Lotto: Secure Participant Selection against Adversarial Servers in Federated Learning

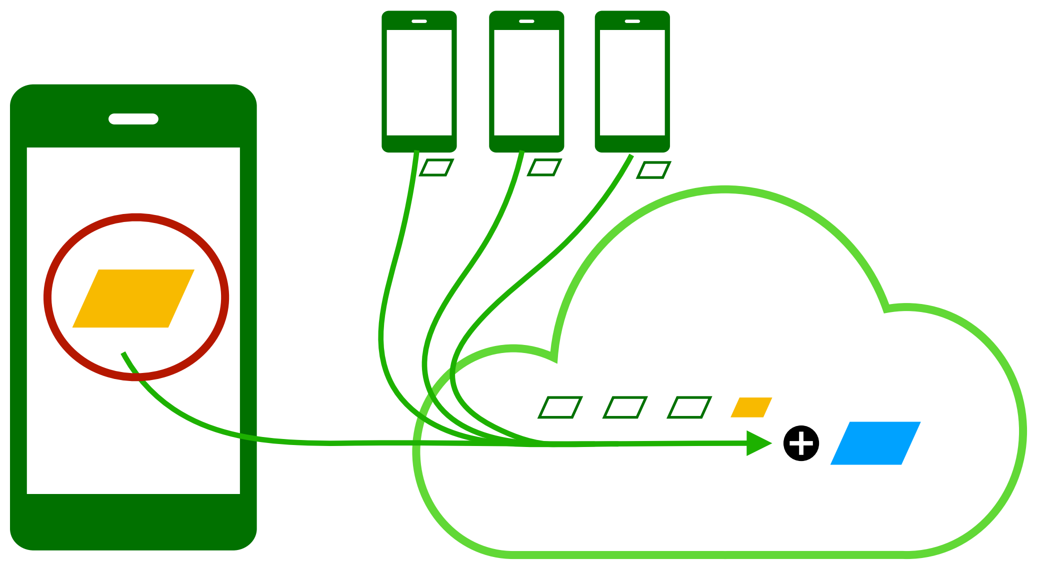
Zhifeng Jiang, Peng Ye, Shiqi He, Wei Wang, Ruichuan Chen, Bo Li

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

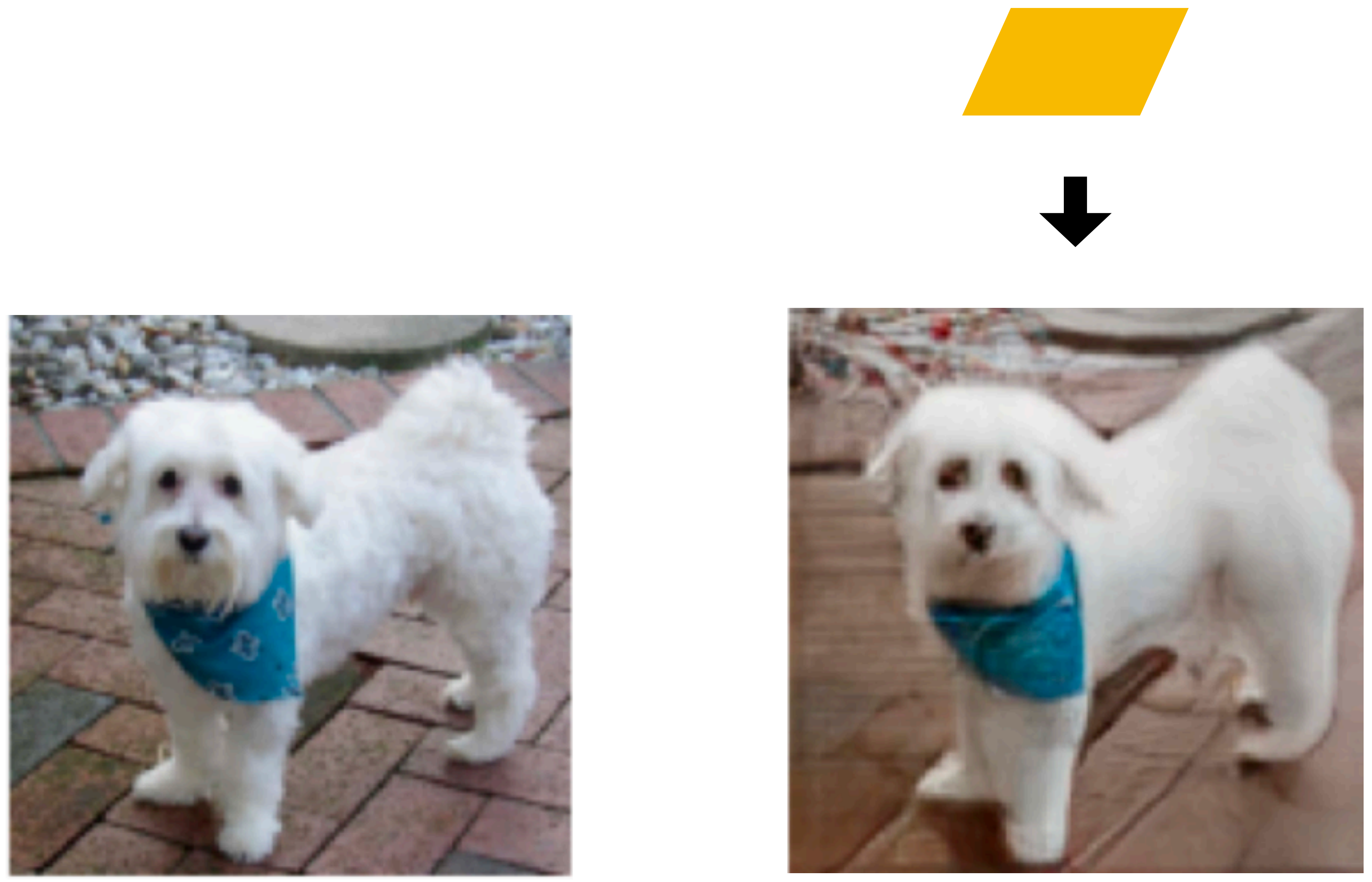
UNIVERSITY OF MICHIGAN

NOKIA BELL LABS

Private learning on the edge



Privacy-Enhancing Technique	Federated Learning[1]
Privacy Guarantee	Data kept on premises

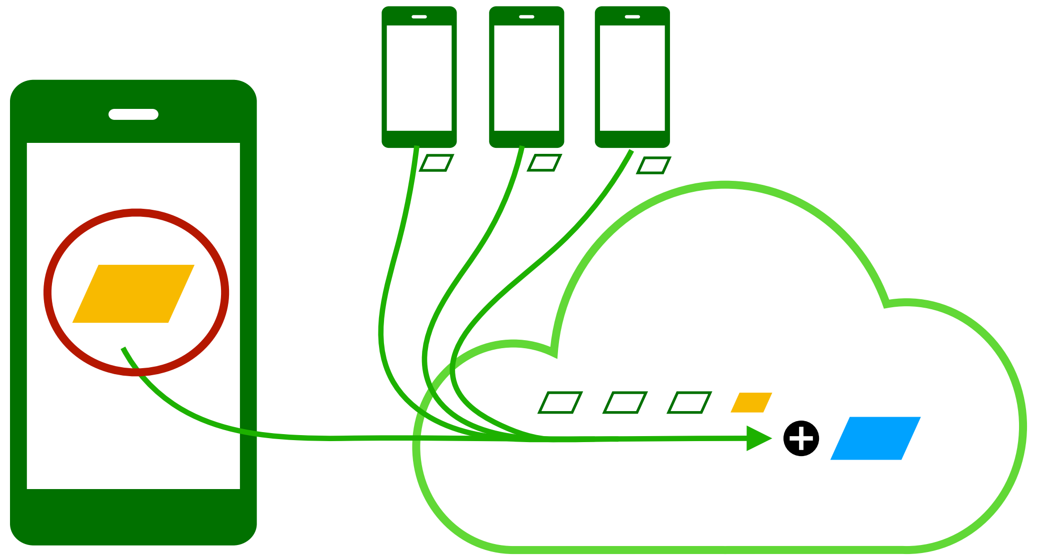


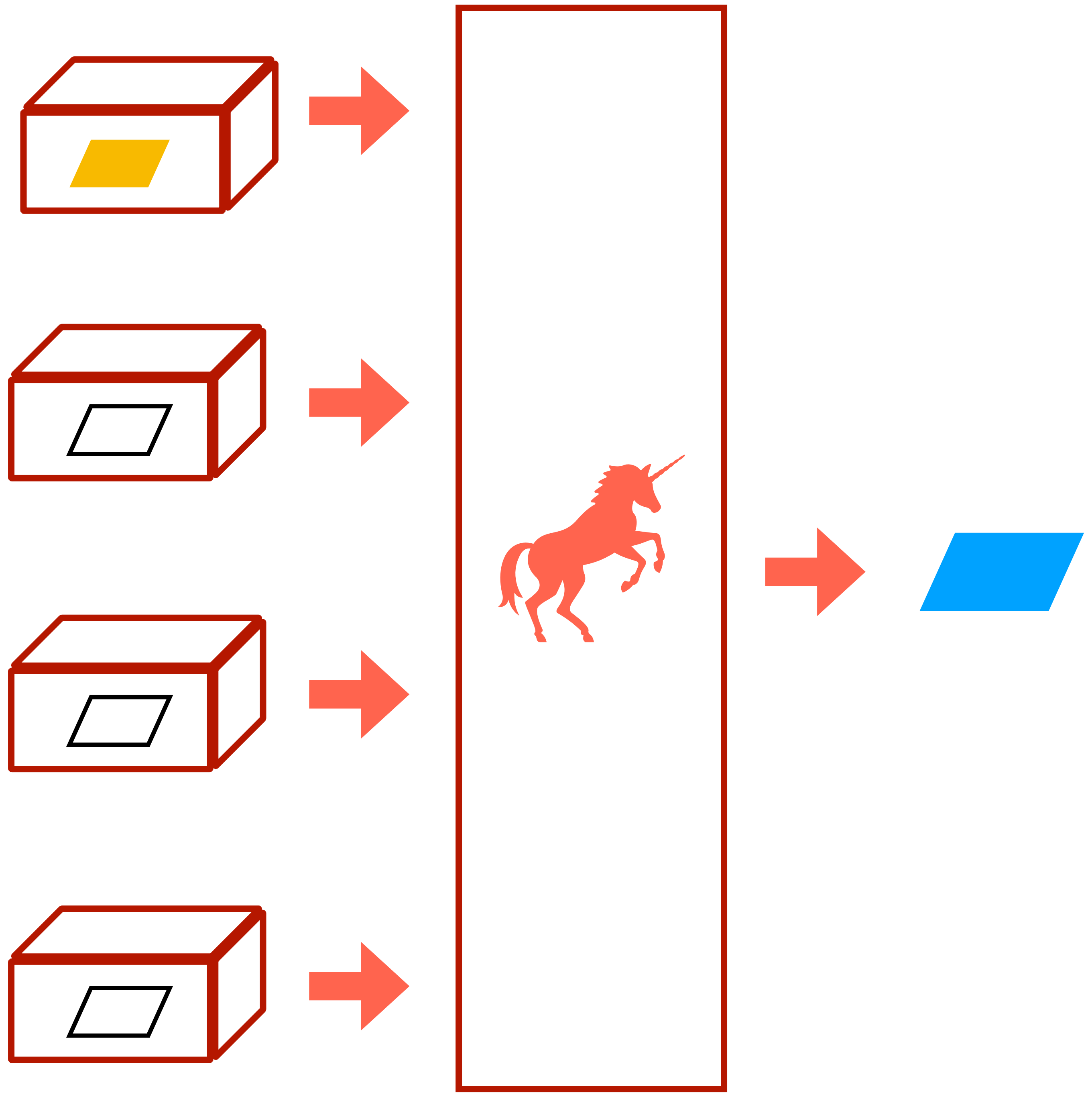
Ground truth   Reconstructed

Problem: Data can be reconstructed from **local model updates**[2]

[1]McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data", In AISTATS '17
[2]Yue et al. "Gradient Obfuscation Gives a False Sense of Security in Federated Learning", In Security '23

Private learning on the edge

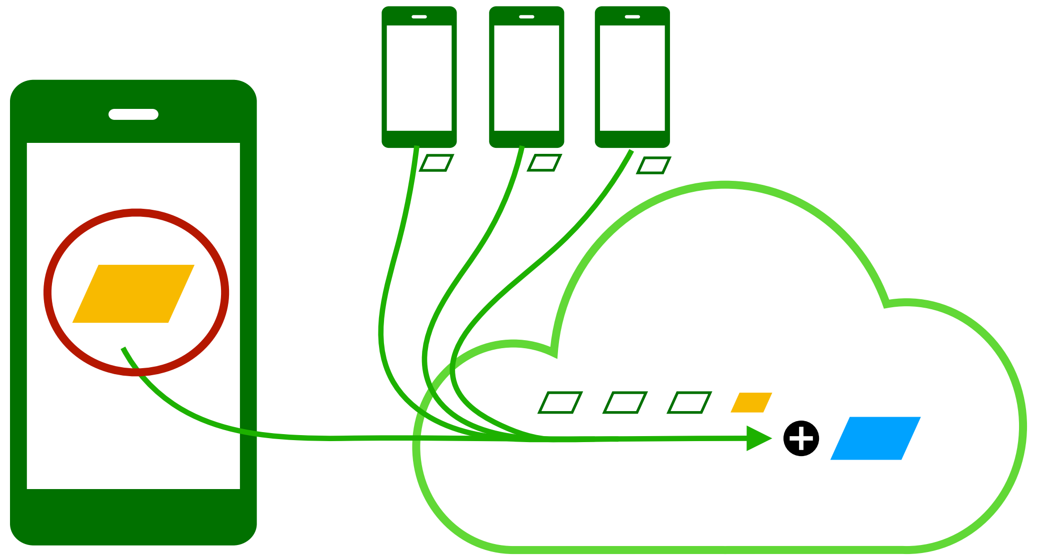


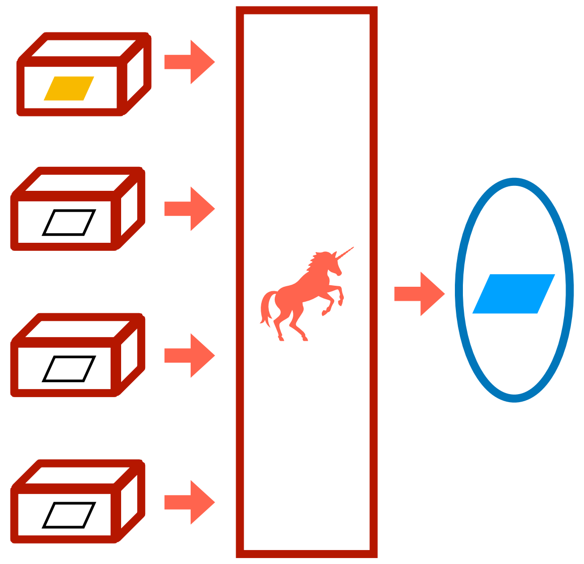


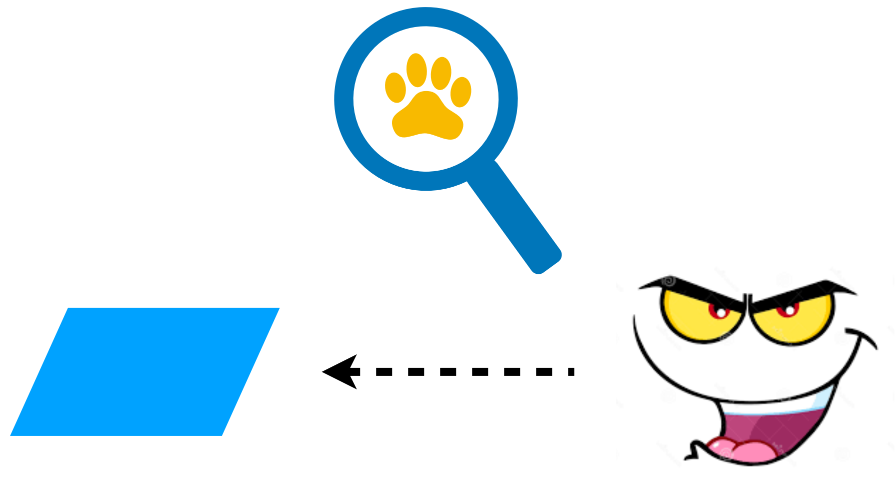
Privacy-Enhancing Technique	Federated Learning[1]	Secure Aggregation[3,4]
Privacy Guarantee	Data kept on premises	Local updates unseen

[1]McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data", In AISTATS '17
[2]Yue et al. "Gradient Obfuscation Gives a False Sense of Security in Federated Learning", In Security '23
[3]Bonawitz et al. "Practical Secure Aggregation for Privacy-Preserving Machine Learning", In CCS '17
[4]Bell et al. "Secure Single-Server Aggregation with (Poly) Logarithmic Overhead", In CCS '20

Private learning on the edge







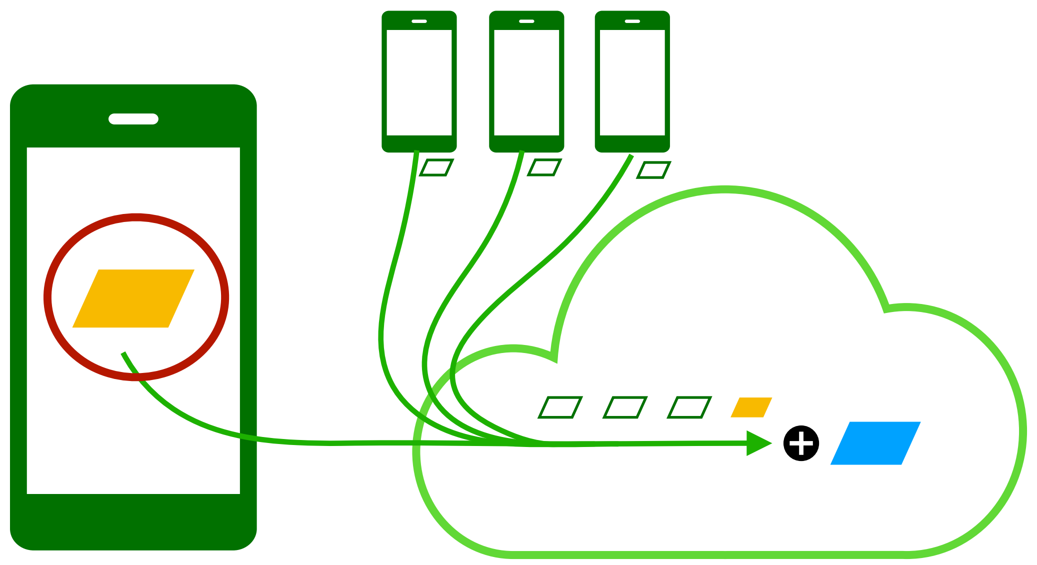
Privacy-Enhancing Technique	Federated Learning[1]	Secure Aggregation[3,4]
Privacy Guarantee	Data kept on premises	Local updates unseen

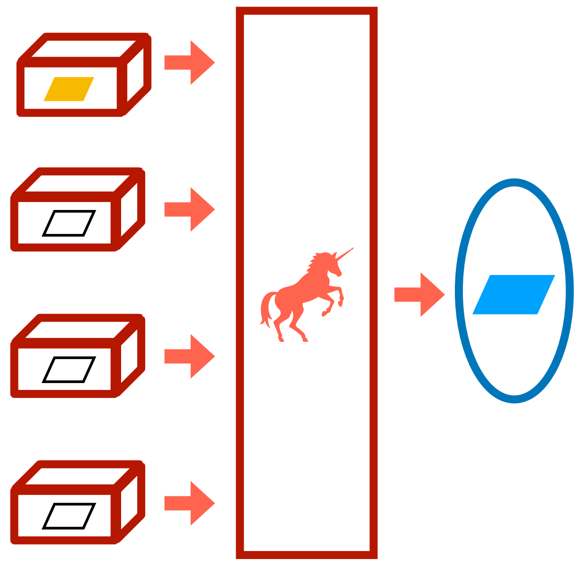
Problem: Data still has footprints in **global model update**[5]

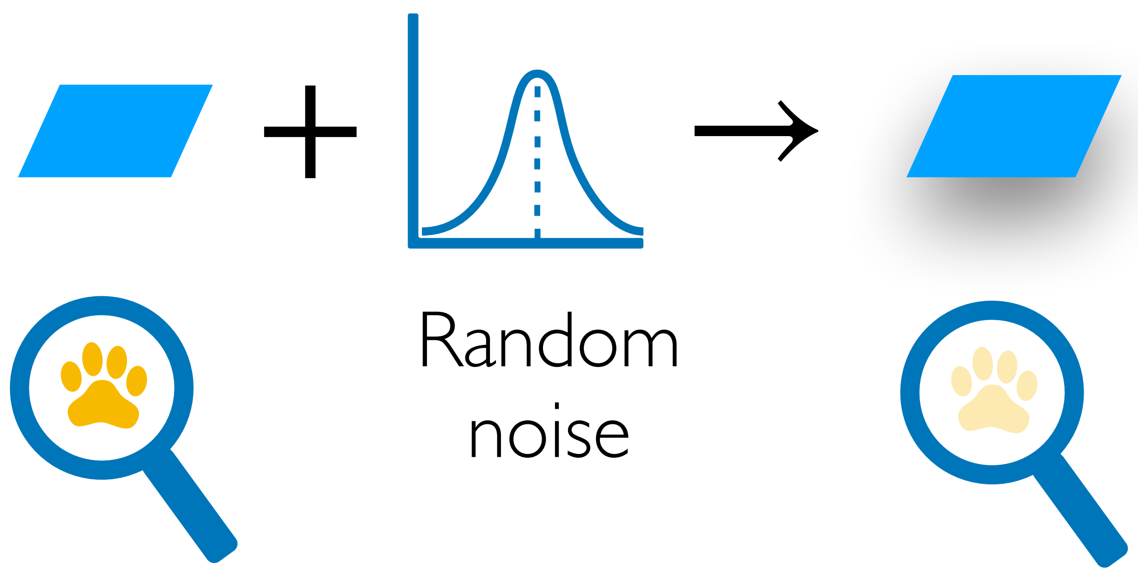
[1]McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data", In AISTATS '17
[2]Yue et al. "Gradient Obfuscation Gives a False Sense of Security in Federated Learning", In Security '23
[3]Bonawitz et al. "Practical Secure Aggregation for Privacy-Preserving Machine Learning", In CCS '17
[4]Bell et al. "Secure Single-Server Aggregation with (Poly) Logarithmic Overhead", In CCS '20

[5]Nasr et al. "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning", In S&P '19

Private learning on the edge





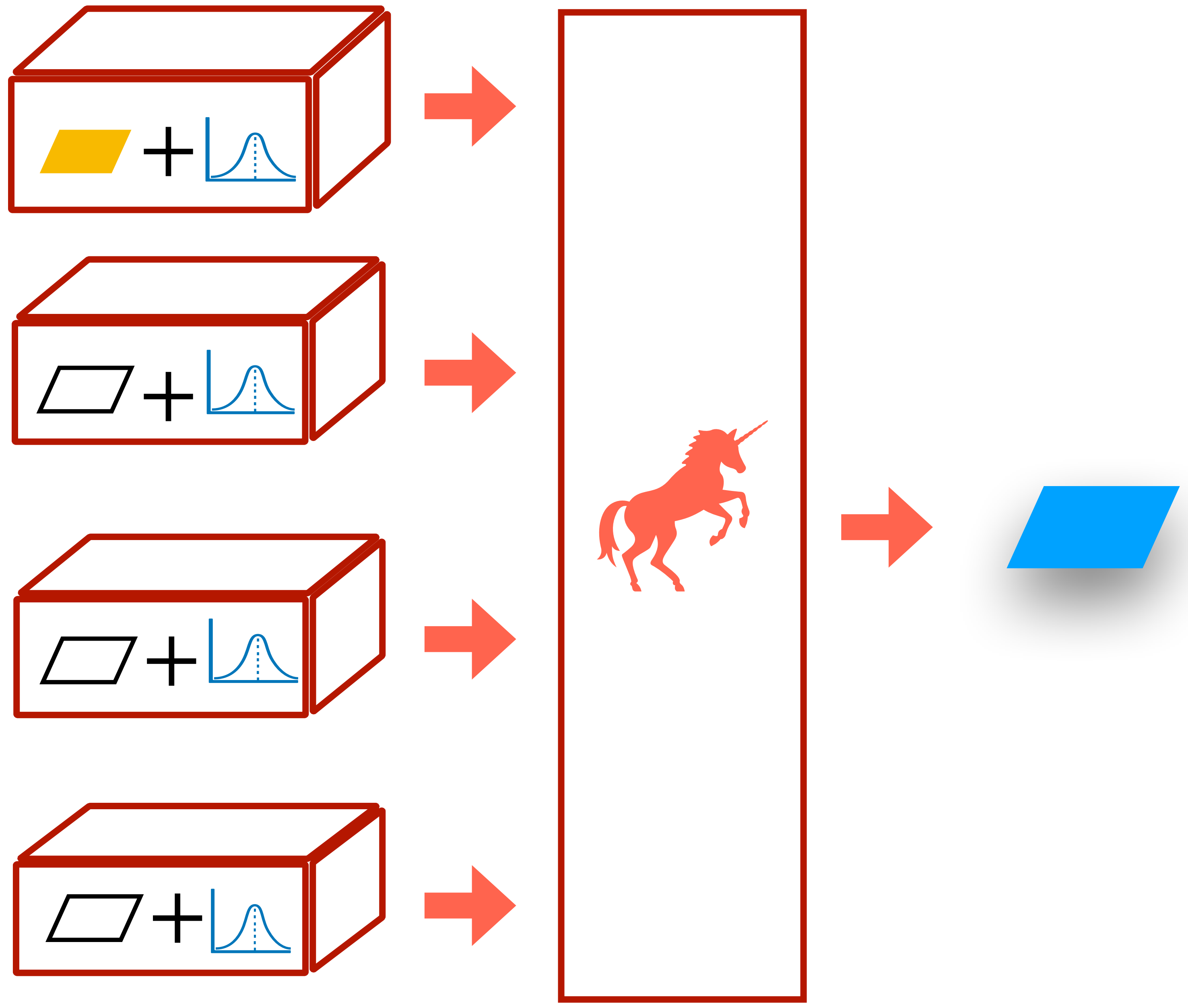


Privacy-Enhancing Technique	Federated Learning[1]	Secure Aggregation[3,4]	Differential Privacy[6]
Privacy Guarantee	Data kept on premises	Local updates unseen	Global update leaks little about any client

[1]McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data", In AISTATS '17
[2]Yue et al. "Gradient Obfuscation Gives a False Sense of Security in Federated Learning", In Security '23
[3]Bonawitz et al. "Practical Secure Aggregation for Privacy-Preserving Machine Learning", In CCS '17
[4]Bell et al. "Secure Single-Server Aggregation with (Poly) Logarithmic Overhead", In CCS '20

[5]Nasr et al. "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning", In S&P '19
[6]Cynthia. "Differential Privacy", 06.

