Jane Doe will help improve your project

Presentation

- Rebeca Sarai 👩‍💻
- Recife, Brazil ☀️🌊🏖️🏝️🇧🇷
- Computer Engineering by University of Pernambuco 👩‍🎓

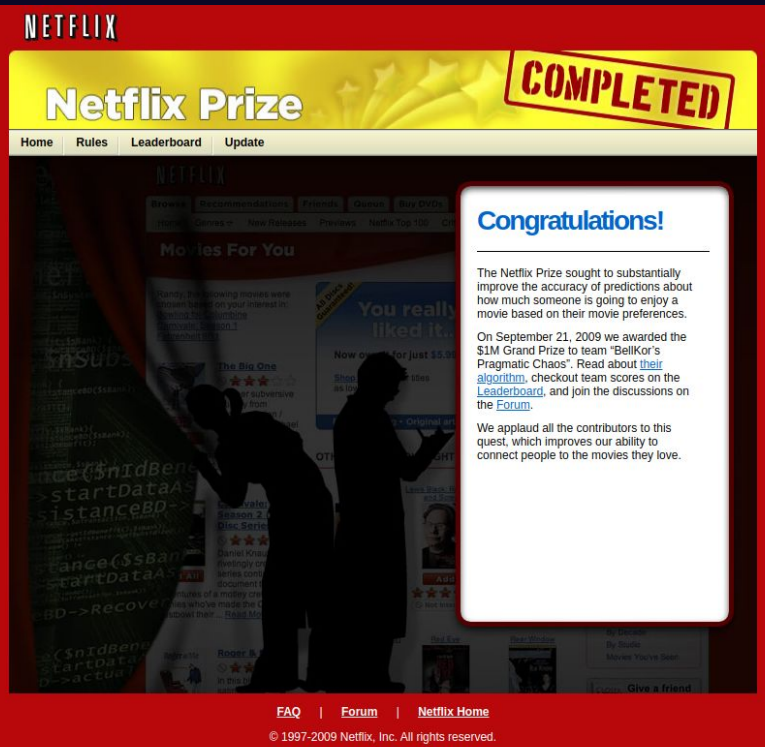

VINTA
We're a **team of experts** from Brazil.
We help our clients **evolve their products** with **top notch** development and UX techniques.
Get to know us: vintasoftware.com

Comunicação Náutico

ão Náutico

Jane Doe will help improve your project

Beyond replacing names...



Netflix Prize

BRUCE SCHNEIER SECURITY 12.12.07 09:00 PM

WHY 'ANONYMOUS' DATA SOMETIMES ISN'T

LAST YEAR, NETFLIX published 10 million movie rankings by 500,000 customers, as part of a challenge for people to come up with better recommendation systems than the one the company was using. The data was anonymized by removing personal details and replacing names with random numbers, to protect the privacy of the recommenders.

Arvind Narayanan and Vitaly Shmatikov, researchers at the University of Texas at Austin, de-anonymized some ofthe Netflix data by comparing rankings and timestamps with public information in the Internet Movie Database, or IMDb.

Their research (.pdf) illustrates some inherent security problems with anonymous data, but first it's important to explain what they did and did not do.

They did *not* reverse the anonymity of the entire Netflix dataset. What they did was reverse the anonymity of the Netflix dataset for those sampled users who also entered some movie rankings, under their own names, in the IMDb. (While IMDb's records are public, crawling the site to get them is against the IMDb's terms of service, so the

WIRED

Fitness tracking app Strava gives away location of secret US army bases

Data about exercise routes shared online by soldiers can be used to pinpoint overseas facilities

- **Latest: Strava suggests military users 'opt out' of heatmap as row deepens**

▲ A military base in Helmand Province, Afghanistan with route taken by joggers highlighted by Strava. Photograph: Strava Heatmap

Sensitive information about the location and staffing of military bases and spy outposts around the world has been revealed by a fitness tracking company.

The Guardian

Wait, is anonymization even possible at all?


Your Data Were 'Anonymized'? These Scientists Can Still Identify You
Computer scientists have developed an algorithm that can pick out almost any American in databases supposedly stripped of personal information.

The Times

Researchers spotlight the lie of 'anonymous' data

Natasha Lomas @riptari / 7:30 am -03 • July 24, 2019

Comment

Researchers from two universities in Europe have published a method they say is able to correctly re-identify 99.98% of individuals in anonymized data sets with just 15 demographic attributes.

TechCrunch

The answer is yes.

Aircloak Blog

Outline

- Regulations and Privacy by Design
- Pseudonymization (real quick)
- Anonymization
  - k-Anonymity
  - Differential Privacy

Photo by Gia Oris on Unsplash

EU General Data Protection Regulation

Privacy by Design

CRPD Article 25

Pseudonymization

Pseudonymization

Pseudonymized data is still considered Personal Data, since it can be used for **re-identification** of the data subject, if combined with **additional information**.

Pseudonymization

- Data Masking
- Approximation
- Encryption
- Tokenization

```python
class AbstractUser(AbstractBaseUser, PermissionsMixin):
    username_validator = UnicodeUsernameValidator()
    username = models.CharField(
        _('username'),
        max_length=150,
        unique=True,
        help_text=_('Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.'),
        validators=[username_validator],
        error_messages={
            'unique': _("A user with that username already exists."),
        },
    )
    first_name = models.CharField(_('first name'), max_length=30, blank=True)
    last_name = models.CharField(_('last name'), max_length=150, blank=True)
    email = models.EmailField(_('email address'), blank=True)
```

```
class AbstractUser(Abstra
    username_validator =
    username = models.Cha
        _('username'),
        max_length=150,
        unique=True,
        help_text=_('Requ
        validators=[usern
        error_messages={
            'unique': _('
        },
    )
    first_name = models.C
    last_name = models.Ch
    email = models.Email
```

THE TOKEN OBJECT


```
{
  "id": "tok_1F7Xn02eZvKYlo2C80cAActP",
  "object": "token",
  "card": {
    "id": "card_1F7Xn02eZvKYlo2CfJ7Z3z0x",
    "object": "card",
    "address_city": null,
    "address_country": null,
    "address_line1": null,
    "address_line1_check": null,
    "address_line2": null,
    "address_state": null,
    "address_zip": null,
    "address_zip_check": null,
    "brand": "Visa",
    "country": "US",
    "cvc_check": null,
    "dynamic_last4": null,
    "exp_month": 8,
    "exp_year": 2020,
    "fingerprint": "Xt5EWLLDS7FJjR1c",
    "funding": "credit",
    "last4": "4242",
    "metadata": {},
    "name": null,
    "tokenization_method": null
  },
```

pseudonymization django

FILTER

DjangoCon US 2018 - Pseu, Pseu, Pseudio. Pseudonymization in Django. by Frank Valcarcel

DjangoCon US • 210 views • 9 months ago

DjangoCon US 2018 - Pseu, Pseu, Pseudio. **Pseudonymization** in **Django**. by Frank Valcarcel The General Data Protection ...

