UiT Norges arktiske universitet

Privacy preserving distributed computation

April 28th 2022

Finse Cybersecurity Winter School

Anders Andersen

Department of Computer Science

Faculty of Science and Technology

UiT The Arctic University of Norway

Privacy preserving distributed computation

- Two natural models for privacy preserving mechanisms:
  - non-interactive (offline)
    - a data curator collects the data and publishes a sanitized version of the data (anonymization or de-identification)
  - interactive (online)
    - the trusted entity provides an interface through which users can query the data and obtain (possibly noisy) answers
    - often used if no information about the queries is known in advance

Privacy preserving distributed computation

- Different approaches:
  - Data perturbation / Output perturbation
  - Access control
  - Query restriction / Query auditing
  - Summary statistics
  - Removal of specific identifiers (k-anonymity, l-diversity, t-closeness)
  - Differential privacy
  - Secure multiparty computation (SMC)
  - Information flow properties
  - Homomorphic encryption

Privacy preserving distributed computation

- Different approaches:
  - Data perturbation / Output perturbation
  - Access control
  - Query restriction / Query auditing
  - Summary statistics
  - Removal of specific identifiers (k-anonymity, l-diversity, t-closeness)
  - Differential privacy
  - Secure multiparty computation (SMC)
  - Information flow properties
  - Homomorphic encryption

Privacy preserving distributed computation

- Todays topics
  - Differential privacy
  - Secure multiparty computation (SMC)

Who am I?

- Professor in computer science
  - Cybersecurity
  - Machine learning
  - Mobile systems / context sensitivity / personalization
  - Adaptive middleware / distributed systems
  - Multimedia
- Head of department (IFI @ UiT)
  - Since 2019
- Other
  - Led national workgroup on ICT-security in education

A short advertising break

A short advertising break

(the only one)

A short advertising break

(a few words about IFI @ UiT)

5-year integrated master study programs IFI @ UiT

UiT Norges arktiske universitet

Datamaskinsystemer

– en femårig mastergrad i informatikk

UiT Norges arktiske universitet

Helseteknologi

Studiet kvalifiserer til en teknologisk yrkeskarriere i helseteknologi innen helsevesen og industri

UiT Institutt for informatikk

CYBERSIKKERHET

– En femårig mastergrad i informatikk

Cybersikkerhet sikrer det moderne samfunnet.

Vær med å form den teknologiske framtida.

Study programs IFI @ UiT

- Bachelor in informatics
  - 3-year
- Master in Computer Science
  - 2-year
- Integrated master
  - 5-year (sivilingeniør)
  - computer systems
  - health technology
  - cybersecurity
  - artificial intelligence (with IFT/IMS)

- About the studies
  - practical
  - good evaluation
  - industry relevance
  - good collaboration with industry
- Recruitment
  - mostly from Northern-Norway
  - increasing share of students from other places in Norway
  - some international students

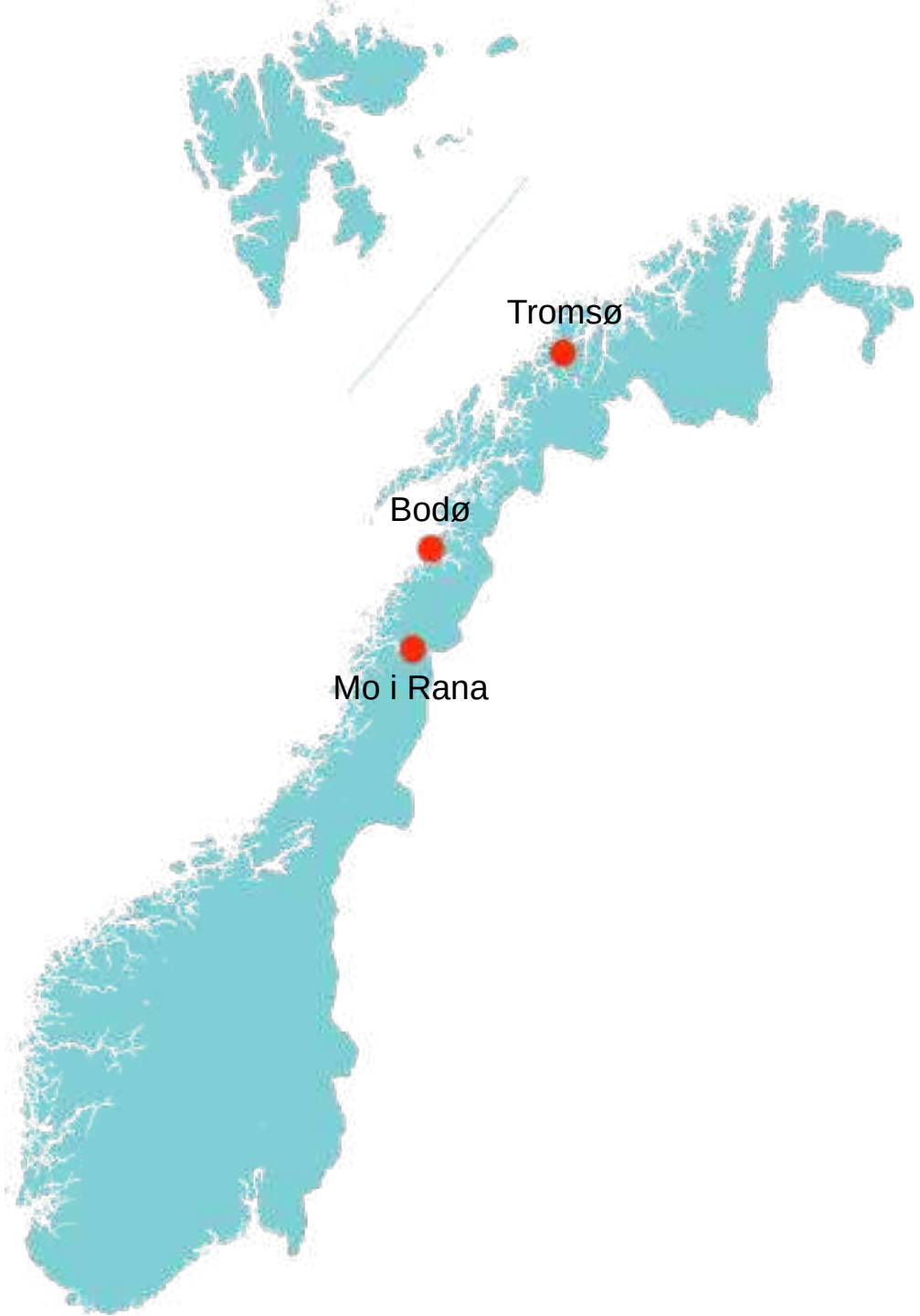
Study programs IFI @ UiT

- Bachelor in informatics
  - 3-year
- Master in Computer Science
  - 2-year
- Integrated master
  - 5-year (sivilingeniør)
  - computer systems
  - health technology
  - cybersecurity
  - artificial intelligence (with IFT/IMS)
- About the studies
  - practical
  - good evaluation
  - industry relevance
  - good collaboration with industry
- Recruitment
  - mostly from Northern-Norway
  - increasing share of students from other places in Norway
  - some international students

