Privacy

(with some material from Lorrie Cranor, Pete Keleher)

- Common phrase: “Security and privacy”
- So far we have mostly talked about security
- What is the difference?

Privacy is a value so complex, so entangled in competing and contradictory dimensions, so engorged with various and distinct meanings, that I sometimes despair whether it can be usefully addressed at all.

Robert C. Post
Three Concepts of Privacy, 89 Geo. L.J. 2087 (2001)

Boundaries of self?

"The right to be let alone"

- Samuel D. Warren and Louis D. Brandeis,
The Right to Privacy, 4 Harv. L. Rev. 193 (1890)

"Our concern over our accessibility to others: the extent to which we are known to others, the extent to which others have physical access to us, and the extent to which we are the subject of others' attention."

- Ruth Gavison,
Privacy and the Limits of the Law, Yale Law Journal 89 (1980

Control over information?

"Privacy is the claim of individuals, groups or institutions to determine for themselves when, how, and to what extent information about them is communicated to others. … Each individual is continually engaged in a personal adjustment process in which he balances the desire for privacy with the desire for disclosure and communication."

- Alan Westin, Privacy and Freedom, 1967

Privacy definitions and goals

- Solitude, uninterrupted
- Unseen, unheard, unread
- Not talked about
- Not judged/misjudged
- Not profiled, targeted, treated differently
- Free to practice, make errors
- Being unknown
- Being forgotten
- Intimacy
- Control
- Boundaries

How privacy is protected

- Laws
- Self-regulation
- Technology

Case study: Databases

- Several possibilities
  - Medical data
  - Scientific research (on human subjects)
  - US census data
  - Employment data
  - …

  - Data about oneself (e.g., on smartphone)

Database privacy

- A user (or group of users) has authorized access to certain data in a database, but not to all data
  - E.g., user is allowed to learn certain entries only
  - E.g., user is allowed to learn aggregate data but not individual data (e.g., allowed to learn the average salary but not individual salaries)
  - E.g., allowed to learn trends (i.e., data mining) but not individual data
- How to enforce?
- Note: we are assuming that authentication/access control is already taken care of…

Database privacy

- Want to be able to discern statistical trends without violating (individual) privacy
  - An inherent tension!
- Questions:
  - [How to obtain the raw data in the first place?]
  - How to allow effective data mining while still maintaining (some level of) user privacy?
- Serious real-world problem
  - Federal laws regarding medical privacy
  - Data mining on credit card transactions, web browsing, movie recommendations, …

The problem

- A user may be able to learn unauthorized information via inference
  - Combining multiple pieces of authorized data
  - Combining authorized data with “external” knowledge
    - 87% of people identified by ZIP code + gender + date of birth
    - Someone with breast cancer is likely to be female

Database privacy

- The problem is compounded by the fact that 'allowing effective data mining' and 'privacy' are (usually) left vague
  - If so, solutions are inherently heuristic and ad-hoc
- Recent work toward formally pinning down what these notions mean

Two models

- Non-interactive data disclosure
  - Users given access to “all data” (after the data is anonymized/sanitized/ processed in some way)
  - Note: it does not suffice to just delete the names!
- Interactive mechanisms
  - Users given the ability to query the database

Example

- Say not allowed to learn any individual's salary

Name	UID	Years of service	Salary
Alice	001	12	$65,000
Evan	101	4	$90,000
Frank	110	8	$58,000

Request denied!