CC

Anonymization

Anonymous data **does not contain** information that can **potentially identify an individual** and is not considered Personal Data by GDPR

Approaches to data anonymization

- Static Anonymization
- Dynamic Anonymization
- Synthetic Data

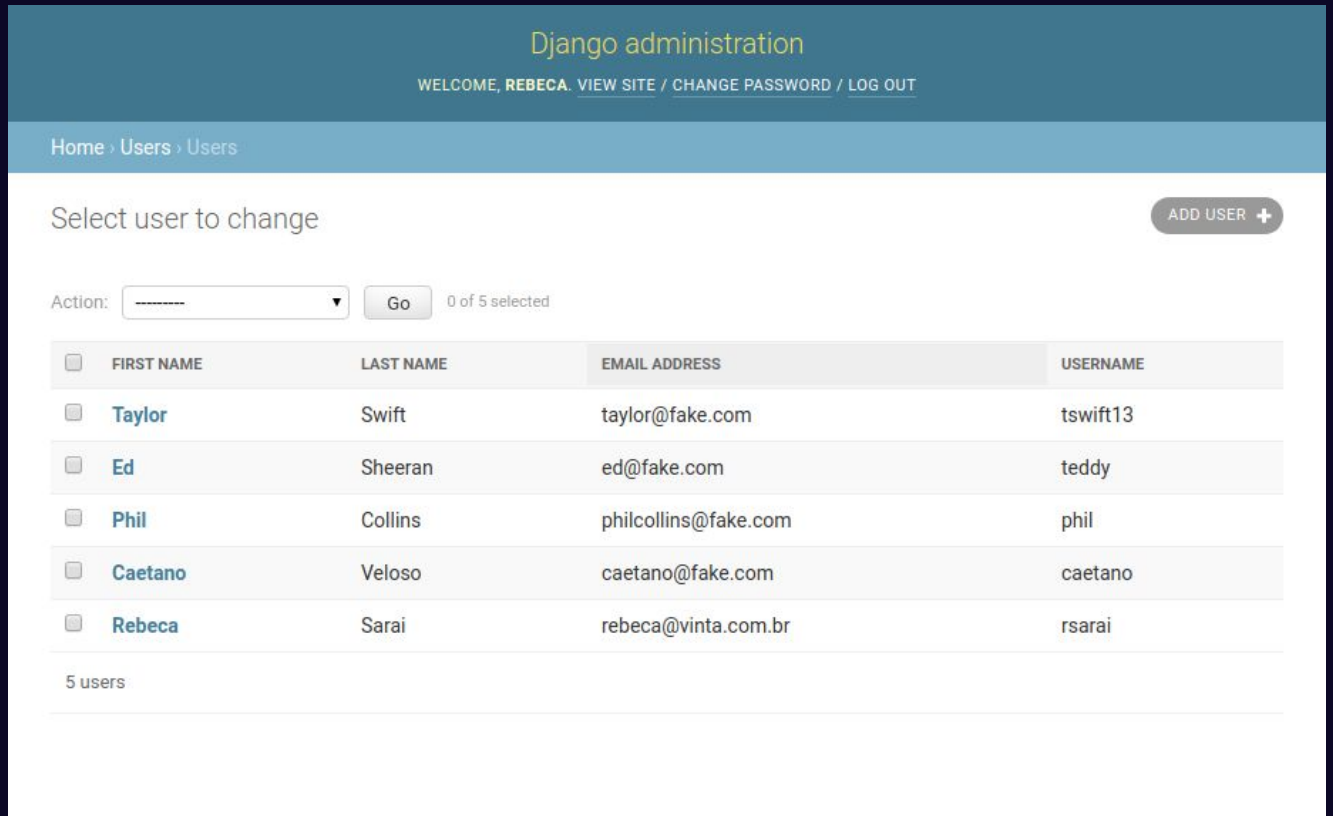
Approaches to data anonymization

- Static Anonymization
- Dynamic Anonymization
- ~~Synthetic Data~~

Database Anonymization

- **Static** Anonymization
- **Destructively** alter data **directly on the database**
- Share data with **third parties**
- **Secure testing** environments

Database Anonymization



Database Anonymization

```
class AbstractUser(AbstractBaseUser, PermissionsMixin):
    username_validator = UnicodeUsernameValidator()
    username = models.CharField(
        _('username'),
        max_length=150,
        unique=True,
        help_text=_('Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.'),
        validators=[username_validator],
        error_messages={
            'unique': _("A user with that username already exists."),
        },
    )
    first_name = models.CharField(_('first name'), max_length=30, blank=True)
    last_name = models.CharField(_('last name'), max_length=150, blank=True)
    email = models.EmailField(_('email address'), blank=True)
    is_staff = models.BooleanField(
        _('staff status'),
        default=False,
        help_text=_('Designates whether the user can log into this admin site.'),
    )
    is_active = models.BooleanField(
        _('active'),
        default=True,
        help_text=_(
            'Designates whether this user should be treated as active. '
            'Unselect this instead of deleting accounts.'
        ),
    )
    date_joined = models.DateTimeField(_('date joined'), default=timezone.now)
```

Database Anonymization

```python
from dj_anonymizer import anonym_field
from dj_anonymizer.register_models import AnonymBase, register_anonym, register_skip
from faker import Factory
fake = Factory.create()


class UserAnonym(AnonymBase):
    email = anonym_field.string('{seq}@fake.com', seq_callback=datetime.datetime.now)
    username = anonym_field.string('username_{seq}@fake.com', seq_callback=datetime.datetime.now)
    first_name = anonym_field.function(fake.first_name)
    last_name = anonym_field.function((fake.last_name))
    password = anonym_field.password('password')
    is_staff = anonym_field.function(lambda: False)
    ssn = anonym_field.function(fake.ssn)

    class Meta:
        queryset = User.objects.exclude(id=1)
        exclude_fields = ['is_active', 'is_superuser', 'last_login', 'date_joined',
                          'avatar', 'phone_number', 'birth_date', 'bio']
register_anonym([
    (User, UserAnonym),
])
register_skip([
    ContentType, Group, Permission, LogEntry, Session,
])
```

