ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing With Imperfect Randomness

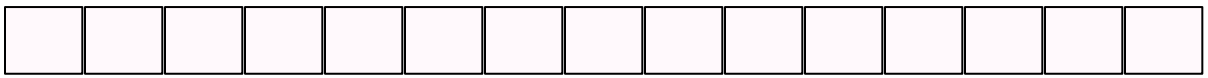
Shafi Goldwasser, Madhu Sudan and Vinod Vaikuntanathan

Presentation by: Daniel Thomas

1. Randomness

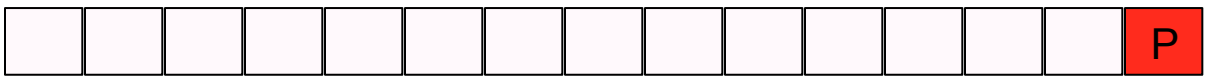
1. Randomness

QuickSort



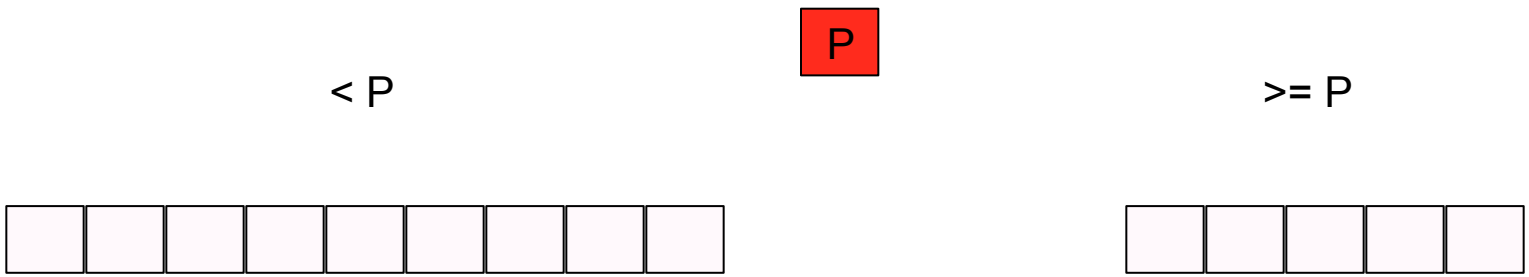
1. Randomness

QuickSort



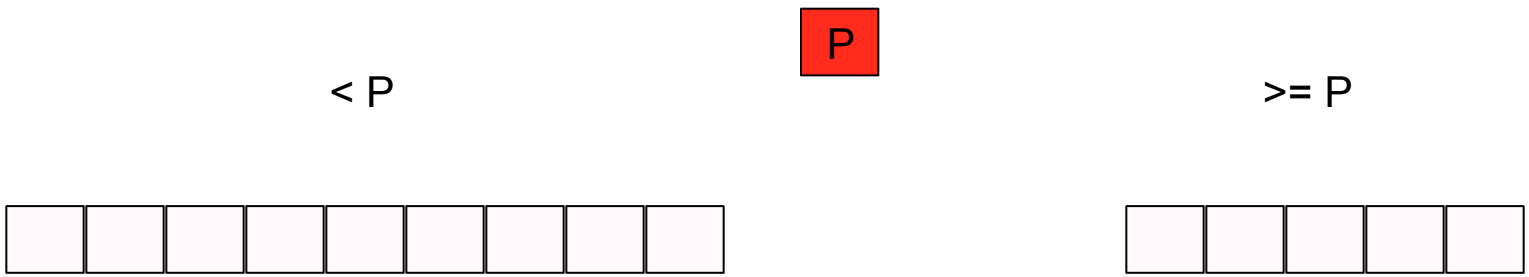
1. Randomness

QuickSort



1. Randomness

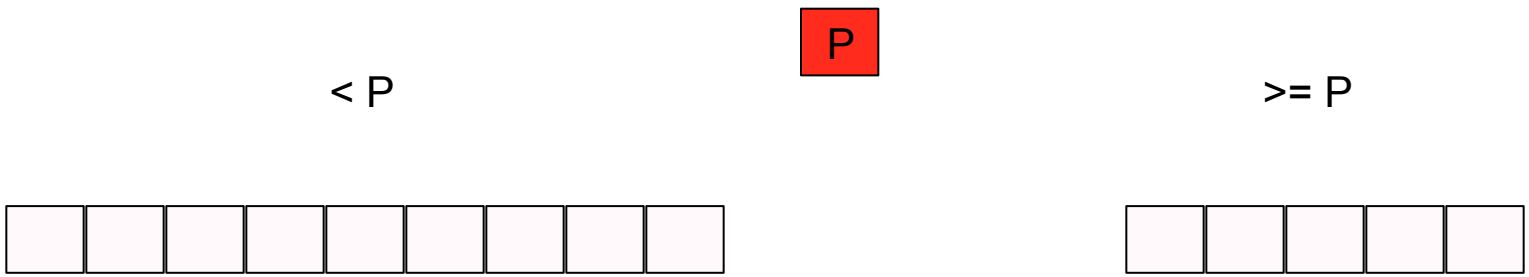
QuickSort



What If Already Sorted?

1. Randomness

QuickSort



What If Already Sorted?

→ Randomized QuickSort

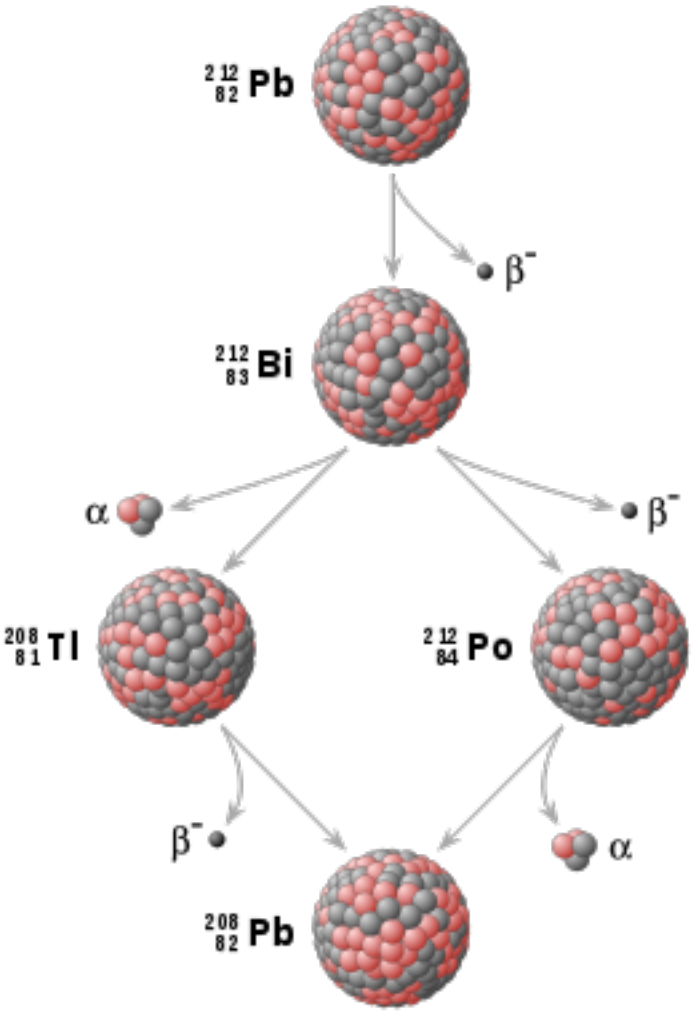
1. Randomness

Real Randomness?
(Strong Randomness)

1. Randomness

Real Randomness?
(Strong Randomness)

Atomic Decay?



