# Validity in Network-Agnostic Byzantine Agreement


Andrei Constantinescu
**Marc Dufay**
Diana Ghinea
Roger Wattenhofer

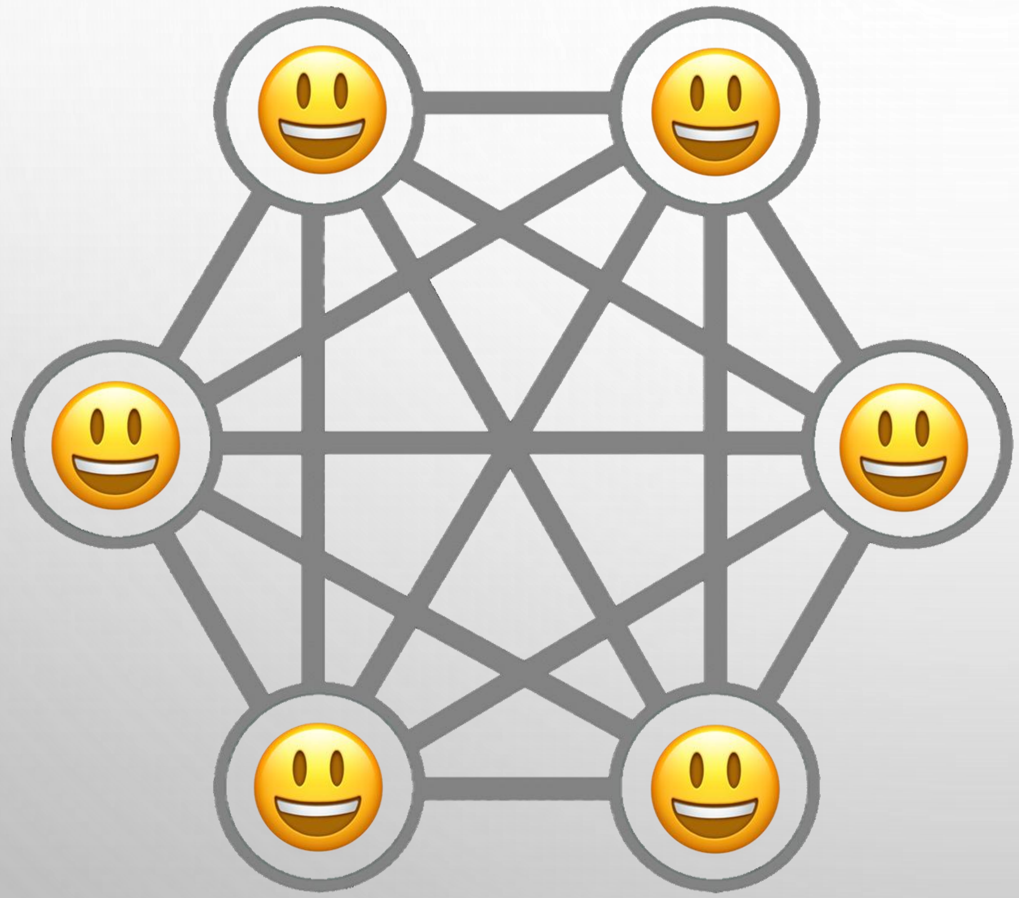




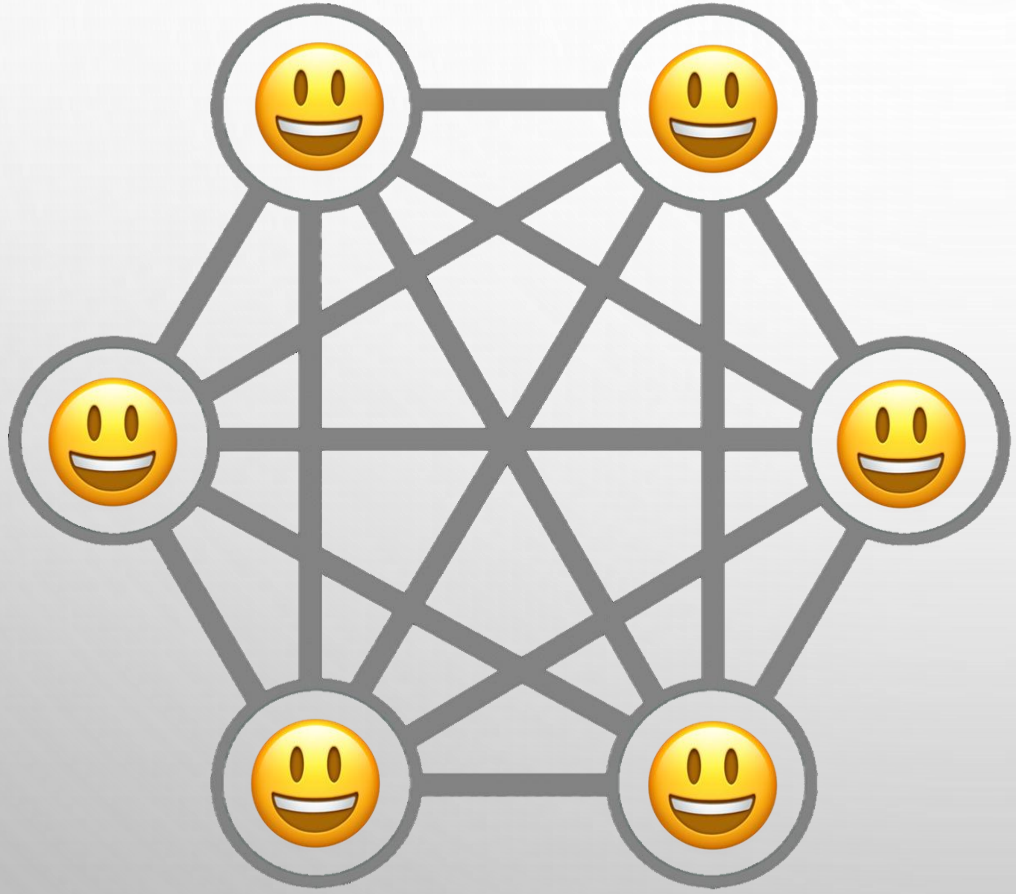

**Dis**tributed **Co**mputing Group
ETH zürich


**DISC 2025**

# MODEL

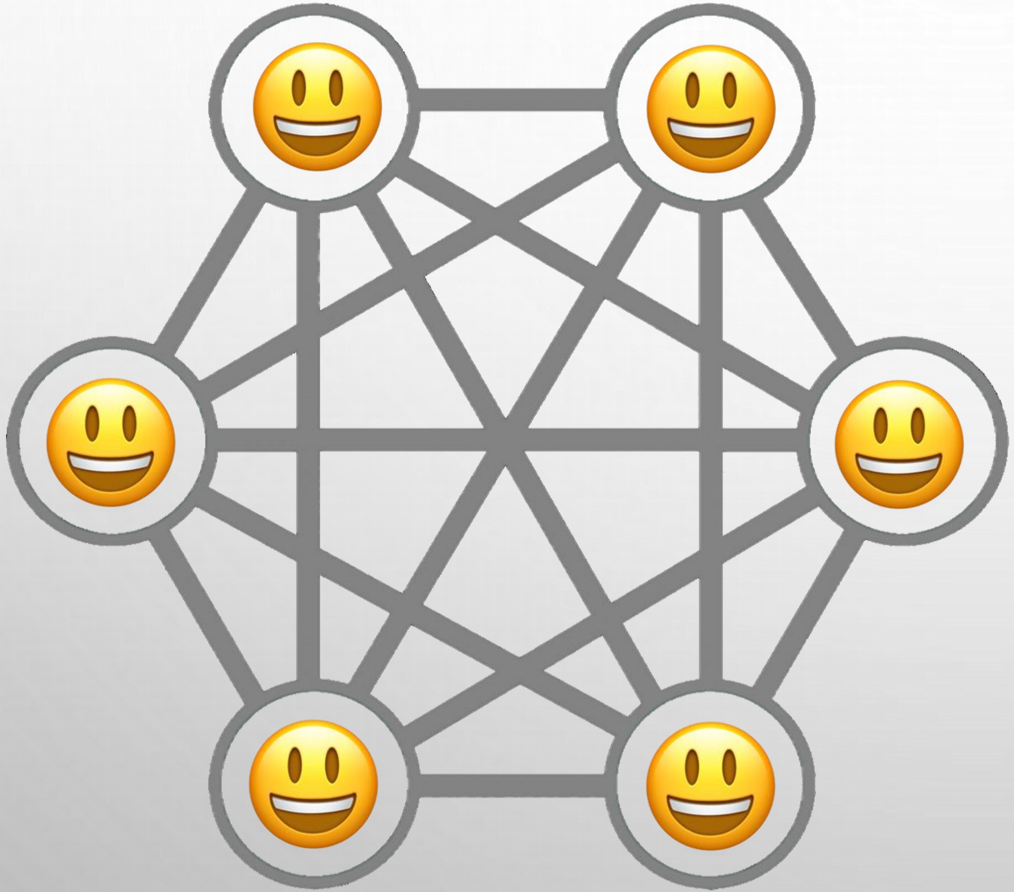


# MODEL



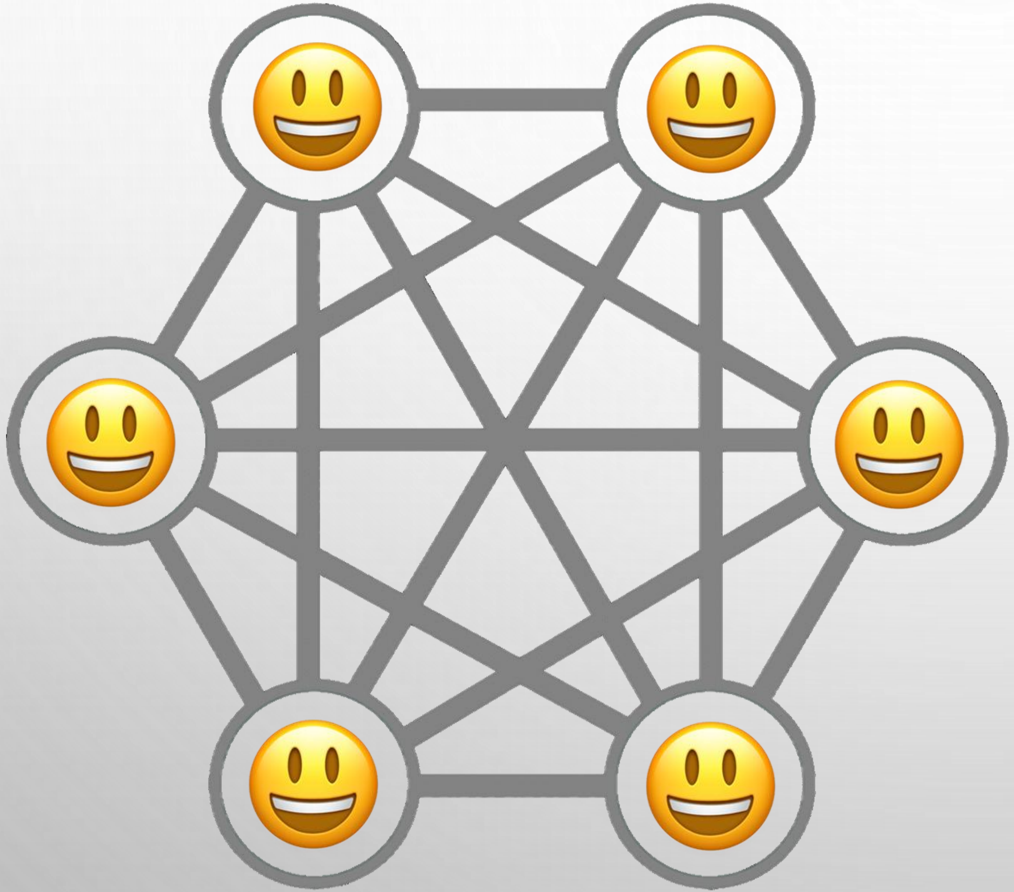
- $n$ parties

# MODEL



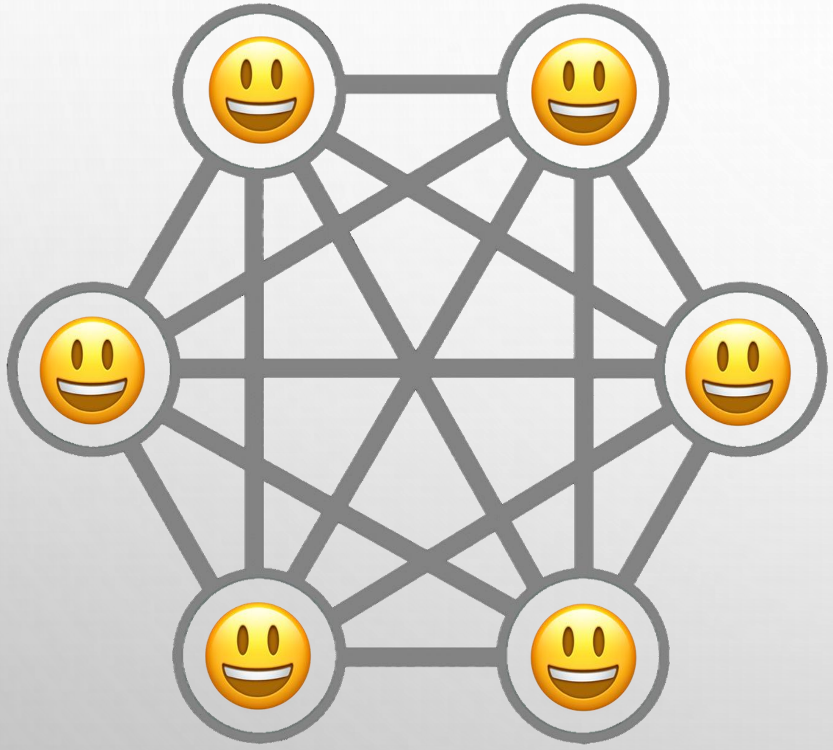
- $\boldsymbol{n}$ parties
- Fully connected network

