CS 458 / 658: Computer Security and Privacy

Module 6 - Data Security and Privacy

Part 1 - On the security of databases

Spring 2022

Outline

Relational Databases

- A (relational) database is a structured collection of data (records).
- Database management system (DBMS) provides support for queries and management of the records.
- Many popular DBMSes are based on the relational model.
- Stores records into one or multiple tables (relations)
  - Table has rows (records) and named columns (attributes).
  - Tables can be related to one another.
- Structure (schema) set by database administrator.

Relations: example

Here is a table that an airline booking agency might use to store details of their customers:

Last	First	Address	City	State	Zip	Airport
ADAMS	Charles	212 Market St.	Columbus	OH	43210	CMH
ADAMS	Edward	212 Market St.	Columbus	OH	43210	CMH
BENCHLY	Zeke	501 Union St.	Chicago	IL	60603	ORD
CARTER	Marlene	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Beth	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Ben	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Lisabeth	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Mary	411 Elm St.	Columbus	OH	43210	CMH

Relations: example

Here is a table that an airline booking agency might use to store details of their customers:

Last	First	Address	City	State	Zip	Airport
ADAMS	Charles	212 Market St.	Columbus	OH	43210	CMH
ADAMS	Edward	212 Market St.	Columbus	OH	43210	CMH
BENCHLY	Zeke	501 Union St.	Chicago	IL	60603	ORD
CARTER	Marlene	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Beth	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Ben	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Lisabeth	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Mary	411 Elm St.	Columbus	OH	43210	CMH

Q: What is the issue with storing data in a flattened table like this?

Relations: normalization

Table: FamilyInfo

Last	Address	City	State	Zip
ADAMS	212 Market St.	Columbus	OH	43210
BENCHLY	501 Union St.	Chicago	IL	60603
CARTER	411 Elm St.	Columbus	OH	43210

Last	First
ADAMS	Charles
ADAMS	Edward
BENCHLY	Zeke
CARTER	Marlene
CARTER	Beth
CARTER	Ben
CARTER	Lisabeth
CARTER	Mary

Table: NameInfo

Zip	Airport
43210	CMH
60603	ORD

Table: AirportInfo

Relations: normalization

Normalization eliminates redundant storage of data, which

- optimizes the storage costs,
- improves query speed, and
- reduces future maintenance costs.

Database queries

The most popular language for query and manipulation of a relational database is SQL.

- A single table query

```
SELECT Address FROM FamilyInfo
 WHERE (Zip = "43210") AND (Name ="ADAMS")
```

Database queries

The most popular language for query and manipulation of a relational database is SQL.

- A single table query

```
SELECT Address FROM FamilyInfo
 WHERE (Zip = "43210") AND (Name ="ADAMS")
```

- A join query across multiple tables

```
SELECT Name, Airport
 FROM FamilyInfo JOIN AirportInfo
 ON FamilyInfo.Zip = AirportInfo.Zip
```

Database queries

The most popular language for query and manipulation of a relational database is SQL.

- A single table query

```
SELECT Address FROM FamilyInfo
 WHERE (Zip = "43210") AND (Name ="ADAMS")
```

- A join query across multiple tables

```
SELECT Name, Airport
 FROM FamilyInfo JOIN AirportInfo
 ON FamilyInfo.Zip = AirportInfo.Zip
```

- An aggregation

```
SELECT COUNT(Last) FROM FamilyInfo
 WHERE City = "Columbus"
```

Database queries

The most popular language for query and manipulation of a relational database is SQL.

- A single table query

```
SELECT Address FROM FamilyInfo
  WHERE (Zip = "43210") AND (Name ="ADAMS")
```

- A join query across multiple tables

```
SELECT Name, Airport
  FROM FamilyInfo JOIN AirportInfo
  ON FamilyInfo.Zip = AirportInfo.Zip
```

- An aggregation

```
SELECT COUNT(Last) FROM FamilyInfo
  WHERE City = "Columbus"
```

- A change of record content

```
UPDATE FamilyInfo SET Address = "1 Town St."
  WHERE Last = "ADAMS"
```

Security requirements for a database

Security requirements for a database

- Access control
  - who can read? who can write?

Security requirements for a database

- Access control
  - who can read? who can write?
- Authentication
  - how do we know if a DB client is not masquerading as someone else

Security requirements for a database

- Access control
  - who can read? who can write?
- Authentication
  - how do we know if a DB client is not masquerading as someone else
- Confidentiality
  - what if the DB server is compromised? what about network tapping?

Security requirements for a database

- Access control
  - who can read? who can write?
- Authentication
  - how do we know if a DB client is not masquerading as someone else
- Confidentiality
  - what if the DB server is compromised? what about network tapping?
- Integrity
  - how do we guarantee that the data is in an intact and sensible state

Security requirements for a database

- Access control
  - who can read? who can write?
- Authentication
  - how do we know if a DB client is not masquerading as someone else
- Confidentiality
  - what if the DB server is compromised? what about network tapping?
- Integrity
  - how do we guarantee that the data is in an intact and sensible state
- Availability
  - redundancy? fault-tolerance? Byzantine fault tolerance?

Security requirements for a database

- Access control
  - who can read? who can write?
- Authentication
  - how do we know if a DB client is not masquerading as someone else
- Confidentiality
  - what if the DB server is compromised? what about network tapping?
- Integrity
  - how do we guarantee that the data is in an intact and sensible state
- Availability
  - redundancy? fault-tolerance? Byzantine fault tolerance?
- Auditability
  - a.k.a. provenance, proving how we ended up with a specific state

Security requirements for a database

- Access control
  - who can read? who can write?
- Authentication
  - how do we know if a DB client is not masquerading as someone else
- Confidentiality
  - what if the DB server is compromised? what about network tapping?
- Integrity
  - how do we guarantee that the data is in an intact and sensible state
- Availability
  - redundancy? fault-tolerance? Byzantine fault tolerance?
- Auditability
  - a.k.a. provenance, proving how we ended up with a specific state

Outline

Access control - Recall OS module

What are some *types* of access control?

Access control - Recall OS module

What are some *types* of access control?

- Discretionary Access Control (DAC)
  - owners can delegate (grant/revoke) privileges to others
- Role-based Access Control (RBAC)
  - ties in users' privileges to their position or roles in the organization
- Mandatory Access Control (MAC)
  - users and objects are assigned labels based on their 'security level'

Access control - Recall OS module

What are some *types* of access control?

- Discretionary Access Control (DAC)
  - owners can delegate (grant/revoke) privileges to others
  - *If you own the data, you can do anything with it.*
- Role-based Access Control (RBAC)
  - ties in users' privileges to their position or roles in the organization
  - *Assign labels to users and assign privileges to labels.*
- Mandatory Access Control (MAC)
  - users and objects are assigned labels based on their 'security level'
  - *You don't own the data even if you create it. The data has labels too and may deny access from its creator.*

Access control for databases

All three types of access control (DAC, RBAC, MAC) apply to databases (with various forms of implementations).

- Most commercial DBs have native support for DAC and RBAC
- Multi-level security database is an implementation of MAC

Access control for databases

All three types of access control (DAC, RBAC, MAC) apply to databases (with various forms of implementations).

- Most commercial DBs have native support for DAC and RBAC
- Multi-level security database is an implementation of MAC

Q: What is the design space of a database access control scheme (i.e., what are the things to consider)?

Access control for databases

All three types of access control (DAC, RBAC, MAC) apply to databases (with various forms of implementations).

- Most commercial DBs have native support for DAC and RBAC
- Multi-level security database is an implementation of MAC

Q: What is the design space of a database access control scheme (i.e., what are the things to consider)?

- Granularity: Access control on *relations*, *records*, *attributes*

Access control for databases

All three types of access control (DAC, RBAC, MAC) apply to databases (with various forms of implementations).

- Most commercial DBs have native support for DAC and RBAC
- Multi-level security database is an implementation of MAC

Q: What is the design space of a database access control scheme (i.e., what are the things to consider)?

- Granularity: Access control on *relations*, *records*, *attributes*
- Supporting different operations: `SELECT`, `INSERT`, `UPDATE`, `DELETE`

DAC for databases

DAC is built-in in the SQL language.

DAC for databases

DAC is built-in in the SQL language.

- Use the `GRANT` keyword to assign a privilege to a user
- Use the `REVOKE` keyword to withdraw a privilege.

DAC for databases

DAC is built-in in the SQL language.

- Use the `GRANT` keyword to assign a privilege to a user
- Use the `REVOKE` keyword to withdraw a privilege.

Different types of privileges have built-in support:

- Account-level privileges:
  - DBMS functionalities (e.g. shutdown server),
  - creating or modifying tables,
  - routines (database functions),
  - users and roles.
- Relation-level privileges:
  - `SELECT`,
  - `UPDATE`,
  - `REFERENCES` privileges on a relation

DAC example: account-level privilege

Accounts A1, A2
Relations: nil

Account-level privilege

```
> Admin:  GRANT CREATE USER TO A1;
```

Sysadmin grants user A1 privilege to create users (and roles).

DAC example: account-level privilege

Accounts A1, A2 , A3
Relations: nil

Account-level privilege

```
> Admin:  GRANT CREATE USER TO A1;
```

Sysadmin grants user A1 privilege to create users (and roles).

Account-level privilege

```
> A1:  CREATE USER A3;
```

User A1 now uses her privilege to create another user.

DAC example: account-level privilege

Accounts A1, A2, A3
Relations: nil

Account-level privilege

> `Admin: GRANT CREATE TABLE TO A2;`

Sysadmin grants user A2 privilege to create new tables.

DAC example: account-level privilege

Accounts A1, A2, A3
Relations: Employee

Account-level privilege

```
> Admin:  GRANT CREATE TABLE TO A2;
```

Sysadmin grants user A2 privilege to create new tables.

Account-level privilege

```
> A2:  CREATE TABLE Employee (...);
```

User A2 now uses her privilege to create the `Employee` table.

DAC example: relation-level privilege

Accounts A1, A2, A3
Relations: `Employee`

Relation-level privilege

```
> A2: GRANT SELECT ON Employee TO A3;
```

The table owner (A2) grants user A3 the privilege to run `SELECT` queries on the `Employee` table.

DAC example: relation-level privilege

Accounts A1, A2, A3

Relations: `Employee`

Relation-level privilege

```
> A2:  GRANT SELECT ON Employee TO A3;
```

The table owner (A2) grants user A3 the privilege to run `SELECT` queries on the `Employee` table.

Relation-level privilege

```
> A2:  GRANT SELECT ON Employee TO A3 WITH GRANT OPTION;
```

The table owner (A2) grants user A3 the privilege to run `SELECT` queries on the `Employee` table and to further delegate that privilege to other users.

DAC example: relation-level privilege

Accounts A1, A2, A3

Relations: `Employee`

Relation-level privilege

```
> A3: GRANT SELECT ON Employee TO A1;
```

A3 now can exercise her delegation rights

DAC example: relation-level privilege

Accounts A1, A2, A3
Relations: `Employee`

Relation-level privilege

```
> A3:  GRANT SELECT ON Employee TO A1;
```

A3 now can exercise her delegation rights

Relation-level privilege

```
> A2:  REVOKE SELECT ON Employee FROM A1;
```

The table owner (A2) however, reserves the rights to revoke any privilege she considers as improper.

Fine-grained DAC

Q: What is missing in the DAC scheme we have seen so far?

Fine-grained DAC

Q: What is missing in the DAC scheme we have seen so far?

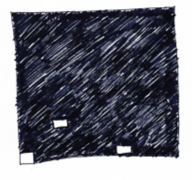


Fig. 74. "Privacy means my life is a black box, except for the items I choose to share with others." By Lauren, age 32

Fine-grained DAC

Q: What is missing in the DAC scheme we have seen so far?

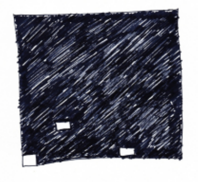


Fig. 74. "Privacy means my life is a black box, except for the items I choose to share with others." By Lauren, age 32

The solution is SQL views:

- For an SQL query, we can generate a view that represents the result of that query.
- Views can be used to only reveal certain columns (attributes after `SELECT`) and rows (defined by the `WHERE` clause) for access control.

