PORTIA *Privacy, Obligations, and Rights in Technologies of Information Assessment*

Accountability in Privacy-Preserving Data Mining

Rebecca Wright
Computer Science Department
Stevens Institute of Technology
www.cs.stevens.edu/~rwright

TAMI/PORTIA Workshop on Privacy and Accounability
29 June, 2006

STEVENS Institute of Technology

Privacy-Preserving Data Mining

Allow multiple data holders to collaborate to compute important information while protecting the privacy of other information.

- Security-related information
- Public health information
- Marketing information
- etc.

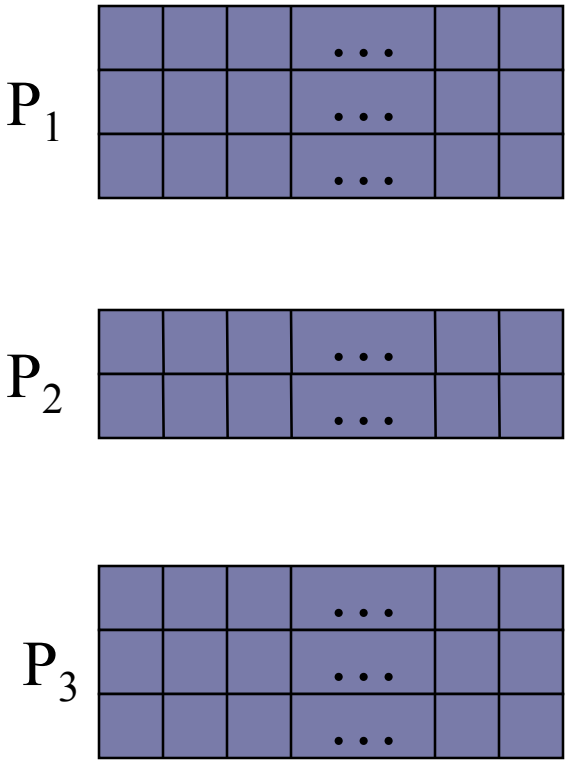
Technological tools include cryptography, data perturbation and sanitization, access control, inference control, trusted platforms.

Advantages of Privacy Protection

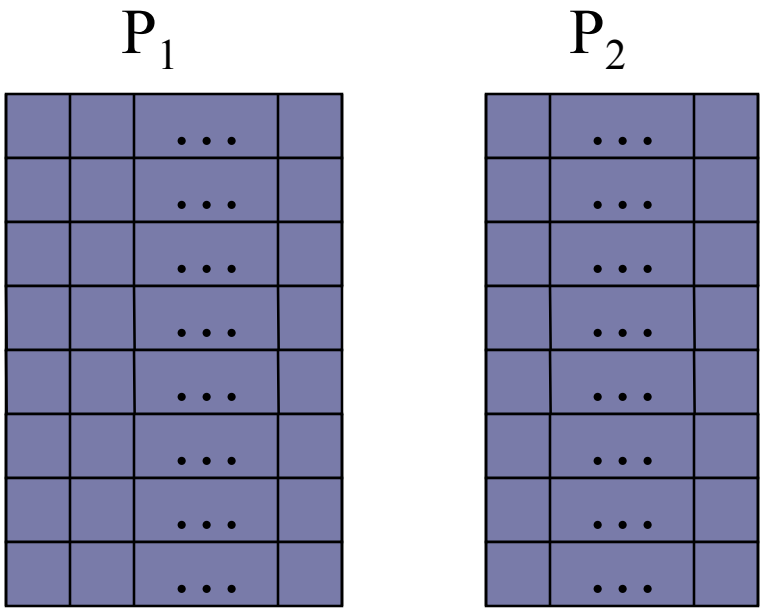
- protection of personal information
- protection of proprietary or sensitive information
- enables collaboration between different data owners (since they may be more willing or able to collaborate if they need not reveal their information)
- compliance with legislative policies (e.g., HIPAA, EU privacy directives)

