Privacy Preserving Data Mining

Presented by Zheng Ma

Outline

- Motivation
- Randomization Approach
  - R. Agrawal and R. Srikant, “Privacy Preserving Data Mining”, SIGMOD 2000.
  - Application: Web Demographics
- Cryptographic Approach
  - Application: Inter-Enterprise Data Mining
- Challenges
  - Application: Privacy-Sensitive Security Profiling

Growing Privacy Concerns

- Popular Press:
  - Economist: The End of Privacy (May 99)
  - Time: The Death of Privacy (Aug 97)
- Govt. directives/commissions:
  - European directive on privacy protection (Oct 98)
  - Canadian Personal Information Protection Act (Jan 2001)
- Special issue on internet privacy, CACM, Feb 99
- S. Garfinkel, "Database Nation: The Death of Privacy in 21st Century", O' Reilly, Jan 2000

Privacy Concerns?

- Surveys of web users
  - 17% privacy fundamentalists, 56% pragmatic majority, 27% marginally concerned (Understanding net users' attitude about online privacy, April 99)
  - 82% said having privacy policy would matter (Freebies & Privacy: What net users think, July 99)
- Fear:
  - "Join" (record overlay) was the original sin.
  - Data mining: new, powerful adversary?
  - How much fear do you have?

Black box

- The primary task in data mining: development of models about aggregated data.
- Can we develop accurate models without access to *precise information* in individual data records?

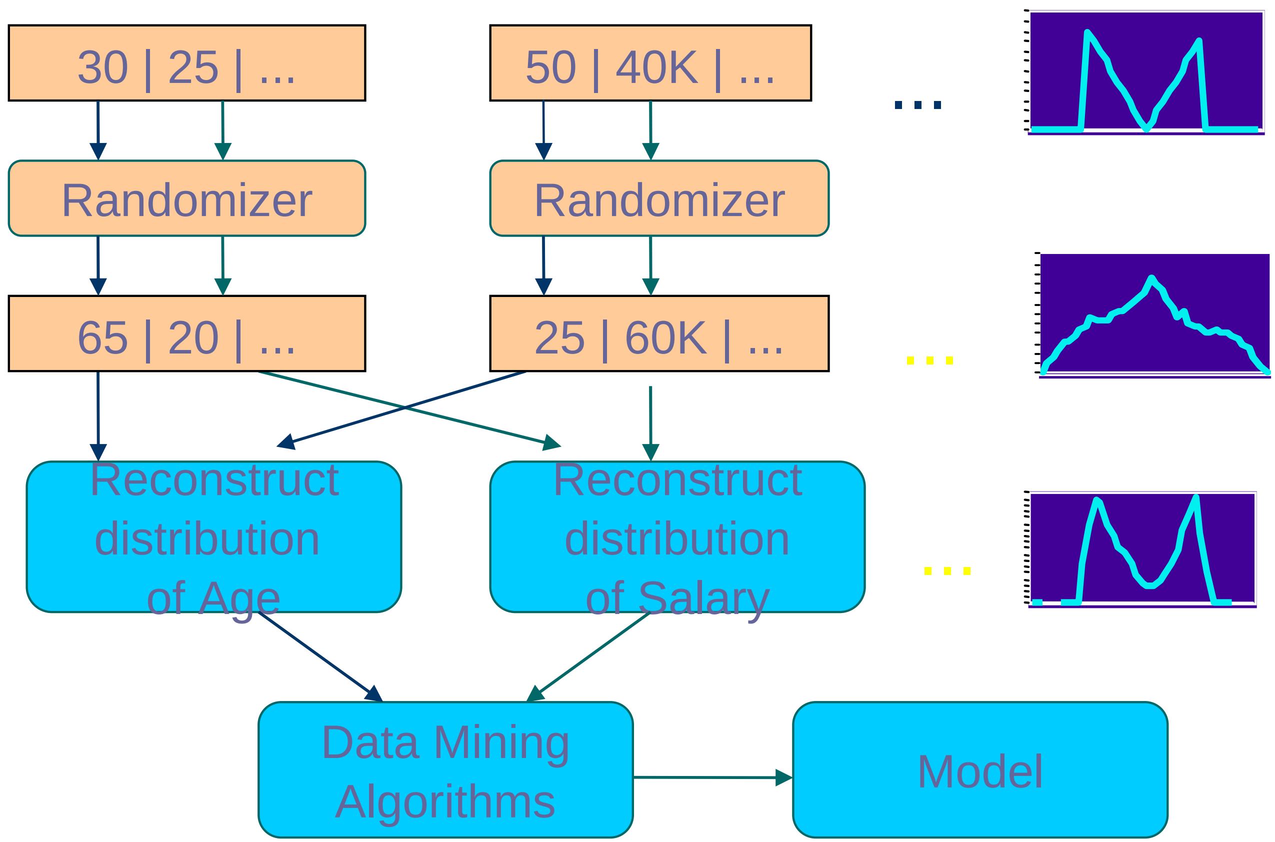
Outline

- Motivation
- Randomization Approach
  - Application: Web Demographics
  - R. Agrawal and R. Srikant, “Privacy Preserving Data Mining”, SIGMOD 2000.
- Cryptographic Approach
  - Application: Inter-Enterprise Data Mining
- Challenges
  - Application: Privacy-Sensitive Security Profiling

Web Demographics (example)

- Volvo S40 website targets people in 20s
  - Are visitors in their 20s or 40s?
  - Which demographic groups like/dislike the website?

Randomization Approach Overview



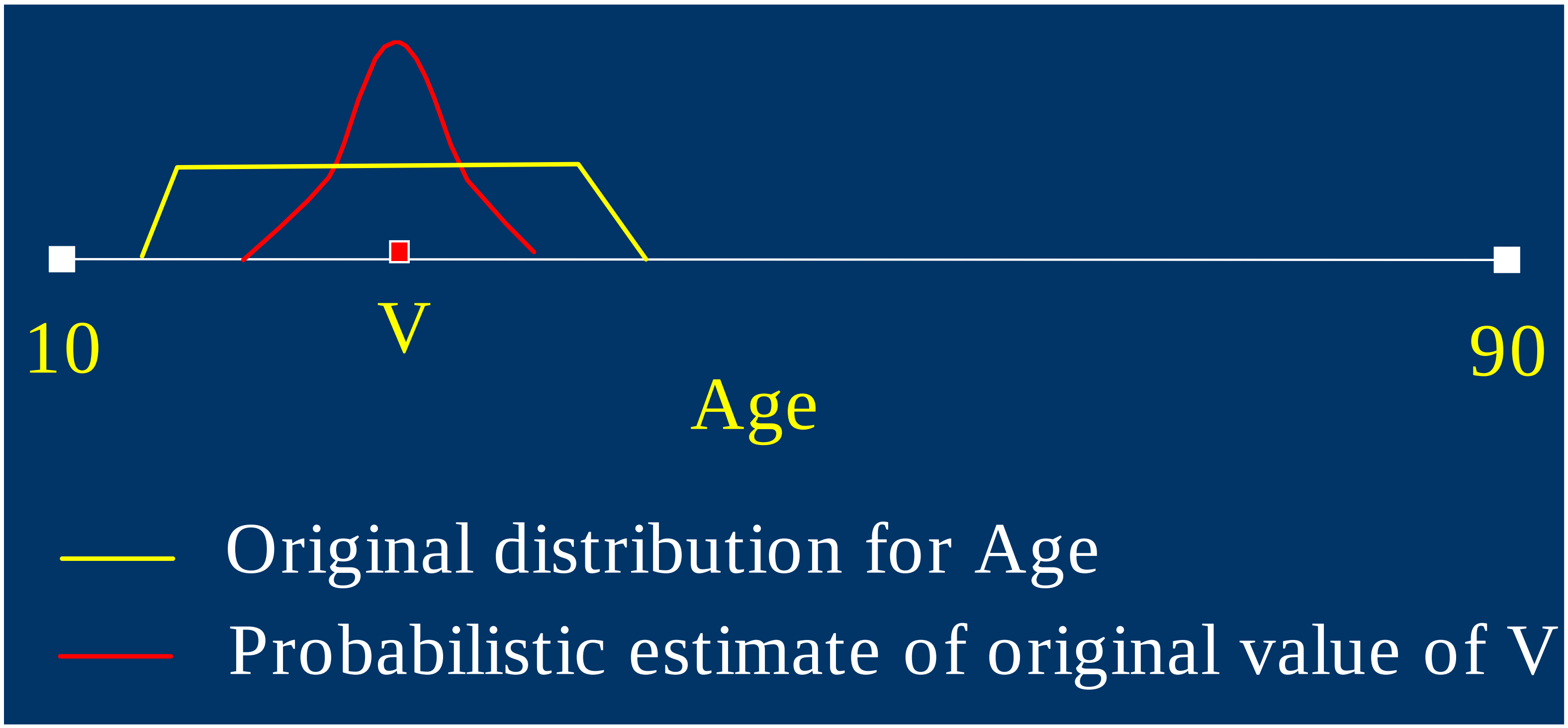
Reconstruction Problem

- Original values $x_1, x_2, \ldots, x_n$
  - from probability distribution X (unknown)
- To hide these values, we use $y_1, y_2, \ldots, y_n$
  - from probability distribution Y (known)
- Given
  - $x_1+y_1, x_2+y_2, \ldots, x_n+y_n$
  - the probability distribution of Y

  Estimate the probability distribution of X.