Database Anonymization

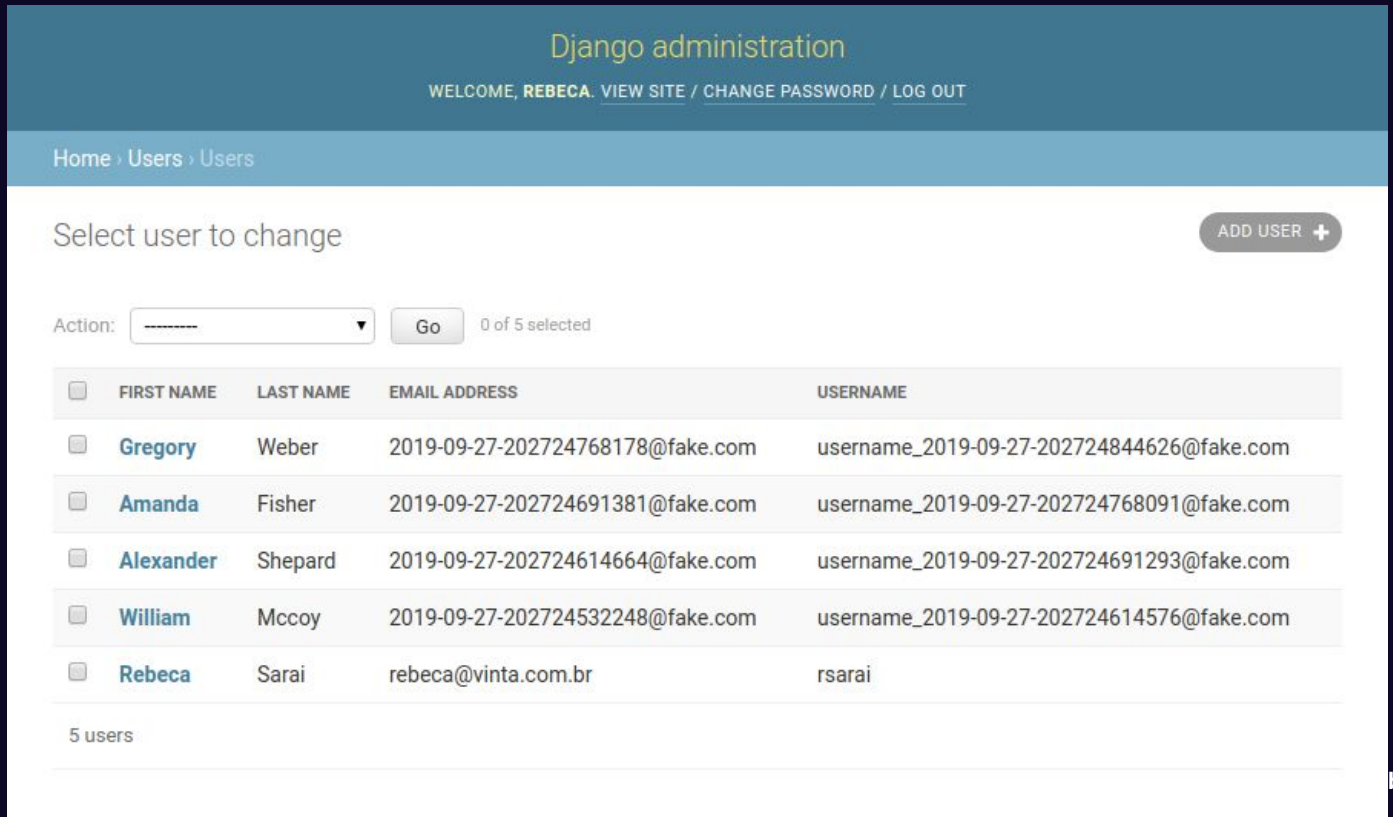
Database Anonymization

```
✔ 21:02:57 rsarai:~/github-projects/anonymization/data-privacy/jane_doe_project (master) [2 days ago] ⤾
$ ls
anonymizer/  anonymizingdb  db.sqlite3  jane_doe_project/  janedoe  janedoeproject  manage.py*  users/
✔ 21:02:59 rsarai:~/github-projects/anonymization/data-privacy/jane_doe_project (master) [2 days ago] ⤾
$ python manage.py anonymize_db --action anonymize
Updating started

Generating fake values for model "User"


Updating finished
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Total time (sec.): 0.3125114440917969
✔ 21:03:12 rsarai:~/github-projects/anonymization/data-privacy/jane_doe_project (master) [2 days ago] ⤾
$ exit
exit
█
```

Database Anonymization



Database Static Anonymization

- dj_anonymizer
- Keeps the **data structure**
- **Performance** issues
- Anonymization must be **precisely defined**
- **Background knowledge** attacks is a risk
- Data is often only pseudonymised

k-Anonymity

- 1° method proposed for microdata anonymization
- Creates **groups** with at least k records sharing the same **quasi-identifiers** values.
- Generalization and Suppression

k-Anonymity: Behavior

Single	20

	ID	QIDs			SA
Tuple#	Name	Marital Stat	Age	ZIP Code	Crime
1	Joe	Separated	29	32042	Murder
2	Jill	Single	20	32021	Theft
3	Sue	Widowed	24	32024	Traffic
4	Abe	Separated	28	32046	Assault
5	Bob	Widowed	25	32045	Piracy
6	Amy	Single	23	32027	Indecency

k-Anonymity: Behavior

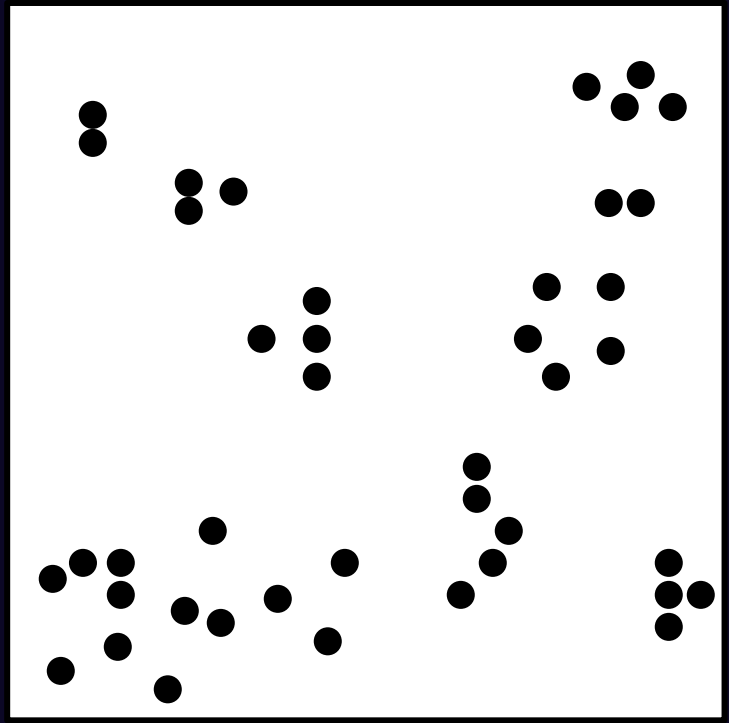
Single	20

Jill committed a theft

	ID	QIDs			SA
Tuple#	Name	Marital Stat	Age	ZIP Code	Crime
1	Joe	Separated	29	32042	Murder
2	Jill	Single	20	32021	Theft
3	Sue	Widowed	24	32024	Traffic
4	Abe	Separated	28	32046	Assault
5	Bob	Widowed	25	32045	Piracy
6	Amy	Single	23	32027	Indecency

k-Anonymity: Behavior

- Using Mondrian implementation
- **Partitions** the domain space recursively into several regions