Models for Distributed Data Mining, I

- Horizontally Partitioned

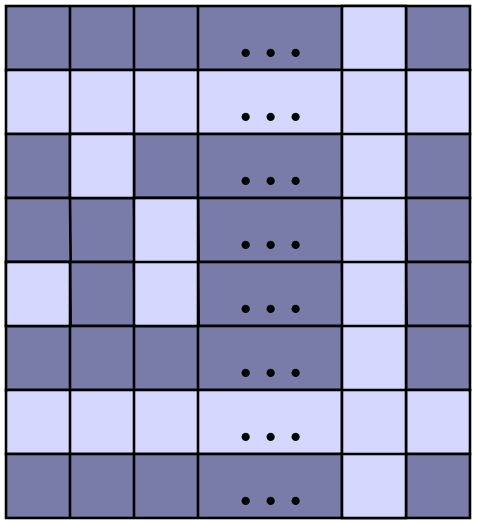


- Vertically Partitioned



Models for Distributed Data Mining, II

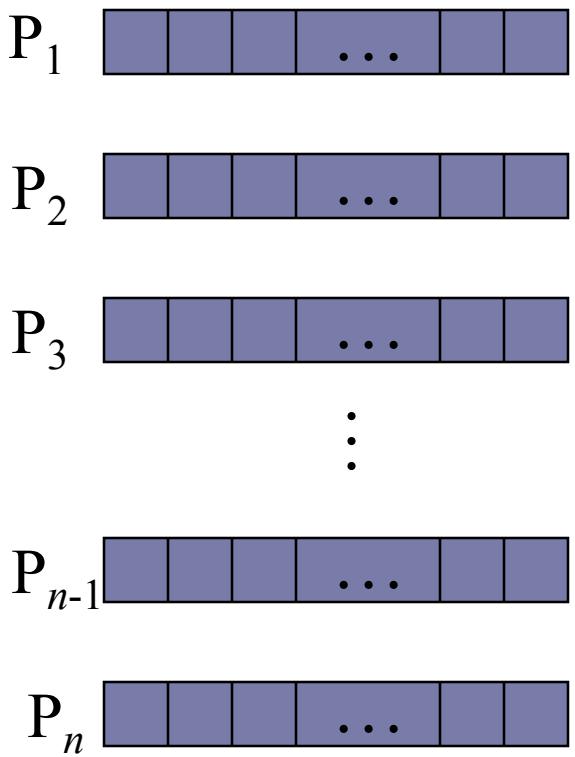