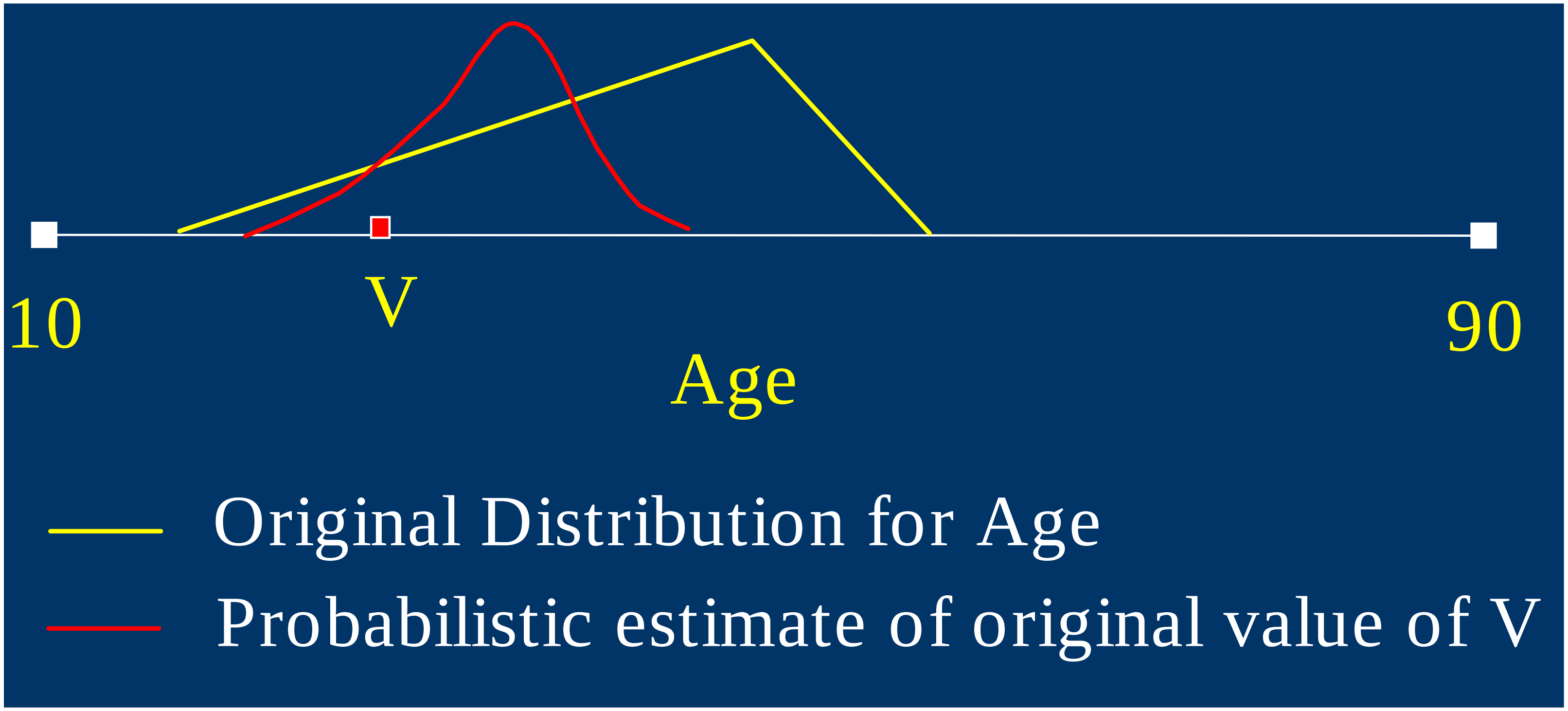
Intuition (Reconstruct single point)

- Use Bayes' rule for density functions



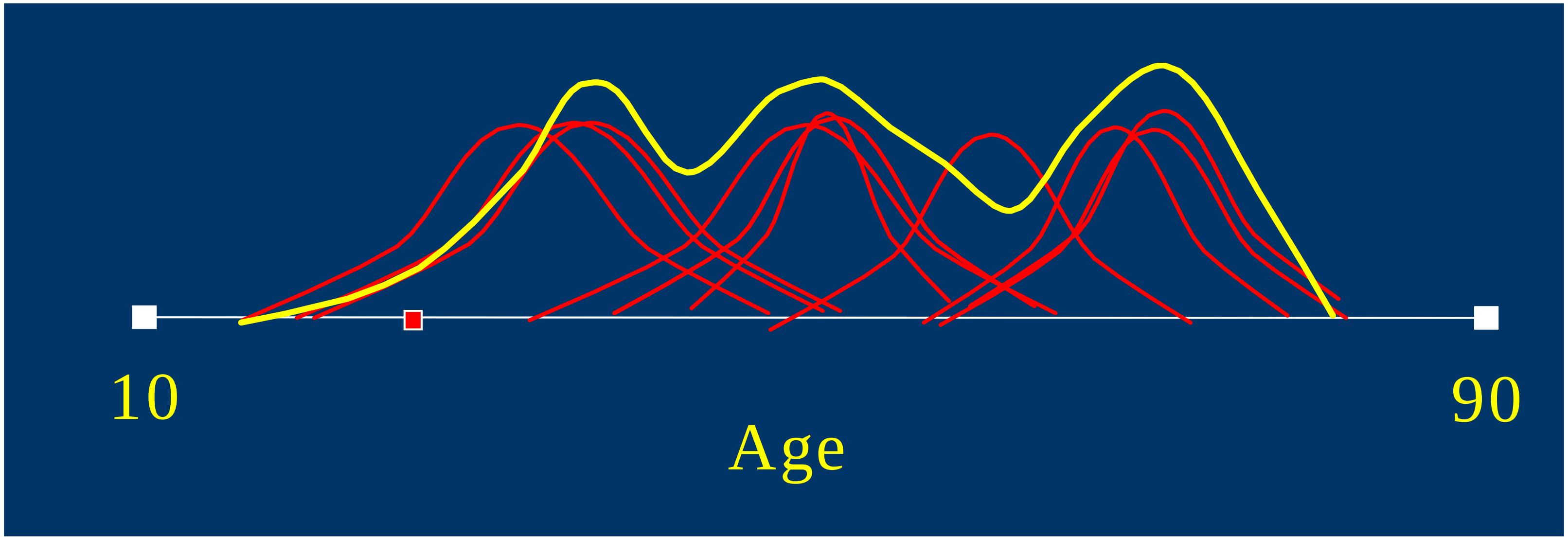
Intuition (Reconstruct single point)

- Use Bayes' rule for density functions



Reconstructing the Distribution

- Combine estimates of where point came from for all the points:
  - Gives estimate of original distribution.



Reconstruction: Bootstrapping

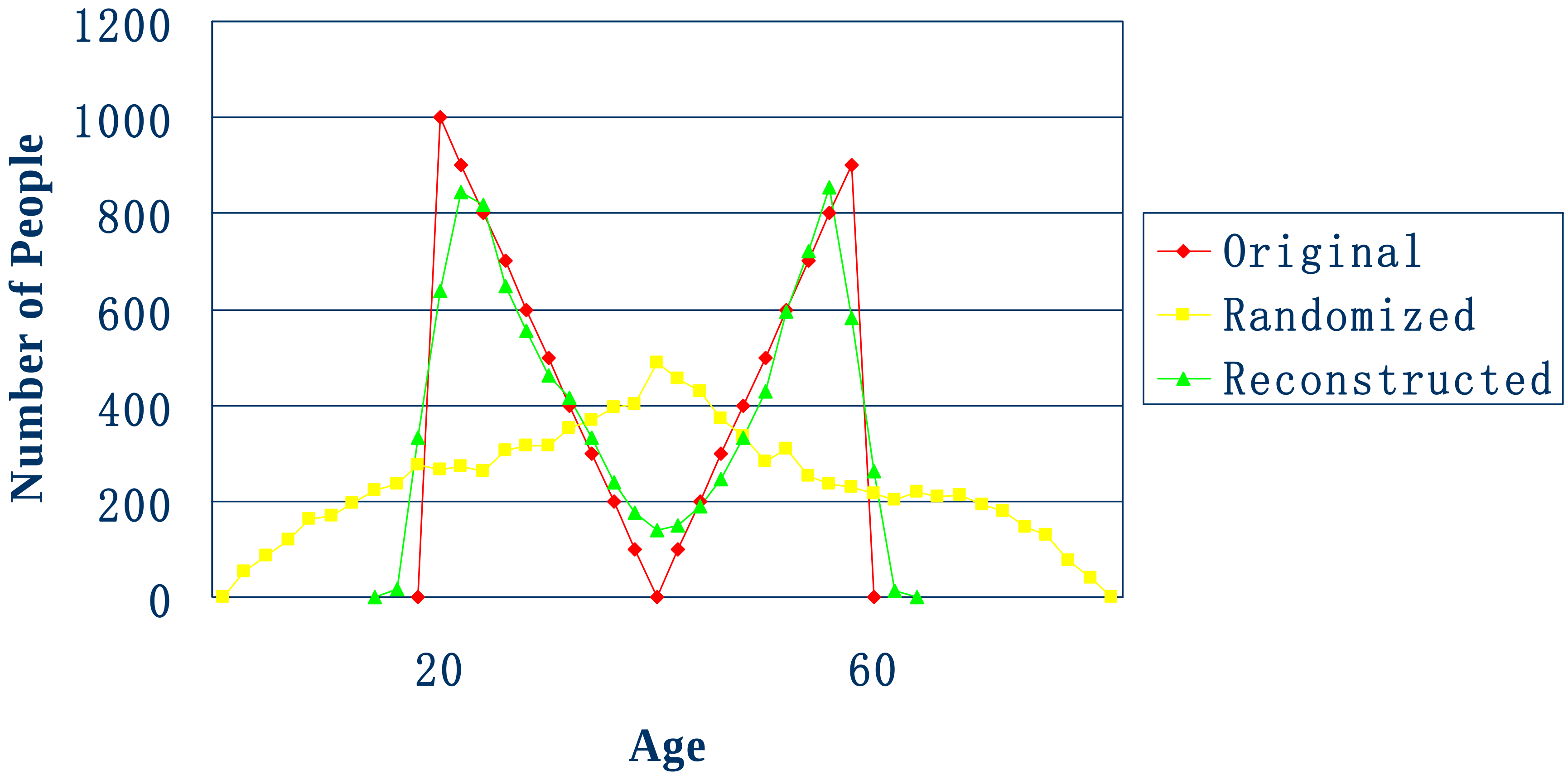
f_X^0 := Uniform distribution
j := 0 // Iteration number
repeat

$$f_X^{j+1}(a) := \frac{1}{n}\sum_{i=1}^{n}\frac{f_Y((x_i+y_i)-a)\,f_X^j(a)}{\int_{-\infty}^{\infty} f_Y((x_i+y_i)-a)\,f_X^j(a)}$$ (Bayes' rule)

j := j+1
until (stopping criterion met)

- Converges to maximum likelihood estimate.
  - D. Agrawal & C.C. Aggarwal, PODS 2001.