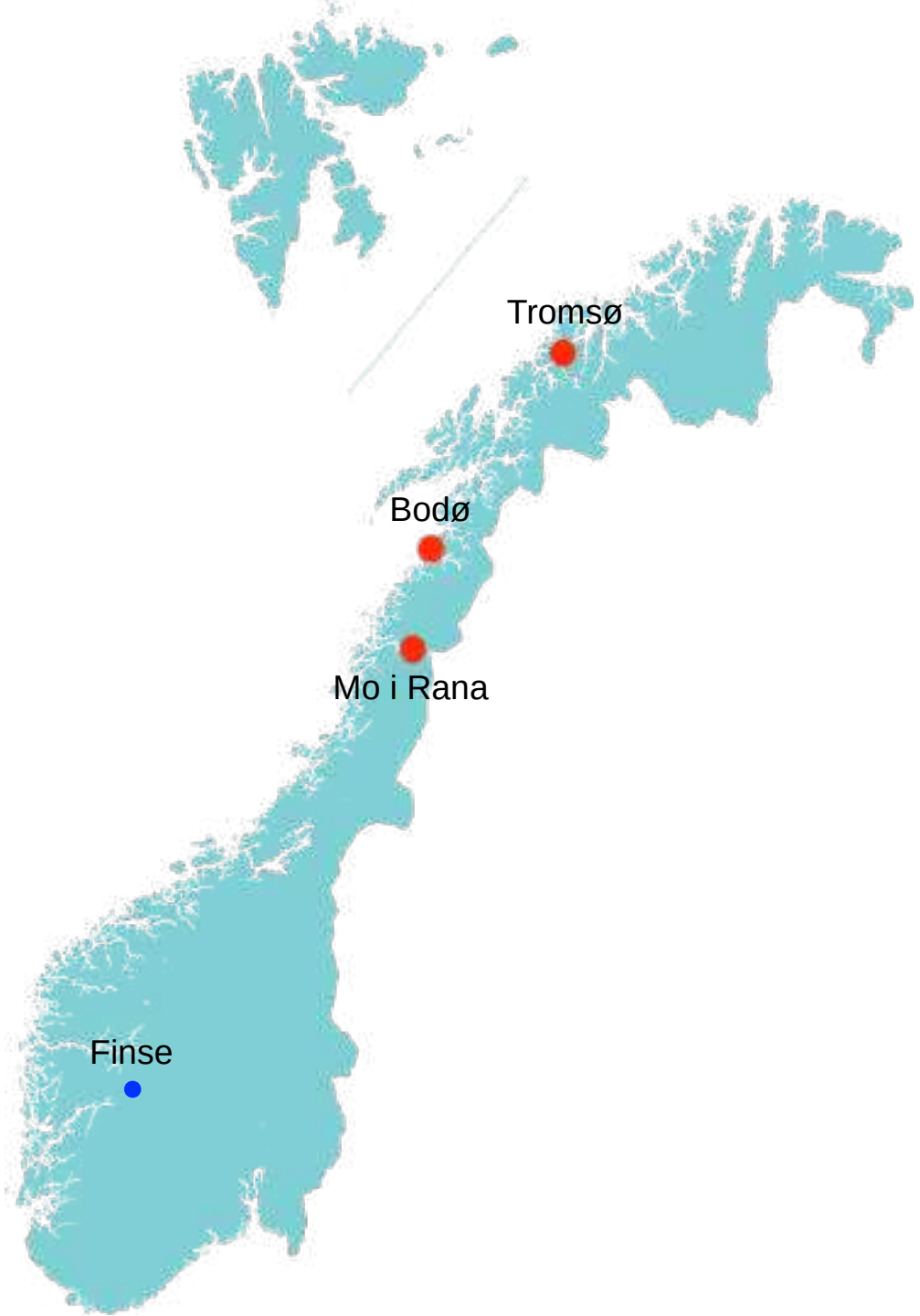
New study programs

- In Nordland, starting autumn 2023
  - 3-year bachelor at Helgeland
  - 2-year master in Bodø
- Continuing education
  - Experienced based master in digital health services, from 2022
    - Pilot at Helgeland in collaboration with the hospital at Helgeland
  - Individual topics
    - Programming, ICT-security, analytics (AI / ML)
  - In collaboration with others
    - Experienced master «Ocean leadership» (BFE, Jur-Fak)

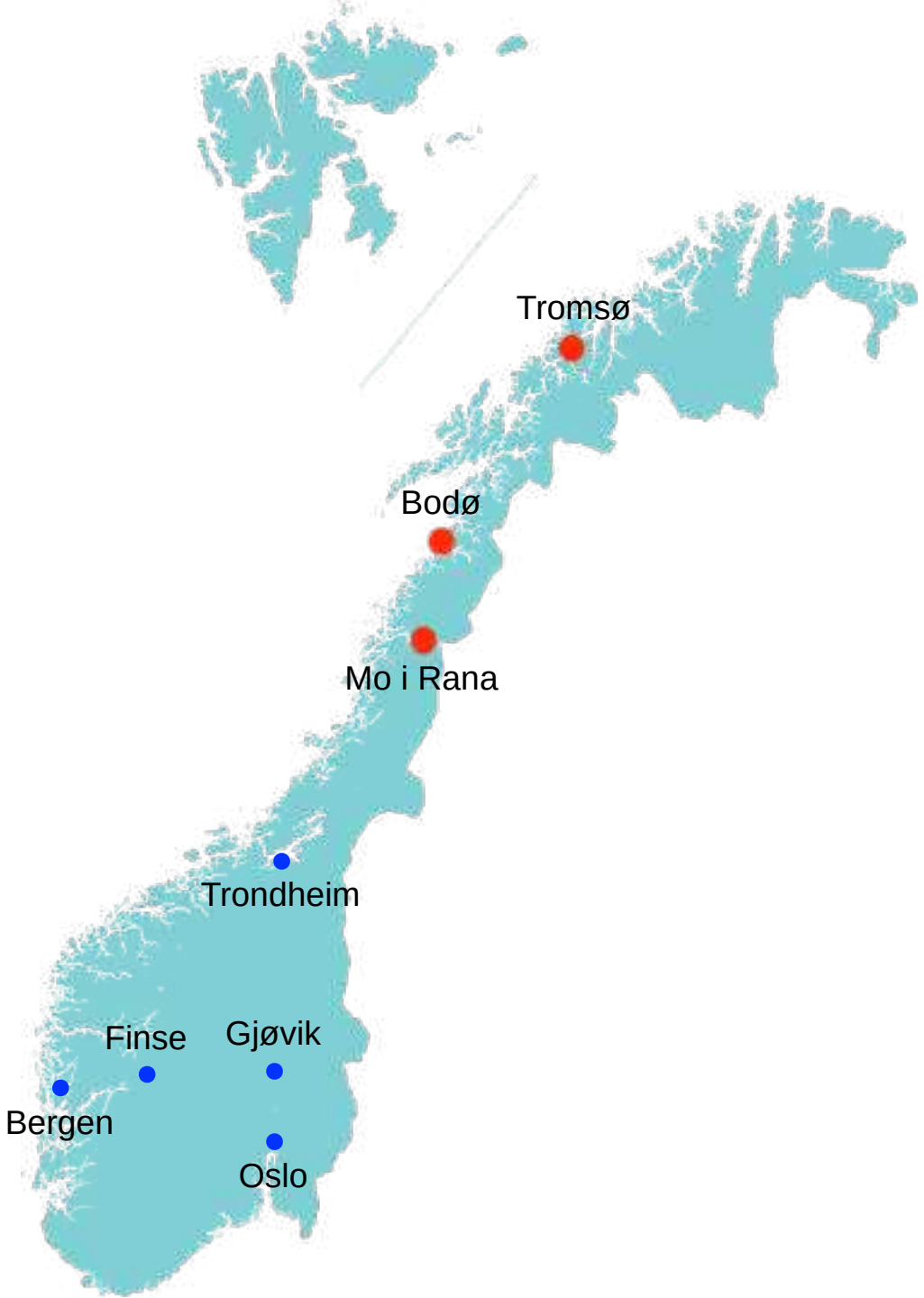
IFI @ UiT has become a multi-campus department



IFI @ UiT has become a multi-campus department



IFI @ UiT has become a multi-campus department



Research groups

- Cyber-Physical Systems
- Open Distributed Systems
- Health informatics and -technology
- Cyber Security Group
- Arctic Green Computing
- Health Data Lab
- Computational Analytics and Intelligence

Faggruppe 1 (O. Anshus)

Faggruppe 2 (R. Karlsen)

Faggruppe 3 (H. D. Johansen)

Back to the main program

Privacy preserving distributed computation

- Todays topics
  - Differential privacy
  - Secure multiparty computation (SMC)

Differential Privacy

Roughly, an algorithm is differentially private if an observer seeing its output cannot tell if a particular individual's information was used in the computation

Differential Privacy — Background

- Example: publish statistical data about businesses:
  - Collect sales-numbers by business categories and not individual businesses
  - Leave out the business categories with a single business
  - Now, the sales-numbers for individual businesses are not known

Category 1	
Category 2	
Category 3	
Category 4	
Category 5	
Category 6	
Total	2500

Differential Privacy — Background

- Example: publish statistical data about businesses:
  - Collect sales-numbers by business categories and not individual businesses
  - Leave out the business categories with a single business (only category 4)
  - Now, the sales-numbers for individual businesses are not known

Category 1	315
Category 2	890
Category 3	545
Category 4	x
Category 5	140
Category 6	360
Total	**2500**

Differential Privacy — Background

- Example: publish statistical data about businesses:
  - Collect sales-numbers by business categories and not individual businesses
  - Leave out the business categories with a single business (only category 4)
  - ~~Now, the sales-numbers for individual businesses are not known~~ (not true)

Category 1	315
Category 2	890
Category 3	545
Category 4	x
Category 5	140
Category 6	360
Total	**2500**

$x = 2500 - (315+890+545+140+360) = 250$

Differential Privacy — Background

- Tracker:
  - an adversary that could learn the confidential contents of a statistical database by creating a series of targeted queries and remembering the results (Denning, Denning, Schwartz 1979)
- Fundamental Law of Information Recovery:
  - it is impossible to publish arbitrary queries on a private statistical database without revealing some amount of private information (Nissim, Dinur 2003)
  - in the most general case, privacy cannot be protected without injecting some amount of noise → this led to the development of differential privacy

Differential Privacy

- Calibrating Noise to Sensitivity in Private Data Analysis (Dwork, McSherry, Nissim, Smith 2006):
  - formalized the amount of noise that needed to be added
  - proposed a generalized mechanism for doing so
- The concept of ε-differential privacy:
  - a mathematical definition for the privacy loss associated with any data release drawn from a statistical database
  - a person's privacy cannot be compromised by a statistical release if their data are not in the database
  - with differential privacy, the goal is to give each individual roughly the same privacy that would result from having their data removed

Differential Privacy

- Each individuals contribution to the result of a query:
  - how much any individual contributes to the result of a database query depends in part on how many people's data are involved in the query
  - If the database contains data from a single person, that person's data contributes 100%
  - If the database contains data from a hundred people, each person's data contributes 1%
- Key insight of differential privacy:
  - as the query is made on the data of fewer and fewer people, more noise needs to be added to the query result to produce the same amount of privacy

Differential Privacy — final comments

- The tradeoff between data utility and individual privacy:
  - If the privacy loss parameter is set to favor utility, the privacy benefits are lowered (less “noise” is injected into the system)
  - if the privacy loss parameter is set to favor heavy privacy, the accuracy and utility of the dataset are lowered (more “noise” is injected into the system)
- Data privacy and security:
  - differential privacy is robust to still unknown privacy attacks
  - however, it encourages greater data sharing, which if done poorly, increases privacy risk
  - depends on the privacy loss parameter chosen and may lead to a false sense of security

Differential Privacy — some publication

- *Differential Privacy: A Survey of Results*, C. Dwork. TAMC 2008, LNCS 4978, pp. 1–19, 2008. Springer-Verlag 2008.
- *The Algorithmic Foundations of Differential Privacy*, C. Dwork, A. Roth. Foundations and Trend in Theoretical Computer Science Vol. 9, Nos. 3–4 (2014) 211–407, 2014.
- *The Tracker: A Threat to Statistical Database Security,* D. E. Denning; P. J. Denning; M. D. Schwartz. ACM Transactions on Database Systems, Vol. 4, No. 1, March 1979, Pages 76-96.
- *Revealing information while preserving privacy,* I. Dinur and K. Nissim. In Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '03). ACM.
- *Calibrating Noise to Sensitivity in Private Data Analysis,* Cy. Dwork, F. McSherry, K. Nissim, A. Smith. In Theory of Cryptography Conference (TCC), Springer, 2006 / Journal of Privacy and Confidentiality, 7 (3), 17-51.