- Arbitrarily partitioned



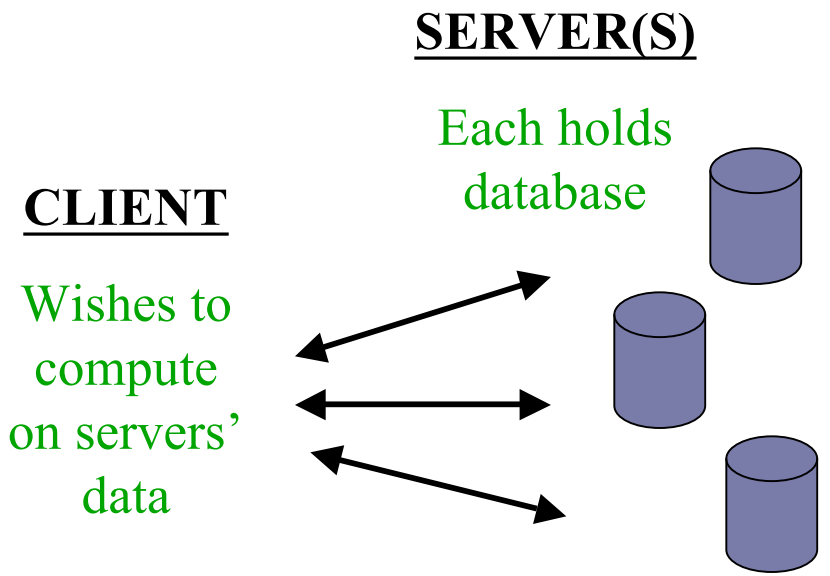
P_1 P_2

Models for Distributed Data Mining, III

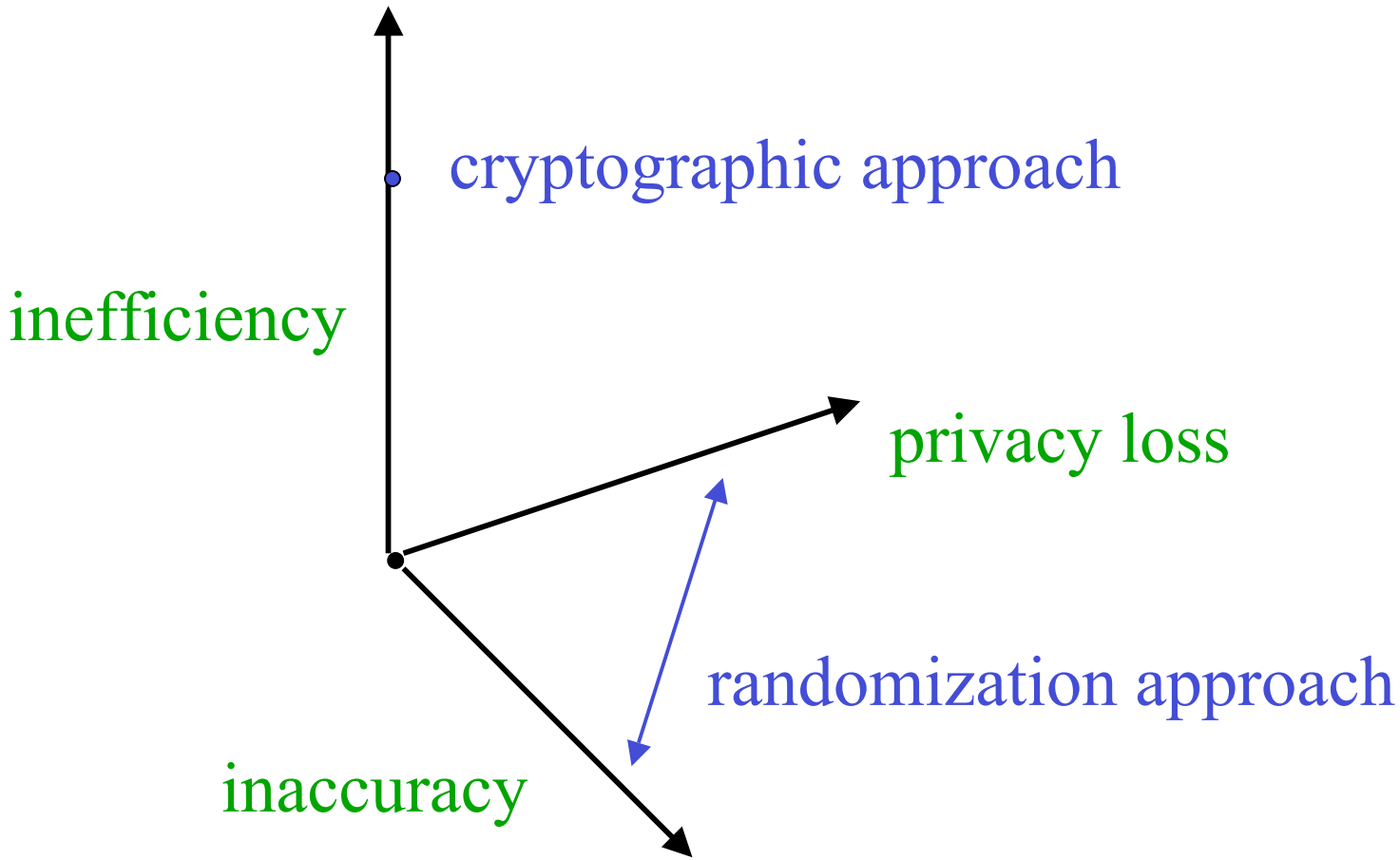
- Fully Distributed



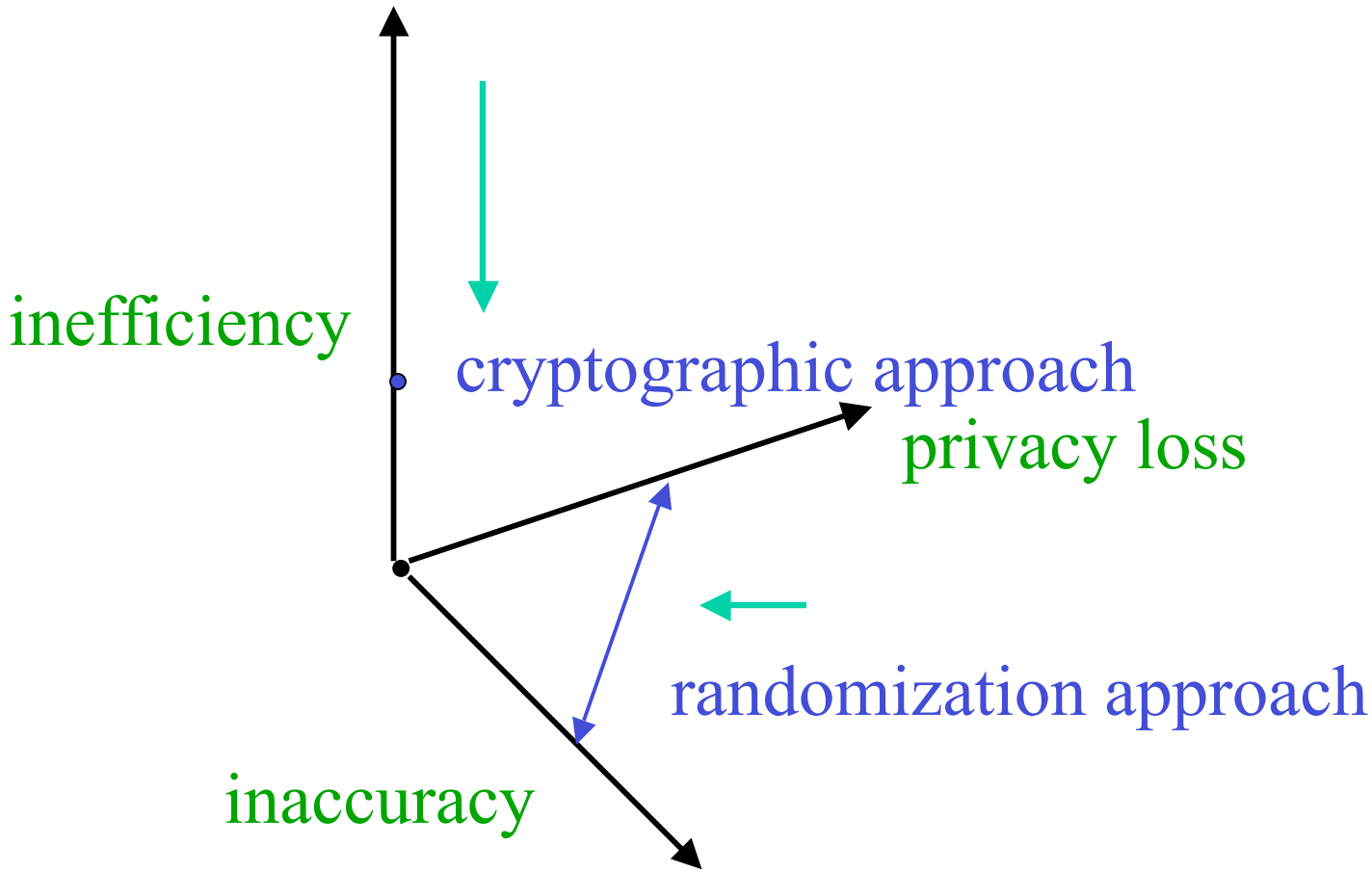
- Client/Server(s)