Seems to work well!



Classification

- Naïve Bayes
  - Assumes independence between attributes.
- Decision Tree
  - Correlations are weakened by randomization, not destroyed.

Algorithms

- “Global” Algorithm
  - Reconstruct for each attribute once at the beginning
- “By Class” Algorithm
  - For each attribute, first split by class, then reconstruct separately for each class.

Experimental Methodology

- Compare accuracy against
  - Original: unperturbed data without randomization.
  - Randomized: perturbed data but without making any corrections for randomization.
- Test data not randomized.
- Synthetic data benchmark from [AGI+92].
- Training set of 100,000 records, split equally between the two classes.

Synthetic Data Functions

- F3

  ((age < 40) and

  (((elevel in [0..1]) and (25K <= salary <= 75K)) or

  ((elevel in [2..3]) and (50K <= salary <= 100K))) or

  ((40 <= age < 60) and ...

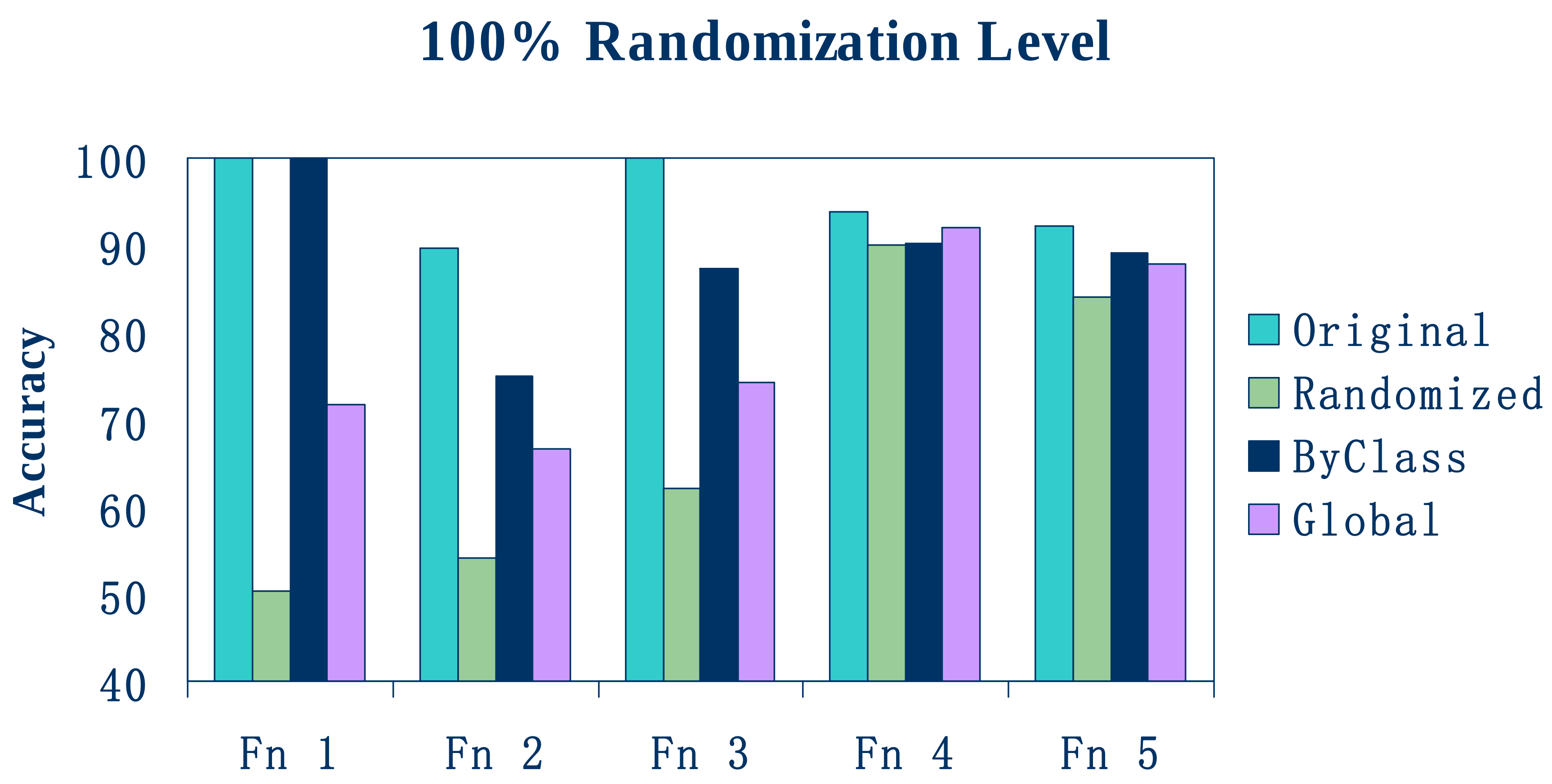
- F4

  (0.67 x (salary+commission) - 0.2 x loan - 10K) > 0

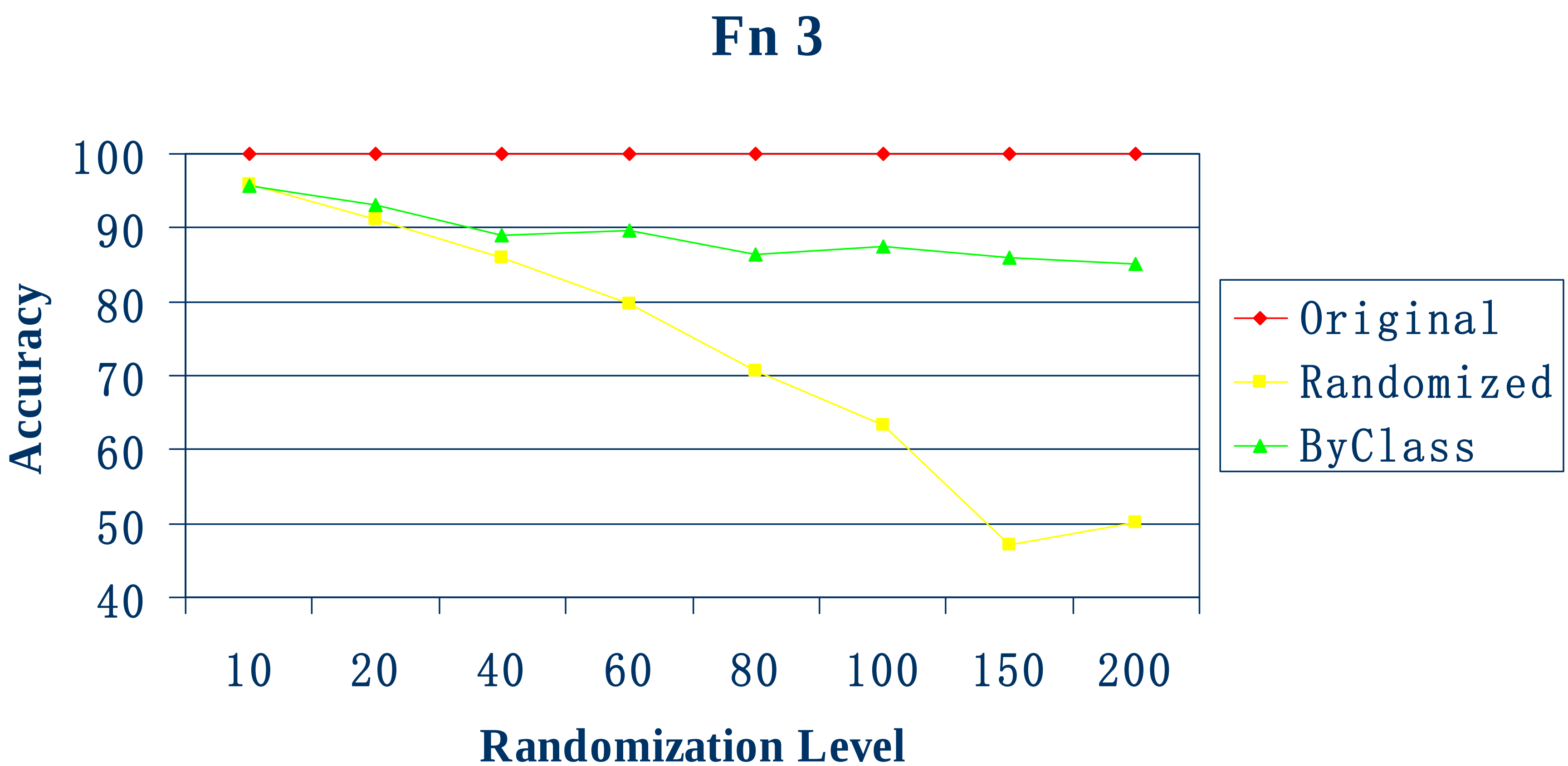
Quantifying Privacy

- Add a random value between -30 and +30 to age.
- If randomized value is 60
  - know with 90% confidence that age is between 33 and 87.
- Interval width “amount of privacy”.
  - Example: (Interval Width : 54) / (Range of Age: 100) ♏ 54% randomization level @ 90% confidence

Acceptable loss in accuracy



Accuracy vs. Randomization Level



Outline

- Motivation
- Randomization Approach
  - Application: Web Demographics
- Cryptographic Approach
  - Application: Inter-Enterprise Data Mining
  - Y. Lindell and B. Pinkas, “Privacy Preserving Data Mining”, *Crypto 2000*, August 2000.
- Challenges
  - Application: Privacy-Sensitive Security Profiling

Inter-Enterprise Data Mining

- Problem: Two parties owning confidential databases wish to build a decision-tree classifier on the union of their databases, without revealing any unnecessary information.
- Horizontally partitioned.
  - Records (users) split across companies.
  - Example: Credit card fraud detection model.
- Vertically partitioned.
  - Attributes split across companies.
  - Example: Associations across websites.