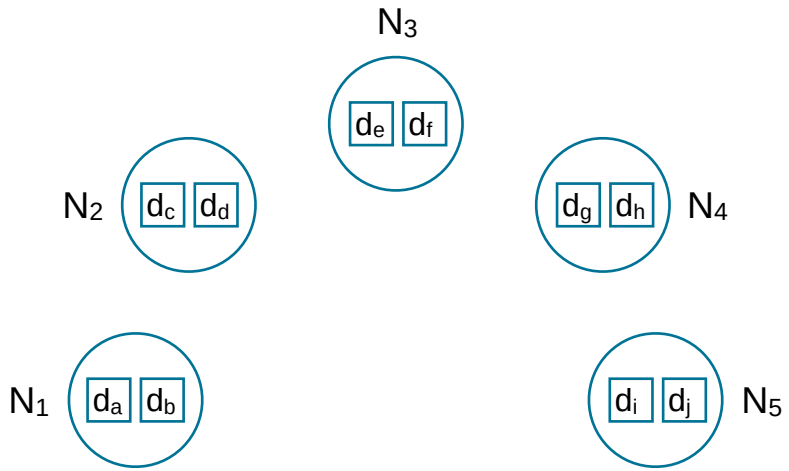
This is a break in the lecture

—

This slide separates the previous part from the next part

Privacy preserving distributed computation

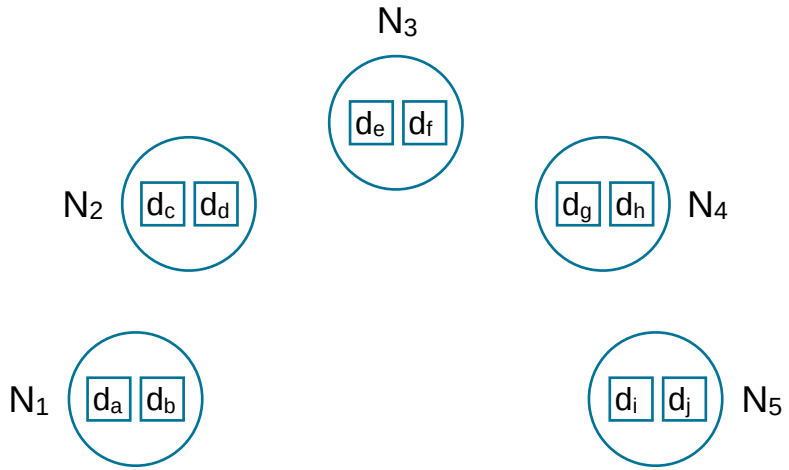
- The problem
  - Data distributed on nodes
  - Data from multiple nodes included in computation
  - Leakage of original data between nodes problematic



Computation including data d_a, …, d_j from node N_1, …, N_5

Privacy preserving distributed computation

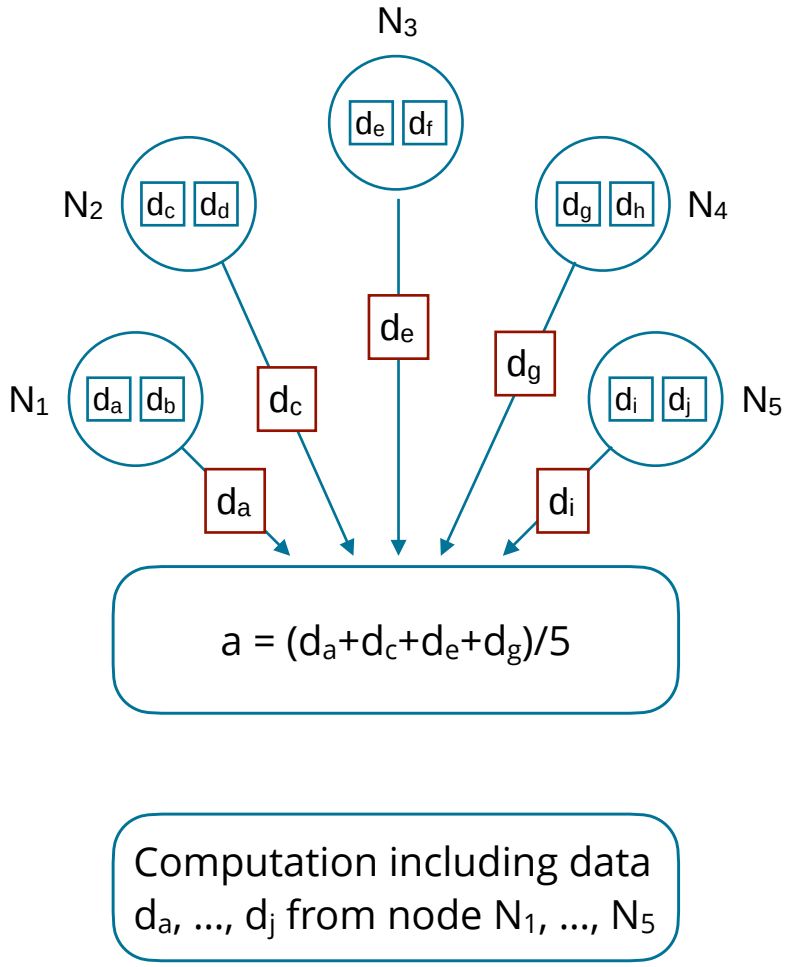
- Simple example
  - Compute the average value of d_a, d_c, d_e, d_g, and d_i



Computation including data d_a, …, d_j from node N_1, …, N_5

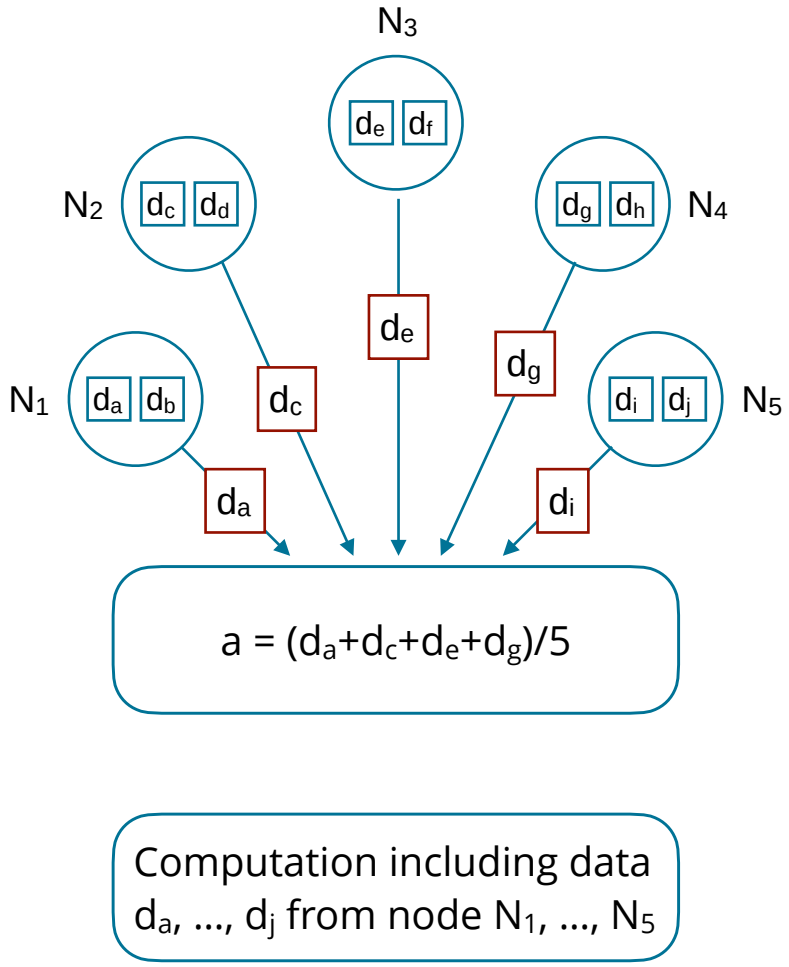
Privacy preserving distributed computation