Cryptography vs. Randomization

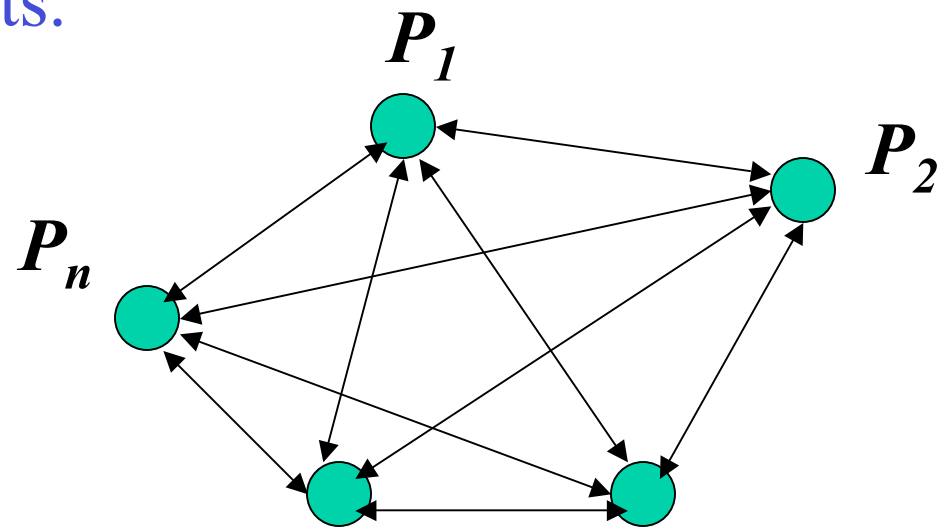


Cryptography vs. Randomization



Secure Multiparty Computation

- Allows *n* players to privately compute a function *f* of their inputs.