Private learning on the edge



Combined

Secure Aggregation	Differential Privacy
Local updates unseen	Global update leaks little about any client

Practice[1,2]: = 1/4

Each client adds an **even share** of the target noise to its local model update

[1] Kairouz et al. "The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation", In ICML '21
[2] Agarwal. "The Skellam Mechanism for Differentially Private Federated Learning", In NeurIPS '21

Private learning on the edge

Privacy-Enhancing Technique	Federated Learning[1]	Secure Aggregation	Differential Privacy
Privacy Guarantee	Data kept on premises	Local updates unseen	Global update leaks little about any client

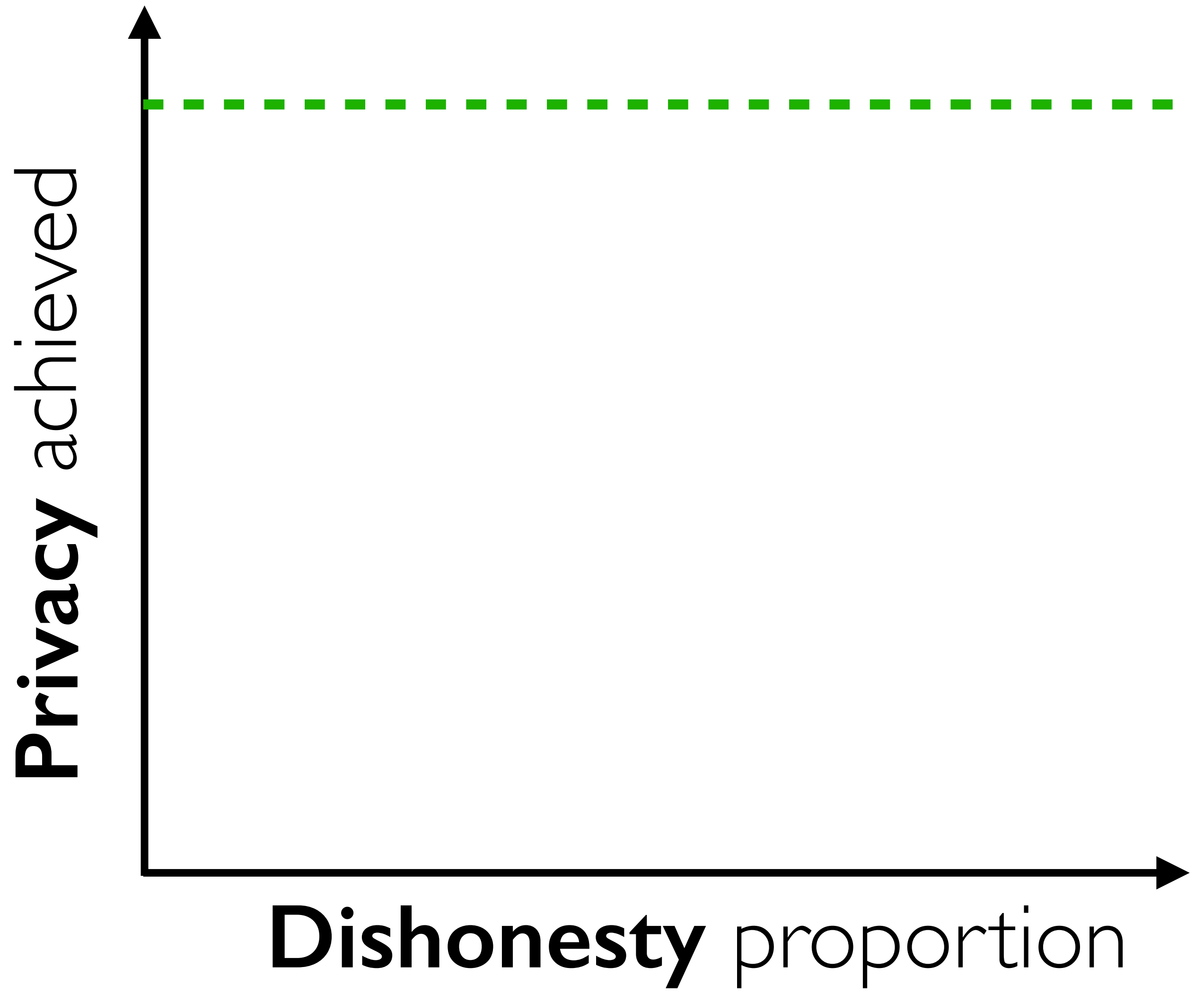
Need for Lotto

Relied assumption

May **not** hold

Privacy-Enhancing Technique	Federated Learning	Secure Aggregation	Differential Privacy
Privacy Guarantee	Data kept on premises	Local updates unseen	Global update leaks little about any client

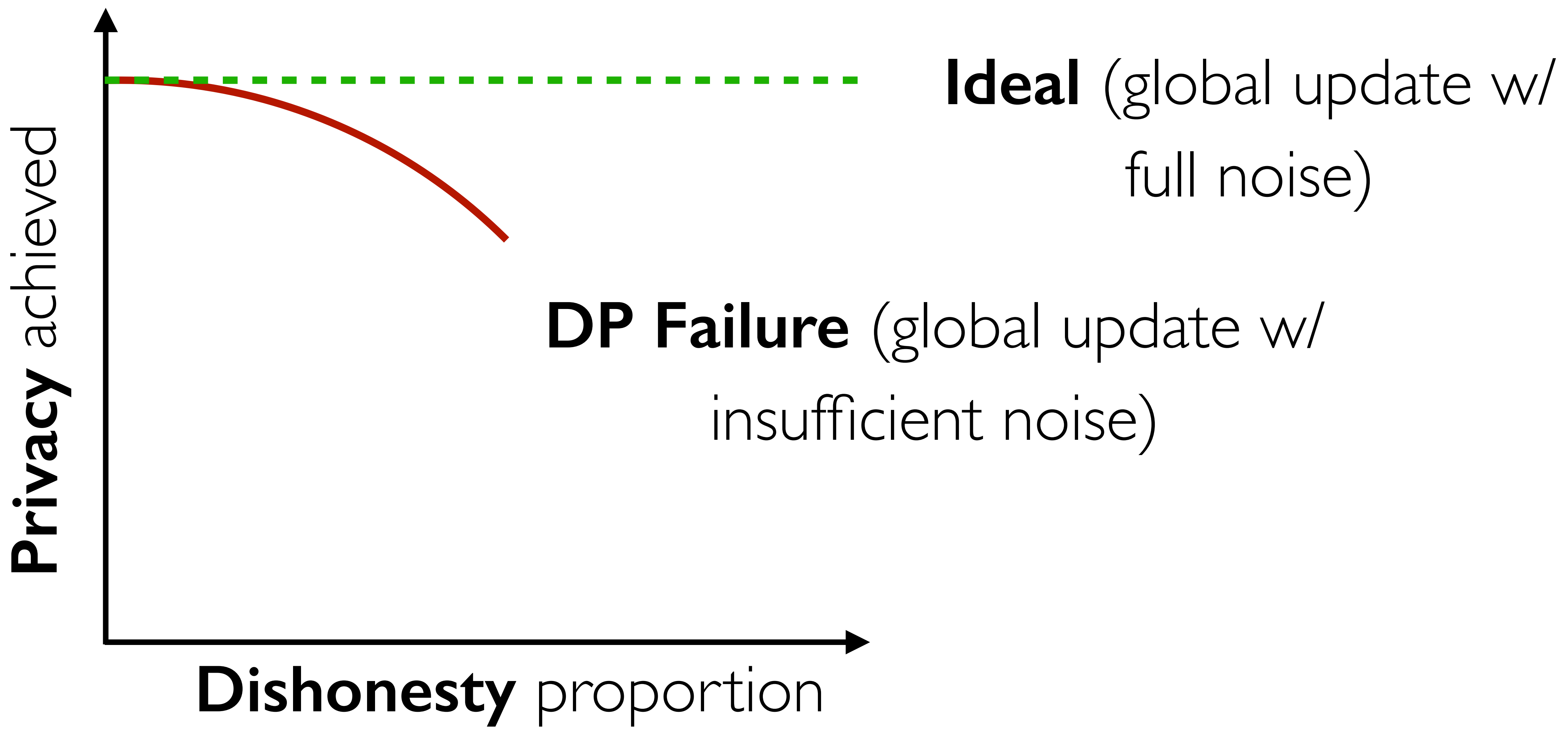
Need for Lotto



Ideal (global update w/ full noise)

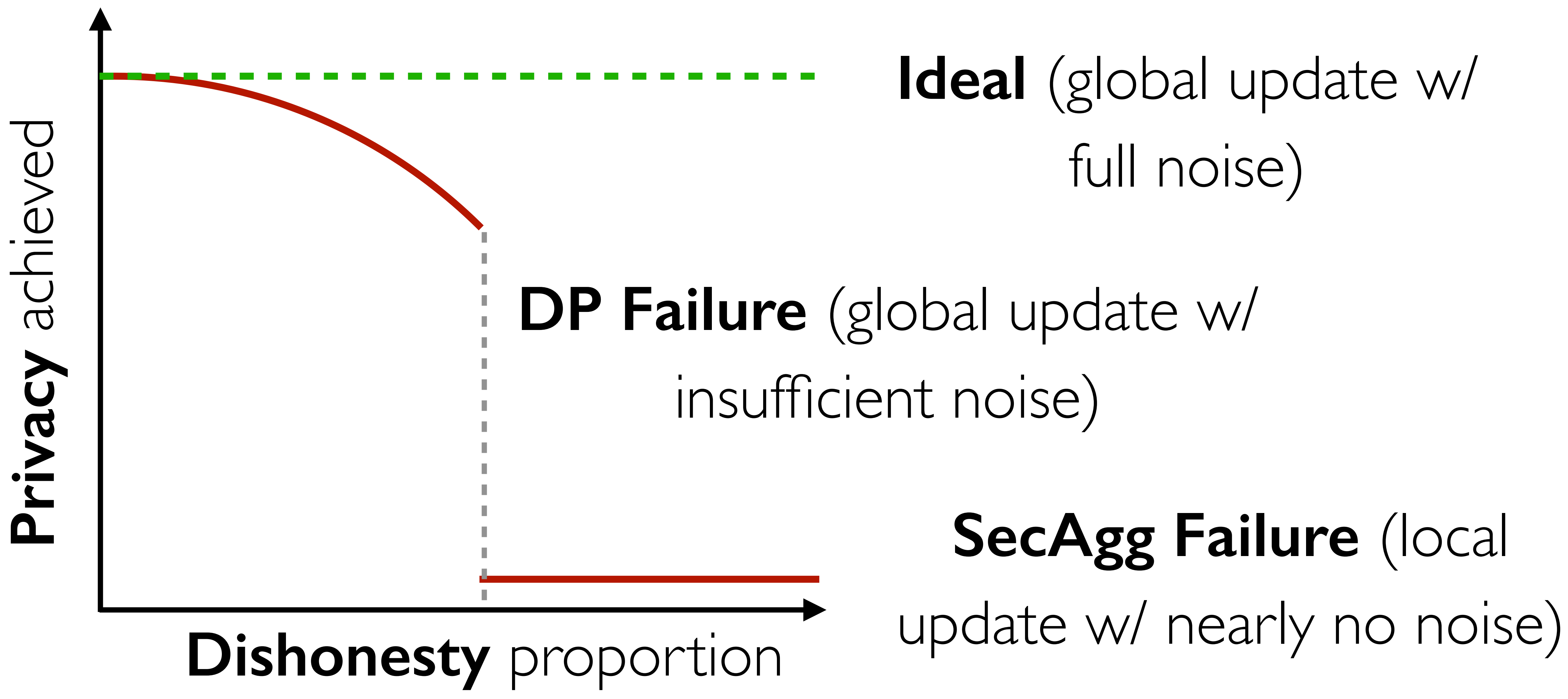
Secure Aggregation | Differential Privacy

Need for Lotto



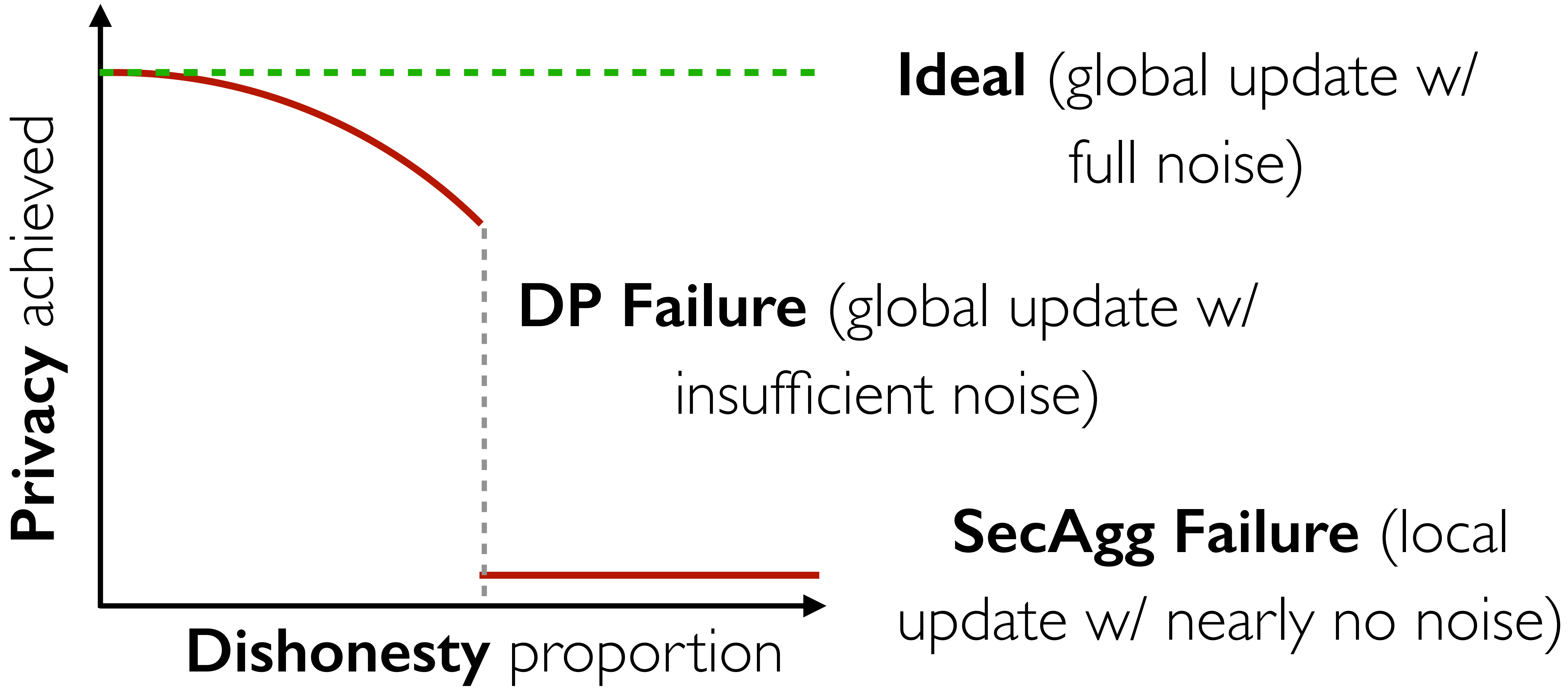
Secure Aggregation | Differential Privacy

Need for Lotto



Secure Aggregation | Differential Privacy

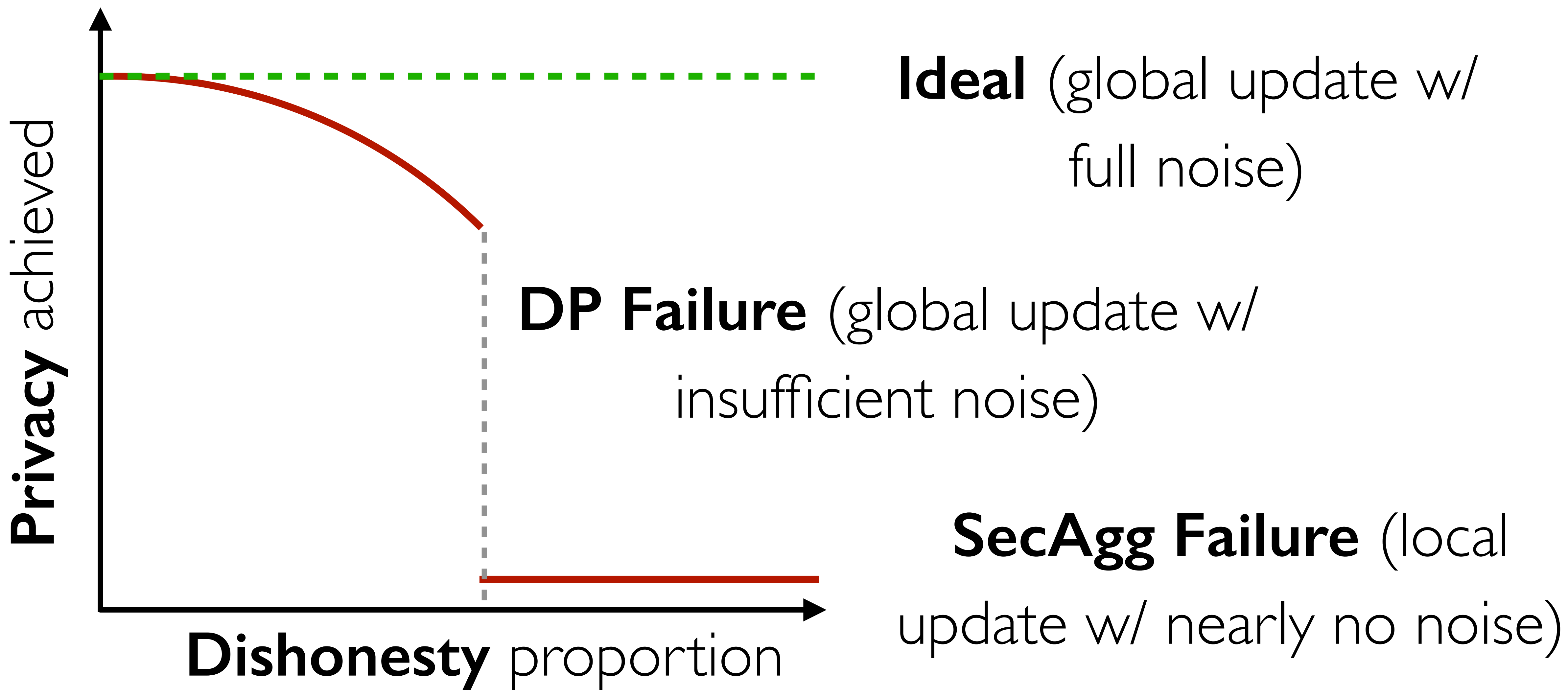
Need for Lotto



Assumption: honest participants

Secure Aggregation	Differential Privacy

Need for Lotto

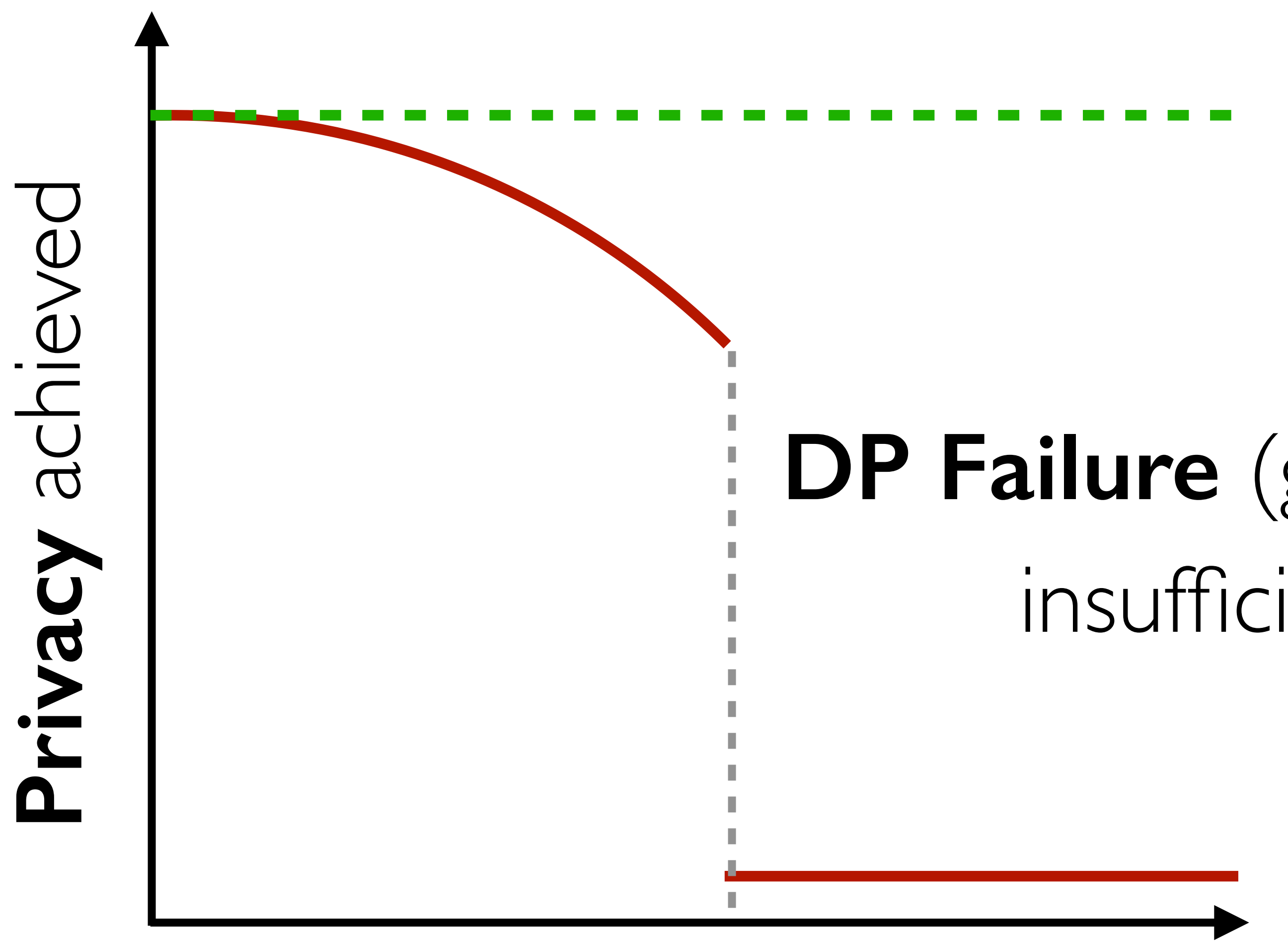


Assumption: honest participants

Secure Aggregation | Differential Privacy

Federated Learning

Need for Lotto



Ideal (global update w/ full noise)