- Simple example
  - Compute the average value of d_a, d_c, d_e, d_g, and d_i



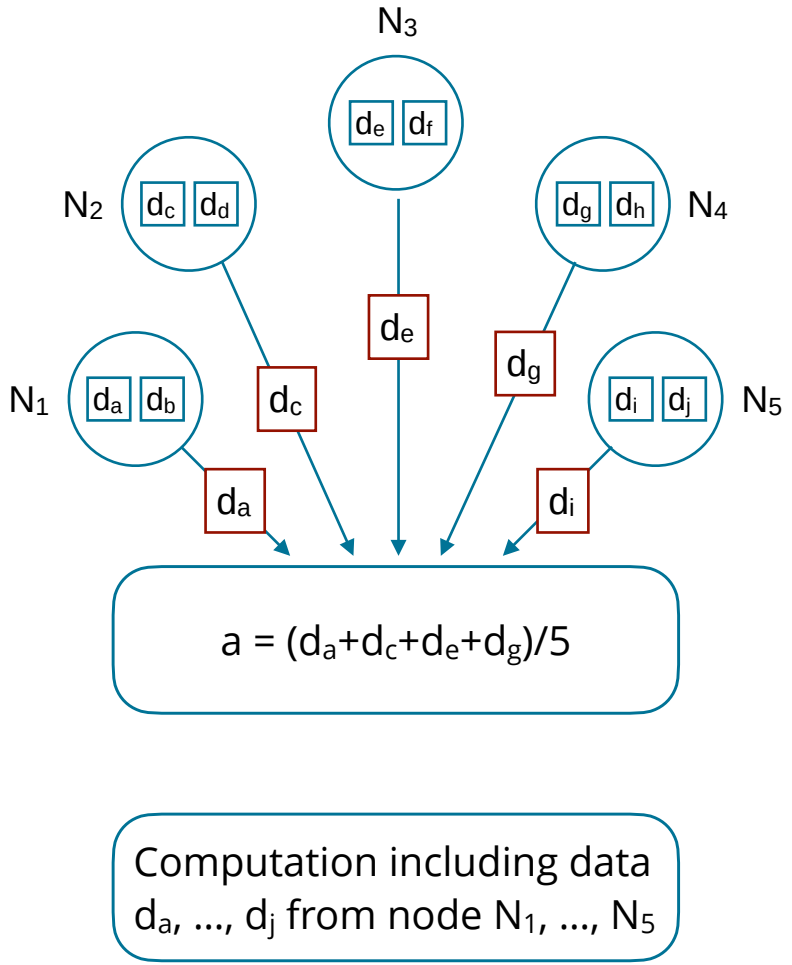
Privacy preserving distributed computation

- Simple example
  - Compute the average value of d_a, d_c, d_e, d_g, and d_i
  - Cannot collect the values at a single node (leakage)



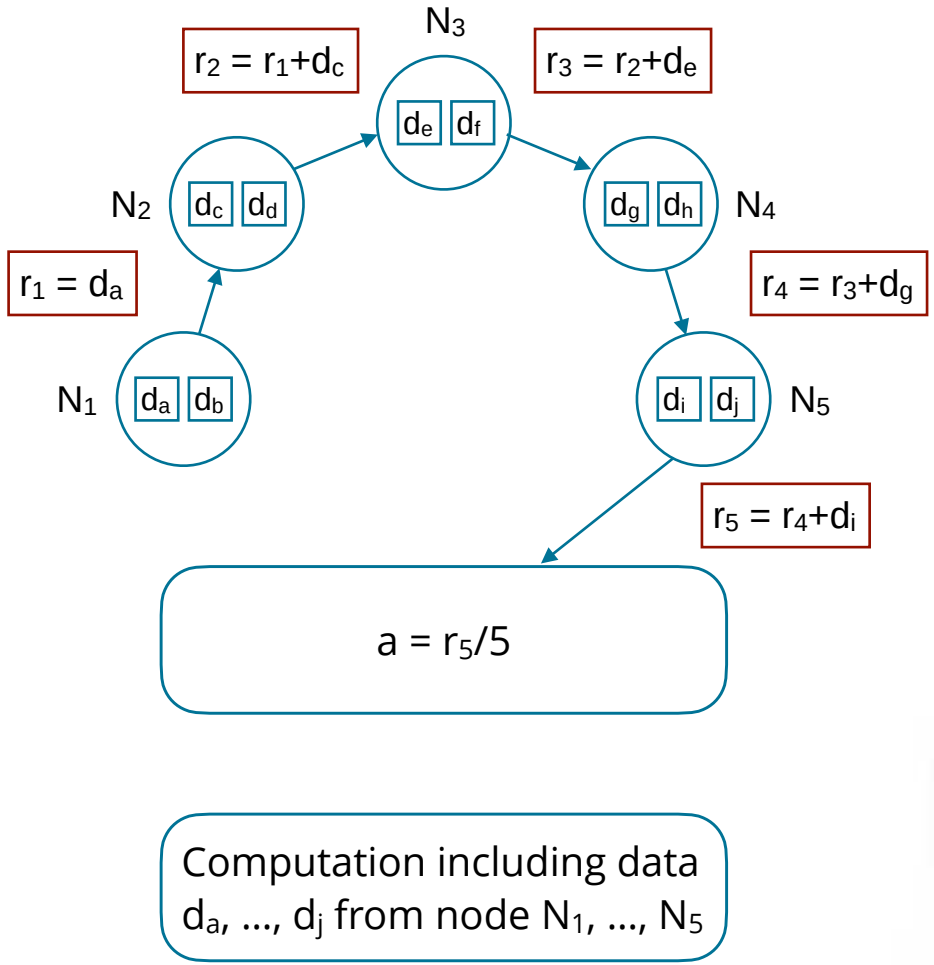
Privacy preserving distributed computation

- Simple example
  - Compute the average value of d_a, d_c, d_e, d_g, and d_i
  - Cannot collect the values at a single node (leakage)
  - Combine local and distributed computation?
  - Prepare values?



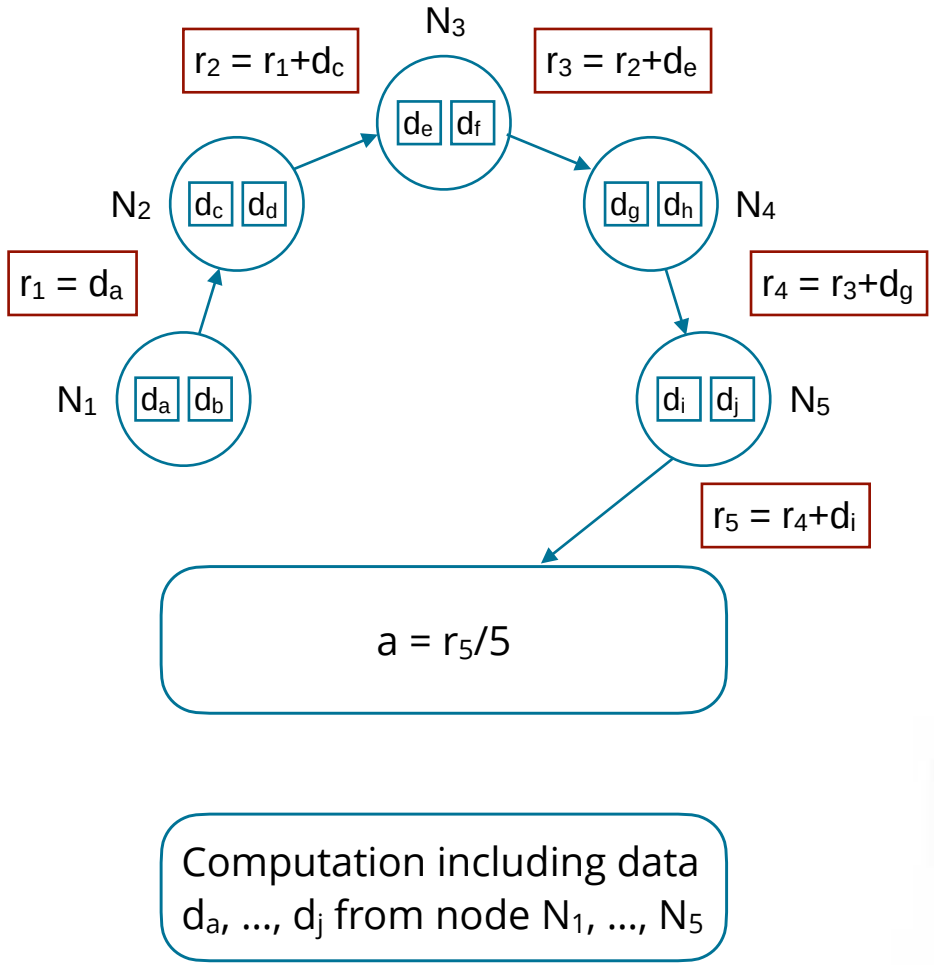
Secure multiparty computation

- Improved example
  - Don’t send individual data directly to node doing the computation
  - Distribute the current sum between the nodes