Example

Name	UID	Years of service	Salary
Alice	001	12	$65,000

Give me the list of all salaries

Alice ← 01 → $65,000

Bob ← 10 → $40,000

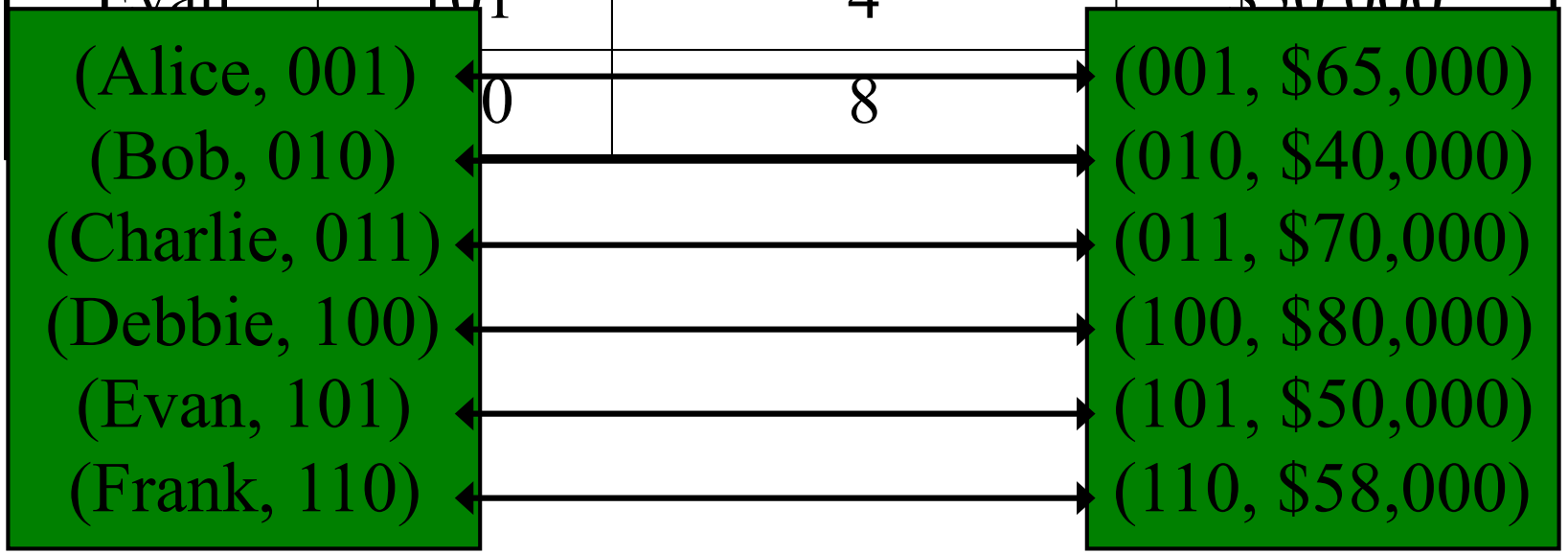
$40,000
$50,000
$58,000
$65,000
$70,000
$80,000

Solution: return … is independent of the tab… sorted)