Fine-grained DAC using SQL views

Accounts A1, A2, A3

Relations: `Employee(Name, SIN, DOB, Address, Salary, Dpt)`

Create a view

```
> A2:  CREATE VIEW CSEmployeePublicInfo
          SELECT Name, DOB, Address FROM Employee
          WHERE Dpt = "CS";
```

The table owner (A2) creates a view that only expose the (`Name`, `DOB`, `Address`) information for `Employees` in the CS department.

Fine-grained DAC using SQL views

Accounts A1, A2, A3

Relations: `Employee(Name, SIN, DOB, Address, Salary, Dpt)`

Create a view

```
> A2:  CREATE VIEW CSEmployeePublicInfo
           SELECT Name, DOB, Address FROM Employee
           WHERE Dpt = "CS";
```

The table owner (A2) creates a view that only expose the (`Name`, `DOB`, `Address`) information for `Employees` in the CS department.

Relation-level privilege via views

```
> A2:  GRANT SELECT ON CSEmployeePublicInfo TO A3;
```

The table owner (A2) grants user A3 the privilege to run `SELECT` queries on the restrict view instead of the whole `Employee` table.

Fine-grained DAC: what about write operations?

Accounts A1, A2, A3

Relations: `Employee(Name, SIN, DOB, Address, Salary, Dpt)`

Fine-grained DAC: what about write operations?

Accounts A1, A2, A3

Relations: `Employee(Name, SIN, DOB, Address, Salary, Dpt)`

Column-specific update privilege

```
> A2:  GRANT UPDATE ON Employee (Address) TO A3;
```

The table owner (A2) grants user A3 the privilege to `UPDATE` the `Employee` table but only on the `Address` attribute.

Fine-grained DAC: what about write operations?

Accounts A1, A2, A3

Relations: `Employee(Name, SIN, DOB, Address, Salary, Dpt)`

Column-specific update privilege

```
> A2:  GRANT UPDATE ON Employee (Address) TO A3;
```

The table owner (A2) grants user A3 the privilege to `UPDATE` the `Employee` table but only on the `Address` attribute.

Q: How to restrict the `UPDATE` to selective rows only?

Fine-grained DAC: what about write operations?

Accounts A1, A2, A3

Relations: `Employee(Name, SIN, DOB, Address, Salary, Dpt)`

Column-specific update privilege

```
> A2:  GRANT UPDATE ON Employee (Address) TO A3;
```

The table owner (A2) grants user A3 the privilege to `UPDATE` the `Employee` table but only on the `Address` attribute.

Q: How to restrict the `UPDATE` to selective rows only?

Hint: use `UPDATE` triggers.

From DAC to RBAC

Q: If we have DAC in the SQL language, why do we need RBAC?

From DAC to RBAC

Q: If we have DAC in the SQL language, why do we need RBAC?

- DAC requires users to implement the principle of least privilege. (Not done in practice.) Can lead to privilege escalation.
- System administrator needs to know how privileges are inter-related and assign multiple privileges for a user's tasks.
- Need to manually change privileges for multiple users who want to perform the same task, or when a user changes positions in an organization (i.e., roles).

RBAC for databases

Creating and using roles

```
> Admin:  CREATE ROLE "DptAdmin", "CompanyHR";
```

RBAC for databases

Creating and using roles

```
> Admin:  CREATE ROLE "DptAdmin", "CompanyHR";

> Admin:  GRANT "DptAdmin" TO A1;
> Admin:  GRANT "CompanyHR" TO A3;
```

RBAC for databases

Creating and using roles

```
> Admin:  CREATE ROLE "DptAdmin", "CompanyHR";

> Admin:  GRANT "DptAdmin" TO A1;
> Admin:  GRANT "CompanyHR" TO A3;

> A2:  GRANT SELECT ON CSEmployeePublicInfo TO "DptAdmin";
> A2:  GRANT UPDATE ON Employee(Address) TO "CompanyHR";
```

What about MAC?

We show a case study that aims to implement MAC for a database: multi-level security (MLS).

The theory behind MLS is the Bell-La Padula confidentiality model:

- There are security classifications or security levels applied to
  - *Subjects*: i.e., database users — security clearances
  - *Objects*: i.e., each cell in a table — security classifications
- An example of security levels:
  Top Secret > Secret > Classified > Unclassified
- Security goal: ensures that information does not flow to those not cleared for that level.
- Principles (simplified view):
  - *The simple security property*: S can read O iff $L(S) \geq L(O)$.
  - *The star property*: S can write O iff $L(S) \leq L(O)$.

MLS table example

- Users with different clearances see different versions of reality

Name		**Salary**		**Perf**		**TC**
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

- Each attribute has a classification label and a value at that label.
- TC label $=$ *Highest* clearance for any of its attributes.
- Primary key label $\leq$ *Lowest* clearance for any of its attributes.

MLS table example

- Users with different clearances see different versions of reality

Name		**Salary**		**Perf**		**TC**
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

- Each attribute has a classification label and a value at that label.
- TC label $=$ *Highest* clearance for any of its attributes.
- Primary key label $\leq$ *Lowest* clearance for any of its attributes.

Q: Why having this requirement?

MLS table example

- Users with different clearances see different versions of reality

Name		**Salary**		**Perf**		**TC**
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

- Each attribute has a classification label and a value at that label.
- TC label $=$ *Highest* clearance for any of its attributes.
- Primary key label $\leq$ *Lowest* clearance for any of its attributes.

Q: Why having this requirement?
A: Otherwise a user may see a partial record without knowing what that record is about.

MLS read-down by filtering

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

MLS read-down by filtering

Name		**Salary**		**Perf**		**TC**
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

Filtering the table for users having classified clearance:

Name		**Salary**		**Perf**		**TC**
Smith	U	40000	C	-	C	C
Brown	C	-	C	Good	C	C

MLS read-down by filtering

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

Filtering the table for users having classified clearance:

Name		Salary		Perf		TC
Smith	U	40000	C	-	C	C
Brown	C	-	C	Good	C	C

Filtering the table for users having unclassified clearance:

Name		Salary		Perf		TC
Smith	U	-	U	-	U	U

MLS invisible polyinstantiation

- A user with low clearence attempts to insert data in a field that already contains high data.
- Rejecting an update could leak information downwards.

Name		**Salary**		**Perf**		**TC**
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

A user with classified clearance issues a write-up:

```
UPDATE Employee SET Perf = "Great" WHERE Name = "Smith";
```

MLS invisible polyinstantiation

- A user with low clearence attempts to insert data in a field that already contains high data.
- Rejecting an update could leak information downwards.

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

A user with classified clearance issues a write-up:

```
UPDATE Employee SET Perf = "Great" WHERE Name = "Smith";
```

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Smith	U	40000	C	Great	C	C
Brown	C	80000	S	Good	C	S

MLS invisible polyinstantiation

- A user with low clearence attempts to insert data in a field that already contains high data.
- Rejecting an update could leak information downwards.

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

A user with classified clearance issues a write-up:

```
UPDATE Employee SET Perf = "Great" WHERE Name = "Smith";
```

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Smith	U	40000	C	Great	C	C
Brown	C	80000	S	Good	C	S

Q: Why not just override the original record?

MLS invisible polyinstantiation

- A user with low clearence attempts to insert data in a field that already contains high data.
- Rejecting an update could leak information downwards.

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

A user with classified clearance issues a write-up:

```
UPDATE Employee SET Perf = "Great" WHERE Name = "Smith";
```

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Smith	U	40000	C	Great	C	C
Brown	C	80000	S	Good	C	S

Q: Why not just override the original record?

A: An explicit approval is needed to merge the instantiations.

MLS visible polyinstantiation

- A user with high clearence attempts to insert data in a field that already contains low data.
- Overwriting the low data would result in leaking information downwards.

Name		**Salary**		**Perf**		**TC**
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

A user with secret clearance issues a write-down:

```
UPDATE Employee SET Perf = "Bad" WHERE Name = "Brown";
```

MLS visible polyinstantiation

- A user with high clearence attempts to insert data in a field that already contains low data.
- Overwriting the low data would result in leaking information downwards.

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

A user with secret clearance issues a write-down:

```
UPDATE Employee SET Perf = "Bad" WHERE Name = "Brown";
```

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S
Brown	C	80000	S	Bad	S	S

MLS visible polyinstantiation

- A user with high clearence attempts to insert data in a field that already contains low data.
- Overwriting the low data would result in leaking information downwards.

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

A user with secret clearance issues a write-down:

```
UPDATE Employee SET Perf = "Bad" WHERE Name = "Brown";
```

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S
Brown	C	80000	S	Bad	S	S

Q: Why not just override the original record?

MLS visible polyinstantiation

- A user with high clearence attempts to insert data in a field that already contains low data.
- Overwriting the low data would result in leaking information downwards.

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

A user with secret clearance issues a write-down:

```
UPDATE Employee SET Perf = "Bad" WHERE Name = "Brown";
```

Name		Salary		Perf		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S
Brown	C	80000	S	Bad	S	S

Q: Why not just override the original record?

A: An explicit declassification is needed to merge the instantiations.

Or maybe you'd like to keep some information private...

Outline

Security requirements for a database

- Access control
  - who can read? who can write?
- Authentication
  - how do we know if a DB client is not masquerading as someone else
- Confidentiality
  - what if the DB server is compromised? what about network tapping?
- Integrity
  - how do we guarantee that the data is in an intact and sensible state
- Availability
  - redundancy? fault-tolerance? Byzantine fault tolerance?
- Auditability
  - a.k.a. provenance, proving how we ended up with a specific state

Isn't integrity covered in crypto-protocols?

Isn't integrity covered in crypto-protocols?

We are talking about a different type of integrity here.

- In cryptography: integrity means that data cannot be changed without being detected
- In database: integrity means that the data records are in a sensible/correct state

Isn't integrity covered in crypto-protocols?

We are talking about a different type of integrity here.

- In cryptography: integrity means that data cannot be changed without being detected
- In database: integrity means that the data records are in a sensible/correct state

  We will cover the following types of integrity properties:
  - Element integrity
  - All-or-nothing
  - Atomicity
  - Referential integrity

Isn't integrity covered in crypto-protocols?

We are talking about a different type of integrity here.

- In cryptography: integrity means that data cannot be changed without being detected
- In database: integrity means that the data records are in a sensible/correct state

  We will cover the following types of integrity properties:

  - Element integrity
  - All-or-nothing
  - Atomicity
  - Referential integrity

  The goal of ensuring integrity is to prevent users from making changes that will result in an invalid database state. These changes can be either intentional or unintentional.

Element integrity

Example on element integrity violations

```
CREATE TABLE Employee (Name VARCHAR(255), Age INTEGER);
INSERT INTO Employee VALUES ("SMITH", 400);
```

Element integrity

Example on element integrity violations

```
CREATE TABLE Employee (Name VARCHAR(255), Age INTEGER);
INSERT INTO Employee VALUES ("SMITH", 400);
```

Q: What is the problem here? Developer mistake?

Element integrity

Example on element integrity violations

```
CREATE TABLE Employee (Name VARCHAR(255), Age INTEGER);
INSERT INTO Employee VALUES ("SMITH", 400);
```

Q: What is the problem here? Developer mistake?

A: The type system is not expressive enough. There is no way to restrict that `Age` must be in a proper range (e.g., 0-150).

Element integrity

Example on element integrity violations

```
CREATE TABLE Employee (Name VARCHAR(255), Age INTEGER);
INSERT INTO Employee VALUES ("SMITH", 400);
```

Q: What is the problem here? Developer mistake?

A: The type system is not expressive enough. There is no way to restrict that `Age` must be in a proper range (e.g., 0-150).

And there are even more tricky situations, for example:

- At all times, there is at most one employee can have the `Position` attribute set to "CEO".
- A salary increase cannot exceed 100% of the current salary.