1. Random

Real Ran
(Strong R

Atomic D

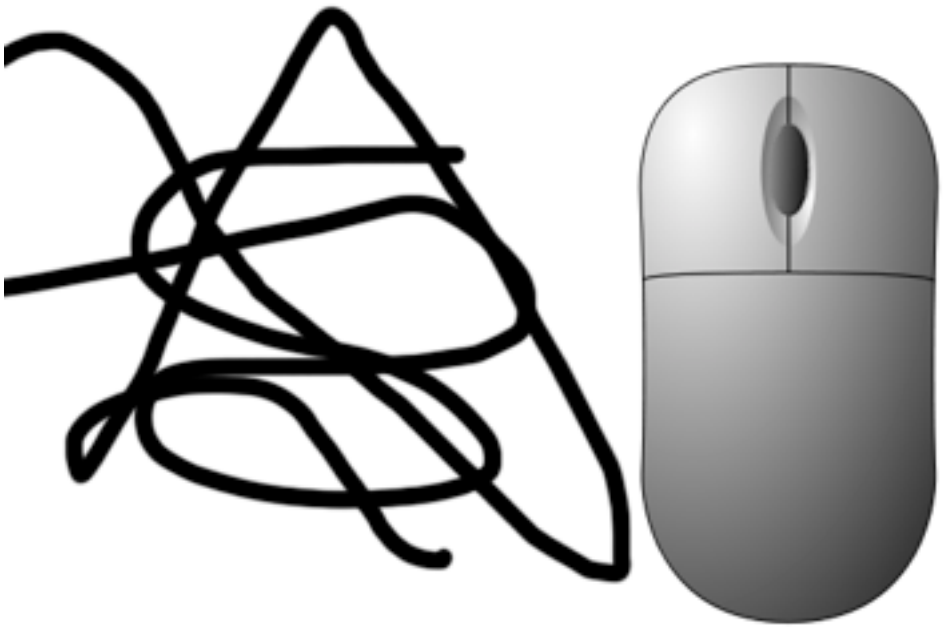
1. Randomness

Pseudorandomness?
(Weak Randmoness?)

1. Randomness

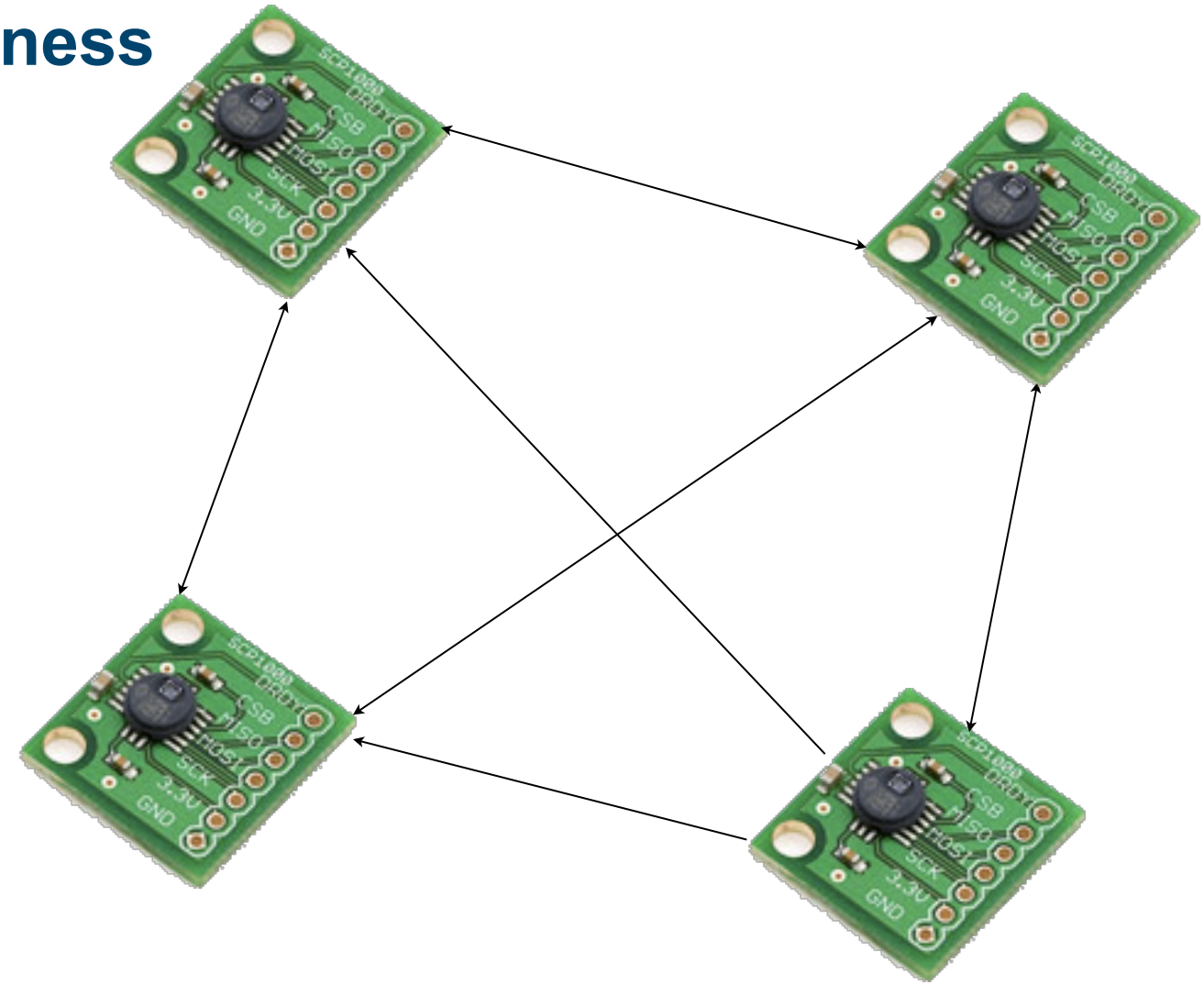
Pseudorandomness?
(Weak Randmoness?)

Get Randomness From System?