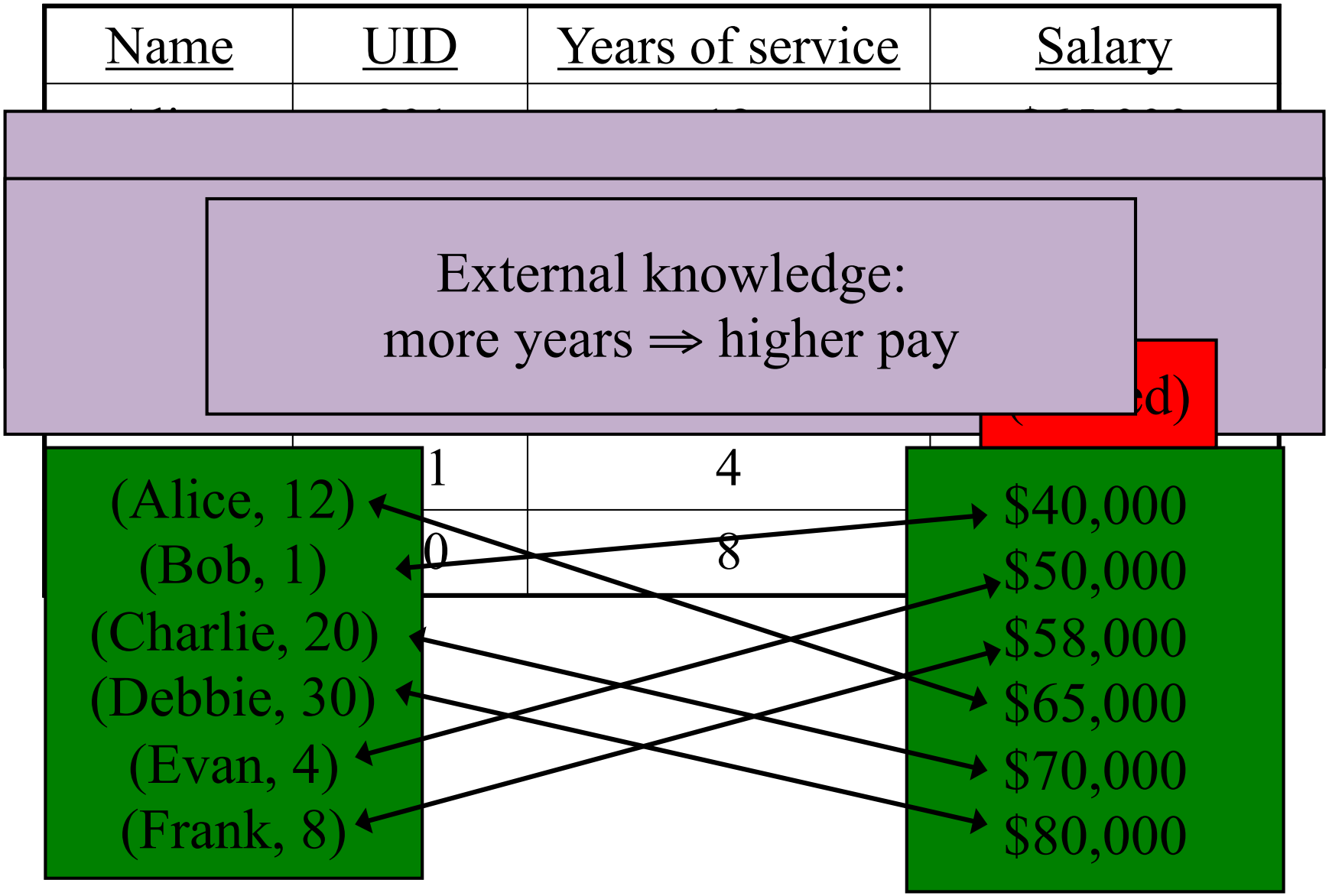
Example

Name	UID	Years of service	Salary
Alice	001	12	$65,000
Evan	101	4	$50,000
Frank	110	8	$58,000

Give me all UIDs and salaries



Example


Name
UID
Years of service
Salary
External knowledge:
more years ⇒ higher pay
1
4
0
8
(Alice, 12)
(Bob, 1)
(Charlie, 20)
(Debbie, 30)
(Evan, 4)
(Frank, 8)
$40,000
$50,000
$58,000
$65,000
$70,000
$80,000

Some solutions

- In general, an unsolved problem
- One idea: split data across several databases
- Another idea: Inference detection at query time
  - Store the set of all queries asked by a particular user, and look for disallowed inferences before answering any query
  - Note: will not prevent collusion among multiple users
  - Can also store the set of all queries asked by anyone, and look for disallowed inference there
- As always, tradeoff privacy/security and utility

Using several databases

- DB1 stores (name, address), accessible to all
- DB2 stores (UID, salary), accessible to all
- DB3 stores (name, UID), accessible to admin

What if I want to add data for “start-date” (and make it accessible to all)?

- Adding to DB2 can be problematic (why?)
- Adding to DB1 seems ok (can we prove this?)

Statistical databases

- Database that only provides data of a statistical nature (average, standard deviation, etc.)
  - Pure statistical database: only stores statistical data
  - Statistical access to ordinary database: stores all data but only answers statistical queries
  - Focus on the second type
- Aim is to prevent inference about any particular piece of information
  - One might expect that by limiting to aggregate information, individual privacy can be preserved

Turning raw data into a statistical database

- Two general methods
- **Query restriction:** Limit what queries are allowed. Allowed queried are answered correctly, while disallowed queries are simply not answered
- **Perturbation:** Queries answered “noisily”. Also includes “scrubbing” (or suppressing) some data
- (Could also be combined)

Query restriction

- Most basic: Only allow queries that involve more than some threshold t of users
- Example: only allow sum/average queries about a set S of people, where $|S| \geq 5$ (say)

Example

Name	Gender	Years of service	Salary
Alice	F	12	$65,000
Bob	M	1	$40,000
Charlie	M	20	$70,000
Dan	M	30	$80,000
Evan	M	4	$50,000

Give me SUM Salary WHERE Gender='F'

Request denied!

This won’t work

- Can you spot the problem?

Example

Name	Gender	Years of service	Salary
Frank	M	8	$58,000

Give me SUM Salary WHERE Gender=*

Give me SUM Salary WHERE Gender='M'

$363, 000

Alice's salary: $65,000

$298, 000

What went wrong?