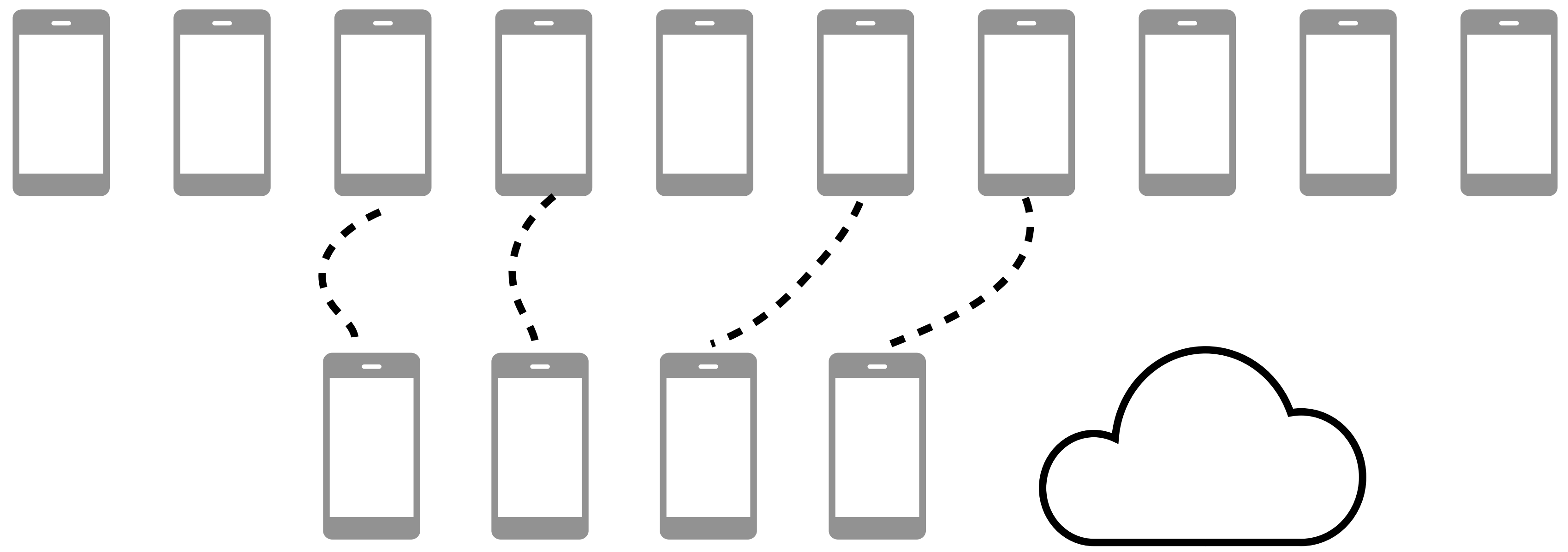
DP Failure (global update w/ insufficient noise)

SecAgg Failure (local update w/ nearly no noise)

Assumption: honest participants

Secure Aggregation | Differential Privacy

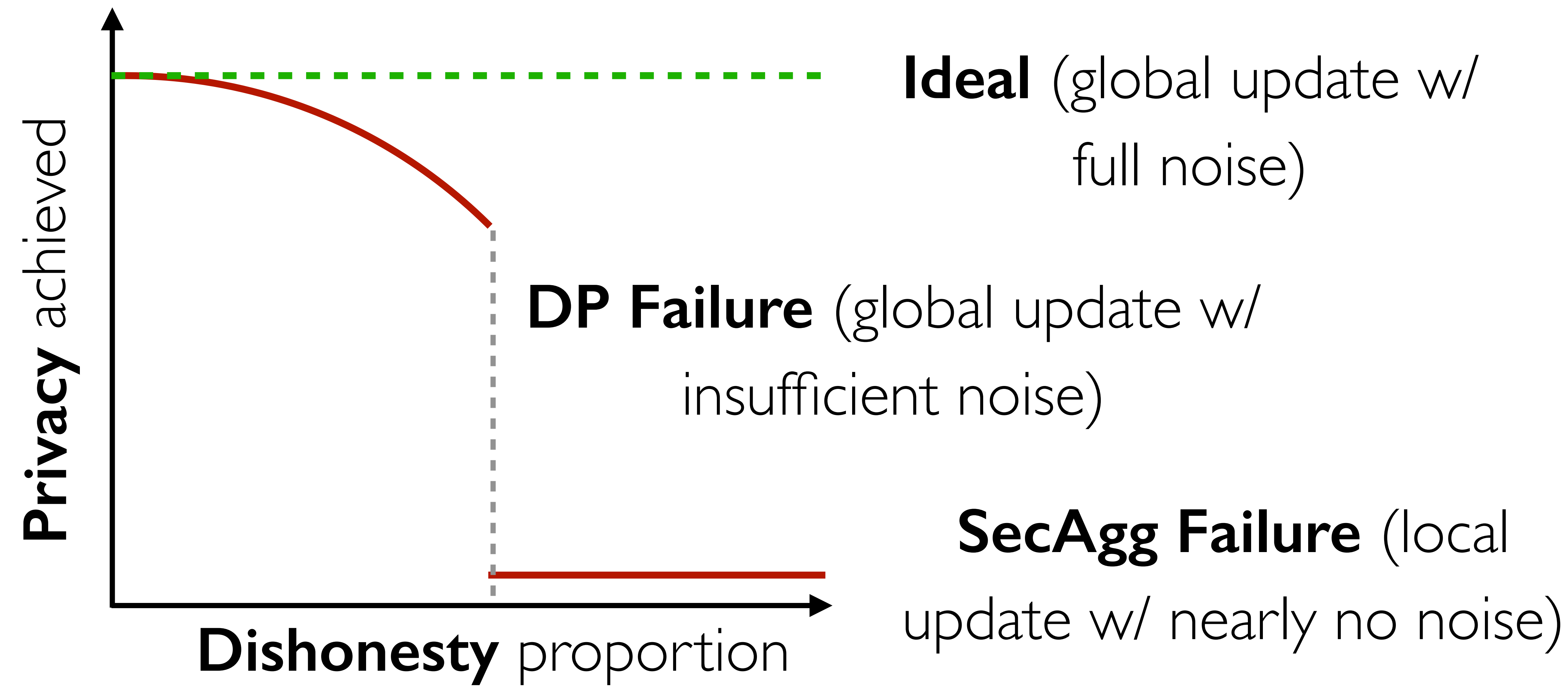
Population (10^4 - 10^8)



Selected participants (10^2 - 10^3)

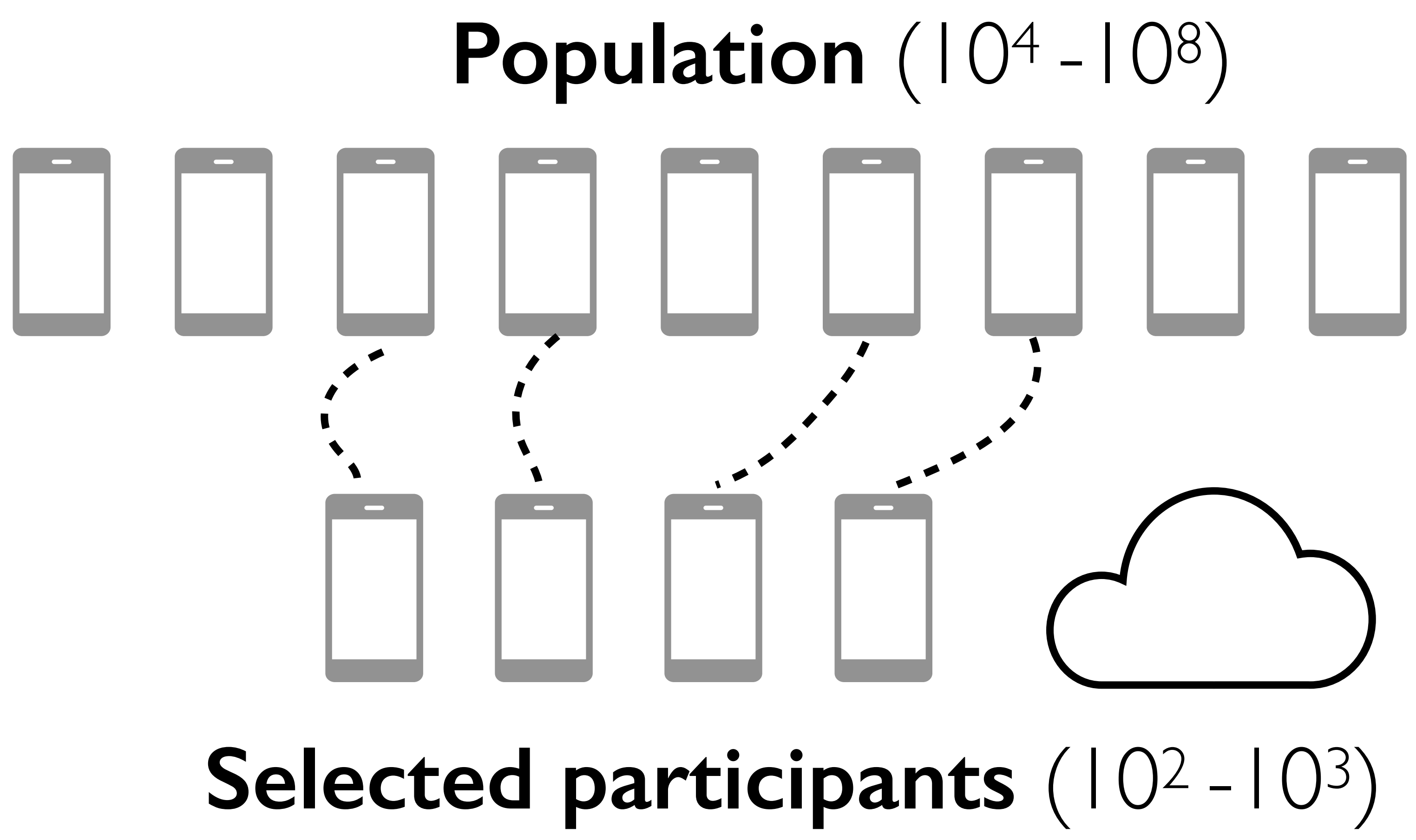
Federated Learning

Need for Lotto



Assumption: honest participants

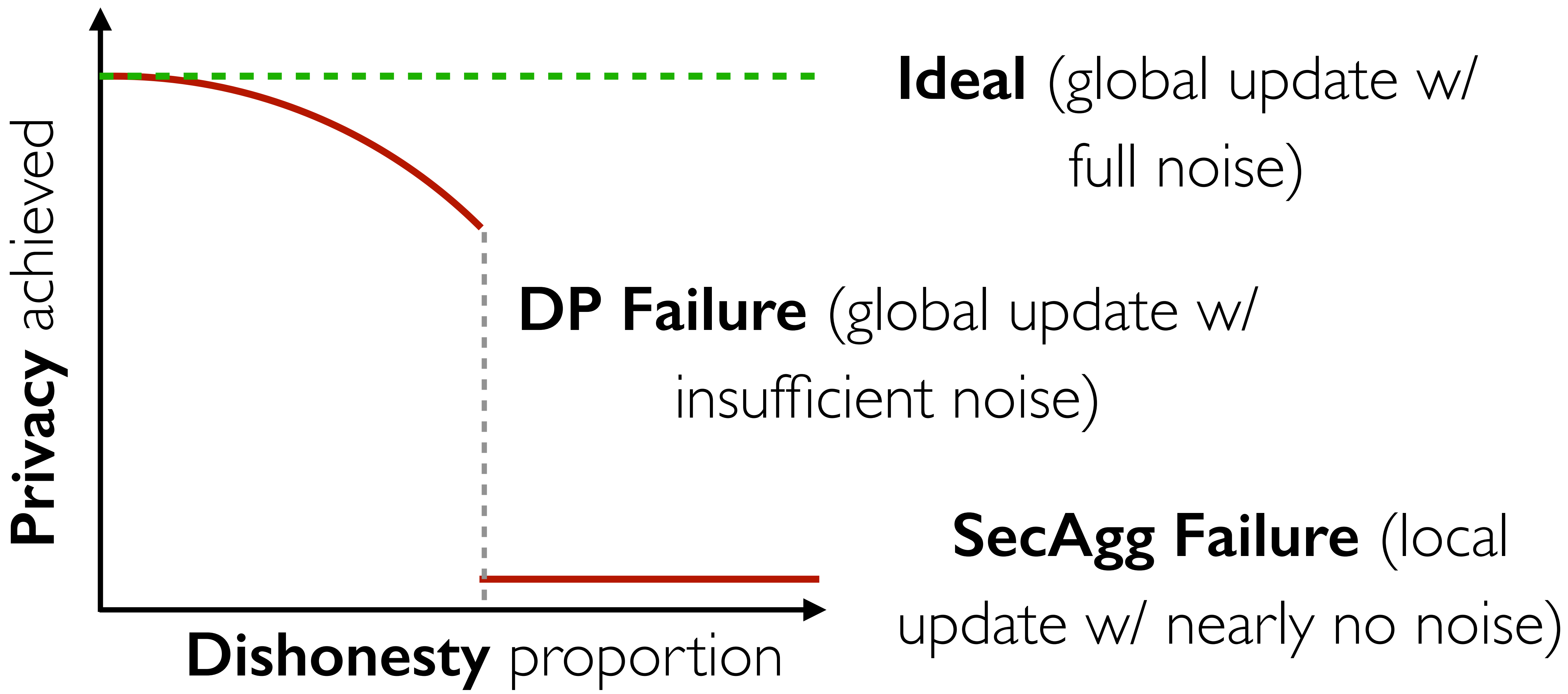
Secure Aggregation | Differential Privacy



- **Random**: uniform chance

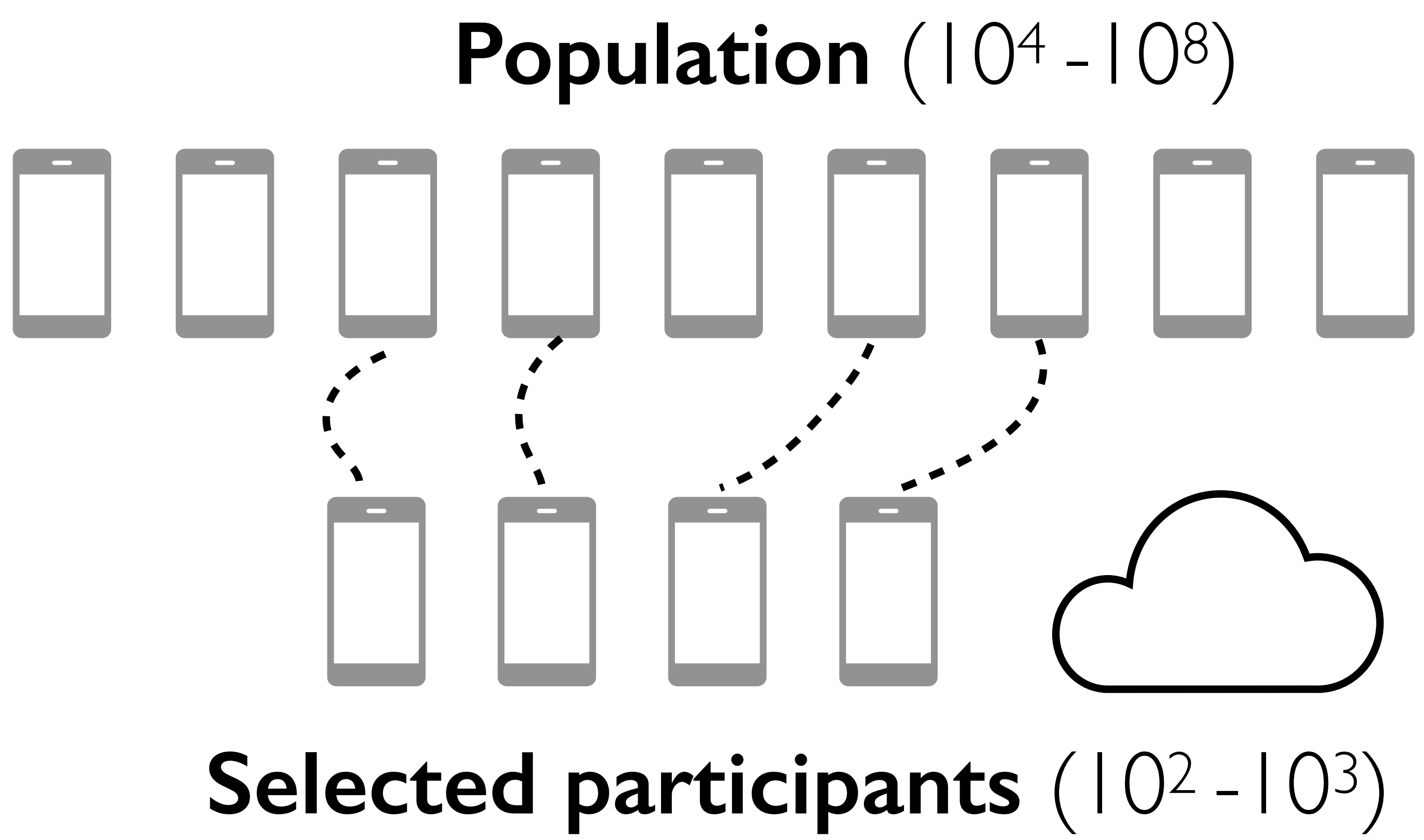
Federated Learning

Need for Lotto



Assumption: honest participants

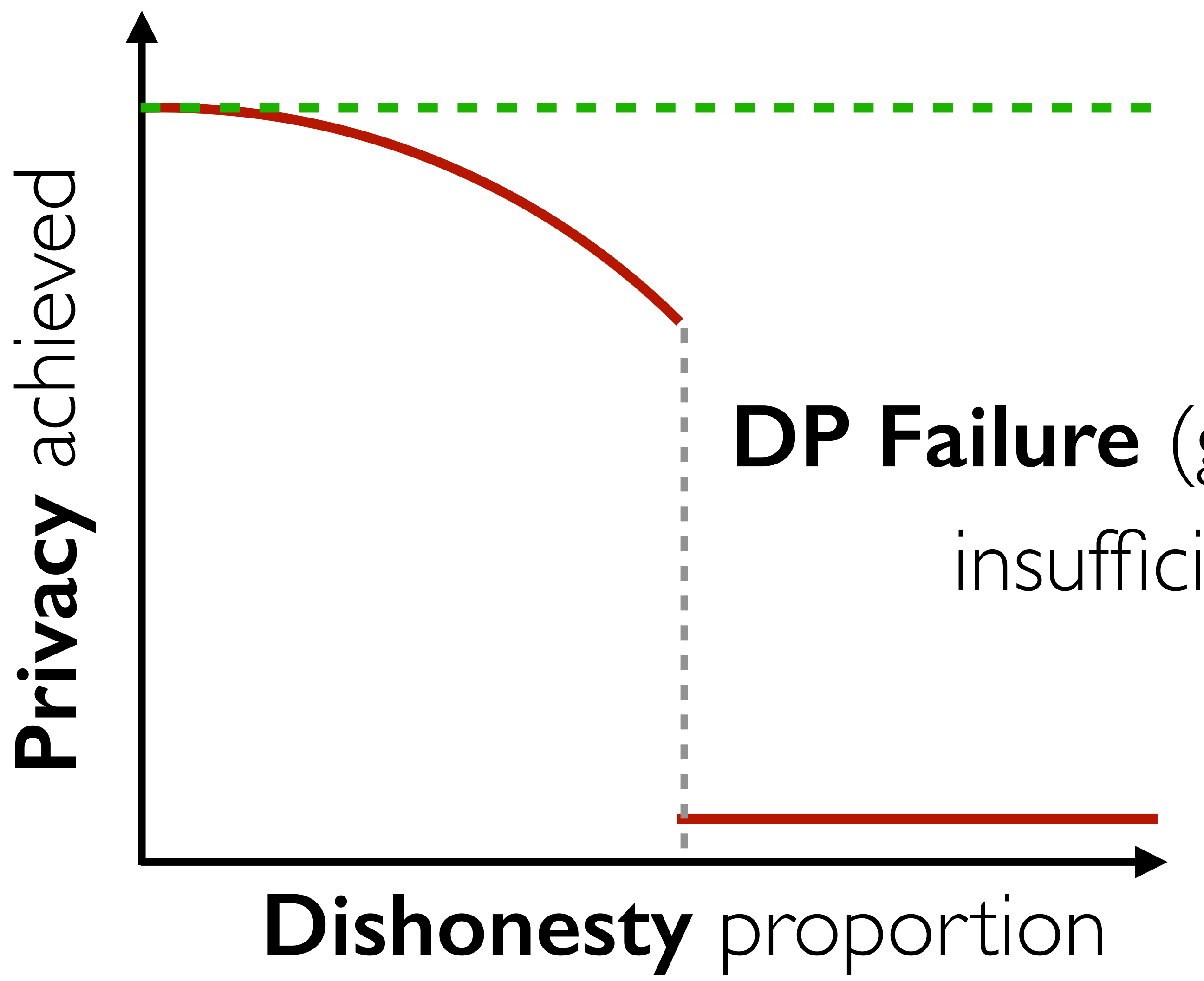
Secure Aggregation | Differential Privacy



- **Random**: uniform chance
- **Informed**: "best-performing" clients are preferred (e.g., high speed and/or rich data)

Federated Learning

Need for Lotto



Ideal (global update w/ full noise)