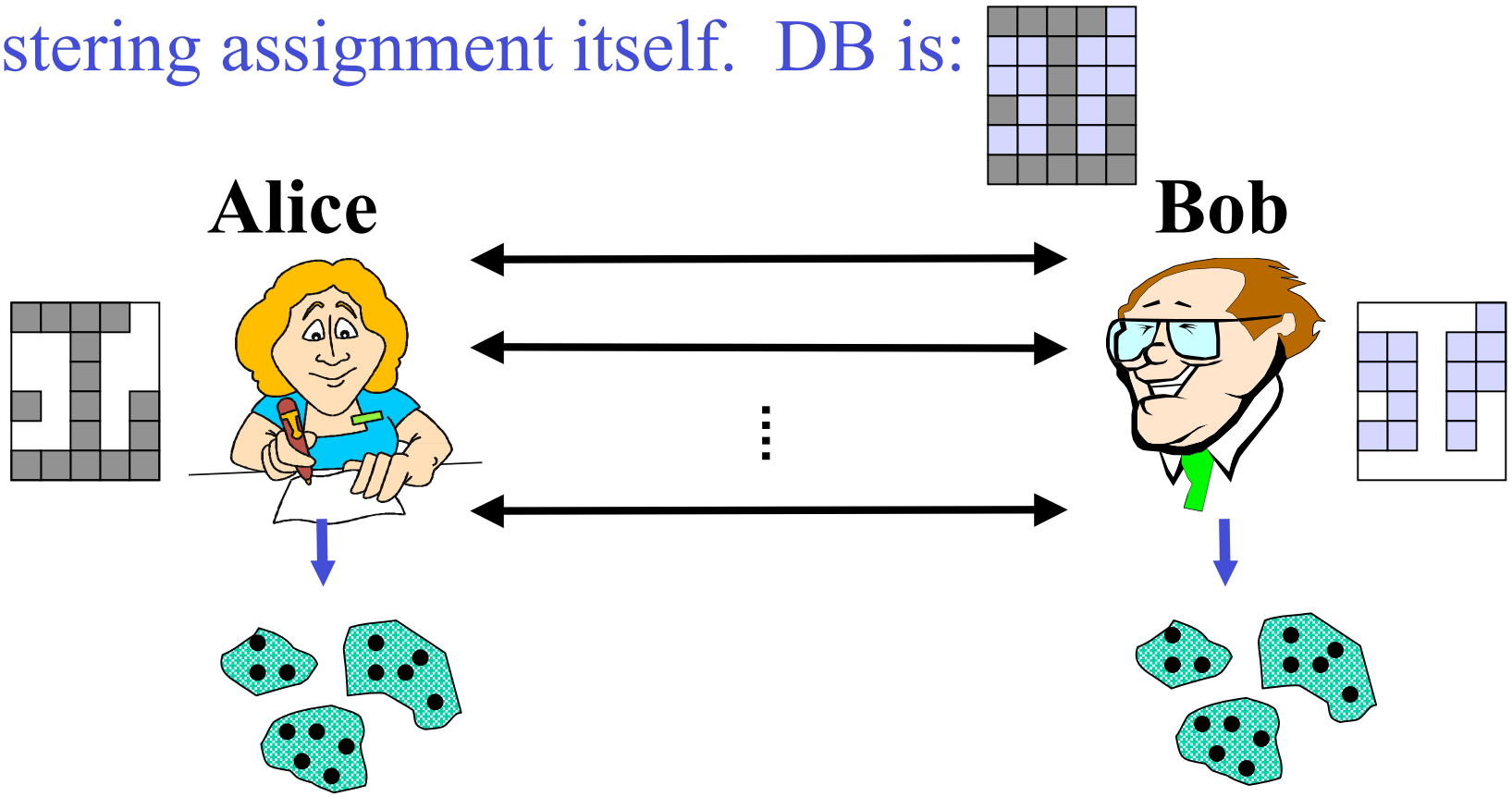
- Overhead is polynomial in size of inputs and complexity of *f* [Yao86, GMW87, BGW88, CCD88, ...]
- In theory, can solve any private distributed data mining problem. In practice, not efficient for large data.

Our PPDM Work

- [WY04,YW05]: privacy-preserving construction of Bayesian networks from vertically partitioned data.
- [YZW05]: privacy-preserving frequency mining in the fully distributed model (enables naïve Bayes classification, decision trees, and association rule mining).
- [JW05, JPW06]: privacy-preserving clustering: *k*-means clustering for arbitrarily partitioned data and a divide-and-merge clustering algorithm for horizontally partitioned data.
- [ZYW05]: privacy-preserving solutions for a data publisher to learn a *k*-anonymized version of a fully distributed database.

Privacy-Preserving Clustering [JW05]

Goal: Cooperatively learn *k*-means clustering on database arbitrarily partitioned between Alice and Bob, ideally without either party learning anything except the clustering assignment itself. DB is:



k-means Clustering [Llo82]

Input: Database D, integer k.
Output: Assignment of database objects to k clusters.

- Randomly select k objects from D as initial cluster centers.
- Iteratively try to improve clusters:
  - For each object d_i, determine the closest cluster center and assign d_i to that cluster.
  - Recompute the new cluster centers.

until the change is sufficiently small.

Privacy-Preserving Clustering

Input: Database D, integer k.
Output: Assignment of database objects to k clusters.

