RSAConference2019
San Francisco | March 4–8 | Moscone Center

SESSION ID: CRYP-W03

Error Detection in Monotone Span Programs with Application to Communication-Efficient Multiparty Computation

Tim Wood

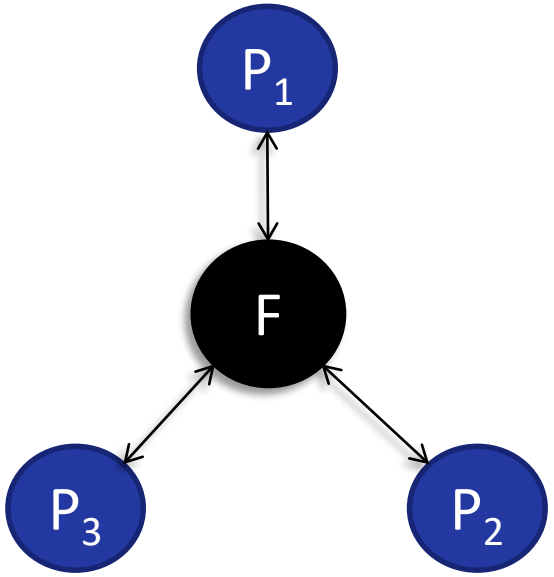
University of Bristol/KU Leuven COSIC-imec

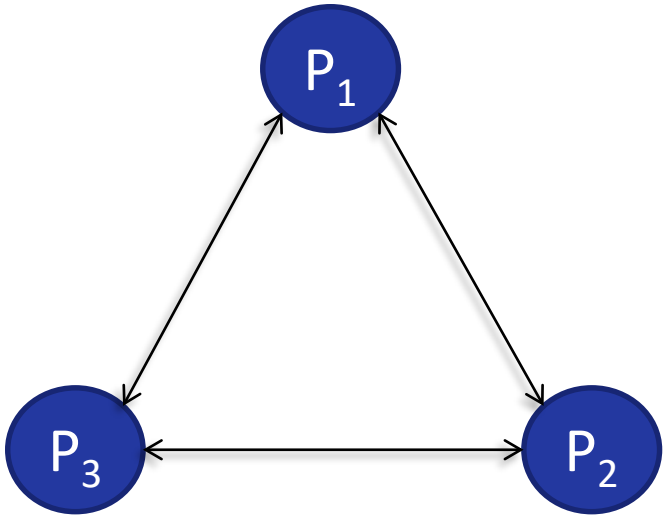
Co-authored with Nigel Smart

Outline

- What is MPC?
- Goal
- Tools
- Protocol

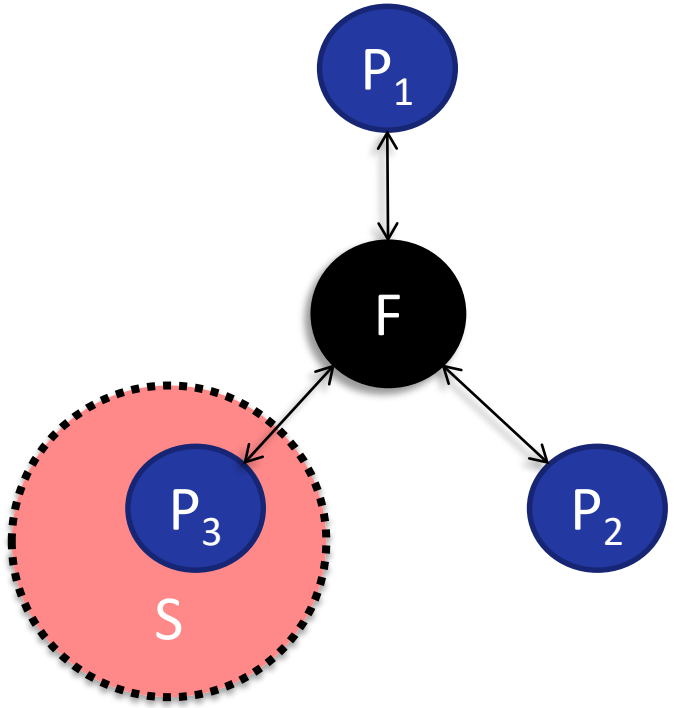
What is Multiparty Computation?

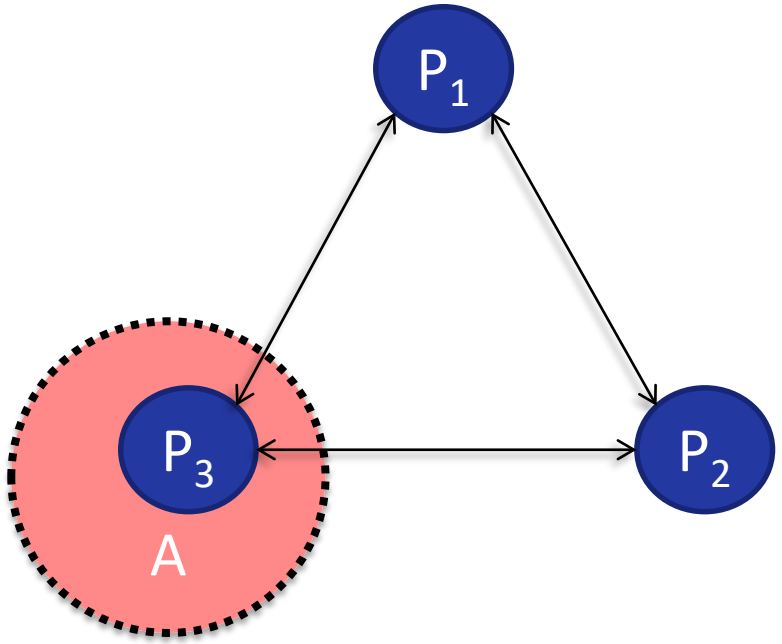




Guaranteeing e.g. correctness, privacy, fairness, etc.

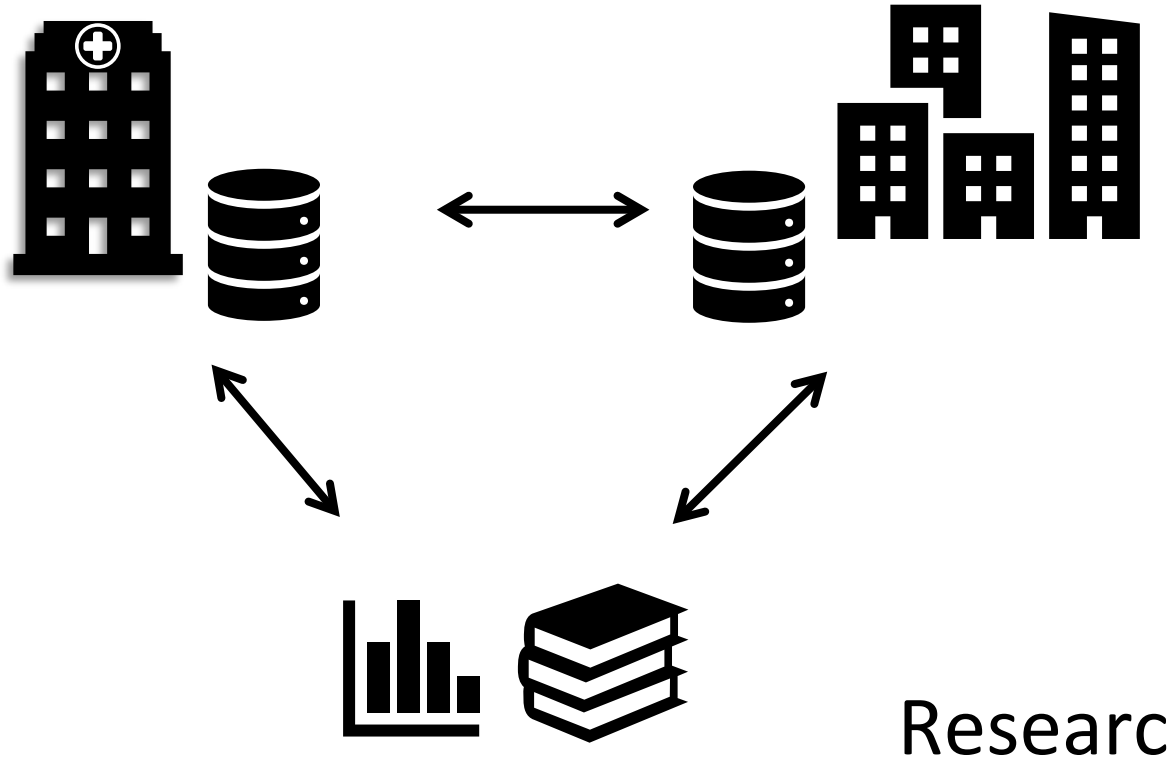
What is MPC?





Guaranteeing e.g. correctness, privacy, fairness, etc.

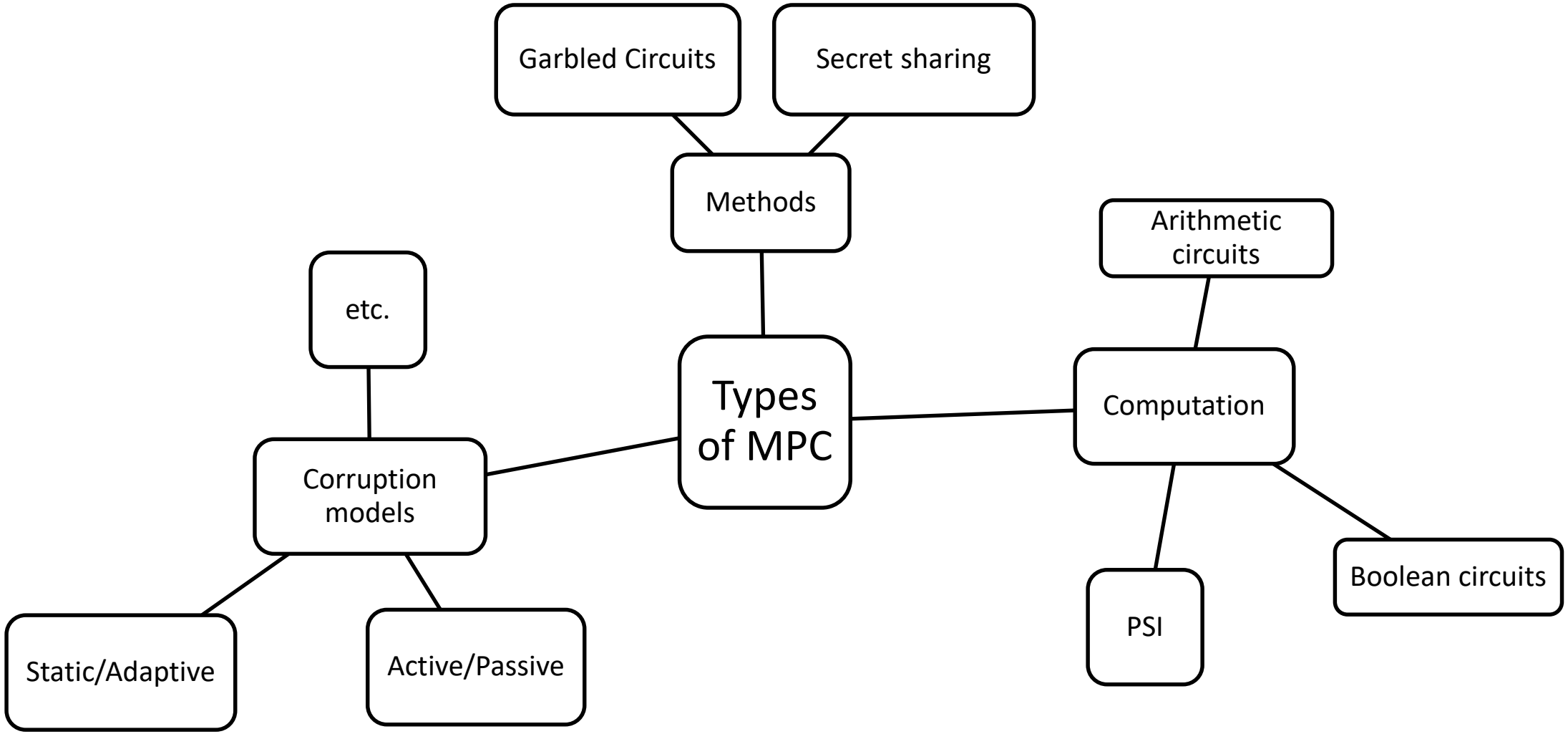
For example…



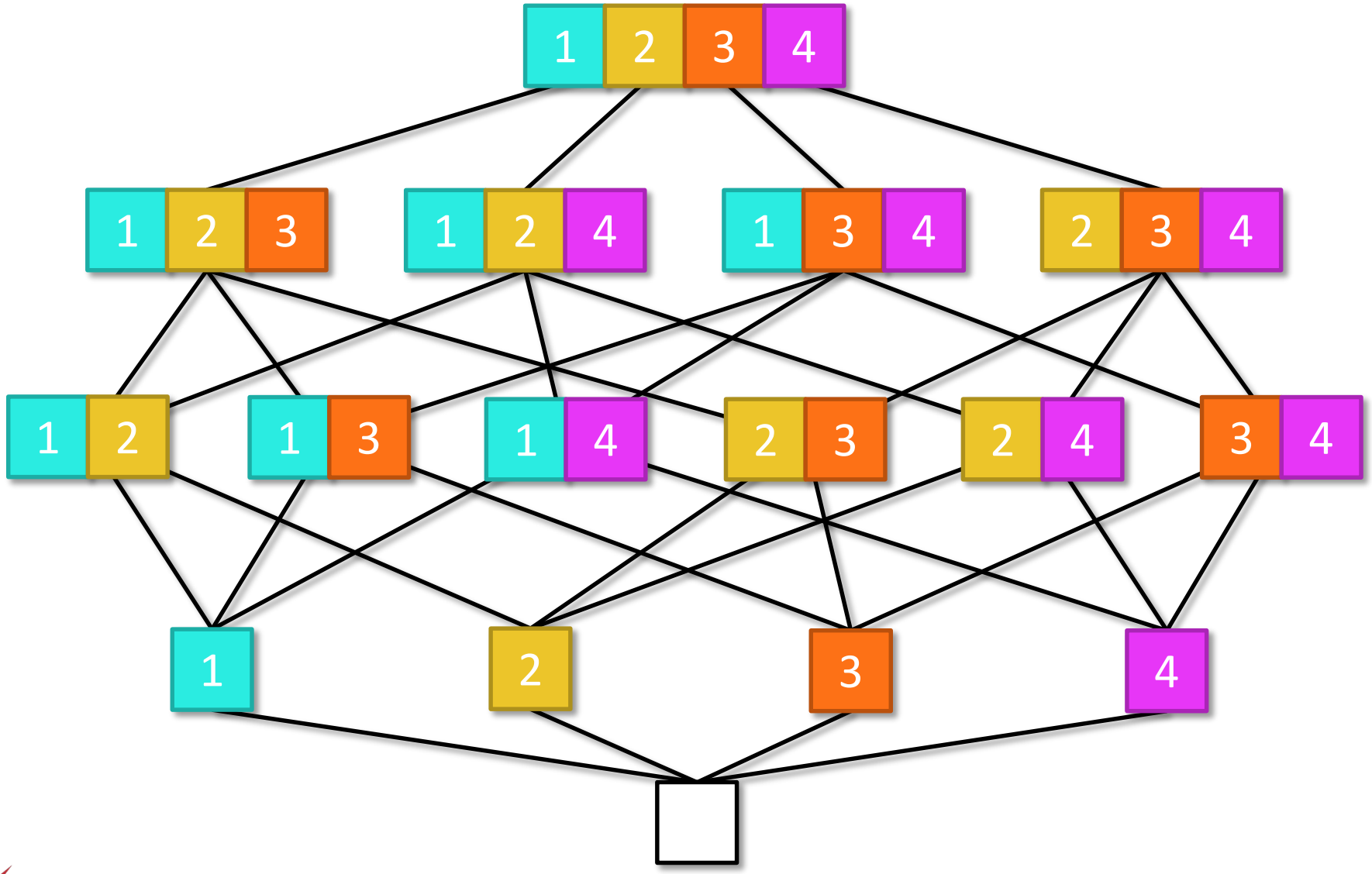
Research on medical data…

…without pooling patient data

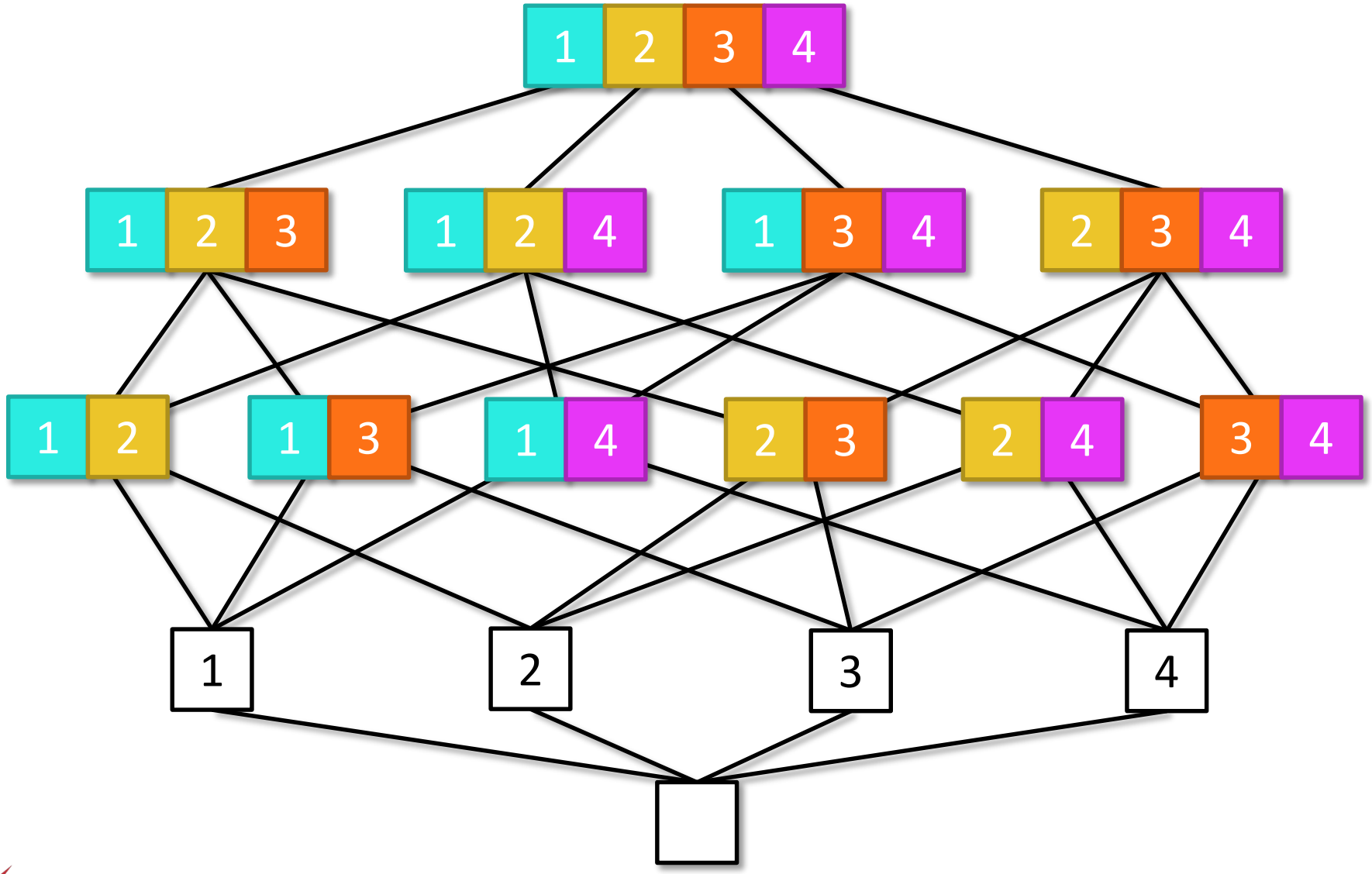
Categories


Garbled Circuits
Secret sharing
Methods
etc.
Arithmetic circuits
Types of MPC
Computation
Corruption models
Static/Adaptive
Active/Passive
PSI
Boolean circuits