Cryptographic Adversaries

- Malicious adversary: can alter its input, e.g., define input to be the empty database.
- Semi-honest (or passive) adversary: Correctly follows the protocol specification, yet attempts to learn additional information by analyzing the messages.

Yao's two-party protocol

- Party 1 with input *x*
- Party 2 with input *y*
- Wish to compute *f(x,y)* without revealing *x*,*y*.
- Yao, “How to generate and exchange secrets”, FOCS 1986.

Private Distributed ID3

- Key problem: find attribute with highest information gain.
- We can then split on this attribute and recurse.
  - Assumption: Numeric values are discretized, with *n*-way split.

Information Gain

- Let
  - T = set of records (dataset),
  - $T(c_i)$ = set of records in class i,
  - $T(c_i,a_j)$ = set of records in class i with value(A) = a_j.
  - Entropy(T) $= \sum_i -\frac{|T(c_i)|}{|T|} \log \frac{|T(c_i)|}{|T|}$
  - Gain(T,A) = Entropy(T) $- \sum_j \frac{|T(a_j)|}{|T|} \times \text{Entropy}(T(a_j))$
- Need to compute
  - $\Sigma_j \, \Sigma_i \, |T(a_j, c_i)| \; \log |T(a_j, c_i)|$
  - $\Sigma_j \; |T(a_j)| \; \log |T(a_j)|$.

Selecting the Split Attribute

- Given v1 known to party 1 and v2 known to party 2, compute (v1 + v2) log (v1 + v2) and output random shares.
  - Party 1 gets Answer - δ
  - Party 2 gets δ, where δ is a random number
- Given random shares for each attribute, use Yao's protocol to compute information gain.

Summary (Cryptographic Approach)

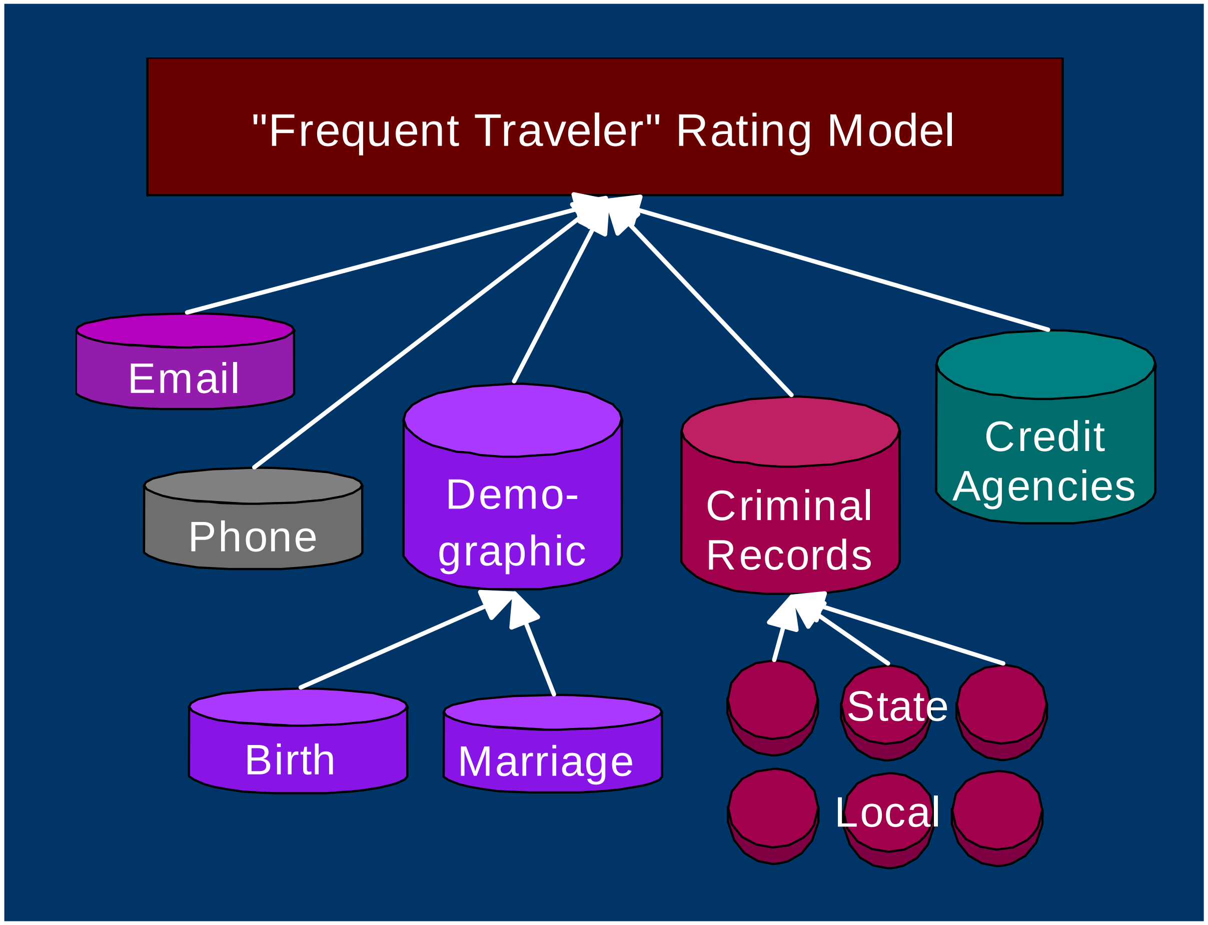
- Solves different problem (vs. randomization)
  - Efficient with semi-honest adversary and small number of parties.
  - Gives the same solution as the non-privacy-preserving computation (unlike randomization).
  - Will not scale to individual user data.
- Can we extend the approach to other data mining problems?
  - J. Vaidya and C.W. Clifton, “Privacy Preserving Association Rule Mining in Vertically Partitioned Data”. (SIGKDD02)

Outline

- Motivation
- Randomization Approach
  - Application: Web Demographics
- Cryptographic Approach
  - Application: Inter-Enterprise Data Mining
- Challenges
  - Application: Privacy-Sensitive Security Profiling
  - Privacy Breaches
  - Clustering & Associations

Privacy-sensitive Security Profiling

- Heterogeneous, distributed data.
- New domains: text, graph



Potential Privacy Breaches

- Distribution is a spike.
  - Example: Everyone is of age 40.
- Some randomized values are only possible from a given range.
  - Example: Add U[-50,+50] to age and get 125 ♏ True age is ❍ 75.
  - Not an issue with Gaussian.

Potential Privacy Breaches (2)

- Most randomized values in a given interval come from a given interval.
  - Example: 60% of the people whose randomized value is in [120,130] have their true age in [70,80].
  - Implication: Higher levels of randomization will be required.
- Correlations can make previous effect worse.
  - Example: 80% of the people whose randomized value of age is in [120,130] and whose randomized value of income is [...] have their true age in [70,80].
- Challenge: How do you limit privacy breaches?

Clustering

- Classification: ByClass partitioned the data by class & then reconstructed attributes.
  - Assumption: Attributes independent given class attribute.
- Clustering: Don't know the class label.
  - Assumption: Attributes independent.
- Global (latter assumption) does much worse than ByClass.
- Can we reconstruct a set of attributes together?
  - Amount of data needed increases exponentially with number of attributes.

Associations

- Very strong correlations ◆ Privacy breaches major issue.
- Strawman Algorithm: Replace 80% of the items with other randomly selected items.
  - 10 million transactions, 3 items/transaction, 1000 items
  - <a, b, c> has 1% support = 100,000 transactions
  - <a, b>, <b, c>, <a, c> each have 2% support
    - 3% combined support excluding <a, b, c>
  - Probability of retaining pattern = $0.2^3 = 0.8\%$
    - 800 occurrences of <a, b, c> retained.
  - Probability of generating pattern = 0.8 * 0.001 = 0.08%
    - 240 occurrences of <a, b, c> generated by replacing one item.
  - Estimate with 75% confidence that pattern was originally present!
  - PODS2003

Associations (cont.)

- "Where does a wise man hide a leaf? In the forest. But what does he do if there is no forest?" ... "He grows a forest to hide it in." -- G.K. Chesterton
- A. Evfimievski, R. Srikant, R. Agrawal, J. Gehrke, “Privacy Preserving Mining of Association Rules”, KDD 2002.
- S. Rizvi, J. Haritsa, “Privacy-Preserving Association Rule Mining”, VLDB 2002.

Summary

- Have your cake and mine it too!
  - Preserve privacy at the individual level, but still build accurate models.
- Challenges
  - Privacy Breaches, Security Applications, Clustering & Associations
- Opportunities
  - Web Demographics, Inter-Enterprise Data Mining, Security Applications

My several cents

- When does randomization fail?
- How about the privacy preserving search in encrypted data?
- Practical tools with reasonable efficiency.