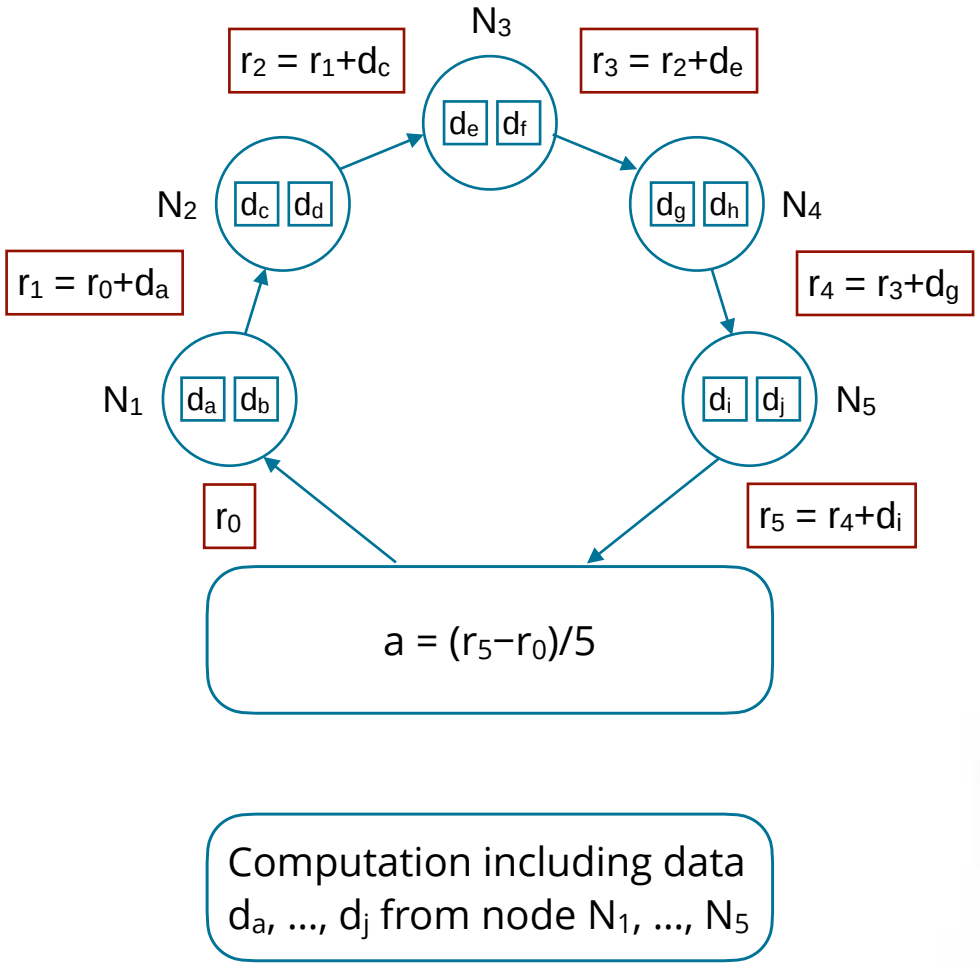
Secure multiparty computation

- Improved example
  - Don’t send individual data directly to node doing the computation
  - Distribute the current sum between the nodes
  - Individual data and small set data still exposed



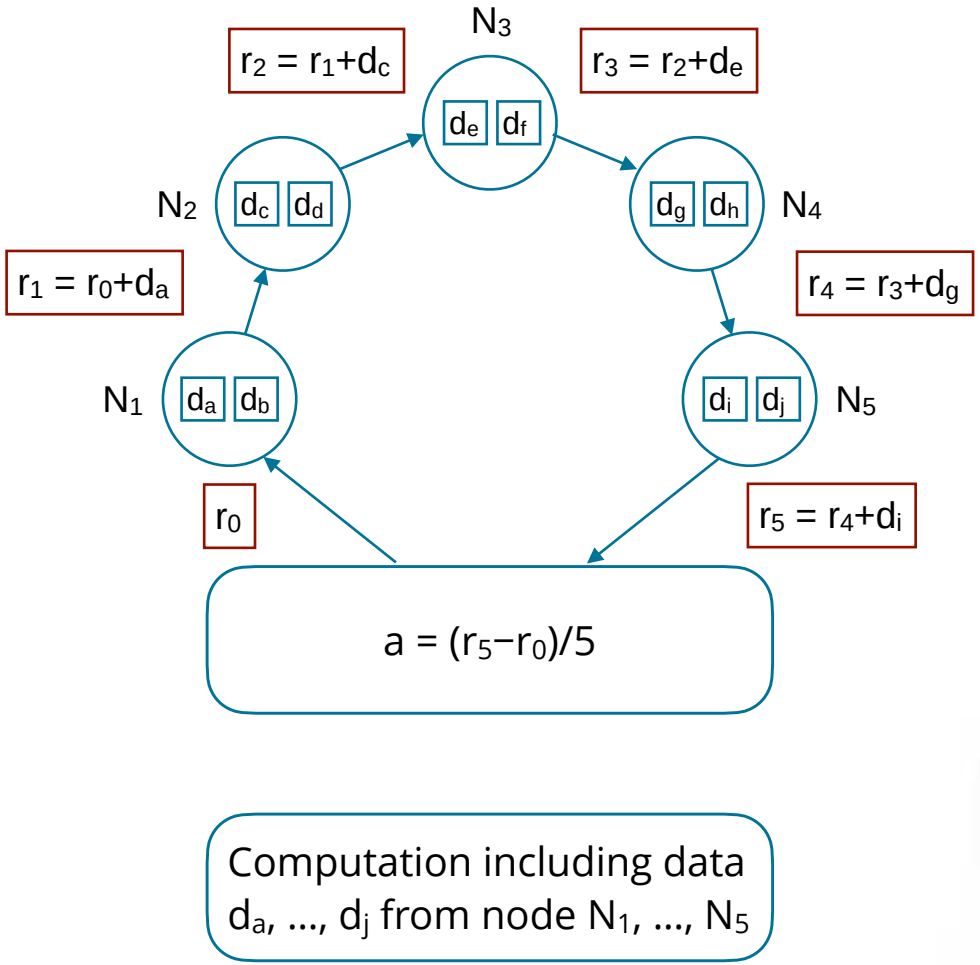
Secure multiparty computation

- Second improved example
  - Start with a large random number r_0



Secure multiparty computation

- Second improved example
  - Start with a large random number r_0
  - We have a secure multiparty computation (SMC) algorithm
  - Each node does not increase their knowledge about the other nodes data



Secure multiparty computation

SMC for N institutions with different data sets who wish to evaluate a result r (e.g. the correlation of usage of medication x and the adverse effect y) based on the data sets is subject to four constrains:

1. The correct value result r is obtained and known to all institutions.
2. No institution learns more about the other institutions values than it can deduce from its own data set and the result r.
3. No trusted third party—human or machine—is part of the process.
4. Semi-honesty: Institutions perform agreed upon computations correctly using their true data. However, they are permitted to retain the results of intermediate computations.

Secure multiparty computation

The four constrains

SMC (Goldwasser 1997, Karr 2009) for N institutions with different data sets who wish to evaluate a result r (e.g. the correlation of usage of medication x and the adverse effect y) based on the data sets is subject to four constrains:

1. The correct value result r is obtained and known to all institutions.
2. No institution learns more about the other institutions values than it can deduce from its own data set and the result r.
3. No trusted third party—human or machine—is part of the process.
4. Semi-honesty: Institutions perform agreed upon computations correctly using their true data. However, they are permitted to retain the results of intermediate computations.

Secure multiparty computation
The four constrains

For practical purposes, the zero information disclosure implied by constrain 3 is sometimes difficult to implement efficiently (Du, Zhan 2002).

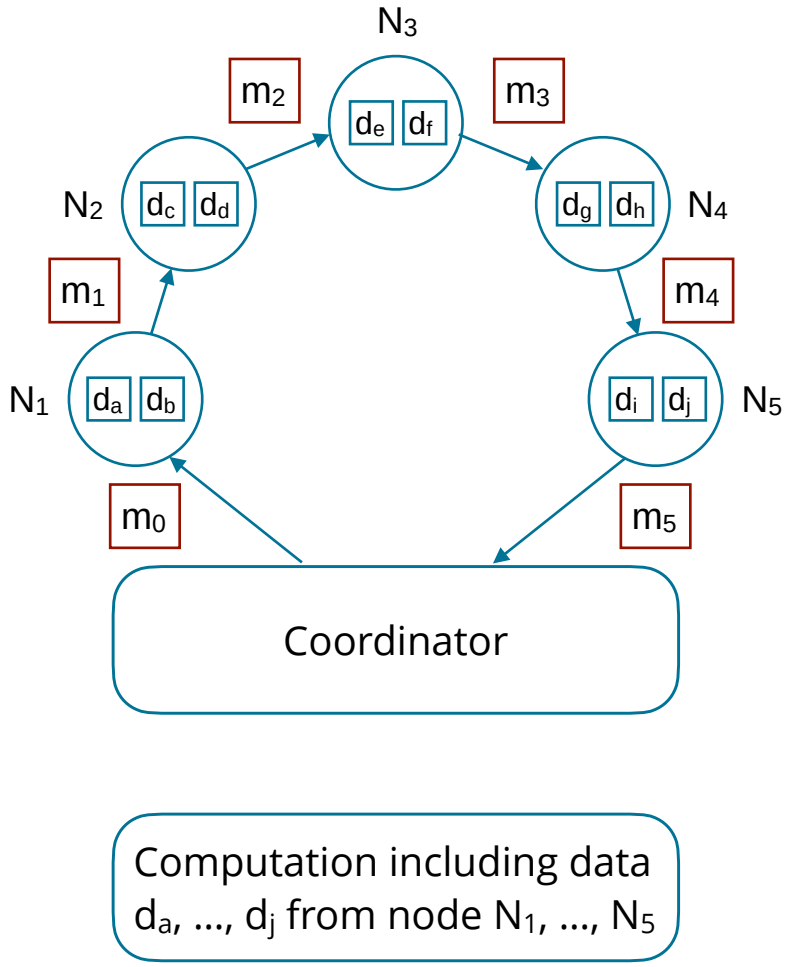
SMC (Goldwasser 1997, Karr 2009) for N institutions with different data sets who wish to evaluate a result r (e.g. the correlation of usage of medication x and the adverse effect y) based on the data sets is subject to four constrains:

1. The correct value result r is obtained and known to all institutions.
2. No institution learns more about the other institutions values than it can deduce from its own data set and the result r.
3. *No human or machine is allowed to have access to both the patient identifier, and data of that patient not previously known.*
4. Semi-honesty: Institutions perform agreed upon computations correctly using their true data. However, they are permitted to retain the results of intermediate computations.