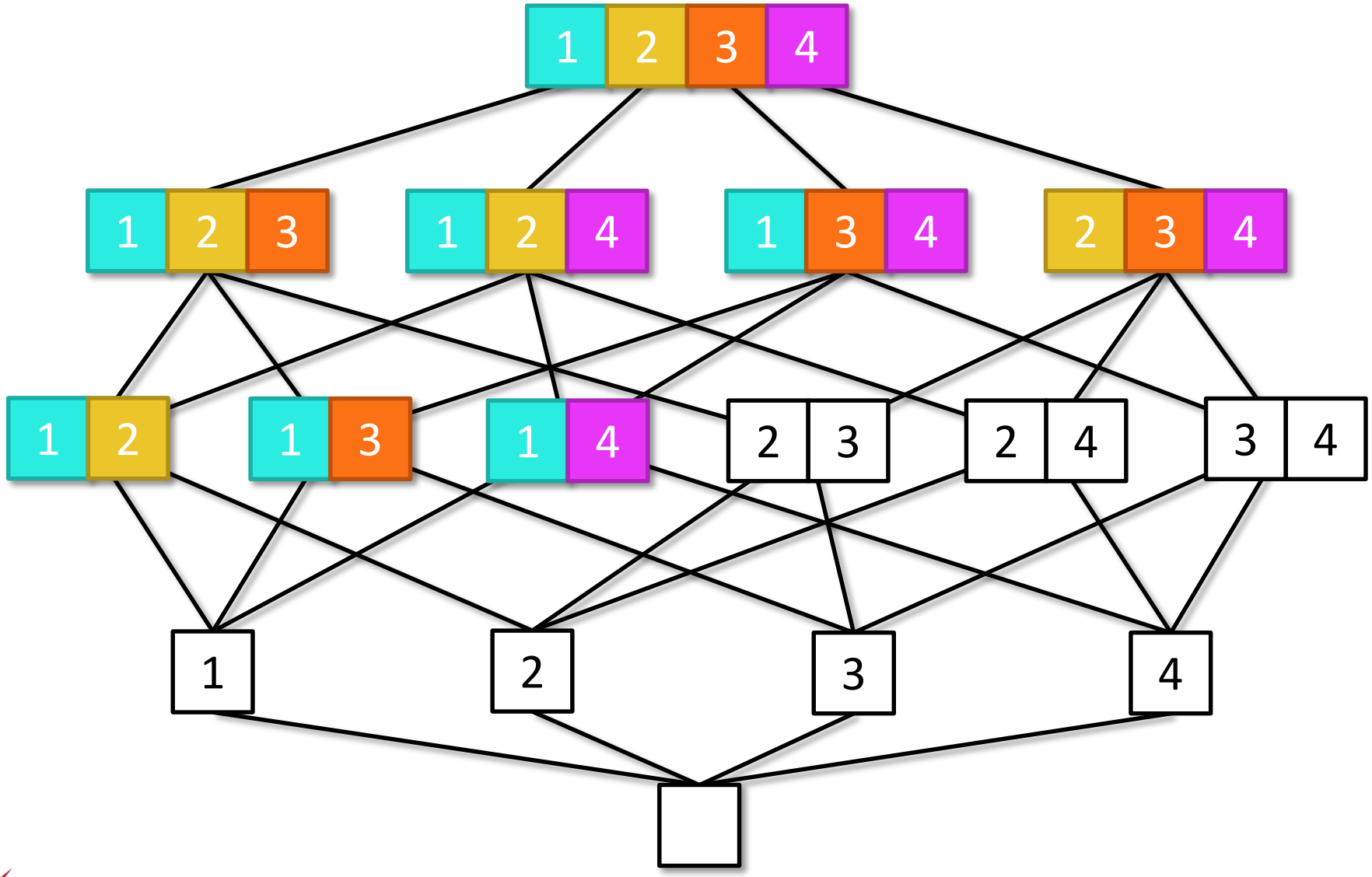
Access structures



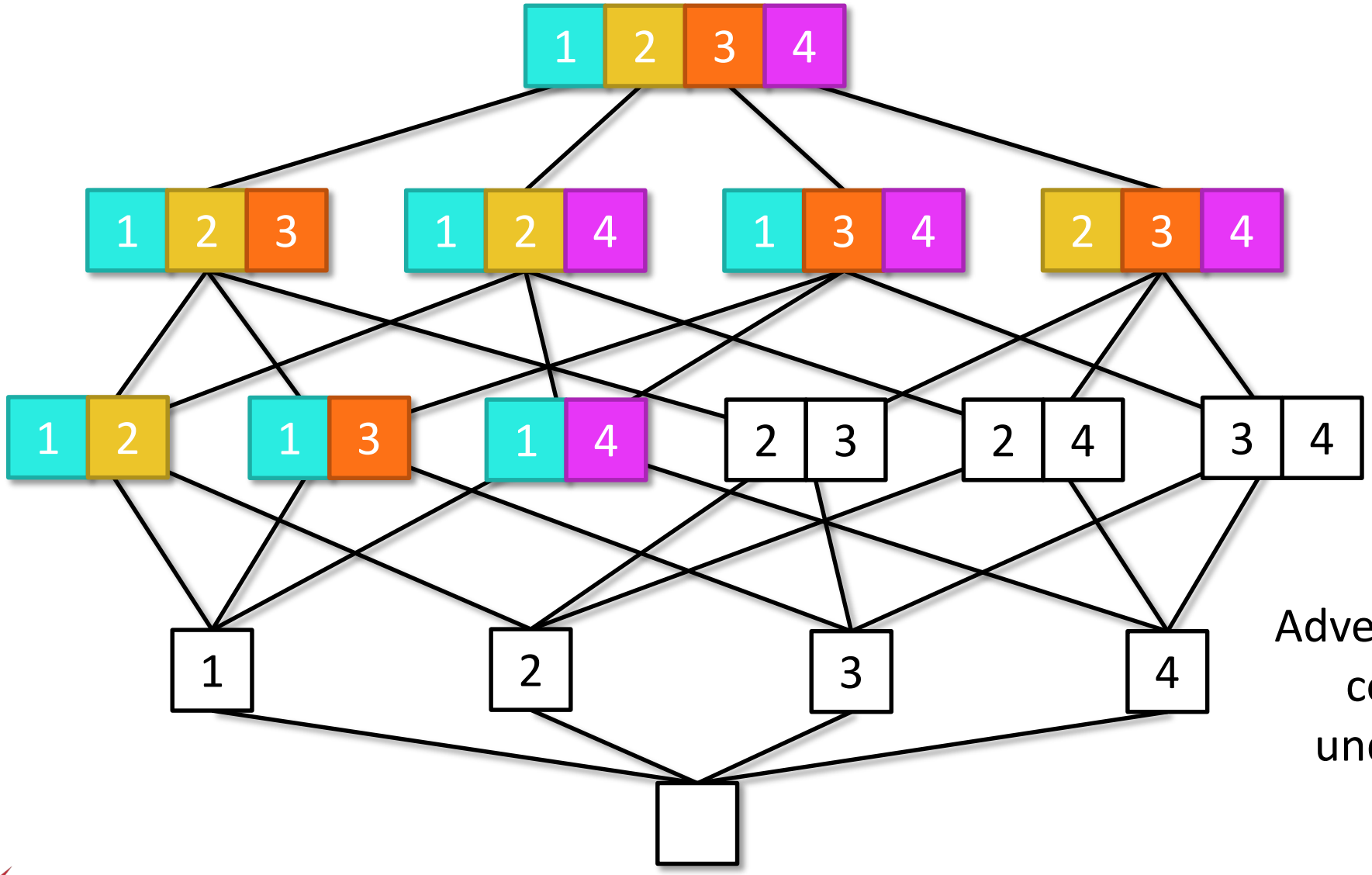
(4,1) threshold access structure



Another access structure



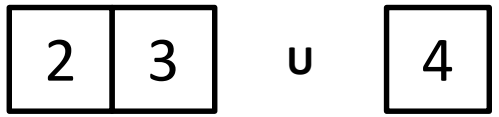
Another access structure



Adversary can only corrupt one unqualified set

Q_2?

Union of any two unqualified sets is missing at least one party

e.g. 

Can think of as “generalised honest majority”

Goal

- This work's focus: Computation…

  of arithmetic circuits

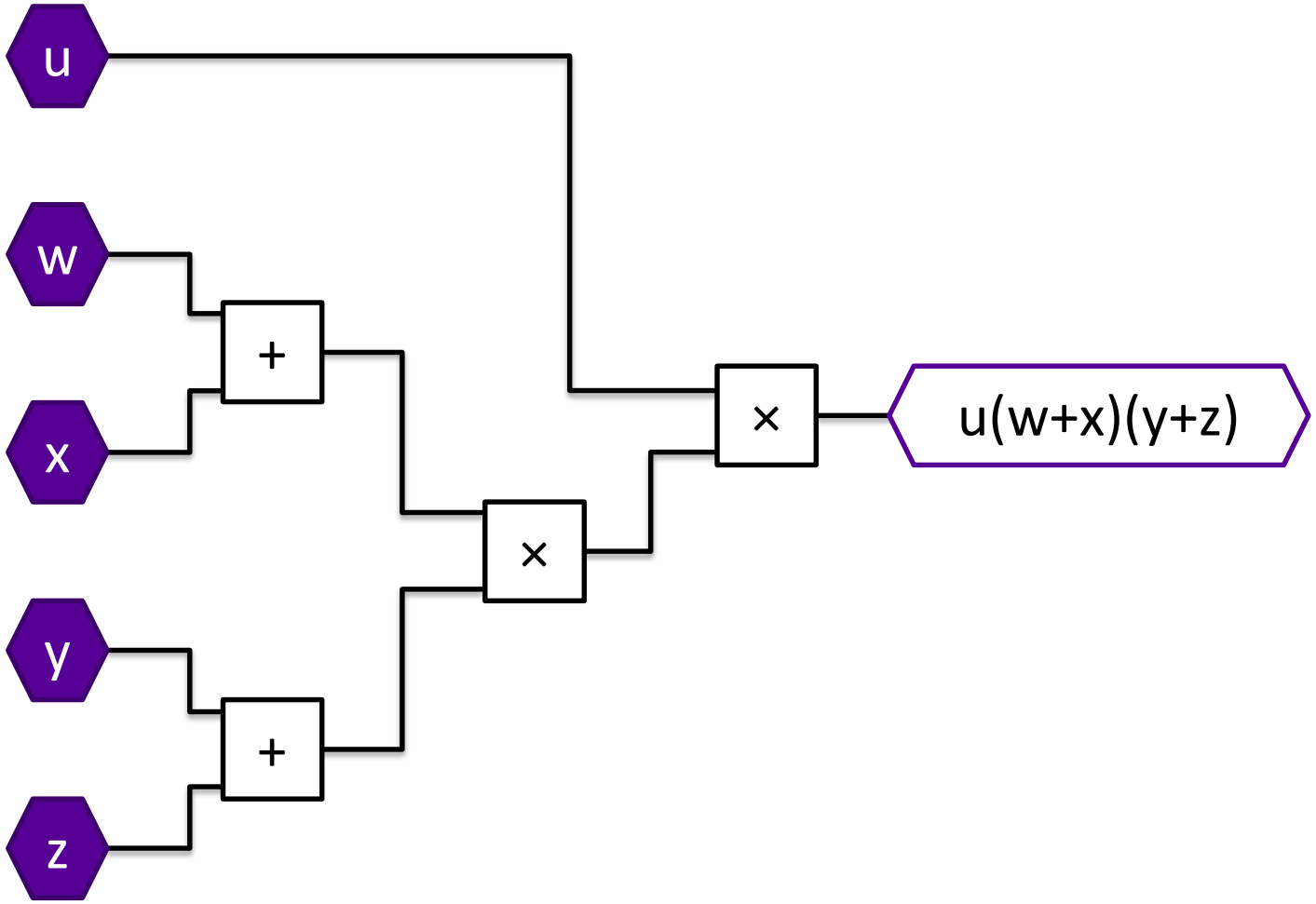
  with efficient communication

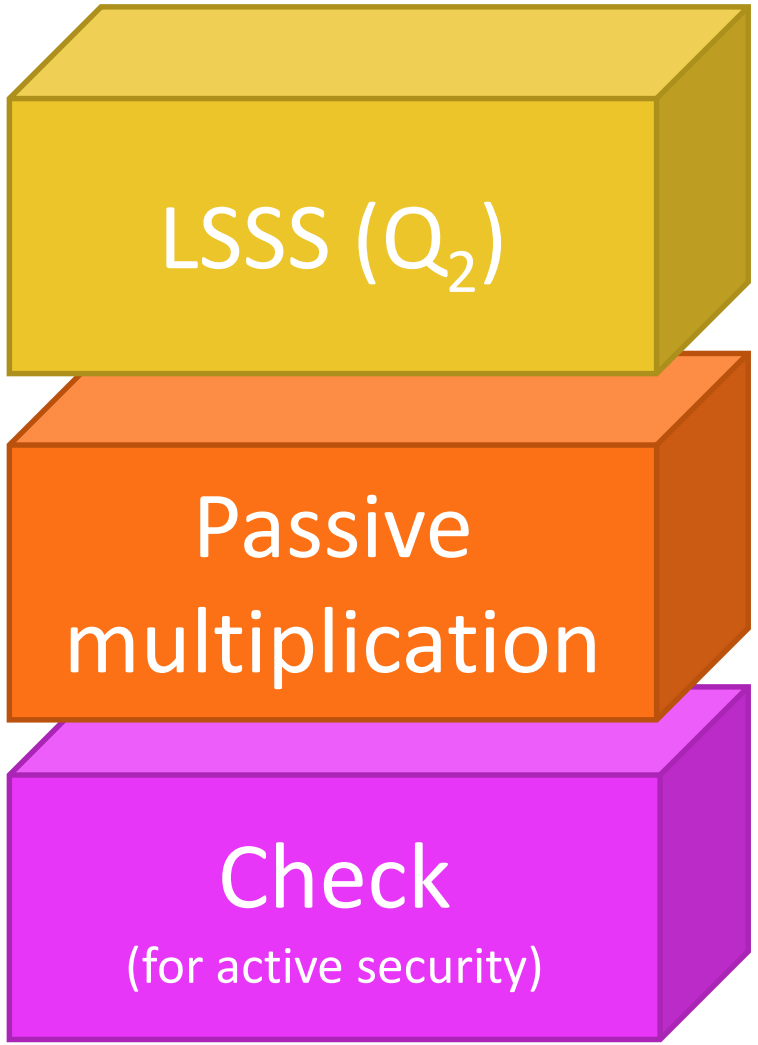
  and active security

  for Q_2 access structures

- General goal:
  - Above, for *any* access structure (see SPDZ family)

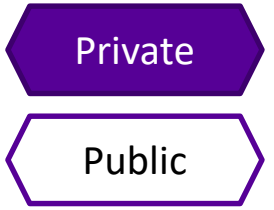
Arithmetic Circuits

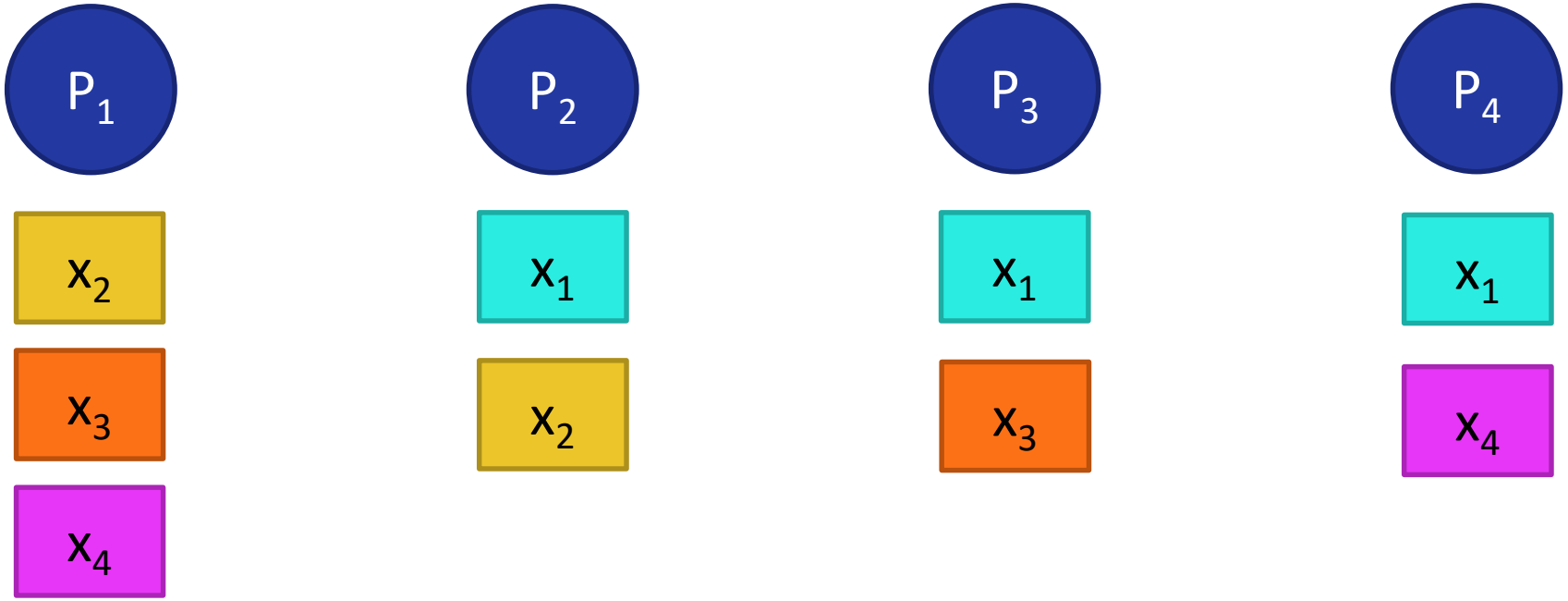




Linear Secret Sharing Scheme (LSSS)