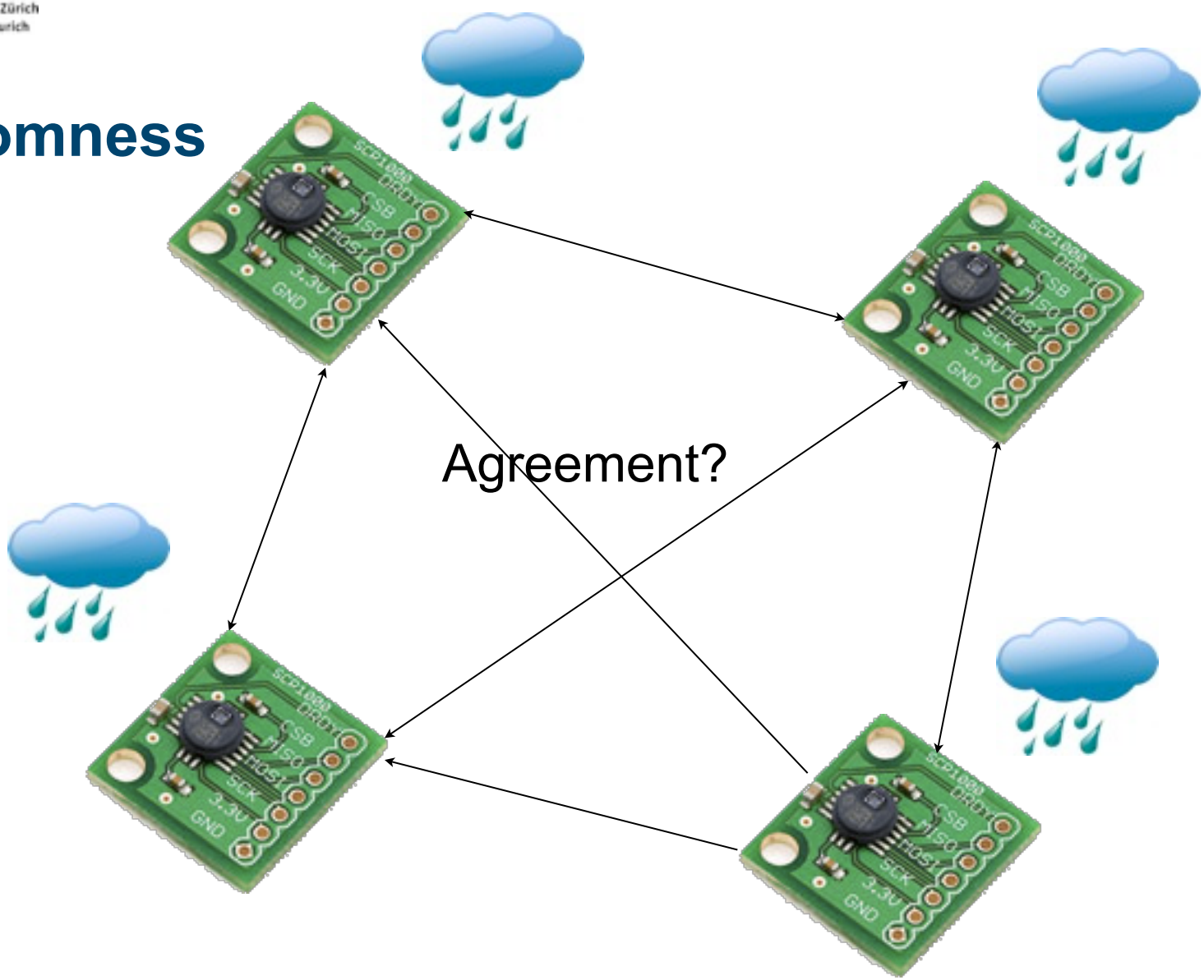
1. Randomness

Setting



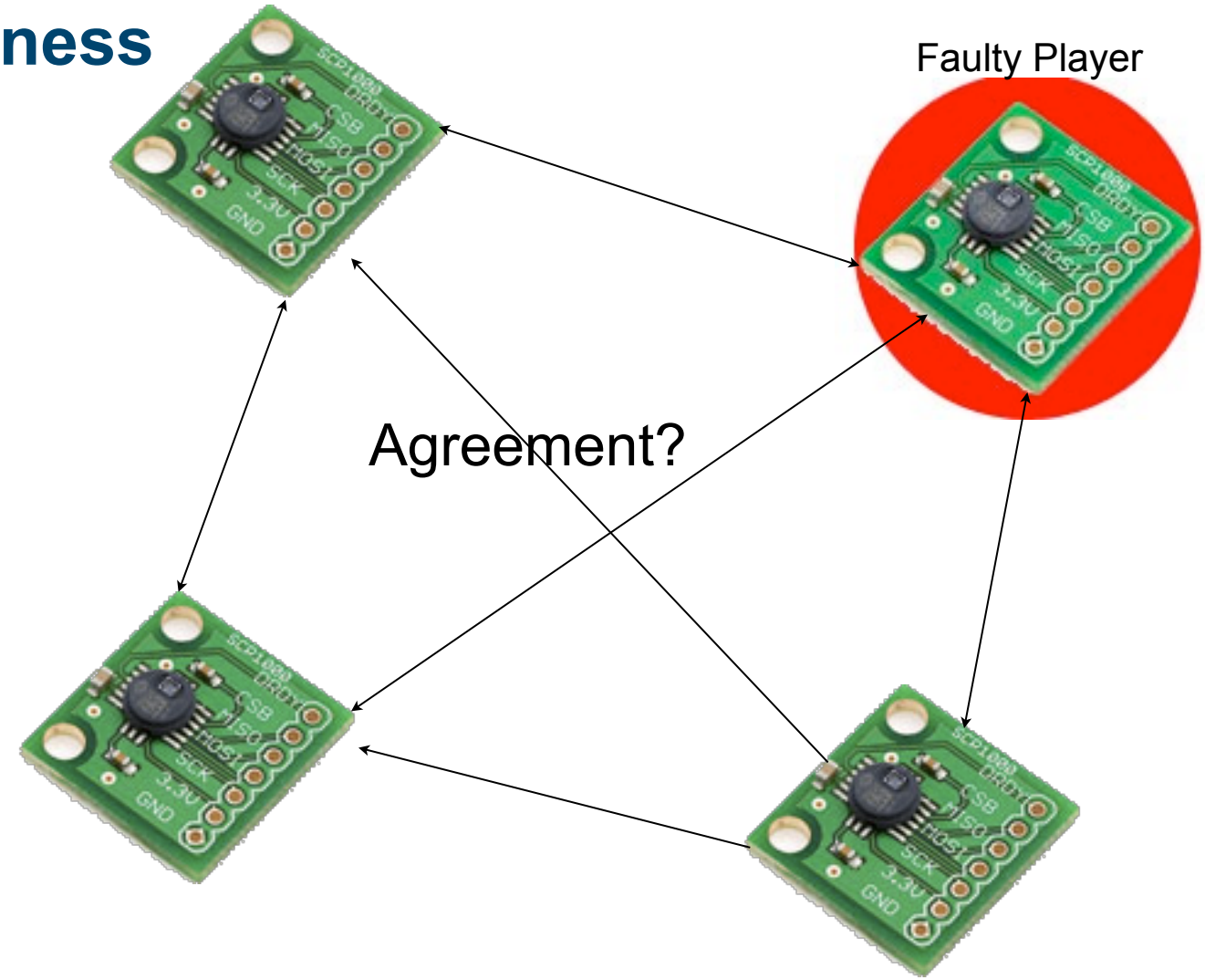
1. Randomness

Setting



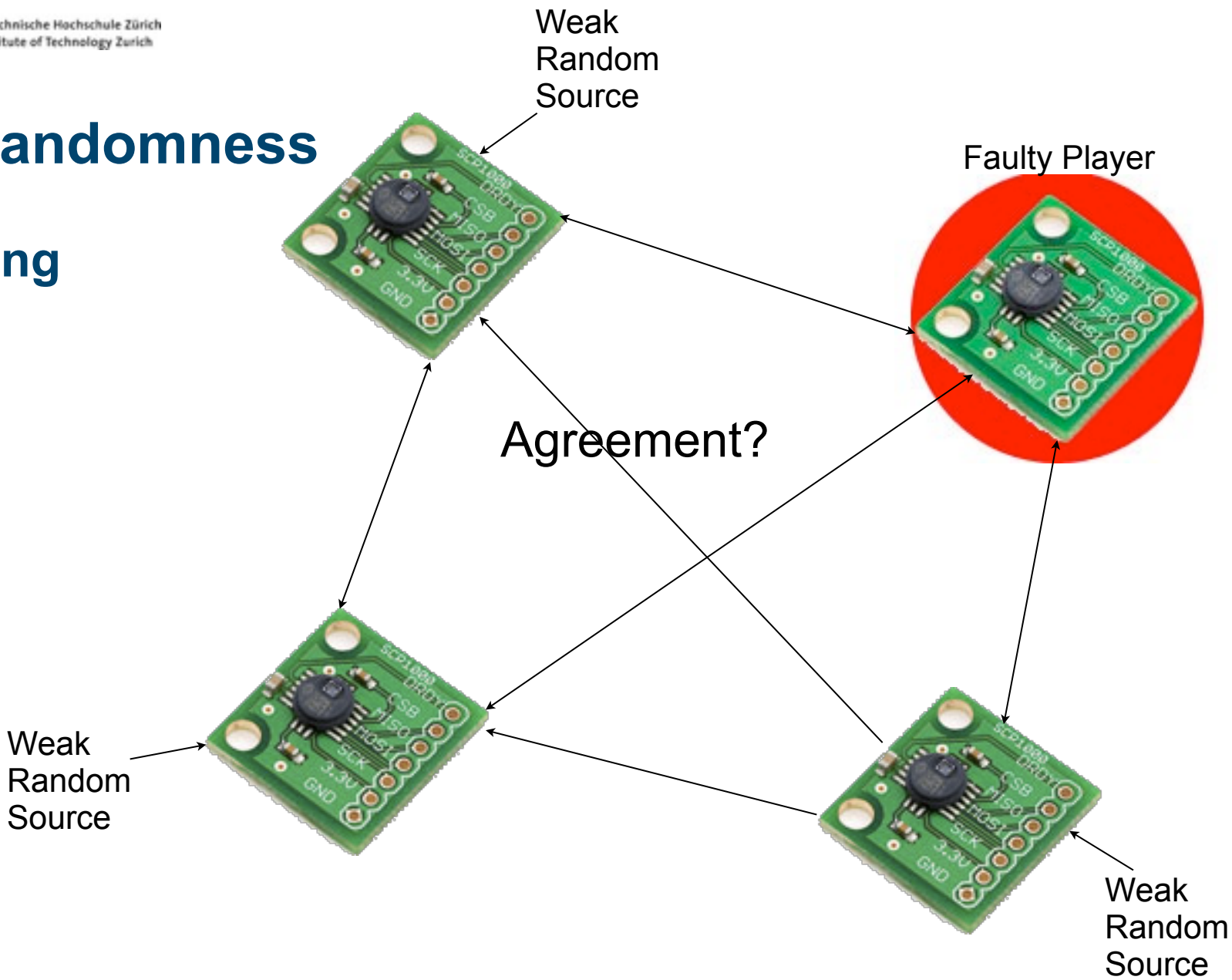
1. Randomness

Setting

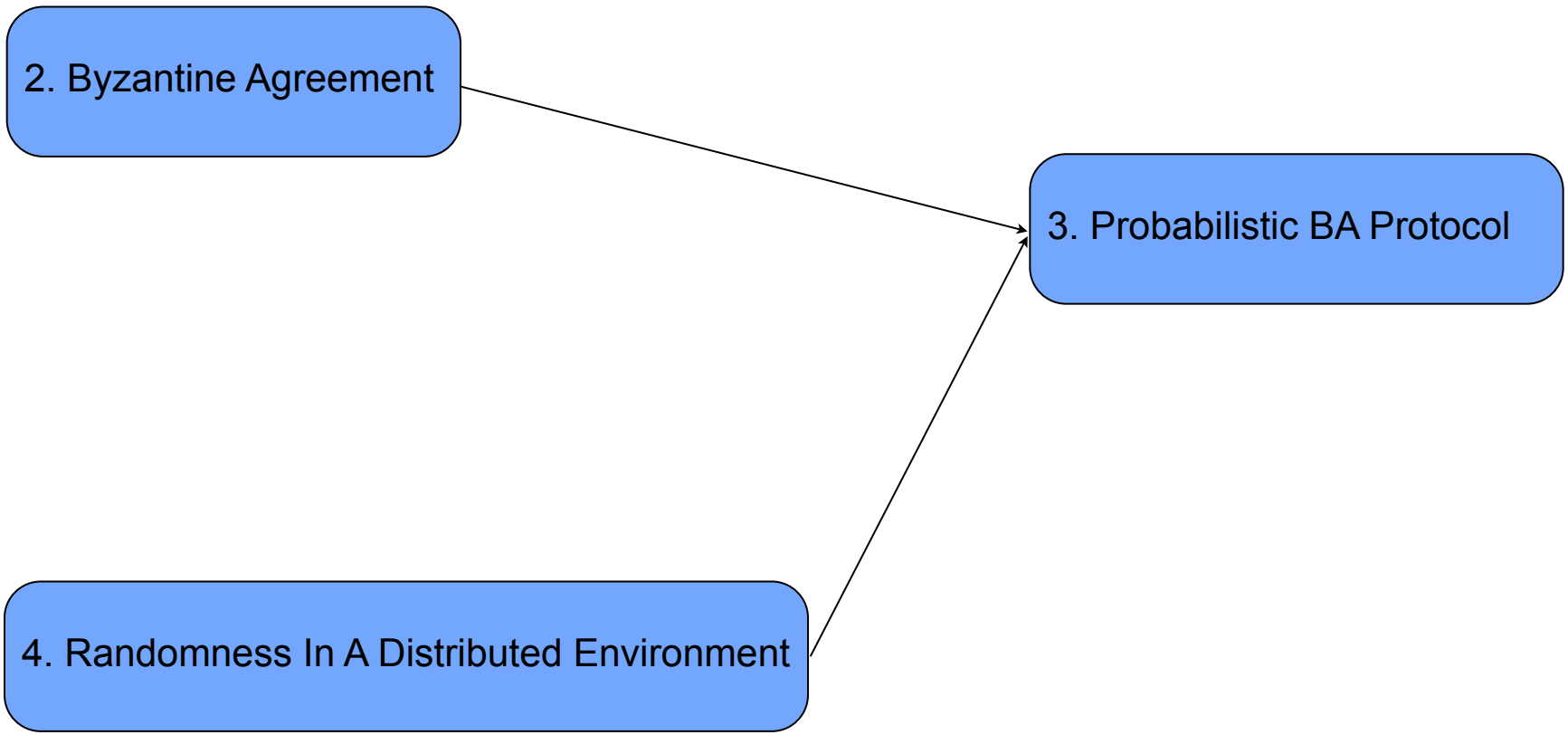


1. Randomness

Setting



1. Randomness, Overview



2. Byzantine Agreement (single fault case)

2. Byzantine Agreement (single fault case)

2. Byzantine Agreement (single fault case)

A. The non-faulty generals compute exactly the same vector

B. The element of this vector corresponding to a given non-faulty general is the private value/ opinion of this general

2. Byzantine Agreement (single fault case)

Works if n >= 3m + 1
And for m+1 rounds are required.

3. Probabilistic Byzantine Agreement Protocols

Deterministic Algorithm	Randomized Algorithm A Simple and Efficient Randomized Byzantine Agreement
m+1 rounds	O(m/log n) rounds

3. Probabilistic Byzantine Agreement Protocols

Deterministic Algorithm	Randomized Algorithm A Simple and Efficient Randomized Byzantine Agreement
m+1 rounds	O(m/log n) rounds

Randomized Algorithm Expects Players To Have Real Random Source!

4. Randomness In A Distributed Environment

Formalizing Our Randomness Model

4. Randomness In A Distributed Environment

Formalizing Our Randomness Model

- Strong Randomness: U_m
- Weak Randomness: (k, δ)-weak source
- No Randomness

4. Randomness In A Distributed Environment

Randomness Model

- Each player (sensor) has access to its own weak source that is independent of the sources of all other players

- Is there a way to get real randomness (a string of unbiased and independent random bits) by combining weak random sources?