Privacy preserving distributed computation

Represent the computation as a graph

- A directed graph where the nodes are sub-processes in the computation
- The arrows are messages between nodes
- Local data and processing at each node, including at the coordinator



Privacy preserving distributed computation

A note on the notation used in the following examples

$\{m\}$	A message containing *m*
$s\{m\}$	*m* encrypted with secret key *s*
$\{m\}_p$	*m* encrypted with public key *p*
$\{m\}^a$	*m* signed by *a*
$\{m\}_p^a$	*m* signed by *a and* encrypted with public key *p*
$\{a,p\}^c$	CA *c* binds public key *p* to identity (address) *a*
$A \rightarrow B : \{m\}$	Message $\{m\}$ sent from *A* to *B*

Implementation detail:

$\{m\}_p = \{\{s\}_p, s\{m\}\}$

Privacy preserving distributed computation

A real SMC example: Calculate Pearson's *r*

- Pearson's *r* is used to measure the correlation (linear dependence) between *n* samples of two variables *x* and *y*

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- In the case of *m* health institutions with s_j samples of x_{ji} and y_{ji} at each institution, *r* can be rewritten

$$r = \frac{\sum_{j=1}^{m}\sum_{i=1}^{s_j}(x_{j_i} - \bar{x})(y_{j_i} - \bar{y})}{\sqrt{\sum_{j=1}^{m}\sum_{i=1}^{s_j}(x_{j_i} - \bar{x})^2 \sum_{j=1}^{m}\sum_{i=1}^{s_j}(y_{j_i} - \bar{y})^2}}$$

Privacy preserving distributed computation

A real SMC example: Calculate Pearson's *r*

- Pearson's *r* is used to measure the correlation (linear dependence) between *n* samples of two variables *x* and *y*

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- In the case of *m* health institutions with s_j samples of x_{ji} and y_{ji} at each institution, *r* can be rewritten

$$r = \frac{\sum_{j=1}^{m} \overbrace{\sum_{i=1}^{s_j}(x_{j_i} - \bar{x})(y_{j_i} - \bar{y})}^{u_j}}{\sqrt{\sum_{j=1}^{m} \underbrace{\sum_{i=1}^{s_j}(x_{j_i} - \bar{x})^2}_{v_j} \sum_{j=1}^{m} \underbrace{\sum_{i=1}^{s_j}(y_{j_i} - \bar{y})^2}_{w_j}}}$$

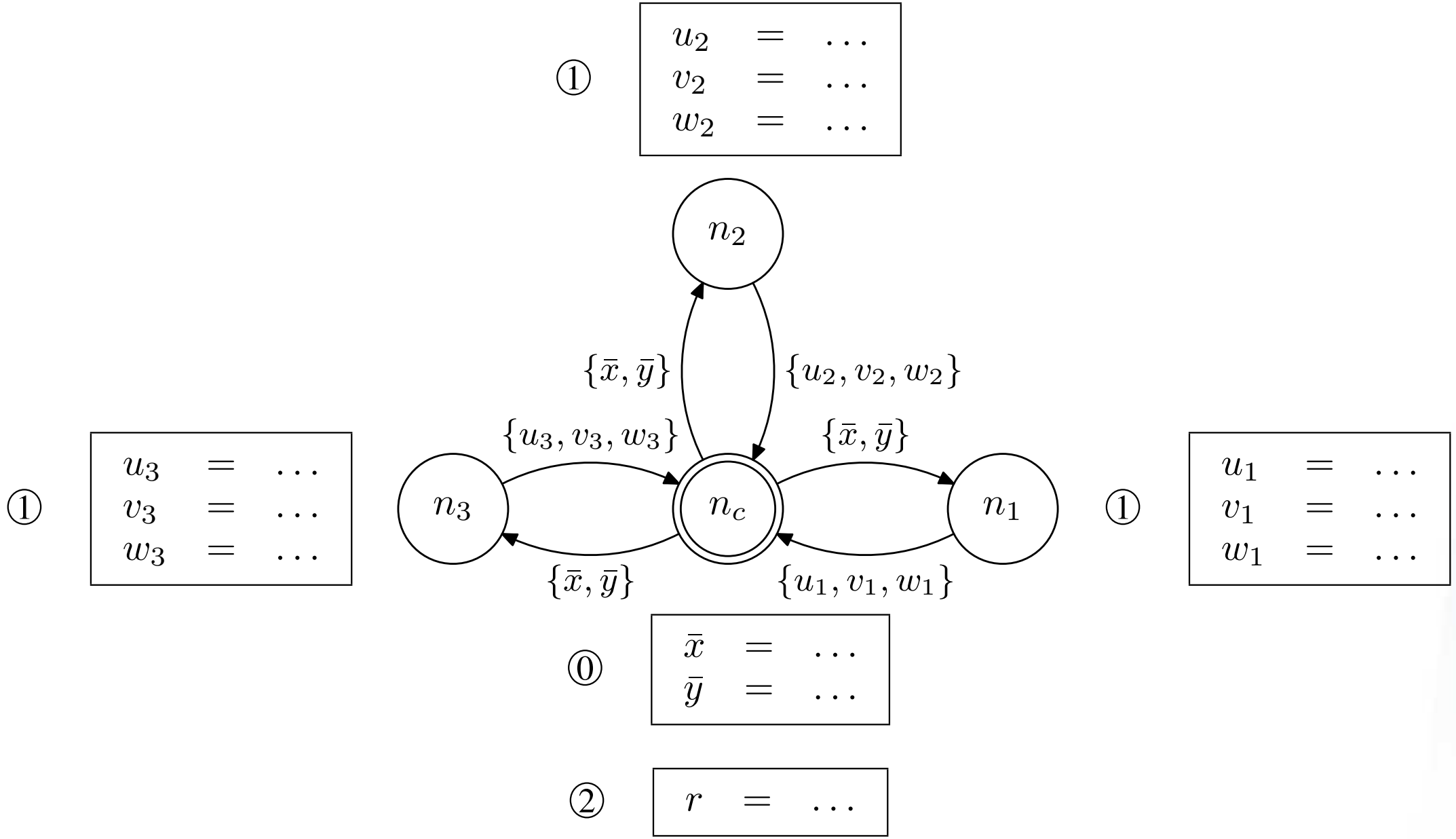
Privacy preserving distributed computation

A real SMC example: Calculate Pearson's *r*

- At each node *j* (health institution) the following three intermediate results have to be calculated:

① $$\begin{aligned} u_j &= \sum_{i=1}^{s_j} (x_{j_i} - \bar{x})(y_{j_i} - \bar{y}) \\ v_j &= \sum_{i=1}^{s_j} (x_{j_i} - \bar{x})^2 \\ w_j &= \sum_{i=1}^{s_j} (y_{j_i} - \bar{y})^2 \end{aligned}$$

- The initial mean values $\bar{x}$ and $\bar{y}$ can be securely calculated using an approach similar to the SMC example calculating the mean value


①
$u_2 = \ldots$
$v_2 = \ldots$
$w_2 = \ldots$
n_2
$\{\bar{x}, \bar{y}\}$
$\{u_2, v_2, w_2\}$
$\{u_3, v_3, w_3\}$
$\{\bar{x}, \bar{y}\}$
①
$u_3 = \ldots$
$v_3 = \ldots$
$w_3 = \ldots$
n_3
n_c
n_1
①
$u_1 = \ldots$
$v_1 = \ldots$
$w_1 = \ldots$
$\{\bar{x}, \bar{y}\}$
$\{u_1, v_1, w_1\}$
⓪
$\bar{x} = \ldots$
$\bar{y} = \ldots$
②
$r = \ldots$

