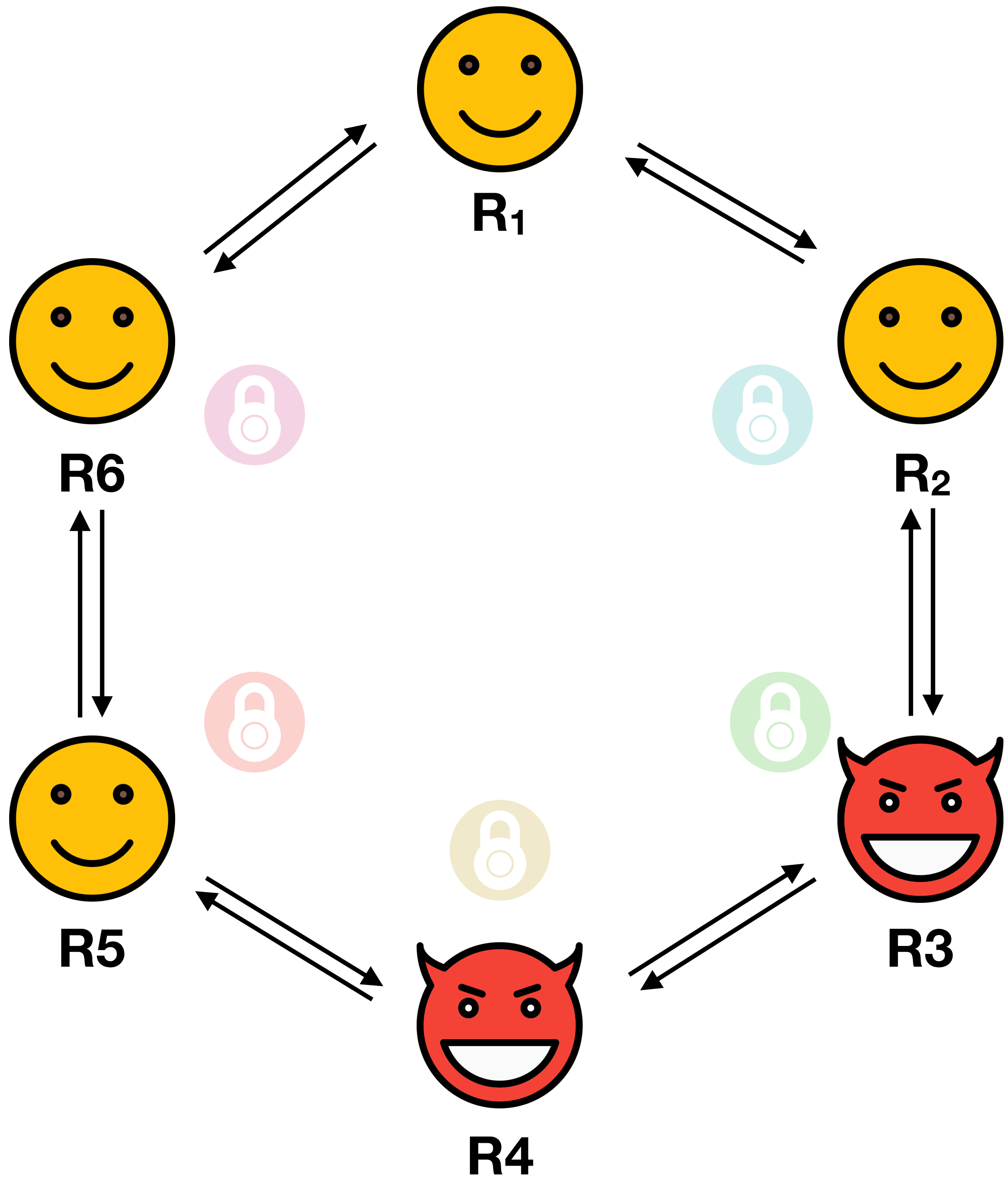
Using classical MPC:
update classical key

$$C \mapsto C' := C(G^\dagger \otimes I^{\otimes n})$$

Then will decode to

$$(C')^\dagger C(|\psi\rangle \otimes |0\rangle^{\otimes n})$$

Single-qubit Cliffords



 $= C(|\psi\rangle \otimes |0\rangle^{\otimes n})$

Using classical MPC:
update classical key

$$C \mapsto C' := C(G^\dagger \otimes I^{\otimes n})$$

Then  will decode to

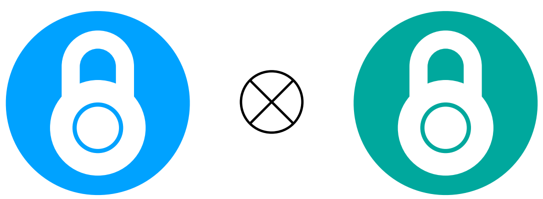
$$(C')^\dagger C(|\psi\rangle \otimes |0\rangle^{\otimes n})$$

$$= G|\psi\rangle \otimes |0\rangle^{\otimes n}$$

CNOT

CNOT



CNOT

 $\otimes$  $= (C_1 \otimes C_2)(|\psi_1\rangle \otimes |0^n\rangle \otimes |\psi_2\rangle \otimes |0^n\rangle)$

CNOT

 $\otimes$  $= (C_1 \otimes C_2)(|\psi_1\rangle \otimes |0^n\rangle \otimes |\psi_2\rangle \otimes |0^n\rangle)$

Same strategy does not work:
$(C_1 \otimes C_2)(CNOT^\dagger \otimes I^{\otimes 2n})$ is not in product form.

CNOT

 $\otimes$  $= (C_1 \otimes C_2)(|\psi_1\rangle \otimes |0^n\rangle \otimes |\psi_2\rangle \otimes |0^n\rangle)$

Same strategy does not work:
$(C_1 \otimes C_2)(CNOT^\dagger \otimes I^{\otimes 2n})$ is not in product form.

Instead:

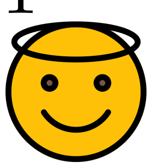
CNOT

 $\otimes$  $= (C_1 \otimes C_2)(|\psi_1\rangle \otimes |0^n\rangle \otimes |\psi_2\rangle \otimes |0^n\rangle)$

Same strategy does not work:
$(C_1 \otimes C_2)(CNOT^\dagger \otimes I^{\otimes 2n})$ is not in product form.

Instead:

- Player 1 applies $(C_1' \otimes C_2')CNOT(C_1^\dagger \otimes C_2^\dagger)$ for freshly random C_1', C_2' .



CNOT

 $\otimes$  $= (C_1 \otimes C_2)(|\psi_1\rangle \otimes |0^n\rangle \otimes |\psi_2\rangle \otimes |0^n\rangle)$

Same strategy does not work:
$(C_1 \otimes C_2)(CNOT^\dagger \otimes I^{\otimes 2n})$ is not in product form.

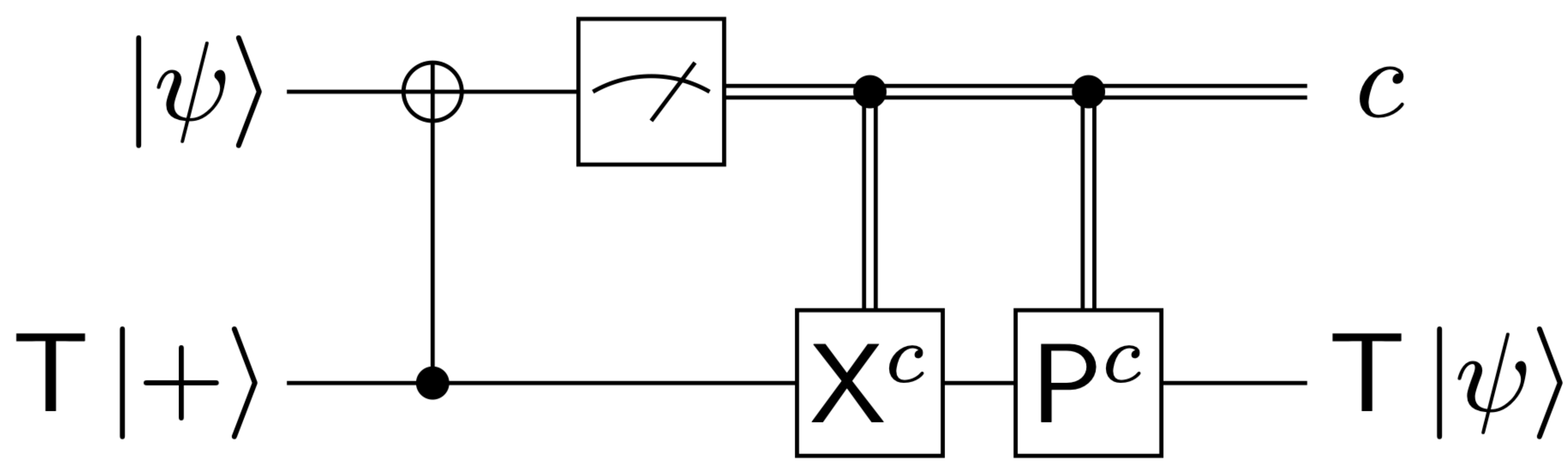
Instead:

- Player 1 applies $(C_1' \otimes C_2')CNOT(C_1^\dagger \otimes C_2^\dagger)$ for freshly random C_1', C_2' . 😇
- Player 1 executes public authentication test. 😇

Non-Clifford gate $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{bmatrix}$

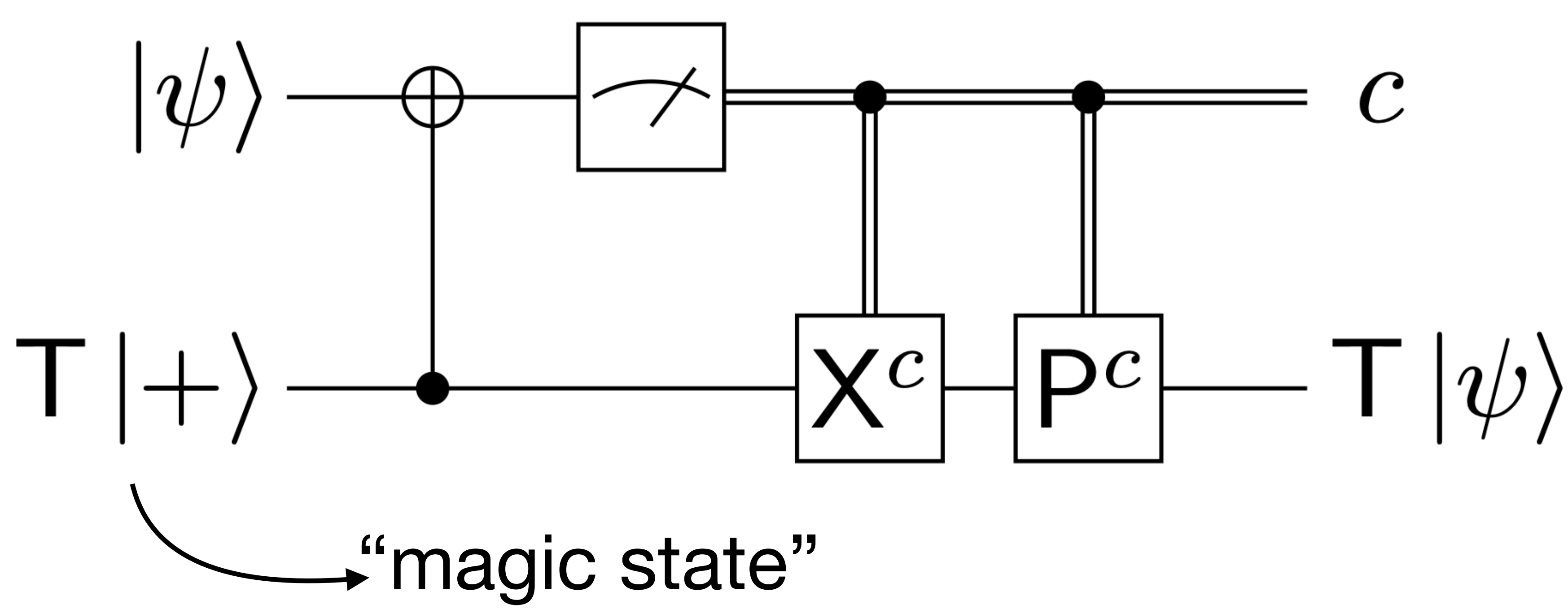
Non-Clifford gate $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{bmatrix}$

Magic-state computation:



Non-Clifford gate $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{bmatrix}$