4. Randomness In A Distributed Environment

Randomness Model

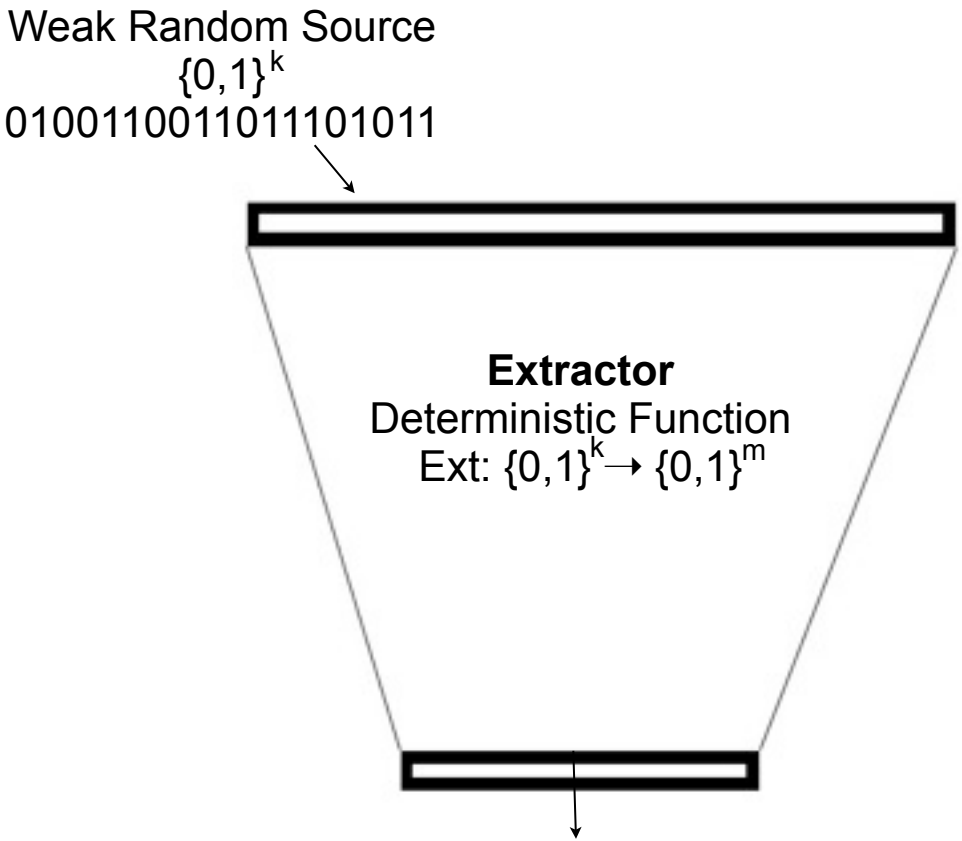
- Each player (sensor) has access to its own weak source that is independent of the sources of all other players

- Is there a way to get real randomness (a string of unbiased and independent random bits) by combining weak random sources?

Extractors!

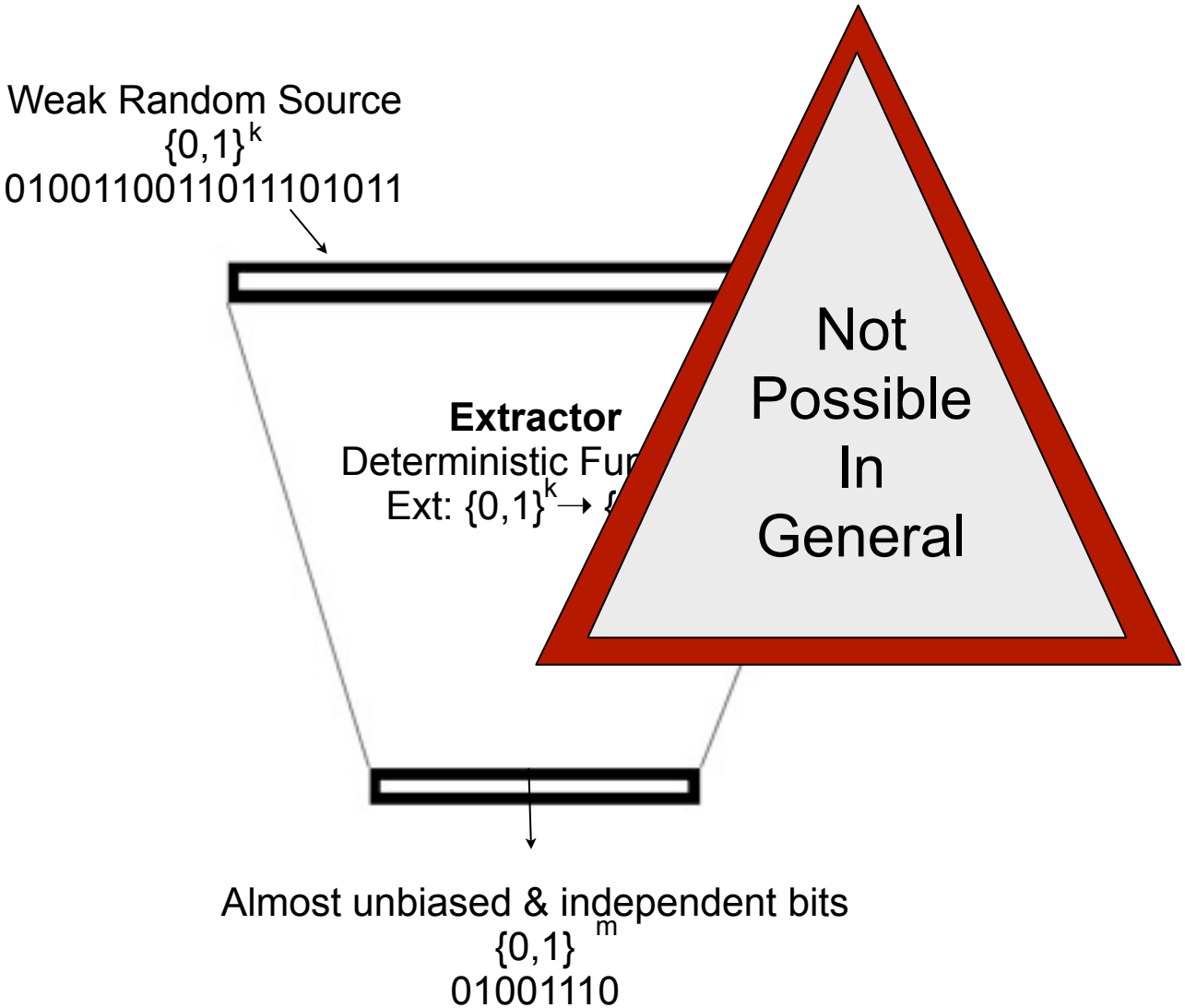
4. Randomness In A Distributed Environment

Extractors



4. Randomness In A Distributed Environment

Extractors



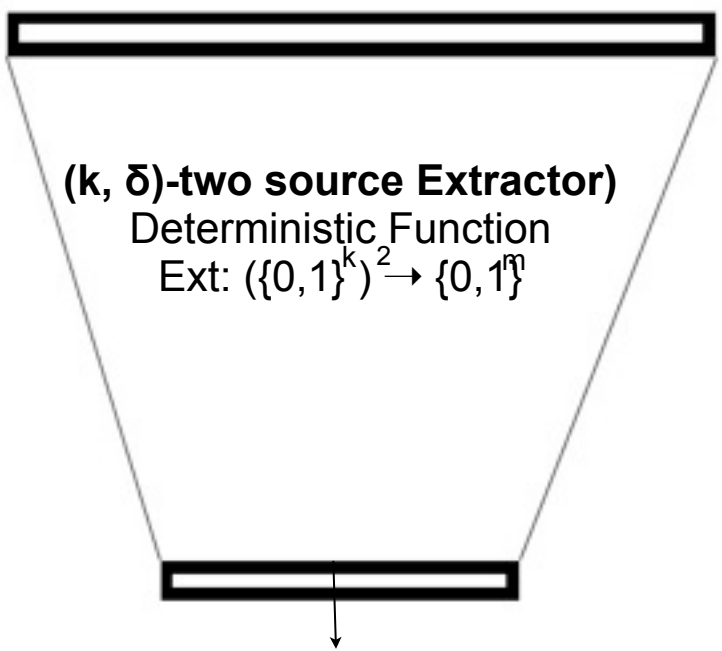
4. Randomness In A Distributed Environment