# MODEL

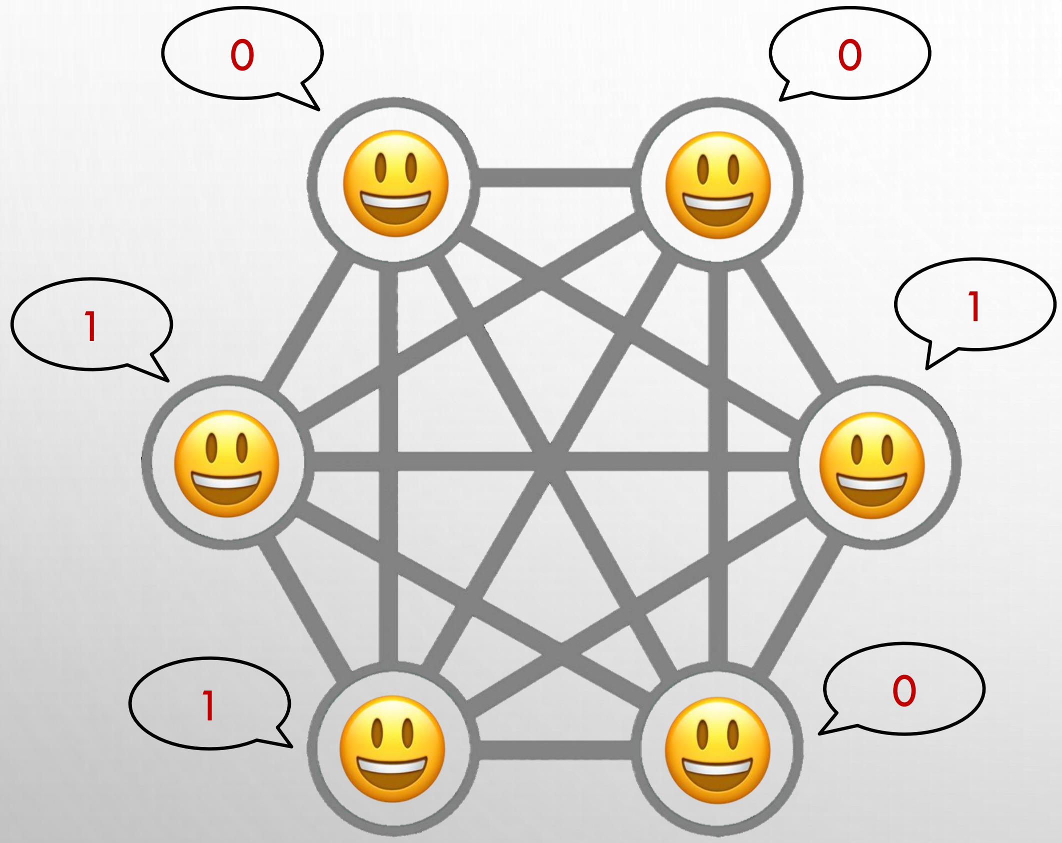


- $n$ parties
- Fully connected network
- Authenticated channels

# AGREEMENT

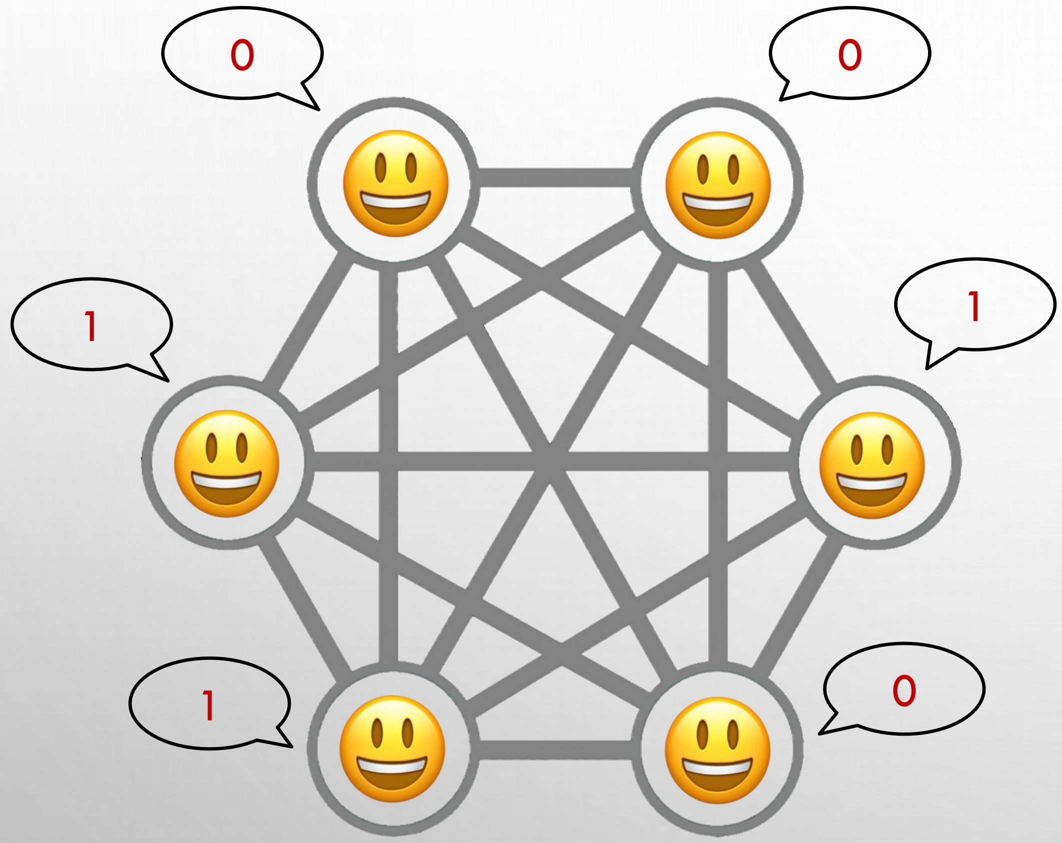


# AGREEMENT




$$V_{IN} = \{0,1\}$$

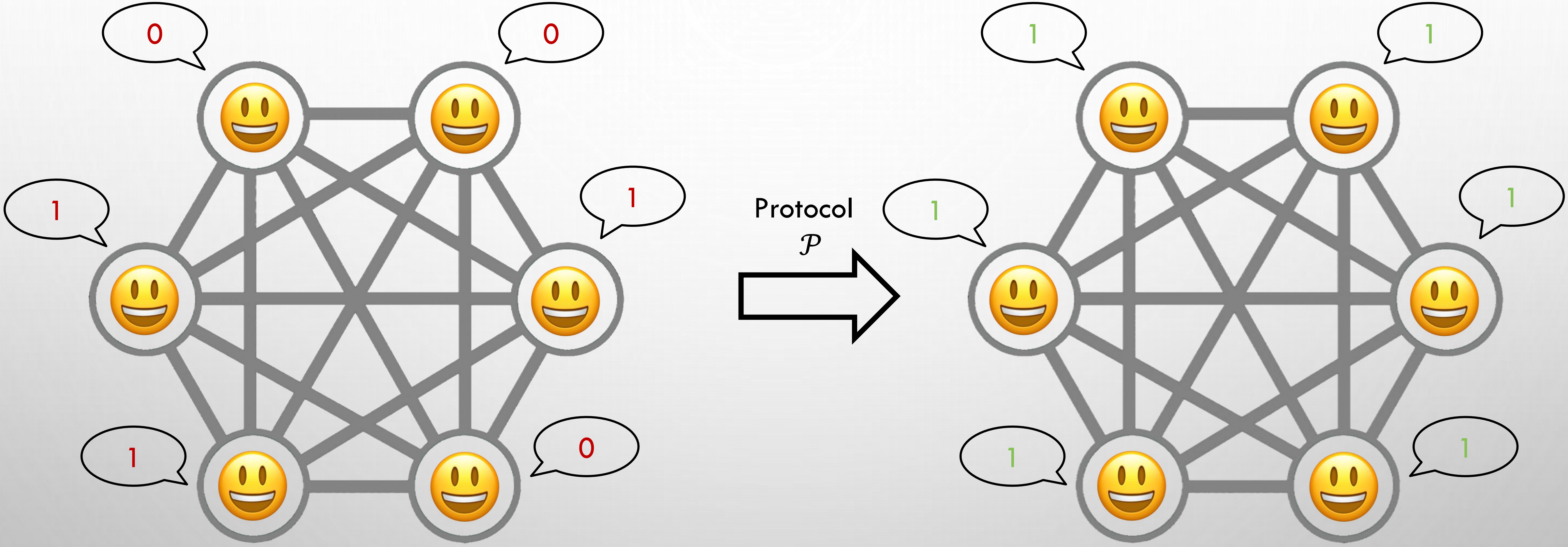
# AGREEMENT




$V_{IN} = \{0,1\}$

Protocol $\mathcal{P}$

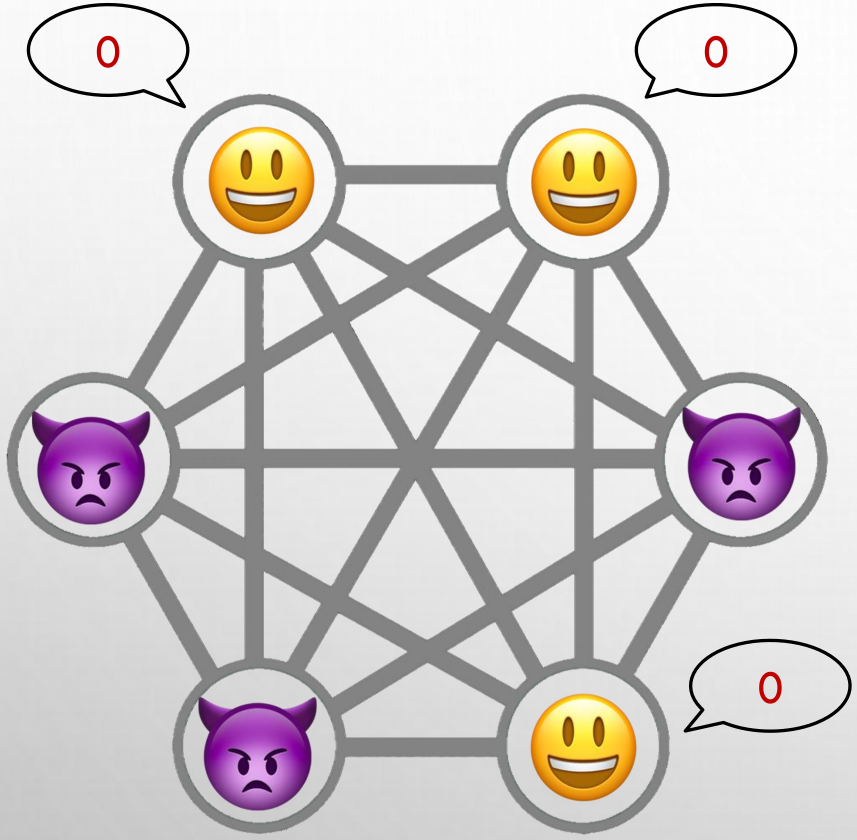