- Randomly select k objects from D as initial cluster centers. Alice and Bob share these centers.
- Iteratively try to improve clusters:
  - For each object d_i, determine the closest cluster center and assign d_i to that cluster.
  - Recompute shares of the new cluster centers.

until the change is sufficiently small.

Computing Closest Cluster

- For an object *d*, compute distance to each shared cluster center:
  - Alice owns some attributes and Bob owns some attributes.
  - Distance can be written as a quadratic function of these attributes and Alice and Bob's shares of the cluster center.
  - Can be computed as shares using local computation and secure scalar products.
- Use Yao's secure 2-party computation on the *k* shared distances to determine which is minimum.

Overall Performance

k number of clusters c bits for encrypted attribute	m number of attributes s number of iterations

- **Computation:** O($kmns$) encryptions and multiplications for each party.
- **Communication:** O($ckmns$) bits.

Privacy-Preserving Clustering Summary

- This solution works for arbitrarily partitioned data.
- It leaks assignment to candidate cluster centers (though not the candidate cluster centers themselves) at each iteration, but nothing else about data.
- A straightforward modification not to leak cluster centers would be inefficient.
- For horizontally partitioned data, we also have an alternate efficient no-leakage solution based on a new divide-and-merge clustering algorithm [JPW06].

Using PPDM

- To actually use privacy-preserving data mining, this kind of PPDM is not sufficient. Also needed:
  - Policies and enforcement for what queries should and shouldn't be allowed. (And methods/tools for helping to choose such policies and understanding the implications).
  - Methods for data-preprocessing, including data cleaning, error handling, adherence to standards for how to represent the data.
  - Integration of many PPDM solutions into a common framework to provide sufficient usability and utility to users.
  - For many applications, ability to prove that policies were met, ability to selectively obtain more information in appropriate cases, audit logs (with their own sets of policies and enforcement issues), etc.

Beyond Privacy-Preserving Data Mining

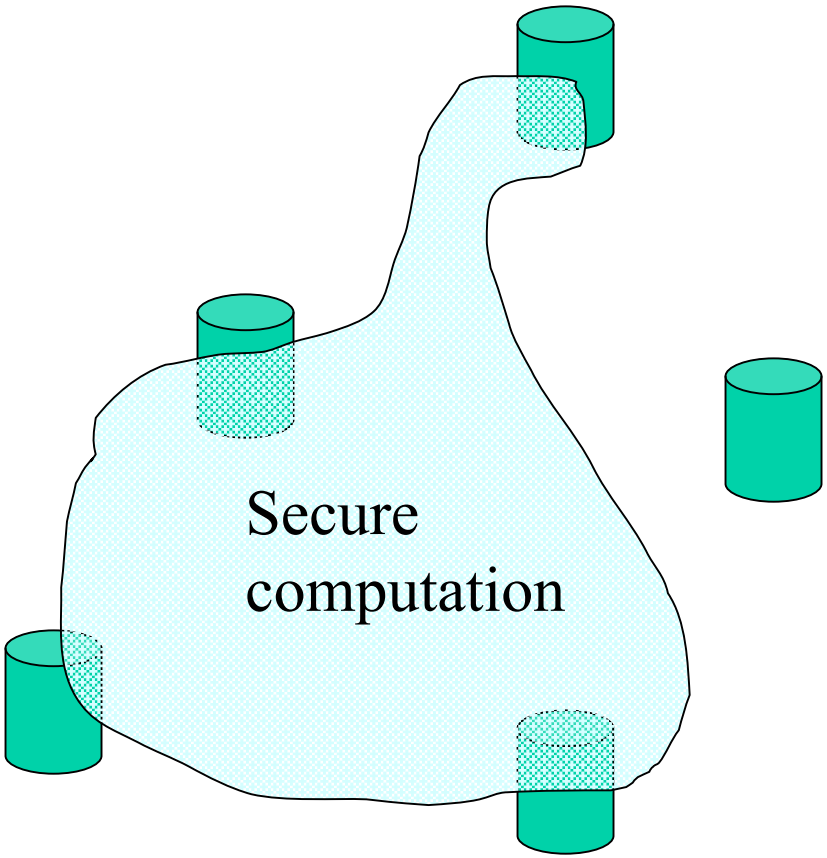
Enforce policies about what kind of queries or computations on data are allowed.

- [JW06]: In the client/server model, allows a server to ensure that inference control policies on aggregate queries are satisfied, without learning which queries the user makes. Extends private inference control work of [WS04].
- [KMN05]: Simulatable auditing to ensure that query denials do not leak information.