We write  if x is secret and parties hold *shares*



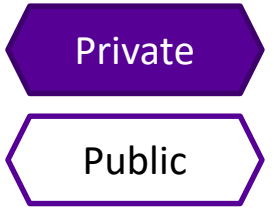


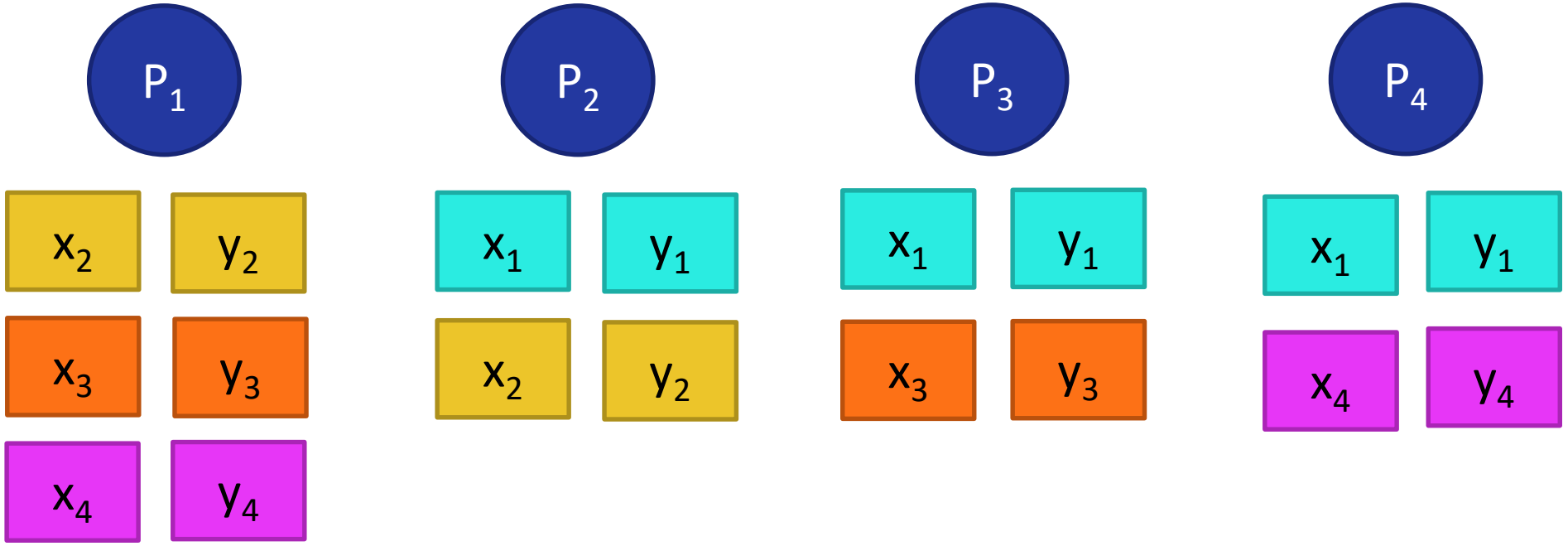
$$x_1 + x_2 + x_3 + x_4 = x$$

Linear Secret Sharing Scheme (LSSS): Adding Secrets

We write 

if x is secret and parties hold *shares*

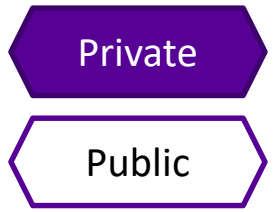


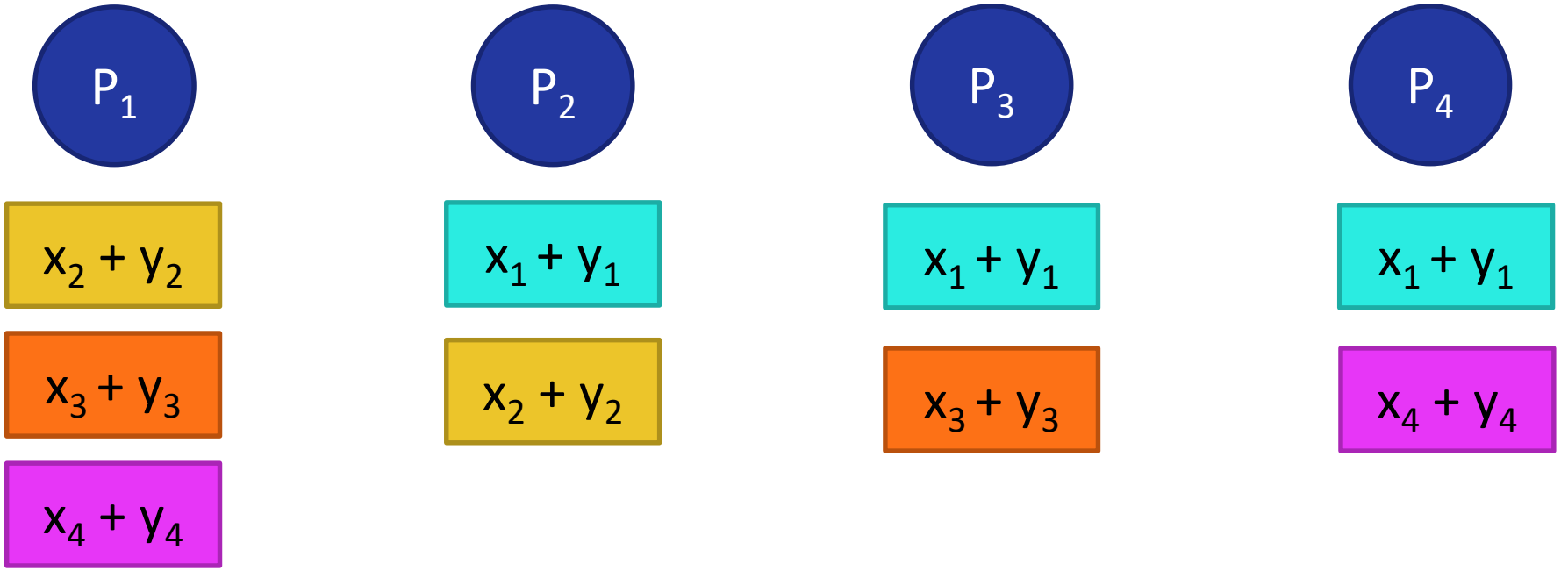


Linear Secret Sharing Scheme (LSSS): Adding Secrets

We write 

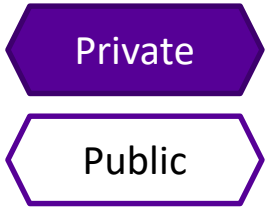
if x is secret and parties hold *shares*:

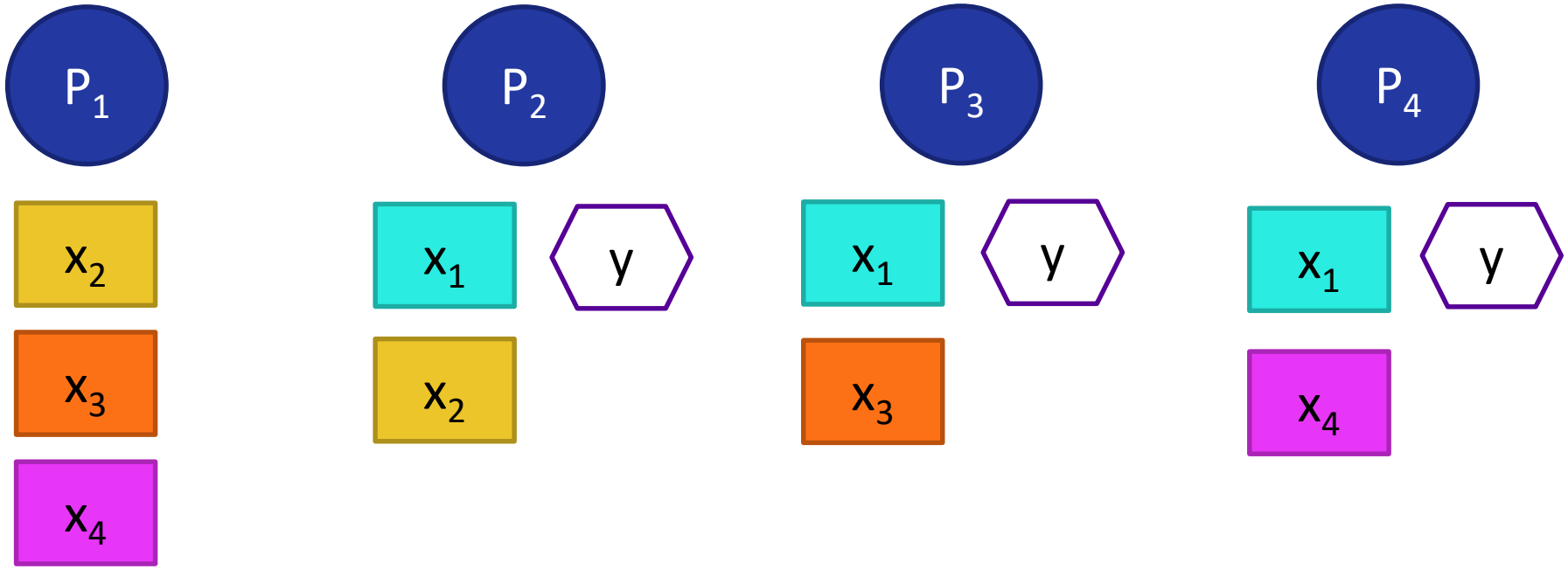




Linear Secret Sharing Scheme (LSSS): Adding Public values

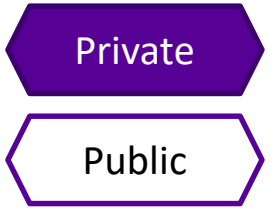
We write  if x is secret and parties hold *shares*

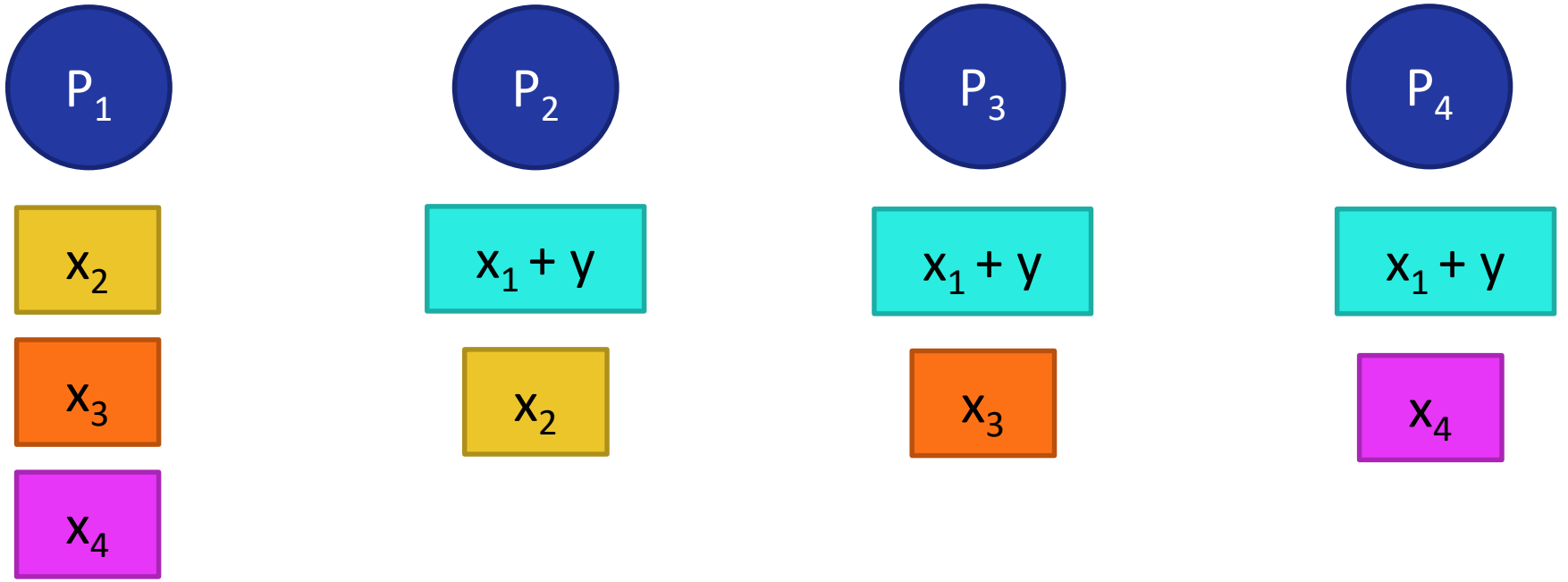




Linear Secret Sharing Scheme (LSSS): Adding Public values

We write  if x is secret and parties hold *shares*





KRSW17: Active security from RSS

This type of sharing is called replicated secret sharing

Every share is held by at least one honest party because the access structure is Q_2

Thus for every share, local additions are always performed by at least one honest party

New result

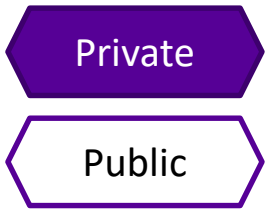
This is not special to RSS!

For any Q_2 access structure, for any LSSS realising it, the adversary cannot add errors without "invalidating" the shares…

…because all *shares* (not just the secret) can be reconstructed from any set of shares held by qualified parties

Multiplication: Beaver's Circuit Randomisation

Suppose the parties want to multiply two secrets



…and suppose they *already* have

where a and b are random, secret, and unknown to any party.

Multiplication: Beaver's Circuit Randomisation

Private

Public

Parties (locally) compute

$x + a$ $y + b$

and "open" the secrets

$x + a$ $y + b$

then locally compute

$$xy = (x + a) \times y + (y + b) \times x - (x + a) \times (y + b) + ab$$

i.e. linear combination produces secret-shared product.

Costs

"Offline"

Produce lots of Beaver triples (see paper)

"Online"

Addition gates: "for free" (no communication)

Multiplication gates: opening two secrets

Hash comparison at the end for active security

Opening

Online efficiency depends (almost) only on efficiency of “opening”.

Need active security too…

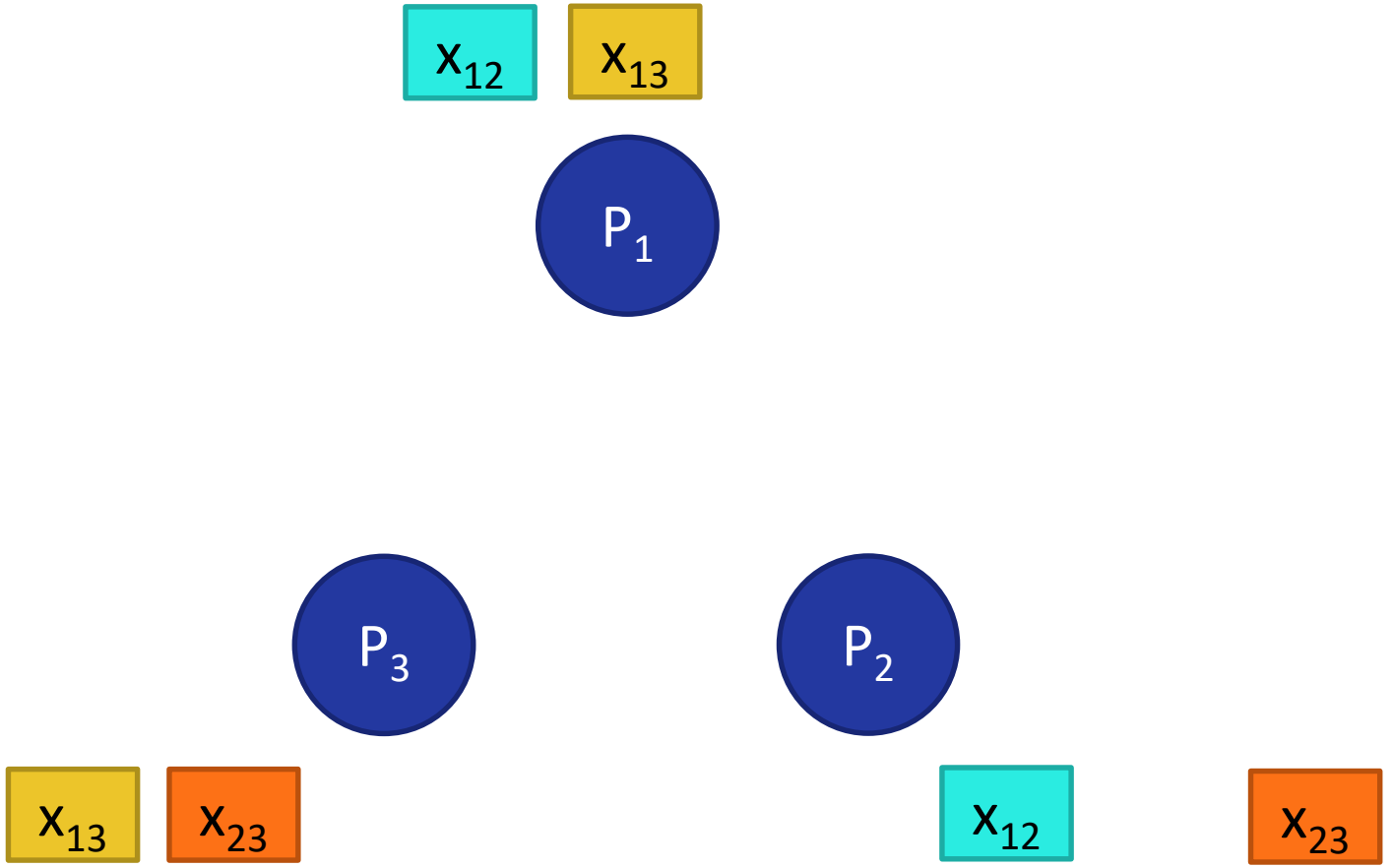
FLNW16 idea

For three parties, exploit there are two honest parties and shares are replicated:

Hash the shares and compare

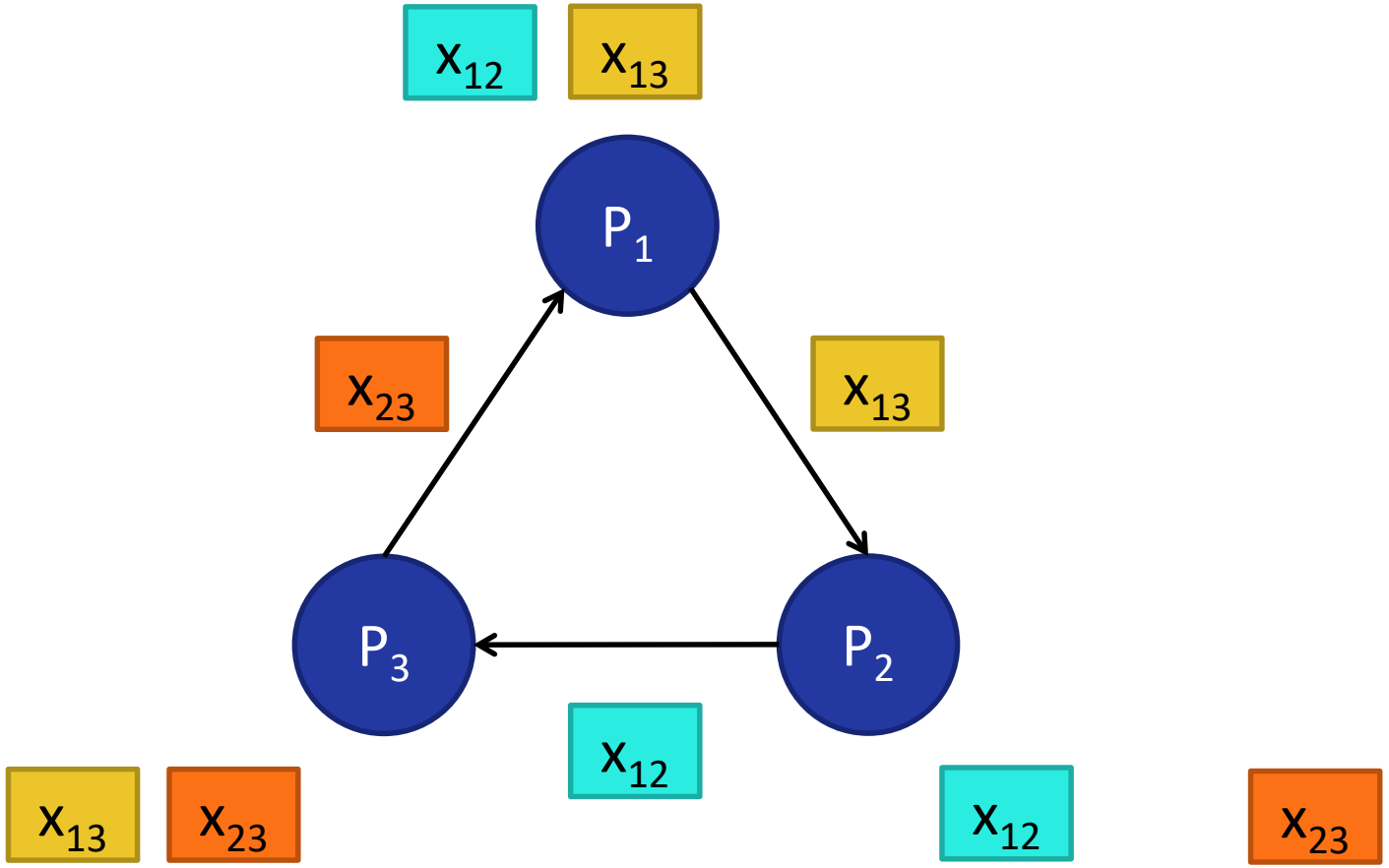
Previous work: FLNW16

$x = x_{12} + x_{13} + x_{23}$



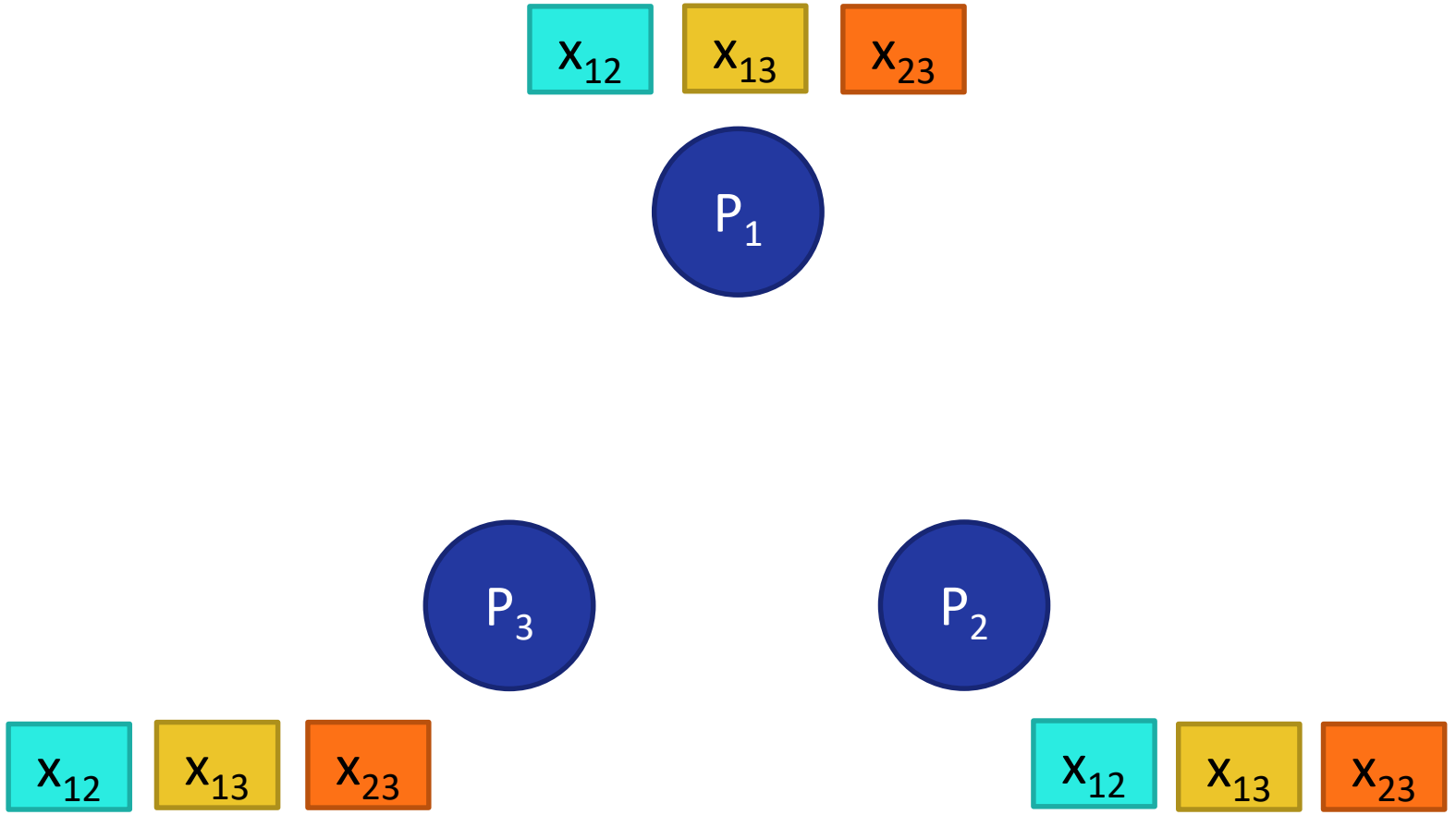
Previous work: FLNW16

$x = x_{12} + x_{13} + x_{23}$



Previous work: FLNW16

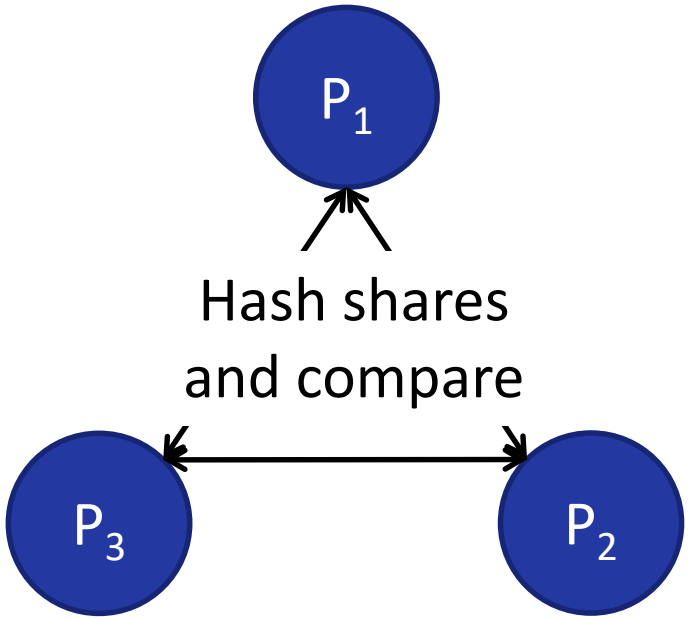
$x = x_{12} + x_{13} + x_{23}$



Previous work: FLNW16

$x = x_{12} + x_{13} + x_{23}$

$h_1 \leftarrow H(x_{12}, x_{13}, x_{23})$



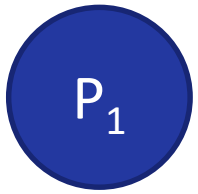
$h_3 \leftarrow H(x_{12}, x_{13}, x_{23})$

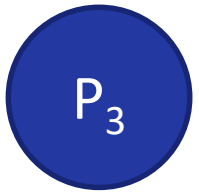
$h_2 \leftarrow H(x_{12}, x_{13}, x_{23})$

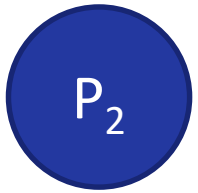
Previous work: FLNW16

$x = x_{12} + x_{13} + x_{23}$

If $h_1 = h_2 = h_3$ then output x







If $h_1 = h_2 = h_3$ then output x

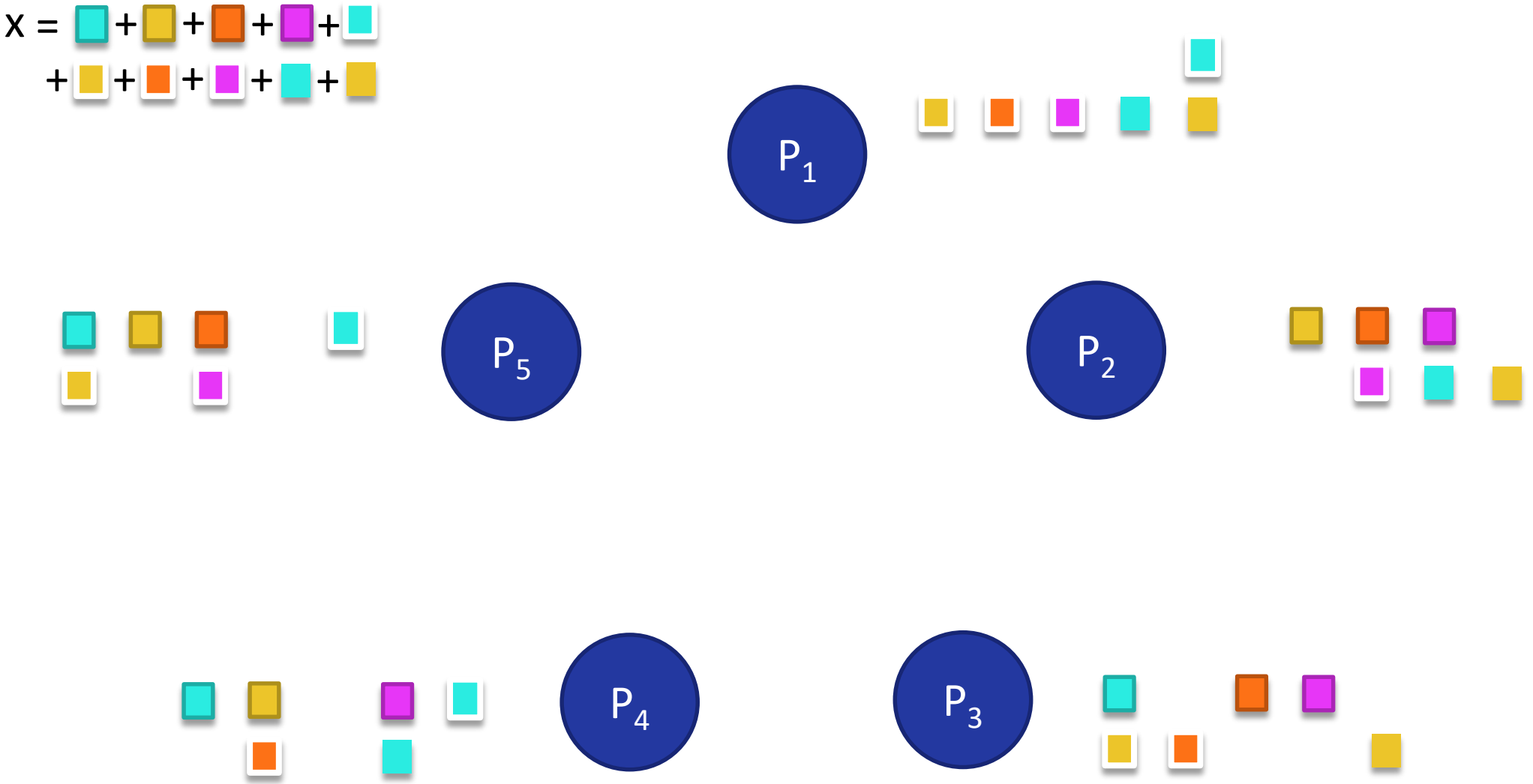
If $h_1 = h_2 = h_3$ then output x

KRSW18 idea

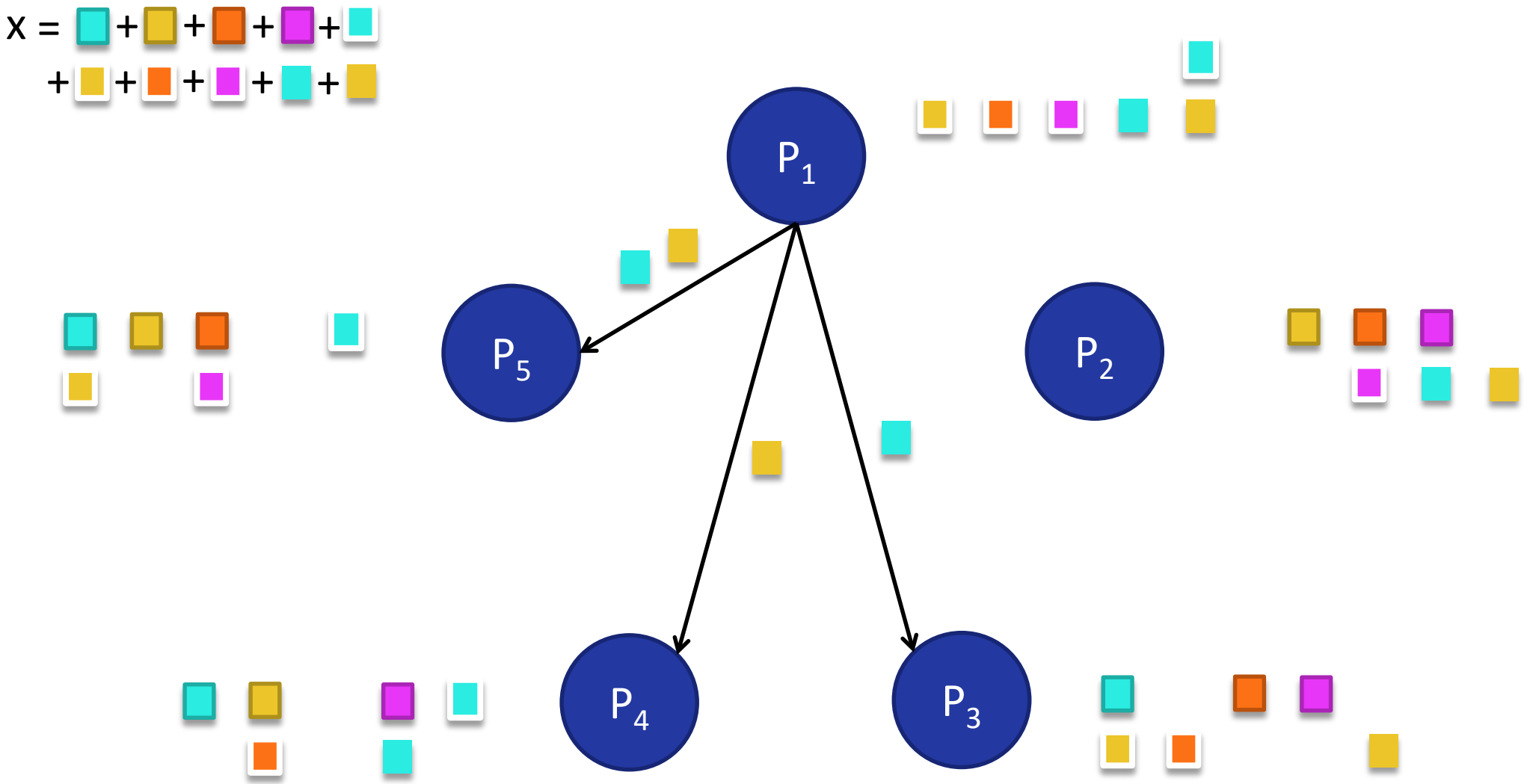
For n parties, exploit that the structure is Q_2 and shares are replicated: again,