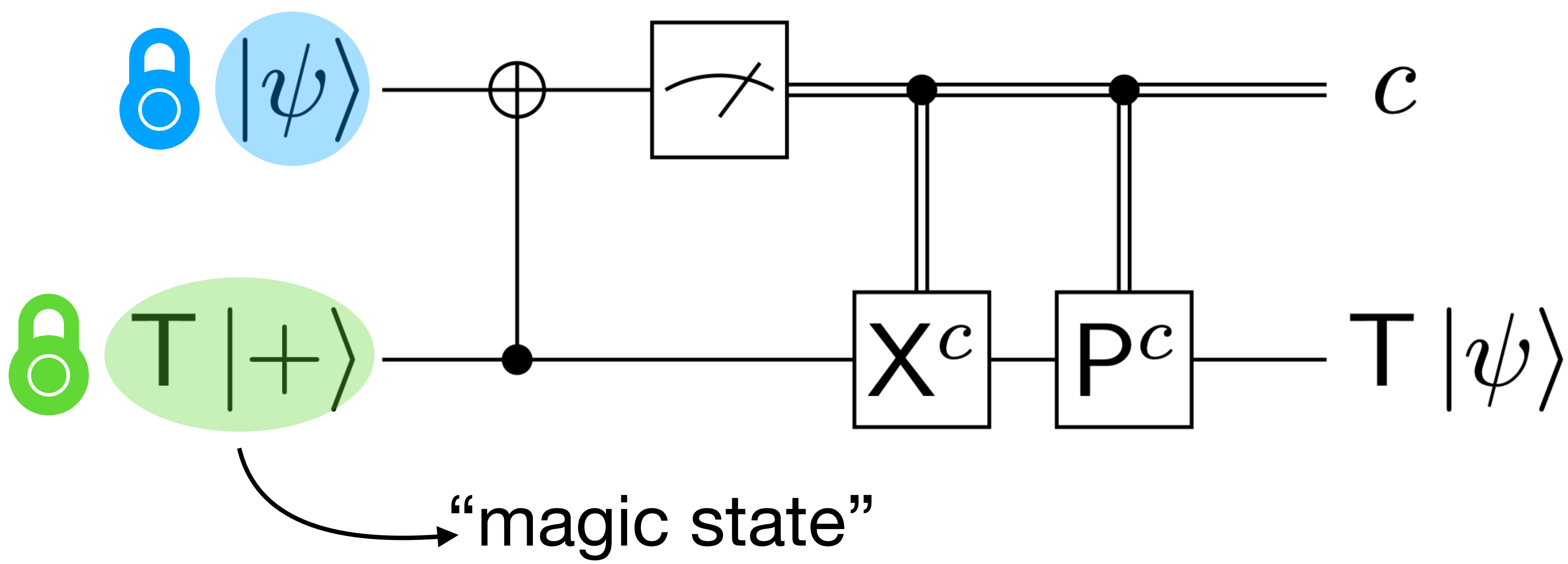
Magic-state computation:



Non-Clifford gate

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{bmatrix}$$

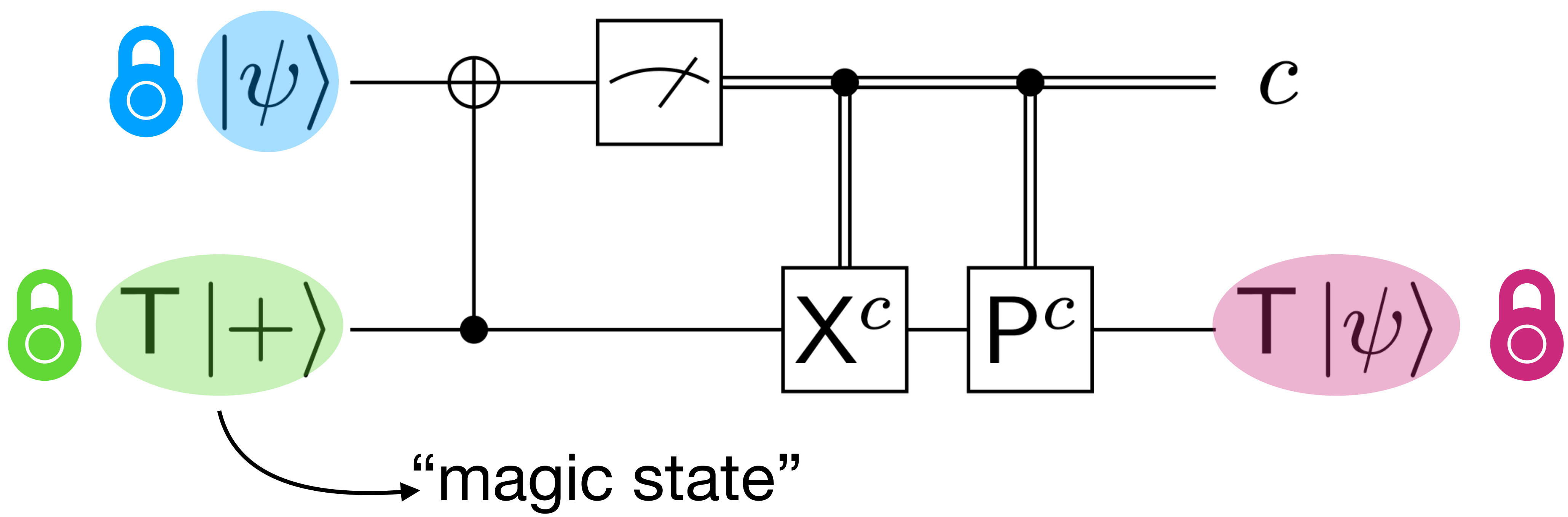
Magic-state computation:



$$C_1(|\psi\rangle \otimes |0^n\rangle) \otimes C_2(T|+\rangle \otimes |0^n\rangle)$$

Non-Clifford gate $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{bmatrix}$

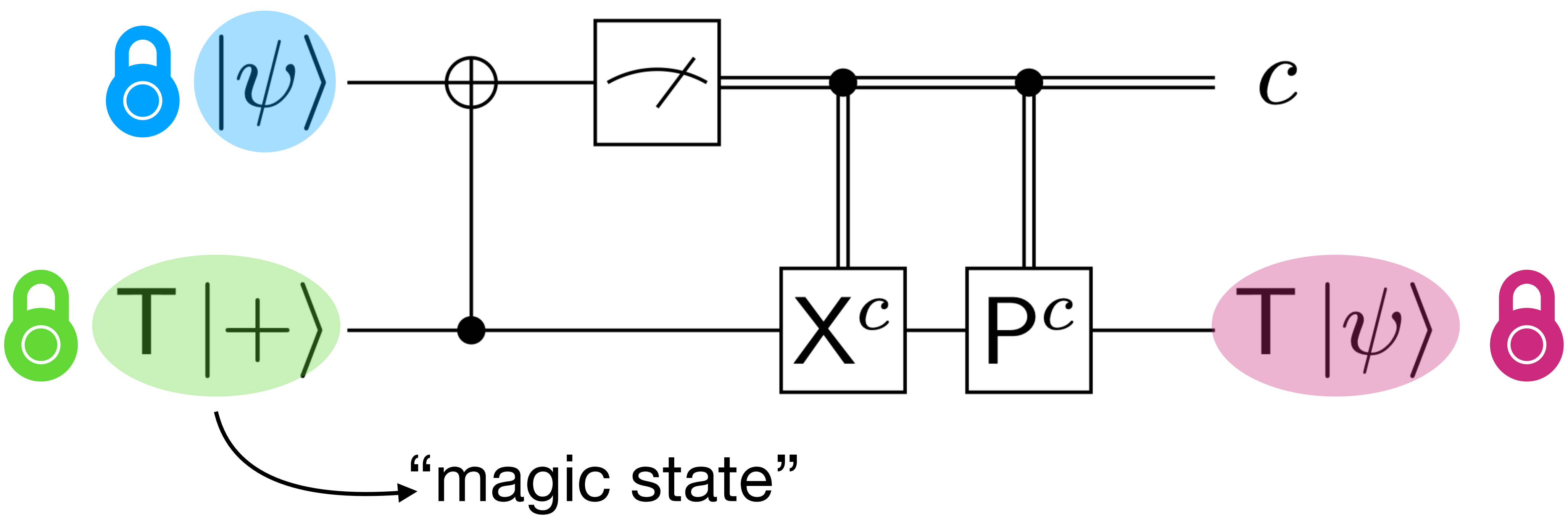
Magic-state computation:



$$C_1(|\psi\rangle \otimes |0^n\rangle) \otimes C_2(T|+\rangle \otimes |0^n\rangle) \mapsto C_3(T|\psi\rangle \otimes |0^n\rangle)$$

Non-Clifford gate $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{bmatrix}$

Magic-state computation:



$$C_1(|\psi\rangle \otimes |0^n\rangle) \otimes C_2(T|+\rangle \otimes |0^n\rangle) \mapsto C_3(T|\psi\rangle \otimes |0^n\rangle)$$

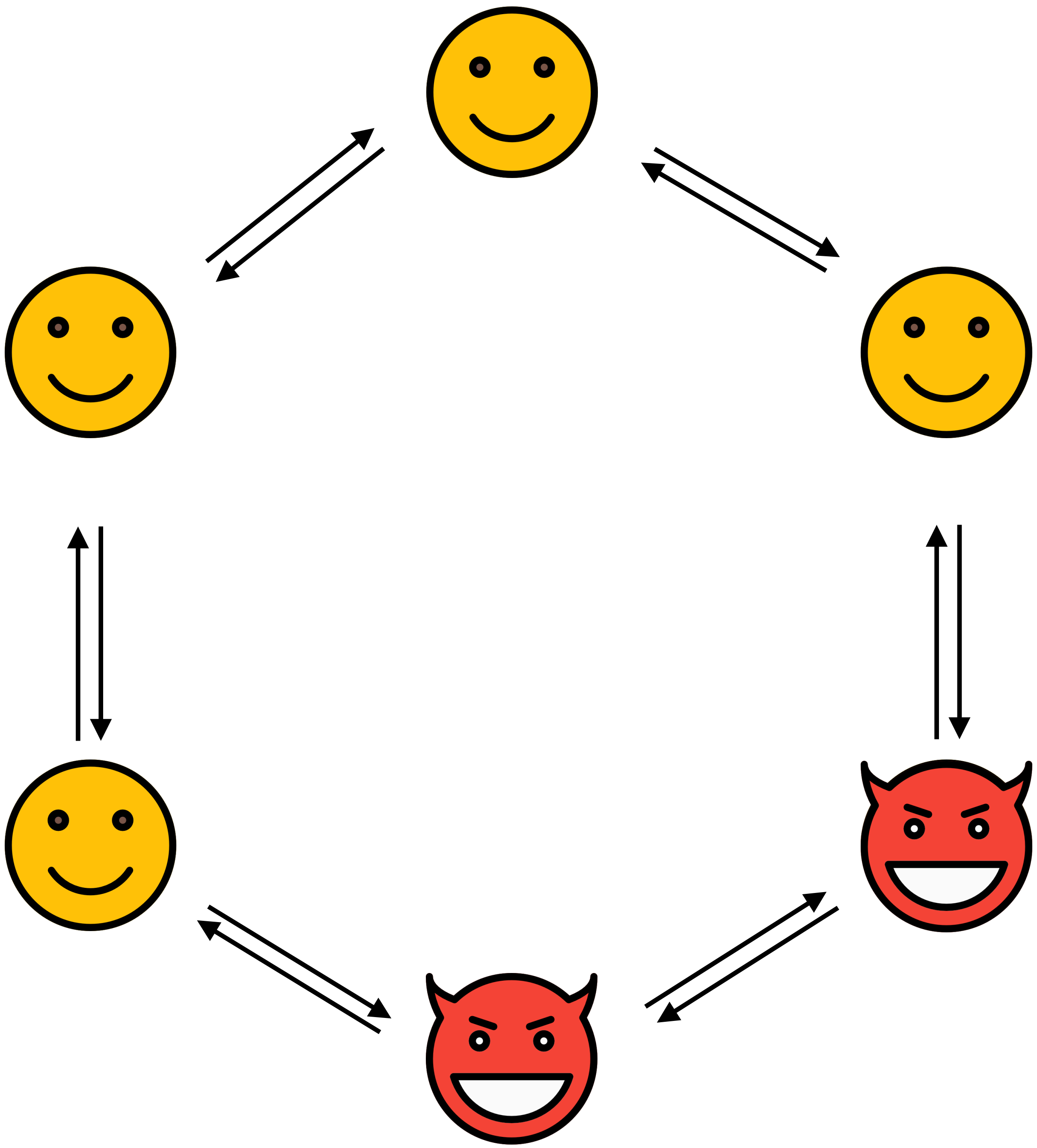
Nobody can be trusted to create encoded magic states!