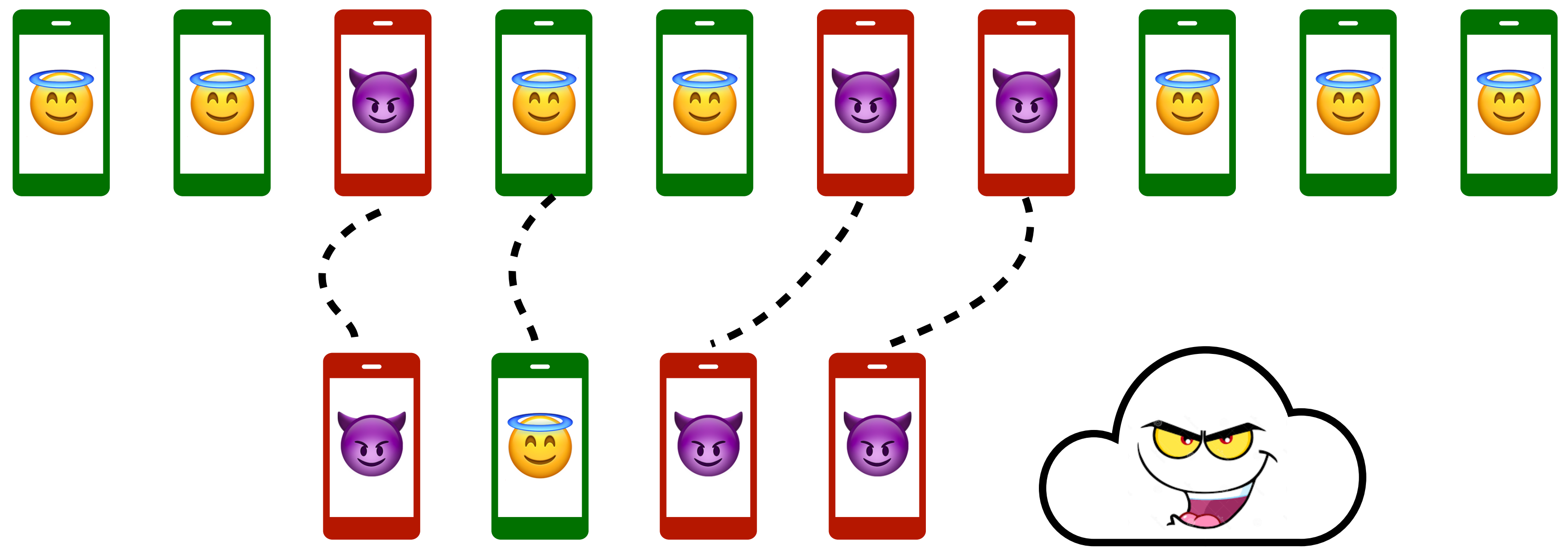
DP Failure (global update w/ insufficient noise)

SecAgg Failure (local update w/ nearly no noise)

Assumption: honest participants

Secure Aggregation | Differential Privacy

Population (10^4 - 10^8)



Selected participants (10^2 - 10^3)

Problem: participant selection can be **manipulated** by the **malicious** server

Federated Learning

Lotto: Random selection

Lotto: Random selection

Current round: 2

	Randomness
#1	$\mathbf{RF}_{pk1}(2) = 9$
#2	$\mathbf{RF}_{pk2}(2) = 1$
#3	$\mathbf{RF}_{pk3}(2) = 7$
…	…

Public keys

Selection criteria: <3

Lotto: Random selection

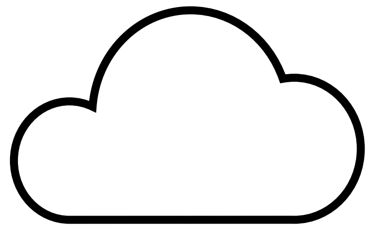
Current round: 2

	Randomness	Select
#1	$\mathbf{RF}_{pk1}(2) = 9$	No
#2	$\mathbf{RF}_{pk2}(2) = 1$	Yes
#3	$\mathbf{RF}_{pk3}(2) = 7$	No
…	…	…

Selection criteria: <3

Lotto: Random selection

Current round: 2



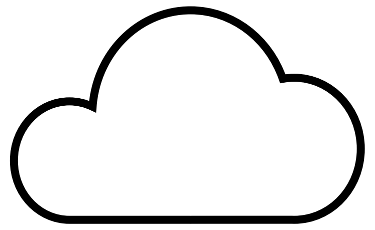
	Randomness	Select	Randomness	Select
#1	$\mathbf{RF}_{pk1}(2) = 9$	No		Yes
#2	$\mathbf{RF}_{pk2}(2) = 1$	Yes	Does NOT matter.	No
#3	$\mathbf{RF}_{pk3}(2) = 7$	No		No
…	…	…		…

Selection criteria: <3

For dishonest majority

Lotto: Random selection

Current round: 2



	Randomness	Select	Randomness	Select
#1	$\mathbf{RF}_{pk1}(2) = 9$	No		Yes
#2	$\mathbf{RF}_{pk2}(2) = 1$	Yes	Does NOT matter.	No
#3	$\mathbf{RF}_{pk3}(2) = 7$	No		No
…	…	…		…

Selection criteria: <3

For dishonest majority

Potential approach:

- Mutual verification

Lotto: Random selection

Current round: 2

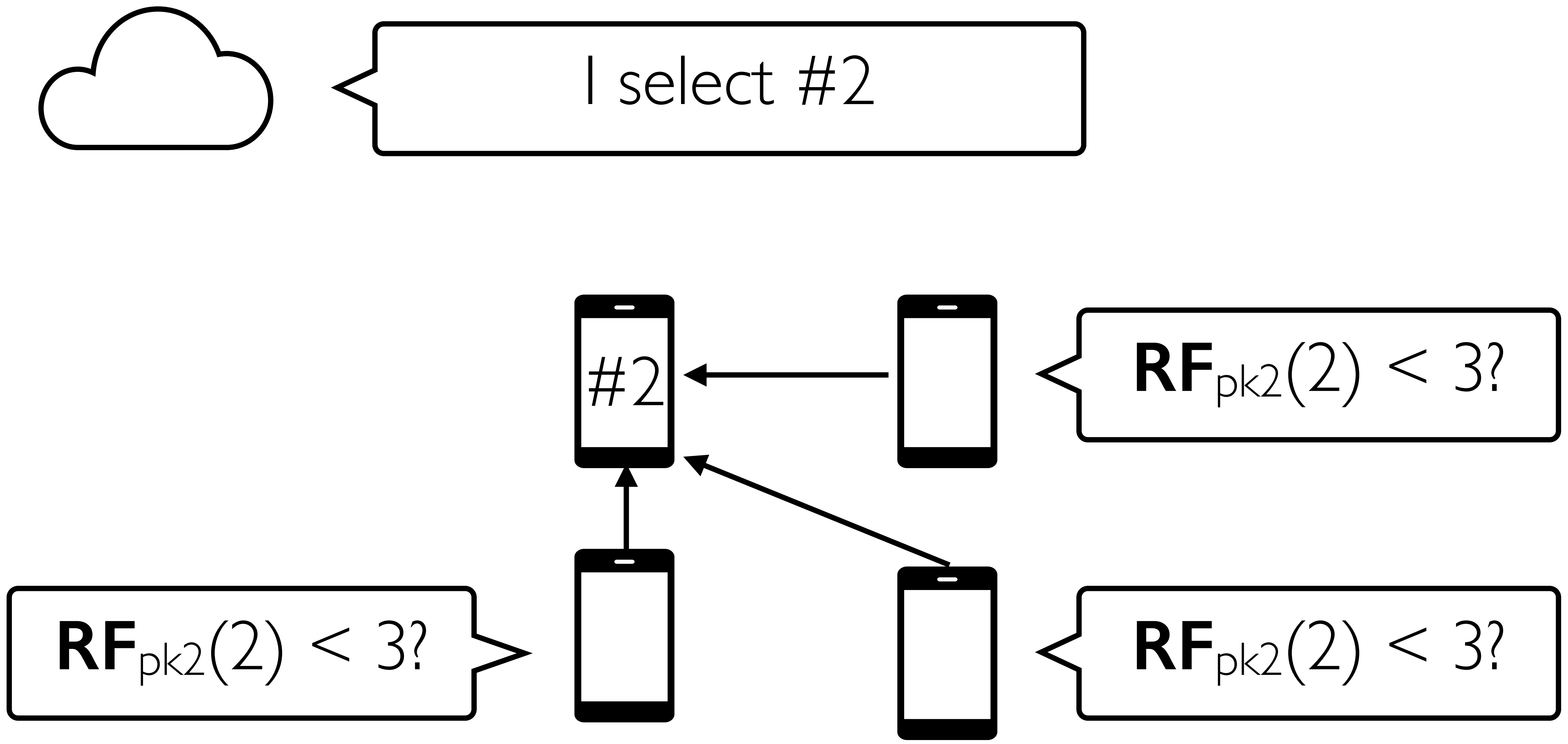
	Randomness	Select	Randomness	Select
#1	$\mathbf{RF}_{pk1}(2) = 9$	No	Does NOT matter.	Yes
#2	$\mathbf{RF}_{pk2}(2) = 1$	Yes		No
#3	$\mathbf{RF}_{pk3}(2) = 7$	No		No
…	…	…		…

Selection criteria: <3

For dishonest majority

Potential approach:

- Mutual verification



Lotto: Random selection

Current round: 2

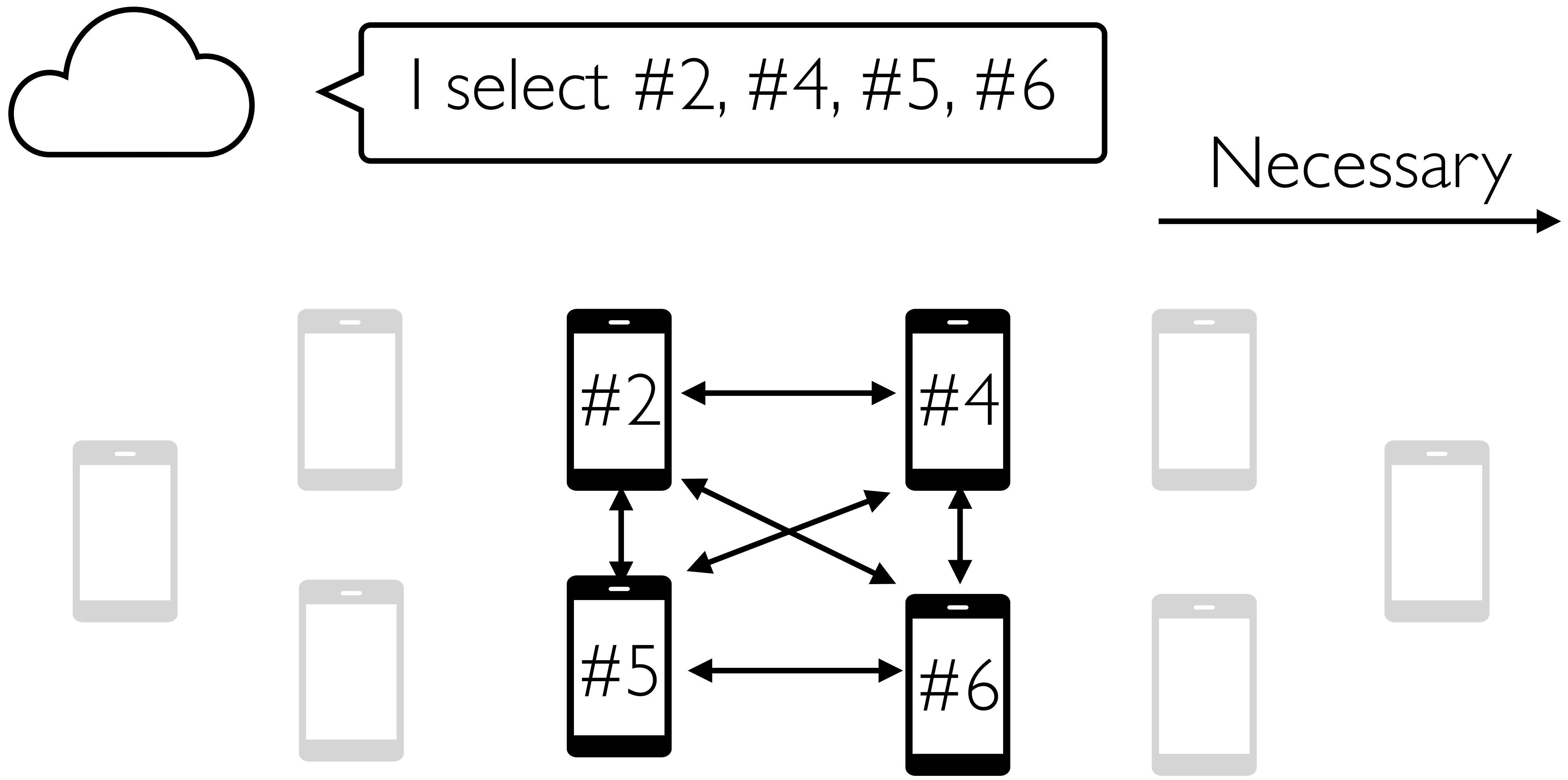
	Randomness	Select	Randomness	Select
#1	$\mathbf{RF}_{pk1}(2) = 9$	No		Yes
#2	$\mathbf{RF}_{pk2}(2) = 1$	Yes	Does NOT matter.	No
#3	$\mathbf{RF}_{pk3}(2) = 7$	No		No
…	…	…		…

Selection criteria: <3

For dishonest majority

Potential approach:

- Mutual verification
- Only within participants (10^2 - 10^3)



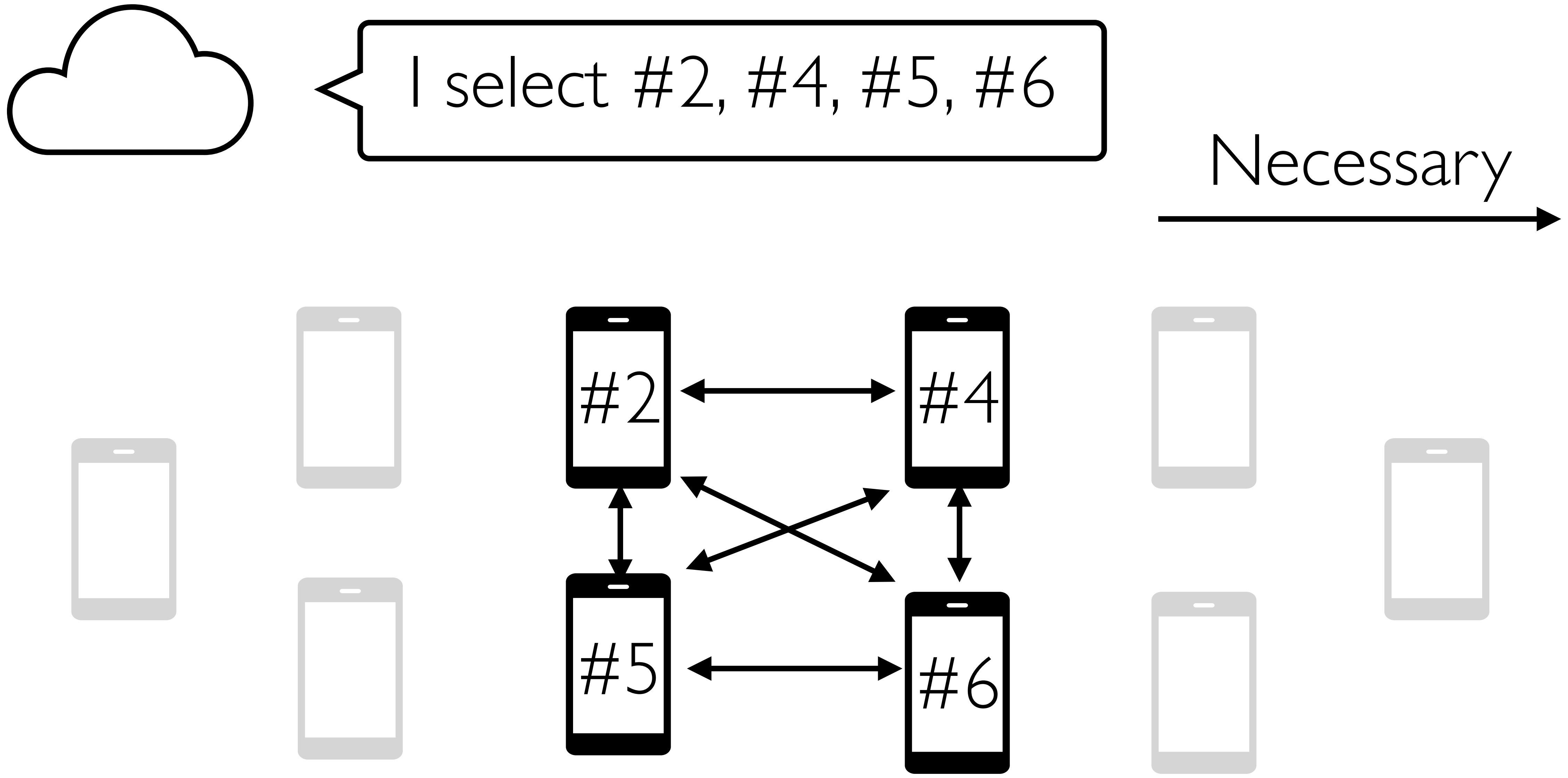
Lotto: Random selection

What is achieved:

Each participant
sees a list of peers

Potential approach:

- Mutual verification
- Only within participants (10^2 - 10^3)



Lotto: Random selection

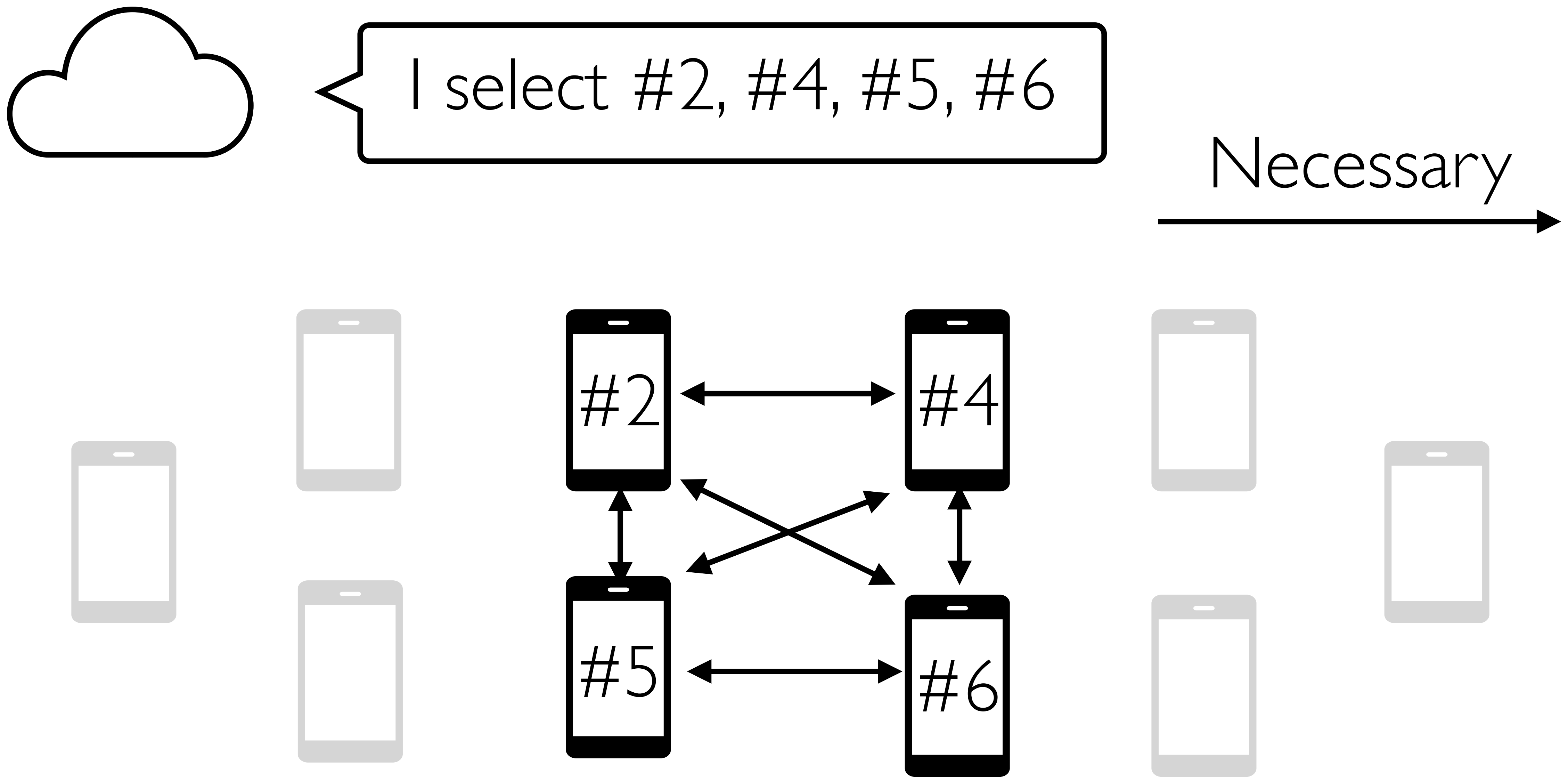
What is achieved:

Each participant
sees a list of peers who
presents only by chance.

E.g., $$\frac{\text{Selection criteria: } \mathbf{<3}}{\text{Output range: } \mathbf{[0, 10)}} = 3/10$$

Potential approach:

- Mutual verification
- Only within participants (10^2 - 10^3)



Lotto: Random selection

What is achieved:

Each participant

sees a list of peers who

presents only by chance.

What happens to the absent?

Lotto: Random selection

What is achieved:

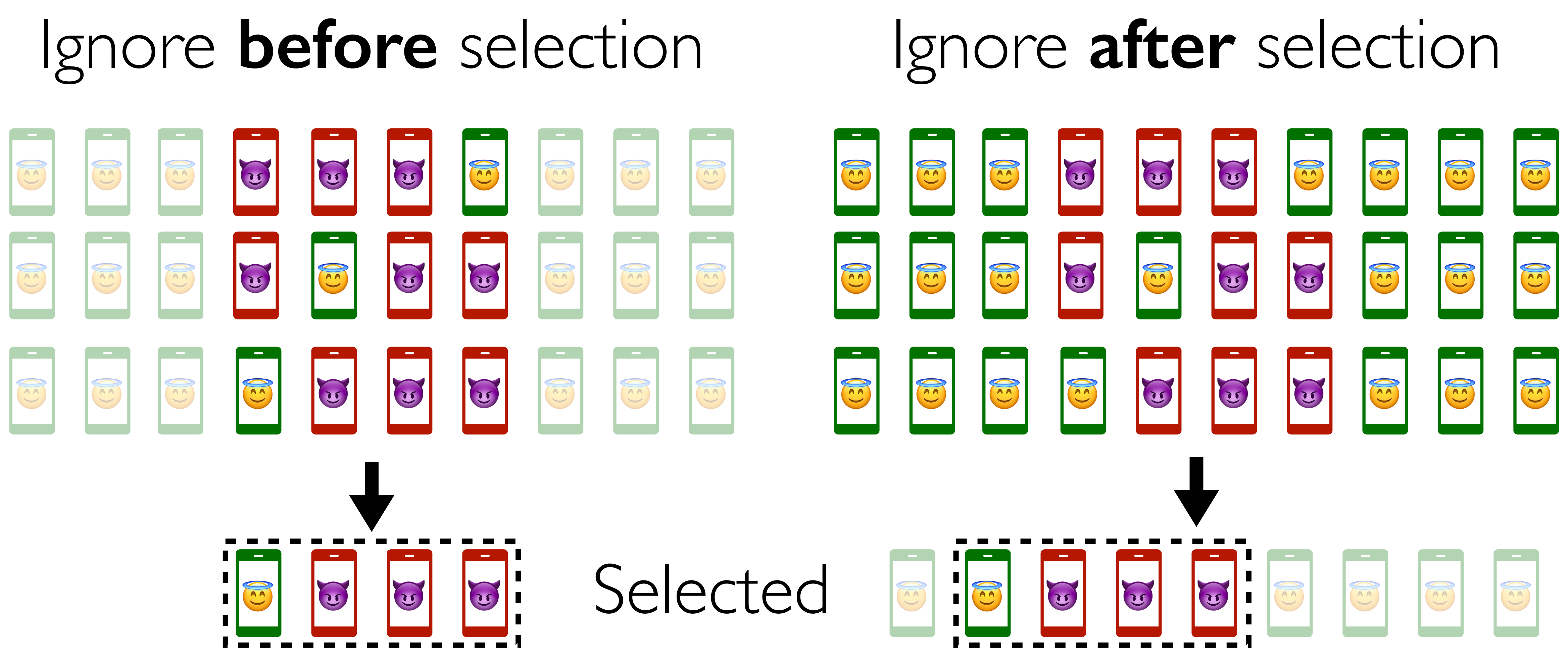
Each participant

sees a list of peers who

presents only by chance.