Check element integrity with triggers

A typical way to enforce element integrity is to use triggers, i.e., procedures that are automatically executed after each write operation, including `INSERT`, `UPDATE`, `DELETE`, ... queries

Check element integrity with triggers

A typical way to enforce element integrity is to use triggers, i.e., procedures that are automatically executed after each write operation, including `INSERT`, `UPDATE`, `DELETE`, ... queries

An example on SQL trigger

```
CREATE TRIGGER AgeCheck ON Employee
    AFTER INSERT, UPDATE
    FOR EACH ROW
    BEGIN
        IF NEW.Age >= 150
        BEGIN
            RAISERROR ("Invalid age")
        END
    END;
```

Foreign key

Table: FamilyInfo

Last	Address	City	State	Zip
ADAMS	212 Market St.	Columbus	OH	43210
BENCHLY	501 Union St.	Chicago	IL	60603
CARTER	411 Elm St.	Columbus	OH	43210

Last	First
ADAMS	Charles
ADAMS	Edward
BENCHLY	Zeke
CARTER	Marlene
CARTER	Beth
CARTER	Ben
CARTER	Lisabeth
CARTER	Mary

Table: NameInfo

Zip	Airport
43210	CMH
60603	ORD

Table: AirportInfo

Foreign key

Table: FamilyInfo

Last (PK)	**Address**	**City**	**State**	**Zip (FK)**
ADAMS	212 Market St.	Columbus	OH	43210
BENCHLY	501 Union St.	Chicago	IL	60603
CARTER	411 Elm St.	Columbus	OH	43210

Last (FK)	**First**
ADAMS	Charles
ADAMS	Edward
BENCHLY	Zeke
CARTER	Marlene
CARTER	Beth
CARTER	Ben
CARTER	Lisabeth
CARTER	Mary

Table: NameInfo

Zip (PK)	**Airport**
43210	CMH
60603	ORD

Table: AirportInfo

Foreign key

Foreign key in table creation

```
CREATE TABLE FamilyInfo (
  Last VARCHAR(255) NOT NULL,
  Address VARCHAR(1024),
  City VARCHAR(128),
  State VARCHAR(128),
  Zip VARCHAR(128),
  PRIMARY KEY (Last),
  FOREIGN KEY (Zip) REFERENCES AirportInfo(Zip),
);
```

Q: Why do we need this line here?

Referential integrity

Referential integrity ensures that each value of a foreign key *refers* to a valid primary key value, i.e. there are no dangling foreign keys.

One use case: to prevent accidental or intentional deletion of records that are still being used.

Inconsistent state

Recall that integrity is about ensuring the data records are in a sensible/correct state at all times.

But what if a transaction requires two or more write operations? For example: transfer money from Alice to Bob requires two `UPDATE`:

- `UPDATE Ledger SET Balance = Balance - 100 WHERE Name = "Alice";`
- `UPDATE Ledger SET Balance = Balance + 100 WHERE Name = "Bob";`

Inconsistent state

Recall that integrity is about ensuring the data records are in a sensible/correct state at all times.

But what if a transaction requires two or more write operations? For example: transfer money from Alice to Bob requires two `UPDATE`:

- `UPDATE Ledger SET Balance = Balance - 100 WHERE Name = "Alice";`
- `UPDATE Ledger SET Balance = Balance + 100 WHERE Name = "Bob";`

Q: What happens if the database fails after the first `UPDATE`?

Transaction as an all-or-nothing mechanism

Transaction (abort)

```
BEGIN TRANSACTION;
  UPDATE Ledger SET Balance = Balance - 100 WHERE Name = "Alice";
  UPDATE Ledger SET Balance = Balance + 100 WHERE Name = "Bob";
COMMIT TRANSACTION;
```

Transaction as an all-or-nothing mechanism

Transaction (commit or rollback)

```
BEGIN TRANSACTION;
  UPDATE Ledger SET Balance = Balance - 100 WHERE Name = "Alice";
  SELECT @balance = Balance FROM Ledger WHERE Name = "Alice";
  IF @balance < 100
    BEGIN
      ROLLBACK TRANSACTION;
    END
  ELSE
    BEGIN
      UPDATE Ledger SET Balance = Balance + 100 WHERE Name = "Bob";
      COMMIT TRANSACTION;
    END
```

Data race

Notice that in the prior example, we used an unusual syntax to update the balance:

Atomic update (implicit)

```
UPDATE Ledger SET Balance = Balance - 100 WHERE Name = "Alice";
```

Data race

Notice that in the prior example, we used an unusual syntax to update the balance:

Atomic update (implicit)

```
UPDATE Ledger SET Balance = Balance - 100 WHERE Name = "Alice";
```

If used on its own (i.e., not in a transaction context), this is implicitly translated into a transaction:

Atomic update (explicit)

```
BEGIN TRANSACTION;
  SELECT @balance = Balance FROM Ledger WHERE Name = "Alice";
  UPDATE Ledger SET Balance = @balance - 100 WHERE Name = "Alice";
COMMIT TRANSACTION;
```

Data race

Notice that in the prior example, we used an unusual syntax to update the balance:

Atomic update (implicit)

```
UPDATE Ledger SET Balance = Balance - 100 WHERE Name = "Alice";
```

If used on its own (i.e., not in a transaction context), this is implicitly translated into a transaction:

Atomic update (explicit)

```
BEGIN TRANSACTION;
  SELECT @balance = Balance FROM Ledger WHERE Name = "Alice";
  UPDATE Ledger SET Balance = @balance - 100 WHERE Name = "Alice";
COMMIT TRANSACTION;
```

Q: Why must we enclose it within a transaction?

Data race

If two clients send the request concurrently, what will be the result?

Client 1

```
SELECT @balance = Balance
  FROM Ledger WHERE Name = "Alice";

UPDATE Ledger SET Balance =
  @balance - 100 WHERE Name = "Alice";
```

Client 2

```
SELECT @balance = Balance
  FROM Ledger WHERE Name = "Alice";

UPDATE Ledger SET Balance =
  @balance - 100 WHERE Name = "Alice";
```

Data race

If two clients send the request concurrently, what will be the result?

Client 1

```
SELECT @balance = Balance
  FROM Ledger WHERE Name = "Alice";
UPDATE Ledger SET Balance =
  @balance - 100 WHERE Name = "Alice";
```

Client 2

```
SELECT @balance = Balance
  FROM Ledger WHERE Name = "Alice";
UPDATE Ledger SET Balance =
  @balance - 100 WHERE Name = "Alice";
```

One possible interleaving:

Transaction interleavings

```
SELECT @balance = Balance FROM Ledger WHERE Name = "Alice";
SELECT @balance = Balance FROM Ledger WHERE Name = "Alice";
UPDATE Ledger SET Balance = @balance - 100 WHERE Name = "Alice";
UPDATE Ledger SET Balance = @balance - 100 WHERE Name = "Alice";
```

Q: How much is deducted from Alice's balance?

Transaction as a serialization mechanism

Transaction interleavings

```
BEGIN TRANSACTION;
  SELECT @balance = Balance FROM Ledger WHERE Name = "Alice";
  UPDATE Ledger SET Balance = @balance - 100 WHERE Name = "Alice";
COMMIT TRANSACTION;
BEGIN TRANSACTION;
  SELECT @balance = Balance FROM Ledger WHERE Name = "Alice";
  UPDATE Ledger SET Balance = @balance - 100 WHERE Name = "Alice";
COMMIT TRANSACTION;
```

Outline

Security requirements for a database

- Access control
  - who can read? who can write?
- Authentication
  - how do we know if a DB client is not masquerading as someone else
- Confidentiality
  - what if the DB server is compromised? what about network tapping?
- Integrity
  - how do we guarantee that the data is in an intact and sensible state
- Availability
  - redundancy? fault-tolerance? Byzantine fault tolerance?
- Auditability
  - a.k.a. provenance, proving how we ended up with a specific state

Authentication

This is a recap of what we learned from last module. . .

- **Q**: How does a client authenticate a DBMS server?
- **Q**: How does a DBMS server authenticate a client?

Authentication

This is a recap of what we learned from last module...

- **Q**: How does a client authenticate a DBMS server?
  - Certificates
- **Q**: How does a DBMS server authenticate a client?

Authentication

This is a recap of what we learned from last module...

- **Q**: How does a client authenticate a DBMS server?
  - Certificates
- **Q**: How does a DBMS server authenticate a client?
  - Passwords
  - Certificates
  - LDAP (Lightweight Directory Access Protocol) server

Confidentiality

Now we have:

- *Authentication*, which reduces the risk that someone gains unauthorized access to the database.
- *Access control*, which further reduces the risks of leakage of secret information.
- *Correctness*, which guarantees that the DBMS software never has a bug (as we see in the Program Security module) and always comply with the policies.

Q: then what else can go wrong?

Confidentiality

The DBMS is simply an application that runs on some OS, along side with other applications.

- Perhaps that machine itself is stolen and an attacker then removes the hard-drive, and attempts to read off the database contents from the hard-drive.
- Perhaps that other applications are compromised and attackers simply scan over your file system and extract all files related to the database content.
- Perhaps that storage provider itself is malicious, especially in the cloud computing setting, and are curious about what you store in your database.

Confidentiality

Solution? If trust is an issue, check if cryptography can be helpful.

- File-level encryption
- Column-level encryption

Confidentiality

Solution? If trust is an issue, check if cryptography can be helpful.

- File-level encryption
- Column-level encryption

Q: Obviously the key cannot be stored alongside the data, then in this case, how do you supply the key to the DBMS?

Availability

Availability is about recognizing the fact that:

- Transactions can fail due to physical problems.
  - System crashes. Disk failures.
  - Physical problems/catastrophes: power failures, floods, fire, thefts.

Availability

Availability is about recognizing the fact that:

- Transactions can fail due to physical problems.
  - System crashes. Disk failures.
  - Physical problems/catastrophes: power failures, floods, fire, thefts.
- Contingency plans are needed to *recover* from these events

High availability in enterprise settings

- Redundancy: reduce risk that service is affected from some component failure transparently transfer operations to another functioning component.
  - Uninterrupted power supplies.
  - Multiple hard-drives in RAID configurations (with error-detection codes or error-correction codes).

High availability in enterprise settings

- Redundancy: reduce risk that service is affected from some component failure transparently transfer operations to another functioning component.
  - Uninterrupted power supplies.
  - Multiple hard-drives in RAID configurations (with error-detection codes or error-correction codes).
- Database clusters: Redundancy by more machines.
  Load-balancing among clustered machines.

High availability in enterprise settings

- Redundancy: reduce risk that service is affected from some component failure transparently transfer operations to another functioning component.
  - Uninterrupted power supplies.
  - Multiple hard-drives in RAID configurations (with error-detection codes or error-correction codes).
- Database clusters: Redundancy by more machines.
  Load-balancing among clustered machines.
- Failover: deal with catastrophes etc., when machines are down.
  - Clustered machines are in the same physical location, so all machines may be down.
  - Primary system handles traffic regularly WHILE secondary system takes over in case of failures.

Auditability

Expecting the DBMS will never fail in access control or integrity is a dangerous thought!

In the event of a data breach, we want to be able to:

- retroactively identify who has run these queries without authorization.
- hold users accountable and deter such accesses.
- comply with relevant legislation, e.g. HIPAA for health data.

Auditability

- Set an audit policy (or policies) to observe queries received by the DBMS.
- DBMS generates an audit trail or log of events that comply with the audit policy. This log can be processed later into DB tables.
- Archive the audit log periodically to ensure *availability* of the logs for future.

CS 458 / 658: Computer Security and Privacy

Module 6 - Data Security and Privacy

Part 2 - Attacks and defences on data inference

Spring 2022

Outline

1. Intra-database inference
2. Linking against other sources
3. k-anonymity
4. ℓ-diversity
5. t-closeness
6. Limitations of the above privacy notions

A conflict of privacy and utility

How to deal with a (large) collection of data?

- Utility — we want to allow certain SQL queries, as data analysts want to learn interesting properties of the data.
  - e.g., get the average salary of everyone in this company
- Privacy — We also want to protect the privacy of the users whose data is in the database.
  - e.g., without revealing each individual's salary

A conflict of privacy and utility

How to deal with a (large) collection of data?

- Utility — we want to allow certain SQL queries, as data analysts want to learn interesting properties of the data.
  - e.g., get the average salary of everyone in this company
- Privacy — We also want to protect the privacy of the users whose data is in the database.
  - e.g., without revealing each individual's salary

Unfortunately, these two criteria often go against each other:

- the most private strategy has the least utility
- the most powerful analytics has no privacy

A compromise?

Now, what about a compromise solution?

- You're forbidden to issue queries that fetch a particular attribute
  - e.g., `SELECT Salary FROM Employee ...`
- but using aggregates are allowed
  - e.g., `SELECT AVG(Salary) FROM Employee ...`

A compromise?

Now, what about a compromise solution?

- You're forbidden to issue queries that fetch a particular attribute
  - e.g., `SELECT Salary FROM Employee ...`
- but using aggregates are allowed
  - e.g., `SELECT AVG(Salary) FROM Employee ...`

Q: What is the privacy issue with this approach?