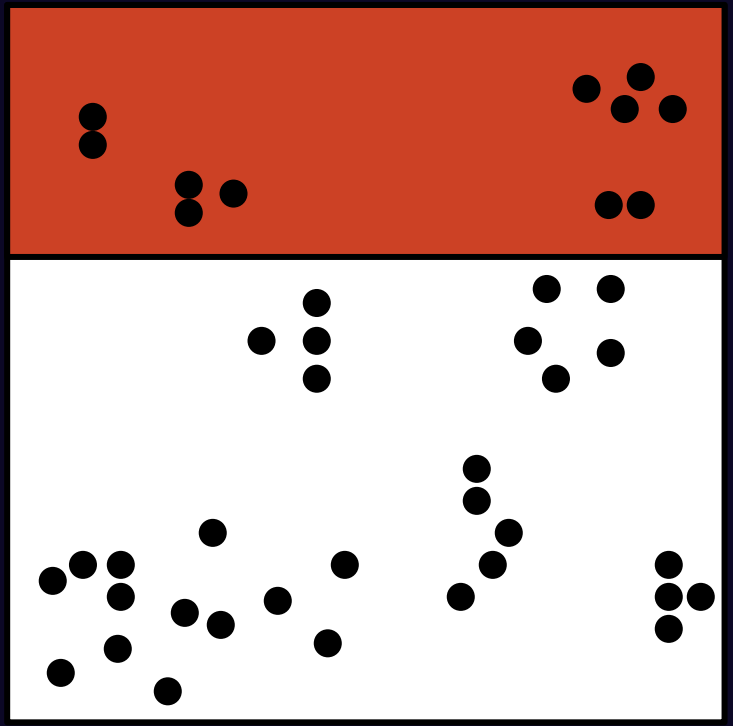
k-Anonymity: Behavior

- Using Mondrian implementation
- **Partitions** the domain space recursively into several regions

	ID	QIDs			SA
Tuple#	Name	Marital Stat	Age	ZIP Code	Crime
1	Joe	Separated	29	32042	Murder
2	Jill	Single	20	32021	Theft
3	Sue	Widowed	24	32024	Traffic
4	Abe	Separated	28	32046	Assault
5	Bob	Widowed	25	32045	Piracy
6	Amy	Single	23	32027	Indecency

k-Anonymity: Behavior

- Using Mondrian implementation
- **Partitions** the domain space recursively into several regions



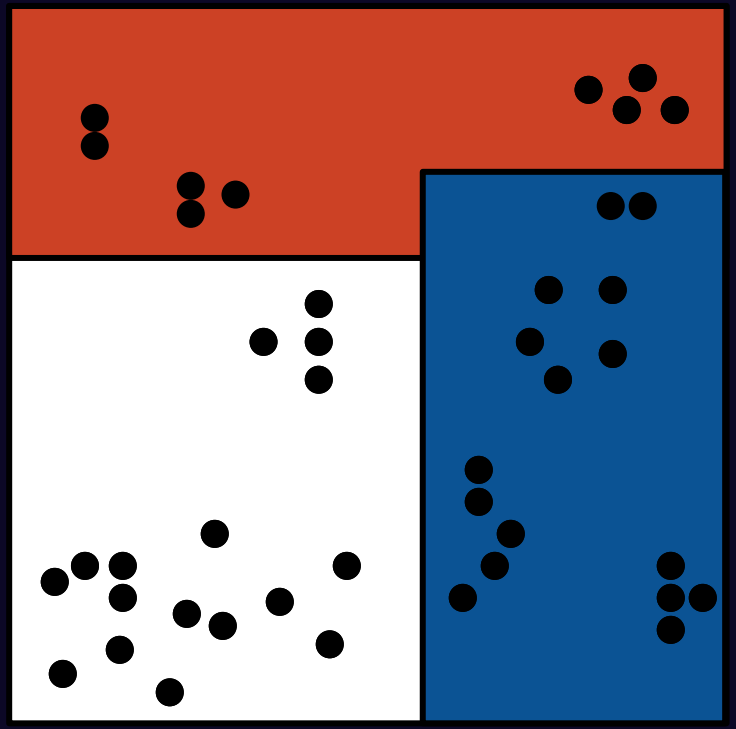
k-Anonymity: Behavior

- Using Mondrian implementation
- **Partitions** the domain space recursively into several regions

	ID	QIDs			SA
Tuple#	Name	Marital Stat	Age	ZIP Code	Crime
1	Joe	Separated	29	32042	Murder
2	Jill	Single	20	32021	Theft
3	Sue	Widowed	24	32024	Traffic
4	Abe	Separated	28	32046	Assault
5	Bob	Widowed	25	32045	Piracy
6	Amy	Single	23	32027	Indecency

k-Anonymity: Behavior

- Using Mondrian implementation
- **Partitions** the domain space recursively into several regions



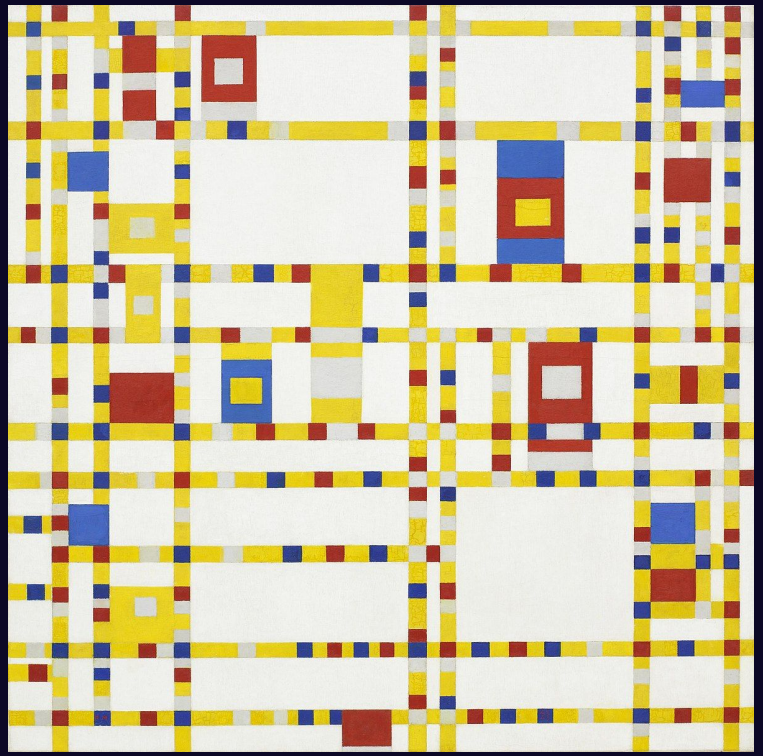
k-Anonymity: Behavior

- Using Mondrian implementation
- **Partitions** the domain space recursively into several regions