# AGREEMENT



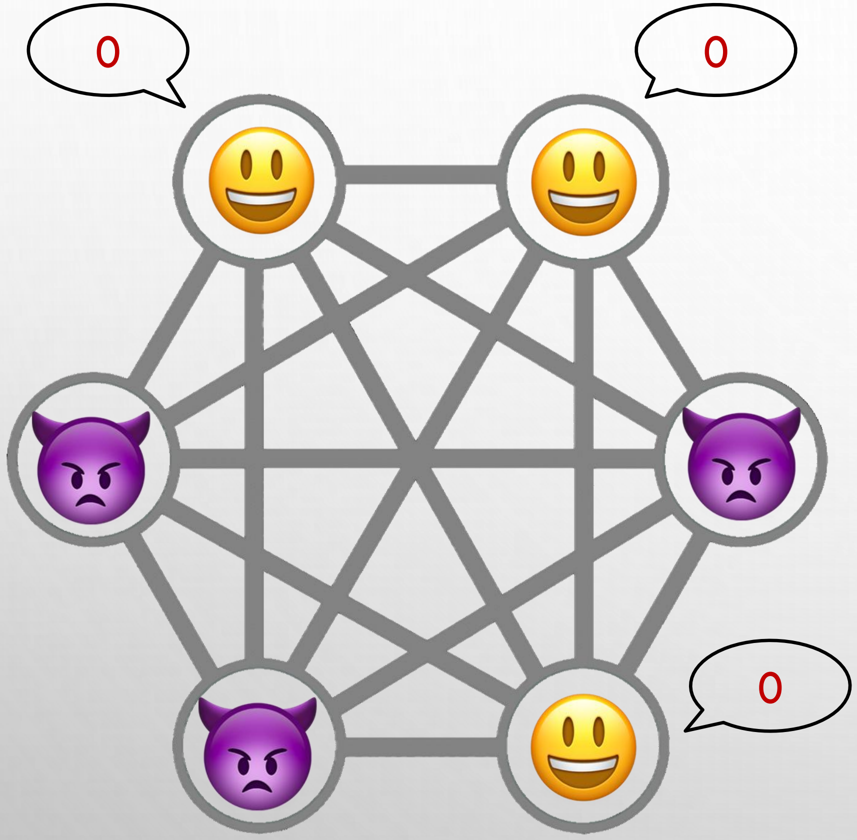

$V_{IN} = \{0,1\}$

$V_{\mathrm{OUT}} = \{0,1\}$

# Byzantine Agreement

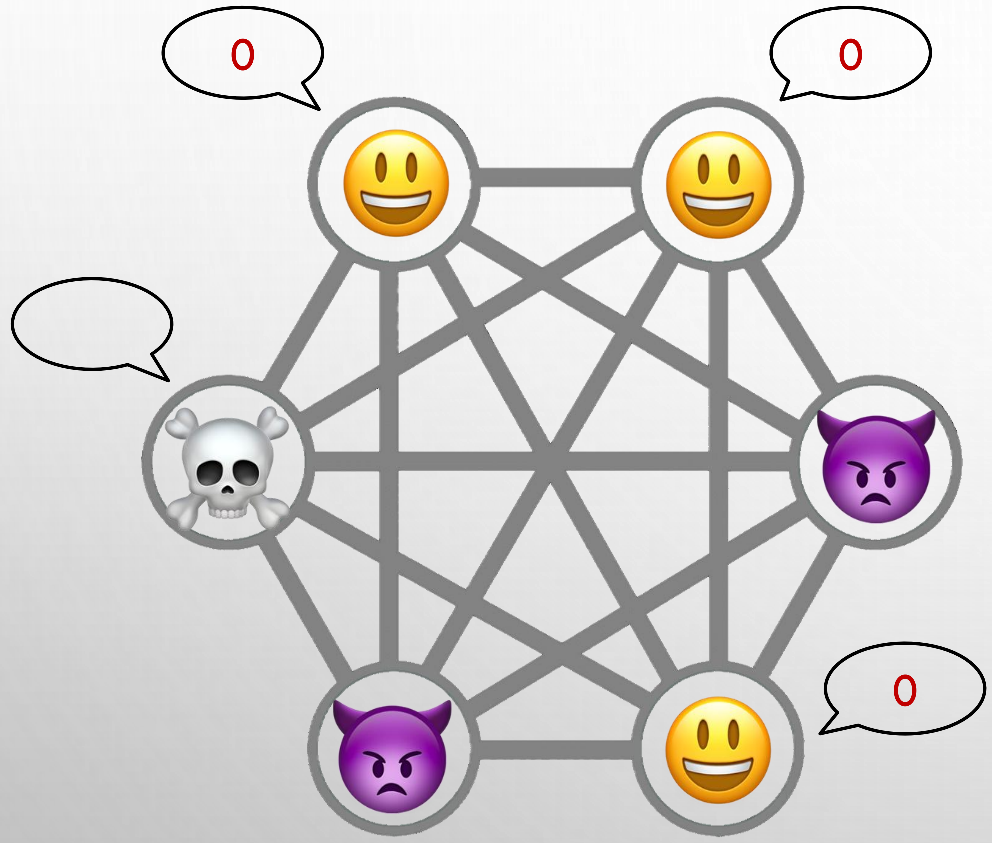


# Byzantine Agreement



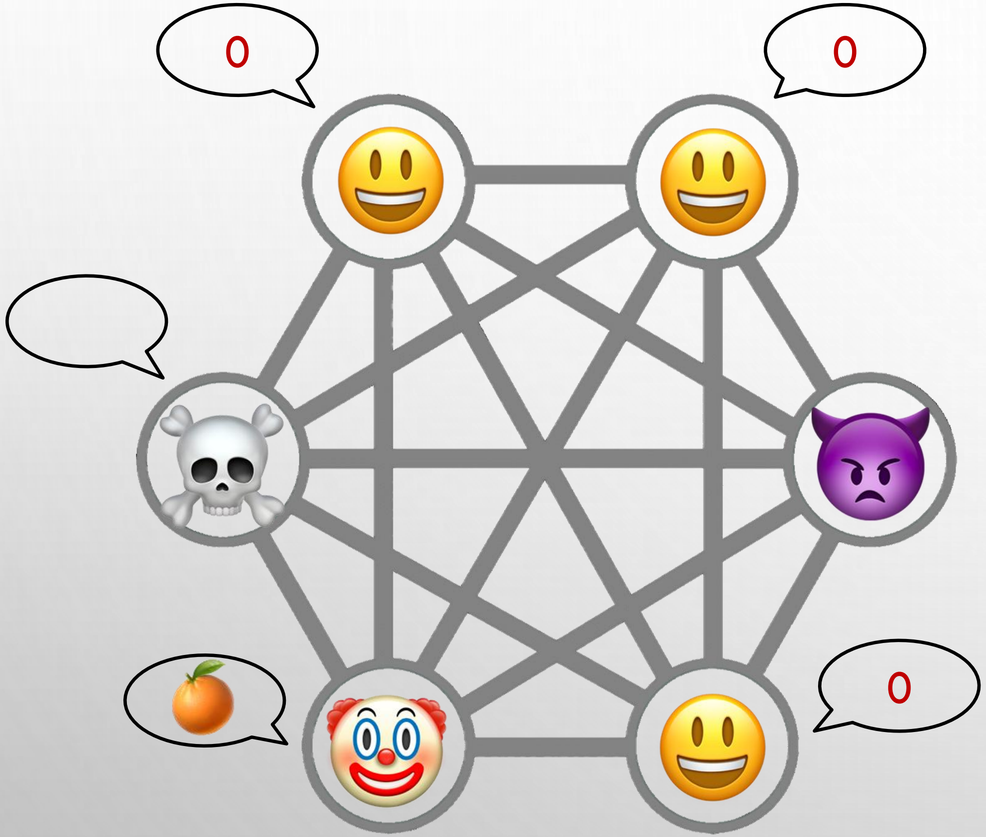

- Up to $t$ parties can be byzantine

# Byzantine Agreement



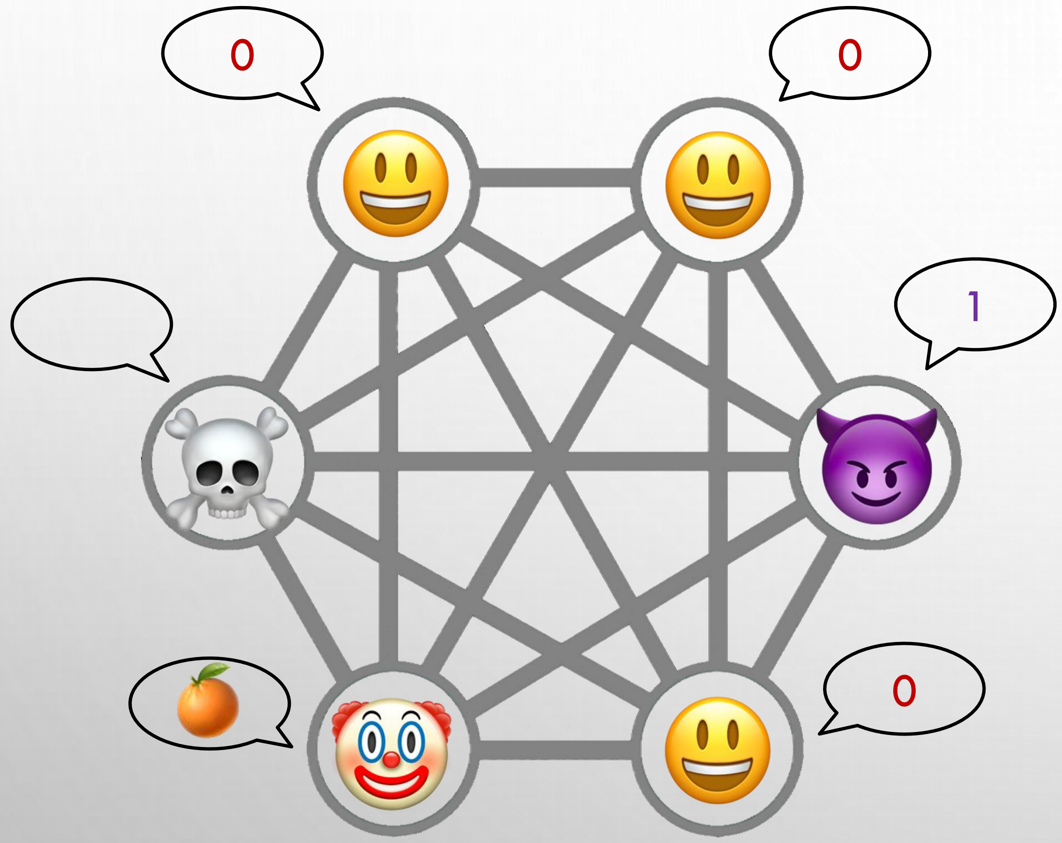