Hash the shares and compare

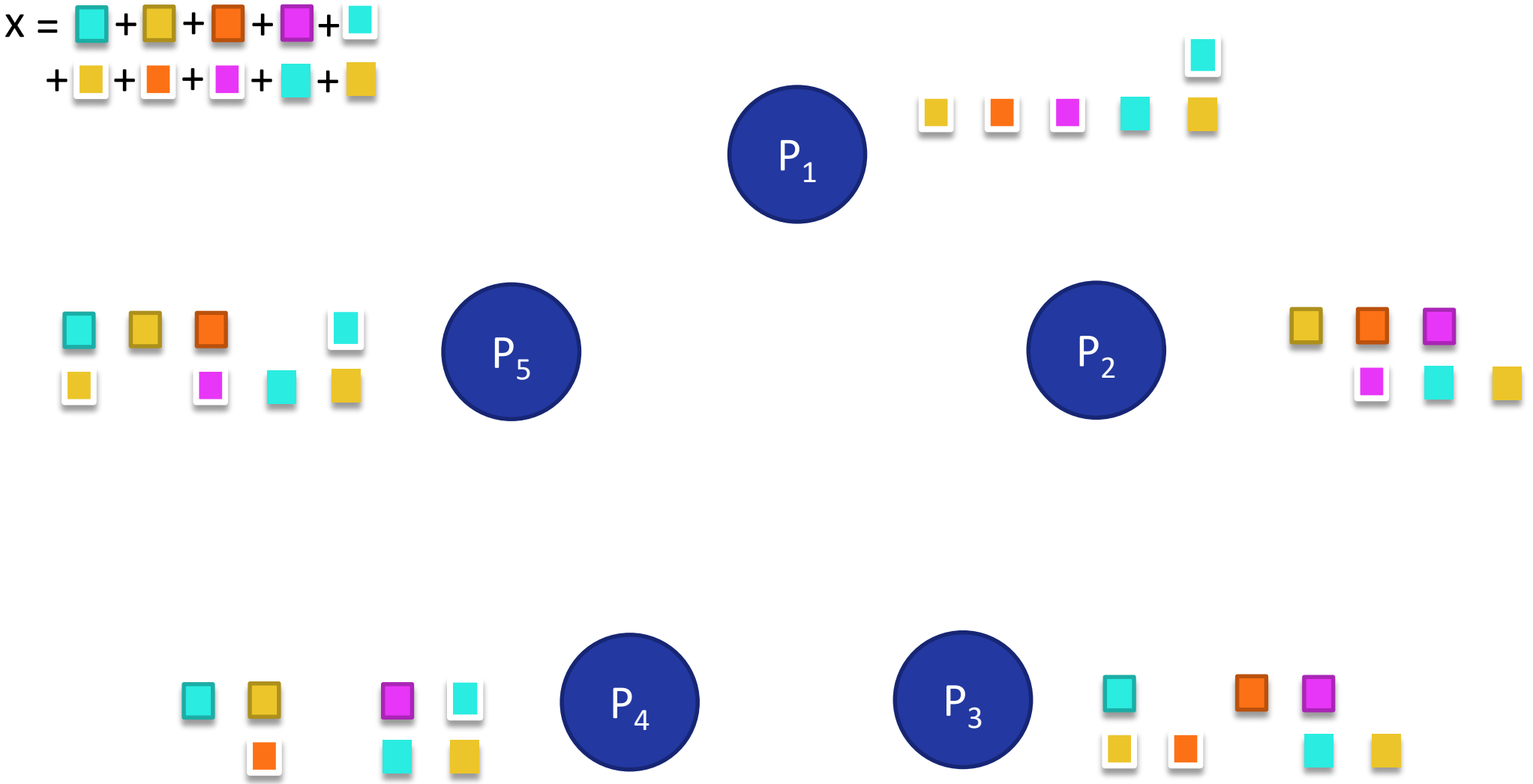
Previous work: KRSW18



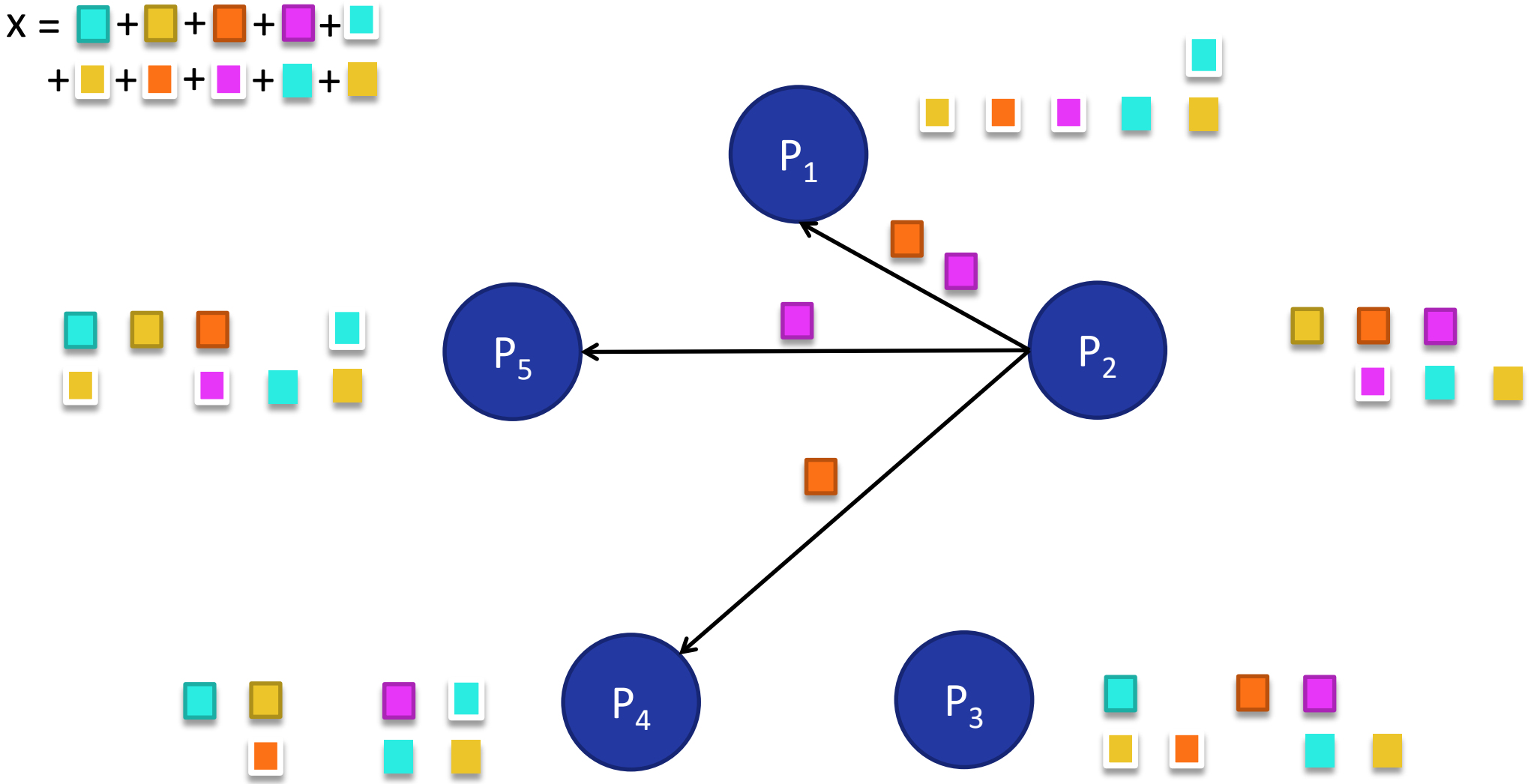
Previous work: KRSW18



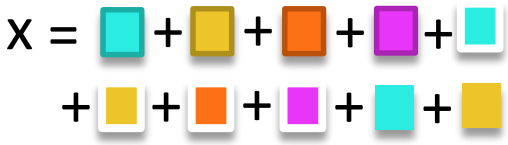
Previous work: KRSW18

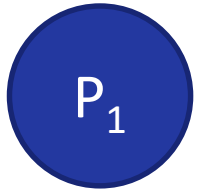


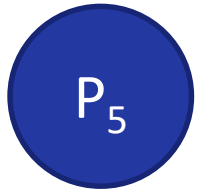
Previous work: KRSW18

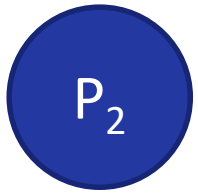


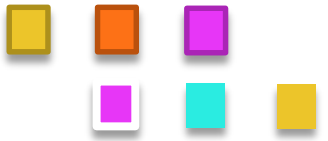
Previous work: KRSW18

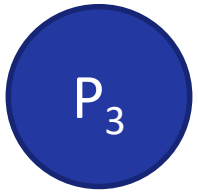




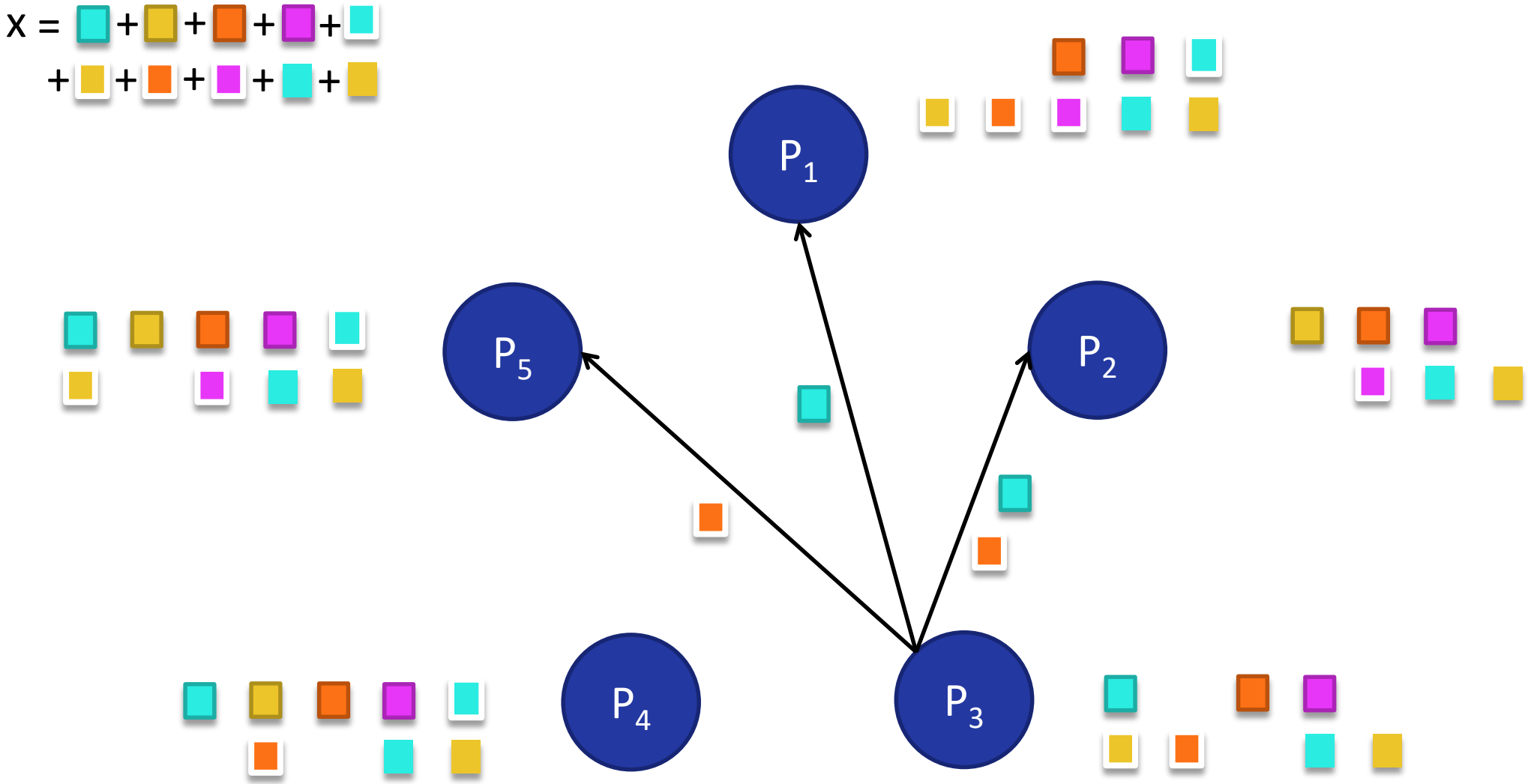




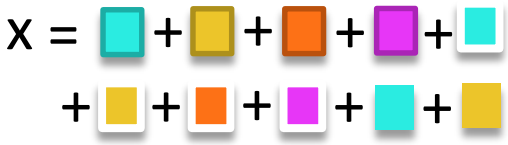


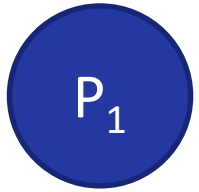


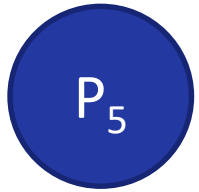
Previous work: KRSW18

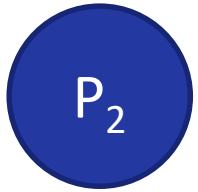


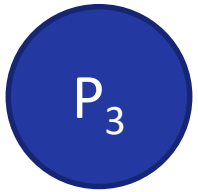
Previous work: KRSW18



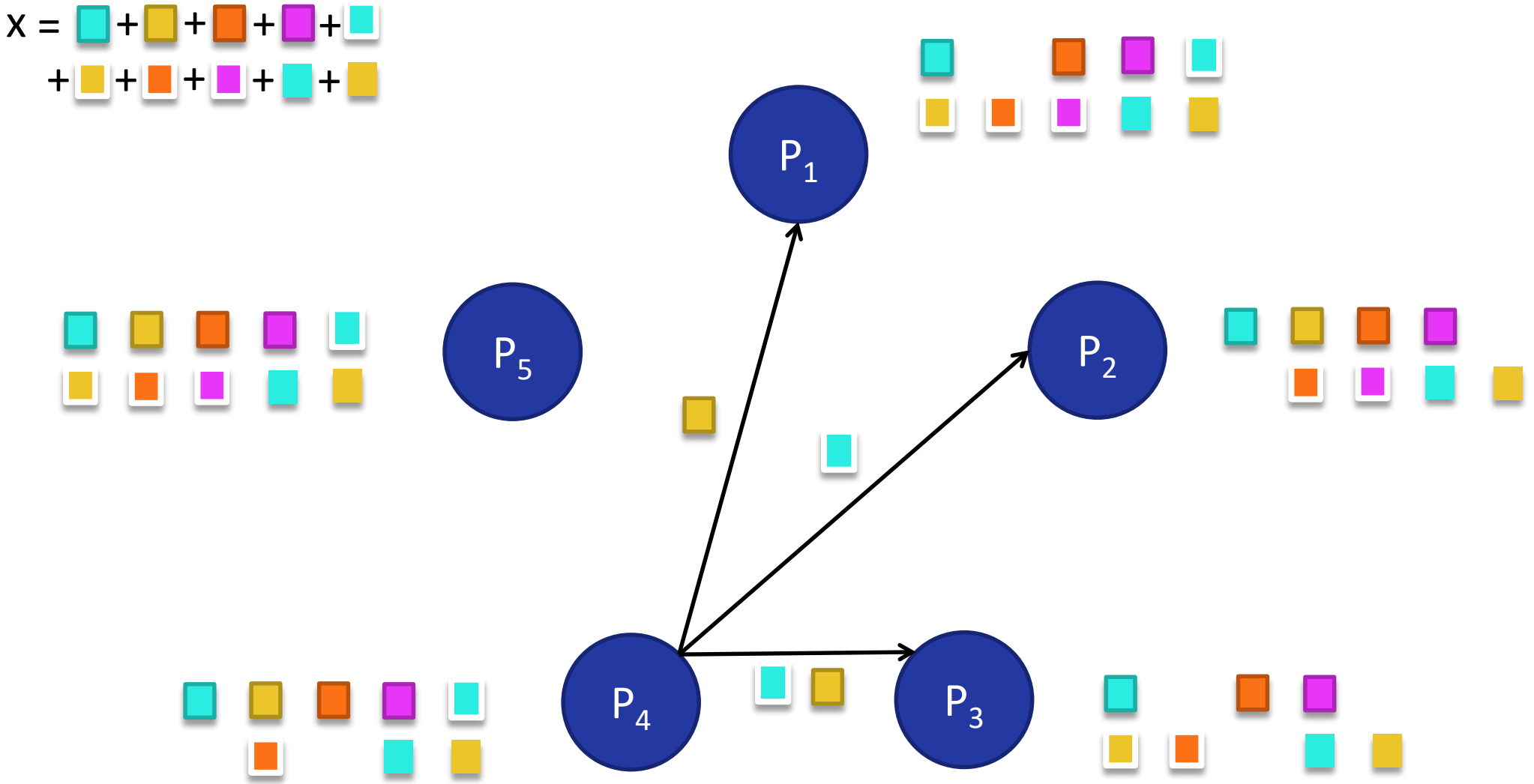




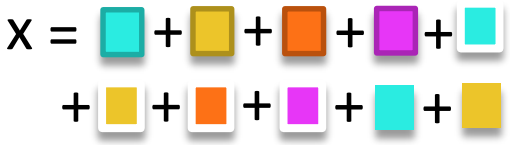


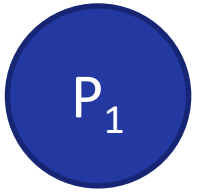


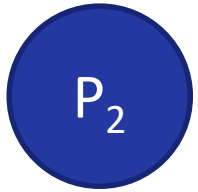
Previous work: KRSW18