2 Source Extractors

Weak Random Source
0100110011011101011

←independent→

Weak Random Source
0100111011010011011



Almost unbiased & independent bits
$\{0,1\}^m$
01001110

4. Randomness In A Distributed Environment

A Very Simple Extractor

4. Randomness In A Distributed Environment

A Very Simple Extractor

0 | 1

4. Randomness In A Distributed Environment

A Very Simple Extractor

0	1
0	0

4. Randomness In A Distributed Environment

A Very Simple Extractor

0	1
0	0
1	1

4. Randomness In A Distributed Environment

A Very Simple Extractor

0	1
0	0
1	1
1	0

4. Randomness In A Distributed Environment

A Very Simple Extractor

0	1
0	0
1	1
1	0
0	0

4. Randomness In A Distributed Environment

A Very Simple Extractor

0	1
0	0
1	1
1	0
0	0
1	0

4. Randomness In A Distributed Environment

A Very Simple Extractor

0	1
0	0
1	1
1	0
0	0
1	0
0	1

4. Randomness In A Distributed Environment

A Very Simple Extractor

0	1	0
0	0	
1	1	
1	0	1
0	0	
1	0	1
0	1	0

4. Randomness In A Distributed Environment

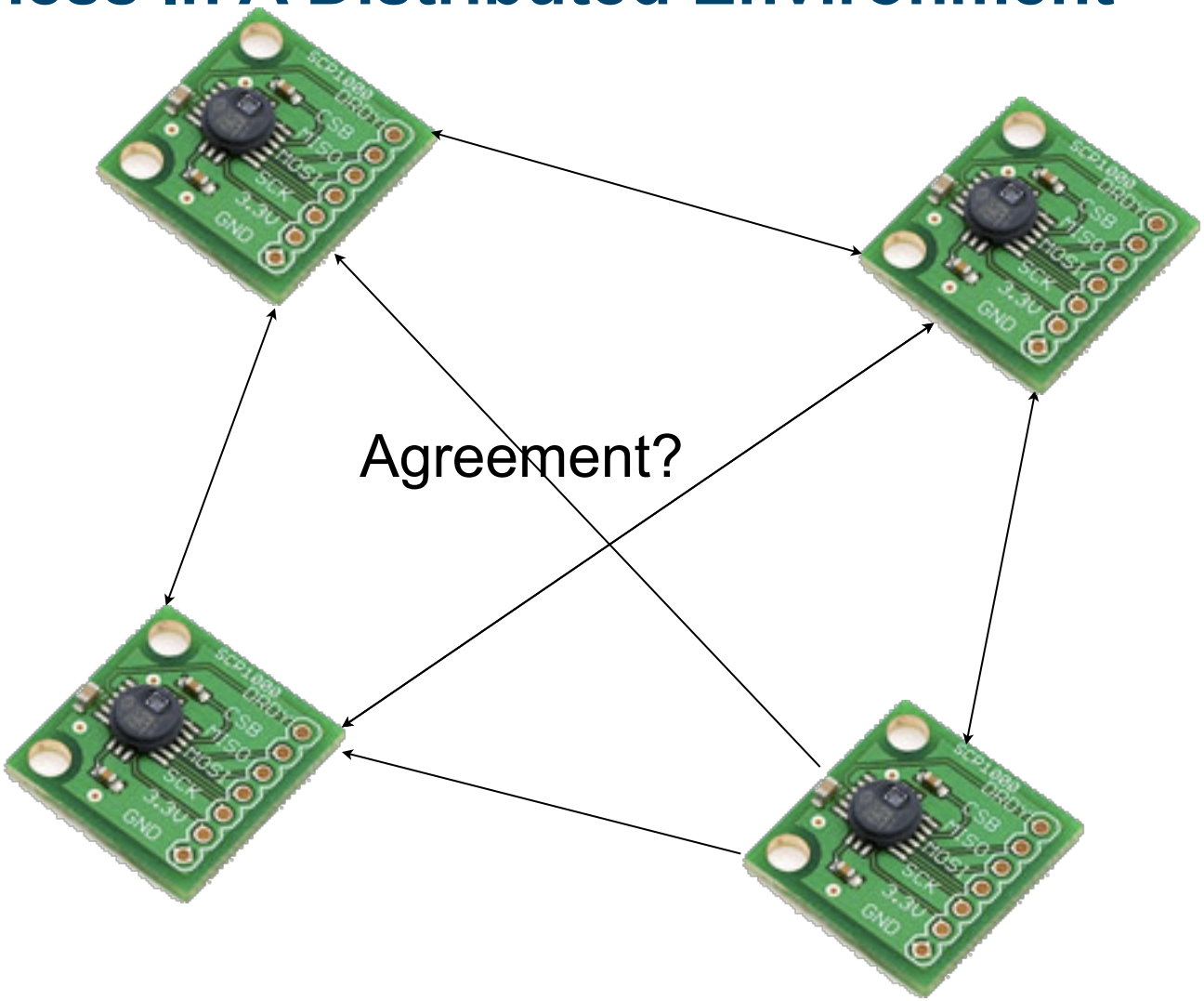
A Very Simple Extractor

0	1	0
0	0	
1	1	
1	0	1
0	0	
1	0	1
0	1	0

Not a (k, δ)-two source Extractor!
(do you see why?)

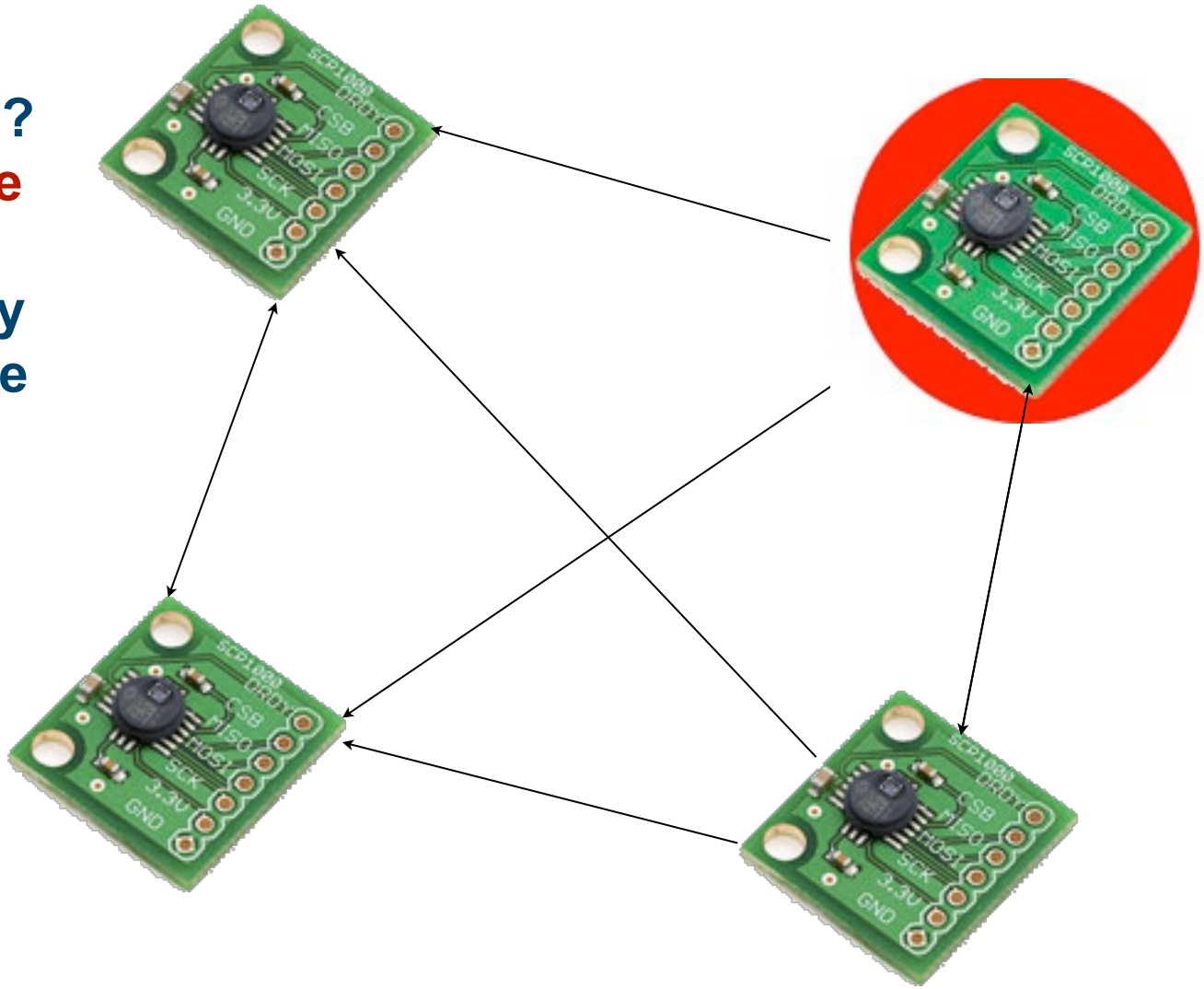
4. Randomness In A Distributed Environment

Is this enough?