- Each query on its own is allowed
- But inference possible once both queries are made
- Similar problems arise if the database is dynamic
  - E.g., determine a person's salary after they are hired by making the same query (over the entire database) before and after their hire date

Query restriction, redux

- More complicated: based on all prior history
  - E.g., if query for S was asked, do not allow query for a set S' if $|S' \Delta S|$ is “small”
- Drawbacks
  - Maintaining the entire query history is expensive
  - Difficult to define a privacy “breach”
  - What about adversary's external information?

Pairwise is not enough!

- Example
  - Say you want information about user i
  - Let S, T be non-overlapping sets, not containing i
  - Ask for SUM(Salary, S), SUM(salary, T), and SUM(salary, $S \cup T \cup \{i\}$)
- Inference: very difficult to detect and prevent…
  - NP-complete (in general) to determine whether a breach has occurred

Restrict whose queries?

- Across all users, or on a per-user basis?
  - If the former, utility is limited
  - If the latter, colluding users can cheat

Even worse news

- Query restriction itself may reveal information!
- Example: Averages released only if there are at least 2 data points being averaged
  - Request average salary of employees whose GPA is $\geq X$
  - No response: Fewer than 2 employees with GPA $\geq X$
  - If query(GPA $\geq$ X) answered but query(GPA $\geq$ X+Δ) not, there is at least one employee whose GPA lies between X and X+Δ

Another query restriction example

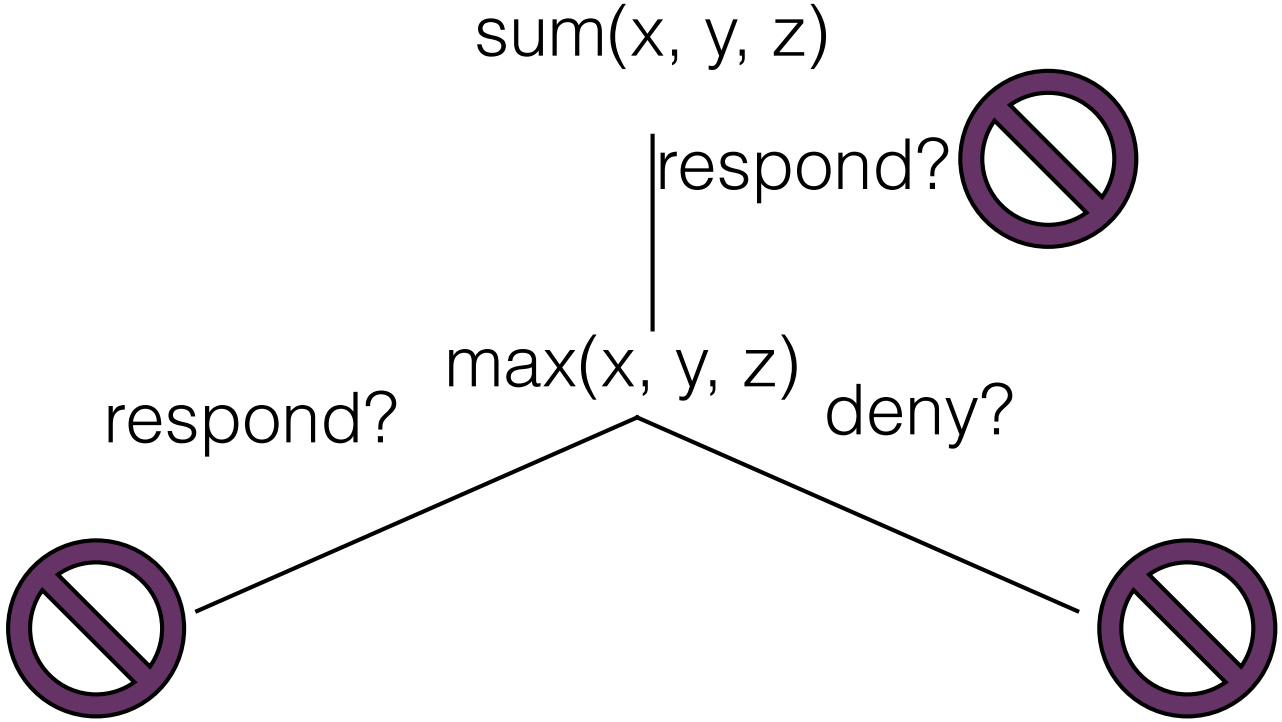
- Say we don't want adversary to learn our exact age
  - Deny query if the answer would exactly reveal the age
- Say age=30
- Adversary asks “is age ≥ 30?”, gets response “yes”
- Adversary asks “is age ≤ 30?”
  - Correct answer reveals the exact age!
  - But denying the query reveals the exact age also…