		QIDs		Non-SA	SA
Tuple#	EQ	Marital Stat	Age	ZIP Code	Crime
1	1	Single,Separated	(23-30)	32042	Murder
4		Single,Separated	(23-30)	32046	Assault
2	2	Single,Separated	[20-23]	32021	Theft
6		Single,Separated	[20-23]	32027	Indecency
3	3	Div.,Wid.,Married,Remarried	[20-30)	32024	Traffic
5		Div.,Wid.,Married,Remarried	[20-30)	32045	Piracy

k-Anonymity

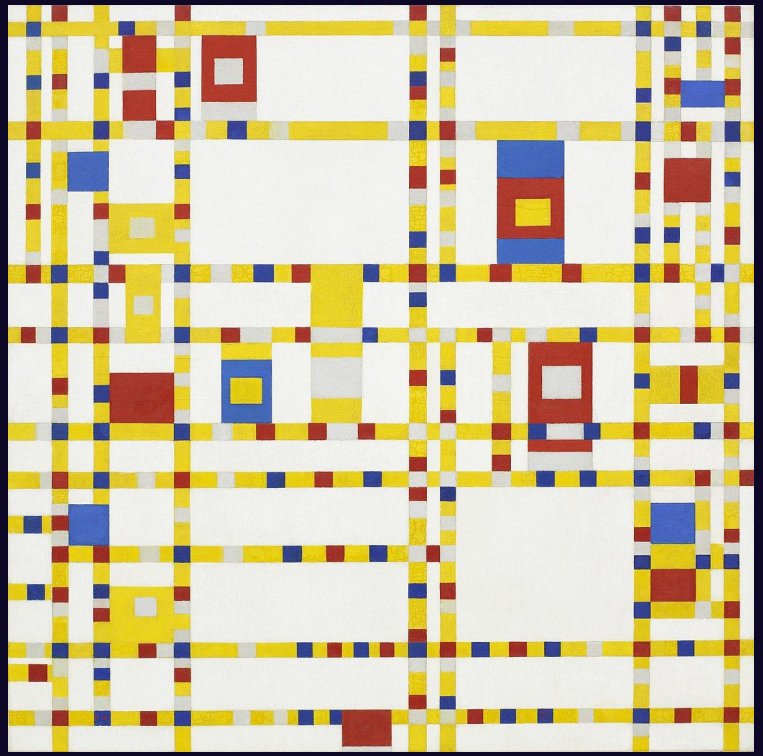
- Provides protection against **identity disclosure**
- Does **not** prevent **attribute disclosure**
- In case of multiple releases **coordination is required**
- **Background knowledge** attacks is a risk



Broadway Boogie-Woogie

Refinements of the k-Anonymity

- *l*-diversity
- *t*-closeness
- ß-likeness
- Require **variability** in the sensitive attributes



Broadway Boogie-Woogie

k-Anonymity use cases

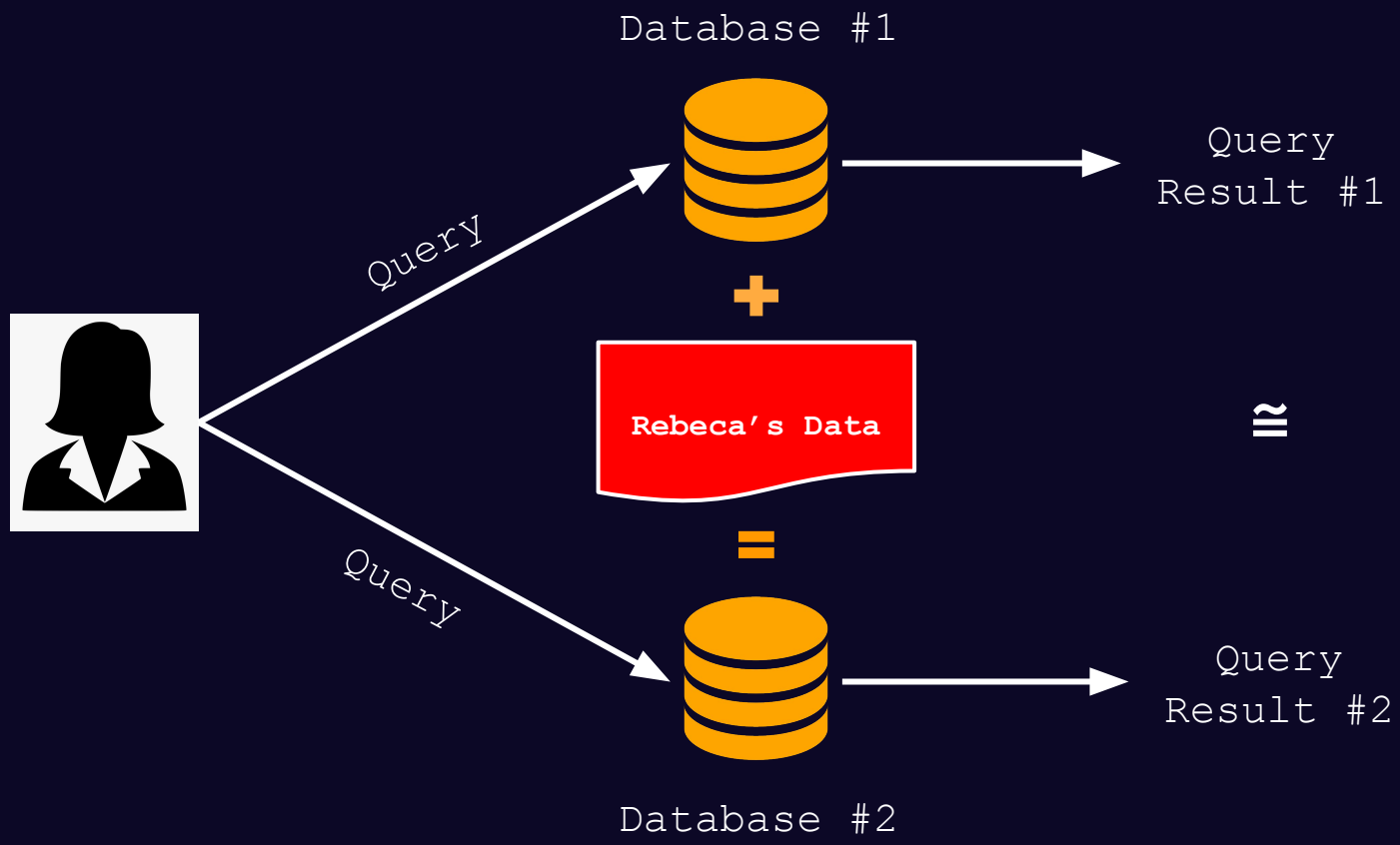
- Haveibeenpwned
  - Validating Leaked Passwords with k-Anonymity
  - Cloudflare, Privacy and k-Anonymity
- Okta's PassProtect
- Your project
  - ARX (https://github.com/arx-deidentifier/arx)

k-Anonymity use cases

- Haveibeenpwned
  - Validating Leaked Passwords with k-Anonymity
  - Cloudflare, Privacy and k-Anonymity
- Okta's PassProtect
- Your project
  - ARX (https://github.com/arx-deidentifier/arx)