Data inference

Data inference problem: Data analysts could infer sensitive data, through output of allowed aggregate queries.

Inference does not have to be a full and accurate recovery of the sensitive data.

- e.g., the employee's salary is $12,345.67

Instead, even a partial revealing of the data is considered as a successful inference and hence a privacy leak.

- e.g., the salary is within the range of $10,000 and $20,000

Our goal is to minimize (unintentional) leaks of sensitive data to the data analysts through the allowed queries.

Inference attack: single query

One single query that directly outputs the sensitive data

Direct attack

```
SELECT SUM(Salary) FROM Employee
  WHERE Name = "Adams"
  AND (Sex = "M" OR Sex = "F" OR Sex = "U");
```

Inference attack: single query

One single query that directly outputs the sensitive data

Direct attack

```
SELECT SUM(Salary) FROM Employee
  WHERE Name = "Adams"
  AND (Sex = "M" OR Sex = "F" OR Sex = "U");
```

Countermeasure: If the `SELECT` clause output includes less than k results, then drop the query. k is usually application specific.

Inference attack: multiple queries

Now, with this k value as a countermeasure, what can we do?

We can use set theory to dictate what queries to send, such that when their outputs are combined, the sensitive value is revealed.

Indirect attack

Q_1: `SELECT SUM(Salary) FROM Employee;` (outputs s)
Q_2: `SELECT SUM(Salary) FROM Employee WHERE Name != "Adams";` (outputs r)

$s - r$ reveals the secret salary.

Inference attack: multiple queries

Now, with this k value as a countermeasure, what can we do?

We can use set theory to dictate what queries to send, such that when their outputs are combined, the sensitive value is revealed.

Indirect attack

Q_1: `SELECT SUM(Salary) FROM Employee;` (outputs s)
Q_2: `SELECT SUM(Salary) FROM Employee WHERE Name != "Adams";` (outputs r)

$s - r$ reveals the secret salary.

Countermeasure: Suppose the database has a total of N records. If the `SELECT` clause output includes less than k results, or more than $N - k$ results (but less than N results), then drop the query. *NOTE*: a query that includes N records (i.e., all records) is OK.

Inference attack: tracker attack

How do we overcome the $k \leq |Q| \leq N - k$ countermeasure?

Inference attack: tracker attack

How do we overcome the $k \leq |Q| \leq N - k$ countermeasure?

Suppose that we find a query T that satisfies this constraint:

- e.g., `SELECT SUM(Salary) FROM Employee WHERE Dpt = "CS";`

For genericity, we use C to represent the (`Dpt = "CS"`) constraint that makes T to include a proper number of records.
And this query T is called a tracker.

Inference attack: tracker attack

How do we overcome the $k \leq |Q| \leq N - k$ countermeasure?

Suppose that we find a query T that satisfies this constraint:

- e.g., `SELECT SUM(Salary) FROM Employee WHERE Dpt = "CS";`

For genericity, we use C to represent the (`Dpt = "CS"`) constraint that makes T to include a proper number of records.
And this query T is called a tracker.

Tracker attack

Q_1: `SELECT SUM(Salary) FROM Employee WHERE Name = "Adams" OR` C;
Q_2: `SELECT SUM(Salary) FROM Employee WHERE Name = "Adams" OR NOT` C;
Q_3: `SELECT SUM(Salary) FROM Employee;`

$Q_1 + Q_2 - Q_3$ reveals the secret salary.

The census reconstruction attack

All the examples shown here involves a database that interactively respond to the attacker's queries. What if one does a one-time release of aggregated data only? For example, the census data?

The census reconstruction attack

Suppose that we have some statistical data about a Census block:

1. There are four people in total.
2. Two of these people have age 17.
3. Two of these people self-identify as White.
4. Two of these people self-identify as Asian.
5. The average age of people who self-identify as White is 30.
6. The average age of people who self-identify as Asian is 32.

The census reconstruction attack

Suppose that we have some statistical data about a Census block:

1. There are four people in total.
2. Two of these people have age 17.
3. Two of these people self-identify as White.
4. Two of these people self-identify as Asian.
5. The average age of people who self-identify as White is 30.
6. The average age of people who self-identify as Asian is 32.

- Take the two people aged 17. Points 1, 3 and 4 tell us that:
  - either they both self-identify as White,
  - either they both self-identify as Asian,
  - either one of them self-identifies as White and the other as Asian.

The census reconstruction attack

Suppose that we have some statistical data about a Census block:

1. There are four people in total.
2. Two of these people have age 17.
3. Two of these people self-identify as White.
4. Two of these people self-identify as Asian.
5. The average age of people who self-identify as White is 30.
6. The average age of people who self-identify as Asian is 32.

- Take the two people aged 17. Points 1, 3 and 4 tell us that:
  - either they both self-identify as White,
  - either they both self-identify as Asian,
  - either one of them self-identifies as White and the other as Asian.
- But only one of these is actually possible!
  - we have a 17-year old Asian and a 17-year old White

The census reconstruction attack

Suppose that we have some statistical data about a Census block:

1. There are four people in total.
2. Two of these people have age 17.
3. Two of these people self-identify as White.
4. Two of these people self-identify as Asian.
5. The average age of people who self-identify as White is 30.
6. The average age of people who self-identify as Asian is 32.

- We have a 17-year old Asian and a 17-year old White
  - **Q:** Who's missing?

The census reconstruction attack

Suppose that we have some statistical data about a Census block:

1. There are four people in total.
2. Two of these people have age 17.
3. Two of these people self-identify as White.
4. Two of these people self-identify as Asian.
5. The average age of people who self-identify as White is 30.
6. The average age of people who self-identify as Asian is 32.

- We have a 17-year old Asian and a 17-year old White
  - **Q:** Who's missing?
  - **A:** A 47 years-old Asian person and a 43 years-old White person

The census reconstruction attack

Suppose that we have some statistical data about a Census block:

1. There are four people in total.
2. Two of these people have age 17.
3. Two of these people self-identify as White.
4. Two of these people self-identify as Asian.
5. The average age of people who self-identify as White is 30.
6. The average age of people who self-identify as Asian is 32.

- We have a 17-year old Asian and a 17-year old White
  - **Q:** Who's missing?
  - **A:** A 47 years-old Asian person and a 43 years-old White person
- When we have billions of statistics with many more attributes to work with, we can convert the data into a massive system of equations (and use computers!). See Damien Desfontaines' blog.

What we learned from these exercises?

Having controls on the type and shape of queries is unlikely be sufficient. We need better (and more systematic) solutions to protect data privacy.

What we learned from these exercises?

Having controls on the type and shape of queries is unlikely be sufficient. We need better (and more systematic) solutions to protect data privacy.

Q: What could be these new solutions?

What we learned from these exercises?

Having controls on the type and shape of queries is unlikely be sufficient. We need better (and more systematic) solutions to protect data privacy.

Q: What could be these new solutions?

- Output coarse-grained results or ranges to queries.
- Change sensitive values slightly by adding randomness.

We will further examine how these solutions work out in real-world.

Outline

Inference across multiple sources

What we have seen so far uses information in a single database only. The inference problem is more severe when the adversary has access to multiple data sources as long as they can link and aggregate the information from different sources.

Inference across multiple sources

What we have seen so far uses information in a single database only. The inference problem is more severe when the adversary has access to multiple data sources as long as they can link and aggregate the information from different sources.

Q: Why more severe?

Inference across multiple sources

What we have seen so far uses information in a single database only. The inference problem is more severe when the adversary has access to multiple data sources as long as they can link and aggregate the information from different sources.

Q: Why more severe?
A: Because access controls rarely apply across data sources.

Obtaining data sources

Q: Where do you get these external data sources?

Obtaining data sources

Q: Where do you get these external data sources?

- Use publicly available data, e.g. census data, regional records.
- Purchase data records from a data broker
- Governments might also share their dossiers with each other.
- Large companies may collect information about their customers.

Data linking

Now, what can we learn from combining these datasets that we didn't learn before?

Data linking

Now, what can we learn from combining these datasets that we didn't learn before?

If these datasets include identifiers that are verinyms, or persistent pseudonyms, one can *link* data records across these datasets to learn more information about an individual or an entity.

Data linking

Now, what can we learn from combining these datasets that we didn't learn before?

If these datasets include identifiers that are verinyms, or persistent pseudonyms, one can *link* data records across these datasets to learn more information about an individual or an entity.

Q: I erased all the identification information before I publicly release the data, would that break the link?

Data linking

Now, what can we learn from combining these datasets that we didn't learn before?

If these datasets include identifiers that are verinyms, or persistent pseudonyms, one can *link* data records across these datasets to learn more information about an individual or an entity.

Q: I erased all the identification information before I publicly release the data, would that break the link?

We will see a series of inference attacks on public data releases that are supposed to protect the privacy of the data suppliers but failed.

Anonymity failure: AOL Search Data Set

- August 6, 2006: AOL released 20 million search queries from 658,000 users over a 3-month period in 2006.
- AOL assigned a random number to each user:
  - 4417749 "numb fingers"
  - 4417749 "60 single men"
  - 4417749 "landscapers in Lilburn, GA"
  - 4417749 "dog that urinates on everything"
  - 711391 "life in Alaska"
- August 9: New York Times article re-identified user 4417749
  - Thelma Arnold, 62-year old widow from Lilburn, GA

Anonymity failure: AOL Search Data Set

- August 6, 2006: AOL released 20 million search queries from 658,000 users over a 3-month period in 2006.
- AOL assigned a random number to each user:
  - 4417749 "numb fingers"
  - 4417749 "60 single men"
  - 4417749 "landscapers in Lilburn, GA"
  - 4417749 "dog that urinates on everything"
  - 711391 "life in Alaska"
- August 9: New York Times article re-identified user 4417749
  - Thelma Arnold, 62-year old widow from Lilburn, GA

Takeaway: simply attaching a random number to each users' record is insufficient to get a high level of nymity.

Anonymity failure: NYC Taxi dataset release

- NYC Taxi Commission released 173 million “anonymized” NYC Taxi trip logs due to a FOIA request
- Each trip log includes information about the trip as well as persistent pseudonyms for each taxi itself.
  - pick-up location (latitude, longitude) and time
  - drop-off location (latitude, longitude) and time
  - MD5 hash of the taxi medallion number
  - MD5 hash of the driver license number
- These parameters were collected in order to learn about taxi usage and traffic patterns.

Anonymity failure: NYC Taxi dataset release

Anonymity problem 1 with this data release: Pick-up / drop-off times and locations can be correlated with celebrities' travels (background knowledge from other news sources).

Anonymity failure: NYC Taxi dataset release

Anonymity problem 1 with this data release: Pick-up / drop-off times and locations can be correlated with celebrities' travels (background knowledge from other news sources).

Example:
You know that a celebrity was spotted leaving the JFK airport at 6pm. $\Longrightarrow$ You look for pick-up records near JFK around 6pm and see where they drop-off. $\Longrightarrow$ After filter out infeasible locations, you might be able to identify the taxi that they took and deduce where they lived or visited.

Anonymity failure: NYC Taxi dataset release

Anonymity problem 1 with this data release: Pick-up / drop-off times and locations can be correlated with celebrities' travels (background knowledge from other news sources).

Example:
You know that a celebrity was spotted leaving the JFK airport at 6pm. $\implies$ You look for pick-up records near JFK around 6pm and see where they drop-off. $\implies$ After filter out infeasible locations, you might be able to identify the taxi that they took and deduce where they lived or visited.

Takeaway: Perhaps these drop-offs/pick-ups could be published at a lower granularity, at the cost of lower utility for statistical analysis of traffic etc?

Anonymity failure: NYC Taxi dataset release

Anonymity problem 2 with this data release: Does hashing help with hiding identities of the drivers and taxicabs?

Anonymity failure: NYC Taxi dataset release

Anonymity problem 2 with this data release: Does hashing help with hiding identities of the drivers and taxicabs?

Background: These two identifiers have the following structures:

- License numbers are 6 or 7 digit numbers
- Medallion numbers are either
  - `[0-9][A-Z][0-9][0-9]`
  - `[A-Z][A-Z][0-9][0-9][0-9]`
  - `[A-Z][A-Z][A-Z][0-9][0-9][0-9]`

Q: How would you uncover their identities?