Yet another example

- Don't want adversary to learn any x, y, z exactly
- Consider the table with x = y = z = 1, where it is known that x, y, z ∈ {0,1,2}
- User requests sum(x, y, z), gets response 3
- User requests max(x, y, z)
  - If user learns the answer, can deduce that x = y = z = 1
  - But if the request is denied, the user can still deduce x = y = z = 1 (!!)

Predicting the future

- Try to “look ahead”
- Do not respond if there exists a subsequent query that will reveal information regardless of whether we answer



Restriction with “look-aheads”

- May need to look more than 1 level deep
- Computationally infeasible, even at 1 level deep
- Does it even work?
  - Denying “Is age ≥ 30?” reveals that age=30
  - Denying sum(x, y, z) reveals that x = y = z
- Even if answers don't uniquely reveal a value, they may leak lots of partial information

Instead: “Simulatable Auditing”

- Deny query if there is some database for which that query would reveal information
- Fixes the previous problems
- Even more computationally expensive
- Restricts utility – most queries denied

Belief tracking

- Track attacker's knowledge, making assumptions about initial state
  - Revise after each query is answered
- Refuse to answer any queries that would raise user's knowledge above some threshold
- Still need to be careful of leaking via refusals
  - Deny if there is any secret for which the answer would reveal information

Two methods, revisited

- **Query restriction:** Limit what queries are allowed. Allowed queried are answered correctly, while disallowed queries are simply not answered

- **Perturbation:** Queries answered “noisily”. Also includes “scrubbing” (or suppressing) some of the data

- (Could also be combined)

Perturbation

- Purposely add “noise”
- **Data perturbation**: add noise to entire table. Answer queries accordingly, or release entire perturbed dataset.
- **Output perturbation**: keep table intact, add noise to answers