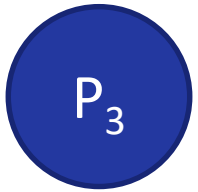


Previous work: KRSW18

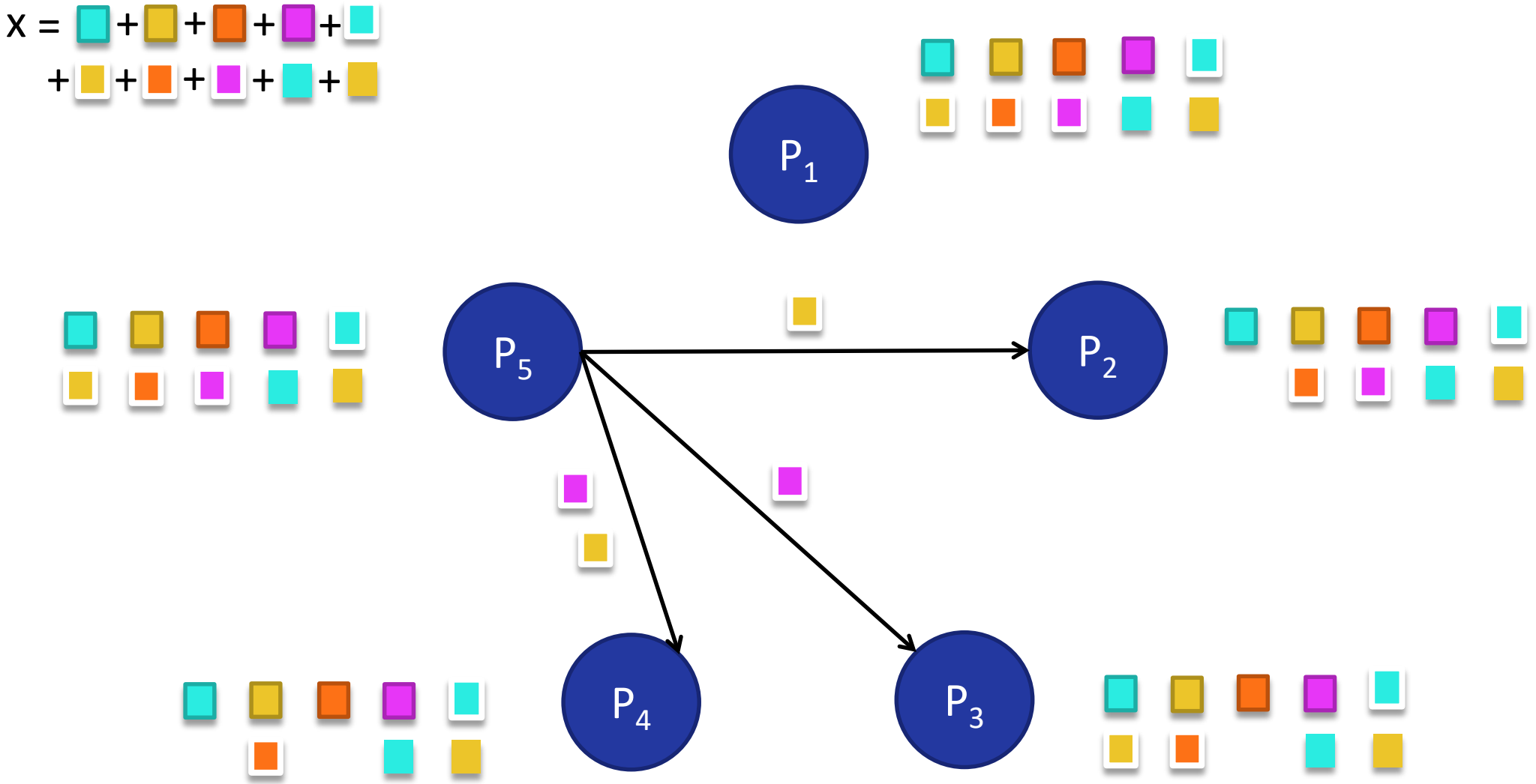




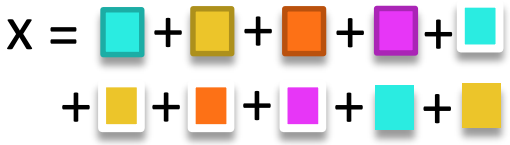


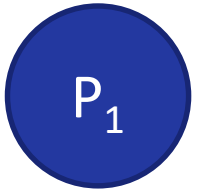


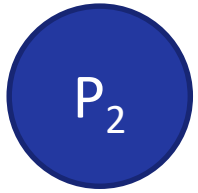
Previous work: KRSW18

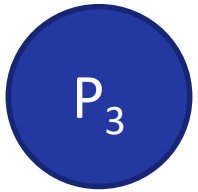


Previous work: KRSW18

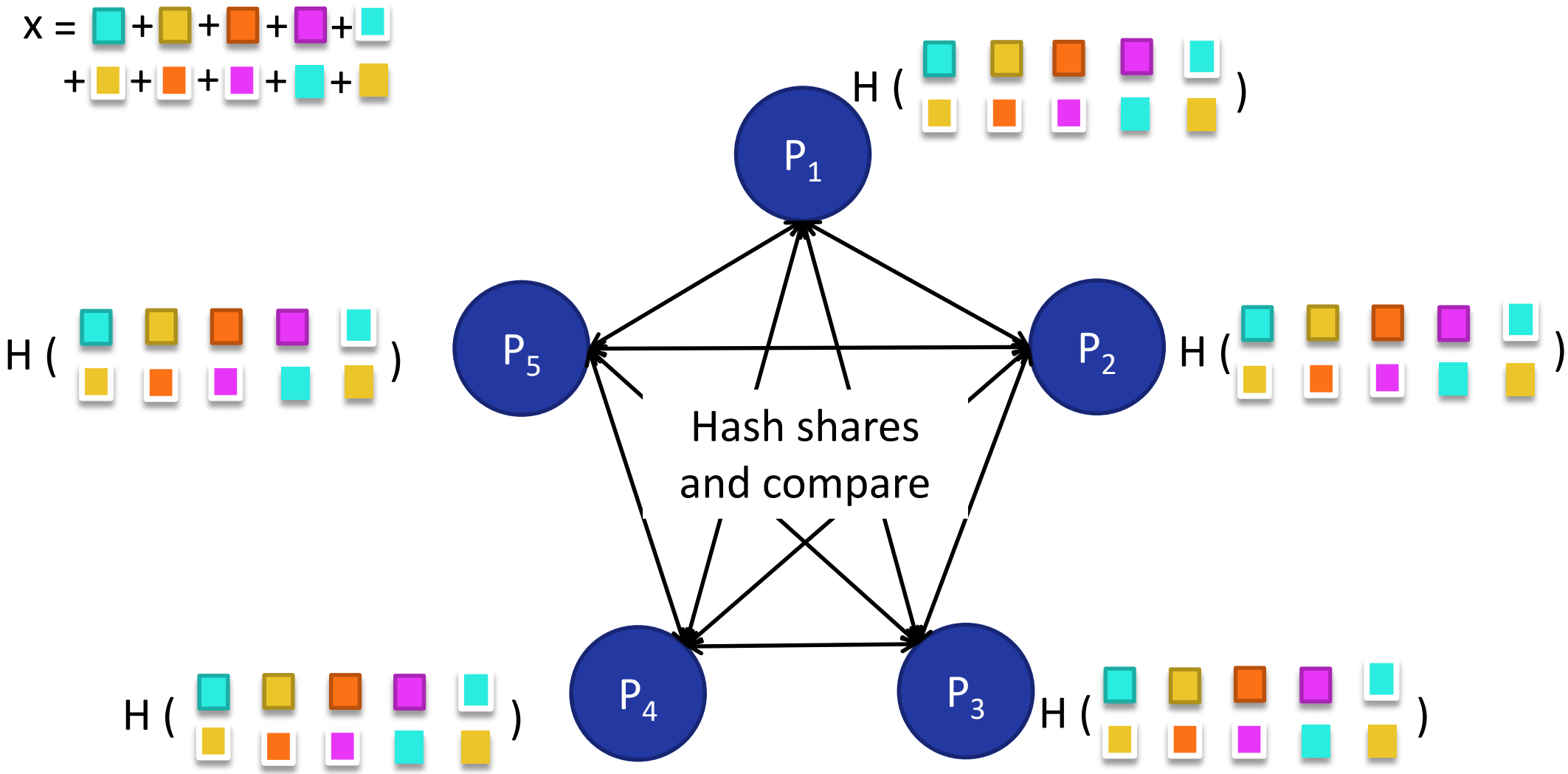




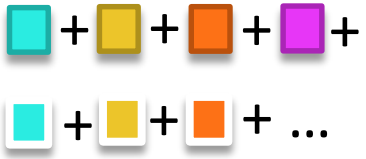




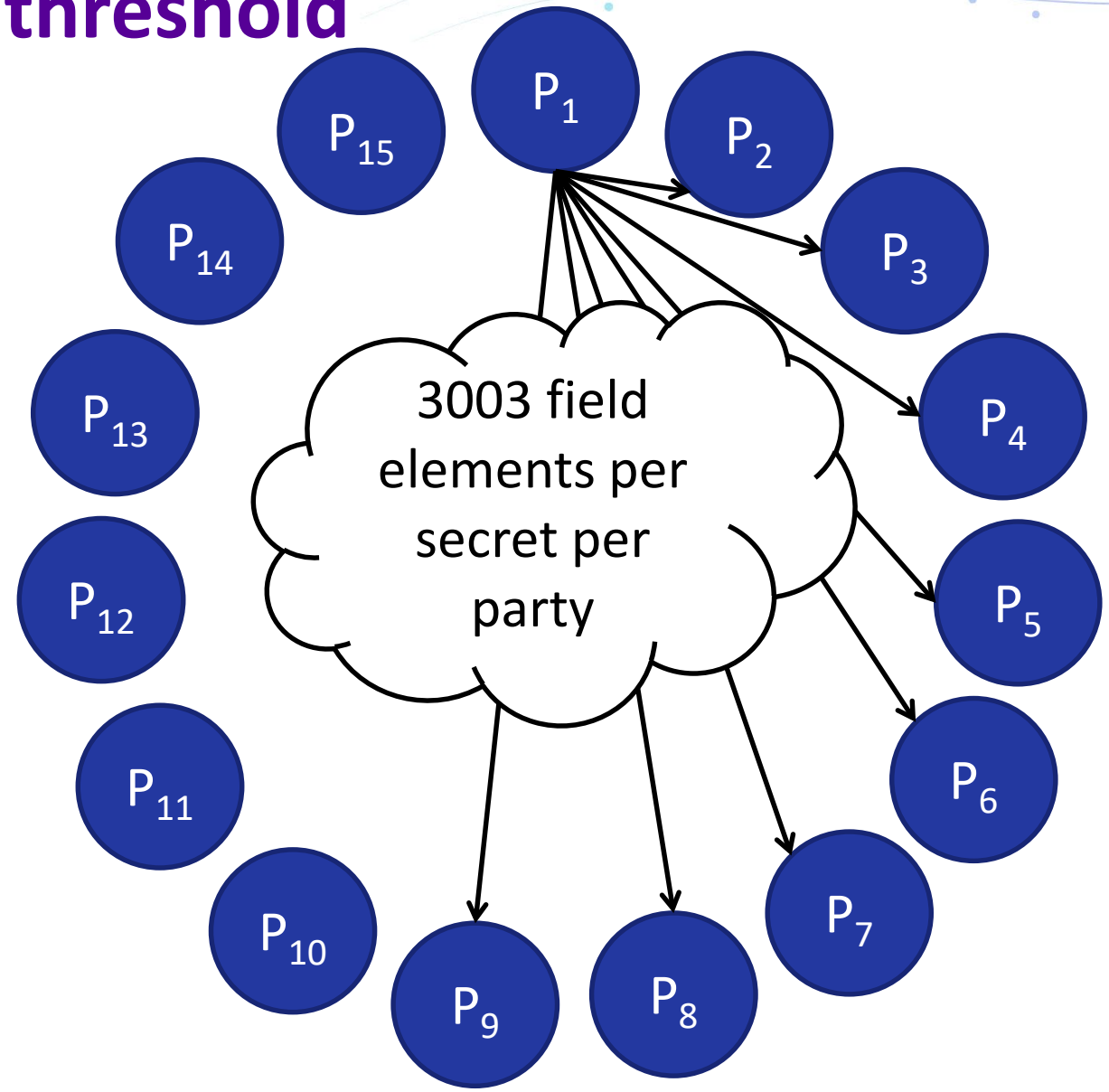
Previous work: KRSW18



KRSW (15,7) threshold



(6435 shares)



New idea

For n parties, exploit that the structure is Q_2: again,

Hash the shares and compare

(i.e. no need for replicated secret sharing!)

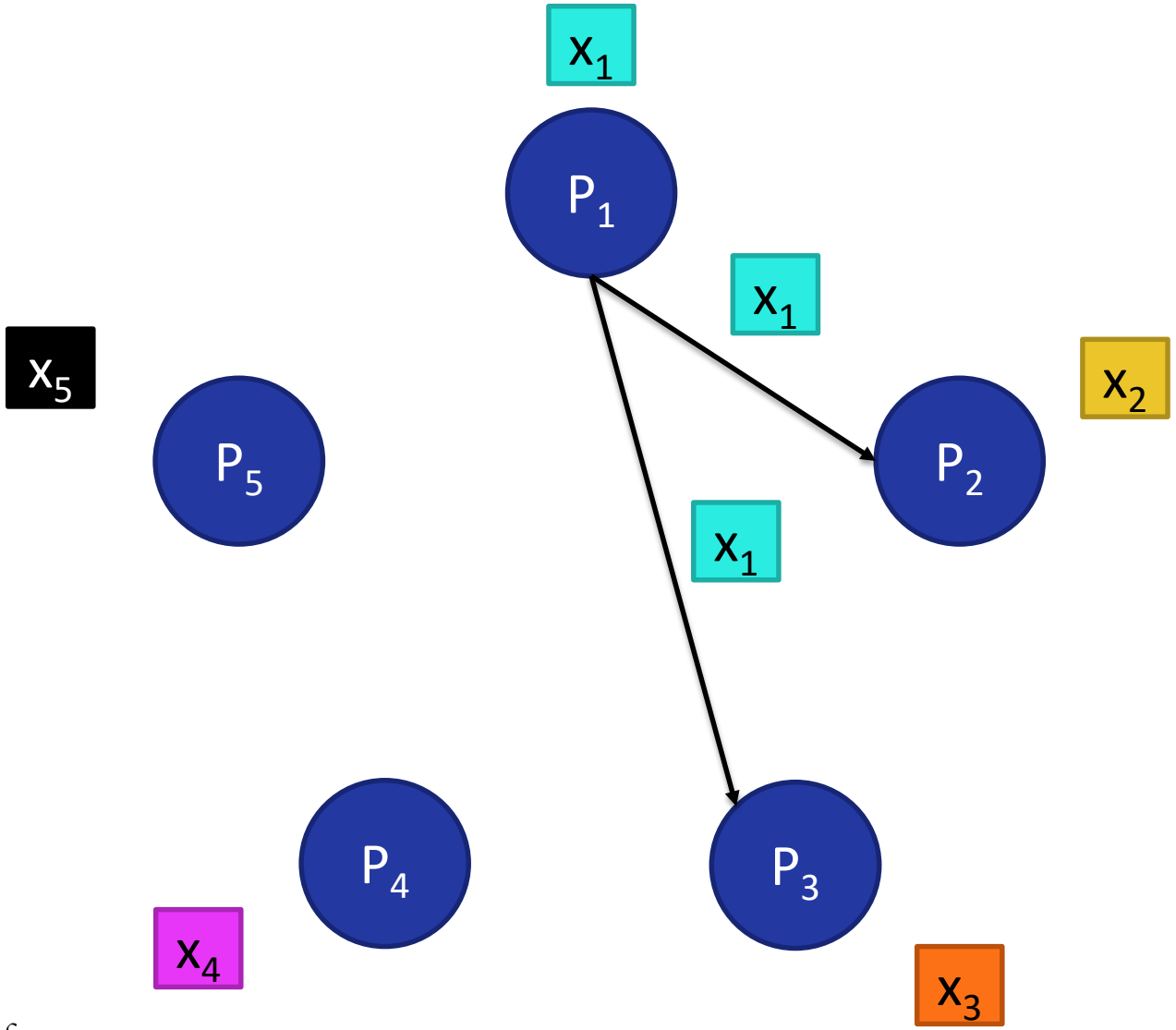
This work: any secret sharing

E.g. for (5,2) threshold, use Shamir:

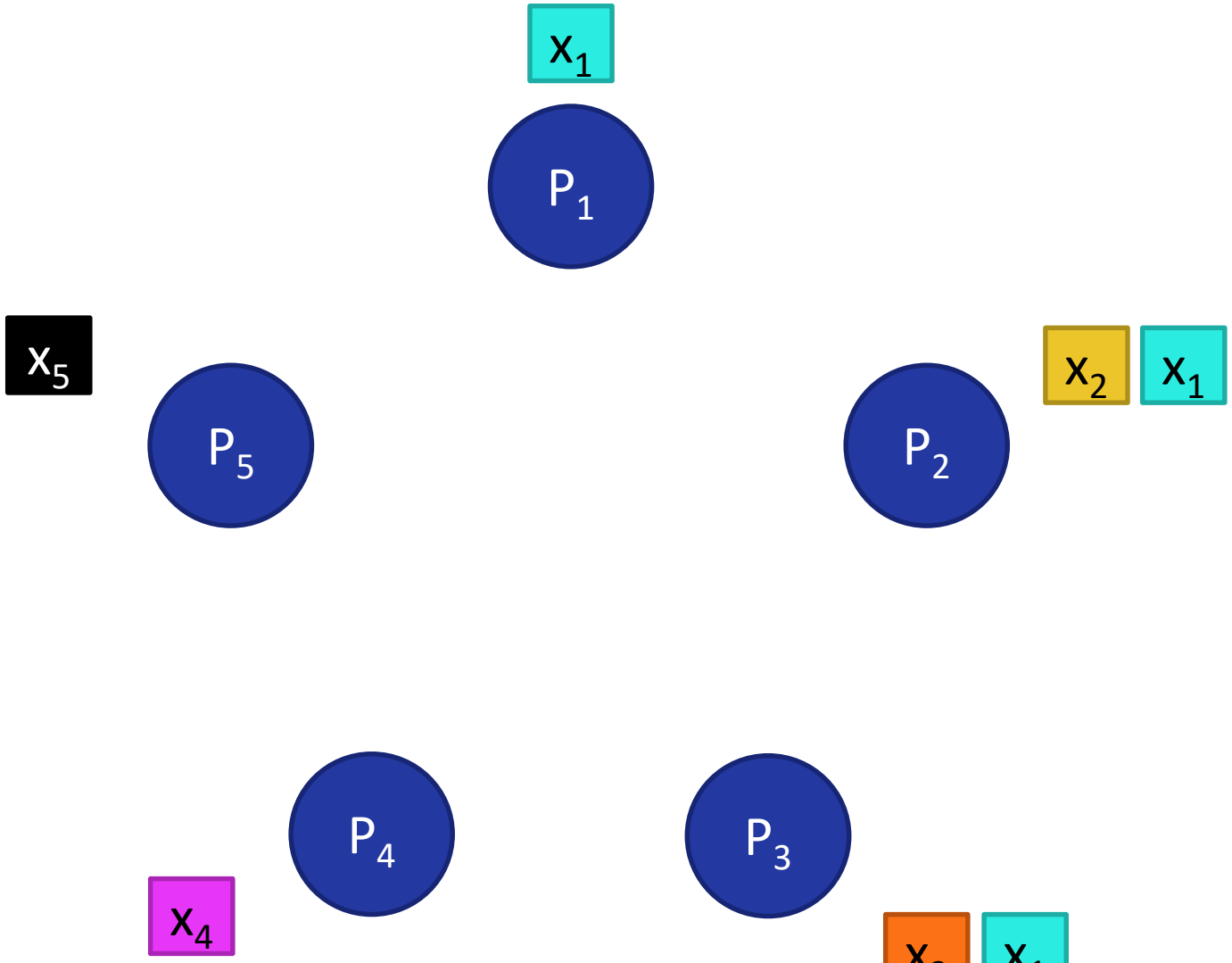
- Sample degree-2 poly f such that f(0) = x
- Fix x_i = (i, f(i)) and give x_i to i^{th} party
- Use Lagrange interpolation to *recover all shares* (and secret)