Introduction
Authentication
Computation

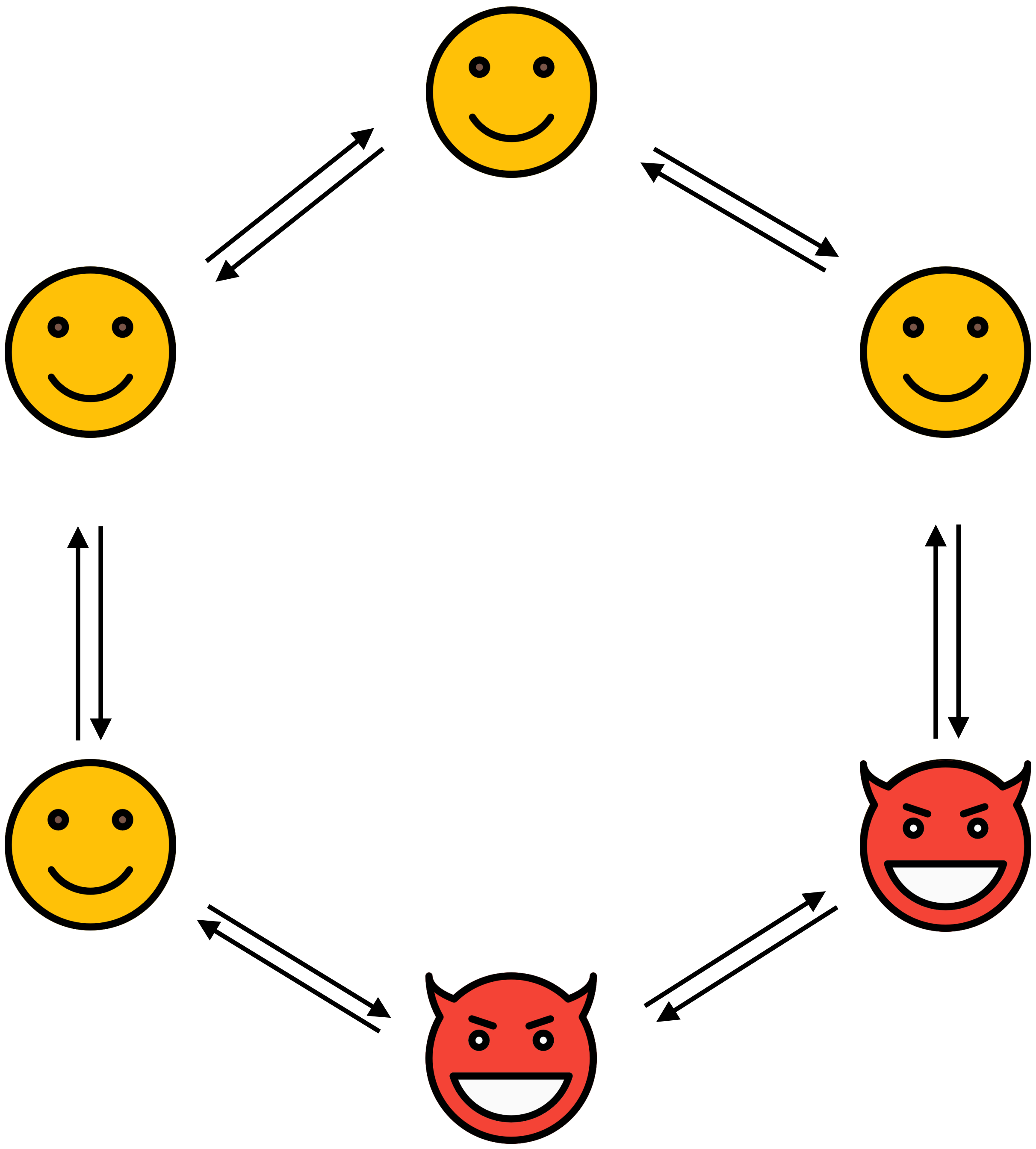
Magic-state generation

Summary

Magic-state generation

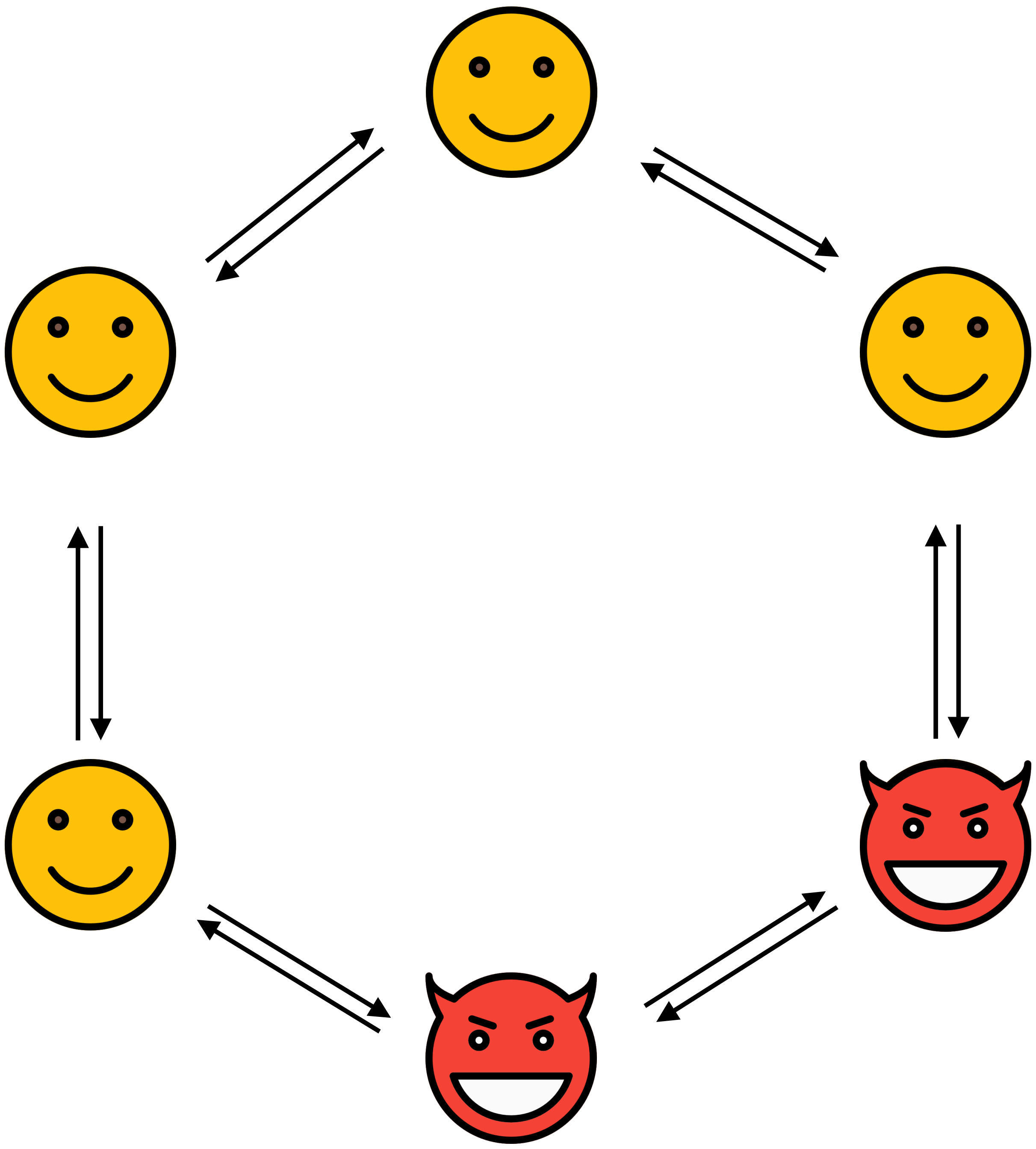


Magic-state generation



 $= C(T|+\rangle \otimes |0^n\rangle)$

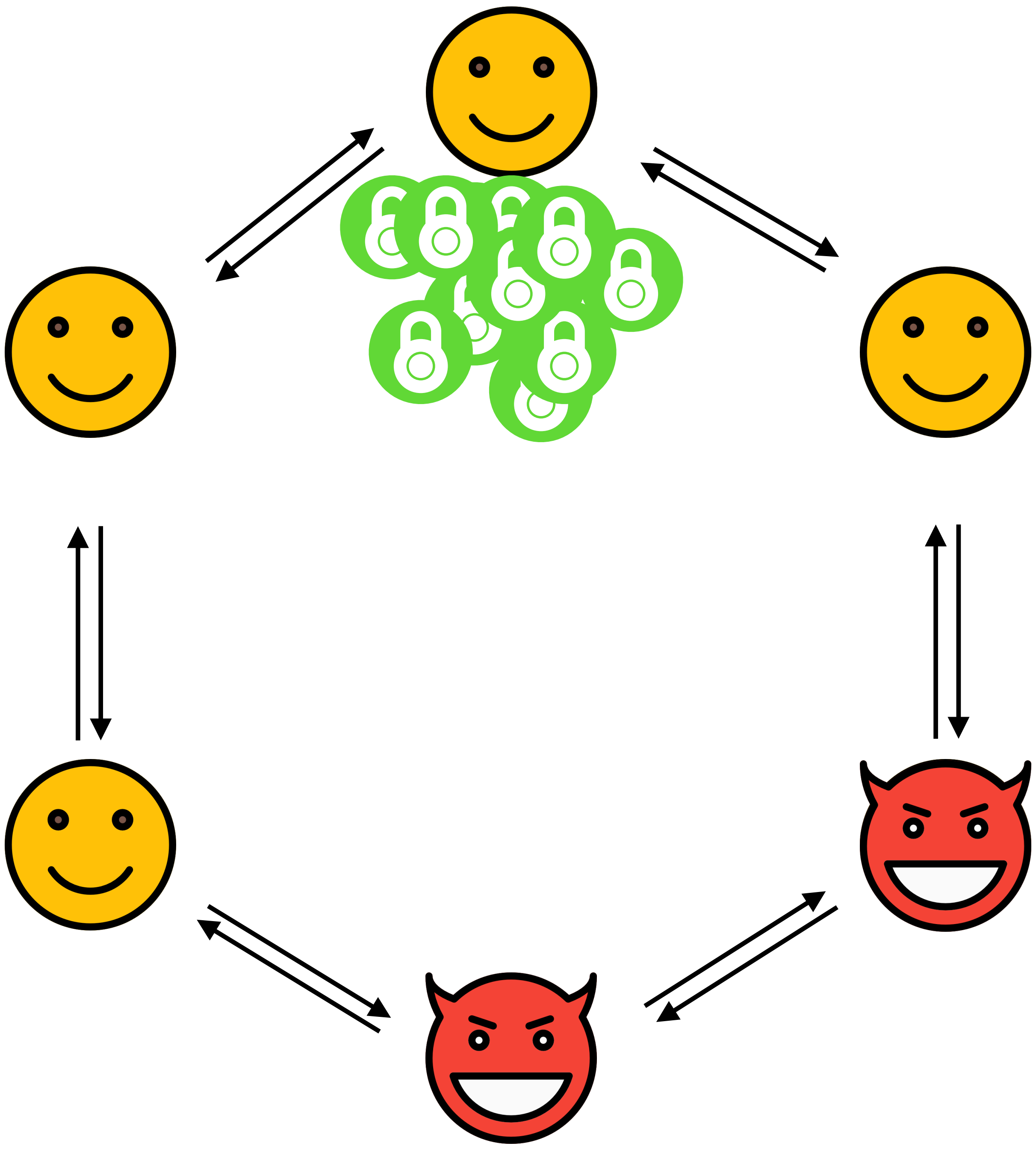
Magic-state generation



 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

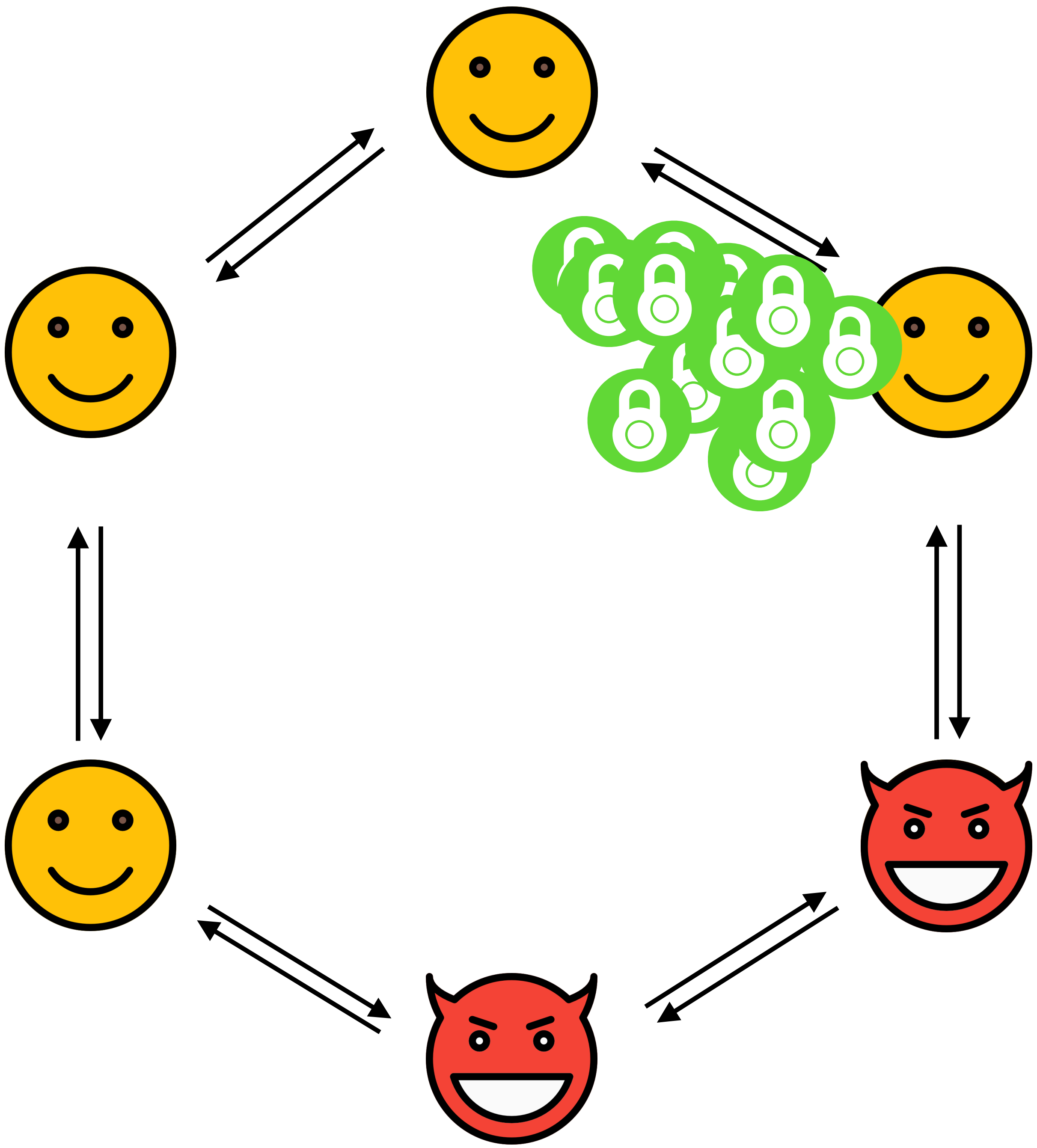
Magic-state generation



 $= C(T|+\rangle \otimes |0^n\rangle)$

1. “cut-and-choose”:

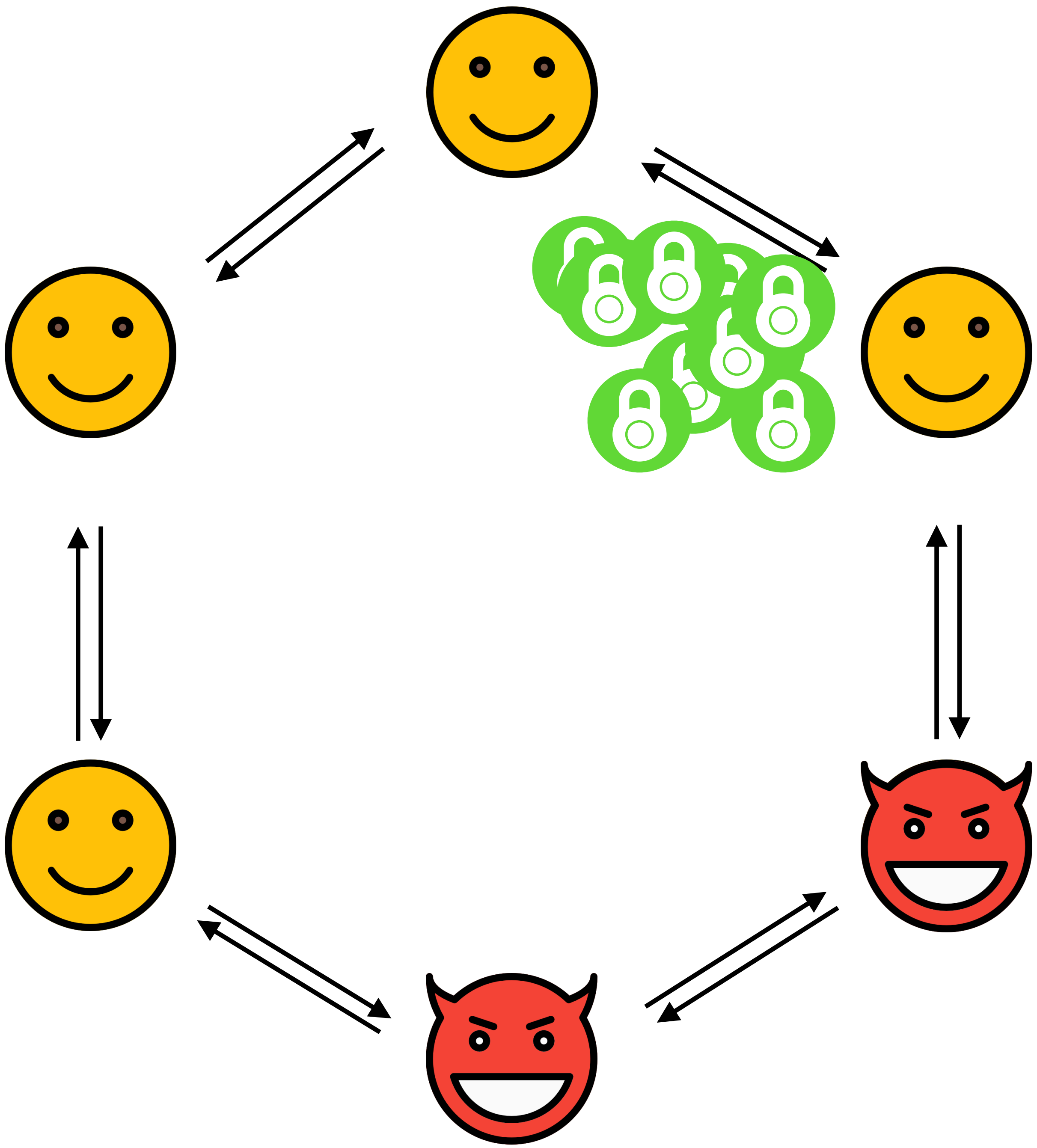
Magic-state generation



 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

Magic-state generation

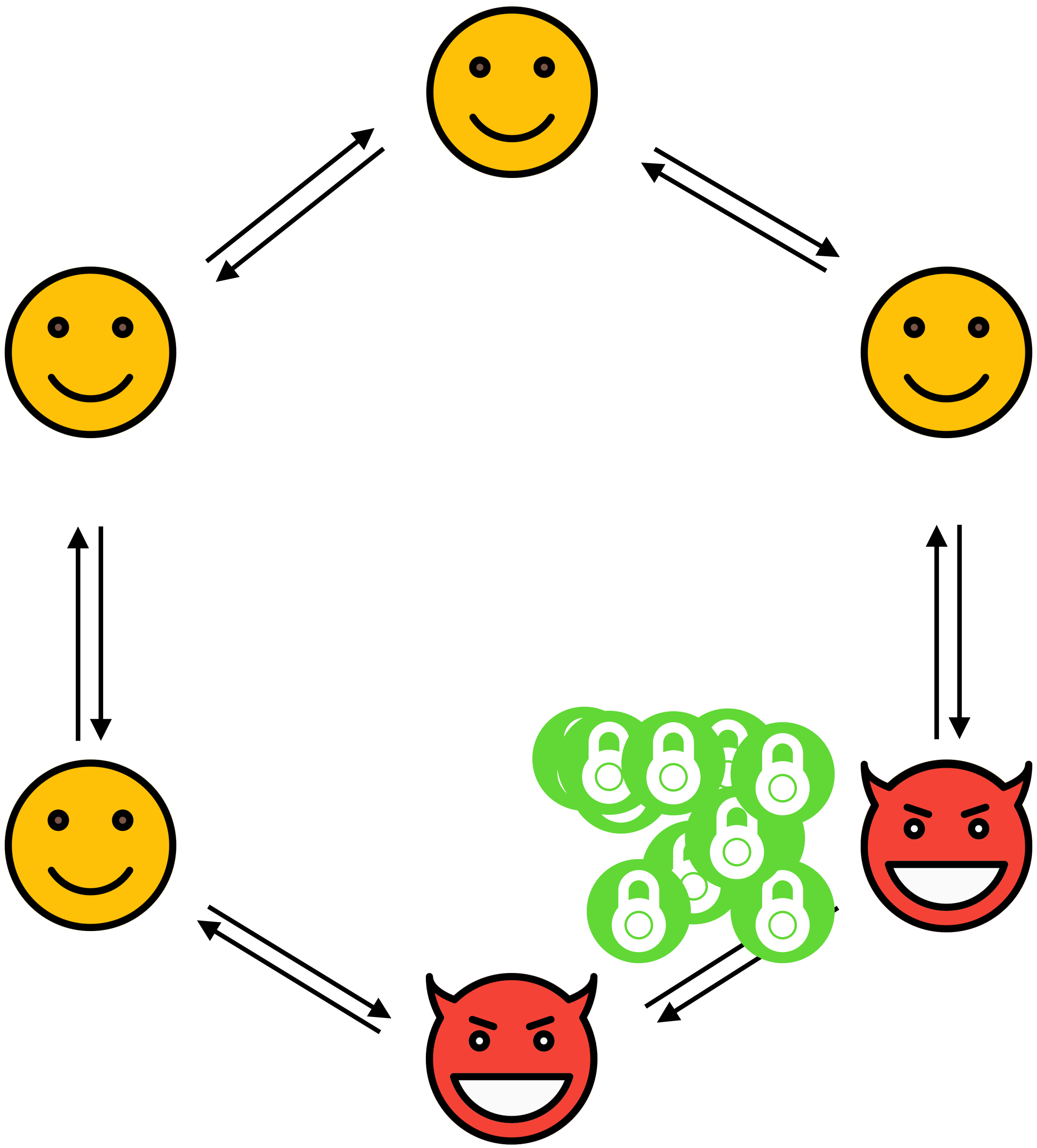


 $= C(T|+\rangle \otimes |0^n\rangle)$

1. “cut-and-choose”:

   - ✦ every player tests n random states 

Magic-state generation

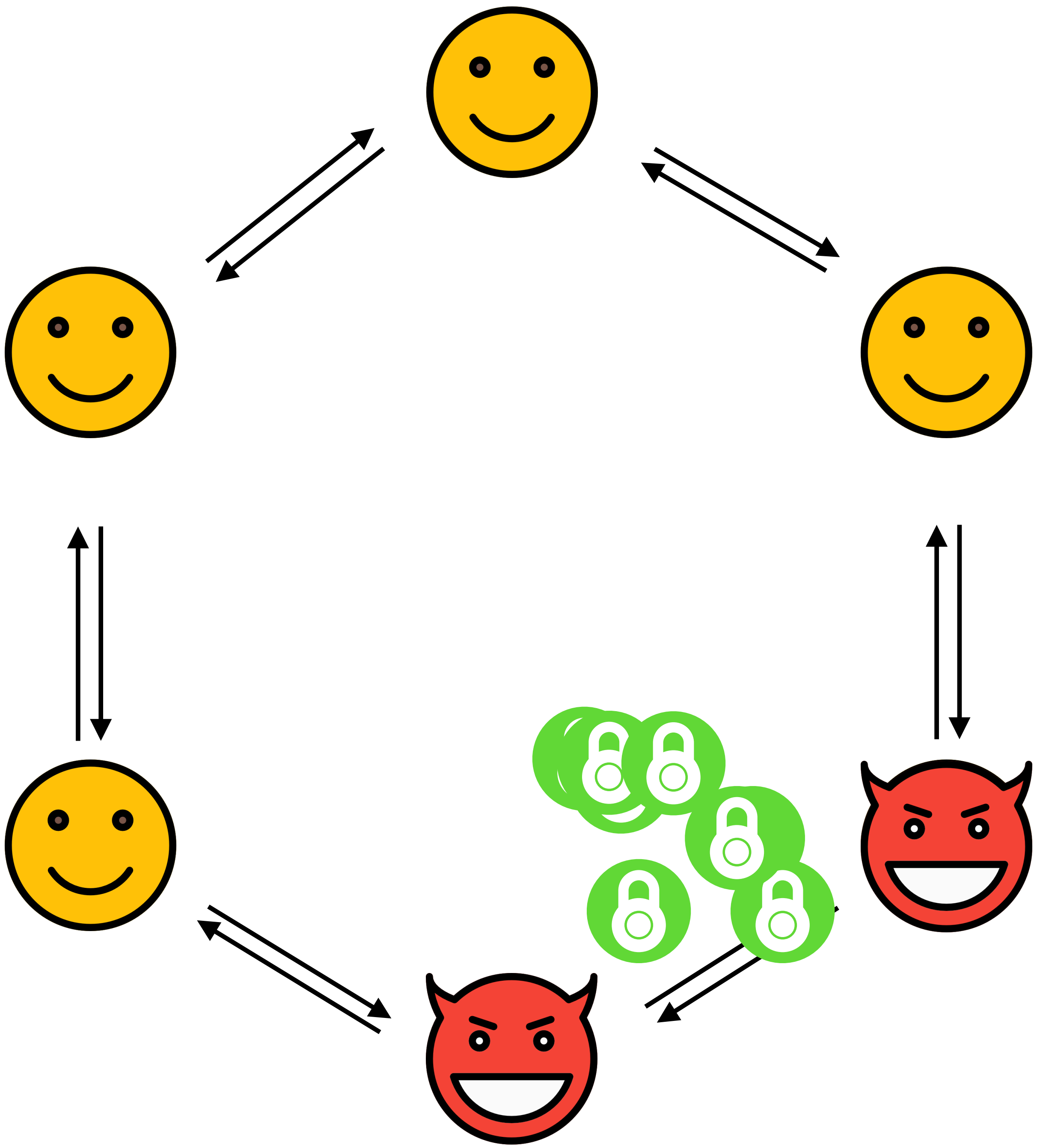


$= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

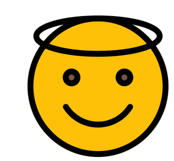
    - every player tests n random states