Accountability in PPDM

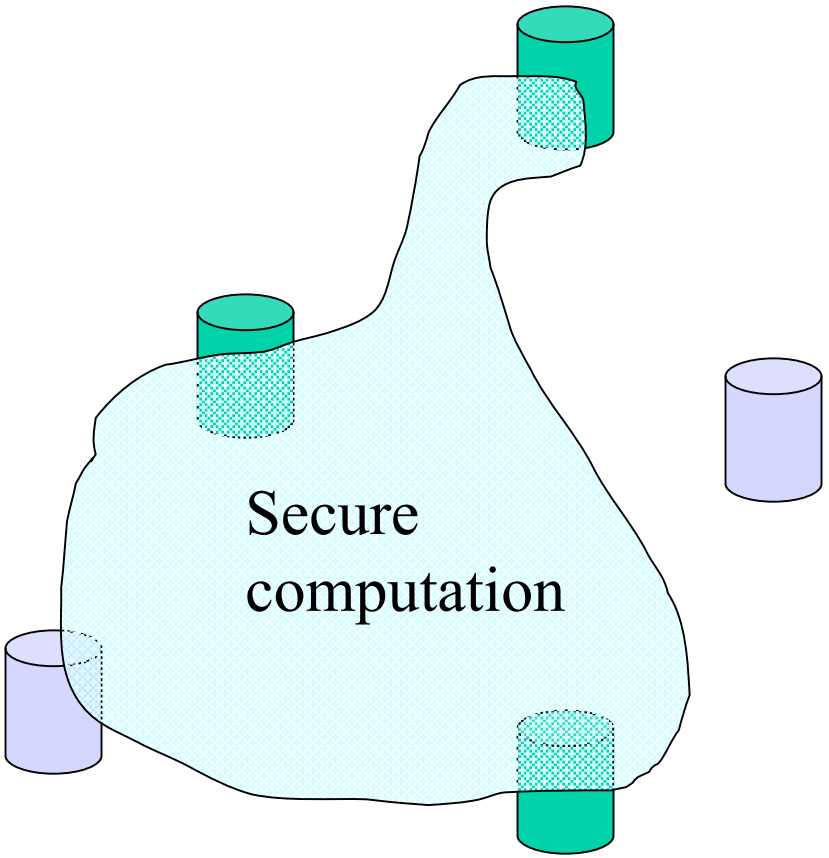
- Is it possible to further "extend the boundary" in privacy-preserving data mining to provide accountability and transparency?
  - May sound impossible at first thought, but then so does the idea of computing on joint data without sharing it.
- What would be needed to make this happen?
  - Combining some existing PPDM solutions with additional existing cryptographic tools (such as zero knowledge proofs, anonymous credentials [CL], mathematically enforced policies [JS])
  - Perhaps some new cryptographic tools and/or lighter-weight mechanisms
  - Policy languages, enforcement, and reconciliation

Potential Solution Framework



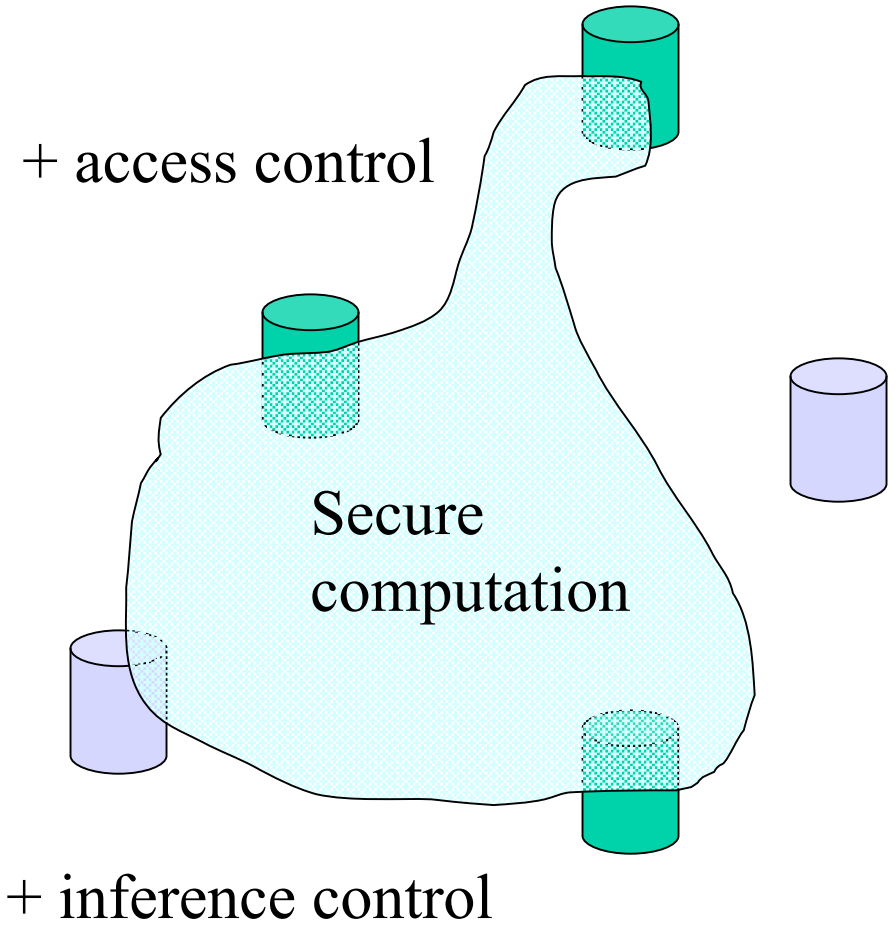
- Secure computation to protect critical data