- Up to $t$ parties can be byzantine

# Byzantine Agreement

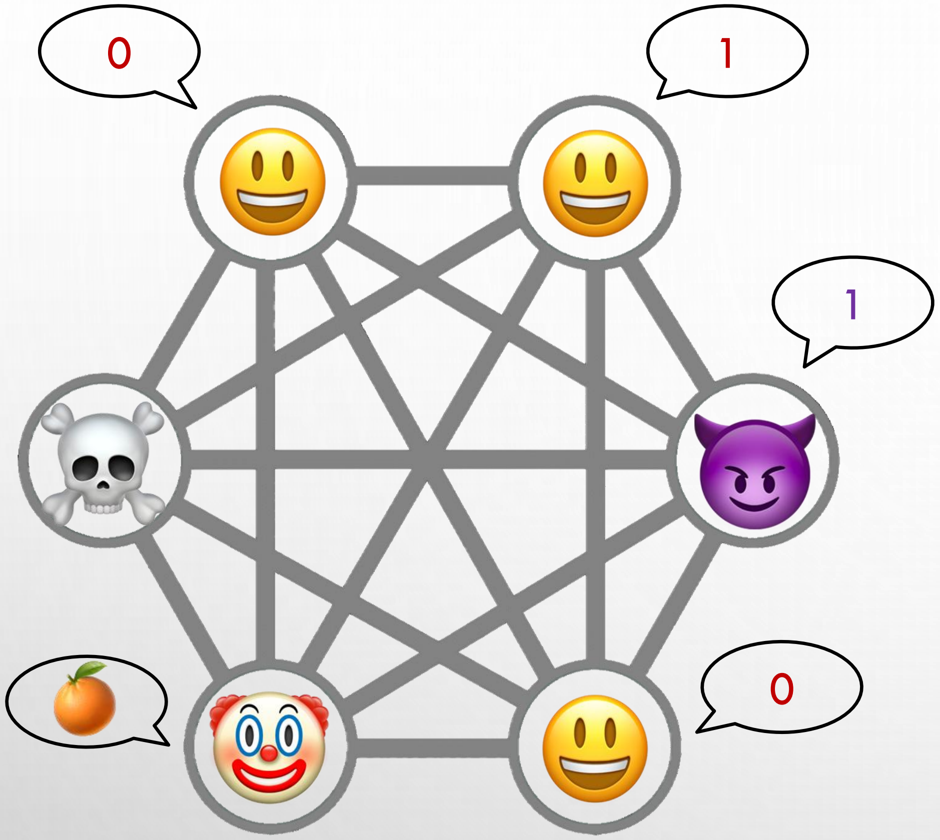


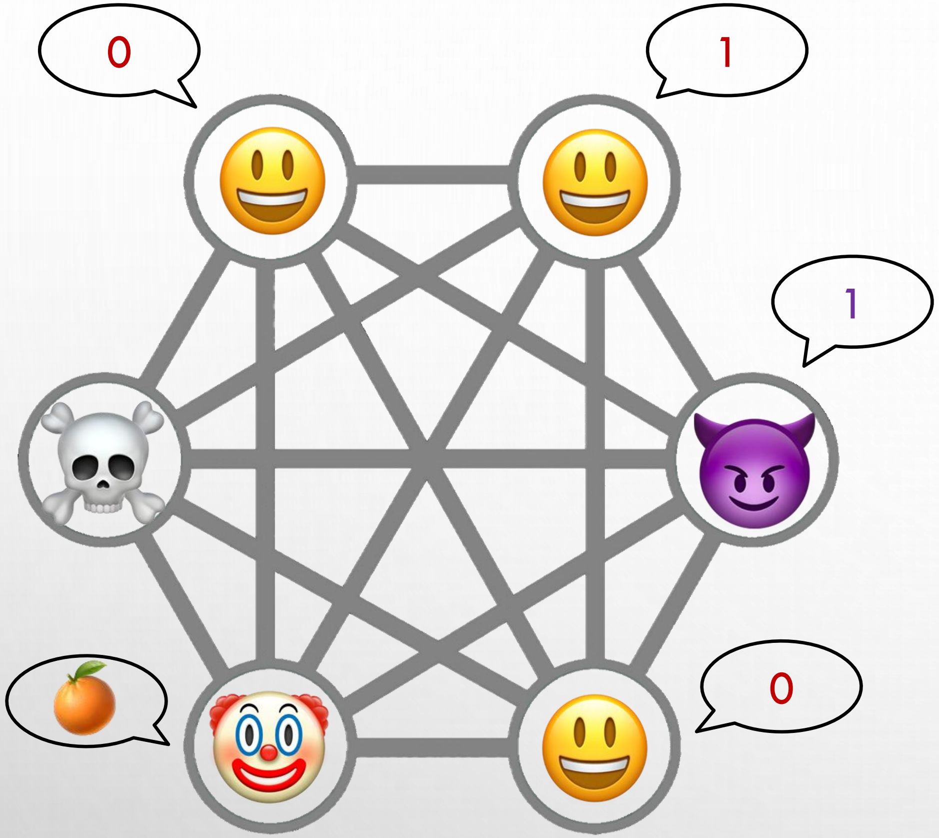

- Up to $t$ parties can be byzantine

# Byzantine Agreement

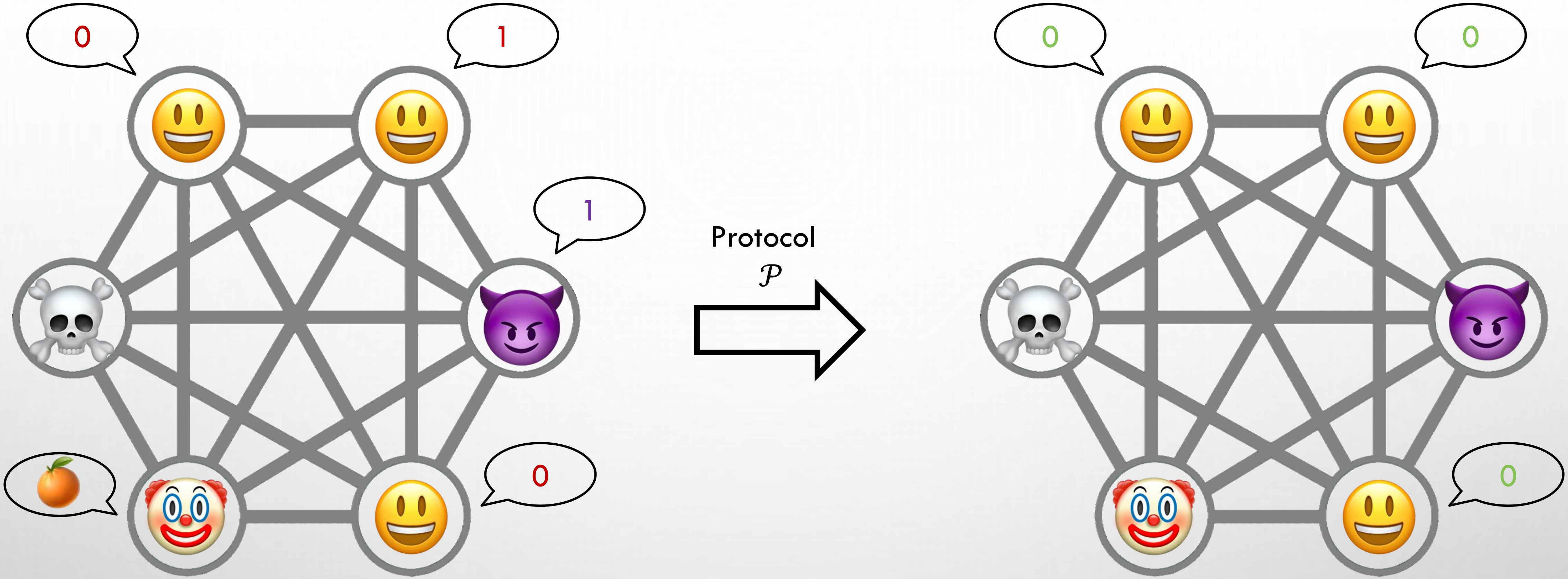


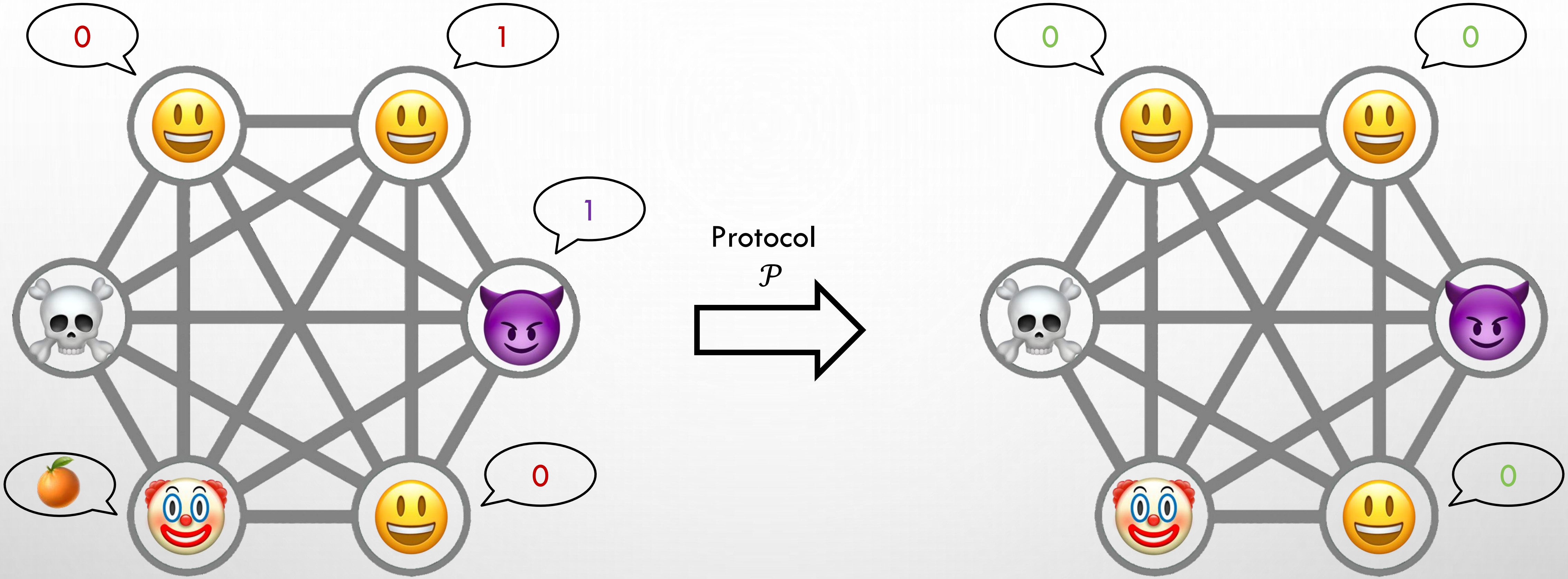

- Up to $t$ parties can be byzantine


0
1
1
0