We show you can do this *because the access structure is* Q_2

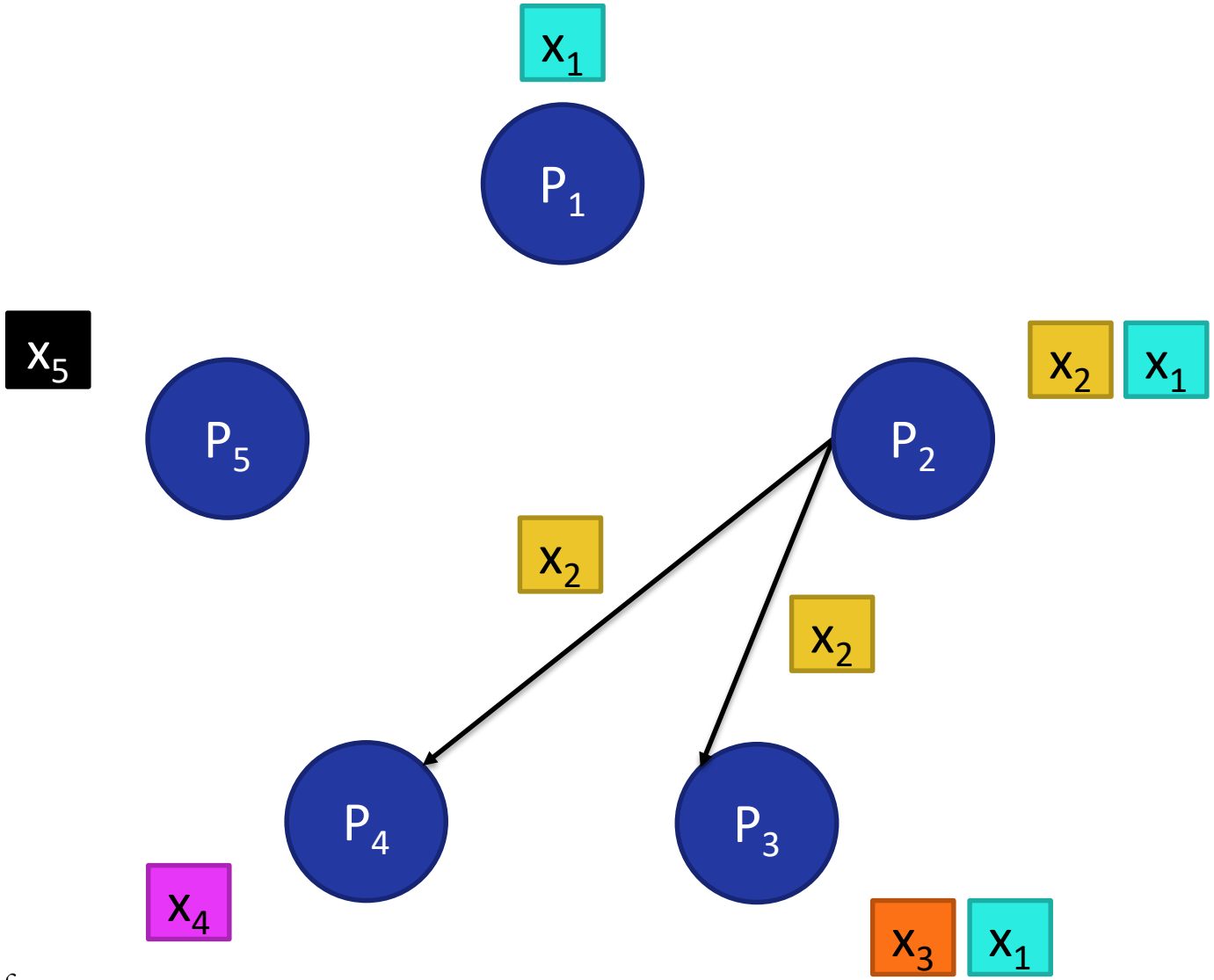
This work: any secret sharing



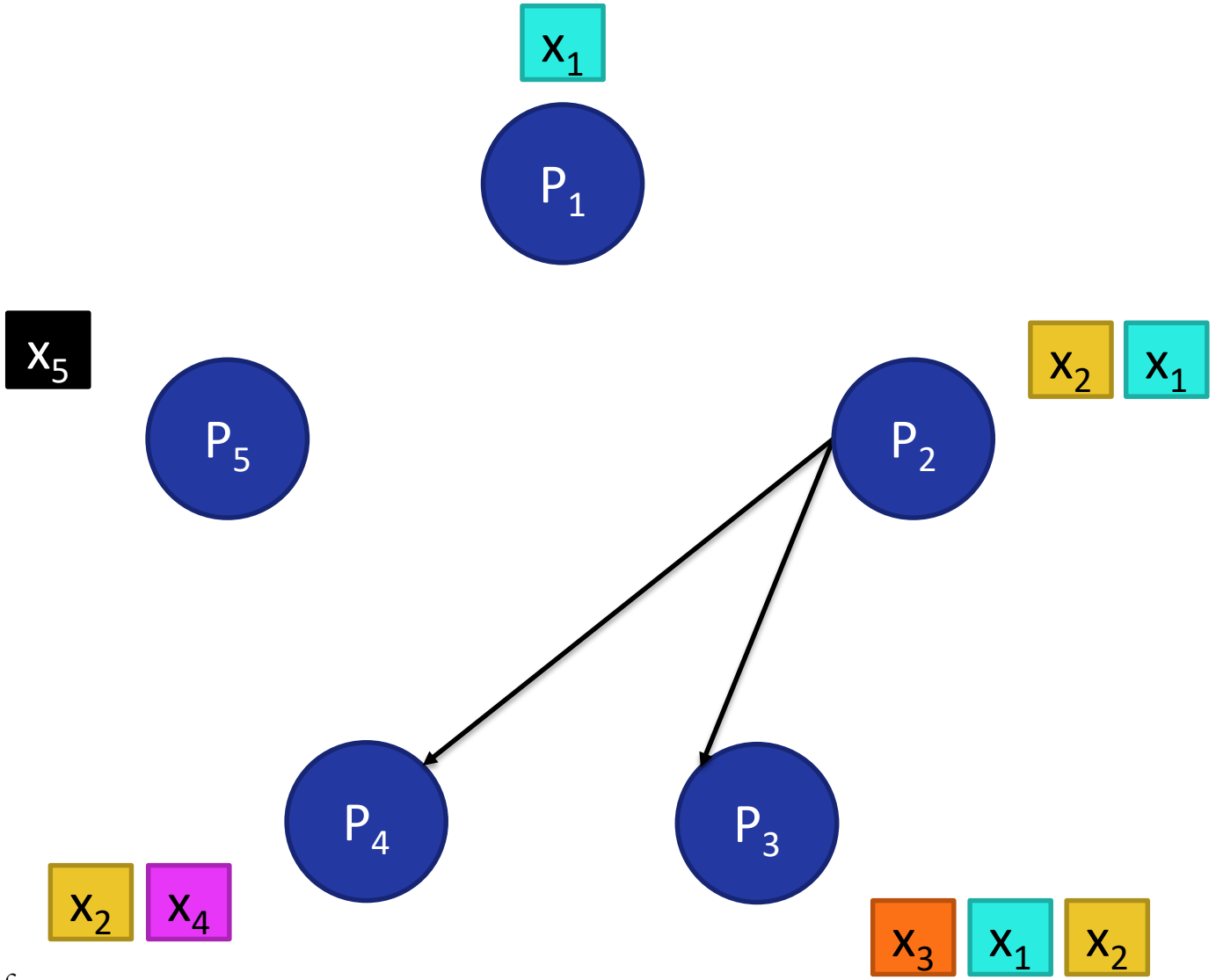
This work: any secret sharing



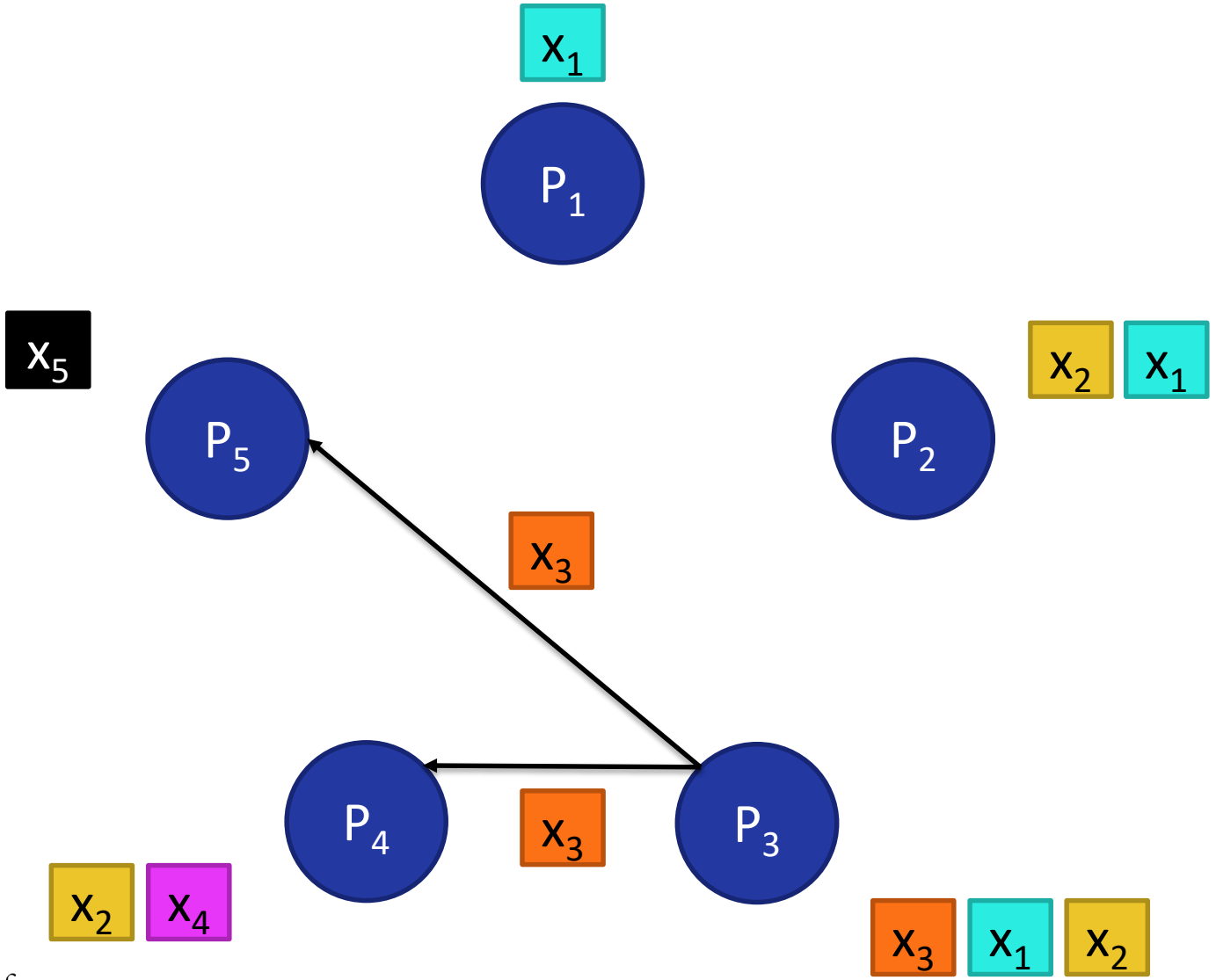
This work: any secret sharing



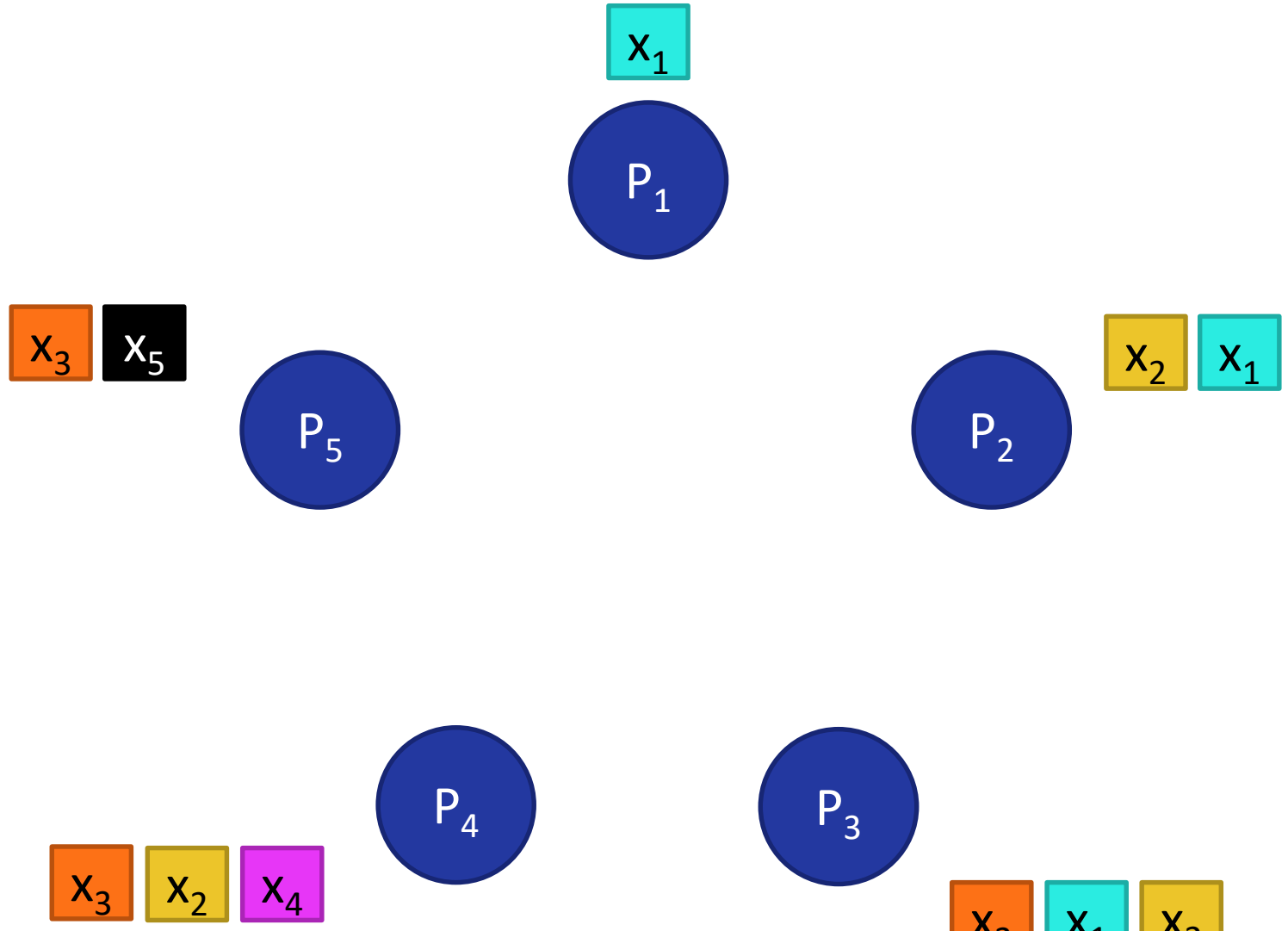
This work: any secret sharing



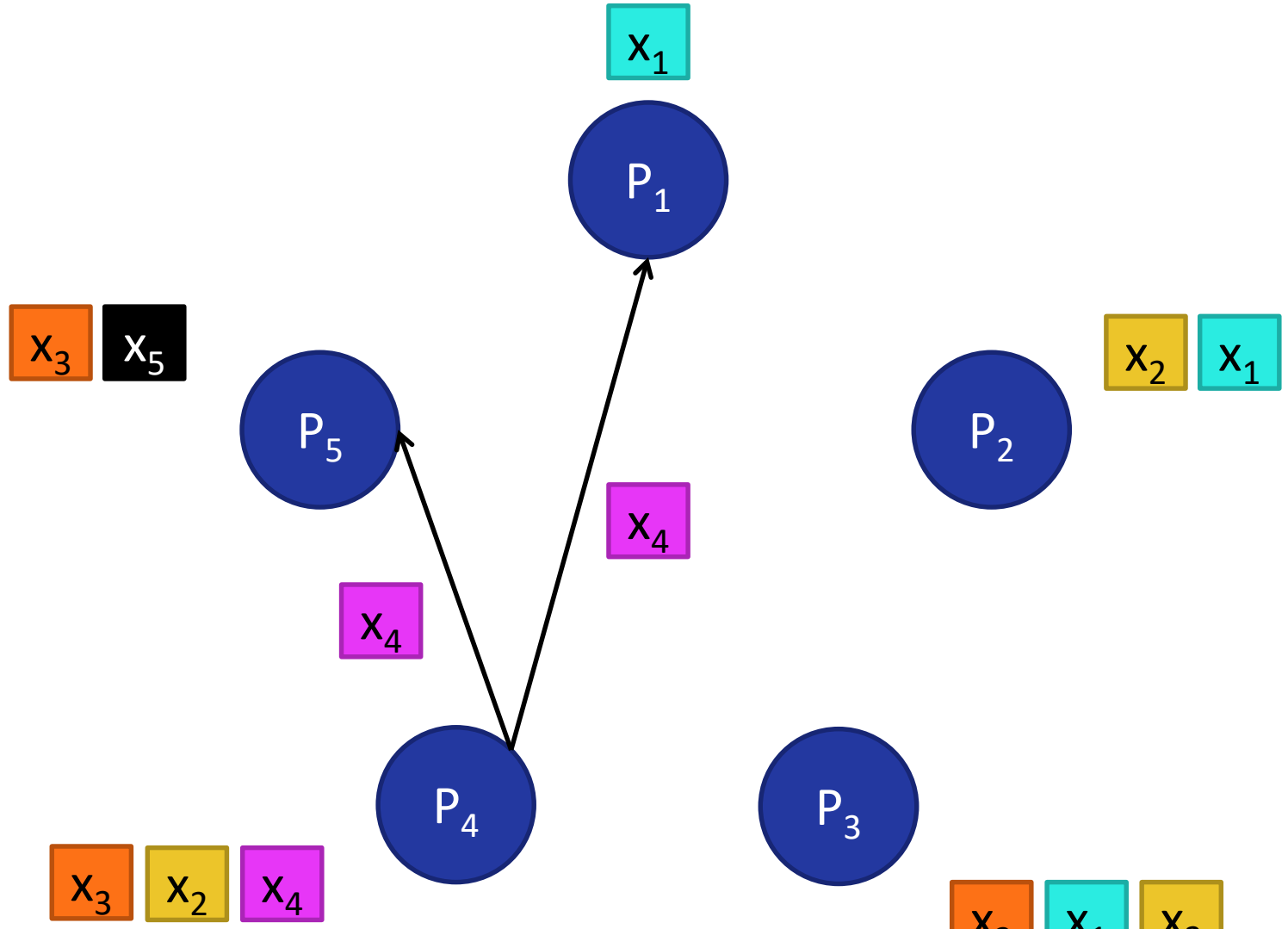
This work: any secret sharing



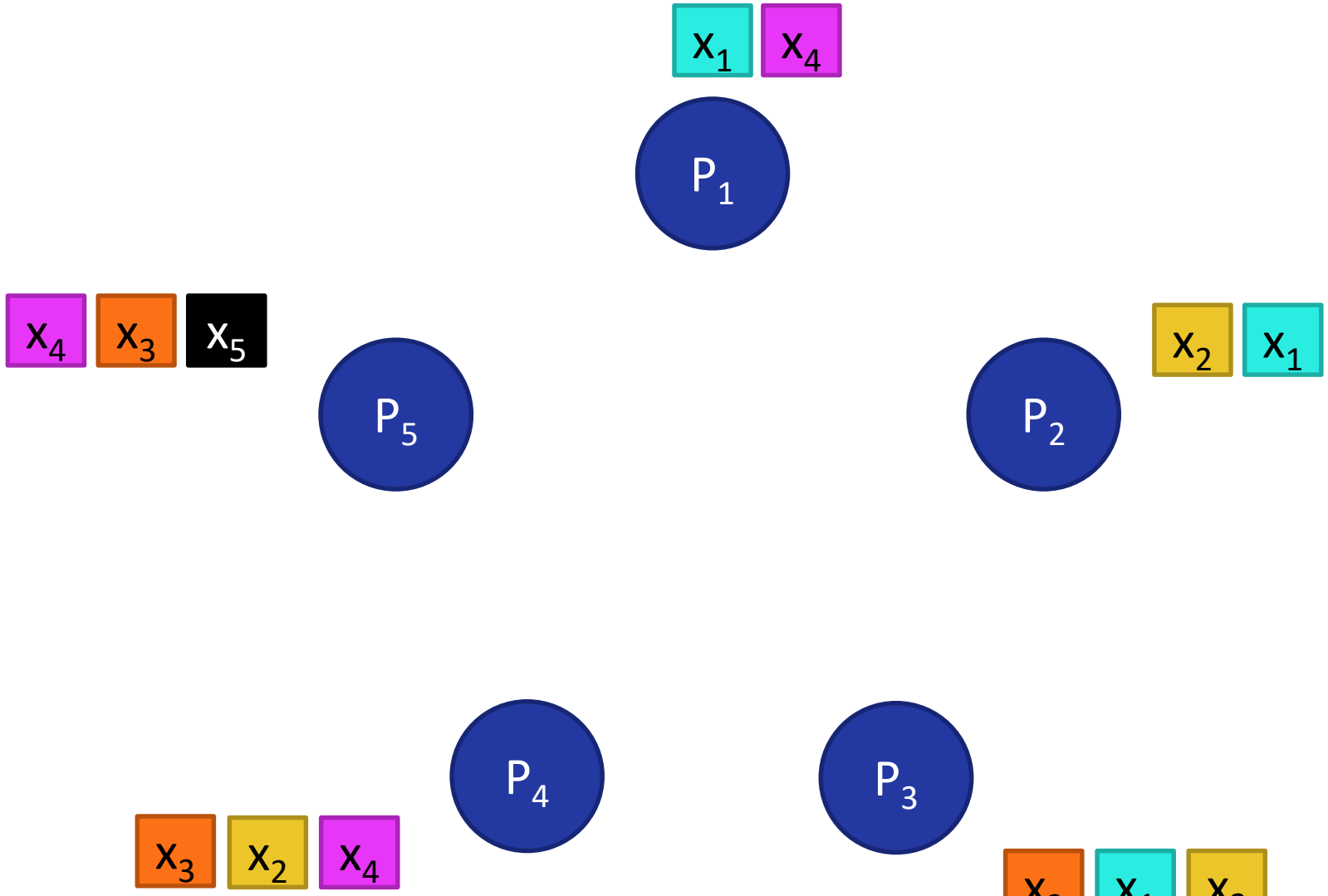
This work: any secret sharing



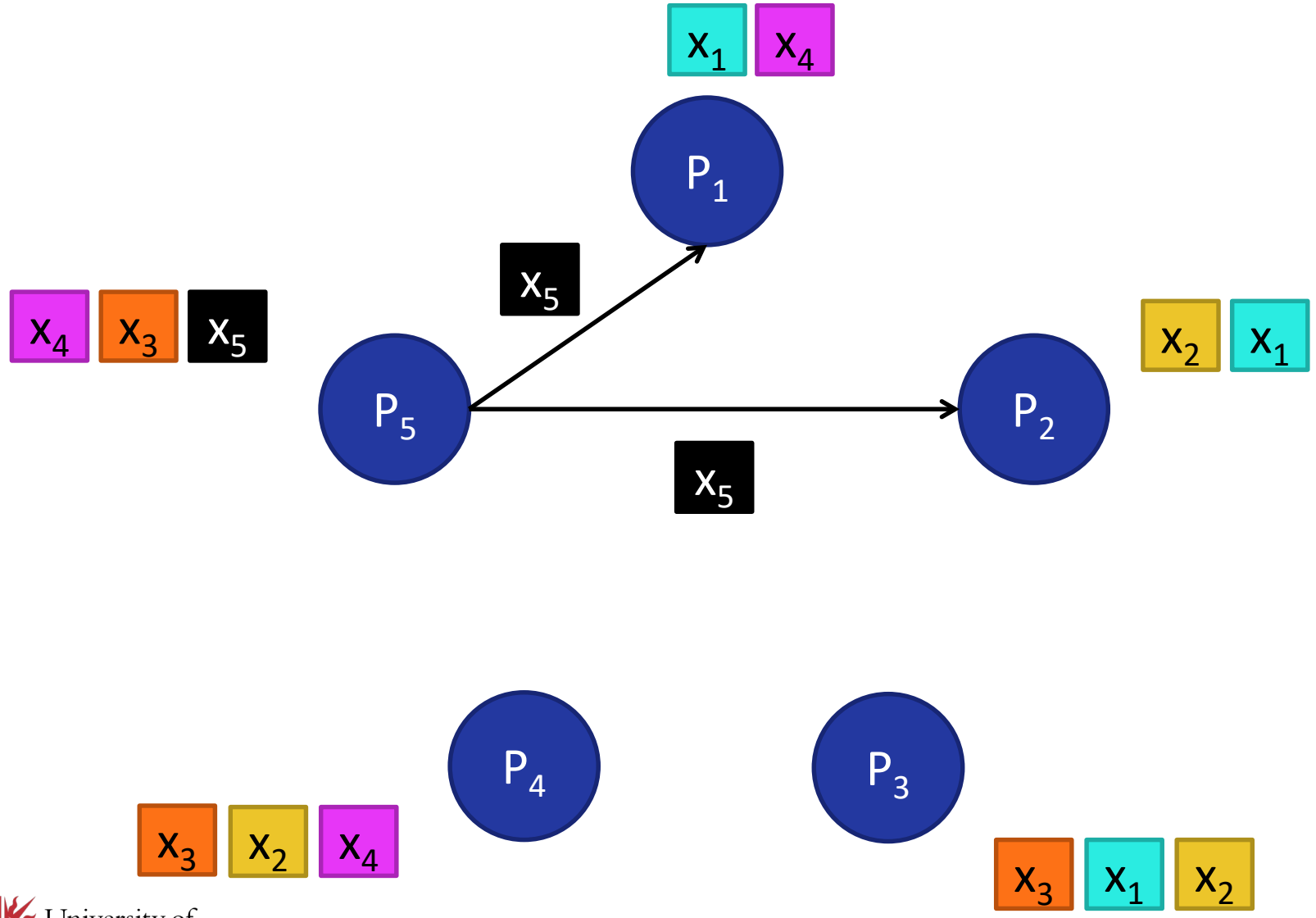
This work: any secret sharing



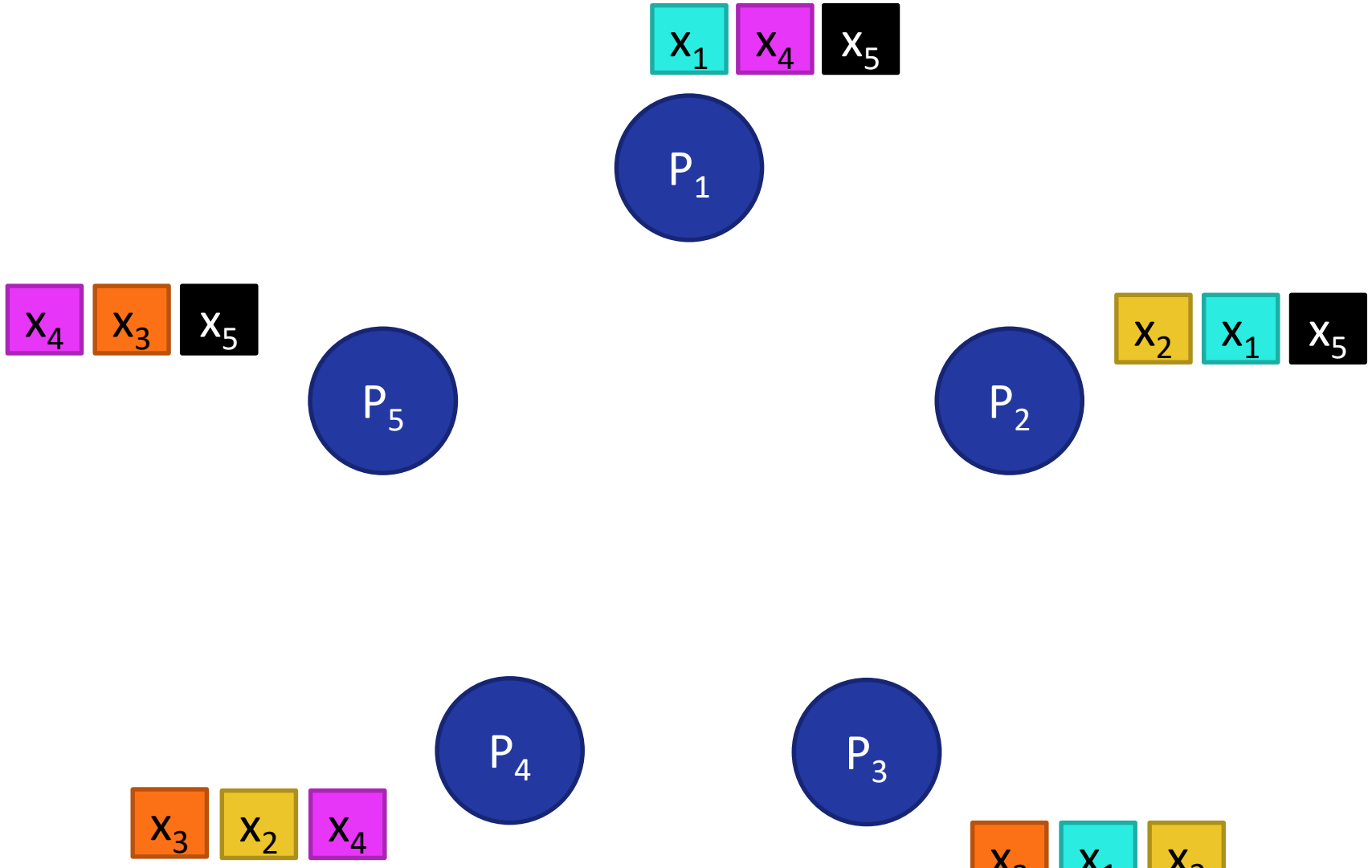
This work: any secret sharing



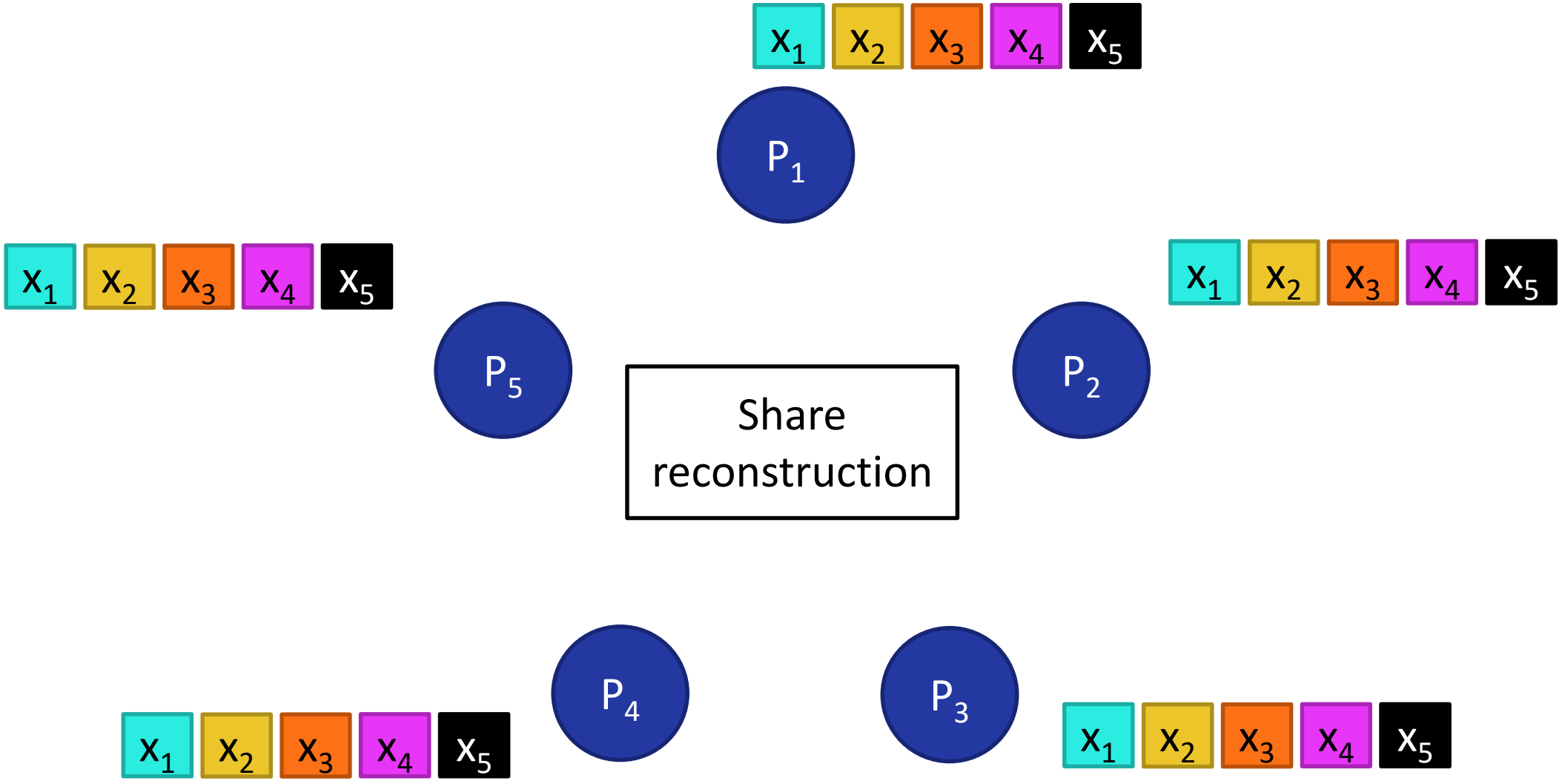
This work: any secret sharing