Differential Privacy

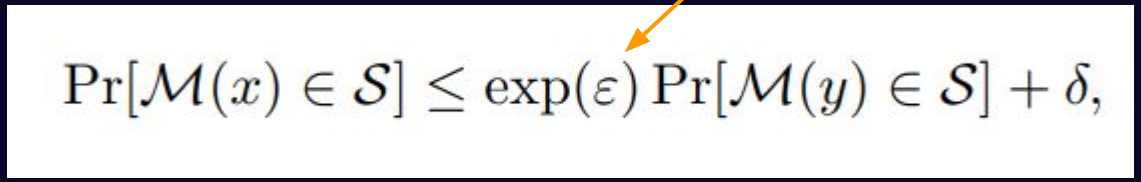
- Privacy-preserving data analysis
- Differential privacy is **not an algorithm**
- Differential privacy is a **formal notion of privacy**
- Learning the teachings of the database
- Without learning about individuals in the database


Database #1
Query
Query Result #1
Rebeca's Data
≅
Query
Query Result #2
Database #2

Differential Privacy

- **Plausible deniability** on individual presence in a database

e raised to the **epsilon** power

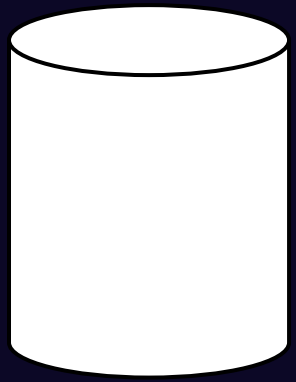


$\mathcal{M}$ is a randomized mechanism that gives ε-differential privacy for all data sets

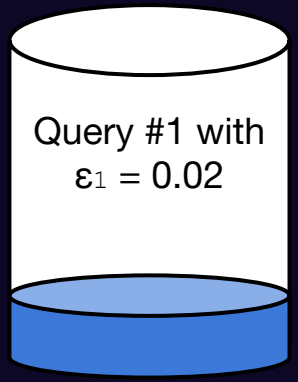
Differential Privacy

- Measure of **privacy loss** ε (**privacy budget**)
- Tune the "amount of privacy"

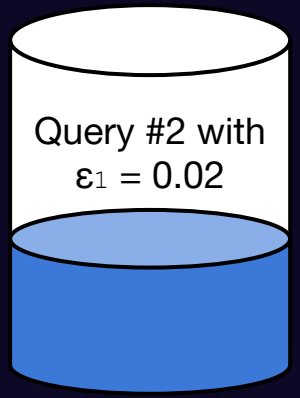
Budget ε = 0.1



One-query with $\varepsilon_1 = 0.02$



Multiple-queries



Challenges of Differential Privacy

- Usability for **non-experts**
- **Support** for analytics queries (allow people do the same queries they do today with SQL)

Mechanisms

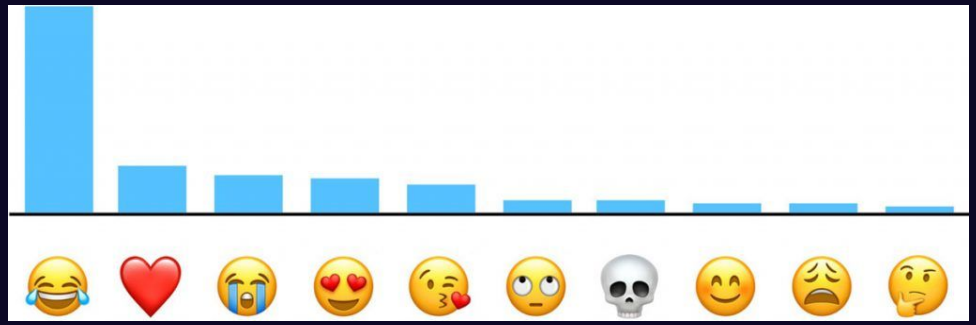
Mechanism	Strengths
Laplace Mechanism	Simple counts
PINQ	Counting and histogram queries
Elastic Sensitivity	Queries with joins
Sample & Aggregate	Statistical Estimators
Restricted Sensitivity	Graph analysis

Challenges of Differential Privacy

- Usability for **non-experts**
- **Support** for analytics queries (allow people do not the same queries they do today with SQL)
- **Integration** with existing data environments
- Sparse **real world** examples
- Fundamental Law of Information Recovery

Usage for Differential Privacy

- Protection against **arbitrary risks**
- **Quantification** of privacy loss
- Promising applications on machine learning
- Used by Microsoft, Google, Apple, Uber, etc