Information Sharing Across Private Databases

Presented by Hong Ge

Motivating Applications

- **Selective Document Sharing**

  compute the join of D_R and D_S using the join predicate $f(|d_R \cap d_S|, |d_R|, |d_S|) > \tau$, for some similarity function f and threshold τ, where f could be $|d_R \cap d_S|/(|d_R|+|d_S|)$

- **Medical Research**

  select pattern, reaction, count(*)
  from T_R, T_S
  where T_R.person_id = T_S.person_id and
  T_S.drug = "true"
  group by T_R.pattern, T_S.reaction

Current Techniques

- Trusted Third Party
  - Requirement too strong, impractical
- Secure Multi-Party Computation
  - Cost too high, impractical

Problem Statement

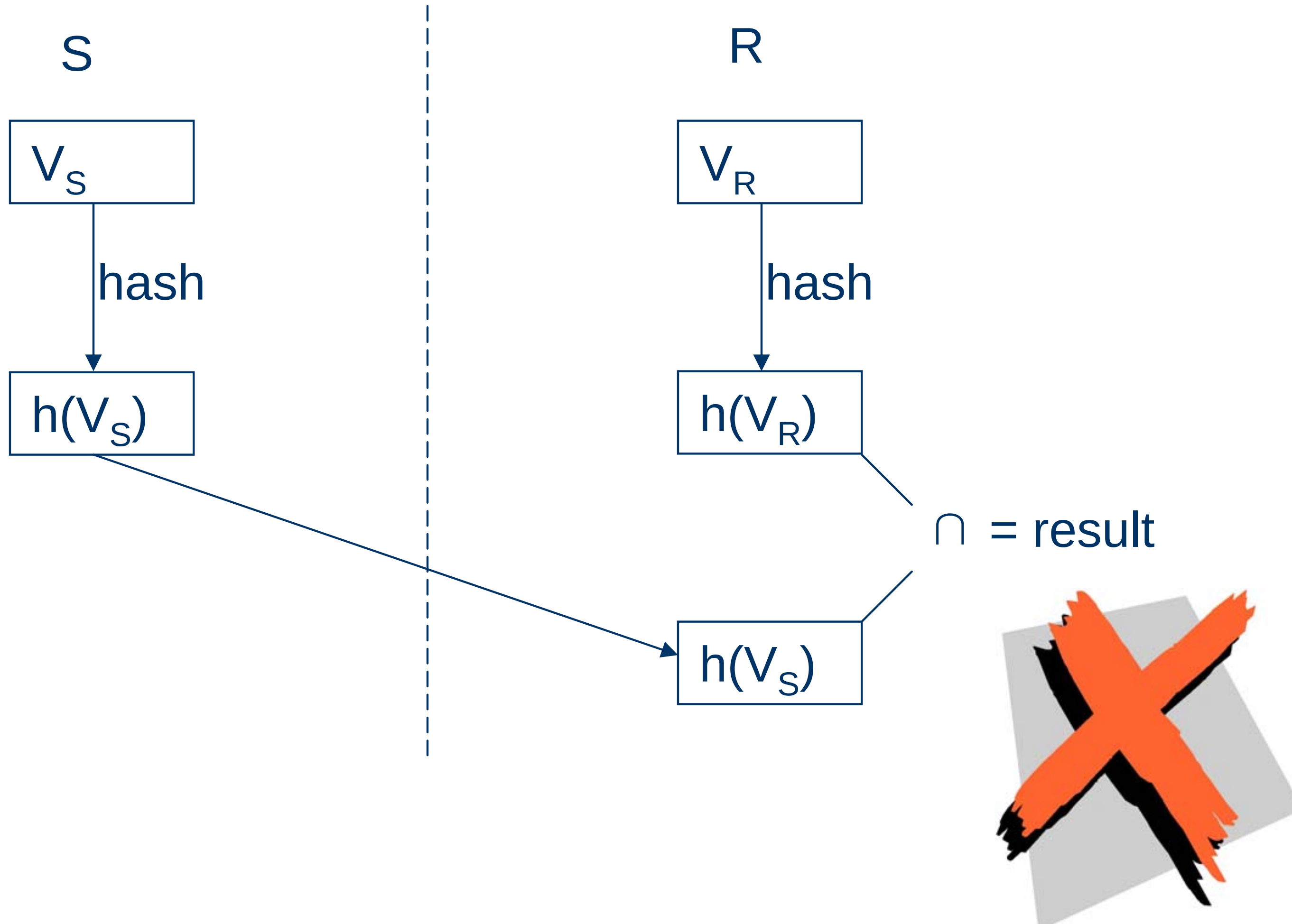
- Ideal case
  - Let there be two parties R (receiver) and S (sender) with databases D_R and D_S respectively. Given a database query Q spanning the tables in D_R and D_S, compute the answer to Q and return it to R without revealing any additional information to either party.

- Minimal Sharing
  - Given some categories of information I, allow revealing information contained in I.

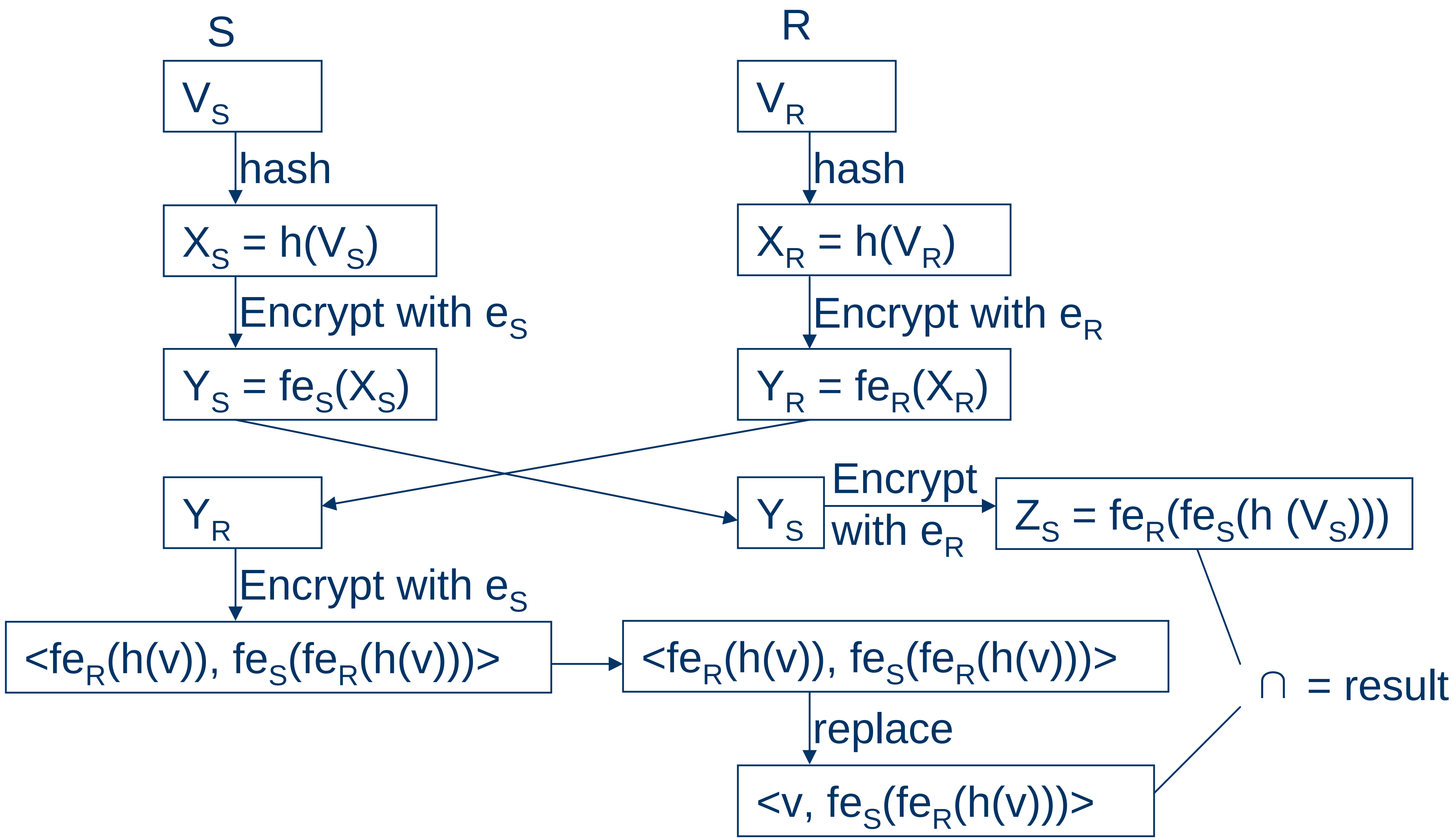
Limitations

- Multiple Queries
  - No guarantee on how much the parties might learn by combining the results of multiple queries
- Schema Discovery and Heterogeneity
  - Assume database schemas are known and don't address heterogeneity