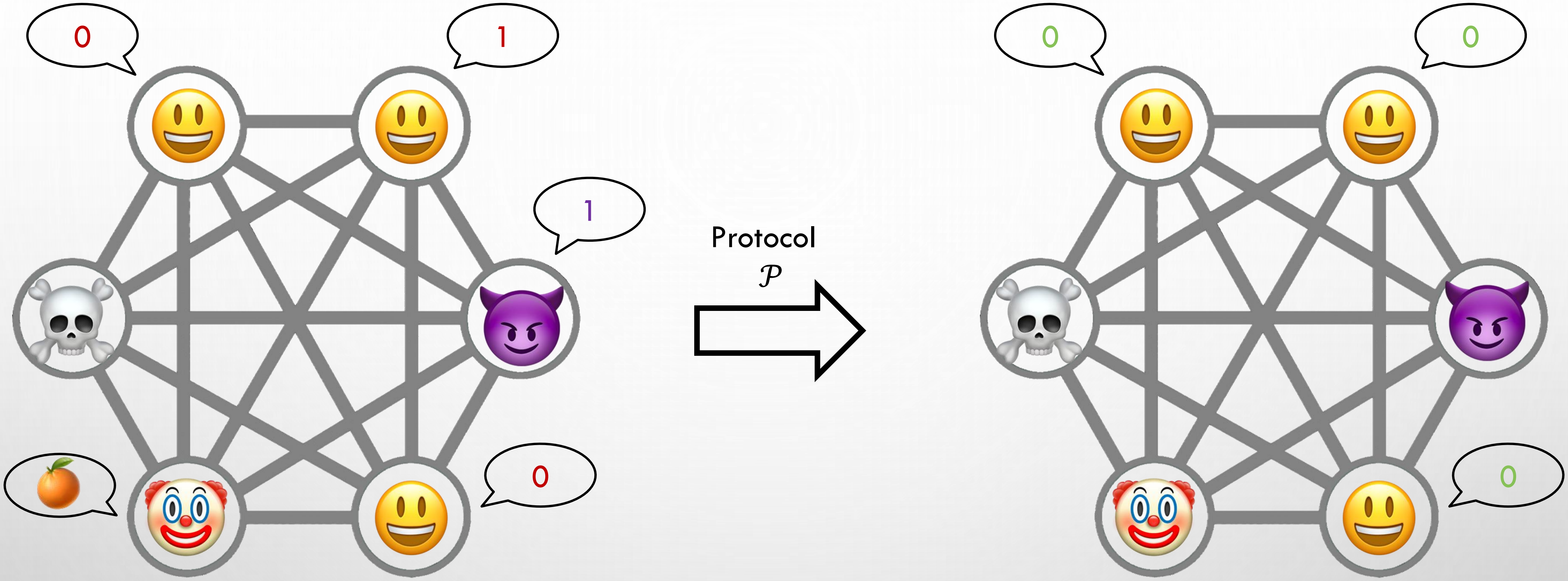



Protocol
$\mathcal{P}$

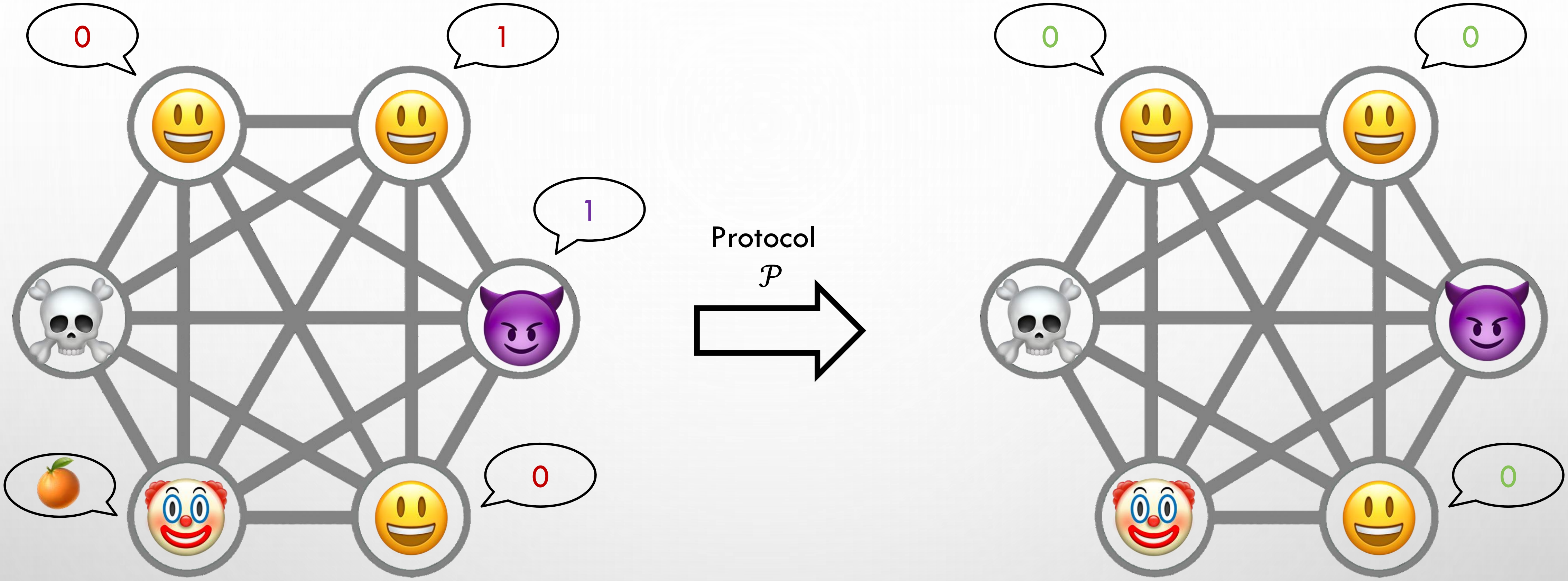

0
1
1
0
Protocol
$\mathcal{P}$
0
0
0




- **Termination**: All honest parties eventually decide a value




- **Termination**: All honest parties eventually decide a value
- **Agreement**: All honest parties decide the same value