Anonymity failure: NYC Taxi dataset release

Anonymity problem 2 with this data release: Does hashing help with hiding identities of the drivers and taxicabs?

Background: These two identifiers have the following structures:

- License numbers are 6 or 7 digit numbers
- Medallion numbers are either
  - `[0-9][A-Z][0-9][0-9]`
  - `[A-Z][A-Z][0-9][0-9][0-9]`
  - `[A-Z][A-Z][A-Z][0-9][0-9][0-9]`

Q: How would you uncover their identities?
A: brute-force! There are only 1 million license numbers at most, and 17 million medallion numbers.

Anonymity failure: NYC Taxi dataset release

Anonymity problem 2 with this data release: Does hashing help with hiding identities of the drivers and taxicabs?

Background: These two identifiers have the following structures:

- License numbers are 6 or 7 digit numbers
- Medallion numbers are either
  - `[0-9][A-Z][0-9][0-9]`
  - `[A-Z][A-Z][0-9][0-9][0-9]`
  - `[A-Z][A-Z][A-Z][0-9][0-9][0-9]`

Q: How would you uncover their identities?
A: brute-force! There are only 1 million license numbers at most, and 17 million medallion numbers.

Takeaway: Hashing identifiers does not provide anonymity. With a small input space, a dictionary attack can be conducted efficiently.

Anonymity failure: Massachusetts Insurance Health Records

Massachusetts released
"anonymized" health records:

- ZIP code
- Gender
- Date of birth
- Health information

Anonymity failure: Massachusetts Insurance Health Records

Massachusetts released "anonymized" health records:

- ZIP code
- Gender
- Date of birth
- Health information

Massachusetts' voter registration lists contains:

- ZIP code
- Gender
- Date of birth
- Name

Fun fact: 87% of U.S. population can be uniquely identified using ZIP code, gender, and date of birth!

Lessons learned

- Datasets included data that was useful for research (primary data), as well as some identifiers ("quasi-identifiers").
- *"Quasi-identifiers"* can be used to link data across multiple records in the same dataset (NYC Taxi dataset or AOL search data) or across different datasets (Massachusetts case).
- *Background knowledge* relating to the primary data, can be used to further de-anonymize records.

Privacy vs utility trade-off

What can be done about each type of data in these data releases?

Privacy vs utility trade-off

What can be done about each type of data in these data releases?

For **quasi-identifiers**:

- Reduce granularity to *deter* linking: e.g. year instead of DOB, only first couple digits of zip code. $\implies$ Increases anonymity set.
- Remove attribute(s) to *prevent* linking altogether: e.g. no random number in AOL dataset or no medallion/license number in NYC taxi dataset. Will reduce utility of the dataset.

Privacy vs utility trade-off

What can be done about each type of data in these data releases?

For **quasi-identifiers**:

- Reduce granularity to *deter* linking: e.g. year instead of DOB, only first couple digits of zip code. $\implies$ Increases anonymity set.
- Remove attribute(s) to *prevent* linking altogether: e.g. no random number in AOL dataset or no medallion/license number in NYC taxi dataset. Will reduce utility of the dataset.

For **primary data**:

- Reduce granularity.
- Remove sensitive attributes.
- Publish aggregate statistics.
- Change values slightly (add randomness).

Outline

k-anonymity

k-**anonymity**: For each published record, there exists at least $k-1$ other records with the same quasi-identifier (where $k \geq 2$).

k-anonymity

k-**anonymity**: For each published record, there exists at least $k - 1$ other records with the same quasi-identifier (where $k \geq 2$).

This can be achieved by pre-processing quasi-identifiers such as

- Remove gender altogether.
- Reduce granularity of ZIP code and date of birth.

k-anonymity example

A simple dataset table

ZIP	DOB	Party affiliation
N1CFF	1962-01-24	Green Party
G0ANF	1975-12-30	Liberal Party
N1C5YN	1966-10-17	Green Party
N2J0HJ	1996-08-14	Conservative Party
N1C4KH	1963-04-06	Green Party
G0A3G4	1977-07-09	Conservative Party
G0A3GN	1973-08-14	Liberal Party
N2JWBV	1990-11-02	New Democratic Party
N2JWBV	1990-01-25	Liberal Party

k-anonymity example

A 3-anonymized table (by using coarser-grained quasi-identifiers)

ZIP	DOB	Party affiliation
N1C***	196*-**-**	Green Party
G0A***	197*-**-**	Liberal Party
N1C***	196*-**-**	Green Party
N2J***	199*-**-**	Conservative Party
N1C***	196*-**-**	Green Party
G0A***	197*-**-**	Conservative Party
G0A***	197*-**-**	Liberal Party
N2J***	199*-**-**	New Democratic Party
N2J***	199*-**-**	Liberal Party

k-anonymity example

A 3-anonymized table (organized by equi-class)

ZIP	DOB	Party affiliation
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Conservative Party
N2J***	199*-**-**	Conservative Party
N2J***	199*-**-**	New Democratic Party
N2J***	199*-**-**	Liberal Party

k-anonymity example

A 3-anonymized table (organized by equi-class)

ZIP	**DOB**	**Party affiliation**
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Conservative Party
N2J***	199*-**-**	Conservative Party
N2J***	199*-**-**	New Democratic Party
N2J***	199*-**-**	Liberal Party

Q: Is this good enough?

Homogeneity attack

If you know Alice (N1C***, 196*-**-**) is in this table, what will you learn?

ZIP	DOB	Party affiliation
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Conservative Party
N2J***	199*-**-**	Conservative Party
N2J***	199*-**-**	New Democratic Party
N2J***	199*-**-**	Liberal Party

Homogeneity attack

If you know Alice (N1C***, 196*-**-**) is in this table, what will you learn?

ZIP	DOB	Party affiliation
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Conservative Party
N2J***	199*-**-**	Conservative Party
N2J***	199*-**-**	New Democratic Party
N2J***	199*-**-**	Liberal Party

Homogeneity attack can happen when sensitive values lack diversity. In the worst case, for a given quasi-identifier, all other data values are identical.

Background knowledge attack

If you know Bob (G0A***, 197*-**-**) is in this table, and Bob does not like Liberal Party, what will you learn?

ZIP	DOB	Party affiliation
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Conservative Party
N2J***	199*-**-**	Conservative Party
N2J***	199*-**-**	New Democratic Party
N2J***	199*-**-**	Liberal Party

Background knowledge attack

If you know Bob (G0A***, 197*-**-**) is in this table, and Bob does not like Liberal Party, what will you learn?

ZIP	DOB	Party affiliation
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
N1C***	196*-**-**	Green Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Liberal Party
G0A***	197*-**-**	Conservative Party
N2J***	199*-**-**	Conservative Party
N2J***	199*-**-**	New Democratic Party
N2J***	199*-**-**	Liberal Party

Background knowledge attack can help filter out infeasible values and in the worst case, narrowing down to a single value only.

Outline

1. Intra-database inference
2. Linking against other sources
3. k-anonymity
4. ℓ-diversity
5. t-closeness
6. Limitations of the above privacy notions

ℓ-diversity

ℓ-**diversity**: For any quasi-identifier value, there should be at least ℓ distinct values of the sensitive fields (again $\ell \geq 2$)

ℓ-diversity example

A 3-anonymized 3-diversified table

ZIP	DOB	Salary
N3P***	199*-**-**	20K
N3P***	199*-**-**	15K
N3P***	199*-**-**	25K
H1A***	196*-**-**	100K
H1A***	196*-**-**	90K
H1A***	196*-**-**	120K
S4N***	197*-**-**	50K
S4N***	197*-**-**	60K
S4N***	197*-**-**	65K

ℓ-diversity example

A 3-anonymized 3-diversified table

ZIP	DOB	Salary
N3P***	199*-**-**	20K
N3P***	199*-**-**	15K
N3P***	199*-**-**	25K
H1A***	196*-**-**	100K
H1A***	196*-**-**	90K
H1A***	196*-**-**	120K
S4N***	197*-**-**	50K
S4N***	197*-**-**	60K
S4N***	197*-**-**	65K

Q: Is this good enough?

Similarity attack

If you know Charles who earns a low salary is in this table, what will you learn?

ZIP	DOB	Salary	Disease
N3P***	199*-**-**	20K	gastric ulcer
N3P***	199*-**-**	15K	gastritis
N3P***	199*-**-**	25K	stomach cancer
H1A***	196*-**-**	100K	heart attack
H1A***	196*-**-**	90K	flu
H1A***	196*-**-**	120K	bronchitis
S4N***	197*-**-**	50K	COVID
S4N***	197*-**-**	60K	kidney stone
S4N***	197*-**-**	65K	pneumonia

Similarity attack

If you know Charles who earns a low salary is in this table, what will you learn?

ZIP	DOB	Salary	Disease
N3P***	199*-**-**	20K	gastric ulcer
N3P***	199*-**-**	15K	gastritis
N3P***	199*-**-**	25K	stomach cancer
H1A***	196*-**-**	100K	heart attack
H1A***	196*-**-**	90K	flu
H1A***	196*-**-**	120K	bronchitis
S4N***	197*-**-**	50K	COVID
S4N***	197*-**-**	60K	kidney stone
S4N***	197*-**-**	65K	pneumonia

Similarity attack can help infer correlations between the semantic meanings of attribute values.

Skewness attack

If you know David (in his 20s) is in this table, what will you learn?

ZIP	DOB	Virus X Test
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
… 45 more positive cases …		
N3P***	199*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
… 945 more negative cases …		
H1A***	196*-**-**	Positive

Skewness attack

If you know David (in his 20s) is in this table, what will you learn?

ZIP	DOB	Virus X Test
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
… 45 more positive cases …		
N3P***	199*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
… 945 more negative cases …		
H1A***	196*-**-**	Positive

Skewness attack: the distribution of sensitive values matters!

Outline

What went wrong?

Re-examine: If you know Charles who earns a low salary is in this table, what will you learn?

ZIP	DOB	Salary	Disease
N3P***	199*-**-**	20K	gastric ulcer
N3P***	199*-**-**	15K	gastritis
N3P***	199*-**-**	25K	stomach cancer
H1A***	196*-**-**	100K	heart attack
H1A***	196*-**-**	90K	flu
H1A***	196*-**-**	120K	bronchitis
S4N***	197*-**-**	50K	COVID
S4N***	197*-**-**	60K	kidney stone
S4N***	197*-**-**	65K	pneumonia

What went wrong?

Re-examine: If you know Charles who earns a low salary is in this table, what will you learn?

ZIP	DOB	Salary	Disease
N3P***	199*-**-**	20K	gastric ulcer
N3P***	199*-**-**	15K	gastritis
N3P***	199*-**-**	25K	stomach cancer
H1A***	196*-**-**	100K	heart attack
H1A***	196*-**-**	90K	flu
H1A***	196*-**-**	120K	bronchitis
S4N***	197*-**-**	50K	COVID
S4N***	197*-**-**	60K	kidney stone
S4N***	197*-**-**	65K	pneumonia

Finding: The concentration of stomach diseases in low-income employees is unexpected.

What went wrong?

Re-examine: If you know David (in his 20s) is in this table, what will you learn?

ZIP	DOB	Virus X Test
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
... 45 more positive cases ...		
N3P***	199*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
... 945 more negative cases ...		
H1A***	196*-**-**	Positive

What went wrong?

Re-examine: If you know David (in his 20s) is in this table, what will you learn?

ZIP	DOB	Virus X Test
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive
… 45 more positive cases …		
N3P***	199*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
H1A***	196*-**-**	Negative
… 945 more negative cases …		
H1A***	196*-**-**	Positive

Finding: The distribution of test results are unexpectedly skewed

Reflection

Revealing the overall distribution of the sensitive attribute in the whole dataset should be considered to have no privacy leakage.

Reflection

Revealing the overall distribution of the sensitive attribute in the whole dataset should be considered to have no privacy leakage.

- $\iff$ removing all quasi-identifier attributes preserves privacy.

Reflection

Revealing the overall distribution of the sensitive attribute in the whole dataset should be considered to have no privacy leakage.

- $\iff$ removing all quasi-identifier attributes preserves privacy.
- Seems unavoidable unless willing to destroy utility.

Reflection

Revealing the overall distribution of the sensitive attribute in the whole dataset should be considered to have no privacy leakage.

- $\iff$ removing all quasi-identifier attributes preserves privacy.
- Seems unavoidable unless willing to destroy utility.