Magic-state generation

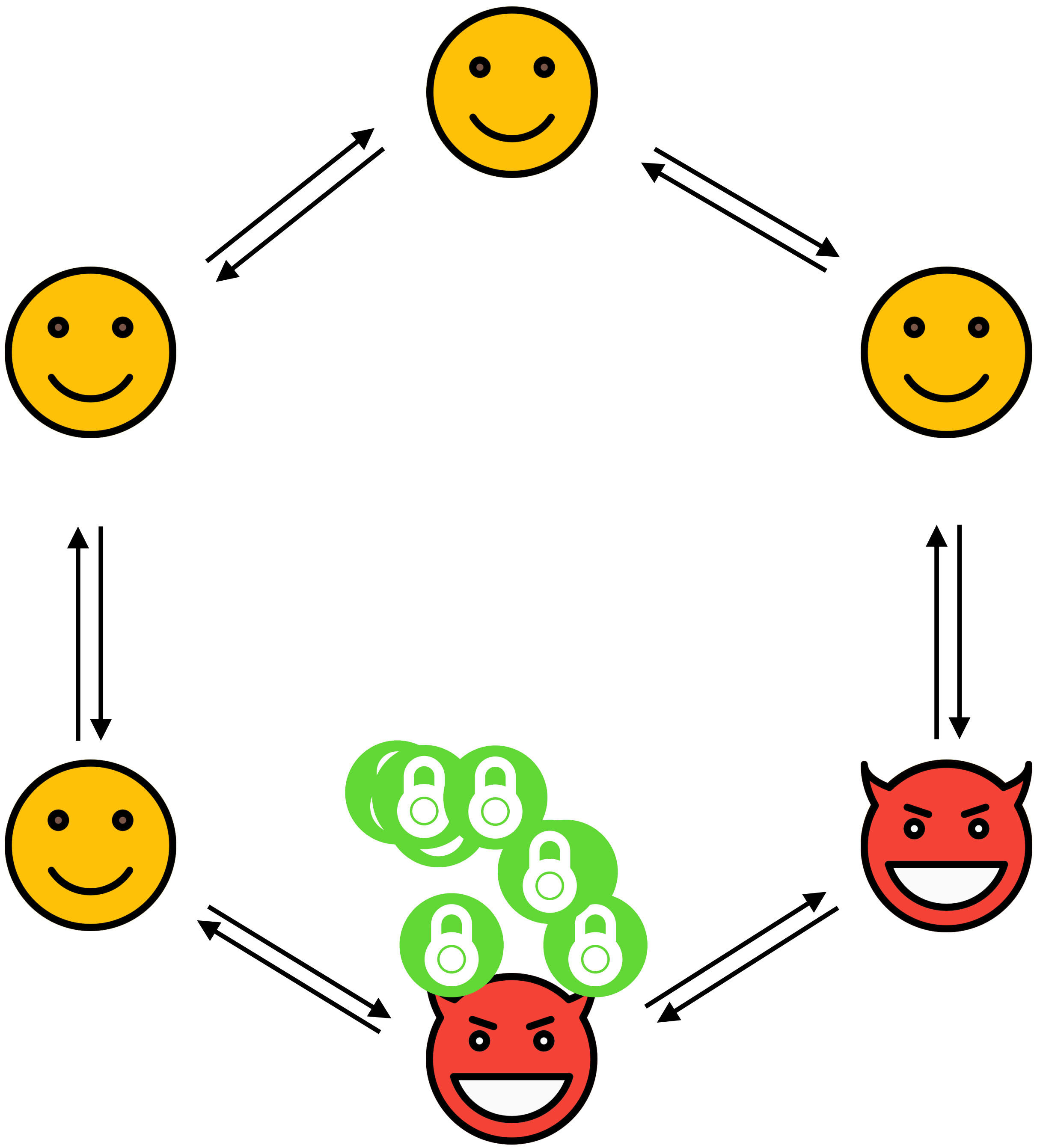


 $= C(T|+\rangle \otimes |0^n\rangle)$

1. “cut-and-choose”:

   ✦ every player tests n random states 

Magic-state generation

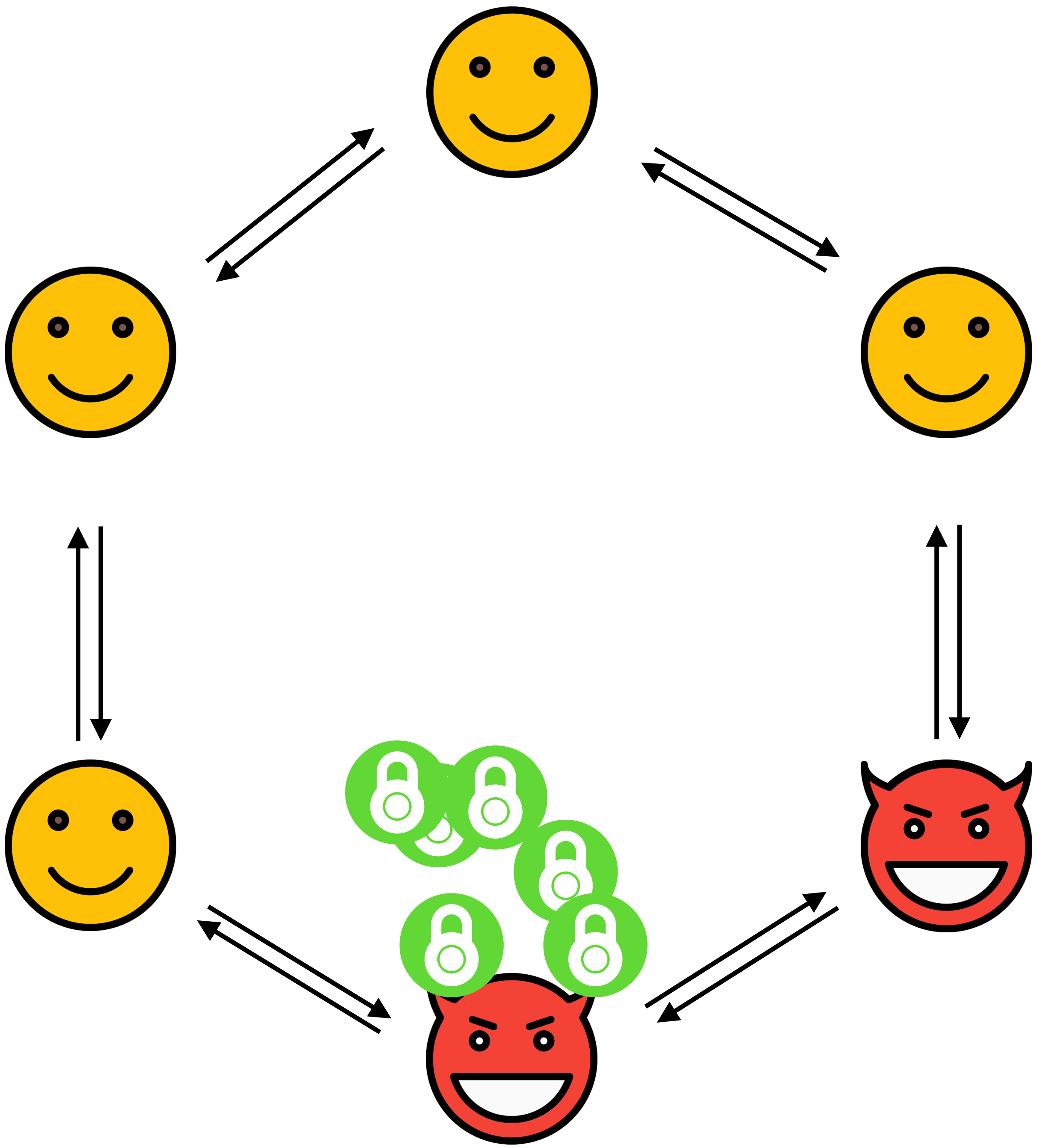


 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

    ✦ every player tests n random states 

Magic-state generation

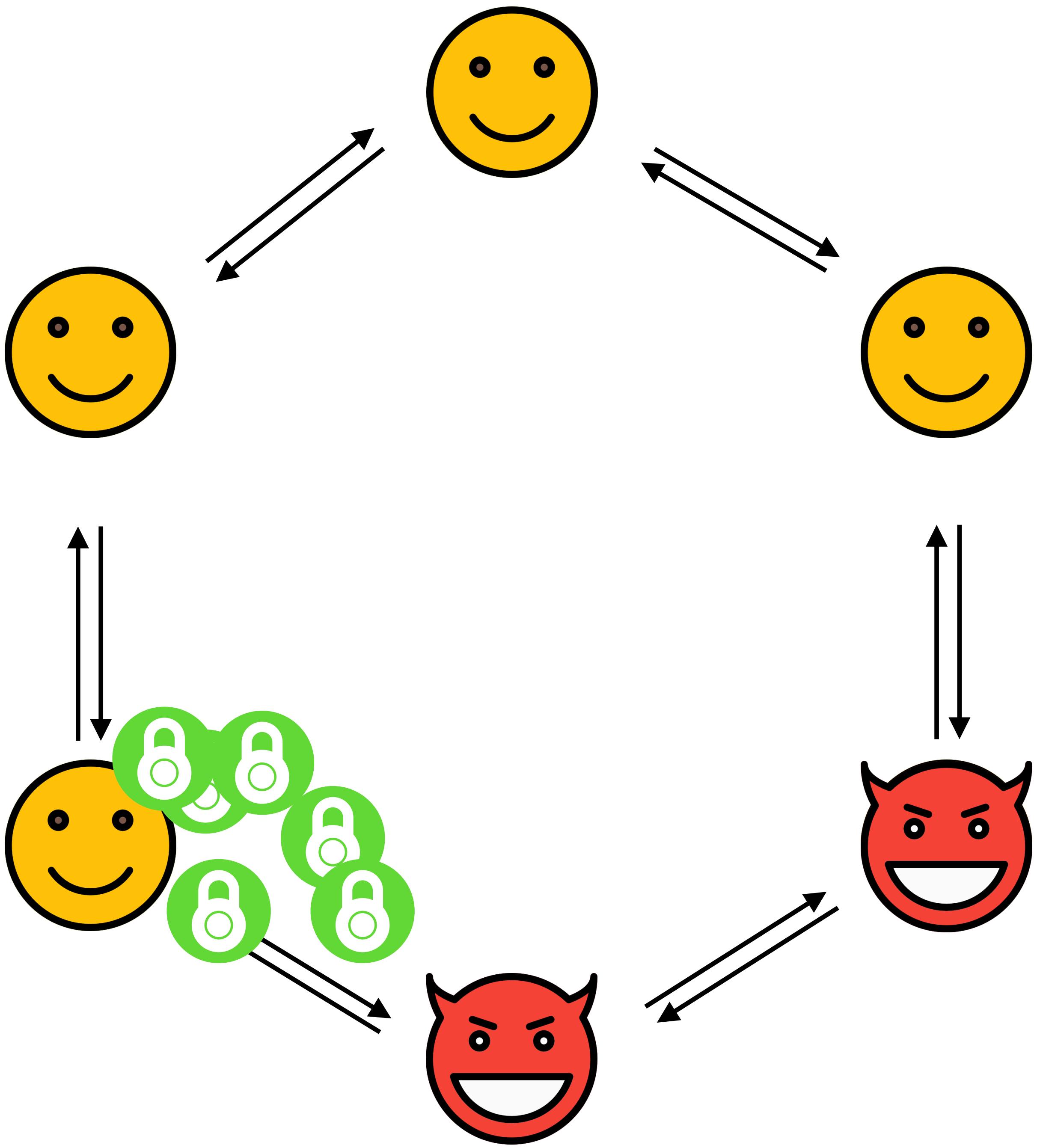


$= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

   ✦ every player tests n random states

Magic-state generation

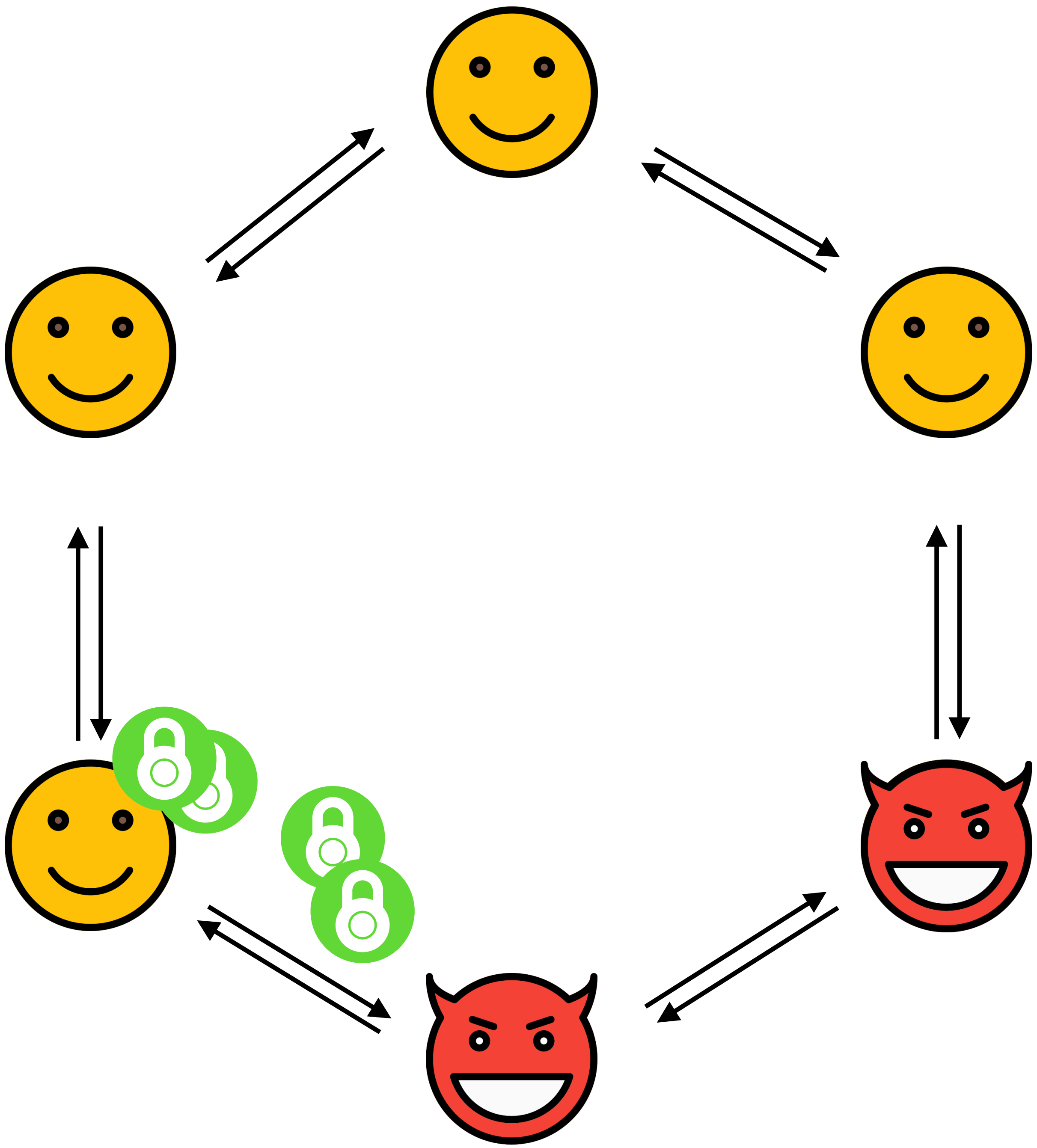


 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

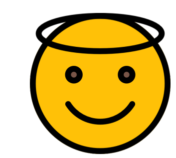
    ✦ every player tests n random states 