- **<u>Termination</u>**: All honest parties eventually decide a value
- **<u>Agreement</u>**: All honest parties decide the same value
- **<u>Validity</u>**: The value decided makes sense

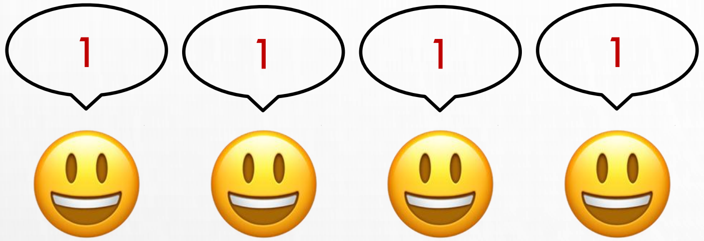
# Validity Conditions

# Validity Conditions

**Weak Validity:**

# Validity Conditions

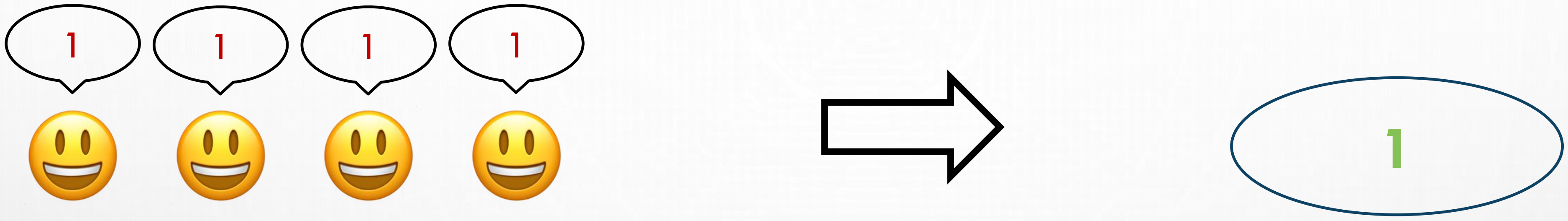
**Weak Validity:**




If all parties are honest and have the same input, they must decide this value.

# Validity Conditions

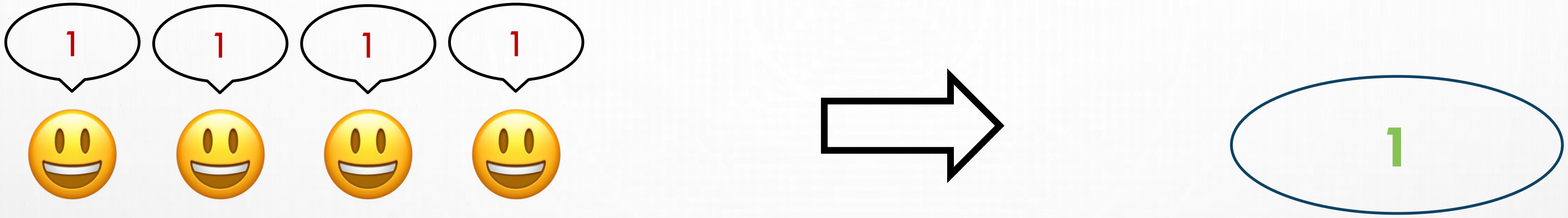
**Weak Validity:**




If all parties are honest and have the same input, they must decide this value.

# Validity Conditions

**Weak Validity:**

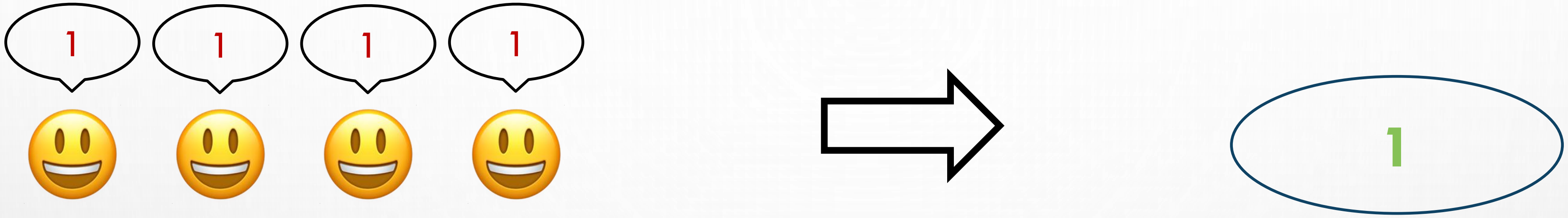



If all parties are honest and have the same input, they must decide this value.

**Strong Validity:**

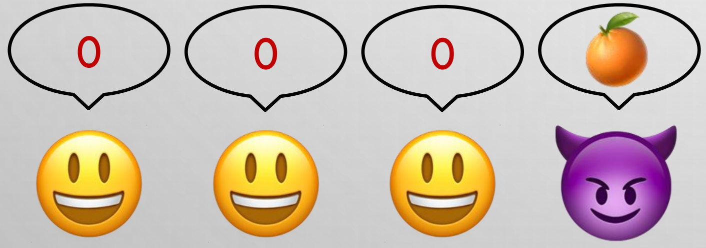
# Validity Conditions

**Weak Validity:**




If all parties are honest and have the same input, they must decide this value.

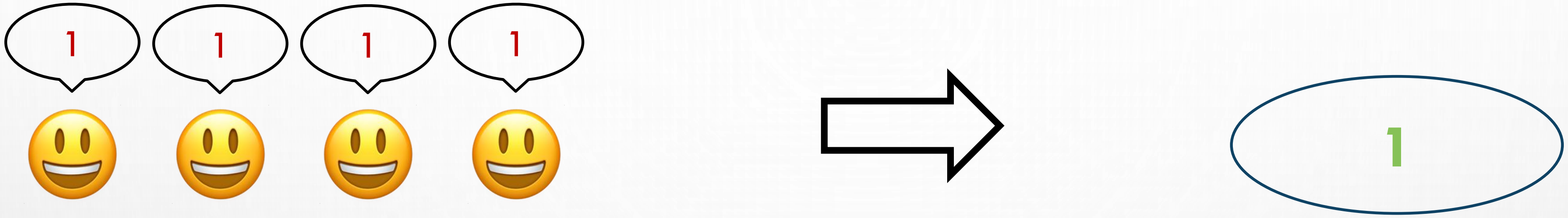
**Strong Validity:**




If all honest parties have the same input, they must decide this value.

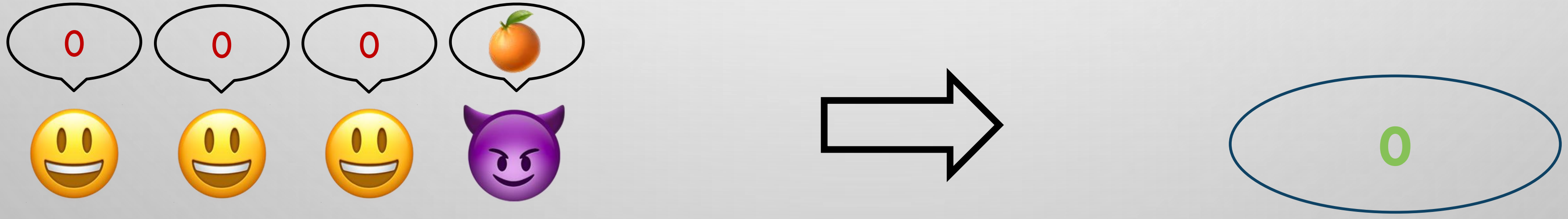
# Validity Conditions

**Weak Validity:**




If all parties are honest and have the same input, they must decide this value.

**Strong Validity:**




If all honest parties have the same input, they must decide this value.

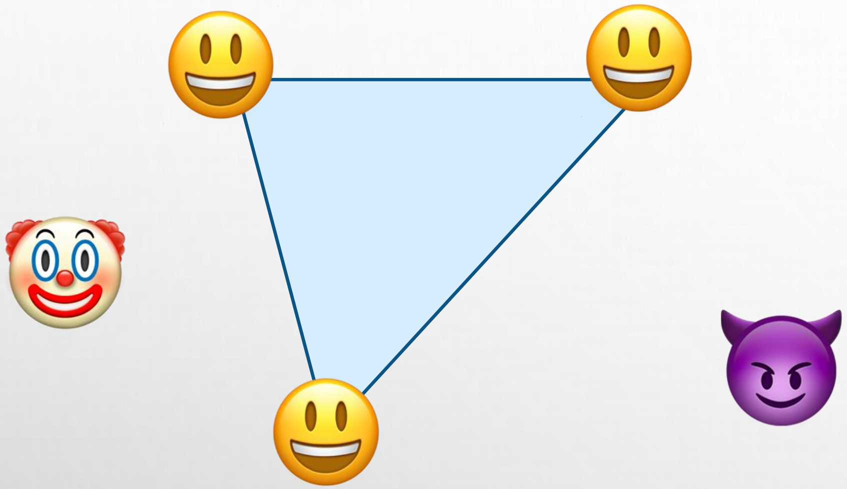
# Validity Conditions

# Validity Conditions

**Convex Validity:**

# Validity Conditions

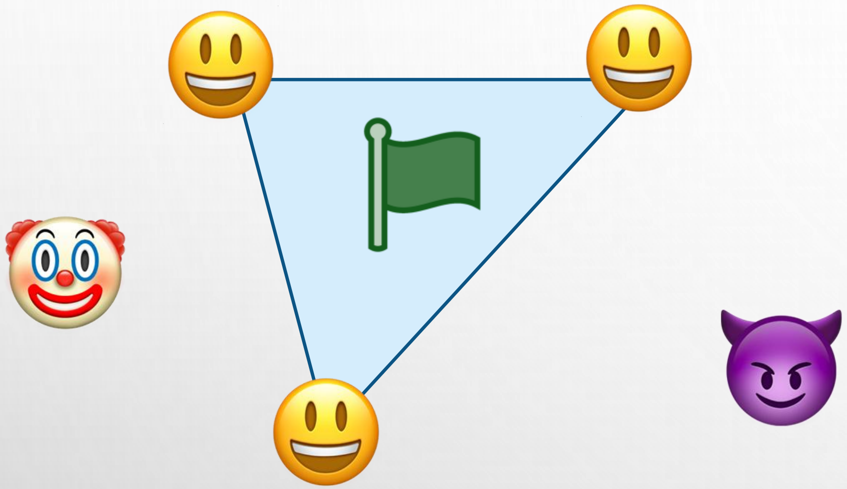
**Convex Validity:**



The output must be in the convex hull of the values of honest parties.