What happens to the absent?

Problem: The server may arbitrarily **ignore honest** clients



Lotto: Random selection

What is achieved:

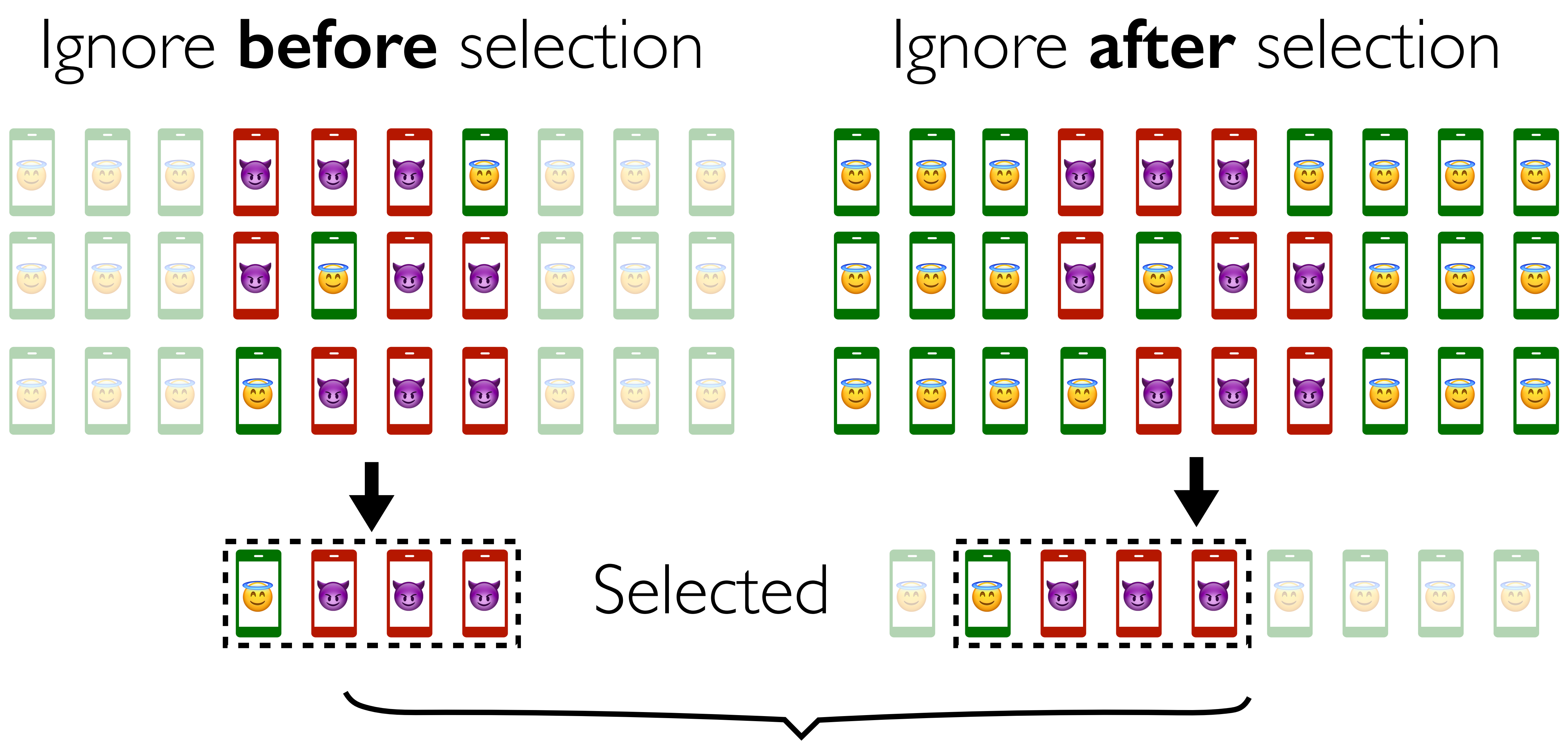
Each participant

sees a list of peers who

presents only by chance.

What happens to the absent?

Problem: The server may arbitrarily **ignore honest** clients



Unbounded advantage in growing dishonesty

Lotto: Random selection

Solution: Enforce a **large enough list** and a **small enough chance**.

What is achieved:

Each participant

sees a list of peers who

presents only by chance.

What happens to the absent?

Lotto: Random selection

What is achieved:

Each participant

sees a list of peers who

presents only by chance.

What happens to the absent?

Solution: Enforce a **large enough list** and a **small enough chance**.

Example

- **len(list)**: ≥ 200
- **Chance**: $\leq 0.1\%$

Lotto: Random selection

What is achieved:

Each participant

sees a list of peers who

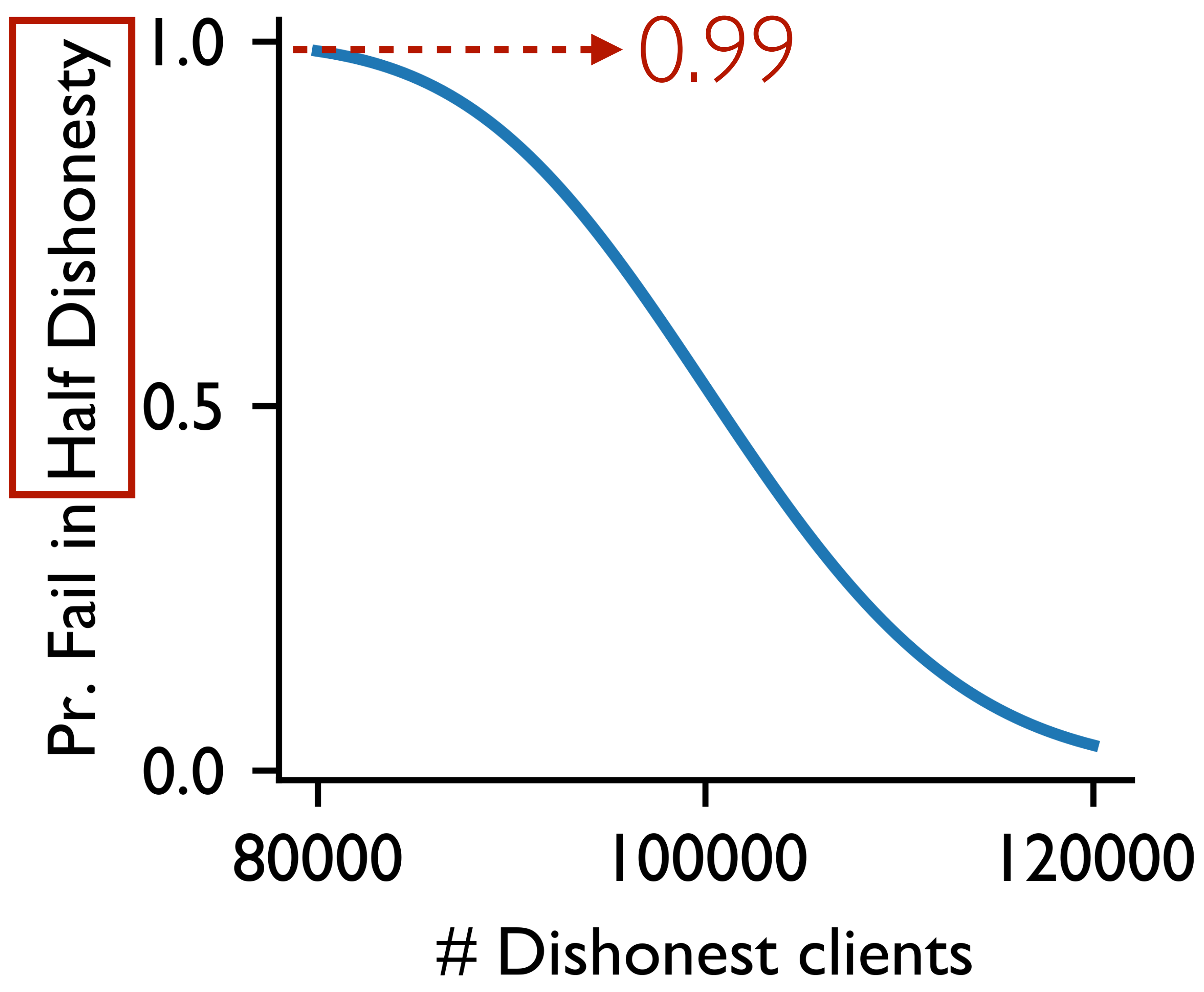
presents only by chance.

What happens to the absent?

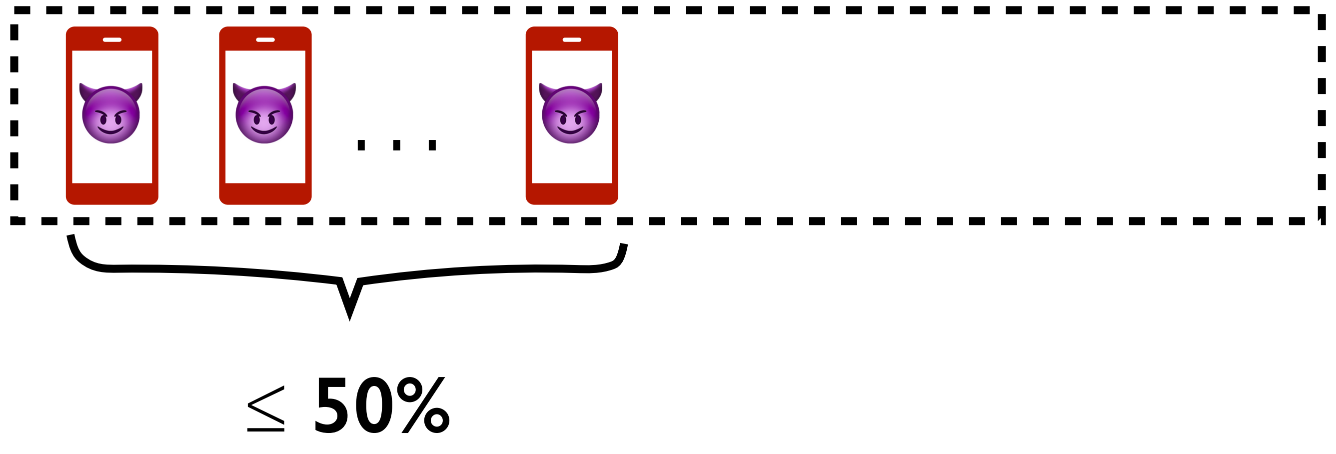
Solution: Enforce a **large enough list** and a **small enough chance**.

Example

- **len(list)**: ≥ 200
- **Chance**: ≤ 0.1%



Selected



Lotto: Random selection

What is achieved:

Each participant

sees a list of peers who

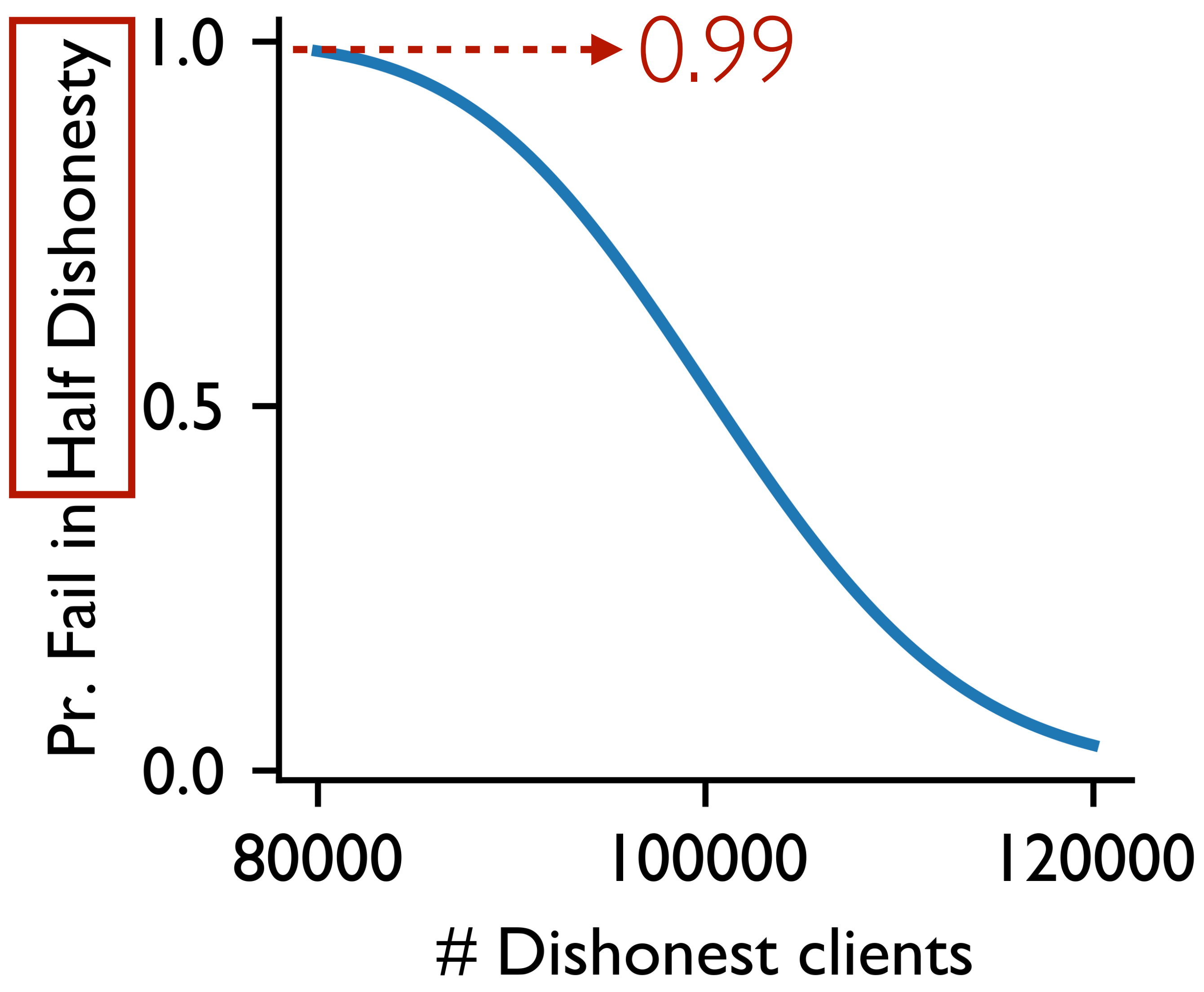
presents only by chance.

The absent will not get arbitrarily ignored

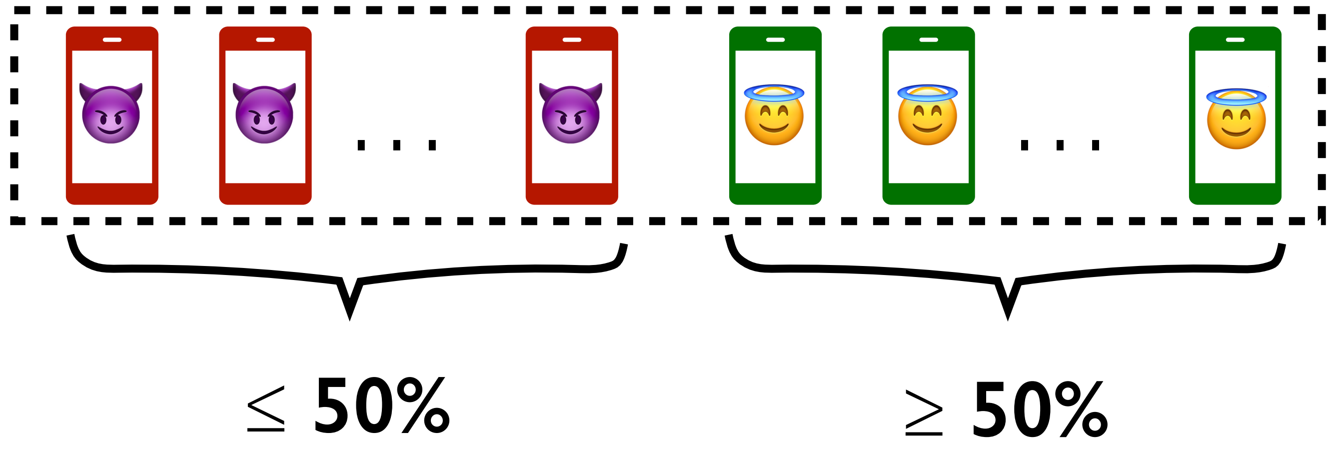
Solution: Enforce a **large enough list** and a **small enough chance**.

Example

- **len(list)**: ≥ 200
- **Chance**: ≤ 0.1%



Selected



Lotto: Random selection

What is achieved:

Each participant

sees a list of peers who

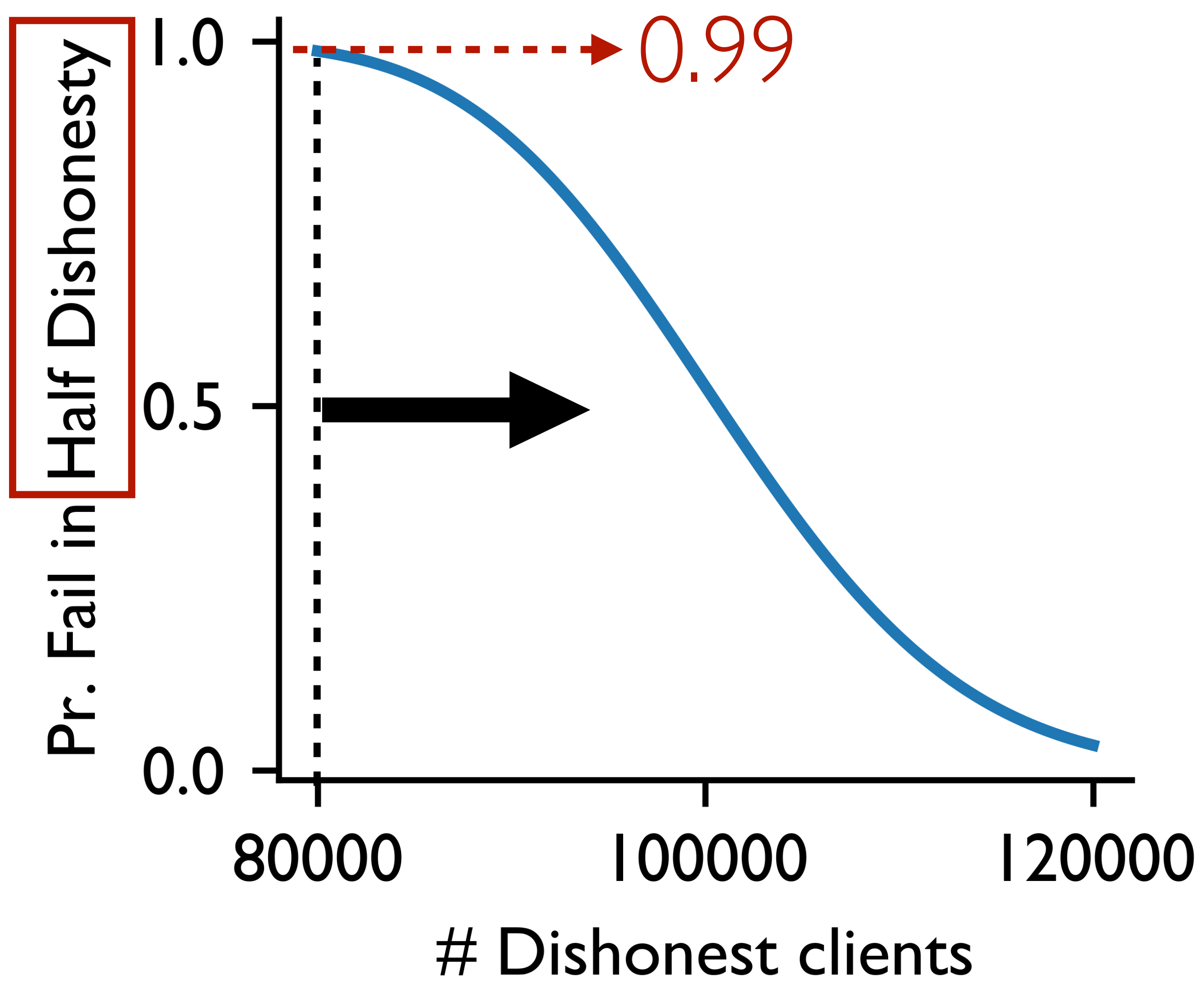
presents only by chance.

The absent will not get arbitrarily ignored

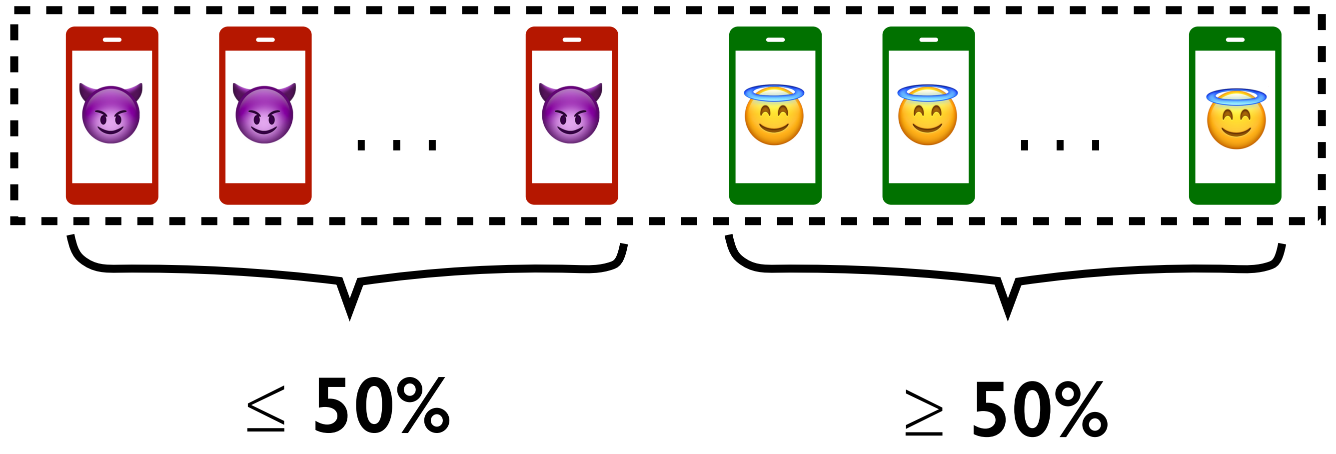
Solution: Enforce a **large enough list** and a **small enough chance**.

Example

- **len(list)**: ≥ 200
- **Chance**: ≤ 0.1%



Selected



Lotto: Random selection

What is achieved:

Each participant

sees a list of peers who

presents only by chance.

Predictable

to server?

The absent will not get

arbitrarily ignored

Public Round index

Examples: #2 will be selected as $\mathbf{RF}_{pk2}(2) = 1 < 3$.

Public Public keys

Lotto: Random selection

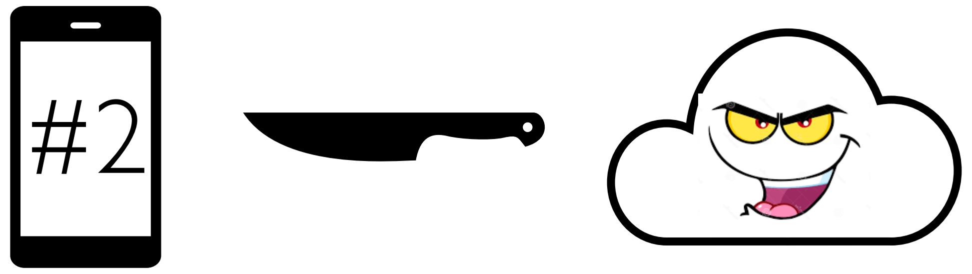
What is achieved:
Each participant
sees a list of peers who
presents only by chance.