4. Randomness In A Distributed Environment

Is this enough?
No, as we have faulty players!
What if they try to "poison" the randomness?



4. Randomness In A Distributed Environment

Immune Extractors

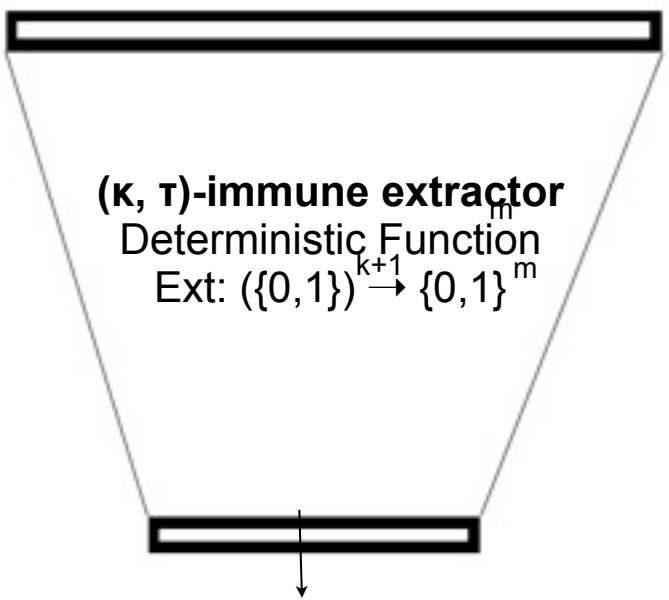
- **Multi-source extractors whose output is random even if an arbitrary subset of the input sources don't send a weak random string (e.g. faulty players) but all sources are independent**

4. Randomness In A Distributed Environment

(κ, τ) Immune Extractors, as a picture

κ + 1 block sources

(k, δ) source (k, δ) source (k, δ) source (k, δ) source



Almost unbiased & independent bits

$\{0,1\}^m$

01001110

4. Randomness In A Distributed Environment

I-Ext: A(t, t-1)-immune extractor

- Let Ext be any (k, δ) two source extractor
- Let X_1^2, X_1^3, ... denote t distinct blocks from the source X_1
- Let X_2, X_3, ... be one block each from the other t sources
- I-Ext($\{X_1^i\}_{i=2}^{t+1}$, X_2, ..., X_{t+1}) = $\oplus_{i=2}^{t+1}$ Ext(X_1^i, X_i)

4. Randomness In A Distributed Environment

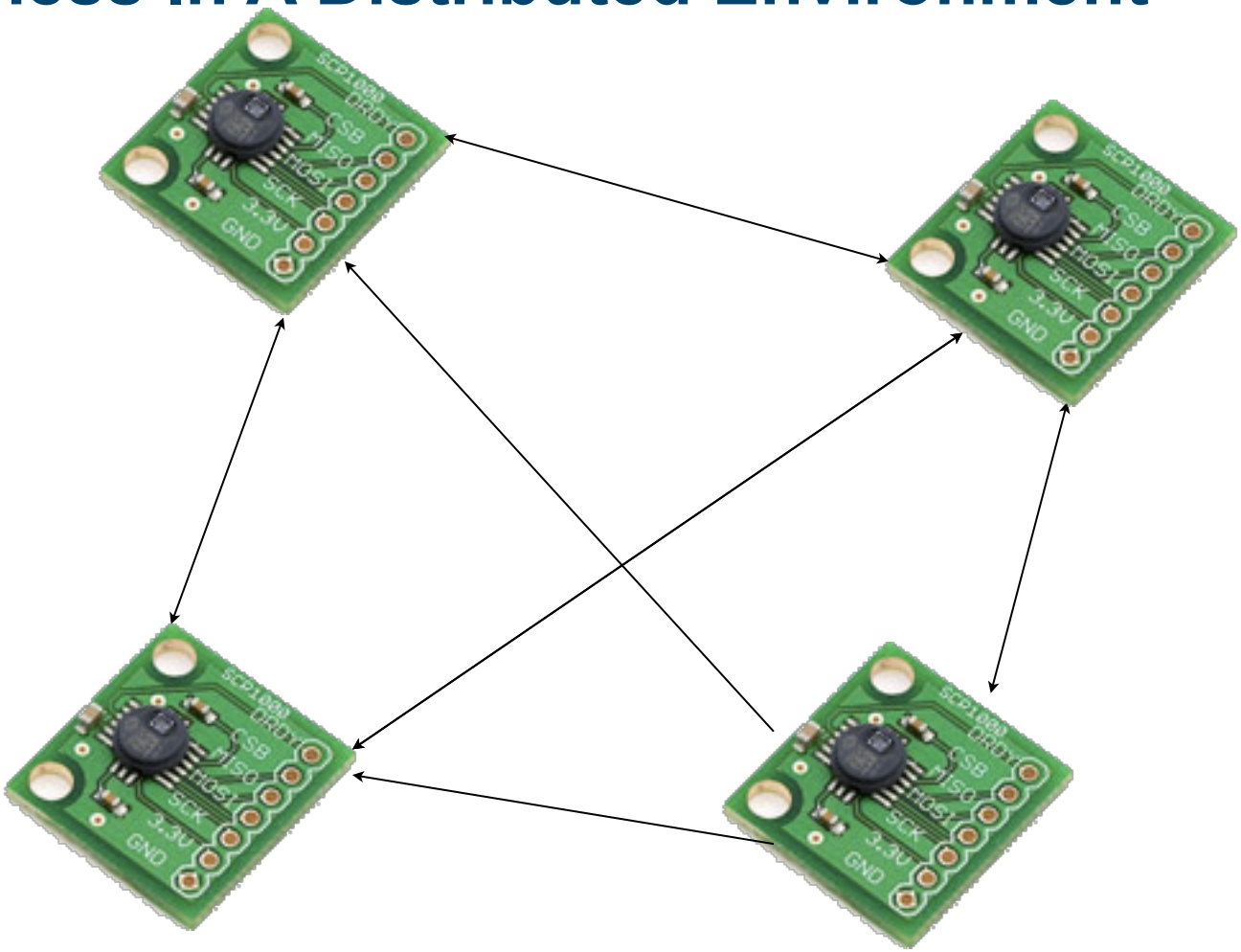
I-Ext: A (t, t-1)-immune extractor

- Let Ext be any (k, δ) two source extractor
- Let X_1^2, X_1^3, ... denote t distinct blocks from the source X_1
- Let X_2, X_3, ... be one block each from the other t sources
- $\text{I-Ext}(\{X_1^i\}_{i=2}^{t+1}, X_2, \ldots, X_{t+1}) = \oplus_{i=2}^{t+1} \text{Ext}(X_1^i, X_i)$

- Theorem: I-Ext is a (t, t-1) immune extractor!