Privacy preserving distributed computation

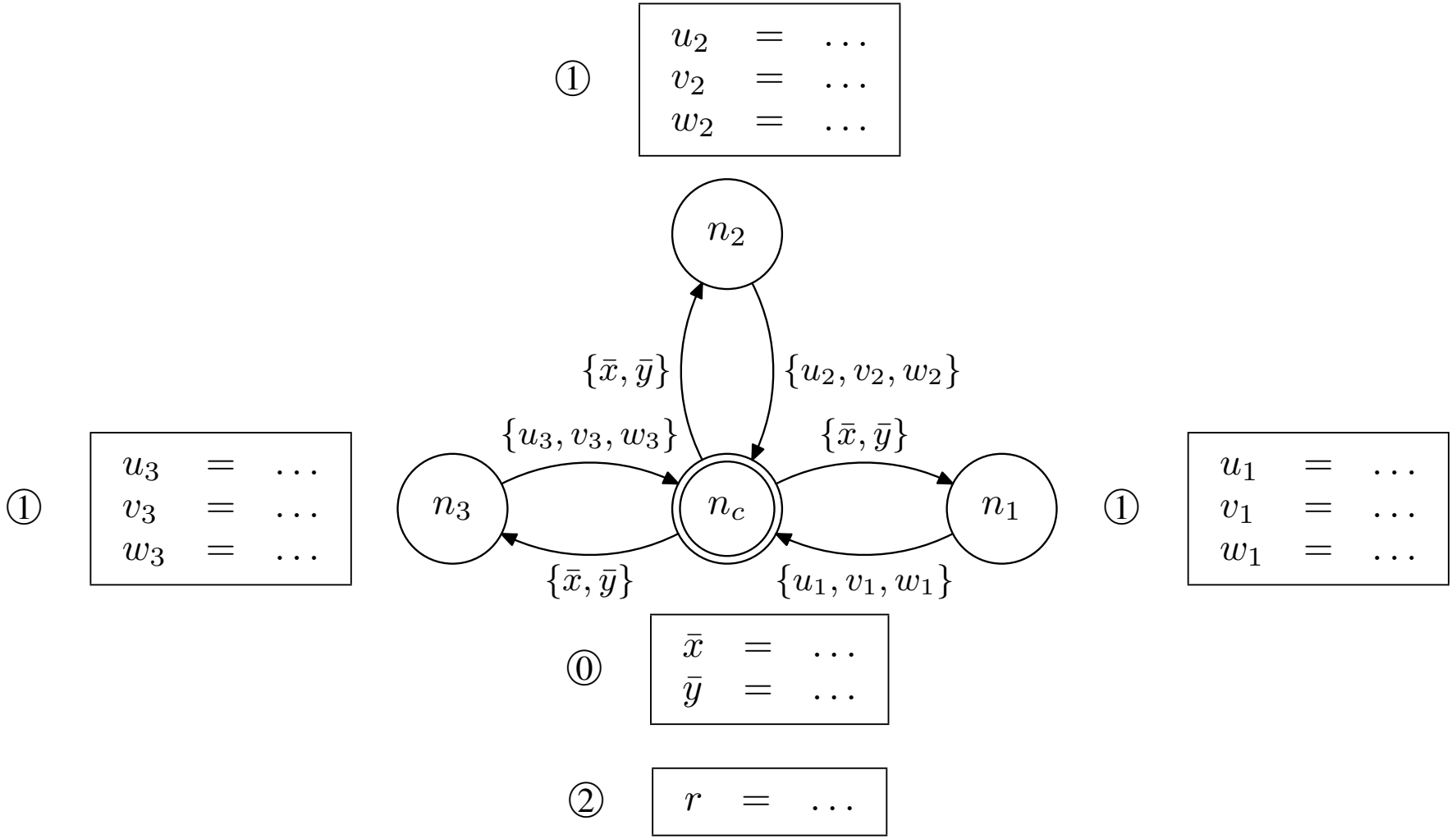
A real SMC example: Calculate Pearson's *r*

- When all intermediate results are received at the coordinator, Pearson's *r* can be calculated:

② $$r = \frac{\sum_{j=1}^{m} u_j}{\sqrt{\sum_{j=1}^{m} v_j \sum_{j=1}^{m} w_j}}$$

Privacy preserving distributed computation

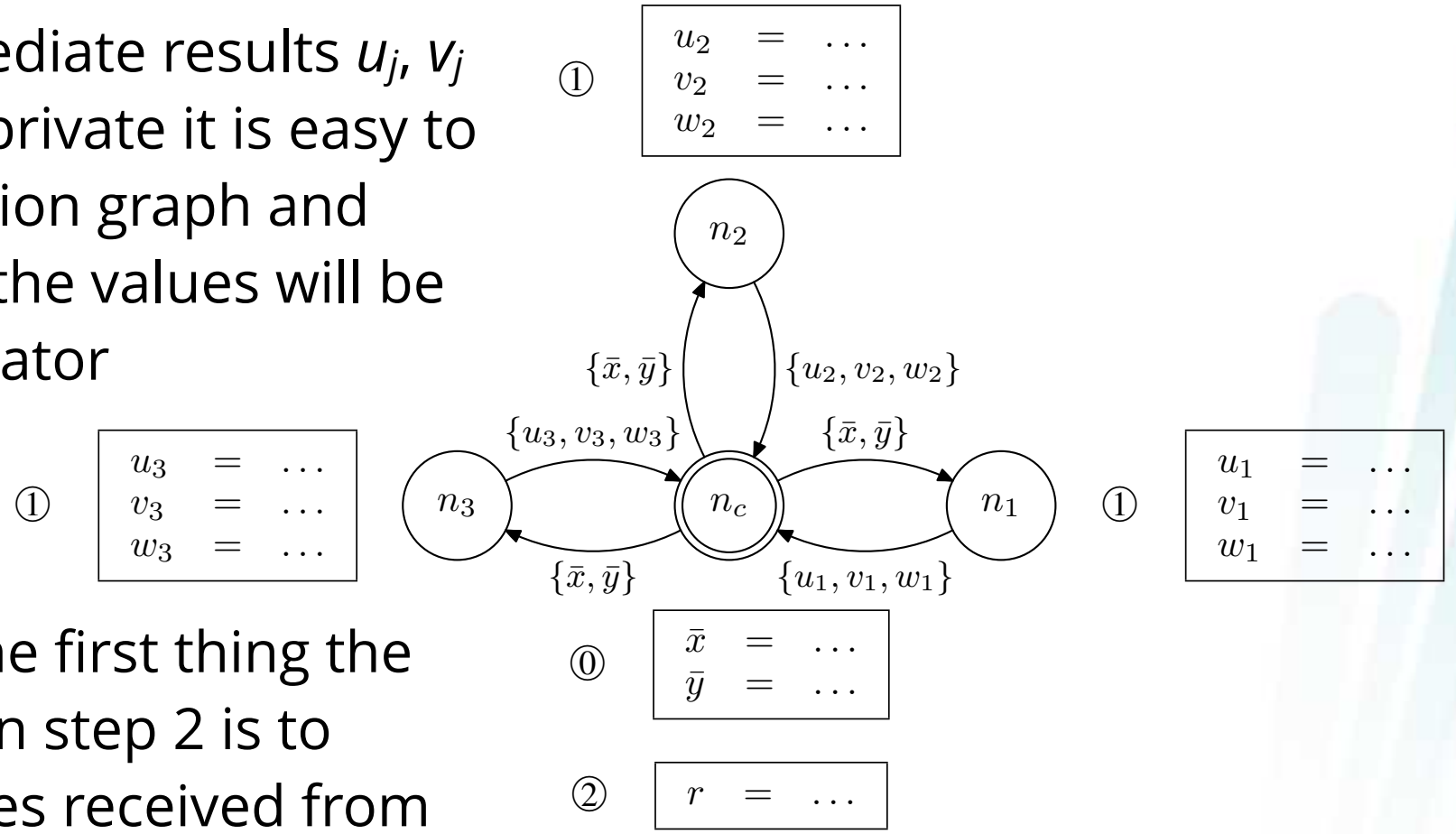
A real SMC example: Calculate Pearson's *r*



Privacy preserving distributed computation

A real SMC example: Calculate Pearson's *r*

- However, if the intermediate results u_j, v_j and w_j are considered private it is easy to see from the computation graph and processing step 2 that the values will be exposed at the coordinator



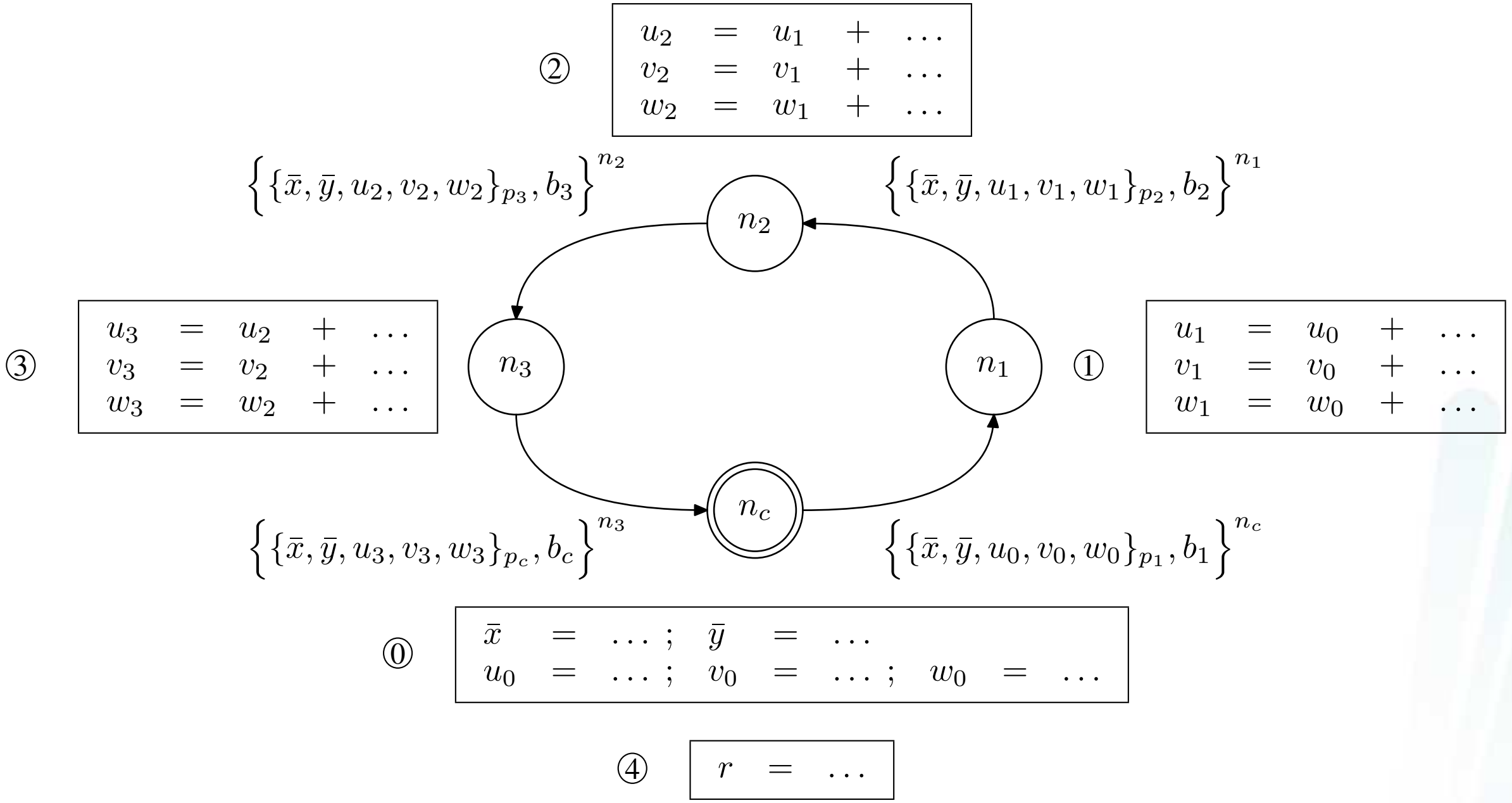
- This is solvable since the first thing the coordinator has to do in step 2 is to summarize all the values received from the nodes