Predictable
to server?

The absent will not get
arbitrarily ignored

Problem: Attack surfaces **enlarged**!

Examples: #2 will be selected as $\mathbf{RF}_{pk2}(2) = 1 < 3$.
Before training, the server may grow its advantage by



Focused hacking

Lotto: Random selection

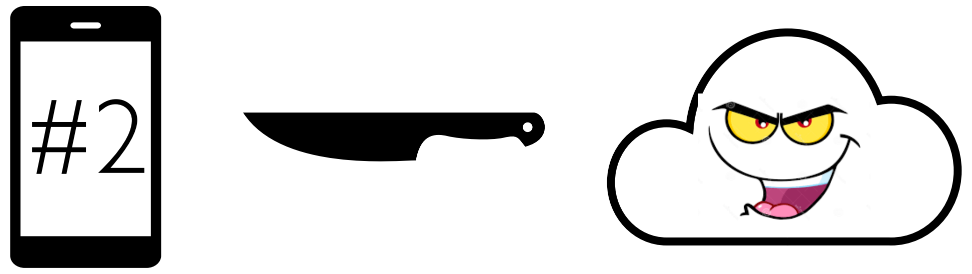
What is achieved:
Each participant
sees a list of peers who
presents only by chance.

Predictable
to server?

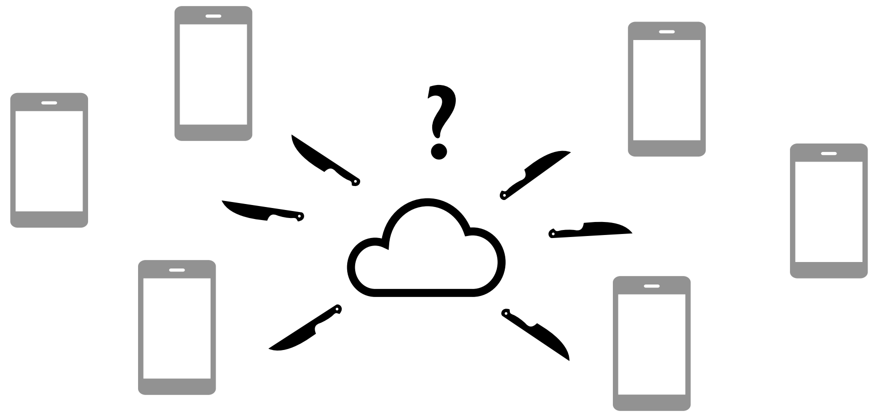
The absent will not get
arbitrarily ignored

Problem: Attack surfaces **enlarged**!

Examples: #2 will be selected as $\mathbf{RF}_{pk2}(2) = 1 < 3$.
Before training, the server may grow its advantage by



vs



Focused hacking

Random compromise

Lotto: Random selection

What is achieved:

Each participant

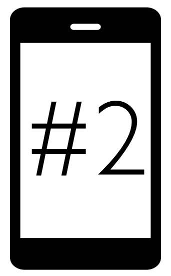
sees a list of peers who

presents only by chance.

Predictable to server?

The absent will not get arbitrarily ignored

Solution: **Self-sampling** with verifiable random functions (**VRFs**)[1,2].



Evaluation: $\textbf{VRF.eval}_{sk2}(2) = (\ 1, \quad)\ (\text{ output}, \quad)$

Secret key

[1]Micali et al. "Verifiable random functions", In FOCS '99
[2]Dodis et al. "A verifiable random function with short proofs and keys", In PKC '05

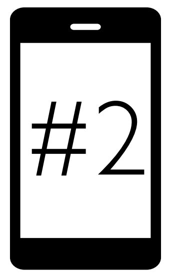
Lotto: Random selection

What is achieved:
Each participant
sees a list of peers who
presents only by chance.

Predictable
to server?

The absent will not get
arbitrarily ignored

Solution: **Self-sampling** with verifiable random functions (**VRFs**)[1,2].



Evaluation: $\mathbf{VRF.eval}_{sk2}(2) = (\ 1, \boldsymbol{\pi_2}\ )$ (output, **proof**)

[1]Micali et al. "Verifiable random functions", In FOCS '99
[2]Dodis et al. "A verifiable random function with short proofs and keys", In PKC '05

Lotto: Random selection

What is achieved:
Each participant
sees a list of peers who
presents only by chance.

Predictable to server?

The absent will not get arbitrarily ignored

Solution: **Self-sampling** with verifiable random functions (**VRFs**)[1,2].

#2

Evaluation: $\textbf{VRF.eval}_{sk2}(2) = (\ 1, \boldsymbol{\pi_2}\ )$ (output, **proof**)

Verification: $\textbf{VRF.ver}_{pk2}(\ 2, 1, \boldsymbol{\pi_2}\ ) = \text{True}$

Public key

[1]Micali et al. "Verifiable random functions", In FOCS '99
[2]Dodis et al. "A verifiable random function with short proofs and keys", In PKC '05

Lotto: Random selection

What is achieved:

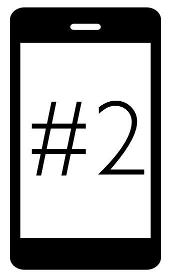
Each participant sees a list of peers who presents only by chance.

Unpredictable to server

The absent will not get arbitrarily ignored

Solution: **Self-sampling** with verifiable random functions (**VRFs**)[1,2].

I self-sample with $(1, \pi_2)$



Evaluation: $\mathbf{VRF.eval}_{sk2}(2) = (\ 1, \boldsymbol{\pi_2}\ )$ (output, **proof**)

Verification: $\mathbf{VRF.ver}_{pk2}(\ 2, 1, \boldsymbol{\pi_2}\ ) = \text{True}$

[1] Micali et al. "Verifiable random functions", In FOCS '99

[2] Dodis et al. "A verifiable random function with short proofs and keys", In PKC '05

Lotto: Random selection

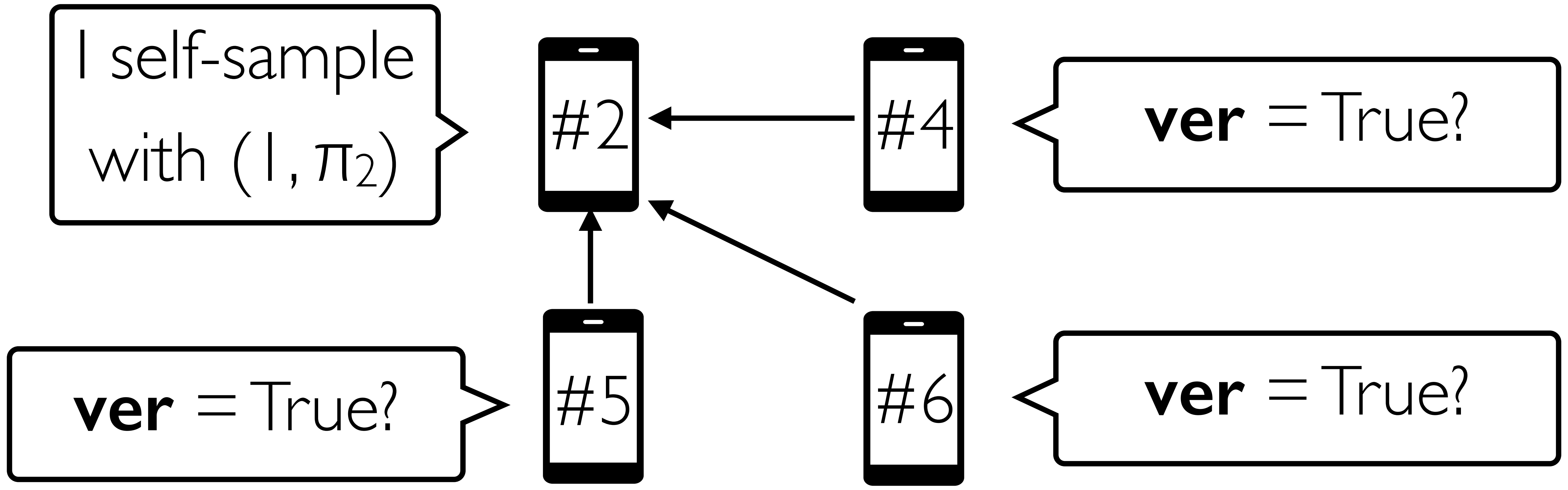
What is achieved:

Each participant sees a list of peers who presents only by chance.

Unpredictable to server

The absent will not get arbitrarily ignored

Solution: **Self-sampling** with verifiable random functions (**VRFs**)[1,2].



Evaluation: $\textbf{VRF.eval}_{sk2}(2) = (\ 1, \boldsymbol{\pi_2}\ )$ (output, **proof**)

Verification: $\textbf{VRF.ver}_{pk2}(\ 2, 1, \boldsymbol{\pi_2}\ ) = \text{True}$

[1]Micali et al. "Verifiable random functions", In FOCS '99
[2]Dodis et al. "A verifiable random function with short proofs and keys", In PKC '05

Lotto: Random selection

What is achieved:
Each participant
sees a list of peers who
presents only by chance.

Unpredictable to server

The absent will not get arbitrarily ignored

[1]Thus also of distributed DP (other privacy-enhancing techniques may not have this feature and this is left for future work).

Minor issues:

- **Participant consistency**: leverage SecAgg
- **Fixed sample size**: over-selection
- **Consistent round index**: uniqueness check

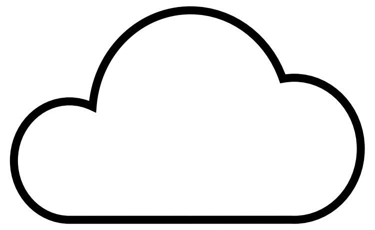
…

Please find more in the paper :)

Lotto: Informed selection

Lotto: Informed selection

Example



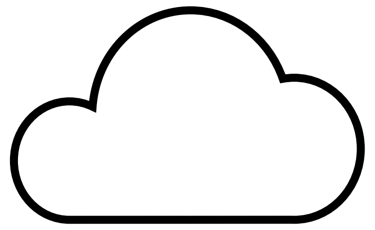
	(Est.) latency	Select	(Est.) latency	Select
#1	1.2s	Yes	Does NOT matter.	Yes
#2	2.7s	No		No
#3	1.6s	Yes		No
…	…	…		…

Selection criteria: the fastest

For dishonest majority

Lotto: Informed selection

Example



	(Est.) latency	Select	(Est.) latency	Select
#1	1.2s	Yes		Yes
#2	2.7s	No	Does NOT matter.	No
#3	1.6s	Yes		No
…	…	…		…

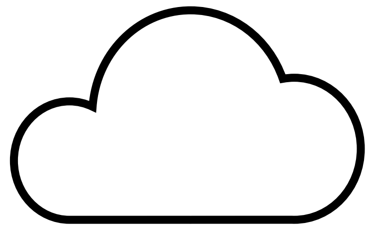
Selection criteria: the fastest

For dishonest majority

Major Challenge: Client metrics are **hard to verify** by honest clients

Lotto: Informed selection

Example



	(Est.) latency	Select	(Est.) latency	Select
#1	1.2s	Yes	Does NOT matter.	Yes
#2	2.7s	No		No
#3	1.6s	Yes		No
…	…	…		…

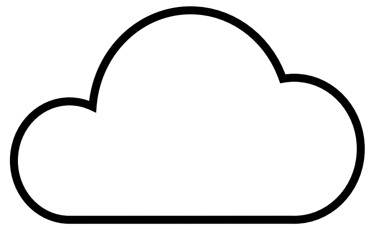
Selection criteria: the fastest For dishonest majority

Major Challenge: Client metrics are **hard to verify** by honest clients

Metrics can be easily fake

Lotto: Informed selection

Example



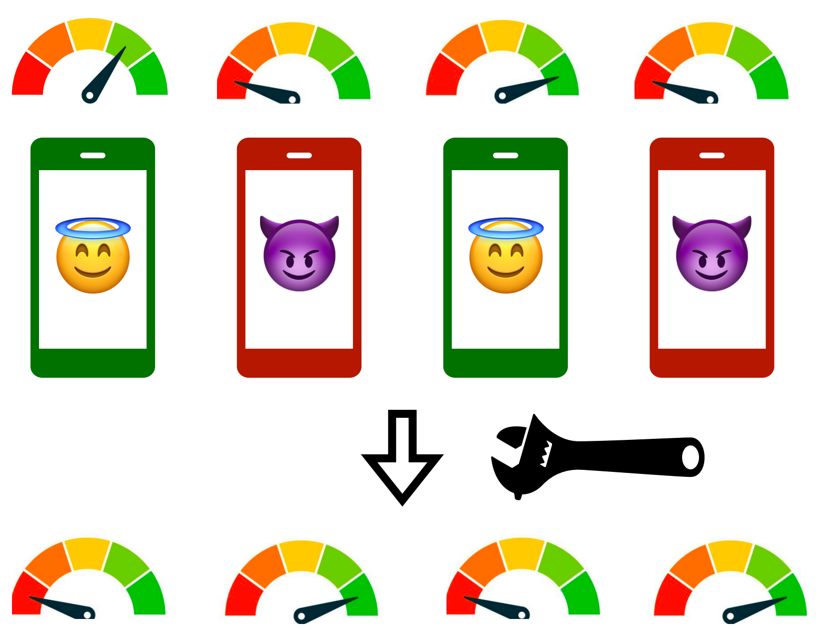

	(Est.) latency	Select	(Est.) latency	Select
#1	1.2s	Yes		Yes
#2	2.7s	No	Does NOT matter.	No
#3	1.6s	Yes		No
...	...	...		...

Selection criteria: the fastest

For dishonest majority

Major Challenge: Client metrics are **hard to verify** by honest clients

Metrics can be easily fake



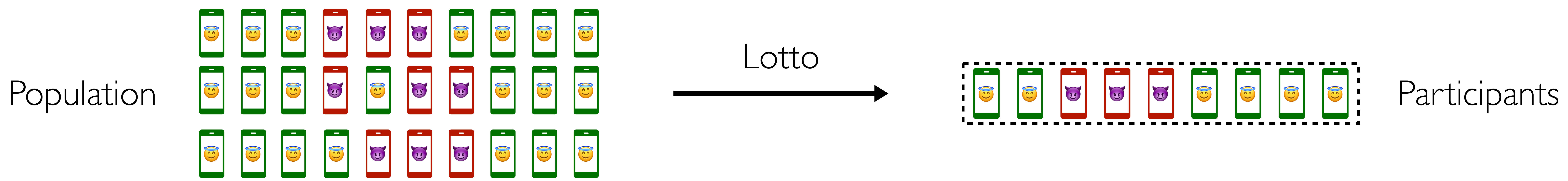
Solution: **Approximate** inform selection by **random** selection

Please find more in the paper :)

Lotto prevents arbitrary manipulation

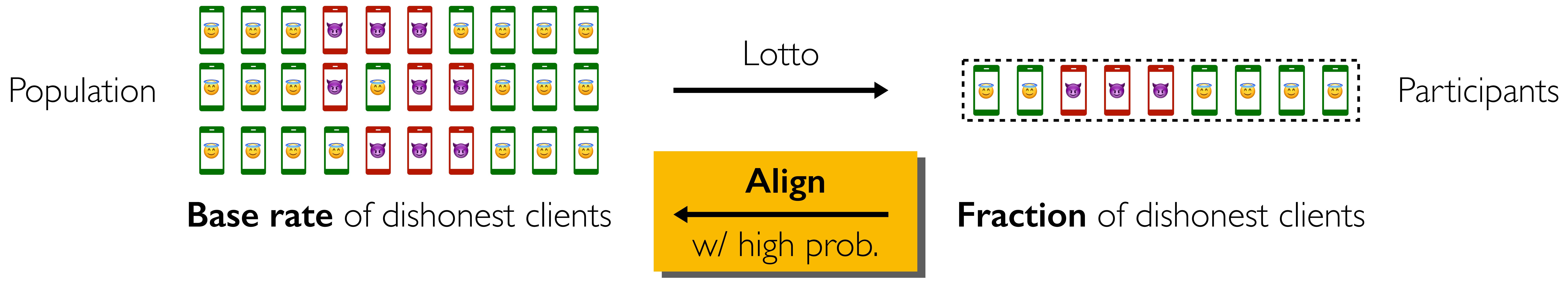
Lotto prevents arbitrary manipulation

What can be **proven**:



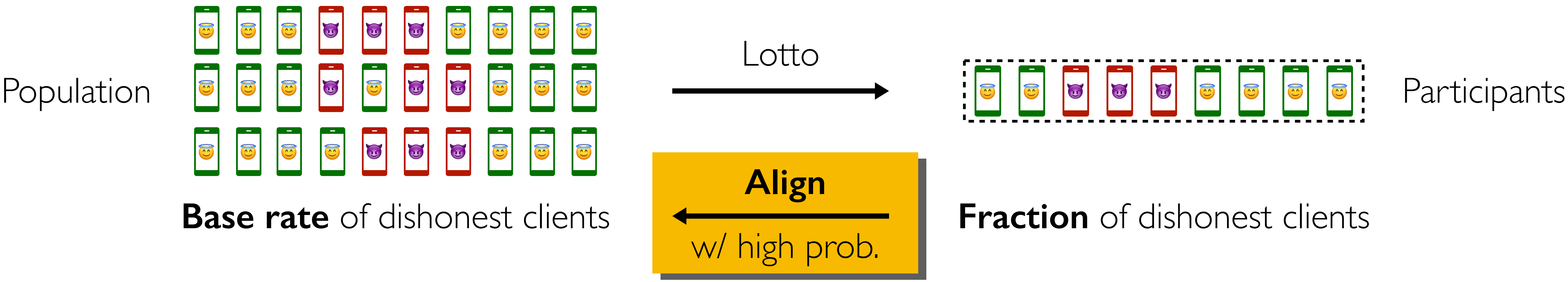
Lotto prevents arbitrary manipulation

What can be **proven**:



Lotto prevents arbitrary manipulation

What can be **proven**:



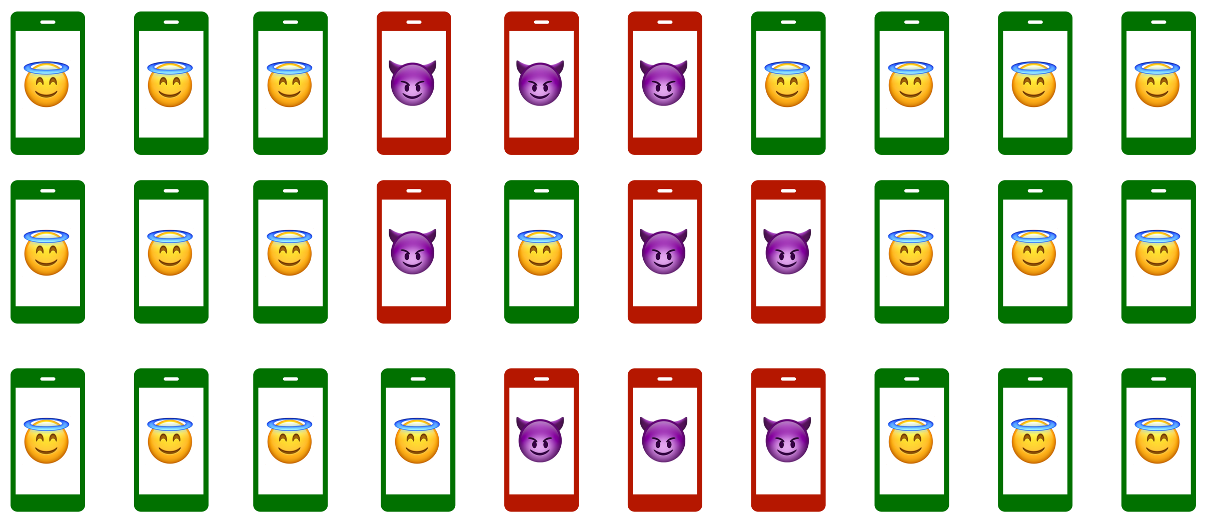
Example

- **Population**: 200,000
- **Dishonesty base rate**: 0.005

Lotto prevents arbitrary manipulation

What can be **proven**:

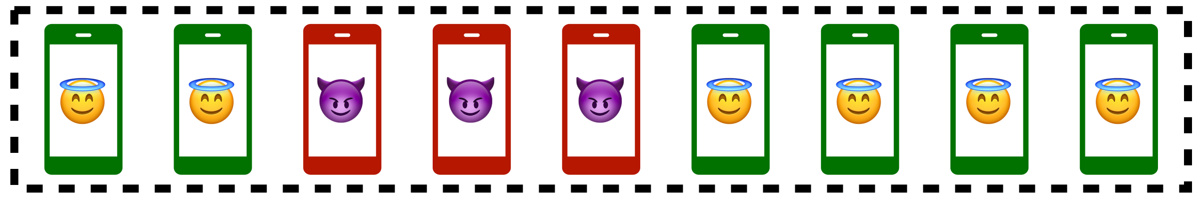
Population



Base rate of dishonest clients

Lotto →

Align ← w/ high prob.