Operation (1)   Intersection



Operation (1) Intersection

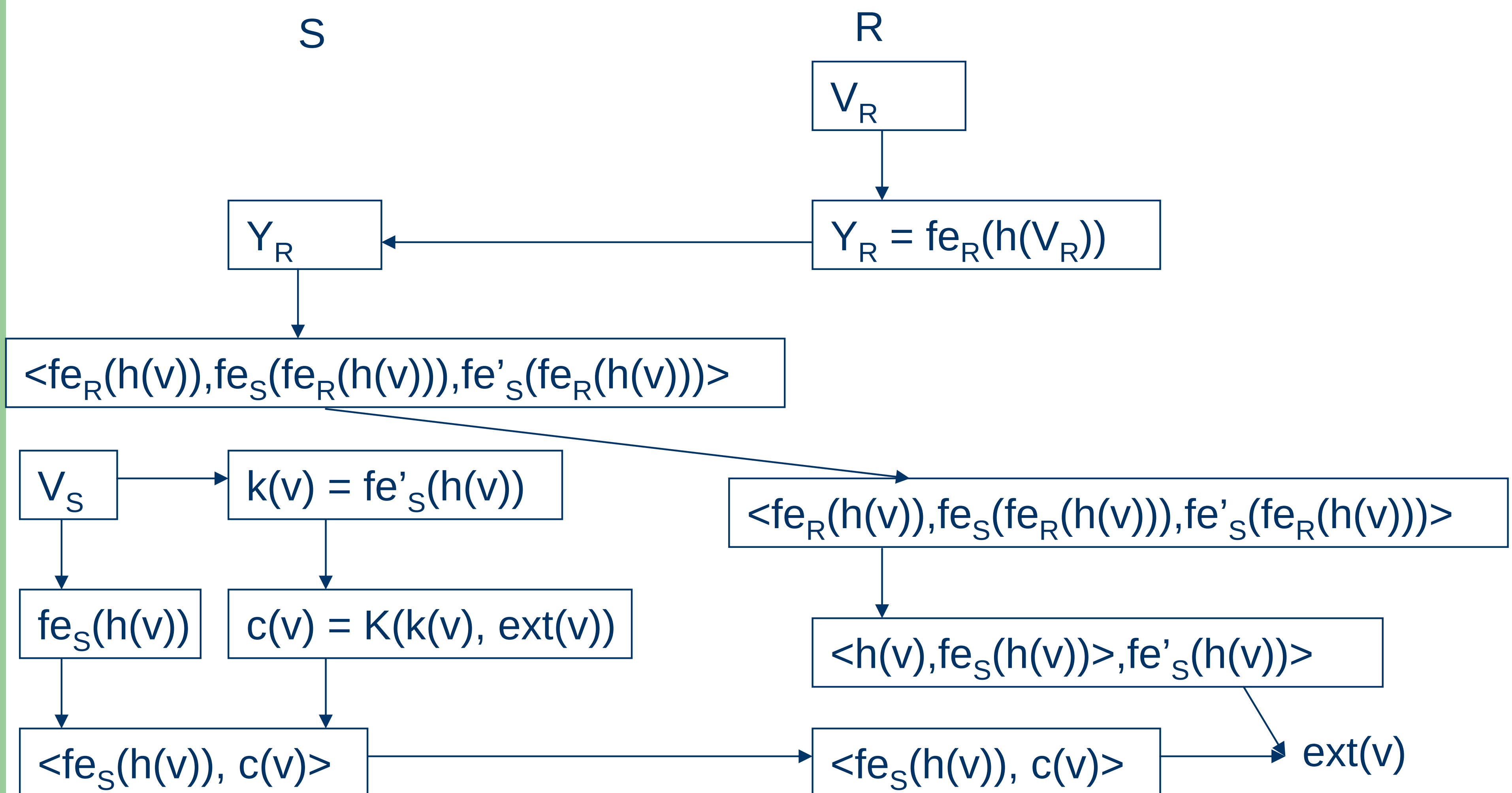


Operation (2) Equijoin

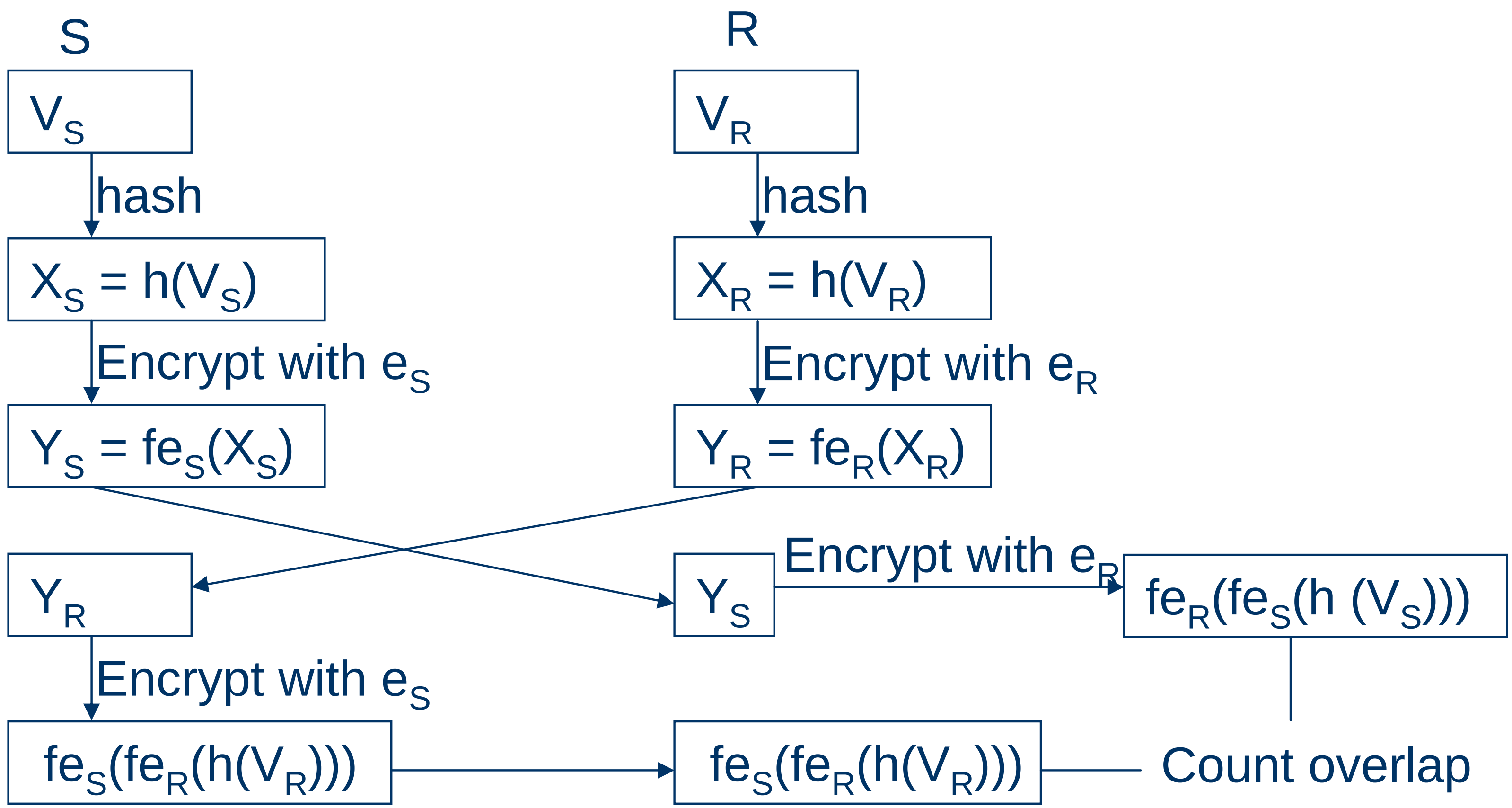
Encrypt ext(v) using h(v)?

Use $k(v) = fe'_S\,(\,h(v))$ instead!

Operation (2) Equijoin



Operation (3) Intersection Size



Operation (4)   Equijoin Size

- Follow the intersection size protocol, except that we allow V_R and V_S to be multi-sets.

- What else besides $|V_R|$, $|V_S|$, $|V_R \bowtie V_S|$ do they learn?
  - R learns distribution of duplicates in S
  - S learns distribution of duplicates in R
  - For each partition $V_R(d)$ and each partition $V_S(d')$, R learns $|VR(d) \cap VS(d')|$
    - If all values have the same number of duplicates, $|V_R \cap V_S|$
    - If no two values have the same number of duplicates, $V_R \cap V_S$

Cost Analysis

- Computation cost:
  - *Intersection:* $2C_e(|V_S| + |V_R|)$
  - *Join:* $2C_e|V_S|+5C_e|V_R|$
- Communication cost:
  - *Intersection:* $(|V_S|+ 2|V_R|)k$ bits
  - *Join:* $(|V_S|+3|V_R|)k +|V_S|k'$ bits

C_e: cost of encryption/decription.

k: length of encrypted v.

k': size of encrypted ext(v).

Cost Analysis for Applications

- Selective Document Sharing
  - Computation: $|D_R|\cdot|D_S|\cdot(|d_R|+|d_S|)\cdot 2C_e$
    - 2 hours given $|D_R| = 10$, $|D_S| = 100$, $|d_R| = |d_S| = 1000$
  - Communication: $|D_R|\cdot|D_S|\cdot(|d_R|+2|d_S|)\cdot k$ bits
    - 35 minutes
- Medical Research
  - Computation: $2(|V_R|+|V_S|)\cdot 2C_e$
    - 4 hours given |VR| = |VS| = 1 million
  - Communication: $2(|V_R|+|V_S|)\cdot 2k$ bits
    - 1.5 hours

Computation speed: 0.02 s for 1024-bit number
Communication speed: 1.544Mb/s
Processors used: 10

Future research

- Will we be able to obtain much faster protocols if we are willing to disclose additional information?
- Can we extend to other database operations such as aggregations?

Hippocratic Databases and Implementing P3P Using Database Technology* - papers by Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu

Presented by Wesley C. Maness

* Platform for Privacy Preferences

Outline

- Brief Overview of Hippocratic Databases
  - Definition
  - Architectural Principles and Proposed ***Strawman*** Model
  - Open Problems/Challenges
- P3P Using Database Technology
  - Definition
  - Example Privacy Policy XML format
  - P3P Implementations
  - DB Schema for P3P & Translation
  - Open Problems/Challenges

"And about whatever I may see or hear in treatment, or even without treatment, in the life of human beings — things that should not ever be blurted out outside — I will remain silent, holding such things to be unutterable..." – Hippocratic Oath

What is a Hippocratic Database?

