Diffix: High Utility Database Anonymization

Paul Francis, Reinhard Munz

MPI-SWS

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Sebastian Probst Eide

Aircloak

aircloak

Background

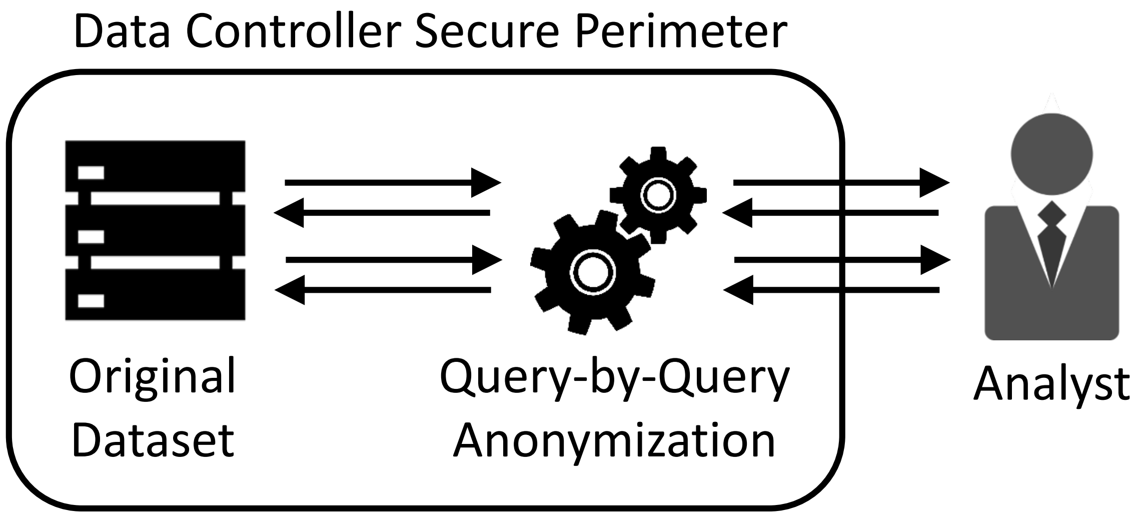
- Researcher at MPI-SWS, co-founder of Aircloak
- Designed and built *Diffix*, a database anonymization technology that provides:
  - Remarkably good analytic utility
  - Easy configuration
  - Very strong anonymity

Background

- Researcher at MPI-SWS, co-founder of Aircloak
- Designed and built *Diffix*, a database anonymization technology that provides:
    - Remarkably good analytic utility
    - Easy configuration
    - Very strong anonymity

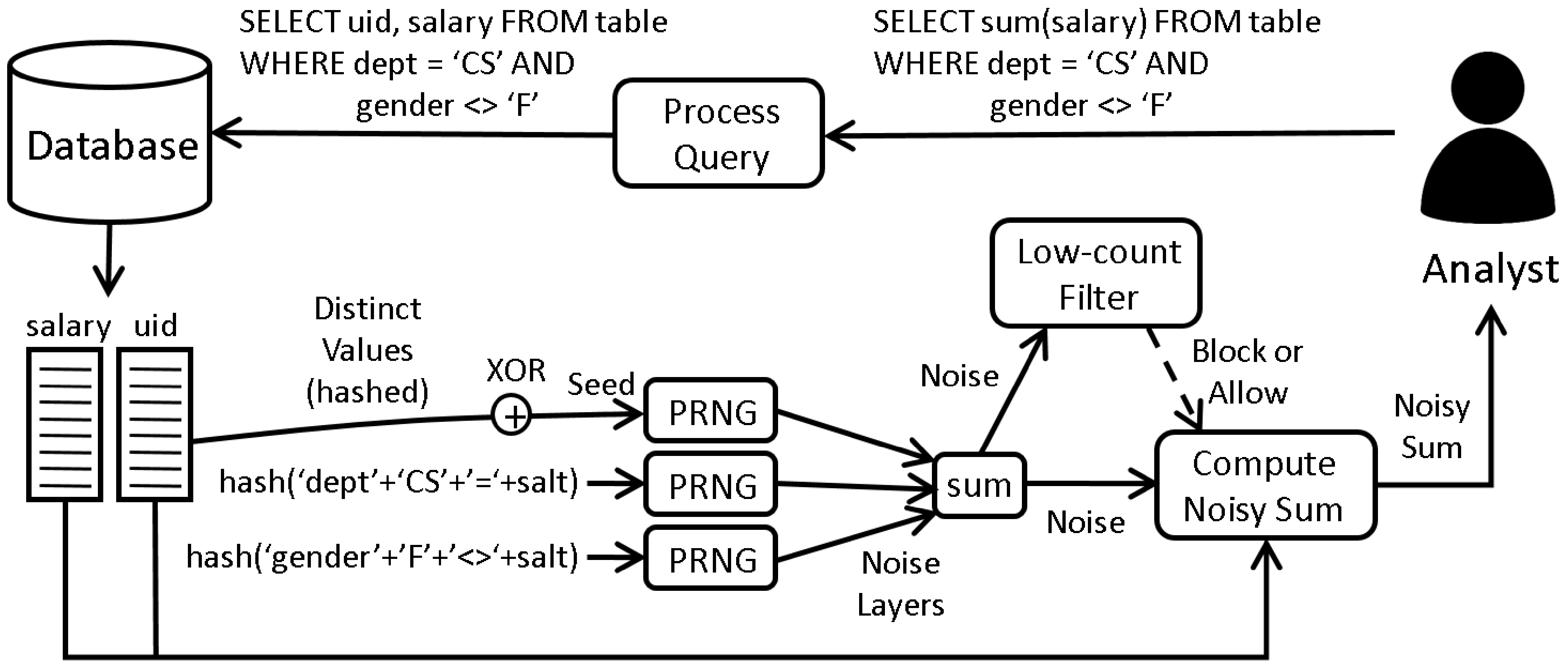
- In my opinion....