However, the distribution of sensitive attribute values in each equi-class (i.e., records that share the same quasi-identifier) are not! And this is where this "unexpected feeling" comes from.

An implied definition of privacy

Privacy is measured by the information gain of an observer.

An implied definition of privacy

Privacy is measured by the information gain of an observer.

The gain is the difference between

- *prior belief*, what the observer knows *before* seeing the data, and
- *posterior belief*: what the observer knowns *after* seeing the data.

An implied definition of privacy

Privacy is measured by the information gain of an observer.

The gain is the difference between

- *prior belief*, what the observer knows *before* seeing the data, and
  - e.g., People have a 5% chance of having Virus X
- *posterior belief*: what the observer knowns *after* seeing the data.
  - e.g., David has 98% chance of having Virus X

t-closeness

t-**closeness**: Distribution of sensitive attribute values in each equi-class should be close to that of the overall dataset. The closeness is measured by some distance calculation method and is bounded by a threshold t.

t-closeness

t-**closeness**: Distribution of sensitive attribute values in each equi-class should be close to that of the overall dataset. The closeness is measured by some distance calculation method and is bounded by a threshold t.

For a list of distance calculation methods, see the original paper that proposes t-closeness on ICDE'07.

Outline

Limitations

- Requires the distinction between quasi-identifiers and sensitive attributes, which is not always possible (and very subjective)

Limitations

- Requires the distinction between quasi-identifiers and sensitive attributes, which is not always possible (and very subjective)
- It is difficult to pin down adversary's background knowledge. For example, the knowledge that a user may have even participated in the dataset helps ultimately to de-anonymize users.

Limitations

- Requires the distinction between quasi-identifiers and sensitive attributes, which is not always possible (and very subjective)
- It is difficult to pin down adversary's background knowledge. For example, the knowledge that a user may have even participated in the dataset helps ultimately to de-anonymize users.
- The privacy notions are syntactic in nature, i.e., the output satisfies the privacy properties but the adversary might be able to infer more information if the adversary knows the algorithm that produces the output.
  - Consider a simple algorithm that produces a 3-anonymized 3-diversified dataset:
    1) repeat the record 2 times and
    2) do a $+1$ and -1 on the sensitive value on each duplicated record.
  - How private is that?

Limitations

However, assuming these limitations,

- k-anonymity
- ℓ-diversity
- t-closeness

is probably the best we can do if we need to release information on an entry-by-entry basis.

But for aggregated data (one-time release or interactive queries), we have a much more powerful tool — *differential privacy*.

CS 458 / 658: Computer Security and Privacy

Module 6 - Data Security and Privacy

Part 3 - Differential privacy

Spring 2022

Outline

We are being too honest...

In all the cases covered in Part 2, we always give a *faithful* aggregation result for each query sent from the data analyst.

We are being too honest...

In all the cases covered in Part 2, we always give a *faithful* aggregation result for each query sent from the data analyst.

For example:

- Inference of the salary
- Census reconstruction attack

We are being too honest...

In all the cases covered in Part 2, we always give a *faithful* aggregation result for each query sent from the data analyst.

For example:

- Inference of the salary
- Census reconstruction attack

Q: How about we add noise to the query response?

Formalize our setup

Formalize our setup

- There is a database, D, which potentially contains sensitive information about individuals.

Formalize our setup

- There is a database, D, which potentially contains sensitive information about individuals.
- The database curator has access to the full database.
  We assume the curator is trusted.

Formalize our setup

- There is a database, D, which potentially contains sensitive information about individuals.
- The database curator has access to the full database.
  We assume the curator is trusted.
- The data analyst consumes the data by asking a series of queries to the curator. Each query is denoted as S and the curator provides a response to query S with R_S.
  The analyst may be honest or malicious.

Formalize our setup

- There is a database, D, which potentially contains sensitive information about individuals.
- The database curator has access to the full database.
  We assume the curator is trusted.
- The data analyst consumes the data by asking a series of queries to the curator. Each query is denoted as S and the curator provides a response to query S with R_S.
  The analyst may be honest or malicious.
- The way in which the curator responds to queries is called the mechanism. Formally, $M : S \to R_S$. We'd like a mechanism that
  - gives statistically useful responses but
  - avoids leaking sensitive information about individuals.

Bad news: adding noise is tricky

Bad news: adding noise is tricky

Dinur-Nissim reconstruction attack: if the mechanism adds too little noise when responding to aggregated queries, an adversary can reconstruct the database *with high accuracy and efficiency*.

Bad news: adding noise is tricky

Dinur-Nissim reconstruction attack: if the mechanism adds too little noise when responding to aggregated queries, an adversary can reconstruct the database *with high accuracy and efficiency*.

This mechanism is called blatantly non-private.

Attack setup

We consider the database to be a collection of n records

$$D = \{d_1, d_2, ..., d_n\}$$

where each record corresponds to one individual.

Attack setup

We consider the database to be a collection of n records

$$D = \{d_1, d_2, ..., d_n\}$$

where each record corresponds to one individual.

Each record d_i may consist of k attributes. For simplicity, we assume that the adversary already knows $k-1$ attribute for all records and the only attribute unknown to the adversary is a single bit.

$$D = \left[\begin{array}{cccc|c} a_{\{1,1\}} & a_{\{1,2\}} & \cdots & a_{\{1,k-1\}} & b_1 \\ a_{\{2,1\}} & a_{\{2,2\}} & \cdots & a_{\{2,k-1\}} & b_2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{\{n,1\}} & a_{\{n,2\}} & \cdots & a_{\{n,k-1\}} & b_n \end{array}\right]$$

Attack setup example

Name	ZIP	DOB	COVID
Alice	K8V 7R6	5/2/1984	1
Bob	V5K 5J9	2/8/2001	0
Charlie	V1C 7J2	10/10/1979	1
David	R4K 5T1	4/4/1944	0
Eve	G7N 8Y3	1/1/1954	1

Threat model

- The attacker is allowed to ask aggregated queries
- Perhaps the most basic type of aggregate query in this case is a counting query
  - how many records in D that satisfies a condition $C(a_{\{*,1\}}, a_{\{*,2\}}, \ldots, a_{\{*,k-1\}})$ have their secret bit set to 1?

Threat model

- The attacker is allowed to ask aggregated queries
- Perhaps the most basic type of aggregate query in this case is a counting query
  - how many records in D that satisfies a condition $C(a_{\{*,1\}}, a_{\{*,2\}}, \ldots, a_{\{*,k-1\}})$ have their secret bit set to 1?

For example: How many rows satisfying condition `(Name = "Charlie" OR DOB > 1980)` have `COVID = 1`.

Threat model

- The attacker is allowed to ask aggregated queries
- Perhaps the most basic type of aggregate query in this case is a counting query
  - how many records in D that satisfies a condition $C(a_{\{*,1\}}, a_{\{*,2\}}, \ldots, a_{\{*,k-1\}})$ have their secret bit set to 1?

For example: How many rows satisfying condition `(Name = "Charlie" OR DOB > 1980)` have `COVID = 1`.

The key point is, the adversary is allowed to pick arbitrary rows in the database using their background knowledge to formulate queries. Formally, $S \in \{0,1\}^n$. An example is $S = [1, 1, 1, 0, 0]$

Curator mechanism

Recall the secret bit vector B = [1, 0, 1, 0, 1].

Upon receiving a query $S = [1, 1, 1, 0, 0]$, the curator will first calculate the true answer $A(S) = S \times [b_1, b_2, \dots, b_n]$.

Curator mechanism

Recall the secret bit vector B = [1, 0, 1, 0, 1].

Upon receiving a query $S = [1, 1, 1, 0, 0]$, the curator will first calculate the true answer $A(S) = S \times [b_1, b_2, \ldots, b_n]$.
True answer $= 2$

Curator mechanism

Recall the secret bit vector B = [1, 0, 1, 0, 1].

Upon receiving a query $S = [1, 1, 1, 0, 0]$, the curator will first calculate the true answer $A(S) = S \times [b_1, b_2, \ldots, b_n]$.
True answer = 2

$$R_S = A(S)$$

Curator mechanism

Recall the secret bit vector B = [1, 0, 1, 0, 1].

Upon receiving a query $S = [1, 1, 1, 0, 0]$, the curator will first calculate the true answer $A(S) = S \times [b_1, b_2, \ldots, b_n]$.
True answer = 2

$$R_S = A(S) + E$$

And subsequently add a random noise E to the true answer.

The inefficient attack

Theorem: If the analyst is allowed to ask 2^n subset queries to a dataset of n users, and the curator adds noise with some bound E, then based on the results, the adversary can reconstruct the database in all but $4E$ positions.

The inefficient attack

Theorem: If the analyst is allowed to ask 2^n subset queries to a dataset of n users, and the curator adds noise with some bound E, then based on the results, the adversary can reconstruct the database in all but $4E$ positions.

e.g., $E = \frac{n}{400} \implies$ reconstruction of 99% entries in the database.

The inefficient attack

Theorem: If the analyst is allowed to ask 2^n subset queries to a dataset of n users, and the curator adds noise with some bound E, then based on the results, the adversary can reconstruct the database in all but $4E$ positions.

e.g., $E = \frac{n}{400} \implies$ reconstruction of 99% entries in the database.

Algorithm:

- For an attacker, there are 2^n candidate databases.
  - e.g., if the true database has 3 users, we have $2^3 = 8$ candidate databases
- For each candidate database $C \in \{0,1\}^n$, if there exists a query S such that $|\Sigma_{i \in S} C[i] - R_S| > E$, rule out C.
- Any database candidate not ruled out (C) differs with the actual database (D) by $4E$ at max.

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

000	001	010	011	100	101	110	111

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)

Q_0=[0, 0, 0] ---> E +0.5

000	001	010	011	100	101	110	111
0	0	0	0	0	0	0	0

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)

Q_0=[0, 0, 0] ---> E +0.5

Q_1=[0, 0, 1] ---> 1 + E -0.5

000	001	010	011	100	101	110	111
0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)

Q_0=[0, 0, 0] ---> E +0.5

Q_1=[0, 0, 1] ---> 1 + E -0.5

Q_2=[0, 1, 0] ---> E +0.5

000	001	010	011	100	101	110	111
0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)

Q_0=[0, 0, 0] ---> E +0.5

Q_1=[0, 0, 1] ---> 1 + E -0.5

Q_2=[0, 1, 0] ---> E +0.5

Q_3=[0, 1, 1] ---> 1 + E +0.5

000	001	010	011	100	101	110	111
0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	1	1	2	0	1	1	2

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)

Query	R(S)	Noise	000	001	010	011	100	101	110	111
Q_0=[0, 0, 0]	---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1]	---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0]	---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1]	---> 1 + E	+0.5	0	1	1	2	0	1	1	2
Q_4=[1, 0, 0]	---> 1 + E	-0.5	0	0	0	0	1	1	1	1

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)

Query	R(S)		000	001	010	011	100	101	110	111
Q_0=[0, 0, 0] ---> E		+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E		-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E		+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E		+0.5	0	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E		-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E		-0.5	0	1	0	1	1	2	1	2

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)

Q_0=[0, 0, 0] ---> E +0.5

Q_1=[0, 0, 1] ---> 1 + E -0.5

Q_2=[0, 1, 0] ---> E +0.5

Q_3=[0, 1, 1] ---> 1 + E +0.5

Q_4=[1, 0, 0] ---> 1 + E -0.5

Q_5=[1, 0, 1] ---> 2 + E -0.5

Q_6=[1, 1, 0] ---> 1 + E -0.5

000	001	010	011	100	101	110	111
0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	1	1	2	0	1	1	2
0	0	0	0	1	1	1	1
0	1	0	1	1	2	1	2
0	0	1	1	1	1	2	2

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)

Query	R(S)	000	001	010	011	100	101	110	111
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	0	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	0	1	0	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	2	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	0	1	1	2	1	2	2	3

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S)		**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	**0**	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	**0**	1	0	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	2	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	**0**	1	1	2	1	2	2	3

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S) ↓		000	001	010	011	100	101	110	111
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	0	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	0	1	0	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	2	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	0	1	1	2	1	2	2	3

| 0 - 1+E | = 1.5

1.5 > E

| 0 - 2+E | = 1.5

1.5 > E

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

Query	R(S)	000	001	010	011	100	101	110	111
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	**0**	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	**0**	1	0	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	2	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	**0**	1	1	2	1	2	2	3