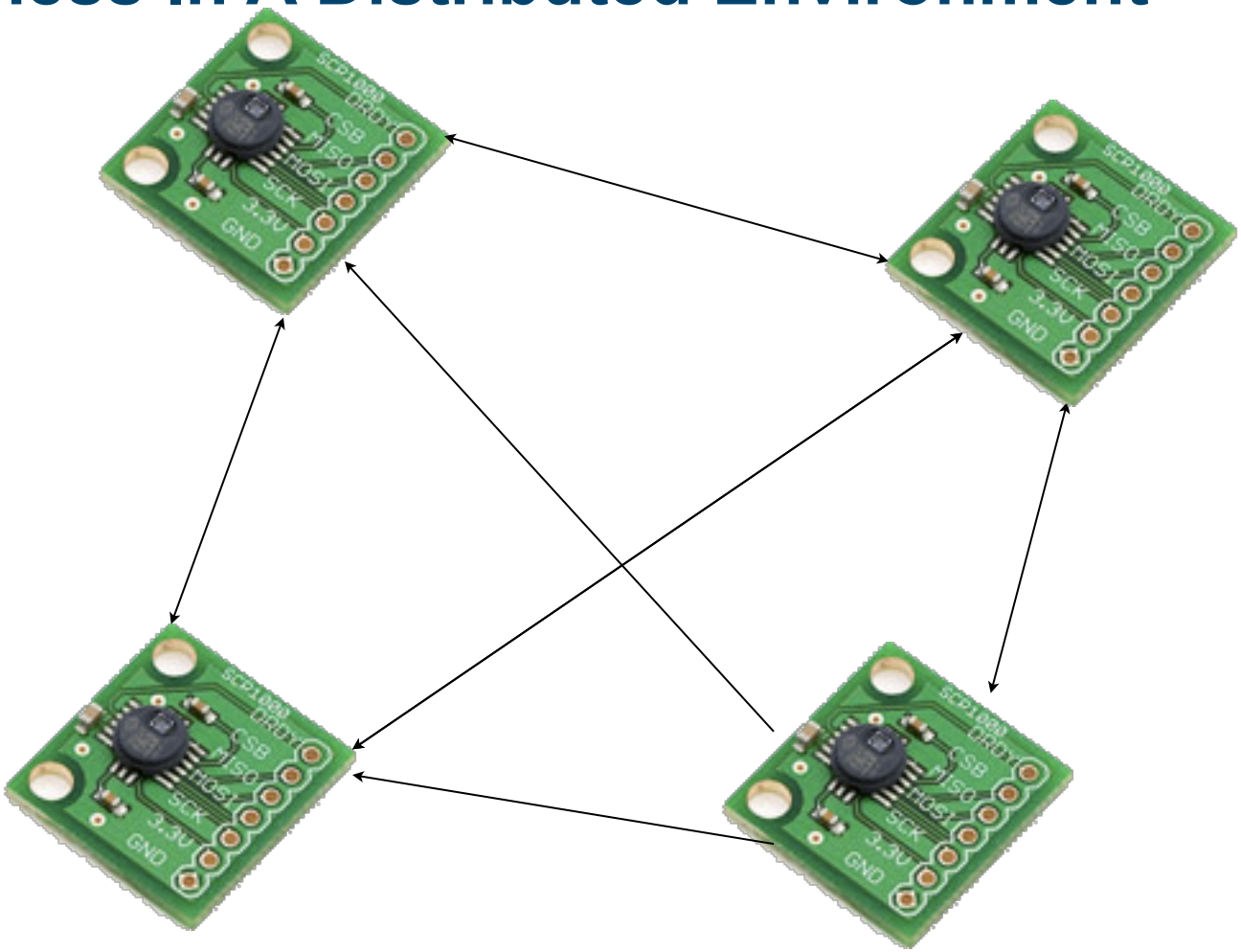
i

4. Randomness In A Distributed Environment



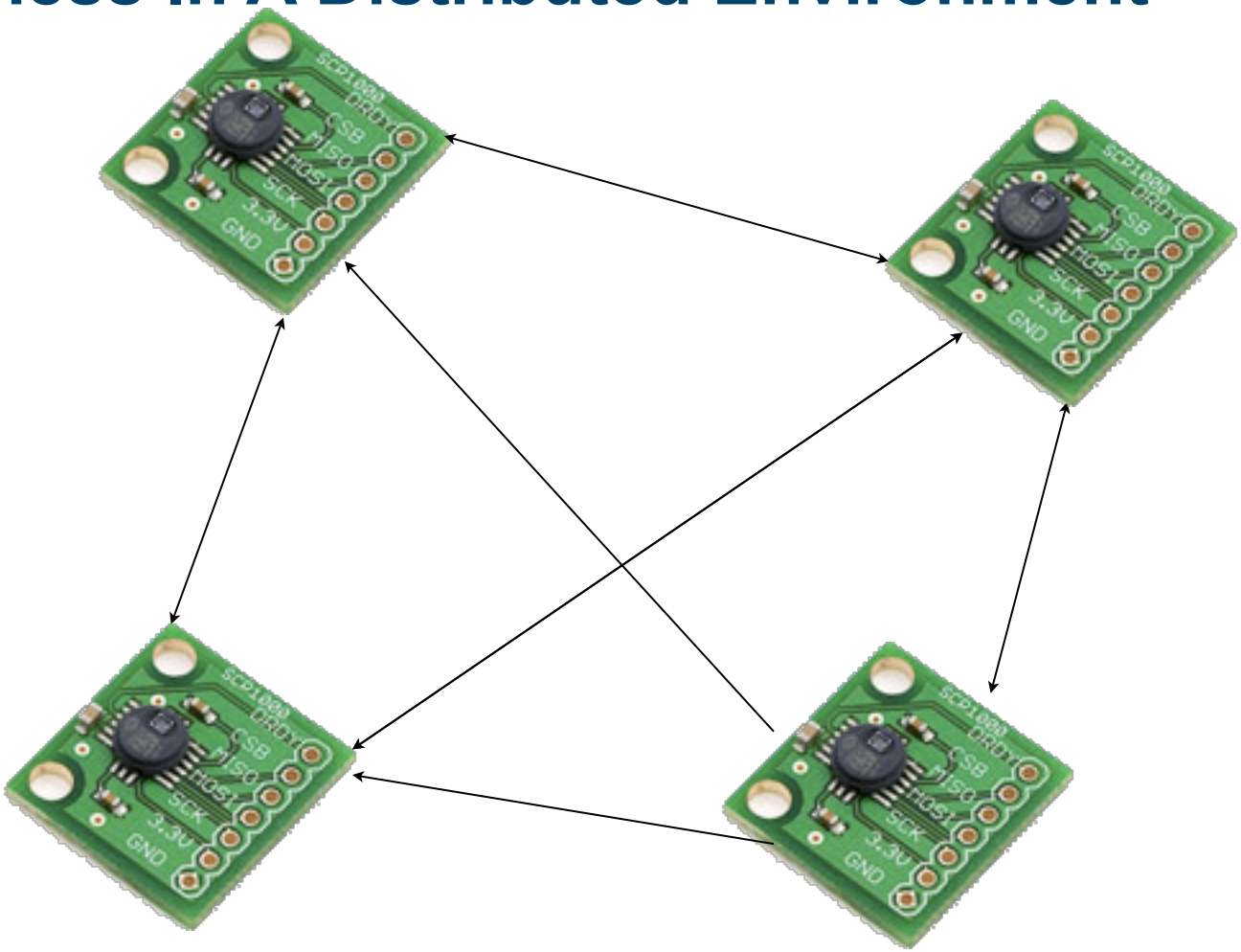
... each player sends each other player a string from its weak random source ...

4. Randomness In A Distributed Environment



... each player can use I-Ext to generate a string very close to U_m ..

4. Randomness In A Distributed Environment



... every player has an almost random string ...

5. Conclusion

The paper mainly showed how it is possible to for each player of a network to extract a almost unbiased, uniform random string by sharing their (mutually independent) random sources, even in presence of faulty or even malicious parties.