Revealing Information while Preserving Privacy

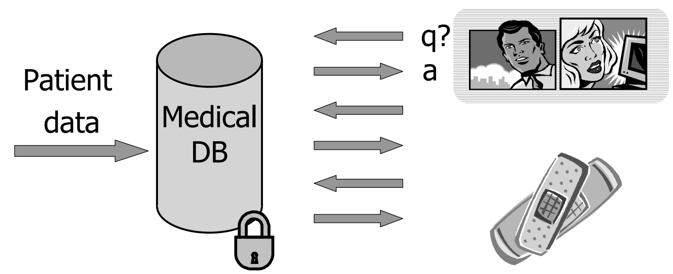
Kobbi Nissim

NEC Labs, DIMACS

Based on work with:

Irit Dinur, Cynthia Dwork and Joe Kilian

The Hospital Story

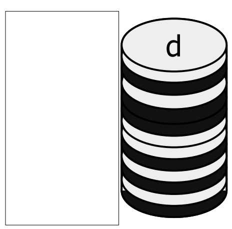


2

A Bad Solution

Idea: a. Remove identifying information (name, SSN, …)

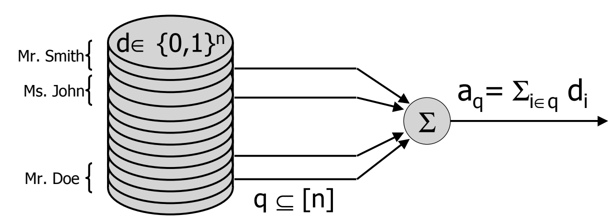
b. Publish data



- Observation: 'harmless' attributes uniquely identify many patients (gender, approx age, approx weight, ethnicity, marital status…)
- Worse: `rare' attribute (CF ≈ 1/3000)

3

Our Model: Statistical Database (SDB)



4

The Privacy Game: Information-Privacy Tradeoff

- Private functions:
  - want to hide $\pi_i(d_1, \ldots, d_n)=d_i$
- Information functions:
  - want to reveal $f_q(d_1, \ldots, d_n)=\Sigma_{i\in q} d_i$
- Explicit definition of private functions
- Crypto: secure function evaluation
  - want to reveal f()
  - want to hide all functions $\pi()$ not computable from f()
  - Implicit definition of private functions

5

Approaches to SDB Privacy [AW 89]

- Query Restriction
  - Require queries to obey some structure
- Perturbation
  - Give `noisy' or `approximate' answers

} This talk

6

Perturbation

- Database: $d = d_1,\ldots,d_n$
- Query: $q \subseteq [n]$
- Exact answer: $a_q = \Sigma_{i\in q} d_i$
- Perturbed answer: $\hat{a}_q$

Perturbation E:
For all q: $|\hat{a}_q - a_q| \leq E$

General Perturbation:
$\Pr_q[|\hat{a}_q - a_q| \leq E] = 1\text{-neg}(n)$
$= 99\%, 51\%$

7

Perturbation Techniques [AW89]

Data perturbation:

- Swapping [Reiss 84][Liew, Choi, Liew 85]
- Fixed perturbations [Traub, Yemini, Wozniakowski 84] [Agrawal, Srikant 00] [Agrawal, Aggarwal 01]
  - Additive perturbation $d'_i=d_i+E_i$

Output perturbation:

- Random sample queries [Denning 80]
  - Sample drawn from query set
- Varying perturbations [Beck 80]
  - Perturbation variance grows with number of queries
- Rounding [Achugbue, Chin 79] Randomized [Fellegi, Phillips 74] …

8

Privacy from $\approx\sqrt{n}$ Perturbation

(an example of a useless database)

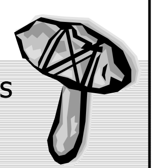
- Database: $d \in_R \{0,1\}^n$
- On query q:
  1. Let $a_q = \Sigma_{i \in q} d_i$
  2. If $|a_q - |q|/2| > E$ return $\hat{a}_q = a_q$
  3. Otherwise return $\hat{a}_q = |q|/2$
- Privacy is preserved
  - If $E \cong \sqrt{n}\,(\lg n)^2$, whp always
    - No information about d i
- No usability!

Can we do better?

- Smaller E ?
- Usability ???

(not) Defining Privacy

- Elusive definition
  - Application dependent
  - Partial vs. exact compromise
  - Prior knowledge, how to model it?
  - Other issues …
- Instead of defining privacy: What is surely non-private…
  - Strong breaking of privacy



11

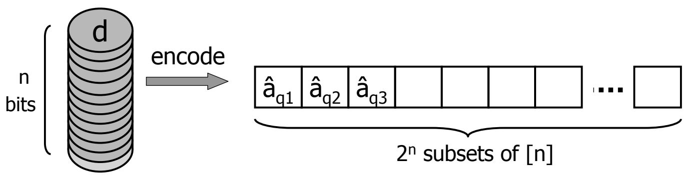
The Useless Database Achieves Best Possible Perturbation: Perturbation << $\sqrt{n}$ Implies no Privacy!

- Main Theorem:
  Given a DB response algorithm with perturbation $E << \sqrt{n}$, there is a poly-time reconstruction algorithm that outputs a database d', s.t. dist(d,d') < o(n).