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS


Data Controller Secure Perimeter
Original Dataset
Query-by-Query Anonymization
Analyst

Demo....

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS


SELECT uid, salary FROM table
WHERE dept = 'CS' AND
gender <> 'F'
SELECT sum(salary) FROM table
WHERE dept = 'CS' AND
gender <> 'F'
Database
Process Query
Analyst
salary
uid
Distinct Values (hashed)
XOR
Seed
PRNG
hash('dept'+'CS'+'='+salt)
PRNG
hash('gender'+'F'+'<>'+salt)
PRNG
Noise
Low-count Filter
Block or Allow
sum
Noise
Noise Layers
Compute Noisy Sum
Noisy Sum

Diffix value proposition

- Anonymized data is **not** personal data (GDPR)
  - Does not have to be protected as personal data
- But, process for getting DPA approval for an anonymity solution:
  - Takes a long time
  - Can be expensive
  - Applies only narrowly (given data for given use case)
  - Might not even be right! (approved solution is not really anonymous)

- Diffix promises *“instant compliance”* for anonymity

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

My problem…

- Who has the authority and ability to declare Diffix to be anonymous?
    - For all use cases!
- What if they are wrong?
    - (I’ve heard stories that suggest that DPAs are often “wrong” about anonymity)
- How do we establish and maintain confidence in the technology?

What we’ve done so far

- Extensive evaluation with CNIL
  - French national data protection authority
  - Very competent in anonymization technologies (Amandine Jambert, Vincent Toubianis)
- Submitted 60 pages of documentation
  - How Diffix works
  - “Best effort” catalogue of attacks and defenses
- Used anonymity criteria in WP29 opinion on anonymity
- CNIL “thought about it”, doesn’t see a problem
  - Is composing a letter to that effect

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Issues

- CNIL (and us) might be missing something
  - Unfortunately, Diffix is complex, no formal model
- Diffix has evolved since CNIL did the evaluation
  - What is in the field now is not exactly what CNIL evaluated
- CNIL cannot possibly approve every change we make
- Conclusion:
  - No authority currently exists that can adequately evaluate Diffix

Our idea moving forward

- Open Diffix to attack by the public
- "Challenge" system
    - Bug bounty (cash prize for breaking anonymity)
    - Capture the flag (prestige among tech community)

Challenge system

- Provide various types of datasets (geo-location, census, banking, ...)
    - Let attacker provide their own datasets, but should be real data
- Attacker has more or less “background knowledge” of the dataset
    - Complete knowledge of many columns and many rows
    - Detailed but incomplete knowledge of many users
- Attacker gets unlimited number of queries
- Attacker makes statements of the following form:
    - There is a single user with attributes A, B, and C ...
    - “Singling out” (WP29)

Challenge system

- Attacker is more successful when:
  - Attacker statements have higher confidence of being correct
  - Attacker makes statements about more users
  - Attacker requires less background knowledge
- Prize value corresponds with level of success

Feedback and questions?

francis@mpi-sws.org

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

aircloak

```
SELECT count(DISTINCT uid), count_noise(DISTINCT uid),
       avg(age), avg_noise(age),
       stddev(age), stddev_noise(age),
       min(age), max(age), median(age)
FROM   (
        SELECT patients.id AS uid,
               2017 - year(date_of_birth) AS age
        FROM patients
       ) t
```

```
SELECT 2017 - year(date_of_birth) AS age,
       count(DISTINCT patients.id),
       count_noise(DISTINCT patients.id)
FROM patients
GROUP BY age
ORDER BY age
```

```sql
SELECT din, extract_match(name, '^\w+') AS drugName,
      count(*), count_noise(*) FROM (
  SELECT DISTINCT t1.patient_id, visit_id, din, name FROM
   (SELECT patient_id, visit_id, din, name
    FROM pt_prescriptions) t1
  JOIN
   (SELECT distinct patient_id
    FROM pt_prescriptions
    WHERE din = '02303671') t2
  ON t1.patient_id = t2.patient_id
  ) t
GROUP BY din, drugName
ORDER BY count(*) desc
```

```sql
SELECT din, name,
      count(*), count_noise(*) FROM (
  SELECT DISTINCT t1.patient_id, visit_id, din, name FROM
   (SELECT patient_id, visit_id, din, name
    FROM pt_prescriptions) t1
  JOIN
   (SELECT distinct patient_id
    FROM pt_prescriptions
    WHERE din = '02303671') t2
  ON t1.patient_id = t2.patient_id
  ) t
GROUP BY din, name
ORDER BY count(*) desc
```