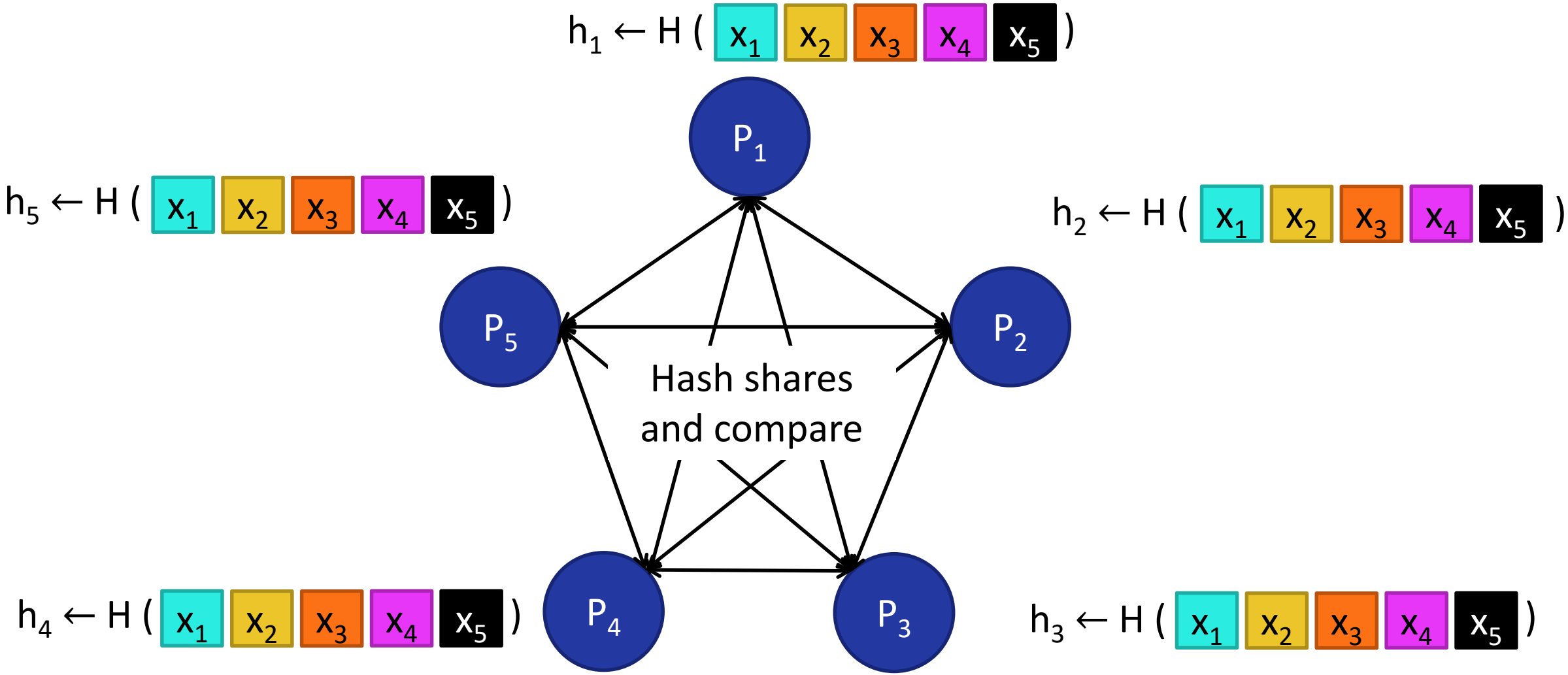
This work: any secret sharing



This work: any secret sharing

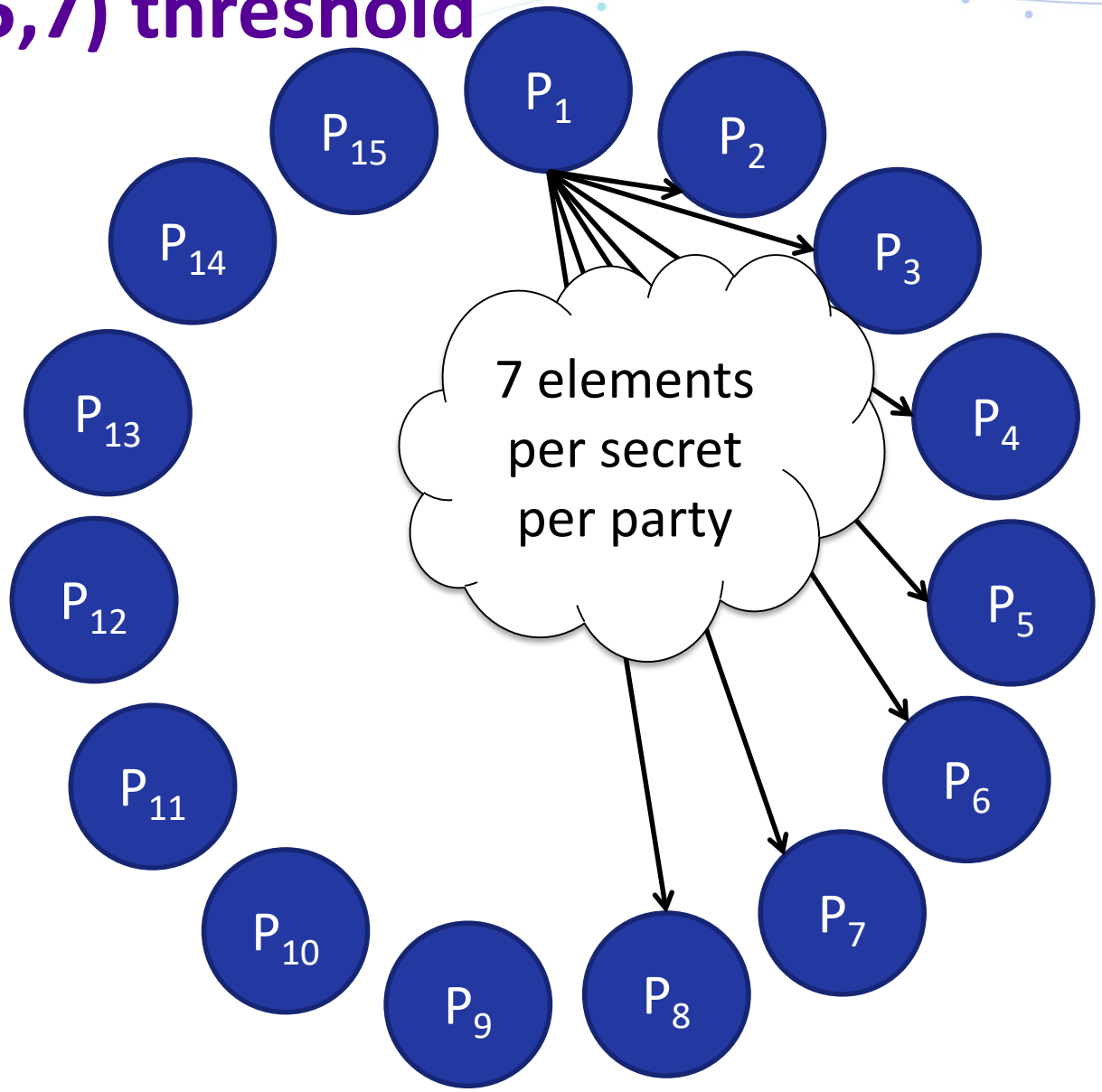


This work: any secret sharing



This work (15,7) threshold

15 shares, 1 per party



More generally we showed…

…this works for any Q_2 access structure

…and any secret sharing scheme realising the access structure

Application/Open questions

- Try it out!

  `https://github.com/KULeuven-COSIC/SCALE-MAMBA`

- Improve offline phase?
  - Producing Beaver triples can be expensive…
- Merge “Online/offline” into one (cf [CGHIKLN18])
- Find optimal secret sharing schemes for specific access structures?
  - Directly leads to more efficient MPC protocols

RSAConference2019

Thanks!

Questions?

RSA®Conference2019