Privacy preserving distributed computation

A real SMC example: Calculate Pearson's *r*

- Instead of sending these values to the coordinator directly, a similar approach to the one done when calculating the mean value in the first SMC example
- At each node *j* the following calculations are performed (where u_0, v_0, and w_0 are large random numbers

$$
\begin{aligned}
u_j &= u_{j-1} + \sum_{i=1}^{s_j} (x_{j_i} - \bar{x})(y_{j_i} - \bar{y}) \\
v_j &= v_{j-1} + \sum_{i=1}^{s_j} (x_{j_i} - \bar{x})^2 \\
w_j &= w_{j-1} + \sum_{i=1}^{s_j} (y_{j_i} - \bar{y})^2
\end{aligned}
$$


②
$u_2 = u_1 + \ldots$
$v_2 = v_1 + \ldots$
$w_2 = w_1 + \ldots$
$\left\{\{\bar{x}, \bar{y}, u_2, v_2, w_2\}_{p_3}, b_3\right\}^{n_2}$
$\left\{\{\bar{x}, \bar{y}, u_1, v_1, w_1\}_{p_2}, b_2\right\}^{n_1}$
n_2
③
$u_3 = u_2 + \ldots$
$v_3 = v_2 + \ldots$
$w_3 = w_2 + \ldots$
n_3
n_1
①
$u_1 = u_0 + \ldots$
$v_1 = v_0 + \ldots$
$w_1 = w_0 + \ldots$
n_c
$\left\{\{\bar{x}, \bar{y}, u_3, v_3, w_3\}_{p_c}, b_c\right\}^{n_3}$
$\left\{\{\bar{x}, \bar{y}, u_0, v_0, w_0\}_{p_1}, b_1\right\}^{n_c}$
⓪
$\bar{x} = \ldots ; \quad \bar{y} = \ldots$
$u_0 = \ldots ; \quad v_0 = \ldots ; \quad w_0 = \ldots$
④
$r = \ldots$

Privacy preserving distributed computation

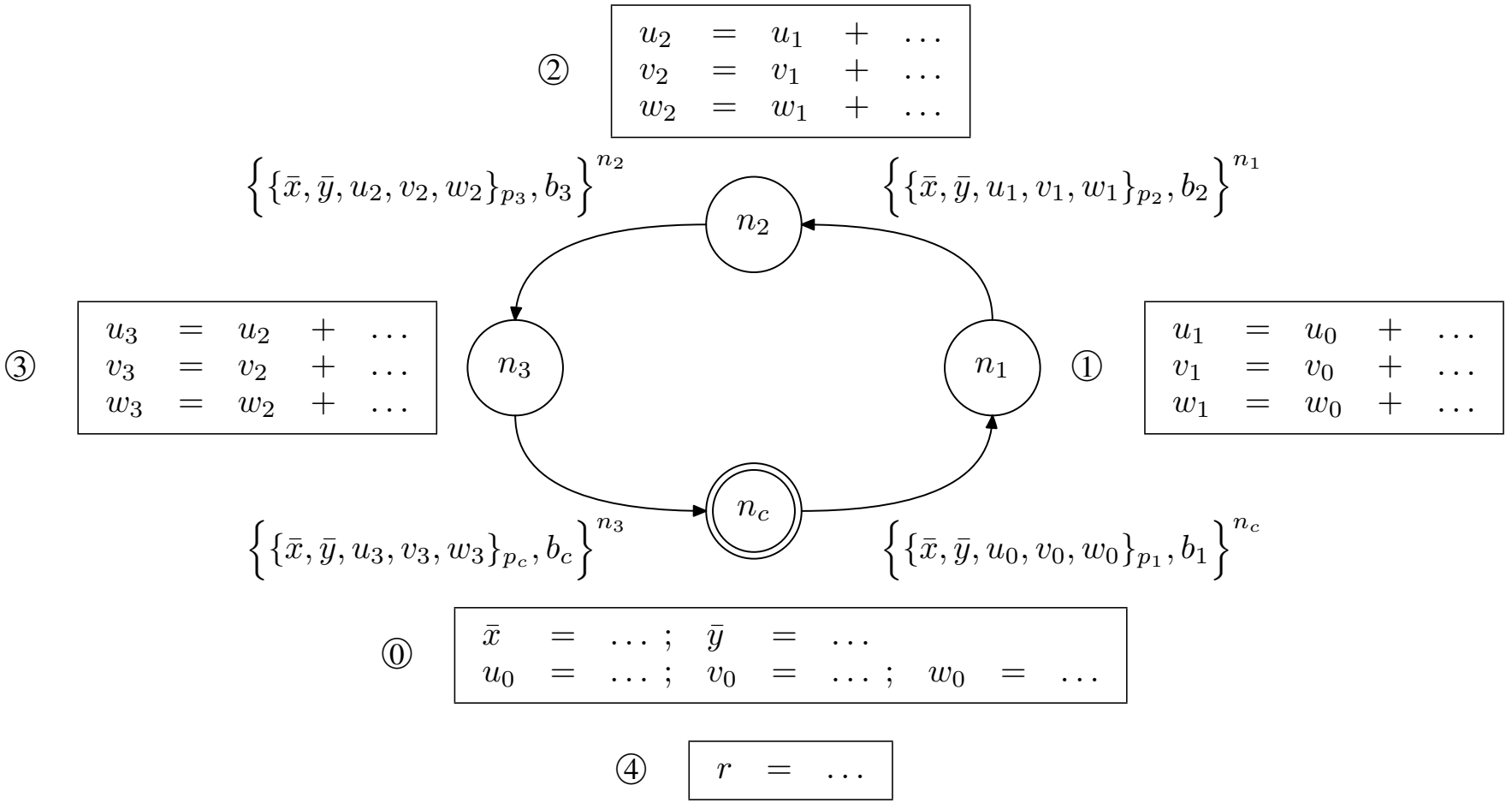
A real SMC example: Calculate Pearson's *r*

- The coordinator calculates Pearson's *r*:

④ $$r = \frac{u_3 - u_0}{\sqrt{(v_3 - v_0)(w_3 - w_0)}}$$

Privacy preserving distributed computation

A real SMC example: Calculate Pearson's *r*



Secure multiparty computation
Some publications

- S. Goldwasser, “Multi party computations: past and present”, in PODC’97, Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing. New York: ACM, 1997, pp. 1–6.
- A. F. Karr, “Secure statistical analysis of distributed databases, empha- sizing what we don’t know”, Journal of Privacy and Confidentiality, vol. 1, no. 2, pp. 197–211, 2009.
- W. Du and Z. Zhan, “A practical approach to solve secure multi-party computation problems”, in Proceedings of the 2002 workshop on New security paradigms. New York: ACM, 2002, pp. 127–135.

Secure multiparty computation

- The security comes from the algorithm/protocol
- Often combined with crypto-concepts (encryption, public-key systems, digital signatures, and so on)
- We have implemented SNOOP for practical SMC
  - Supports the computation graph, the messages, local processing, guards and rules, mechanisms to ensure progress, failure models, and much more

SNOOP

Anders Andersen, Merete Saus. *Privacy preserving distributed computation of community health research data*. In Elhadi Shakshuki, editor, The 4th International Workshop on Privacy and Security in Healthcare (PSCare 2017), volume 113 of Procedia Computer Science, page 633–640. Elsevier, 2017. ISSN 1877-0509.

Anders Andersen, *SNOOP: Privacy preserving middleware for secure multi-party computations*, Proceedings of the 13th Workshop on Adaptive and Reflective Middleware (ARM 2014), 15th International Middleware Conference (Middleware '14), Bordeaux, France, ACM 2014.

Anders Andersen, Kassaye Yitbarek Yigzaw, Randi Karlsen, *Privacy preserving health data processing*, IEEE Healthcom 2014, IEEE Communications Society 2014.

Anders Andersen, *An implementation of secure multi-party computations to preserve privacy when processing EMR data*, 2013 Eleventh Annual Conference on Privacy, Security and Trust, IEEE conference proceedings 2013.

https://uit.no/ifi

Thank you for your attention