Potential Solution Framework



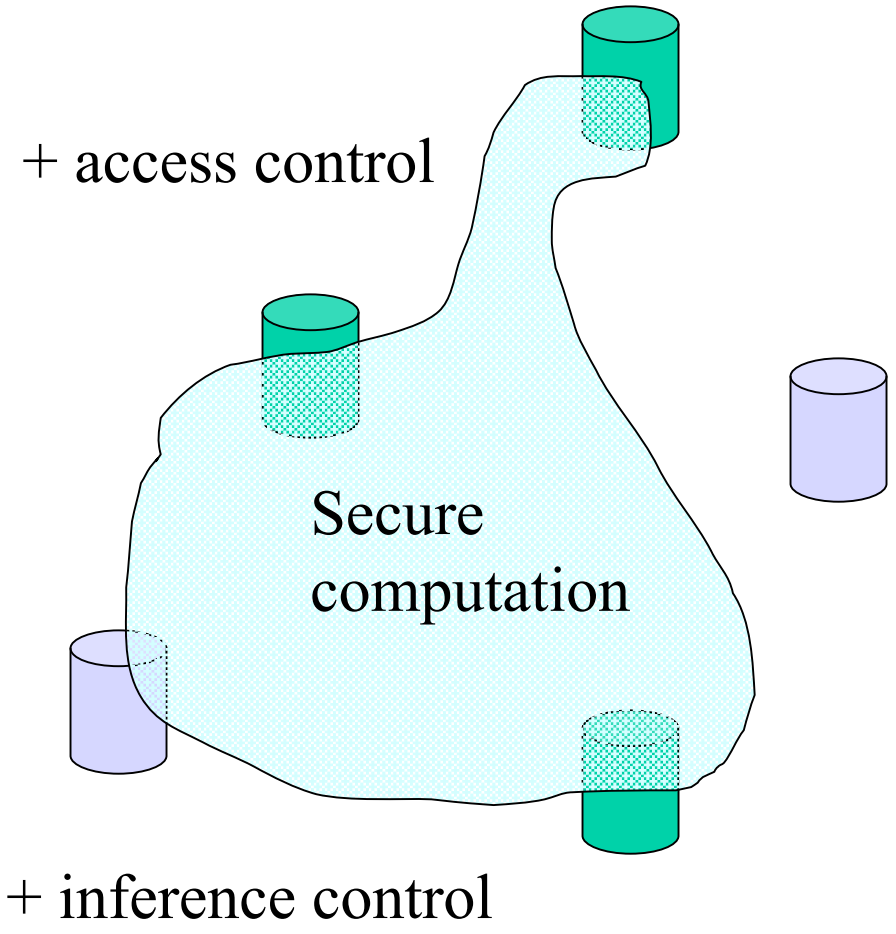
- Secure computation to protect critical data
- Perturbation or aggregation to protect possibly sensitive data
- No protection on completely innocuous data

Potential Solution Framework



- Secure computation to protect critical data
- Perturbation or aggregation to protect possibly sensitive data
- No protection on completely innocuous data
- With policies, access control and inference control to prevent additional leakage

Potential Solution Framework



Issues:

- How to determine which information is critical, possibly sensitive, innocuous?
- How to define appropriate policies?
- How to handle conflicting goals and desires?
- Scalability? Complexity?

Questions for Thought

- Can such an approach work realistically in any reasonable setting, or is it necessarily too rigid and/or too costly?
- Even if solutions are mathematically justified, how can they be handled legally and socially to provide the relevant entities (e.g., data subjects, users, general public) with confidence that accountability is actually provided?
- Can the TAMI architecture be combined with "advanced" cryptography in order to further enhance its privacy and accountability (e.g., cryptographic receipts that can only be obtained if policies are followed)?