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

Query ↓ R(S)		**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	**0**	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	**0**	1	**0**	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	2	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	**0**	1	1	2	1	2	2	3

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

Query	**R(S)**	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	**0**	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	**0**	1	**0**	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	2	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	**0**	1	1	2	1	2	2	3
		✗	✓	✗	✓				

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

	R(S)	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	0	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	0	1	0	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	2	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	0	1	1	2	1	2	2	3
		✗	✓	✗	✓	✗			

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

Query	R(S)	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	0	1	1	2	0	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	0	1	0	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	2	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	0	1	1	2	1	2	2	3
		✗	✓	✗	✓	✗	✓		

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

R(S) ↓

		000	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	**0**	1	1	2	**0**	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	**0**	1	**0**	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	**2**	2
Q_7=[1, 1, 1] ---> 2 + E	-0.5	**0**	1	1	2	1	2	2	3
		✗	✓	✗	✓	✗	✓	✗	

The inefficient attack - Example

True database D = [1, 0, 1]

E = 0.5

Query	R(S)	000	001	010	011	100	101	110	111
Q_0=[0, 0, 0] ---> E	+0.5	0	0	0	0	0	0	0	0
Q_1=[0, 0, 1] ---> 1 + E	-0.5	0	1	0	1	0	1	0	1
Q_2=[0, 1, 0] ---> E	+0.5	0	0	1	1	0	0	1	1
Q_3=[0, 1, 1] ---> 1 + E	+0.5	**0**	1	1	2	**0**	1	1	2
Q_4=[1, 0, 0] ---> 1 + E	-0.5	0	0	0	0	1	1	1	1
Q_5=[1, 0, 1] ---> 2 + E	-0.5	**0**	1	**0**	1	1	2	1	2
Q_6=[1, 1, 0] ---> 1 + E	-0.5	0	0	1	1	1	1	**2**	**2**
Q_7=[1, 1, 1] ---> 2 + E	-0.5	**0**	1	1	2	1	2	2	**3**
		✗	✓	✗	✓	✗	✓	✗	✗

The inefficient attack

- **Intuition:** If we select a query and send it to the not ruled out databases (C), we can guarantee that these databases don't differ from the true database (D) by "too much".
- **Note:** If an adversary is allowed to ask a lot of queries, it does not matter how much (linear) noise is added to the database.
  - The adversary will be able to reconstruct a large fraction of the data!
- But again, for this attack to work, you need to send a large number of queries.
  - That's why it is inefficient / impractical!

The efficient attack

Theorem: If the analyst is allowed to ask $O(n)$ queries to a dataset of n users, and the curator adds noise with some bound $E = O(\alpha\sqrt{n})$, then based on the results, a computationally efficient adversary can reconstruct the database in all but $O(\alpha^2 n)$ positions.

Blatantly non-private

Definition: A mechanism is blatantly non-private if an adversary can reconstruct a database that matches with the true database in all but $O(n)$ entries.

- say, if under M the adversary can construct a database which agrees with the true database on 99% of the entries!

Blatantly non-private

Definition: A mechanism is blatantly non-private if an adversary can reconstruct a database that matches with the true database in all but $O(n)$ entries.

- say, if under M the adversary can construct a database which agrees with the true database on 99% of the entries!

Note 1: According to the efficient attack scenario, adding a noise of $O(\sqrt{n})$ is blatantly non-private.

Blatantly non-private

Definition: A mechanism is blatantly non-private if an adversary can reconstruct a database that matches with the true database in all but $O(n)$ entries.

- say, if under M the adversary can construct a database which agrees with the true database on 99% of the entries!

Note 1: According to the efficient attack scenario, adding a noise of $O(\sqrt{n})$ is blatantly non-private.

Note 2: This definition does not specify whether a mechanism is private. Instead, it defines a criteria to show that a mechanism is clearly not private.

Blatantly non-private

Definition: A mechanism is blatantly non-private if an adversary can reconstruct a database that matches with the true database in all but $O(n)$ entries.

- say, if under M the adversary can construct a database which agrees with the true database on 99% of the entries!

Note 1: According to the efficient attack scenario, adding a noise of $O(\sqrt{n})$ is blatantly non-private.

Note 2: This definition does not specify whether a mechanism is private. Instead, it defines a criteria to show that a mechanism is clearly not private.

Differential privacy, on the other hand, is a definition on whether a mechanism is private.

Outline

So..., more noise maybe?

We've seen that adding too little noise may compromise the privacy of a database.

So maybe we can add more noise such that the adversary cannot reconstruct the database. But how much more is more?

So..., more noise maybe?

We've seen that adding too little noise may compromise the privacy of a database.

So maybe we can add more noise such that the adversary cannot reconstruct the database. But how much more is more?

Well, that depends on what your privacy goal is.

There is a difference between complete database reconstruction and full database privacy

An informal privacy goal

Consider a setting where

- I hand in my data to a database D (which is trusted),
- an algorithm A runs over D and releases a set of data T,
- the adversary knows the details of A and has access to T.

An informal privacy goal

Consider a setting where

- I hand in my data to a database D (which is trusted),
- an algorithm A runs over D and releases a set of data T,
- the adversary knows the details of A and has access to T.

A privacy notion: I don't care if the adversary can reconstruct the entire database or not. All I care is that the adversary learns (almost) nothing new about me even after seeing A and T, and regardless of what other datasets are available.

An informal privacy goal

Consider a setting where

- I hand in my data to a database D (which is trusted),
- an algorithm A runs over D and releases a set of data T,
- the adversary knows the details of A and has access to T.

A privacy notion: I don't care if the adversary can reconstruct the entire database or not. All I care is that the adversary learns (almost) nothing new about me even after seeing A and T, and regardless of what other datasets are available.

This privacy notion makes no assumption about what background knowledge the adversary might possess:

- If the adversary does not know whether I am in the database, it won't know that either after seeing the result.
- If the adversary already knows whether I am in the database, it won't know more about the secret values I supplied.

An example from the attacker's perspective

An example from the attacker's perspective

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of [45, 55].

An example from the attacker's perspective

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of [45, 55].

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total).

An example from the attacker's perspective

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of [45, 55].

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total).

Q: How can you infer whether Alice is enrolled in CS458 or not?

The attack

Just send 5 queries and observe what is returned by the database.

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?
A: Expect [50, 50, 50, 50, 50] in response.

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?
A: Expect [50, 50, 50, 50, 50] in response.

Q: What will happen if Alice IS enrolled (i.e., D1)?

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?
A: Expect [50, 50, 50, 50, 50] in response.

Q: What will happen if Alice IS enrolled (i.e., D1)?
A: For a single response, we either get

- $63 \hookleftarrow \frac{C_{30}^{2}}{C_{30}^{3}} = 10.7\%$
- $50 \hookleftarrow$ otherwise

The attack

Just send 5 queries and observe what is returned by the database.

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Q: What will happen if Alice IS NOT enrolled (i.e., D2)?
A: Expect [50, 50, 50, 50, 50] in response.

Q: What will happen if Alice IS enrolled (i.e., D1)?
A: For a single response, we either get

- $63 \hookleftarrow \frac{C_{30}^2}{C_{30}^3} = 10.7\%$
- $50 \hookleftarrow$ otherwise

For all 5 responses, the chance of getting at least one 63 is $1 - (1 - \frac{C_{30}^2}{C_{30}^3})^5 = 43.26\%!$

What went wrong?

Alice's score has too much impact on the output! As a result, seeing the output of the algorithm allows the attacker to differentiate which database is the underlying database representing the class score.

What went wrong?

Alice's score has too much impact on the output! As a result, seeing the output of the algorithm allows the attacker to differentiate which database is the underlying database representing the class score.

This is exactly what *Differential Privacy (DP)* tries to capture!

What went wrong?

Alice's score has too much impact on the output! As a result, seeing the output of the algorithm allows the attacker to differentiate which database is the underlying database representing the class score.

This is exactly what *Differential Privacy (DP)* tries to capture!

Informally, the DP notion requires any single element in a dataset to have only a limited impact on the output.

The defense

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of [45, 55].

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total)

The defense

Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of [45, 55].

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total)

The defense

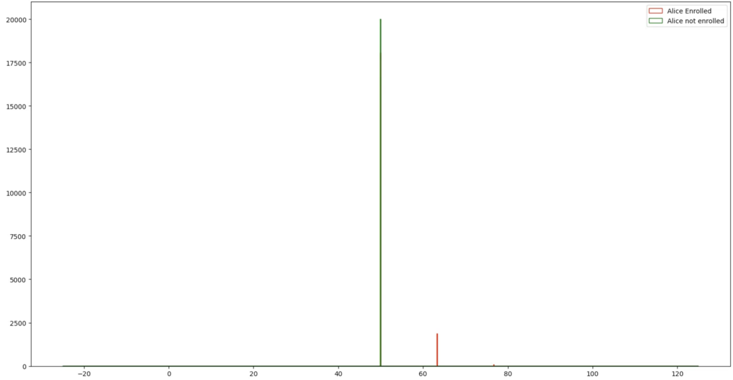
Background knowledge 1: You know that Alice is a top-performer and always gets ≥ 90 in course scores.

Background knowledge 2: CS458 is challenging and historical records show that most students score in the range of [45, 55].

Algorithm: You are given an algorithm that

- allows you to make 5 queries,
- each query returns the average score of 3 randomly selected students (out of 30 scores in total) plus a random value

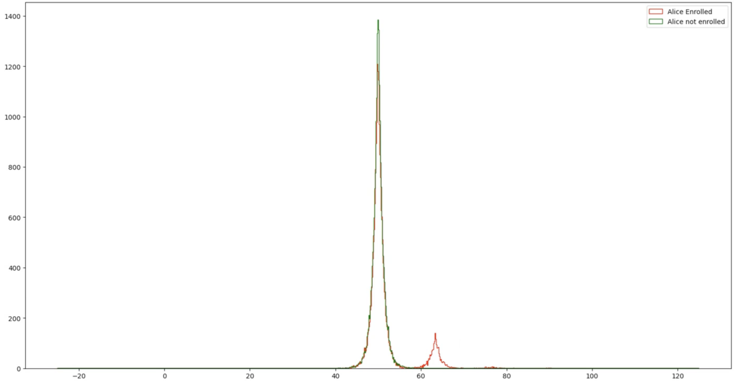
Intuition: No noise



When Alice IS in the database:

- For a given query, most times it will return 50
- Sometimes ($\approx 10\%$) it will return 63

Intuition: Small amount of noise

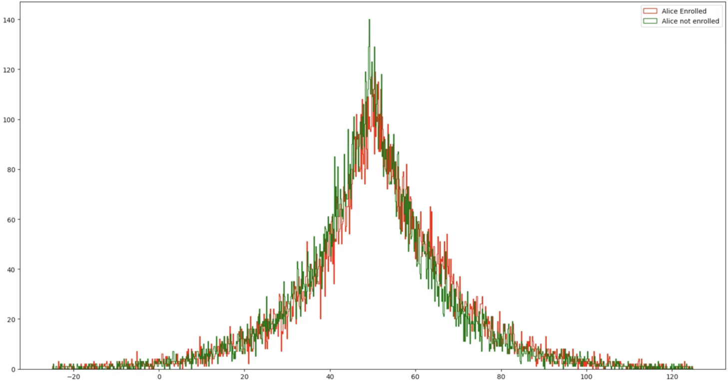


When Alice IS in the database:

- For a given query, most times it will return ≈50
- Sometimes it will return ≈63

Still noticeable!

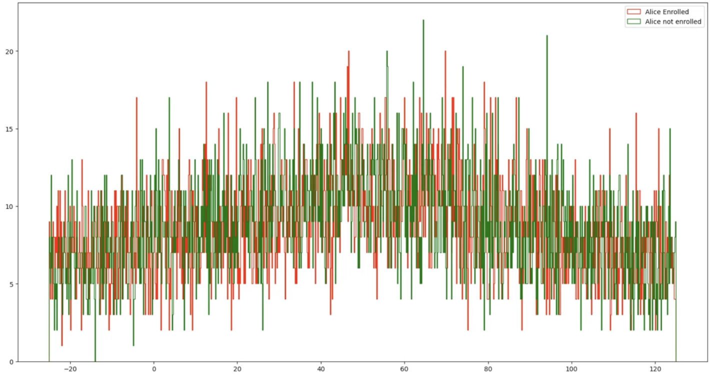
Intuition: Large amount of noise



When Alice IS in the database:

- Query results have a similar probability of occurrence whether Alice is in the database or not (with reasonable utility)
- We may still have a small chance to infer whether Alice is in the database (if we get a query result close to 63)

Intuition: *Very* large amount of noise



When Alice IS in the database:

- We can’t really tell if Alice is in the database or not
- But we completely destroy utility

The appropriate amount of noise

Takeaway: One should set an appropriate amount of noise depending on each particular use case.