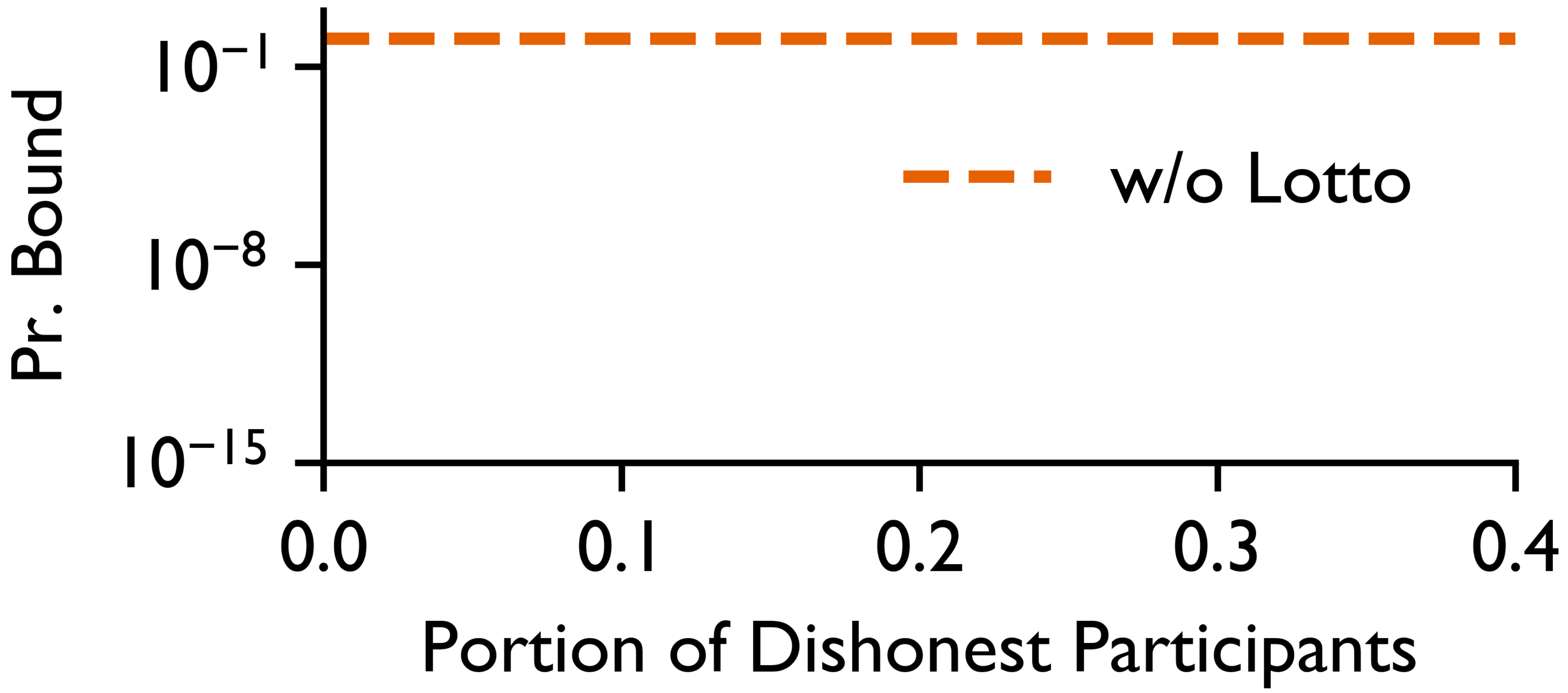


Participants

Fraction of dishonest clients

Example

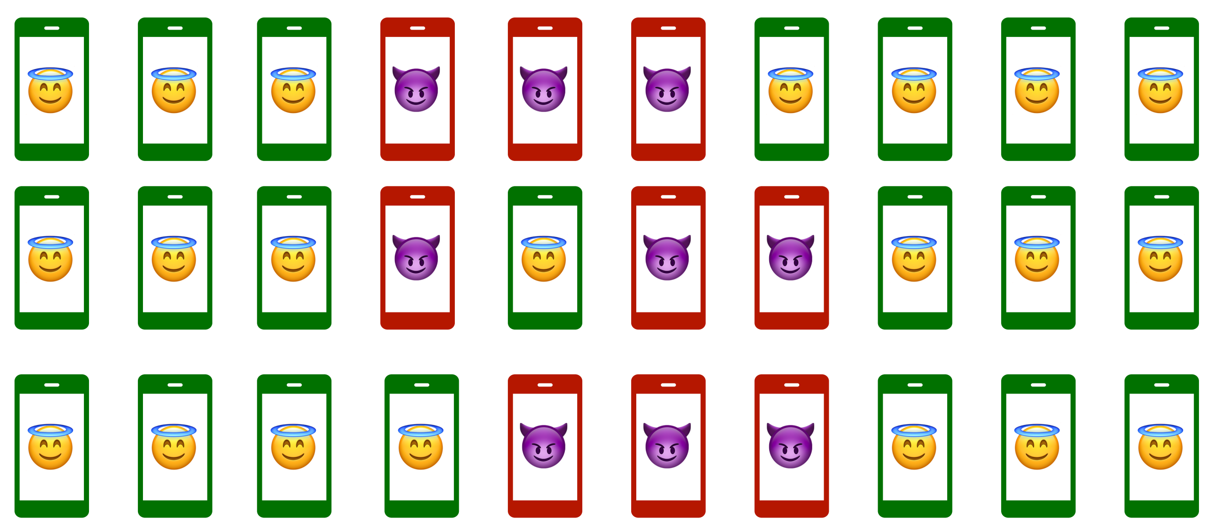
- **Population**: 200,000
- **Dishonesty base rate**: 0.005
- **Target participants**: 200



Lotto prevents arbitrary manipulation

What can be **proven**:

Population

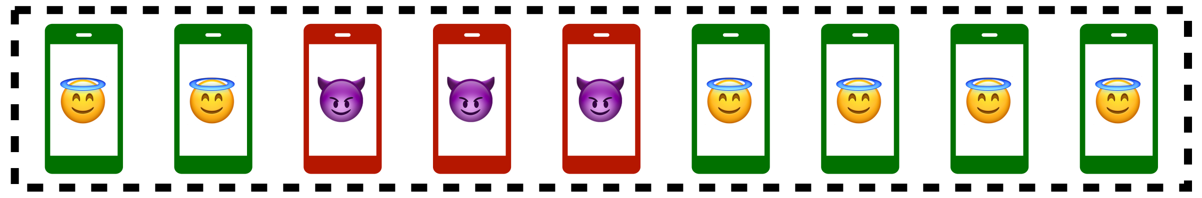


Base rate of dishonest clients

Lotto

Align

w/ high prob.

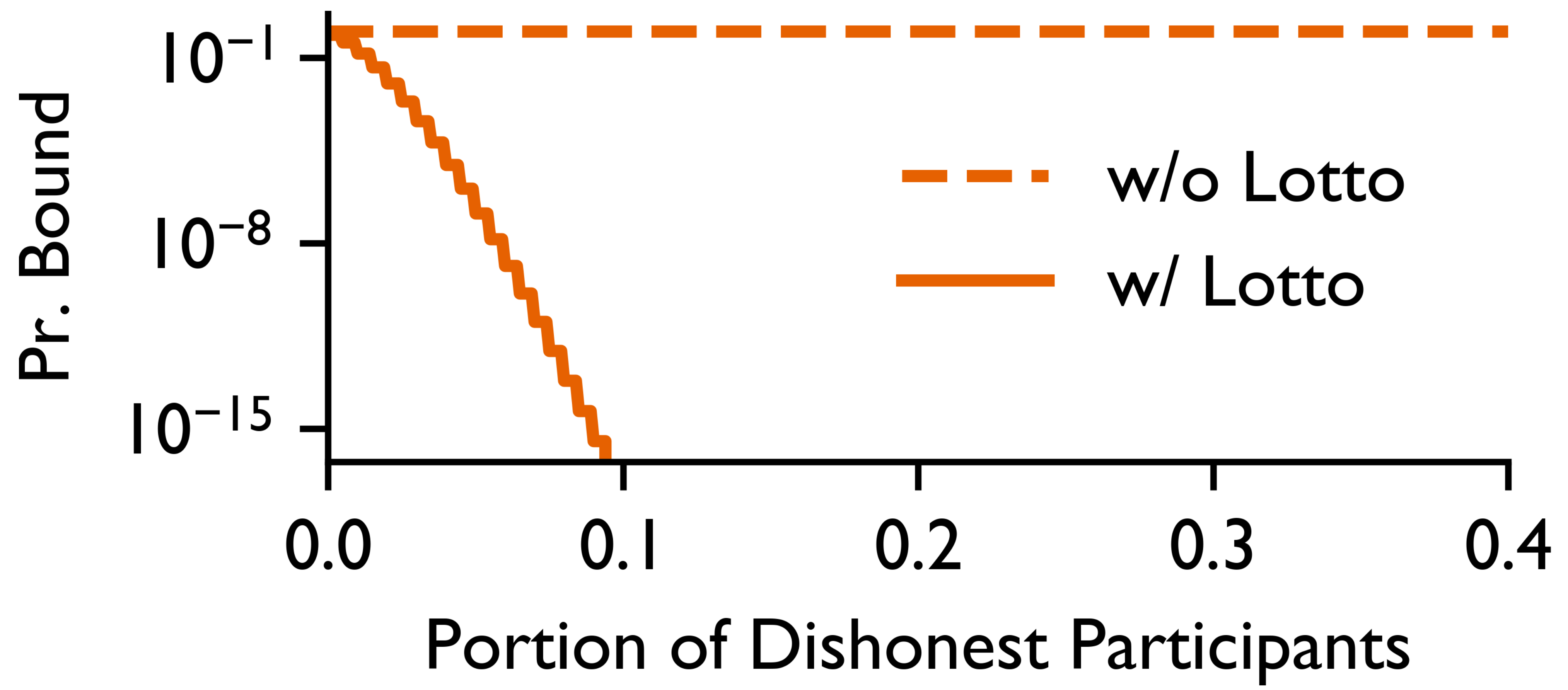


Participants

Fraction of dishonest clients

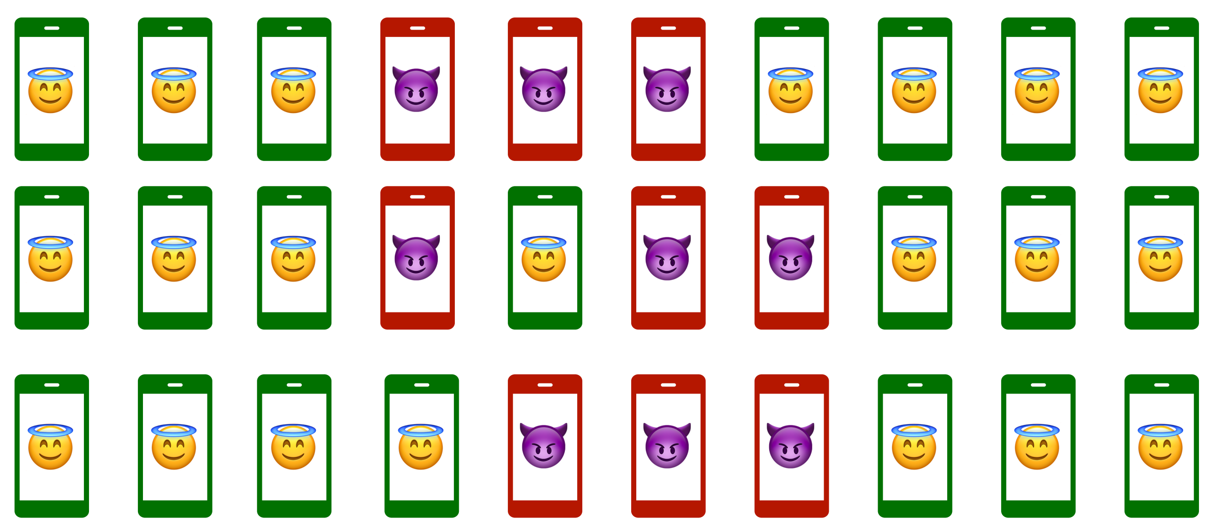
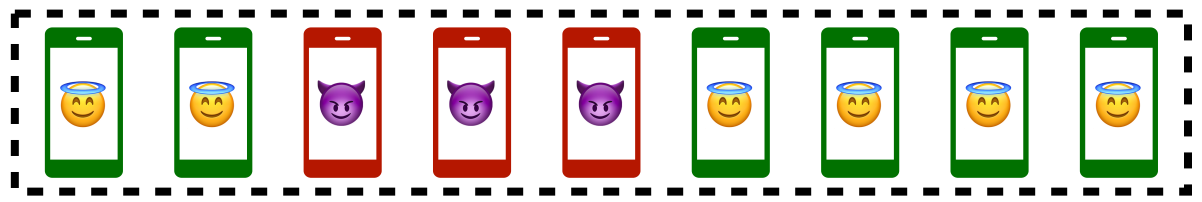
Example

- **Population**: 200,000
- **Dishonesty base rate**: 0.005
- **Target participants**: 200



Lotto prevents arbitrary manipulation

What can be **proven**:

Population  Lotto →  Participants

Base rate of dishonest clients

Align

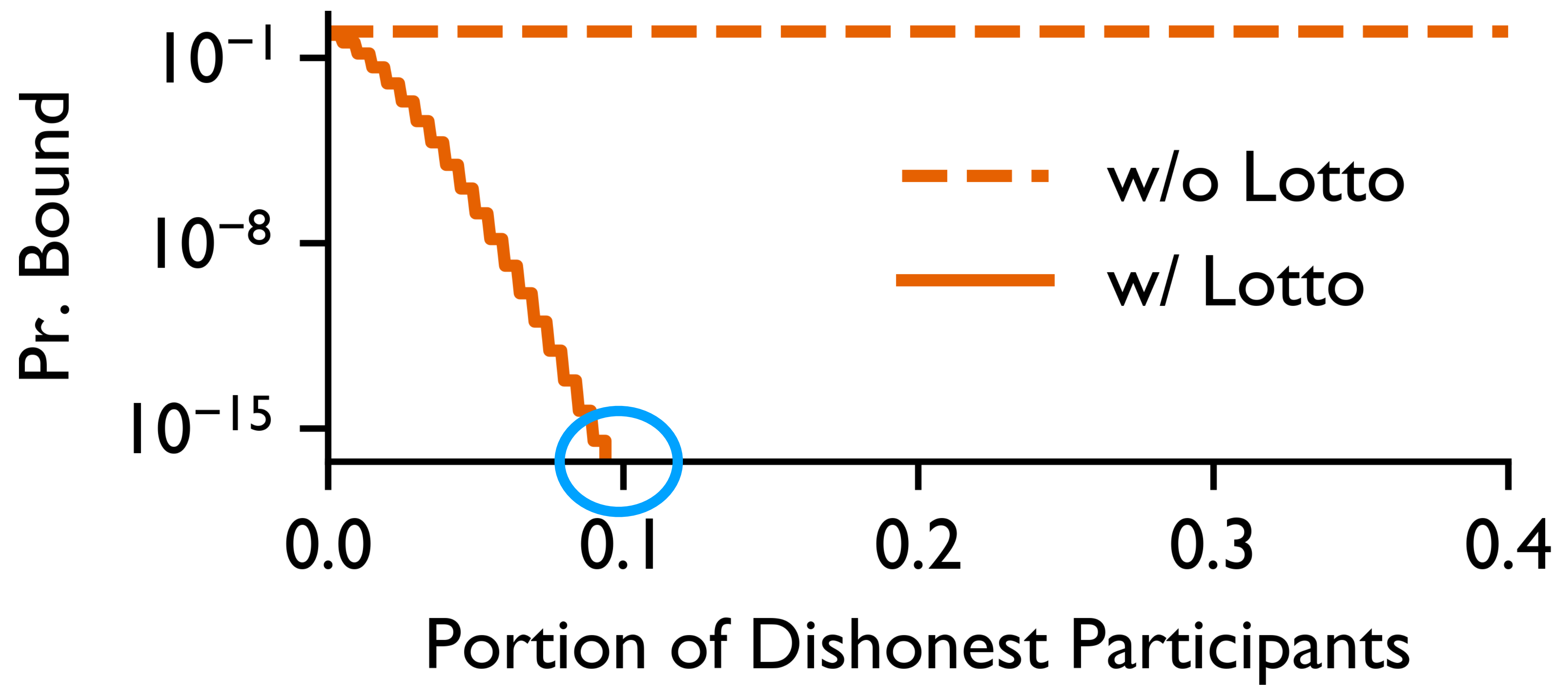
←

w/ high prob.

Fraction of dishonest clients

Example

- **Population**: 200,000
- **Dishonesty base rate**: 0.005
- **Target participants**: 200



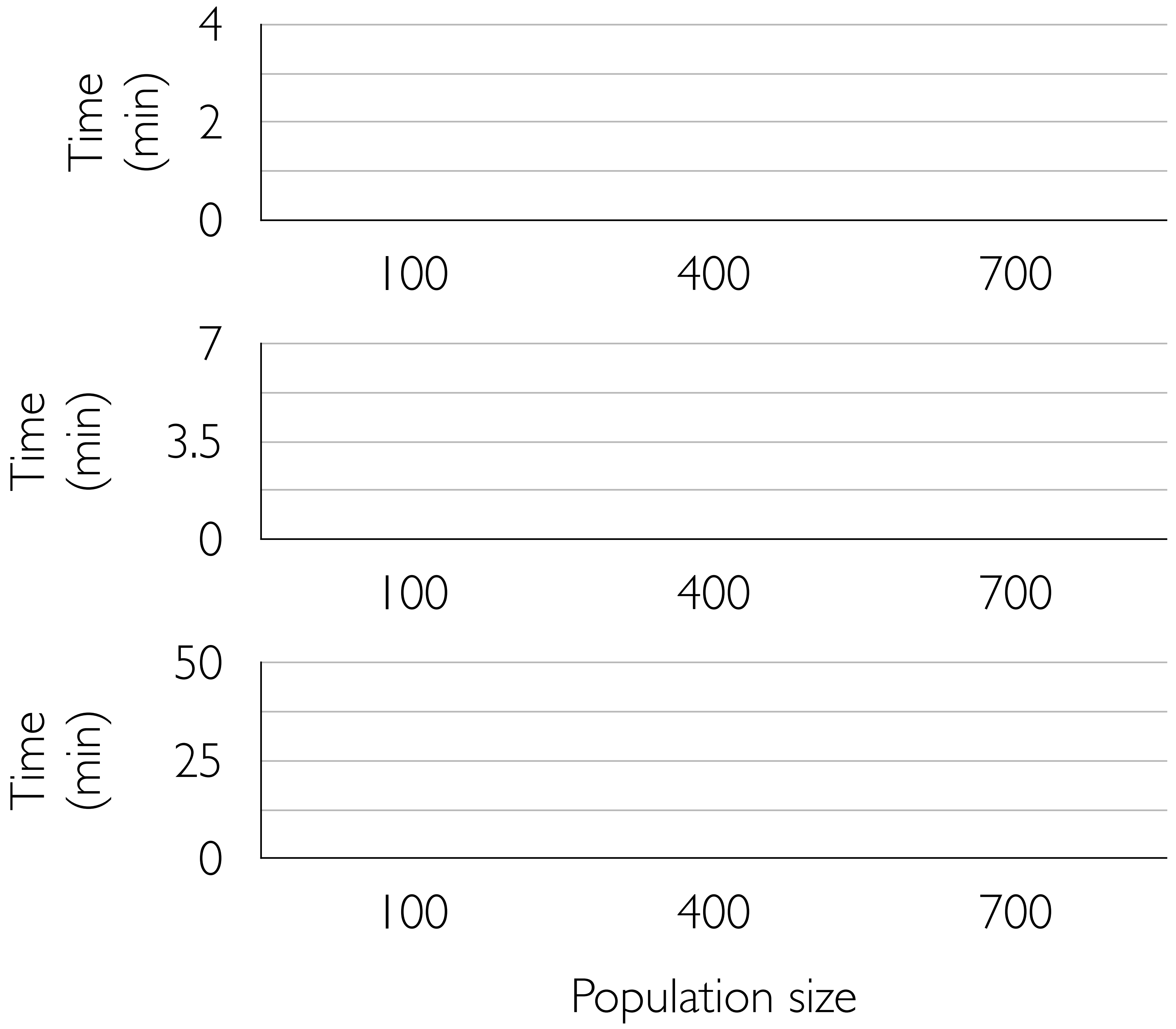
Lotto induces no or mild overhead

Lotto induces no or mild overhead

FEMNIST
@CNN

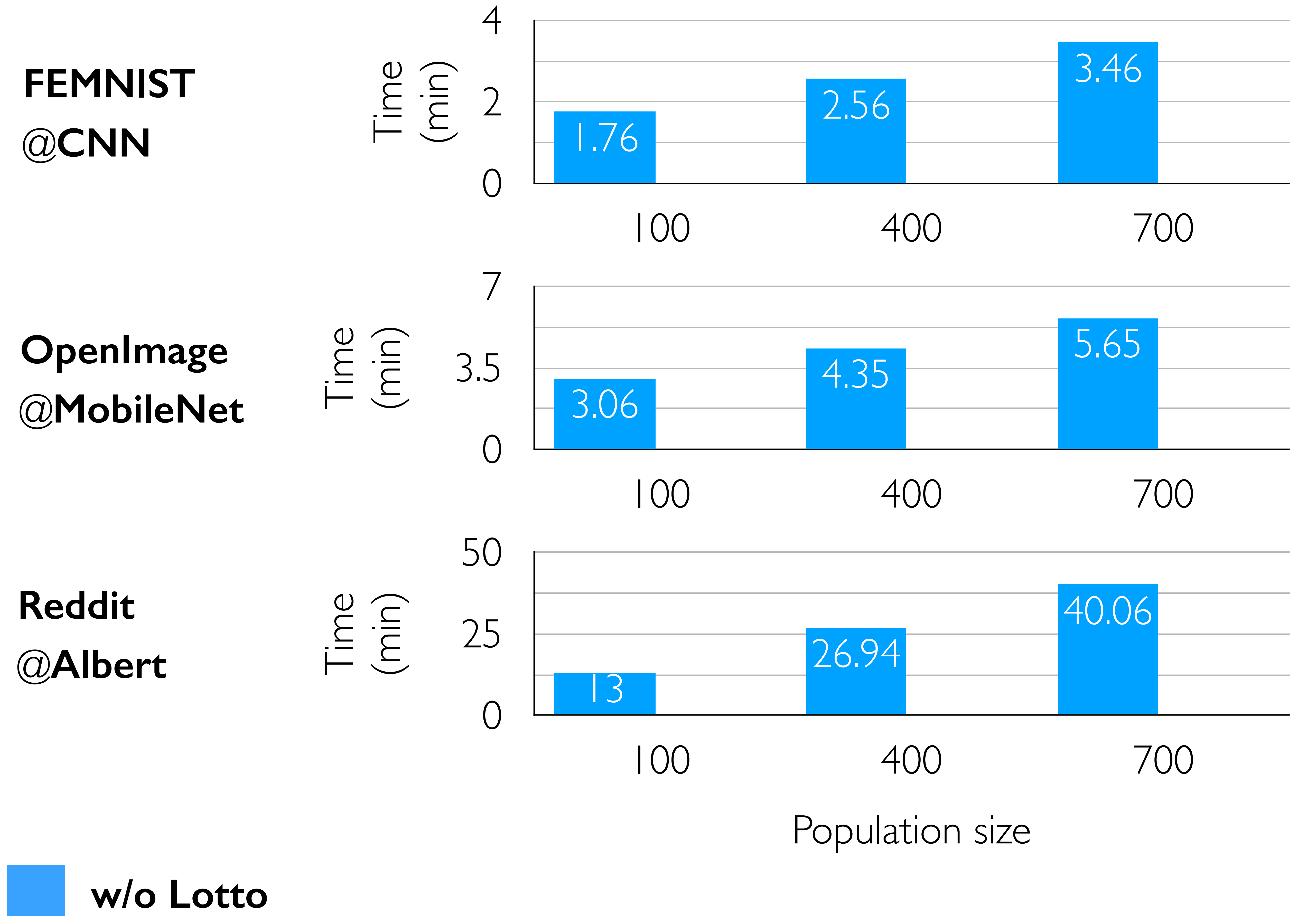
OpenImage
@MobileNet

Reddit
@Albert



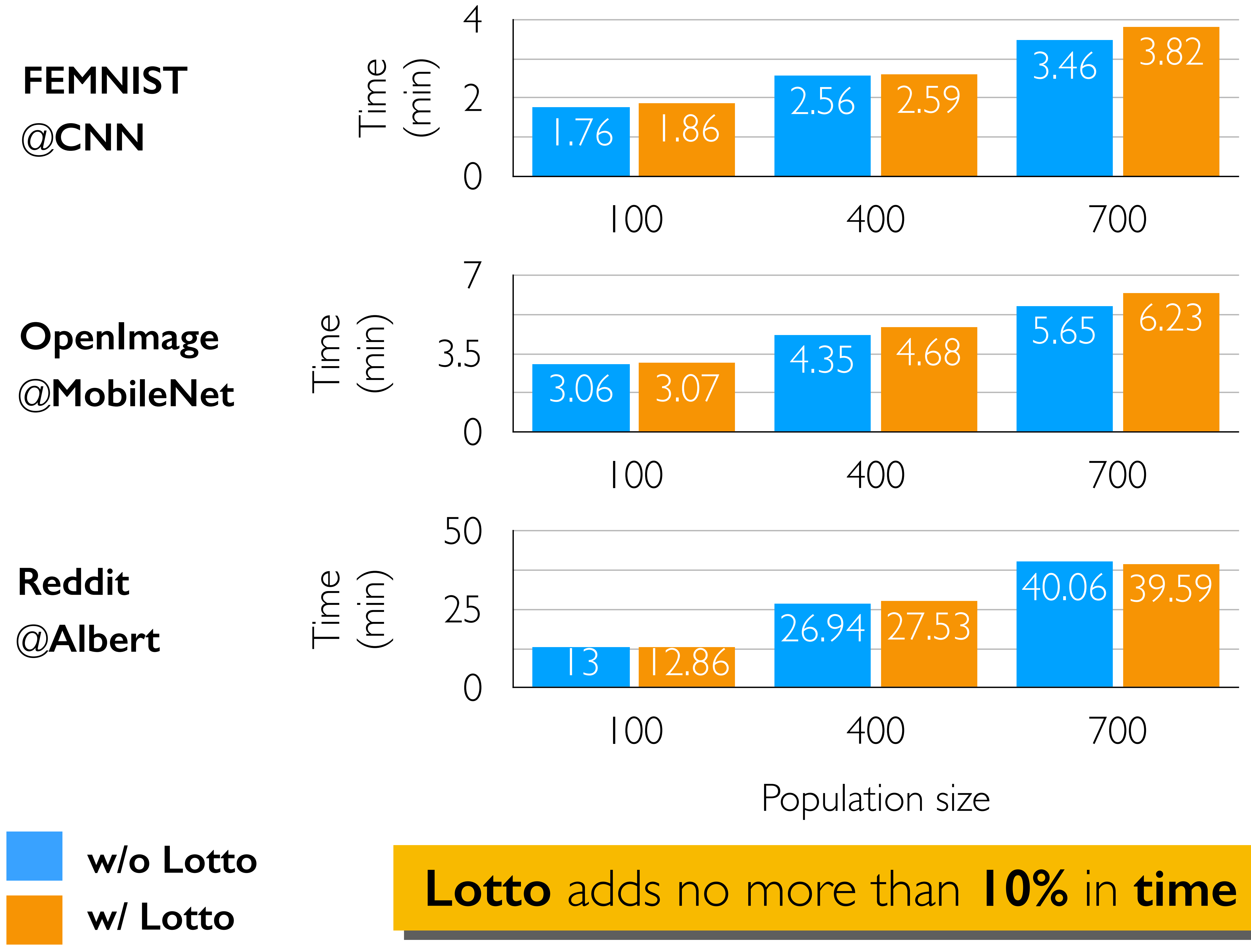
[1]Random selection as an example. See results for informed selection in the paper.