# Validity Conditions

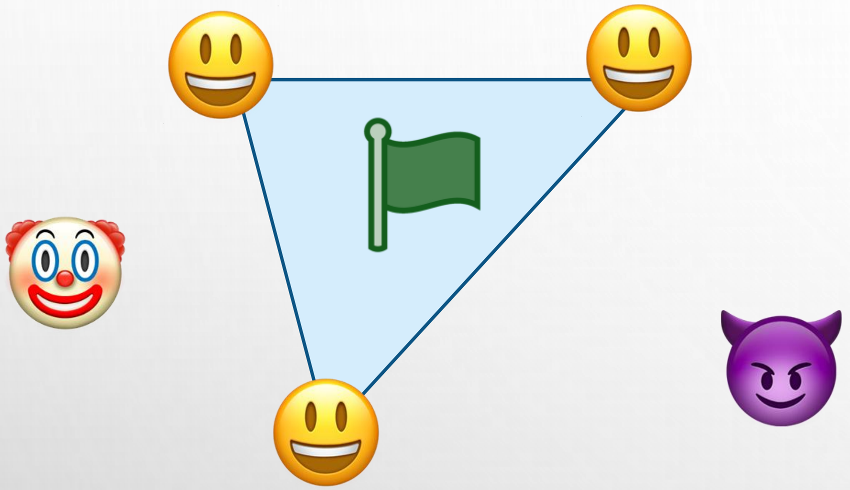
**Convex Validity:**



The output must be in the convex hull of the values of honest parties.

# Validity Conditions

**Convex Validity:**

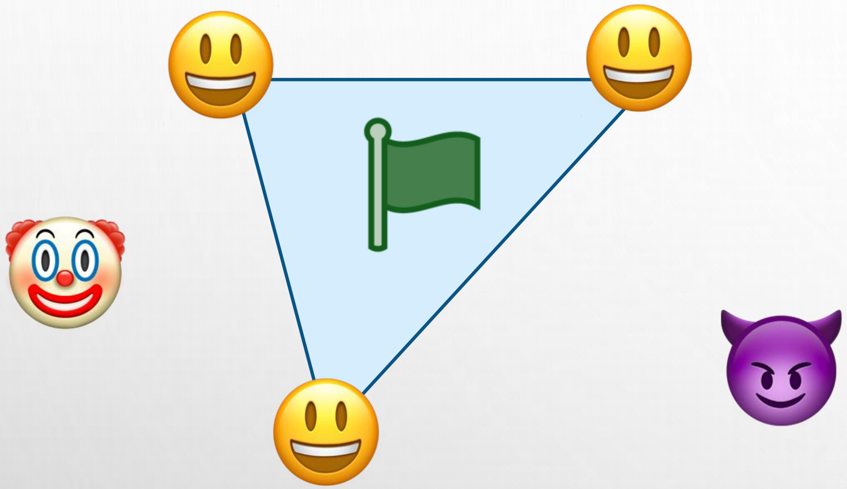


The output must be in the convex hull of the values of honest parties.

**Reliable broadcast:**

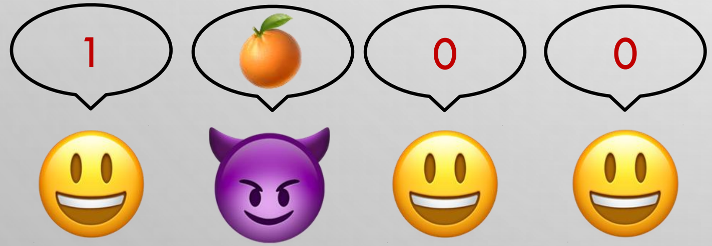
# Validity Conditions

**Convex Validity:**



The output must be in the convex hull of the values of honest parties.

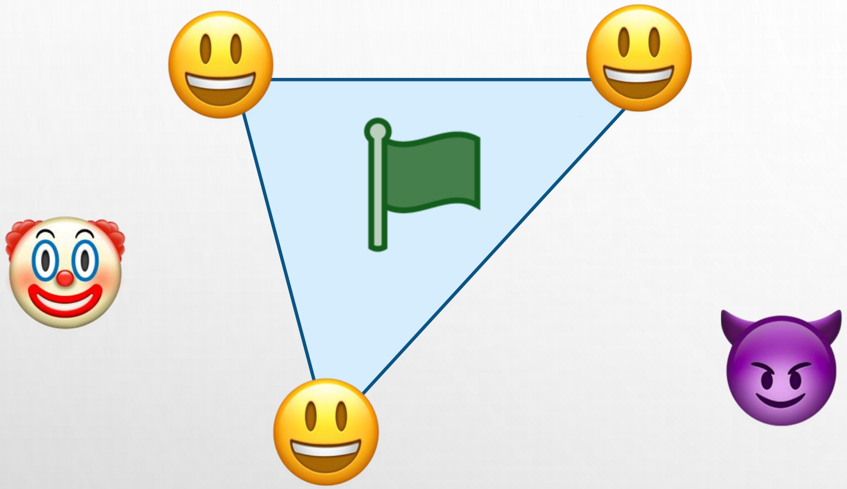
**Reliable broadcast:**




If the first party is honest, its input must be decided.

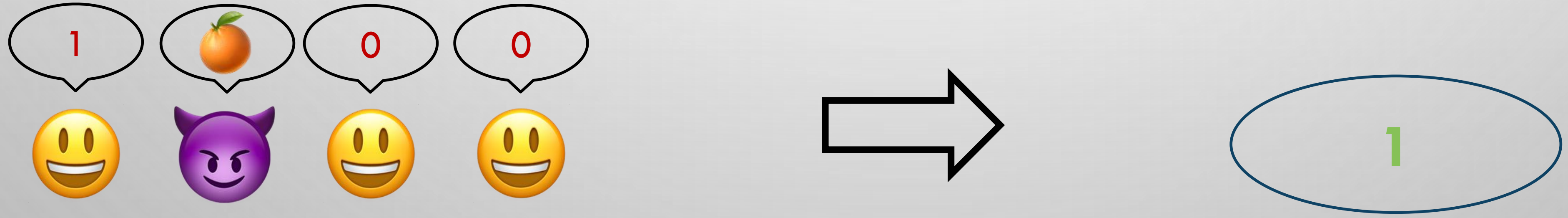
# Validity Conditions

**Convex Validity:**



The output must be in the convex hull of the values of honest parties.

**Reliable broadcast:**




If the first party is honest, its input must be decided.

# Validity Condition

# Validity Condition

$\mathcal{I}$ = "Set of possible input configurations"

# Validity Condition

$\mathcal{I}$ = **"Set of possible input configurations"**

$$\{(p_1, 0), (p_2, 1), (p_4, 0)\} \in \mathcal{I}$$

# Validity Condition

$\mathcal{I} =$ **"Set of possible input configurations"**

$$\{(p_1, 0), (p_2, 1), (p_4, 0)\} \in \mathcal{I}$$

**<u>Validity</u>**: There exists a function val: $\mathcal{I} \mapsto 2^{V_{OUT}}$ such that:

# Validity Condition

$\mathcal{I}$ = **"Set of possible input configurations"**

$$\{(p_1, 0), (p_2, 1), (p_4, 0)\} \in \mathcal{I}$$

**Validity**: There exists a function val: $\mathcal{I} \mapsto 2^{V_{OUT}}$ such that:

$$\text{“Allowed output values”} = \text{val}\left(\textbf{Input Configuration}\right)$$

# Validity Condition

$\mathcal{I}$ = **“Set of possible input configurations”**

$$\{(p_1, 0), (p_2, 1), (p_4, 0)\} \in \mathcal{I}$$

**<u>Validity</u>**: There exists a function val: $\mathcal{I} \mapsto 2^{V_{OUT}}$ such that:

**“Allowed output values”** $=$ val( **Input Configuration** )

A validity property is trivial if there is a value which can always be decided!

# Network Model

# Network Model

<u>**Public Key Infrastructure (PKI):**</u> Messages can be signed.

# Network Model

**Public Key Infrastructure (PKI):** Messages can be signed.

**Synchronous Model**: After a message is sent, it get received within $\Delta$ units of time.

# Network Model

**Public Key Infrastructure (PKI):** Messages can be signed.

**Synchronous Model**: After a message is sent, it get received within $\Delta$ units of time.

**Asynchronous Model**: After a message is sent, it is eventually received.

# Network Model

<u>**Public Key Infrastructure (PKI):**</u> Messages can be signed.

<u>**Synchronous Model**</u>: After a message is sent, it get received within $\Delta$ units of time.

<u>**Asynchronous Model**</u>: After a message is sent, it is eventually received.