- a database that includes privacy as a central concern
- inspired by Hippocratic Oath that serves as basis of doctor-patient relationship
- Another way to provide Privacy Preservation; other, previous systems are
  - Statistical
    - Motivated by the desire to be able to provide statistical information without compromising sensitive information about individuals
    - Query restriction : restricting the size of the query results , controlling the overlap among the queries , keeping the audit trails of all answered queries.
    - Data perturbation : swapping the values between records , adding the noise to the databases and the to query output.
  - Secure
    - Multiple levels of the security to be defined and associated with individual attribute values
    - Query with lower level of security can not read a data item requiring higher level of clearance.
    - Two queries with different levels of security can produce different answers on the same database.

Architectural Principles

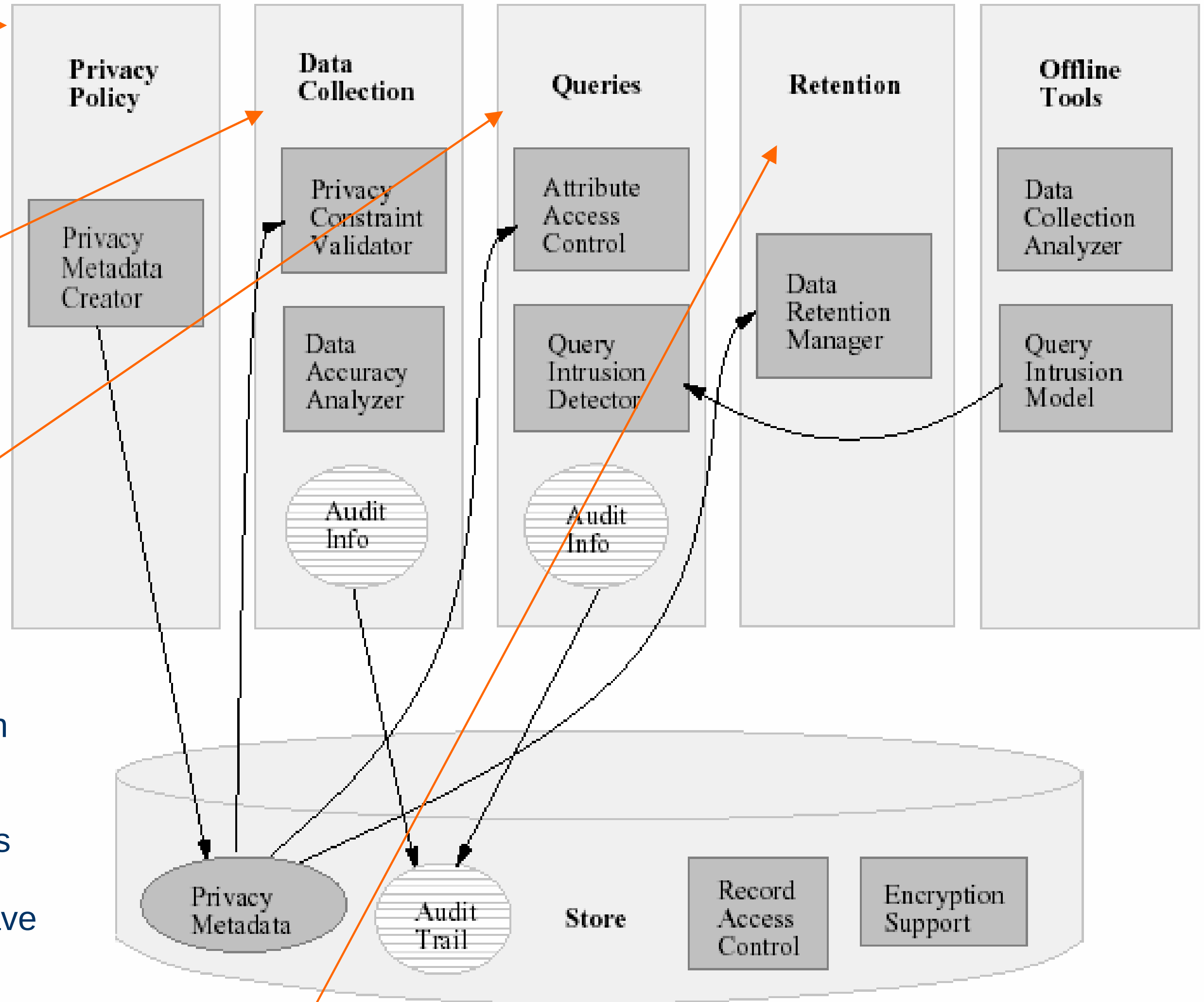
- **Purpose Specification**
  Associate with data the purposes for collection
- **Consent**
  Obtain donor's consent on the purposes
- **Limited Collection**
  Collect minimum necessary data
- **Limited Use**
  Run only queries that are consistent with the purposes
- **Limited Disclosure**
  Do not release data without donor's consent
- **Limited Retention**
  Do not retain data beyond necessary
- **Accuracy**
  Keep data accurate and up-to-date
- **Safety**
  Protect against theft and other misappropriations
- **Openness**
  Allow donor access to data about the donor
- **Compliance**
  Verifiable compliance with the above principles

Strawman Design

- map privacy policy to privacy-policies table
- map access control policy to privacy-authorizations table

- compare privacy policy to user's privacy preferences
- users can opt-in or opt-out of each purpose
- keep audit trail as proof of user's consent
- check data for accuracy before or after insertion

- Before Query:
  - check to make sure that attributes in query are listed for that purpose
- During Query:
  - access to individual tuples of table is restricted by purpose
  - queries have purpose and tuples have purpose
  - do not return tuples where query purpose $\neq$ tuple purpose
- After Query:
  - look for unusual patterns of access that are not typical for that purpose and that user
  - add query to audit trail in order to show who had access to what and when



- delete data that has outlived it's purpose
- if same data collected for more than one purpose use maximum retention period

Conclusion & Open Problems of Hippocratic Databases

- need better language for privacy policies and preferences
- how does privacy management impact performance
- limited collection requires access analysis and granularity analysis
- Impersonation of an authorized user problem.
- Number of purposes; there are performance penalties; way to enhance purpose evaluations.
- Partial retention periods have been mentioned, i.e. how to deal with a three month private and a three month public retention periods.
- QID (Query Intrusion Detection) is reactive; not proactive. Trace Logs, for example don’t protect, they detect.
- Rethinking traditional database design goals.. Is it necessary in implementing a HD?
- “Probably won’t work; the problems presented here aren’t really interesting computer science problems; good idea in concept bad idea in practice” - wcm

P3P Overview

- P3P has two parts:
  - Privacy Policies: An XML format in which a web site can encode its data-collection and data-use practices
  - Privacy Preferences: A machine-readable specification of a user's preferences that can be programmatically compared against a privacy policy
- give web users more control over their personal information
- web sites encode privacy policy in a machine-readable XML format
- user can compare privacy policy to personal privacy preferences
- does not provide mechanism for enforcement

Example Privacy Policy in P3P

```
<POLICY>
  ... ...
  <STATEMENT>
    <PURPOSE><current/></PURPOSE>
    <RECIPIENT><ours/><same/></RECIPIENT>
    <RETENTION><stated-purpose/></RETENTION>
    <DATA-GROUP>
      <DATA ref="#user.name"/>
      <DATA ref="#user.home-info.postal/>
      <DATA ref="#dynamic.miscdata">
        <CATEGORIES><purchase/></CATEGORIES>
      </DATA>
    </DATA-GROUP>
  </STATEMENT>
```

P3P Implementations 1 of 2 (Client-Centric)

There are two parts, in this implementation, in deploying P3P. Web sites first create and install policy files at their sites (Fig. 3).

Then as users browse a web site, their preferences are checked against a site's policy before they access the site (Fig. 4)

Pros/Cons:

- •preference checking at client leads to heavier clients.
- •Upgrade in P3P spec may require upgrade in every client
- •Server-trust is a problem

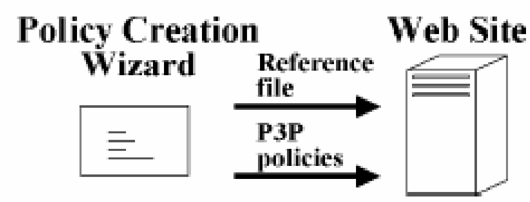


Figure 3: Creation and Installation of Policies (Client-Centric)

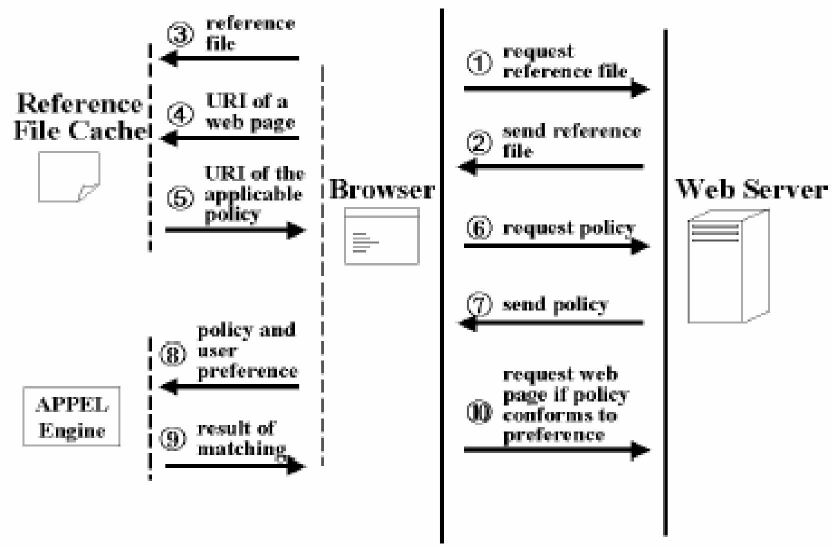


Figure 4: Policy-Preference Matching (Client-Centric)

P3P Implementations 2 of 2 (Server-Centric)

In this architecture, a website deploying P3P first installs its privacy policies in a database system, as seen in Fig. 5.