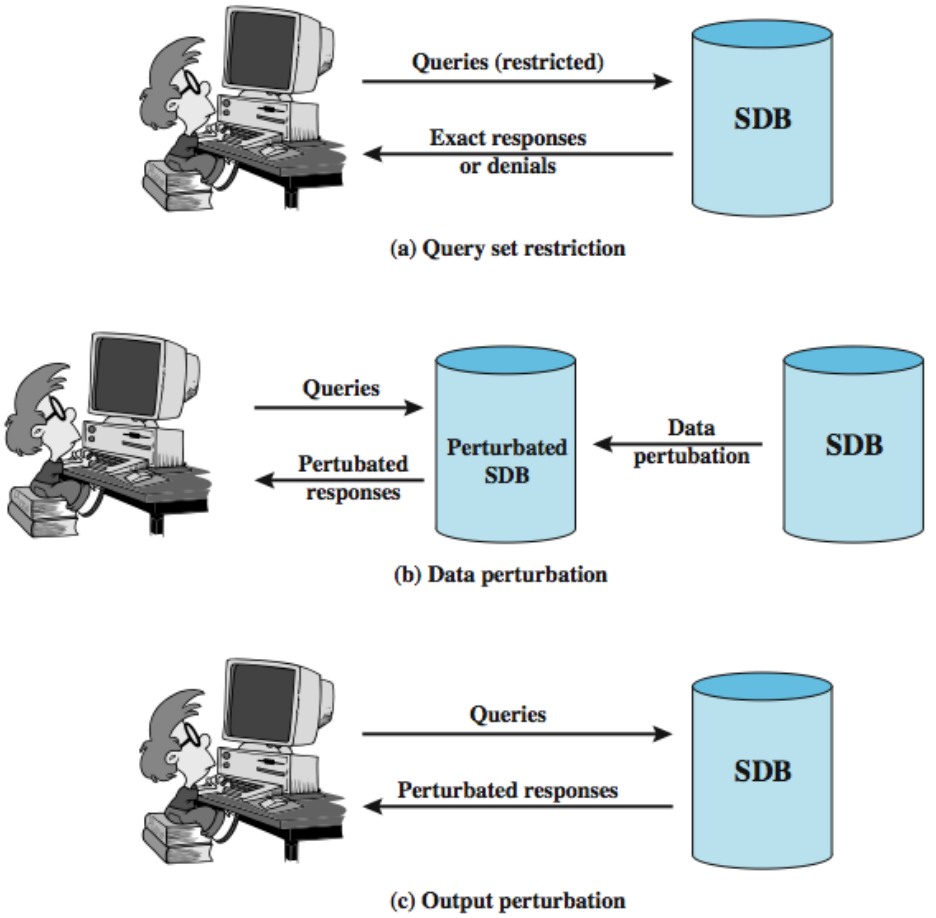


Figure 5.8 Approaches to Statistical Database Security (based on [ADAM89])

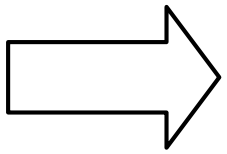
Perturbation: Privacy vs. utility

- No randomization – bad privacy but perfect utility
- Complete randomization – perfect privacy but zero utility

One technique: Data swapping

- Substitute and/or swap values
- While maintaining low-order statistics

F	Bio	4.0
F	CS	3.0
F	EE	3.0
F	Psych	4.0
M	Bio	3.0
M	CS	4.0
M	EE	4.0
M	Psych	3.0



F	Bio	3.0
F	CS	3.0
F	EE	4.0
F	Psych	4.0
M	Bio	4.0
M	CS	4.0
M	EE	3.0
M	Psych	3.0

Stats from any two columns are identical!

Another technique: Derived distribution

- For each sensitive attribute, determine a best-match probability distribution
- Generate fresh data according to distribution
- Populate the table with this fresh data
- Queries on the database can never “learn” more than what was learned initially

Another: Cleaning/Scrubbing

- Remove sensitive data
  - Data that can be used to breach anonymity
- *k-anonymity*: Ensure any “personally identifying information” is shared by at least k members

Example: 2-anonymity

Race	ZIP	Smoke?	Cancer?
Asian	0213x	Y	Y
Asian	0213x	Y	N
Asian	0214x	N	Y
Asian	0214x	Y	Y
Black	0213x	N	N
Black	0213x	N	Y
Black	0214x	Y	Y
Black	0214x	N	N
White	0213x	Y	Y
White	0213x	N	N
White	0214x	Y	Y
White	0214x	Y	Y

Problems with k-anonymity

- Hard to find the right balance of privacy/utility
- Security guarantees are unclear
  - What if I know the Asian person in ZIP code 0214x smokes?
  - Does not deal with out-of-band information
- What if all people who share some identifying information share the same sensitive attribute?

Output perturbation

- One approach: replace the query with a perturbed query, then return an exact answer to that
  - E.g., a query over some set of entries C is answered using some (randomly-determined) subset C' ⊆ C
  - User learns only the answer, not C'
- Second approach: add noise to the exact answer
  - E.g., answer SUM(salary, S) with
    SUM(salary, S) + noise

Negative result [Dinur-Nissim]

- Heavily paraphrased:

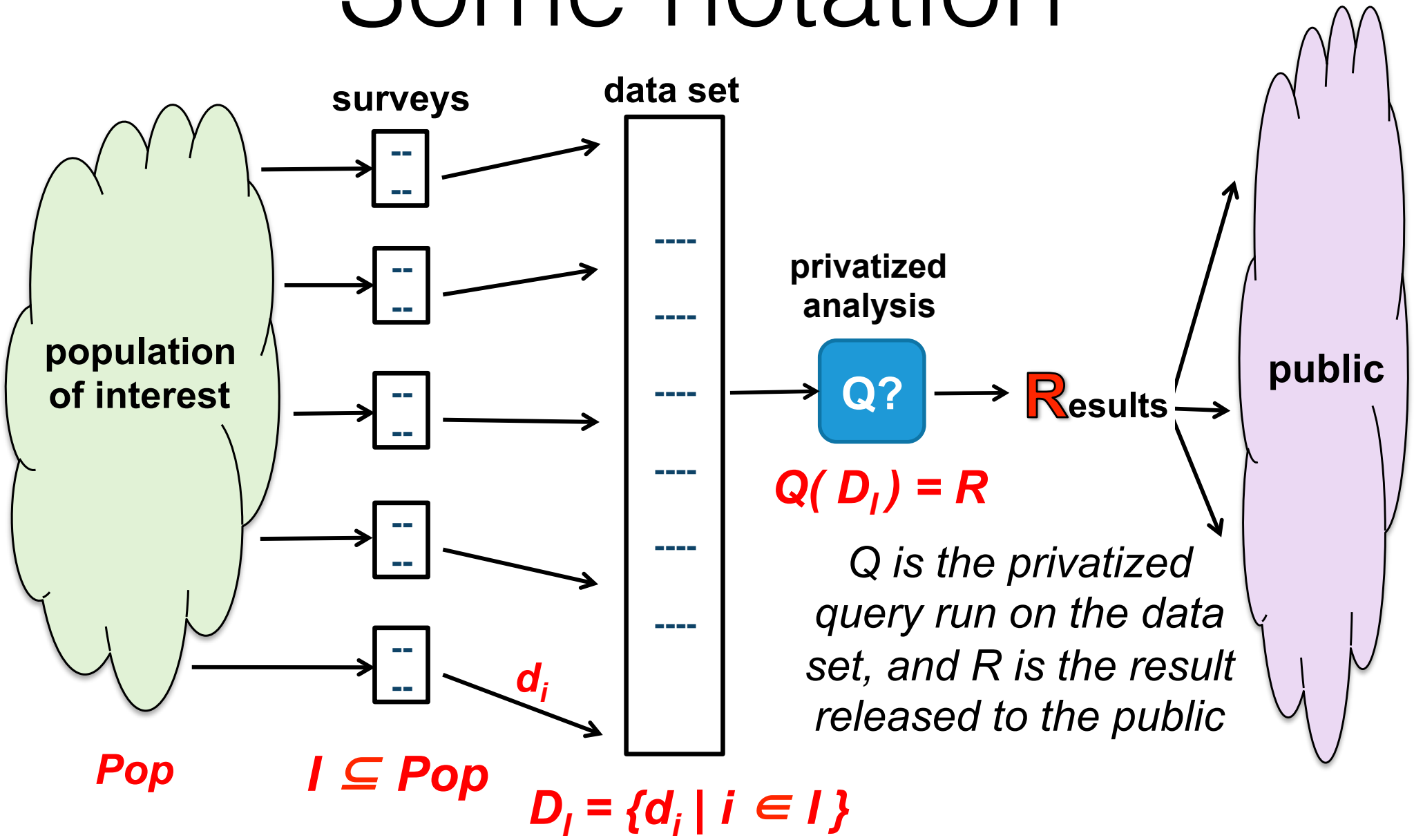
  Given a database with *n* rows, if *O(n)* queries are made to the database then essentially the entire database can be reconstructed *even if* $O(n^{1/2})$ *noise is added to each answer*

- But, very small error can be used when the total number of queries is kept small

Formalizing privacy

- Approaches so far don't formally define privacy
- What is the goal? How well is it achieved?
- Recent work (differential privacy) tries to fix this.
  - Develop definitions
  - Provable schemes to achieve them

Some notation



What do we want? (privacy)

- My answer has no impact on the released results

$$Q(D_{(I\text{-}me)}) = Q(D_I)$$

- Any attacker looking at published **R** can't learn anything about me personally

$$\Pr[secret(me) \mid R] = \Pr[secret(me)]$$

Why can't we have it?

- If individual answers had no impact, results would be useless!

By induction,
$Q(D_{(l)}) = Q(D_{\varnothing})$

- Trends in **R** may be true of me too!

$\Pr[secret(\text{me}) \mid secret(\text{Pop})] > \Pr[secret(\text{me})]$

Why can’t we have it?

- If attacker knows things about me relative to the general population (I’m 1.5x average age), then knows things about me even if I don’t submit a survey!

age(me) = 2*mean_age

mean_age = 16

age(me) = 32

What can we have instead?