<u>**Network-Agnostic Model**</u>: (Best of both world)
The network is either:

# Network Model

**Public Key Infrastructure (PKI):** Messages can be signed.

**Synchronous Model**: After a message is sent, it get received within $\Delta$ units of time.

**Asynchronous Model**: After a message is sent, it is eventually received.

**Network-Agnostic Model**: (Best of both world)

The network is either:

- Synchronous with up to $t_S$ byzantine parties

# Network Model

**<u>Public Key Infrastructure (PKI):</u>** Messages can be signed.

**<u>Synchronous Model</u>**: After a message is sent, it get received within $\Delta$ units of time.

**<u>Asynchronous Model</u>**: After a message is sent, it is eventually received.

**<u>Network-Agnostic Model</u>**: (Best of both world)
The network is either:
- Synchronous with up to $t_S$ byzantine parties
- Asynchronous with up to $t_A$ byzantine parties ($t_A \leq t_S$)

# Protocols

# Protocols

**Deterministic protocols?**

# Protocols

**Deterministic protocols?**

**FLP:** No deterministic protocol can solve agreement in asynchrony!

# Protocols

~~**Deterministic protocols?**~~ **Randomized protocol!**

**FLP:** No deterministic protocol can solve agreement in asynchrony!

# Protocols

~~**Deterministic protocols?**~~ **Randomized protocol!**

**FLP:** No deterministic protocol can solve agreement in asynchrony!

**Probabilistic termination:** All honest parties eventually decide a value almost surely.

# Protocols

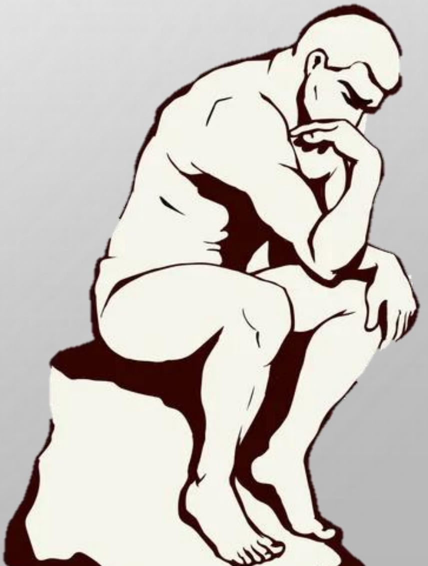
~~**Deterministic protocols?**~~ **Randomized protocol!**

**FLP:** No deterministic protocol can solve agreement in asynchrony!

**Probabilistic termination:** All honest parties eventually decide a value almost surely.

We need to assume $V_{IN}$ or $V_{OUT}$ is not uncountable !

## What are tight conditions for a given validity property to be solvable in the network-agnostic setting?



# Previous Works

# Previous Works

- **Partial synchrony:**
  [Civit, Gilbert, Guerraoui, Komatovic, Vidigueira PODC'23]

# Previous Works

- **Partial synchrony:**
  [Civit, Gilbert, Guerraoui, Komatovic, Vidigueira PODC'23]

- **Synchrony:**
  [Civit, Gilbert, Guerraoui, Komatovic, Paramonov, Vidigueira PODC'24]

# Previous Works

- **Partial synchrony:**
  [Civit, Gilbert, Guerraoui, Komatovic, Vidigueira PODC'23]

- **Synchrony:**
  [Civit, Gilbert, Guerraoui, Komatovic, Paramonov, Vidigueira PODC'24]

Deterministic!

# Previous Works

- **Partial synchrony:**
  [Civit, Gilbert, Guerraoui, Komatovic, Vidigueira PODC'23]

- **Synchrony:**
  [Civit, Gilbert, Guerraoui, Komatovic, Paramonov, Vidigueira PODC'24]

Deterministic!

- **Network Agnostic:** This work

# Synchronous setting

# Synchronous setting

**Theorem**: Without PKI, if a validity property is solvable, then:

$$t_S < n/3$$

# Network agnostic setting

# Network agnostic setting

**Theorem**: If a validity property is solvable, then:

$$2t_S + t_A < n$$

# Necessary conditions

- $2t_S + t_{\mathrm{A}} < n$
- Without PKI, $t_S < n/3$

# Necessary conditions

$$\begin{cases} \bullet\ 2t_S + t_{\mathrm{A}} < n \\ \bullet\ \text{Without PKI, } t_S < n/3 \end{cases}$$

Necessary and sufficient for weak validity, strong validity !

# Necessary conditions

- $2t_S + t_{\mathrm{A}} < n$
- Without PKI, $t_S < n/3$

Necessary and sufficient for weak validity, strong validity !

**Is it enough?**

# Necessary conditions

- $2t_S + t_\mathrm{A} < n$
- Without PKI, $t_S < n/3$

Necessary and sufficient for weak validity, strong validity !

**Is it enough?**

No:

# Necessary conditions

$$\begin{cases} \bullet\ 2t_S + t_{\mathrm{A}} < n \\ \bullet\ \text{Without PKI, } t_S < n/3 \end{cases}$$

Necessary and sufficient for weak validity, strong validity !

**Is it enough?**

No:

- Reliable broadcast: Not solvable if $t_S > 0$.

# Necessary conditions

$$\begin{cases} \bullet\ 2t_S + t_A < n \\ \bullet\ \text{Without PKI, } t_S < n/3 \end{cases}$$

Necessary and sufficient for weak validity, strong validity !

**Is it enough?**

No:

- Reliable broadcast: Not solvable if $t_S > 0$.
- Convex validity: Solvable only if $n > \max(\omega \cdot t_S, \omega \cdot t_A + t_S)$

[Constantinescu, Ghinea, Wattenhofer, Westermann DISC'24]

# Necessary conditions

# Necessary conditions

Assume $\begin{cases} \bullet\ 2t_S + t_{\mathrm{A}} < n \\ \bullet\ \text{Without PKI, } t_S < n/3 \end{cases}$

# Necessary conditions

Assume $\begin{cases} \bullet\ 2t_S + t_{\mathrm{A}} < n \\ \bullet\ \text{Without PKI, } t_S < n/3 \end{cases}$

We can get a common view!

# Necessary conditions

Assume
- $2t_S + t_\mathrm{A} < n$
- Without PKI, $t_S < n/3$

We can get a common view!

How far away is it from the input configuration?

# Necessary conditions

Assume
- $2t_S + t_A < n$
- Without PKI, $t_S < n/3$

We can get a common view!

How far away is it from the input configuration?

- **Synchrony**: A byzantine party may act as honest with an arbitrary input value.

# Necessary conditions

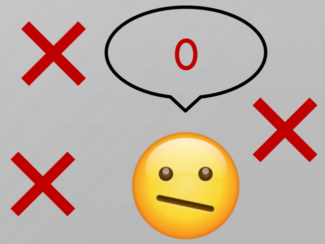
Assume
- $2t_S + t_A < n$
- Without PKI, $t_S < n/3$

We can get a common view!

How far away is it from the input configuration?

- **<u>Synchrony</u>**: A byzantine party may act as honest with an arbitrary input value.


- **<u>Asynchrony</u>**: Messages from an honest party may be delayed until after the common view is established.



# Similarity condition

# Similarity condition

$\mathrm{SIMILAR}(I)$ = “ Input configurations which can result in common view $I$ ”

# Similarity condition

$\mathrm{SIMILAR}(I)$ = " Input configurations which can result in common view $I$ "

$\mathrm{SIMILAR}(I)$

# Similarity condition

$\mathrm{SIMILAR}(I)$ = " Input configurations which can result in common view $I$ "

$$\mathrm{SIMILAR}(I) = \underbrace{\{J \in \mathrm{NEIGHBORS}(I), J \subseteq I\}}_{\text{Synchronous}}$$

# Similarity condition

$\mathrm{SIMILAR}(I) =$ “ Input configurations which can result in common view $I$ ”

$$\mathrm{SIMILAR}(I) = \underbrace{\{J \in \mathrm{NEIGHBORS}(I), J \subseteq I\}}_{\text{Synchronous}} \cup \underbrace{\{J \in \mathrm{NEIGHBORS}(I), |J| \geq n - t_A\}}_{\text{Asynchronous}}$$

# Similarity condition

$\mathrm{SIMILAR}(I) =$ " Input configurations which can result in common view $I$ "

$$\mathrm{SIMILAR}(I) = \underbrace{\{J \in \mathrm{NEIGHBORS}(I), J \subseteq I\}}_{\text{Synchronous}} \cup \underbrace{\{J \in \mathrm{NEIGHBORS}(I), |J| \geq n - t_A\}}_{\text{Asynchronous}}$$

Deciding a value in $\bigcap_{J \in \mathrm{SIMILAR}(\mathcal{J})} \mathrm{val}(J)$ satisfies the validity property!

# Similarity condition

# Similarity condition

A validity property is solvable in the network agnostic setting if and only if:

- $2t_S + t_{\mathrm{A}} < n$
- Without PKI, $t_S < n/3$

# Similarity condition

A validity property is solvable in the network agnostic setting if and only if:

$$\begin{cases} \bullet\ 2t_S + t_{\mathrm{A}} < n \\ \bullet\ \text{Without PKI, } t_S < n/3 \end{cases}$$

- There exists a turing-computable function $\sigma\colon \mathcal{I} \mapsto \mathrm{V_{OUT}}$ such that for all $I \in \mathcal{I}$:

$$\sigma(I) \in \bigcap_{J \in SIMILAR(I)} \mathrm{val}(J)$$

Protocol $\mathcal{P}$

Turing-computable function

$\sigma$

Protocol $\mathcal{P}$

Run offline

Turing-computable function

$\sigma$

# Protocol $\mathcal{P}$

Run offline

Probabilistic
Turing-computable
Function $\sigma_{prob}$

Turing-computable function
σ

Protocol $\mathcal{P}$

Run offline

Probabilistic
Turing-computable
Function $\sigma_{prob}$

Turn deterministic

Turing-computable function
$\sigma$

# Protocol $\mathcal{P}$

Run offline

Probabilistic
Turing-computable
Function $\sigma_{prob}$

Turn deterministic

Use Agreement on Core Set

# Turing-computable function
# σ

# Protocol $\mathcal{P}$

Run offline

Probabilistic
Turing-computable
Function $\sigma_{prob}$

Turn deterministic

Common view $I$

Use Agreement on Core Set

# Turing-computable function
# σ

# Protocol $\mathcal{P}$

Run offline

Probabilistic
Turing-computable
Function $\sigma_{prob}$

Turn deterministic

Run σ locally on the common view
Decide $\sigma(I)$

Common view $I$

Use Agreement on Core Set

# Turing-computable function
# σ

# Conclusion

# Conclusion

- Necessary and sufficient conditions for validity conditions in network-agnostic byzantine agreement.

# Conclusion

- Necessary and sufficient conditions for validity conditions in network-agnostic byzantine agreement.
- Extend to non-zero probability of failure.

# Conclusion

- Necessary and sufficient conditions for validity conditions in network-agnostic byzantine agreement.
- Extend to non-zero probability of failure.
- Extend to weaker agreement problems.

# QUESTIONS?