The database querying is used for matching a user's preferences against privacy policies as show in Fig. 6.

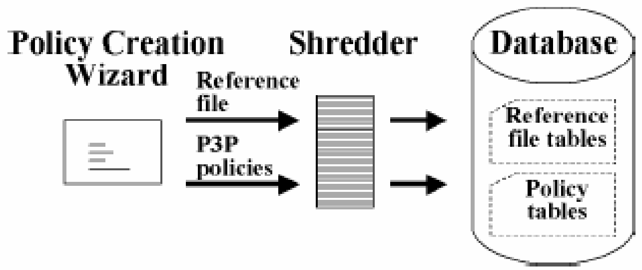
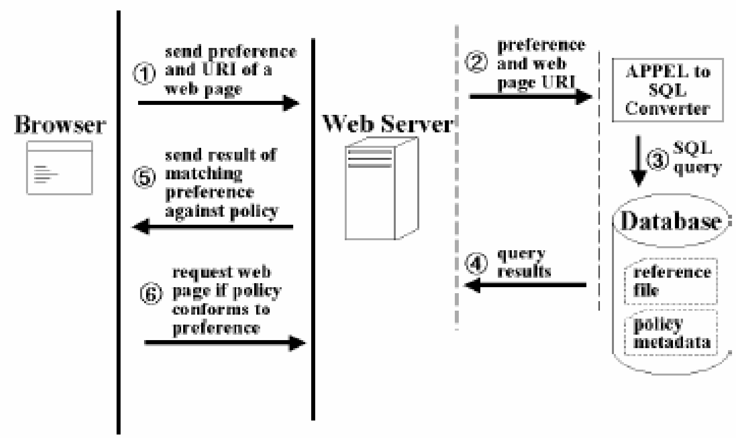


Figure 5: Creation and Installation of Policies (Server-Centric)



DB Schema for P3P

The SQL query corresponding to an APPEL preference will depend on the SQL tables used for storing the P3P policies.

Fig. 8 shows the algorithm for decomposing P3P Schema into tables.

Fig. 9 shows the table created for the DATA element using this algorithm. The Data table will contain one row for every DATA element appearing in a policy

```
// e.name() returns the name of the element e
for each element e defined in the P3P policy do
    create a table such that
        (a) the name of the table is e.name()
        (b) the columns of the table consist of
            (i)   an id column whose name is e.name()
                  concatenated with "_id"
            (ii)  foreign key comprising of the primary key
                  of the table corresponding to the parent element
            (iii) one column for each attribute of e
        (c) the primary key of the table comprises of
            concatenation of columns in (i) and (ii)
```

Figure 8: Schema Decomposition Algorithm

	id column	foreign key columns			attribute columns	
Data	data_id	datagroup_id	statement_id	policy_id	ref	optional

Figure 9: The Data Table

Translation

There must exists a mechanism to translate ones P3P (APPEL) Policies into SQL. This utilizes translation algorithms, not shown here.

```
<appel:RULE behavior="block">
  <POLICY>
    <STATEMENT>
      <PURPOSE appel:connective="or">
        <admin/>
        <contact required="always"/>
      </PURPOSE>
    </STATEMENT>
  </POLICY>
</appel:RULE>
```

Translate APPEL expression into SQL

```
                // main(<appel:RULE>)
SELECT 'block'     // rule's behavior
FROM ApplicablePolicy
                // ApplicablePolicy represents
                // subquery that returns record
                // with ID of applicable policy.
WHERE
EXISTS (
                // match(<POLICY>)
SELECT *
FROM Policy
WHERE Policy.policy_id=ApplicablePolicy.policy_id AND
EXISTS (
                // match(<STATEMENT>)
  SELECT *
  FROM Statement
  WHERE Statement.policy_id = Policy.policy_id AND
  EXISTS (
                // match(<PURPOSE>)
    SELECT *
    FROM Purpose
    WHERE
    Purpose.policy_id = Statement.policy_id AND
    Purpose.statement_id = Statement.statement_id AND
    (EXISTS (
                // match(<admin>)
      SELECT *
      FROM Admin
      WHERE
      Admin.policy_id = Purpose.policy_id AND
      Admin.statement_id = Purpose.statement_id AND
      Admin.purpose_id = Purpose.purpose_id )
                // back to match(<PURPOSE>)
    OR          // line 21 of match()
    EXISTS (
                // match(<contact required=...>)
      SELECT *
      FROM Contact
      WHERE
      Contact.policy_id = Purpose.policy_id AND
      Contact.statement_id = Purpose.statement_id AND
      Contact.purpose_id = Purpose.purpose_id AND
                // lines 16-17 of match()
      Contact.required = 'always' )
      )         // back to match(<PURPOSE>)
    )           // back to match(<STATEMENT>)
  )             // back to match(<POLICY>)
)               // back to match(<appel:RULE>)
```

Open Problems/Challenges

- Major Assumption: how does one enforce P3P in a server-centric DB model? This seems to be the biggest criticism… Compliancy Checks, a local on-site Security Officer. .etc. how to arrange…
- Implicitly requires that server-centric models need to standardize their server-centric architecture… not likely…
- Interesting: there has been significant research in XML DBs however not revealing significant findings, will the same events happen to P3P DBs?
- P3P, initially accepted strongly by community, but recently has disappeared; example; P3P was originally for handing web purchasing agreements and cookie management. Now that most browsers self-include cooking management, not P3P, a need for P3P at the browser is not really needed. Did P3P shoot themselves in the foot?

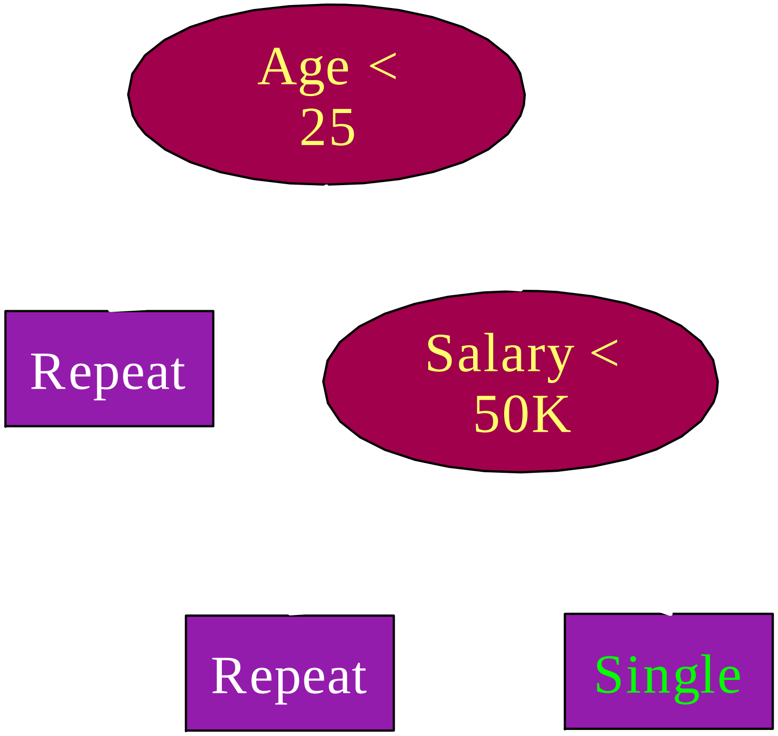
Backup slides for Zheng Ma

Randomization to protect Privacy

- Return $x+r$ instead of x, where r is a random value drawn from a distribution
  - Uniform
  - Gaussian
- Fixed perturbation - not possible to improve estimates by repeating queries
- Reconstruction algorithm knows parameters of r's distribution

Classification Example

Age	Salary	Repeat Visitor?
23	50K	Repeat
17	30K	Repeat
43	40K	Repeat
68	50K	Single
32	70K	Single
20	20K	Repeat



Decision-Tree Classification

```
Partition(Data S)
    begin
        if (most points in S belong to same class)
        return;
    for each attribute A
        evaluate splits on attribute A;
    Use best split to partition S into S1 and S2;
    Partition(S1);
    Partition(S2);
end
```

Training using Randomized Data

- Need to modify two key operations:
  - Determining split point
  - Partitioning data
- When and how do we reconstruct distributions:
  - Reconstruct using the whole data (globally) or reconstruct separately for each class
  - Reconstruct once at the root node or at every node?

Training using Randomized Data (2)

- Determining split attribute & split point:
  - Candidate splits are interval boundaries.
  - Use statistics from the reconstructed distribution.
- Partitioning the data:
  - Reconstruction gives estimate of number of points in each interval.
  - Associate each data point with an interval by sorting the values.

Work in Statistical Databases

- Provide statistical information without compromising sensitive information about individuals (surveys: AW89, Sho82)
- Techniques
  - Query Restriction
  - Data Perturbation
- Negative Results: cannot give high quality statistics and simultaneously prevent partial disclosure of individual information [AW89]

Statistical Databases: Techniques

- Query Restriction
  - restrict the size of query result (e.g. FEL72, DDS79)
  - control overlap among successive queries (e.g. DJL79)
  - suppress small data cells (e.g. CO82)
- Output Perturbation
  - sample result of query (e.g. Den80)
  - add noise to query result (e.g. Bec80)
- Data Perturbation
  - replace db with sample (e.g. LST83, LCL85, Rei84)
  - swap values between records (e.g. Den82)
  - add noise to values (e.g. TYW84, War65)

Statistical Databases: Comparison

- We do not assume original data is aggregated into a single database.
- Concept of reconstructing original distribution.
  - Adding noise to data values problematic without such reconstruction.