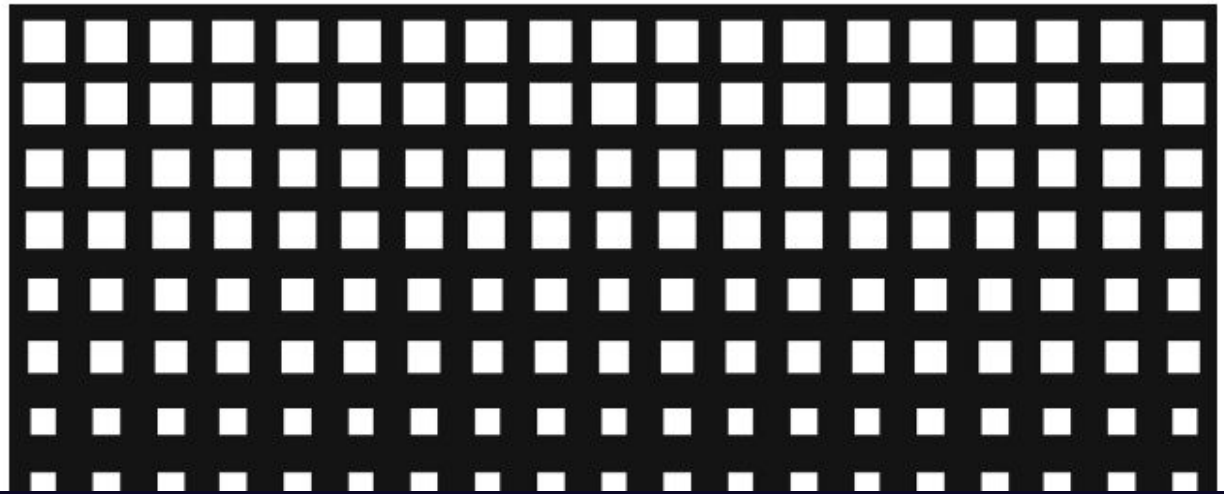
The Mac Observer

WIRED BUSINESS CULTURE GEAR IDEAS SCIENCE SECURITY TRANSPORTATION

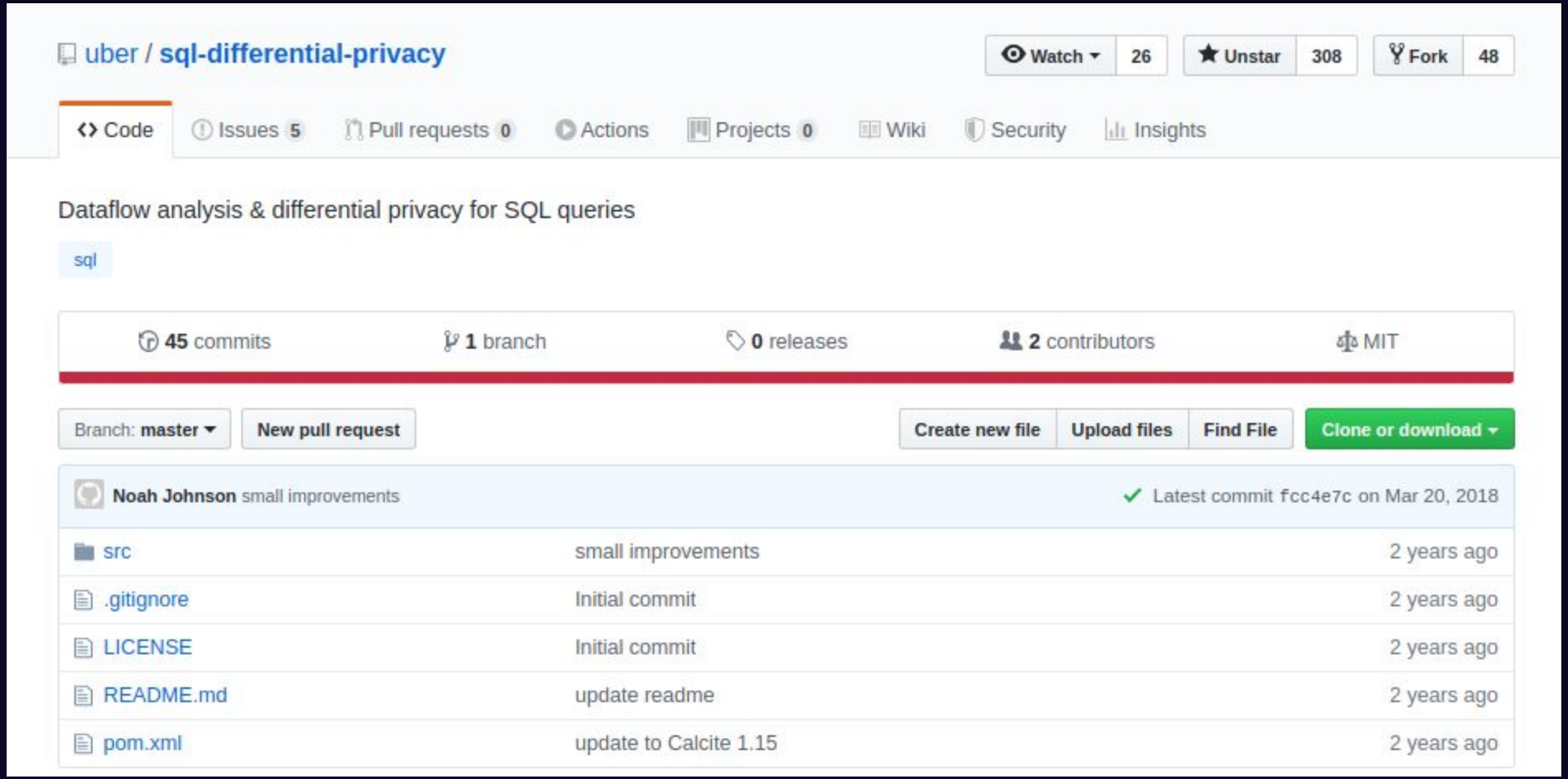
ANDY GREENBERG SECURITY 09.15.2017 09:28 AM

How One of Apple's Key Privacy Safeguards Falls Short

Apple has boasted of its use of a cutting-edge data science known as "differential privacy." Researchers say they're doing it wrong.



WIRED


uber / sql-differential-privacy
Watch 26
Unstar 308
Fork 48
Code
Issues 5
Pull requests 0
Actions
Projects 0
Wiki
Security
Insights
Dataflow analysis & differential privacy for SQL queries
sql
45 commits
1 branch
0 releases
2 contributors
MIT
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
Noah Johnson small improvements
Latest commit fcc4e7c on Mar 20, 2018
src small improvements 2 years ago
.gitignore Initial commit 2 years ago
LICENSE Initial commit 2 years ago
README.md update readme 2 years ago
pom.xml update to Calcite 1.15 2 years ago

uber / **sql-differential-privacy**

Watch 26 | Unstar 308 | Fork 48

Code | Issues 5 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights

Dataflow analysis & differential privacy for SQL queries

sql

frankmcsherry / **blog**

Watch 221 | Star 1,293 | Fork 129

Code | Issues 4 | Pull requests 1 | Projects 0 | Wiki | Security | Insights

Branch: **master** | **blog** / posts / **2018-02-25.md** | Find file | Copy path

Jim Klucar typo fixes — 257215c on Feb 28, 2018

1 contributor

213 lines (114 sloc) | 28 KB | Raw | Blame | History

Uber's differential privacy .. probably isn't

Today we are going to talk through a recently accepted VLDB paper, Toward Practical Differential Privacy for SQL Queries. This paper is partly what is behind Uber's SQL differential privacy project, which they have been happily plugging.

Despite what you might guess, I actually think there is a fair bit to like in the goal of the paper, specifically what is implied by its title, and some of the technical development. The world could use more people aimed at the task of providing differential privacy to people who do not have their own advanced degree in differential privacy, and the framework in this paper is a fine zero-th step.

uber / **sql-differential-privacy**

Watch 26 | Unstar 308 | Fork 48

Code | Issues 5 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights

google / **differential-privacy**

Watch 54 | Unstar 1,238 | Fork 100

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights

No description, website, or topics provided.