Lotto induces no or mild overhead



[1]Random selection as an example. See results for informed selection in the paper.

Lotto induces no or mild overhead



Lotto adds no more than **10%** in **time**

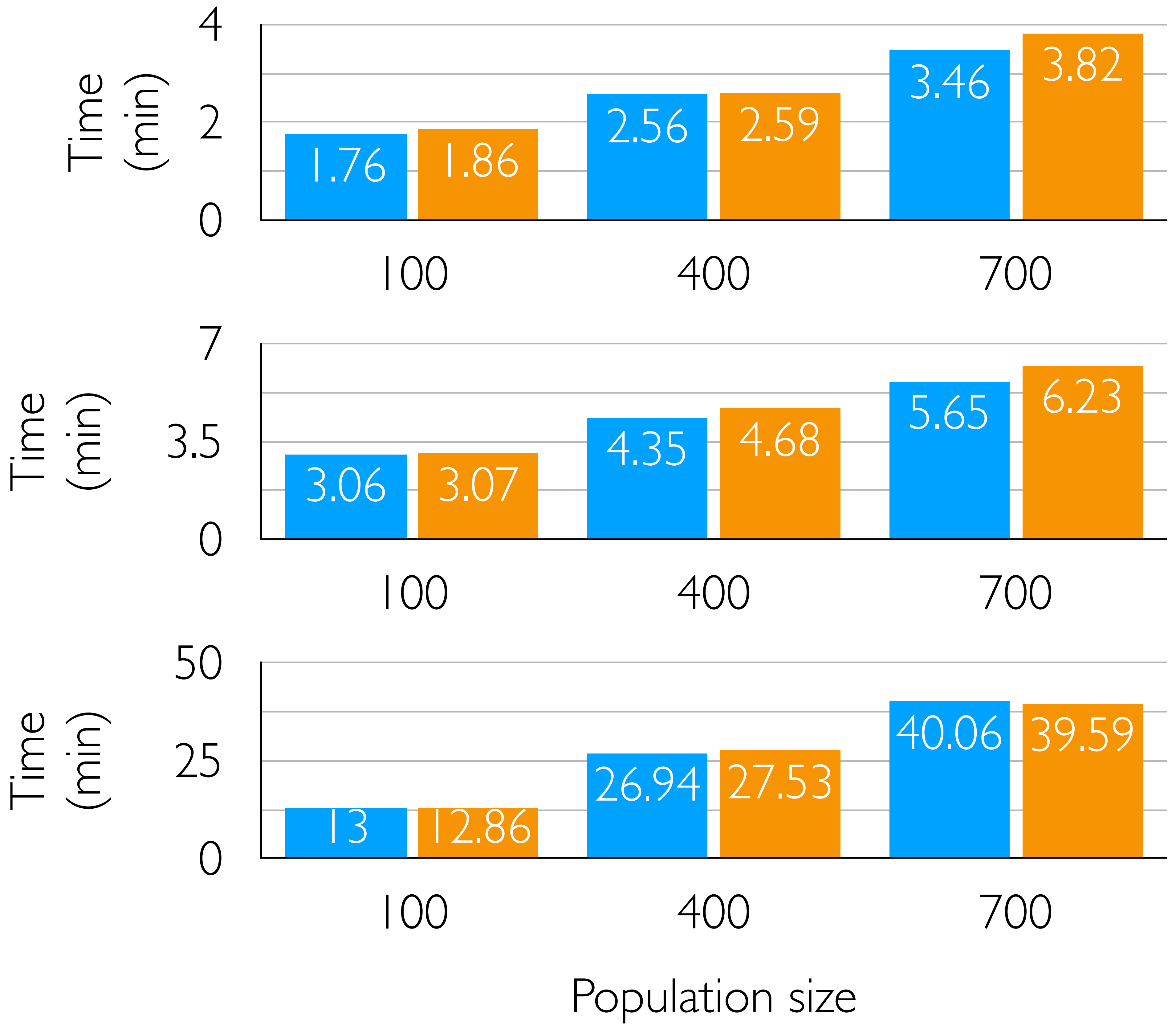
[1]Random selection as an example. See results for informed selection in the paper.

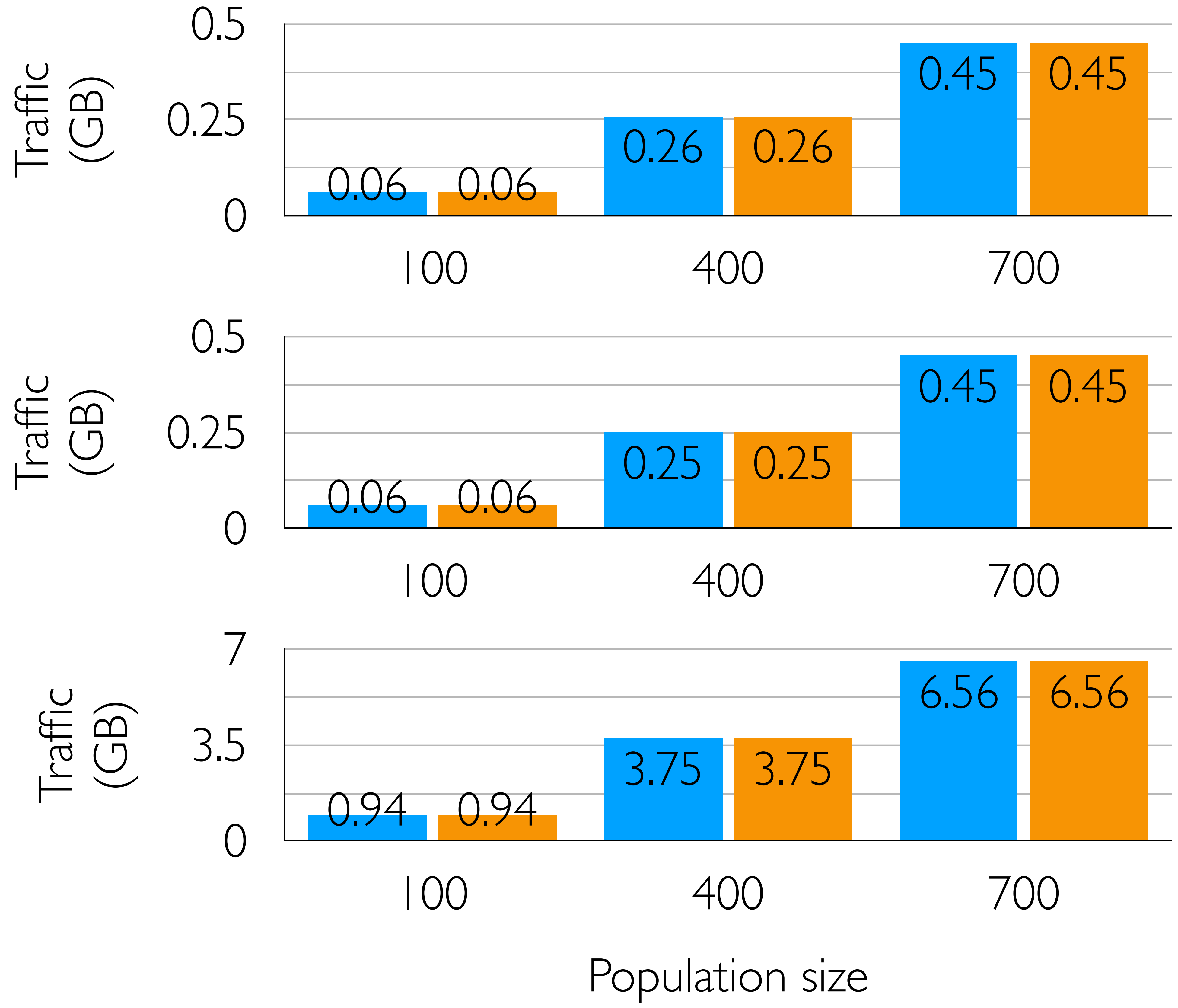
Lotto induces no or mild overhead

FEMNIST @CNN

OpenImage @MobileNet

Reddit @Albert





w/o Lotto

w/ Lotto

Lotto adds no more than **10%** in **time**

Lotto costs **negligible** in **network**

[1]Random selection as an example. See results for informed selection in the paper.

Lotto functions as insecure selectors

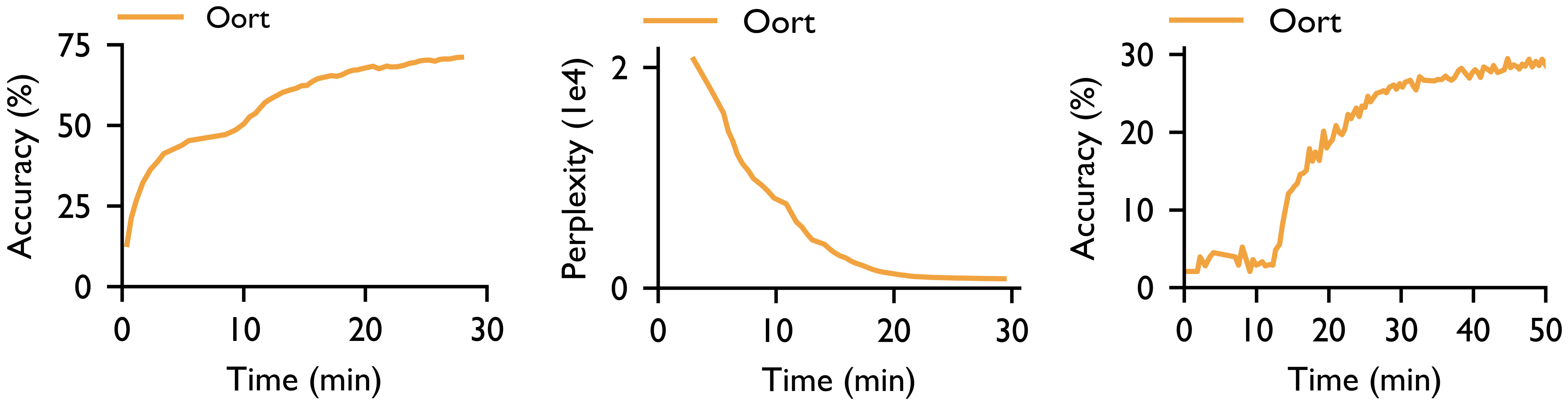
Lotto functions as insecure selectors

Oort[1] → State-of-the-art **informed** selector: optimized for **time-to-accuracy** of training

[1] Lai et al. "Oort: Efficient Federated Learning via Guided Participant Selection", In OSDI '21

Lotto functions as insecure selectors

Oort[1] → State-of-the-art **informed** selector: optimized for **time-to-accuracy** of training

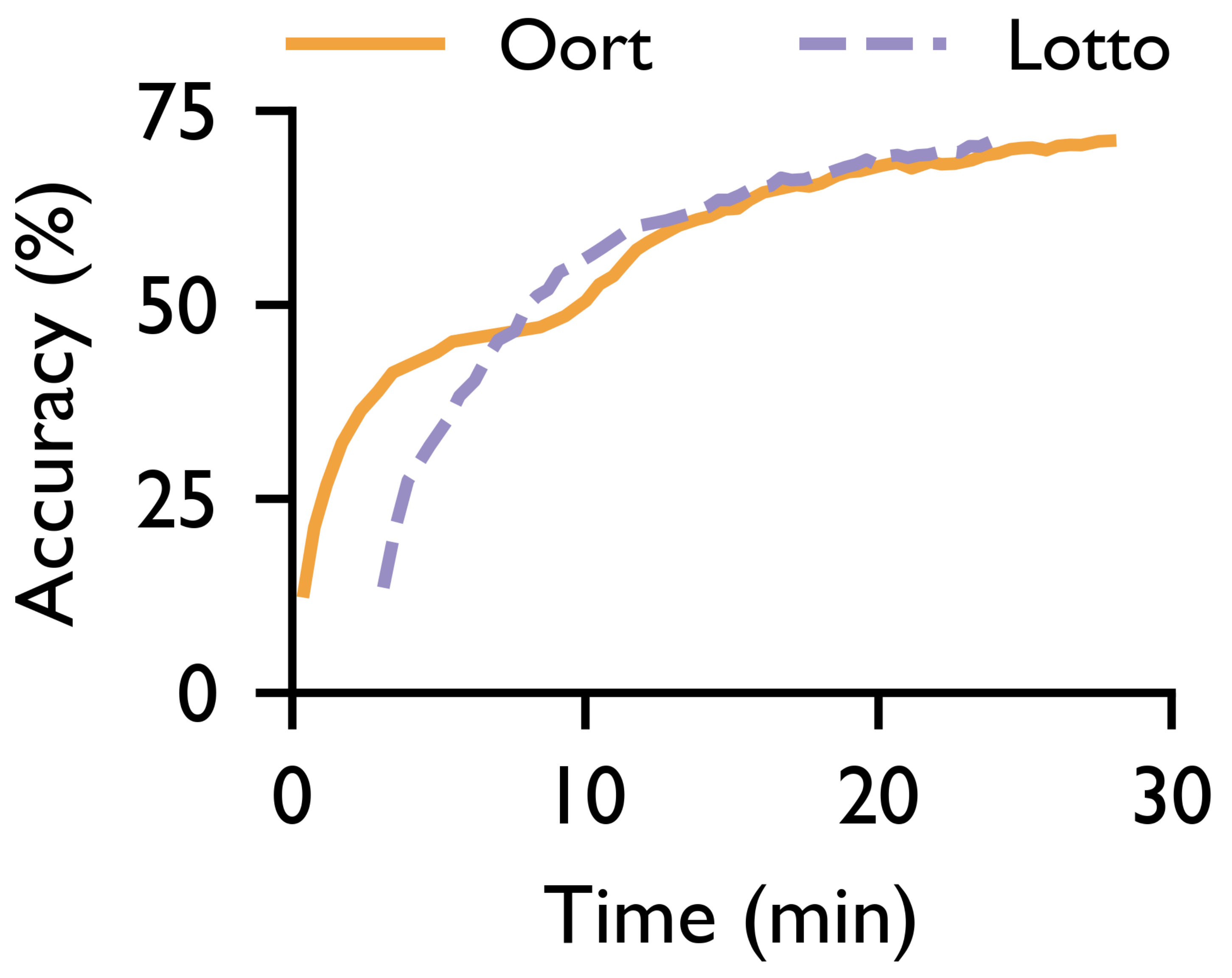


FEMNIST@CNN **OpenImage@MobileNet** **Reddit@Albert**

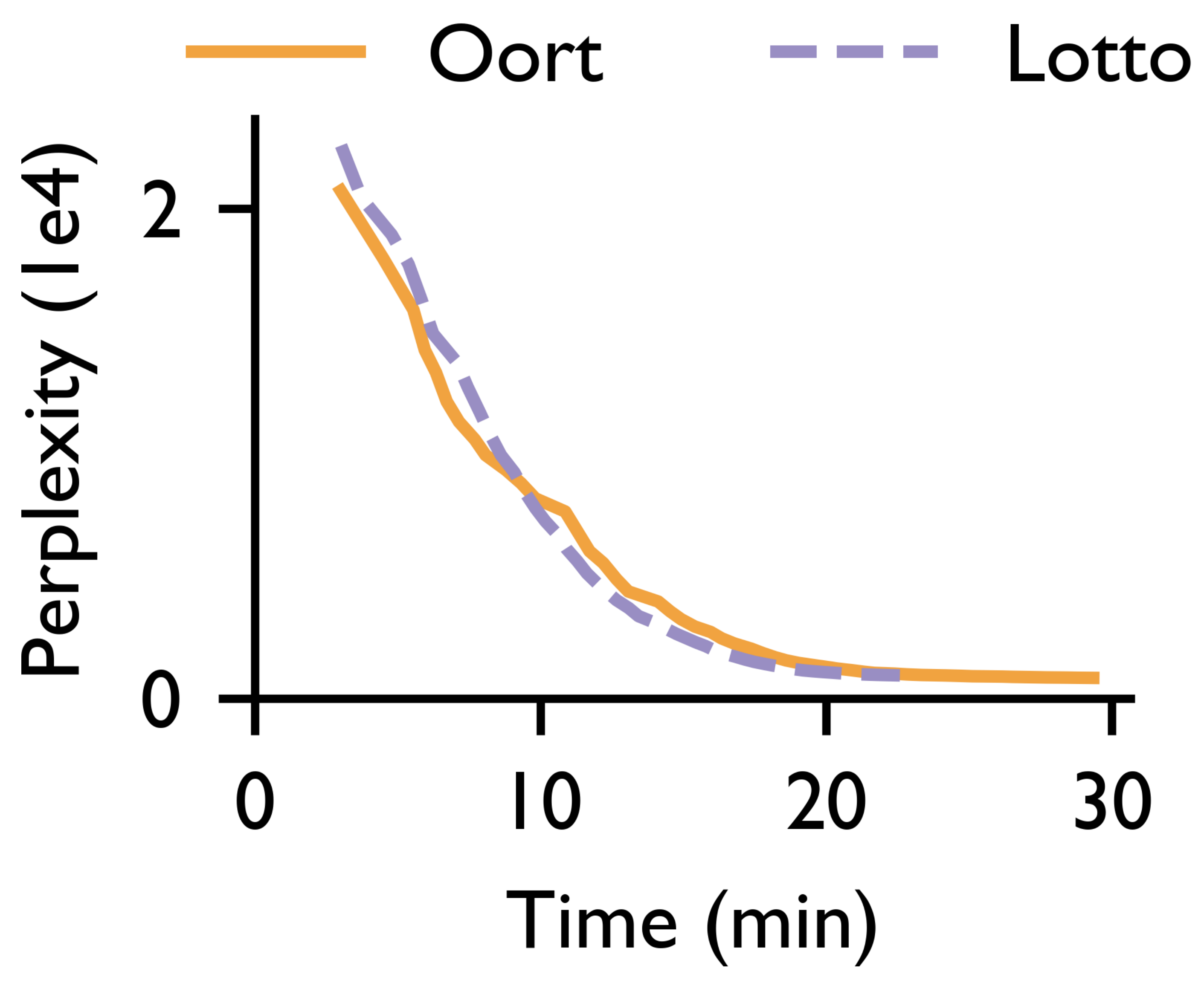
[1]Lai et al. "Oort: Efficient Federated Learning via Guided Participant Selection", In OSDI '21

Lotto functions as insecure selectors

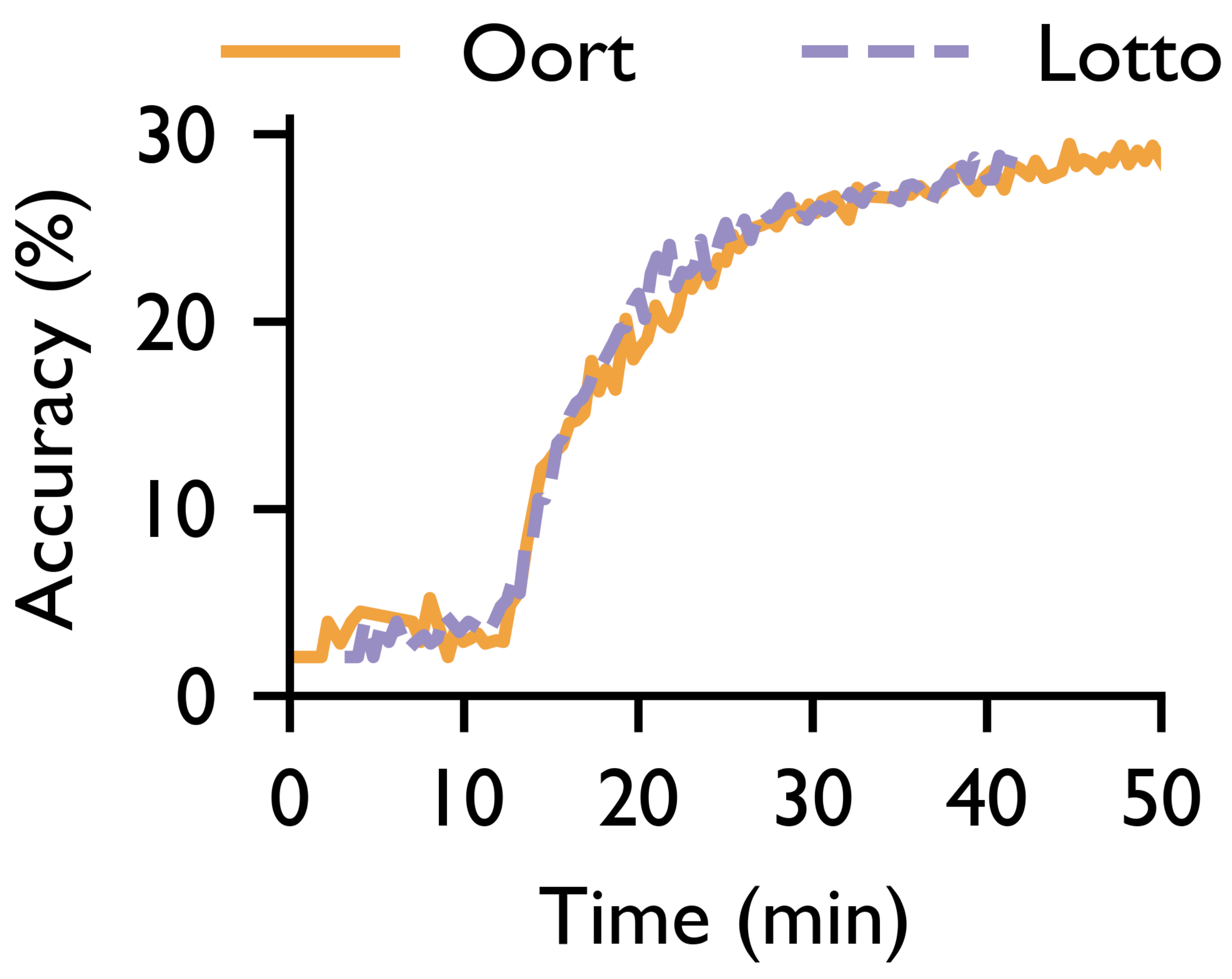
Oort[1] → State-of-the-art **informed** selector: optimized for **time-to-accuracy** of training



FEMNIST@CNN



OpenImage@MobileNet



Reddit@Albert

Lotto well approximate Oort with **no cost in time-to-accuracy** performance

[1] Lai et al. "Oort: Efficient Federated Learning via Guided Participant Selection", In OSDI '21

Lotto: Results summary

Functionality

Support both **random (exact)** and **informed (well approximated)** selection

Security

Theoretical guarantee (tight probability bound) of preventing manipulation

Efficiency

Mild **runtime overhead (≤10%)** with no **network cost (<1%)**

github.com/SamuelGong/Lotto

Thank you

zjiangaj@connect.ust.hk