Strong Breaking of Privacy

12

The Adversary as a Decoding Algorithm



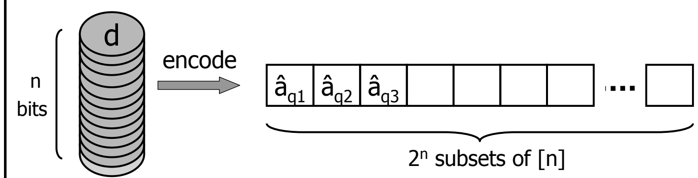
(Recall $\hat{a}_q = \Sigma_{i \in q} d_i + pert_q$)

Decoding Problem: Given access to $\hat{a}_{q1},\ldots, \hat{a}_{q_{2^n}}$ reconstruct d' in time poly(n).

13

Goldreich-Levin Hardcore Bit

Side remark



Where $\hat{a}_q = \Sigma_{i \in q} d_i$ **mod 2** on 51% of the subsets

The GL Algorithm finds in time poly(n) a small list of candidates, containing d

14

Comparing the Tasks

Side remark

Encoding:	$a_q = \Sigma_{i \in q} d_i$ (mod 2)	$a_q = \Sigma_{i \in q} d_i$
Noise:	Corrupt ½-ε of the queries	Additive perturbation ε fraction of the queries deviate from perturbation
Queries:	Dependent	Random
Decoding:	List decoding	d' s.t. dist(d,d') < εn (List decoding impossible)

15

Recall Our Goal: Perturbation << √n Implies no Privacy!

- Main Theorem:
  Given a DB response algorithm with perturbation $E < \sqrt{n}$, there is a poly-time reconstruction algorithm that outputs a database d', s.t. dist(d,d') < o(n).

16

Proof of Main Theorem
The Adversary Reconstruction Algorithm

- Query phase: Get $\hat{a}_{q_j}$ for t random subsets $q_1,\ldots,q_t$ of $[n]$
- Weeding phase: Solve the Linear Program:

$$0 \le x_i \le 1$$
$$\left|\Sigma_{i\in q_j} x_i - \hat{a}_{q_j}\right| \le E$$

- Rounding: Let $c_i = \text{round}(x_i)$, output c

Observation: An LP solution always exists, e.g. x=d.

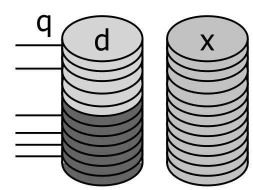
17

Proof of Main Theorem
Correctness of the Algorithm

Consider x=(0.5,…,0.5) as a solution for the LP

Observation: A random q often shows a $\sqrt{n}$ advantage either to 0's or to 1's.

- Such a q disqualifies x as a solution for the LP
- We prove that if dist(x,d) > $\varepsilon \cdot n$, then whp there will be a q among $q_1,\ldots,q_t$ that disqualifies x



18

Extensions of the Main Theorem

- `Imperfect' perturbation:
  - Can approximate the original bit string even if database answer is within perturbation only for 99% of the queries
- Other information functions:
  - Given access to "noisy majority" of subsets we can approximate the original bit-string.

19

Notes on Impossibility Results

- Exponential Adversary:
  - Strong breaking of privacy if E << n
- Polynomial Adversary:
  - Non-adaptive queries
  - Oblivious of perturbation method and database distribution
  - Tight threshold $E \cong \sqrt{n}$
- What if adversary is more restricted?

20

Bounded Adversary Model

- Database: $d \in_R \{0,1\}^n$
- Theorem: If the number of queries is bounded by T, then there is a DB response algorithm with perturbation of $\sim\sqrt{T}$ that maintains privacy.

  With a reasonable definition of privacy

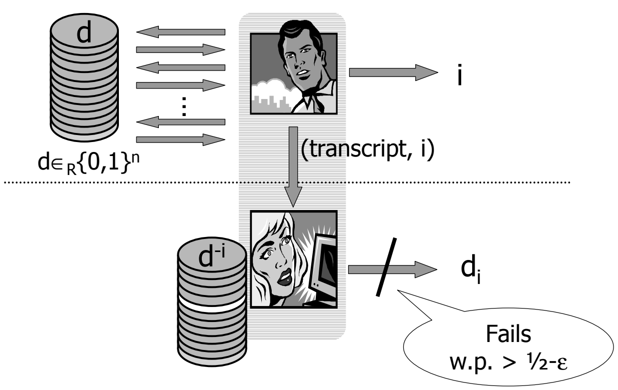
21

Summary and Open Questions

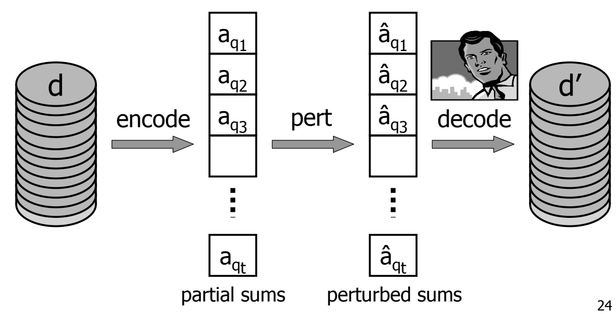
- Very high perturbation is needed for privacy
  - Threshold phenomenon – above $\sqrt{n}$: total privacy, below $\sqrt{n}$: none (poly-time adversary)
  - Rules out many currently proposed solutions for SDB privacy
  - Q: what's on the threshold? Usability?
- Main tool: A reconstruction algorithm
  - Reconstructing an n-bit string from perturbed partial sums/thresholds
- Privacy for a T-bounded adversary with a random database
  - $\sqrt{T}$ perturbation
  - Q: other database distributions
- Q: Crypto and SDB privacy?

22

Our Privacy Definition (bounded adversary model)



The Adversary as a Decoding Algorithm



24