4 commits | **1** branch | **0** releases | **0** contributors | Apache-2.0

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find File** | **Clone or download**

Differential Privacy Team and **dasmdasm** Changes: ··· Latest commit `7c89b65` 7 days ago

differential_privacy	Changes:	7 days ago
BUILD	Project import	19 days ago
CONTRIBUTING.md	Project import	19 days ago
LICENSE	Project import	19 days ago
README.md	Fix typo on the landing page, and incorrect citation.	17 days ago

uber / **sql-differential-privacy**

Watch 26 | Unstar 308 | Fork 48

Code | Issues 5 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights

google / **differential-privacy**

Watch 54 | Unstar 1,238 | Fork 100

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights

No description, website, or t

4 commits

Branch: master | New pull re

Differential Privacy Team a

- differential_privacy
- BUILD
- CONTRIBUTING.md
- LICENSE
- README.md

README.md

Anonymous Functions PostgreSQL Extension

This subdirectory contains a PostgreSQL extension providing several epsilon-DP aggregate functions. We w
as the anonymous functions.

Setup

- Install Postgres 11 using the source code.
  - Source: https://www.postgresql.org/ftp/source/
  - Instructions: https://www.postgresql.org/docs/9.3/install-short.html

uber / sql-differe

Code | Issues

google / differen

Code | Issues

No description, websi

4 commits

Branch: master ▾ | Ne

Differential Privacy

differential_privacy

BUILD

CONTRIBUTING.m

LICENSE

README.md

308 | Fork | 48

1,238 | Fork | 100

Extension

DP aggregate functions. We w

sciendo

Proceedings on Privacy Enhancing Technologies ..; .. (..):1–20

Royce J Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson*

Differentially Private SQL with Bounded User Contribution

Abstract: Differential privacy (DP) provides formal guarantees that the output of a database query does not reveal too much information about any individual present in the database. While many differentially private algorithms have been proposed in the scientific literature, there are only a few end-to-end implementations of differentially private query engines. Crucially, existing systems assume that each individual is associated with at most one database record, which is unrealistic in practice. We propose a generic and scalable method to perform differentially private aggregations on databases, even when individuals can each be associated with arbitrarily many rows. We express this method as an operator in relational algebra, and implement it in an SQL engine. To validate this system, we test the utility of typical queries on industry benchmarks, and verify its correctness with a stochastic test framework we developed. We highlight the promises and pitfalls learned when deploying such a system in practice, and we publish its core components as open-source software.

Keywords: differential privacy, database, SQL

DOI Editor to enter DOI
Received ..; revised ..; accepted ...

a population without revealing too much about individuals is a long-standing field of research. The standard definition used in this context is differential privacy (DP): it provides a formal guarantee on how much the output of an algorithm reveals about any individual in its input [10, 11, 14]. Differential privacy states that the distribution of results derived from private data cannot reveal "too much" about a single person's contribution, or lack thereof, to that data [12]. By using differential privacy when analyzing data, organizations can minimize the disclosure risk of sensitive information about their users.

Query engines are a major analysis tool for data scientists, and one of the most common ways for analysts to write queries is with Structured Query Language (SQL). As a result, multiple query engines have been developed to enable data analysis while enforcing DP [2, 21, 26, 33], and all of them use a SQL-like syntax.

However, as we discuss in Section 2, these differentially private query engines make some implicit assumptions, notably that each individual in the underlying database is associated with at most one database record. This does not hold in many real-world datasets, so the privacy guarantee offered by these systems is weaker than advertised for those databases. To overcome this

sciendo
Proceedings on Privacy Enhancing Technologies ..; .. (..):1–20

Royce J Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson*

Differentially Private SQL with Bounded User Contribution

Abstract: Differential pri too much about indi- guarantees that the outp of research. The stan- not reveal too much infor t is differential privacy present in the database. V ntee on how much the vate algorithms have been bout any individual in erature, there are only a privacy states that the tions of differentially priv m private data cannot existing systems assume t person's contribution, ated with at most one da . By using differential realistic in practice. We p ganizations can mini- method to perform differen ive information about databases, even when indiv with arbitrarily many row analysis tool for data an operator in relational al common ways for an- SQL engine. To validate t tructured Query Lan- of typical queries on industry benchmarks, and verify its correctness with a stochastic test framework we developed. We highlight the promises and pitfalls learned when deploying such a system in practice, and we publish its core components as open-source software.

Keywords: differential privacy, database, SQL

DOI Editor to enter DOI
Received ..; revised ..; accepted ...

guage (SQL). As a result, multiple query engines have been developed to enable data analysis while enforcing DP [2, 21, 26, 33], and all of them use a SQL-like syntax.

However, as we discuss in Section 2, these differentially private query engines make some implicit assumptions, notably that each individual in the underlying database is associated with at most one database record. This does not hold in many real-world datasets, so the privacy guarantee offered by these systems is weaker than advertised for those databases. To overcome this

Concerns when anonymizing datasets

- Data cannot be **fully anonymized** and remain **useful**
- **Re-Identification** is not the only risk
- Queries over **large sets** are not protective
- Query **auditing** is problematic

Concerns when anonymizing datasets

- Summary **Statistics** are Not “Safe.”
- Revealing **“ordinary” facts** may be problematic
- Computer security is not privacy protection

Thank you!

Got any questions?

Rebeca Sarai

Software Developer

rebeca@vinta.com.br

@_rebecasarai

/rsarai

Access this talk on **bit.ly/pygotham2019**

Give me feedback on **@_rebecasarai**

References

- CAO, Jianneng; KARRAS, Panagiotis. **Publishing microdata with a robust privacy guarantee**. Proceedings of the VLDB Endowment, v. 5, n. 11, p. 1388-1399, 2012.
- The Algorithmic Foundations of Differential Privacy (here)
- Differentially Private SQL with Bounded User Contribution (here)
- Differential Privacy at Scale: Uber and Berkeley Collaboration (video)
- Tutorial: Differential Privacy and Learning: The Tools, The Results, and The Frontier (video)
- Keeping Your Data Secure While Learning From It - Andreas Dewes and Katharine Jarmul (video)
- 9 Data Anonymization Use Cases You Need To Know Of (here)
- The Definition of Differential Privacy - Cynthia Dwork (video)
- Protecting Personal Data with Django (because it's the law) (video)
- Pseu, Pseu, Pseudio. Pseudonymization in Django. by Frank Valcarcel (video)
- DOMINGO-FERRER, Josep; SORIA-COMAS, Jordi. **Anonymization in the time of big data**. In: International Conference on Privacy in Statistical Databases. Springer, Cham, 2016. p. 57-68.
- LI, Ninghui; LI, Tiancheng; VENKATASUBRAMANIAN, Suresh. **t-closeness: Privacy beyond k-anonymity and l-diversity**. In: 2007 IEEE 23rd International Conference on Data Engineering. IEEE, 2007. p. 106-115.
- SWEENEY, Latanya. **k-anonymity: A model for protecting privacy**. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, v. 10, n. 05, p. 557-570, 2002.
- Apple Releases Details on Differential Privacy, and the Big Takeaway Is Which Emoji Is Most Popular (here)
- Differential Privacy In Action (here)

Other differential privacy projects

- https://github.com/google/rappor
- https://github.com/prashmohan/GUPT
- https://github.com/LLGemini/PINQ
- https://github.com/ektelo/ektelo