Magic-state generation

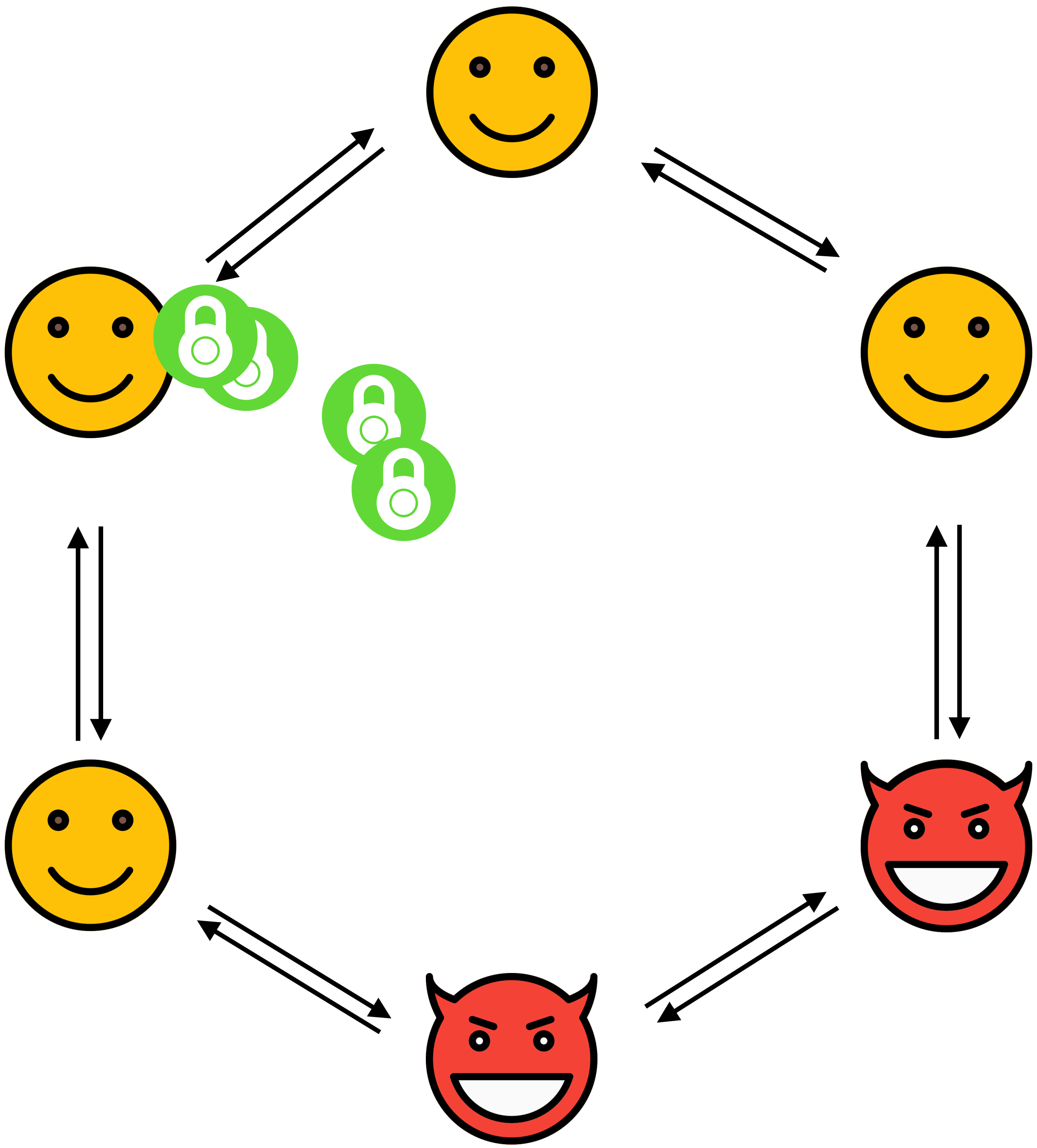


 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

   ✦ every player tests n random states 

Magic-state generation

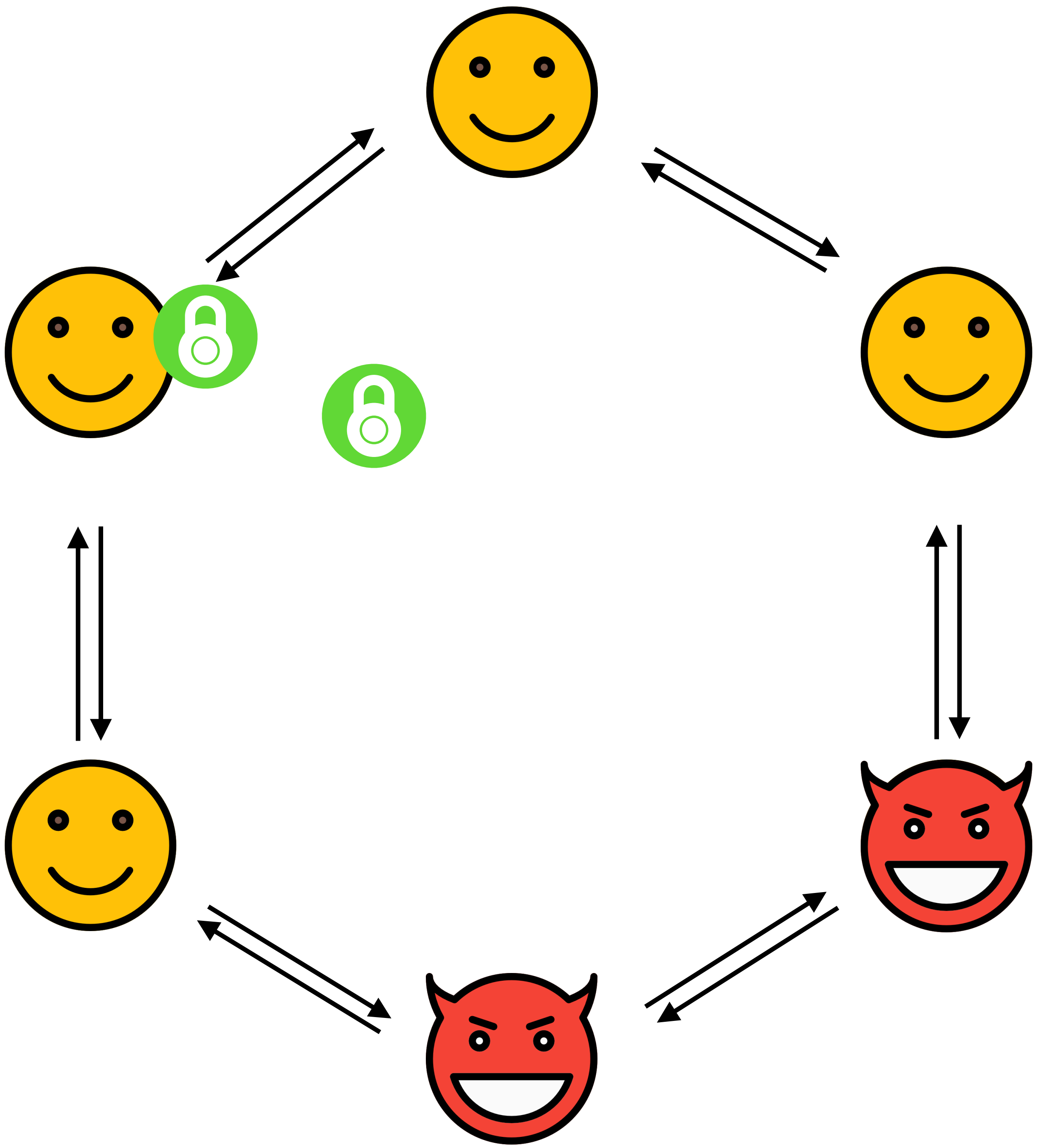


 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

   - every player tests n random states 

Magic-state generation

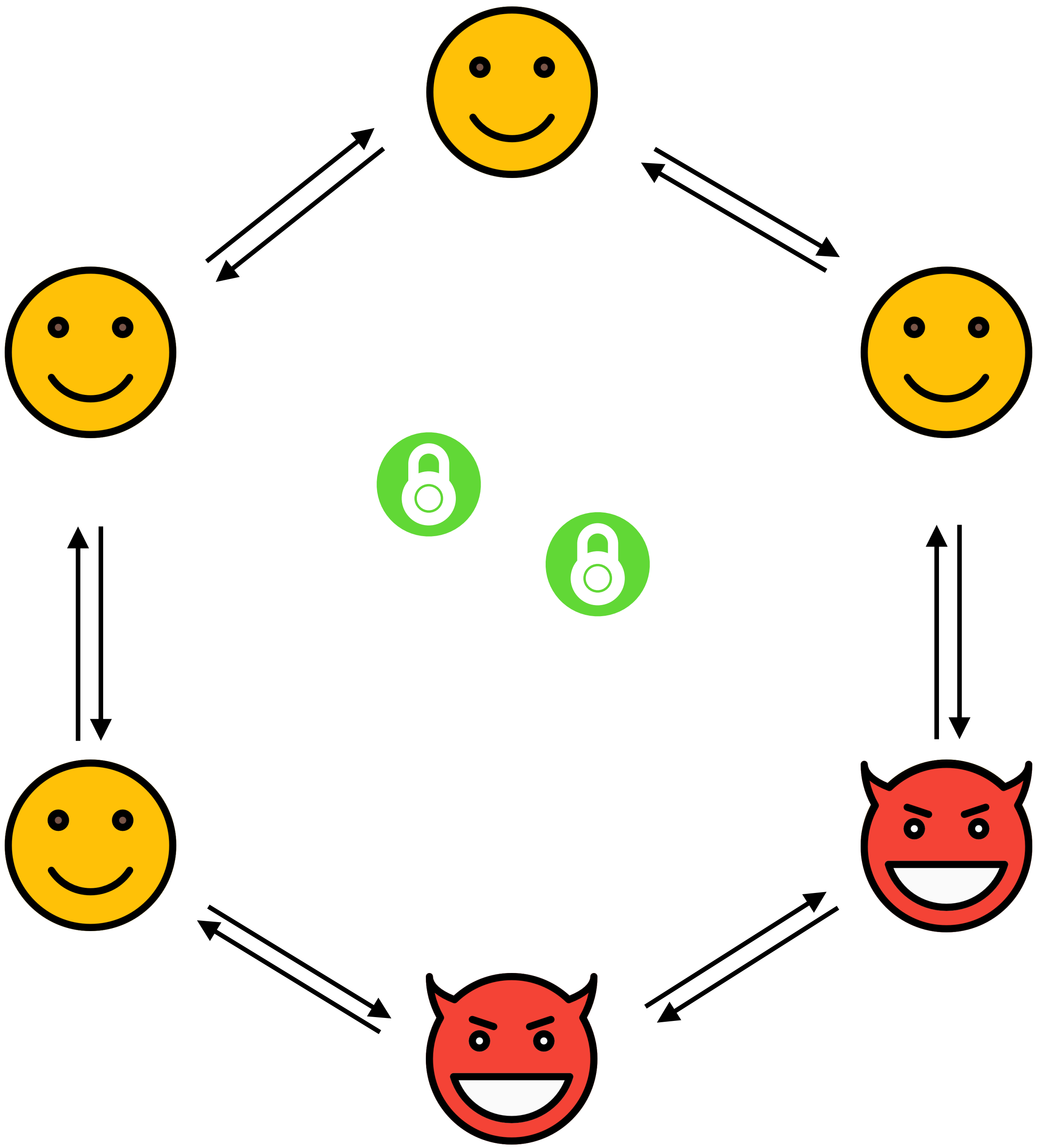


 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

   - every player tests n random states 

Magic-state generation

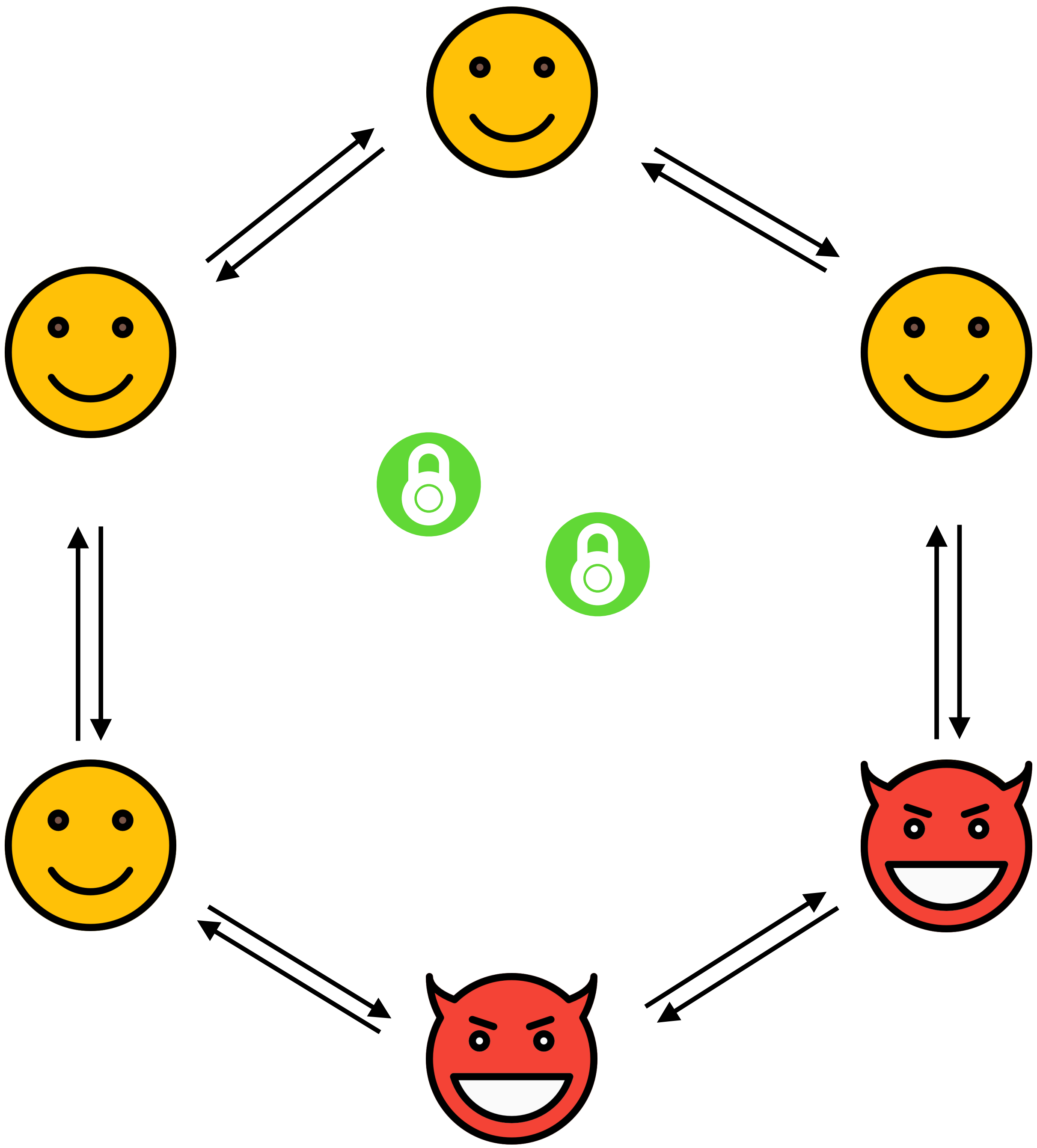


 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":

   - every player tests n random states 

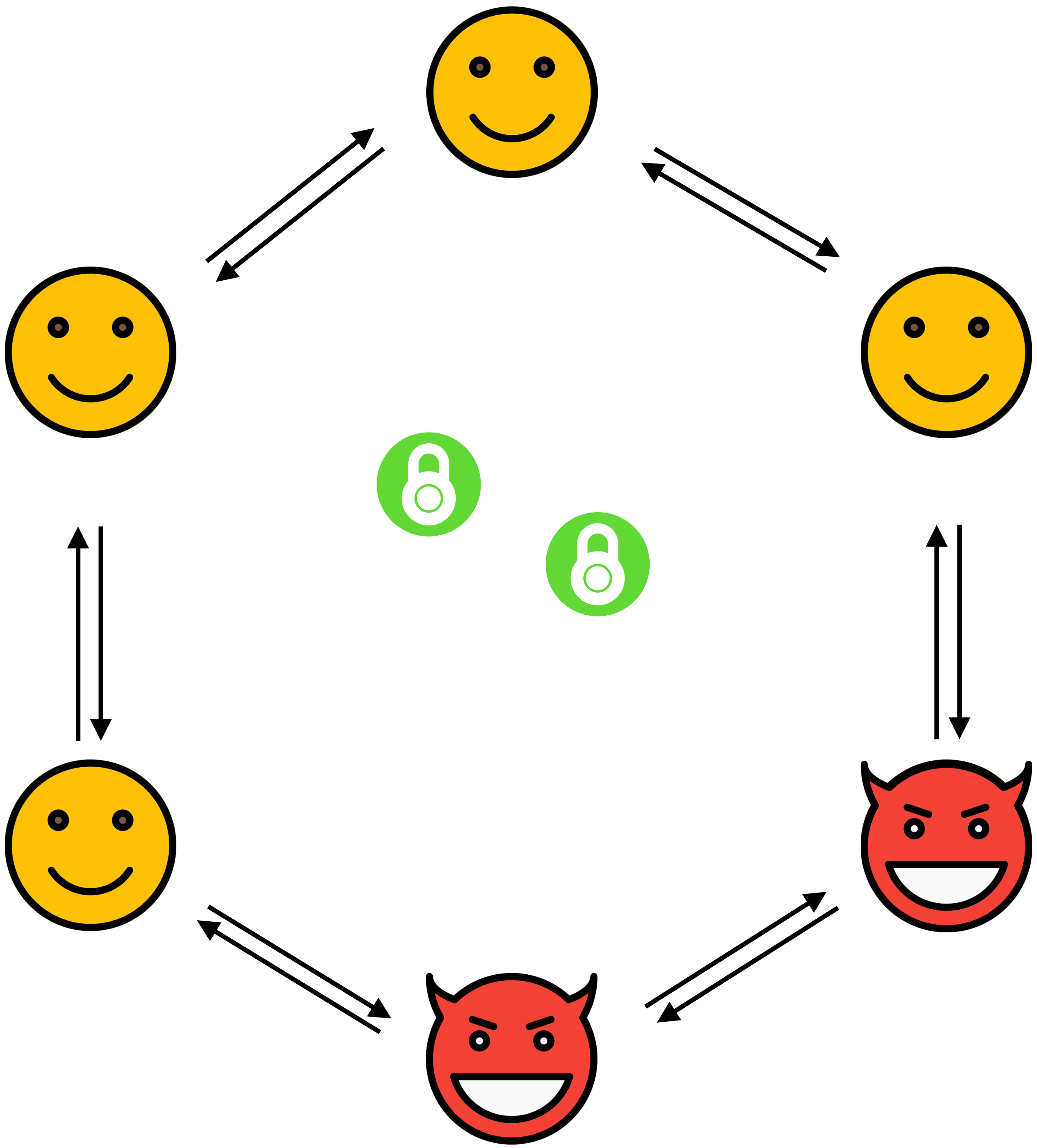
Magic-state generation



 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":
   - every player tests n random states
   - remaining n copies are "pretty good"

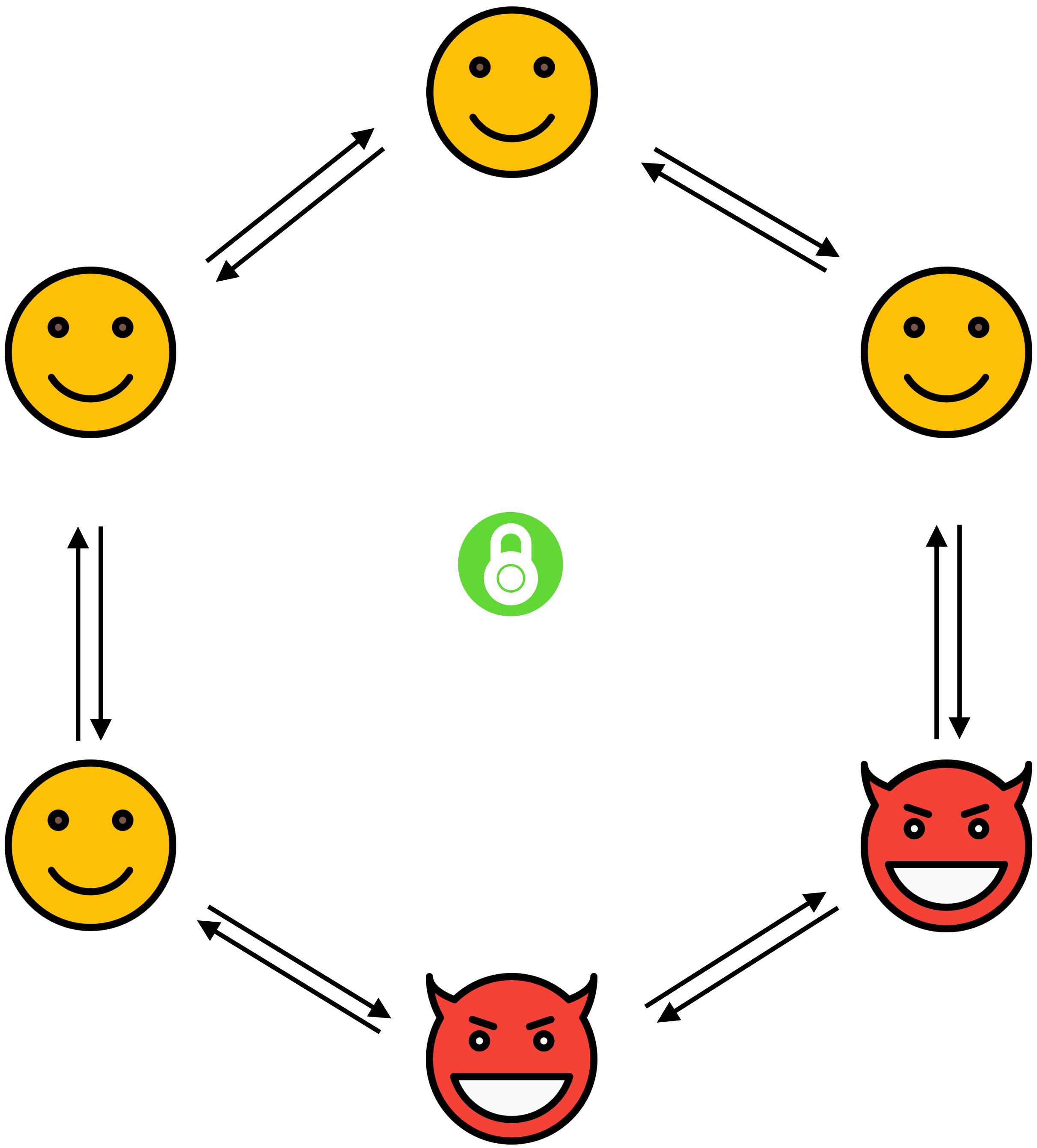
Magic-state generation



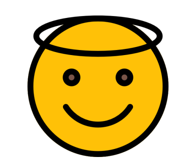
 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":
   - every player tests n random states
   - remaining n copies are "pretty good"
2. magic-state distillation:

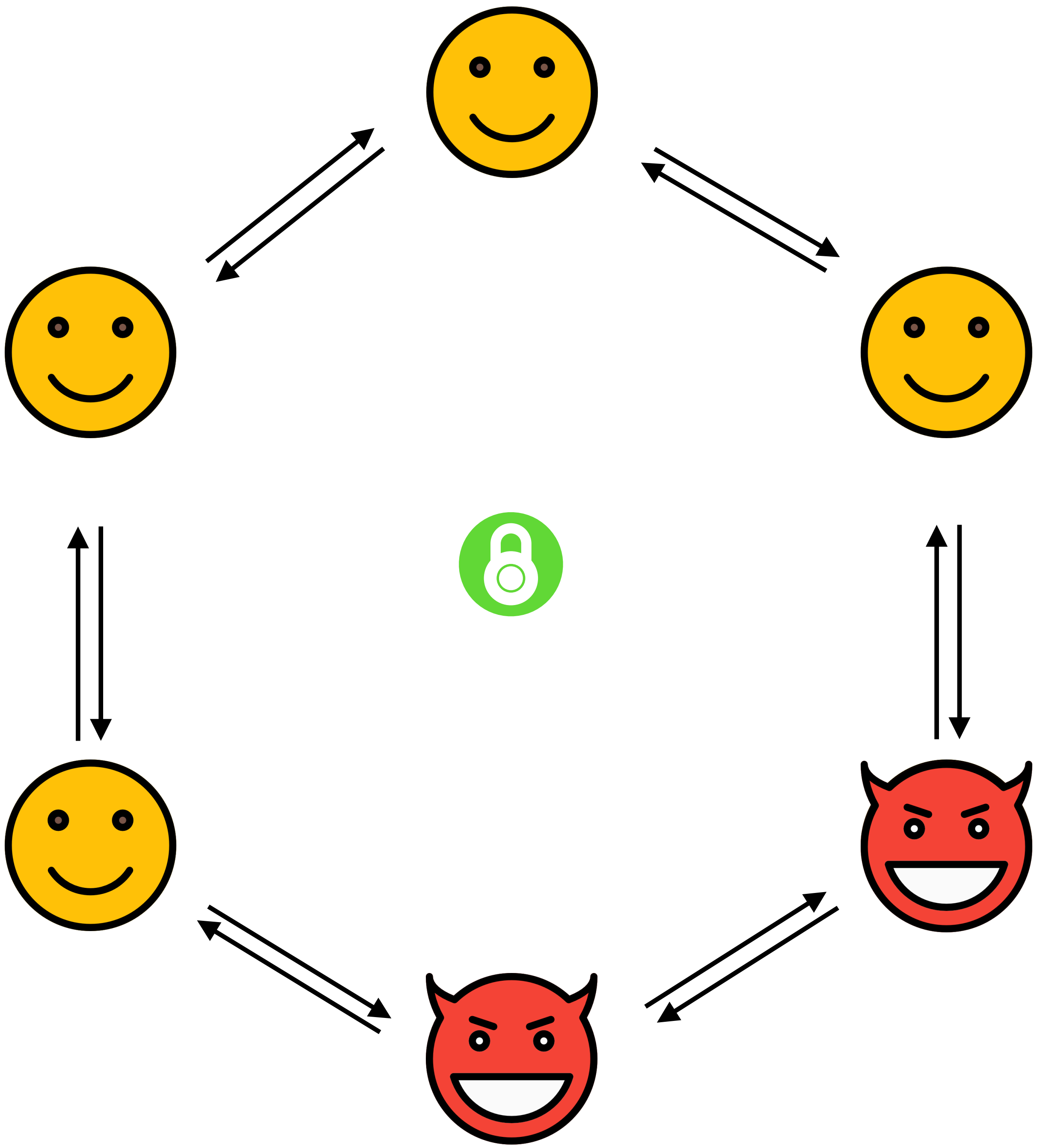
Magic-state generation



 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":
   - every player tests n random states 
   - remaining n copies are "pretty good"
2. magic-state distillation:
   - a Clifford circuit 

Magic-state generation



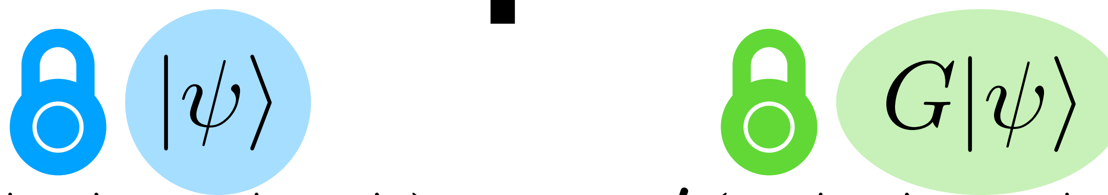
 $= C(T|+\rangle \otimes |0^n\rangle)$

1. "cut-and-choose":
   - every player tests n random states 
   - remaining n copies are "pretty good"
2. magic-state distillation:
   - a Clifford circuit 
   - remaining copy is "very good"

Computation

$|\psi\rangle$ $G|\psi\rangle$

Computation



Protocols ($C(|\psi\rangle \otimes |0^n\rangle) \mapsto C'(G|\psi\rangle \otimes |0^n\rangle)$) for these G :

Computation

 $|\psi\rangle$  $G|\psi\rangle$

Protocols ($C(|\psi\rangle \otimes |0^n\rangle) \mapsto C'(G|\psi\rangle \otimes |0^n\rangle)$) for these G :

- 1-qubit Cliffords $C \;\mapsto\; C' := C(G^\dagger \otimes I^{\otimes n})$ 

Computation

$|\psi\rangle$ $G|\psi\rangle$

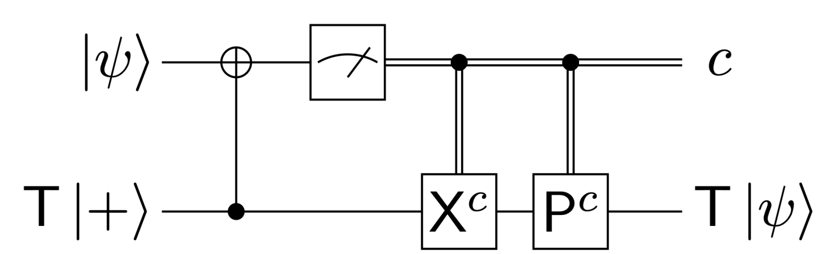
Protocols ($C(|\psi\rangle \otimes |0^n\rangle) \mapsto C'(G|\psi\rangle \otimes |0^n\rangle)$) for these G :

- 1-qubit Cliffords $C \;\mapsto\; C' := C(G^\dagger \otimes I^{\otimes n})$
- CNOT (2-qubit Clifford): apply $(C_1' \otimes C_2')CNOT(C_1^\dagger \otimes C_2^\dagger)$ & perform public authentication test

Computation

Protocols ($C(|\psi\rangle \otimes |0^n\rangle) \mapsto C'(G|\psi\rangle \otimes |0^n\rangle)$) for these G :

- 1-qubit Cliffords $\quad C \;\mapsto\; C' := C(G^\dagger \otimes I^{\otimes n})$
- CNOT (2-qubit Clifford): apply $(C_1' \otimes C_2')CNOT(C_1^\dagger \otimes C_2^\dagger)$ & perform public authentication test
- T (non-Clifford): generate encoded magic states and perform encoded magic-state computation

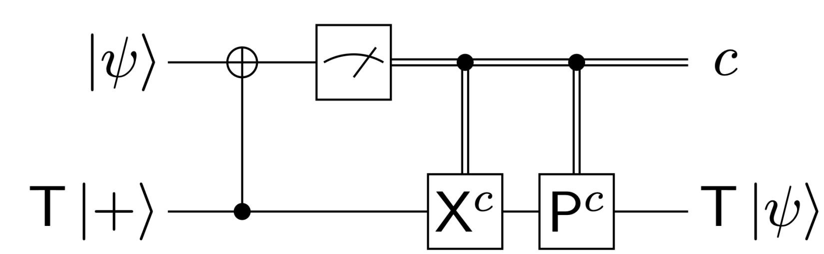


Computation

$|\psi\rangle$ $G|\psi\rangle$

Protocols ($C(|\psi\rangle \otimes |0^n\rangle) \mapsto C'(G|\psi\rangle \otimes |0^n\rangle)$) for these G :

- 1-qubit Cliffords $C \mapsto C' := C(G^\dagger \otimes I^{\otimes n})$ 
- CNOT (2-qubit Clifford): apply $(C_1' \otimes C_2')CNOT(C_1^\dagger \otimes C_2^\dagger)$ & perform public authentication test
- T (non-Clifford): generate encoded magic states and perform encoded magic-state computation



- Computational-basis measurement

Comp-Basis Measurement

 $= C(|\psi\rangle \otimes |0\rangle^{\otimes n}) = C\left(\sum_{m \in \{0,1\}} \alpha_m |m\rangle \otimes |0\rangle^{\otimes n}\right)$

Comp-Basis Measurement

 $= C(|\psi\rangle \otimes |0\rangle^{\otimes n}) = C\left(\sum_{m\in\{0,1\}} \alpha_m |m\rangle \otimes |0\rangle^{\otimes n}\right)$

Comp-Basis Measurement

$$= C(|\psi\rangle \otimes |0\rangle^{\otimes n}) = C\left(\sum_{m\in\{0,1\}} \alpha_m |m\rangle \otimes |0\rangle^{\otimes n}\right)$$

- MPC samples $r, s \leftarrow \{0,1\}^{n+1}, c \leftarrow \{0,1\}^n$

Comp-Basis Measurement

 $$= C(|\psi\rangle \otimes |0\rangle^{\otimes n}) = C\left(\sum_{m\in\{0,1\}} \alpha_m |m\rangle \otimes |0\rangle^{\otimes n}\right)$$

 • MPC samples $r, s \leftarrow \{0,1\}^{n+1}, c \leftarrow \{0,1\}^n$

 • MPC instructs player to apply $V := \mathsf{X}^r \mathsf{Z}^s \prod_{i\in[n]} \mathsf{CNOT}_{1,i}^{c_i}\ \mathsf{C}^\dagger$

Comp-Basis Measurement

$$= C(|\psi\rangle \otimes |0\rangle^{\otimes n}) = C\left(\sum_{m\in\{0,1\}} \alpha_m |m\rangle \otimes |0\rangle^{\otimes n}\right)$$

- MPC samples $r, s \leftarrow \{0,1\}^{n+1}, c \leftarrow \{0,1\}^n$
- MPC instructs player to apply $V := \mathsf{X}^r \mathsf{Z}^s \prod_{i\in[n]} \mathsf{CNOT}_{1,i}^{c_i}\, \mathsf{C}^\dagger$
- player measures in computational basis, outcome r'

Comp-Basis Measurement

$$= C(|\psi\rangle \otimes |0\rangle^{\otimes n}) = C\left(\sum_{m\in\{0,1\}} \alpha_m |m\rangle \otimes |0\rangle^{\otimes n}\right)$$

- MPC samples $r, s \leftarrow \{0,1\}^{n+1}, c \leftarrow \{0,1\}^n$
- MPC instructs player to apply $V := \mathsf{X}^r \mathsf{Z}^s \prod_{i\in[n]} \mathsf{CNOT}_{1,i}^{c_i}\ \mathsf{C}^\dagger$
- player measures in computational basis, outcome r'
- MPC checks whether $r' = r \oplus (m, m\cdot c)$ for some $m \in \{0,1\}$

Summary

Summary

A protocol for multiparty computation of any quantum circuit:

Summary

A protocol for multiparty computation of any quantum circuit:

- Computationally secure against $\leq k-1$ cheaters (out of k)

Summary

A protocol for multiparty computation of any quantum circuit:

- Computationally secure against $\leq k-1$ cheaters (out of k)
- Encoded states of size $2n+1$ (vs. $kn+1$ in [DNS12])

Summary

A protocol for multiparty computation of any quantum circuit:

- Computationally secure against $\leq k-1$ cheaters (out of k)
- Encoded states of size $2n+1$ (vs. $kn+1$ in [DNS12])
- T gate: requires kn magic states (vs. n^k from naive extension of [DNS12])

Summary

A protocol for multiparty computation of any quantum circuit:

- Computationally secure against $\leq k-1$ cheaters (out of k)
- Encoded states of size $2n+1$ (vs. $kn+1$ in [DNS12])
- T gate: requires kn magic states (vs. n^k from naive extension of [DNS12])
- Rounds of q communication: $O\big(k(d+\log(n))\big)$
  for d the $\{\mathsf{CNOT},\mathsf{T}\}$-depth of the quantum computation

Open Questions

Open Questions

- 😇 Post-quantum secure classical MPC protocol for dishonest majority?

Open Questions

- Post-quantum secure classical MPC protocol for dishonest majority?
- CNOT operation without a round of q communication?

Open Questions

- Post-quantum secure classical MPC protocol for dishonest majority?
- CNOT operation without a round of q communication?
- More efficient protocols for more specific functionalities?

Open Questions

- 😇 Post-quantum secure classical MPC protocol for dishonest majority?
- CNOT operation without a round of q communication?
- More efficient protocols for more specific functionalities?
- Classical MPC is a versatile tool (e.g. for zero-knowledge proofs or digital signature). Are there such use cases for MPQC?

Open Questions

- Post-quantum secure classical MPC protocol for dishonest majority?
- CNOT operation without a round of q communication?
- More efficient protocols for more specific functionalities?
- Classical MPC is a versatile tool (e.g. for zero-knowledge proofs or digital signature). Are there such use cases for MPQC?

Nederlandse Organisatie voor Wetenschappelijk Onderzoek

SIMONS INSTITUTE for the Theory of Computing

Thank you!