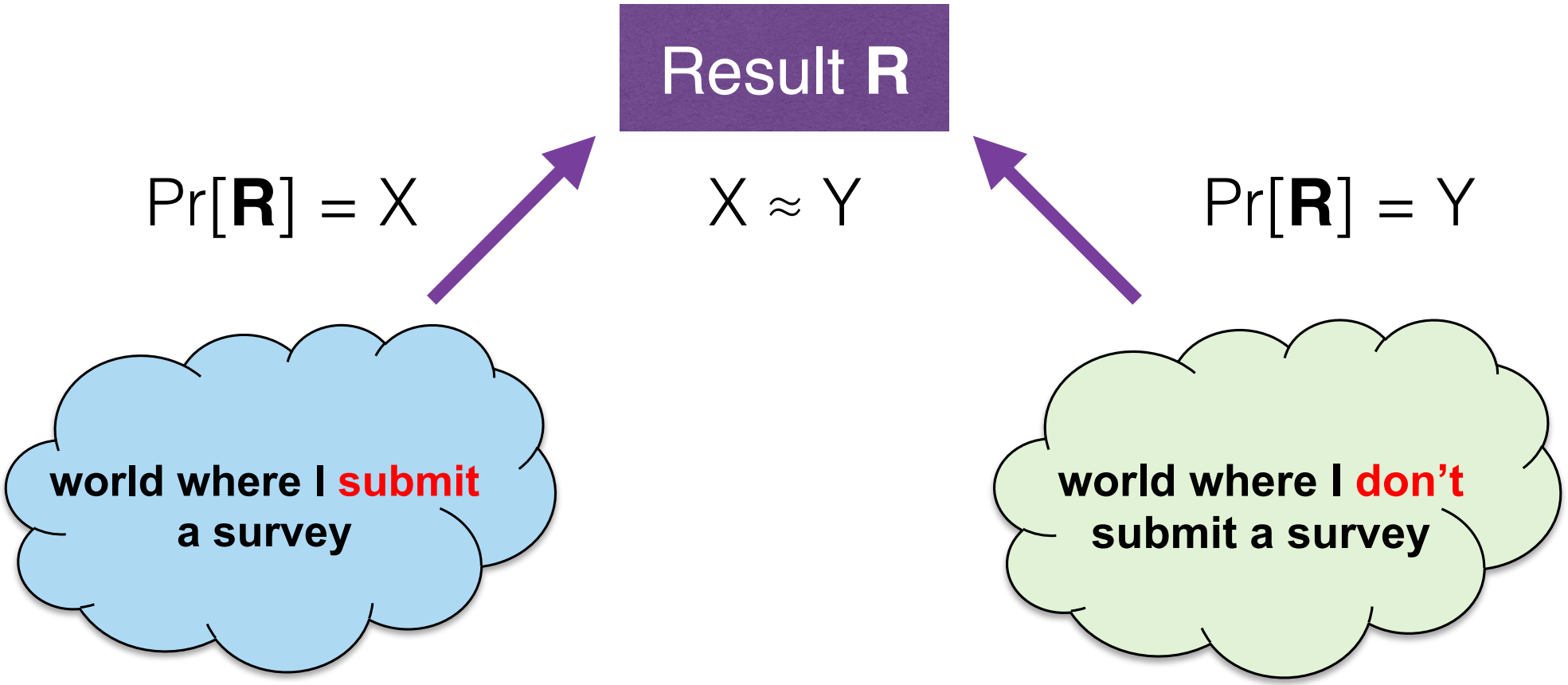
- The chance that the released result will be ***R*** is nearly the same, **regardless** of whether I submit a survey
- There is no (well, *almost* no) additional harm from submitting the survey

Differential privacy

$$\frac{\Pr[Q(D_I) = R]}{\Pr[Q(D_{I \pm i}) = R]} \leq A, \qquad \text{for all } I,i,R$$

- If A=1, there is 0 utility (individuals have no effect)
- If A >> 1, there is little privacy
- A should be chosen by collector to be close to 1

What this means



- Probability of result is nearly the same, regardless of whether I submit a survey
- How can anyone guess which world is true?

What this **doesn't** mean

- Attacker can't learn anything about me from the results (protection from all harms)
- NOPE — Background information still applies
  - Attacker can use *aggregate* results
- Data privacy vs. *personal* privacy

So how does this work?

- A “converse” to the Dinur-Nissim result is that adding *some* (carefully-generated) noise, and limiting the number of queries, *can* be proven to achieve privacy

- (Under this definition of privacy)

Achieving diff. privacy

- E.g., answer SUM(salary, S) with
  SUM(salary, S) + noise,

- Magnitude of the noise depends on the range of plausible salaries (but not on |S|!)
- Automatically handles multiple (arbitrary) queries,
- Privacy degrades as more queries are made
- Gives formal guarantees