- We want to preserve data privacy
- We don't want to destroy utility

The data collectors' argument

... on trying to persuade you to join a differentially private survey:

You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.

The data collectors' argument

... on trying to persuade you to join a differentially private survey:

You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.

But this is only true if they tell you what algorithm they use to release your data and you have verified that their algorithm is indeed differentially private.

Outline

Formalize our setup

- There is a database, D, which potentially contains sensitive information about individuals.
- The database curator has access to the full database.
  We assume the curator is trusted.
- The data analyst consumes the data by asking a series of queries to the curator. Each query is denoted as S and the curator provides a response to query S with R_S.
  The analyst may be honest or malicious.
- The way in which the curator responds to queries is called the mechanism. Formally, $M : S \rightarrow R_S$. We'd like a mechanism that
  - gives statistically useful responses but
  - avoids leaking sensitive information about individuals.

Neighboring databases

Two databases D_1 and D_2 are neighbouring if they agree except for a single entry.

Neighboring databases

Two databases D_1 and D_2 are neighbouring if they agree except for a single entry.

- **Unbounded DP**: D_1 and D_2 are neighboring if D_2 can be obtained from D_1 by adding or removing one element
- **Bounded DP**: D_1 and D_2 are neighboring if D_2 can be obtained from D_1 by replacing one element

ϵ-differential privacy

Idea: If the mechanism M behaves nearly identically for D_1 and D_2, then an attacker can't tell whether D_1 or D_2 was used (and hence can't learn much about the individual).

ϵ-differential privacy

Idea: If the mechanism M behaves nearly identically for D_1 and D_2, then an attacker can't tell whether D_1 or D_2 was used (and hence can't learn much about the individual).

Definition:A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{\epsilon} \Pr[M(D_2) \in T]$$

Meaning: The probability of a subset T of the range of possible responses Y to happen in D_1 is bounded by the probability of the same event to occur in D_2.

ϵ-differential privacy

Definition:
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{\epsilon} \Pr[M(D_2) \in T]$$

ϵ-differential privacy

Definition:
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{\epsilon} \Pr[M(D_2) \in T]$$

The $\forall T \subseteq Y$ means that the attacker cannot find a perspective through which the two databases behaves differently.

ϵ-differential privacy

Definition:
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{\epsilon} \Pr[M(D_2) \in T]$$

The $\forall T \subseteq Y$ means that the attacker cannot find a perspective through which the two databases behaves differently.

In the CS458 grades example, we get an Avg. score as a response:

- $M : \{\text{Name} \times [0 - 100]\} \to [0 - 100]$
- $T : [60 - 100]$
- $\Pr[M(D_1) \in T] = 10.7\% \to$ (Alice in enrolled)
- $\Pr[M(D_2) \in T] = 0\% \to$ (Alice is not enrolled)

ϵ-differential privacy

Recall the definition:
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{\epsilon} \Pr[M(D_2) \in T]$$

Q: Why do we use e^{ϵ} as a multiplicative factor in this bound?

ϵ-differential privacy

Definition (Wrong):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \Pr[M(D_2) \in T] + \epsilon$$

ϵ-differential privacy

Definition (Wrong):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \Pr[M(D_2) \in T] + \epsilon$$

Suppose we have:

- $\epsilon = 0.01$
- $\Pr[M(D_1) \in T] = 0.005$
- $\Pr[M(D_2) \in T] = 0.001$

ϵ-differential privacy

Definition (Wrong):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \Pr[M(D_2) \in T] + \epsilon$$

Suppose we have:

- $\epsilon = 0.01$
- $\Pr[M(D_1) \in T] = 0.005$
- $\Pr[M(D_2) \in T] = 0.001$
- Conforms to the bound,
  but 5x difference

ϵ-differential privacy

Definition (Wrong):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \Pr[M(D_2) \in T] + \epsilon$$

Suppose we have:

- $\epsilon = 0.01$
- $\Pr[M(D_1) \in T] = 0.005$
- $\Pr[M(D_2) \in T] = 0.001$
- Conforms to the bound, but 5x difference

- $\epsilon = 0.01$
- $\Pr[M(D_1) \in T] = 0.96$
- $\Pr[M(D_2) \in T] = 0.94$

ϵ-differential privacy

Definition (Wrong):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \Pr[M(D_2) \in T] + \epsilon$$

Suppose we have:

- $\epsilon = 0.01$
- $\Pr[M(D_1) \in T] = 0.005$
- $\Pr[M(D_2) \in T] = 0.001$
- Conforms to the bound, but 5x difference

- $\epsilon = 0.01$
- $\Pr[M(D_1) \in T] = 0.96$
- $\Pr[M(D_2) \in T] = 0.94$
- Ocurrence is closer, but does not satisfy bound

ϵ-differential privacy

Definition (Better):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \epsilon \times \Pr[M(D_2) \in T]$$

ϵ-differential privacy

Definition (Better):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \epsilon \times \Pr[M(D_2) \in T]$$

Constraints on ϵ:

- It does not make sense for ϵ:
  - to be < 1 (would just switch D_1 and D_2)
  - to be too large

ϵ-differential privacy

Definition (Better):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq \epsilon \times \Pr[M(D_2) \in T]$$

Constraints on ϵ:

- It does not make sense for ϵ:
  - to be < 1 (would just switch D_1 and D_2)
  - to be too large

It seems like we'd like a multiplicative factor close to 1.

ϵ-differential privacy

Definition (Almost):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq (1 + \epsilon) \Pr[M(D_2) \in T]$$

ϵ-differential privacy

Definition (Almost):
A mechanism $M : X \to Y$ is ϵ-differentially private (ϵ-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq (1+\epsilon) \Pr[M(D_2) \in T]$$

NOTE: for small ϵ, $e^{\epsilon} \approx 1 + \epsilon$ by Taylor series:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$

Safety against post-processing

Theorem: Suppose mechanism $M : X \to Y$ is ϵ-differentially private. Then, for any mechanism $A : Y \to Z$, we have that $A \circ M : X \to Z$ is also ϵ-differentially private.

Safety against post-processing

Theorem: Suppose mechanism $M : X \to Y$ is ϵ-differentially private. Then, for any mechanism $A : Y \to Z$, we have that $A \circ M : X \to Z$ is also ϵ-differentially private.

Meaning: Once the data is privatized, it can't be "un-privatized"

Compositional privacy

Theorem: Given

- $M_1 : X \to Y_1$ being ϵ_1-DP, and
- $M_2 : X \to Y_2$ being ϵ_2-DP.

We define a new mechanism $M : X \to Y_1 \times Y_2$ as $M(X) = (M_1(X), M_2(X))$. Then M is $(\epsilon_1 + \epsilon_2)$-DP.

Compositional privacy

Theorem: Given

- $M_1 : X \to Y_1$ being ϵ_1-DP, and
- $M_2 : X \to Y_2$ being ϵ_2-DP.

We define a new mechanism $M : X \to Y_1 \times Y_2$ as $M(X) = (M_1(X), M_2(X))$. Then M is $(\epsilon_1 + \epsilon_2)$-DP.

This has a gossip analogy:

- If A tells you something (potentially with noise),
- and then B tells you some other things (again, with noise).

You may learn more by combining both pieces of information.

Compositional privacy

Theorem: Given

- $M_1 : X \to Y_1$ being ϵ_1-DP, and
- $M_2 : X \to Y_2$ being ϵ_2-DP.

We define a new mechanism $M : X \to Y_1 \times Y_2$ as $M(X) = (M_1(X), M_2(X))$. Then M is $(\epsilon_1 + \epsilon_2)$-DP.

This has a gossip analogy:

- If A tells you something (potentially with noise),
- and then B tells you some other things (again, with noise).

You may learn more by combining both pieces of information.

One may want to set a total privacy loss budget $\epsilon = \epsilon_1 + \epsilon_2 ... + \epsilon_n$.

Group privacy

Theorem: Suppose mechanism $M : X \to Y$ is ϵ-differentially private. Suppose D_1 and D_2 are two databases which differ in exactly k positions. Then:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{k\epsilon} \Pr[M(D_2) \in T]$$

Group privacy

Theorem: Suppose mechanism $M : X \to Y$ is ϵ-differentially private. Suppose D_1 and D_2 are two databases which differ in exactly k positions. Then:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{k\epsilon}\Pr[M(D_2) \in T]$$

If you need to hide the “effects” caused by a whole group, you need to prepare a larger privacy budget.

Outline

Sensitivity

Q: How much noise to add?

Sensitivity

Q: How much noise to add? $\longleftarrow$ Sensitivity is a measurement

Sensitivity

Q: How much noise to add? $\longleftarrow$ Sensitivity is a measurement

Definition: given a query processing function $f : X \to \mathbb{R}^k$, the ℓ_1-sensitivity of f is defined as:

$$\Delta_1^f = \max_{D_1 \sim D_2} \|f(D_1) - f(D_2)\|_1 \quad \text{where } D_1, D_2 \in X$$

Sensitivity

Q: How much noise to add? $\longleftarrow$ Sensitivity is a measurement

Definition: given a query processing function $f : X \to \mathbb{R}^k$, the ℓ_1-sensitivity of f is defined as:

$$\Delta_1^f = \max_{D_1 \sim D_2} \|f(D_1) - f(D_2)\|_1 \quad \text{where } D_1, D_2 \in X$$

Note 1: The range of f is k-dimensional

- e.g., Avg. and Sum. of different attributes in a public data release

Sensitivity

Q: How much noise to add? $\longleftarrow$ Sensitivity is a measurement

Definition: given a query processing function $f : X \rightarrow \mathbb{R}^k$, the ℓ_1-sensitivity of f is defined as:

$$\Delta_1^f = \max_{D_1 \sim D_2} \|f(D_1) - f(D_2)\|_1 \quad \text{where } D_1, D_2 \in X$$

Note 1: The range of f is k-dimensional

- e.g., Avg. and Sum. of different attributes in a public data release

Note 2: ℓ_1-sensitivity is the ℓ_1-norm:
$\|\vec{x_1} - \vec{x_2}\|_1 = \sum_i |\vec{x_1}[i] - \vec{x_2}[i]|$

Sensitivity w/ one pair of neighboring databases

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Algorithm: You are allowed to make a query that returns the average score of this course.

Q: What is the ℓ_1-sensitivity here?

Sensitivity w/ one pair of neighboring databases

D1 with Alice enrolled:

- Alice: 90
- Everyone else (29 of them): 50

D2 with Alice not enrolled:

- Everyone (30 of them): 50

Algorithm: You are allowed to make a query that returns the average score of this course.

Q: What is the ℓ_1-sensitivity here?
A: $|\mathrm{Avg}(D_1) - \mathrm{Avg}(D_2)| = 51.33 - 50 = 1.33$

Laplace distribution

Lap($mean = \mu, scaling = b$) is defined as:

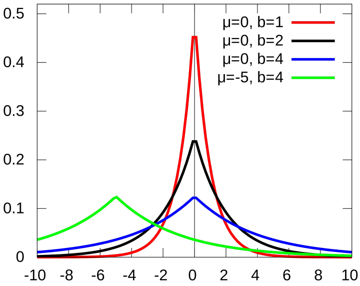
$$\Pr[x = v] = \frac{1}{2b}\exp\left(\frac{-|v-\mu|}{b}\right)$$

Laplace distribution

Lap($mean = \mu, scaling = b$) is defined as:

$$\Pr[x = v] = \frac{1}{2b} \exp\left(\frac{-|v - \mu|}{b}\right)$$

- Usually, for DP, we set $\mu = 0$, so you may see Lap(b) which is essentially Lap($0, b$)
- Lap(μ, b) has variance $\sigma^2 = 2b^2$
- As b increases, the distribution becomes more flat



Laplace mechanism

Definition: Let $f : X \to \mathbb{R}^k$ is the function that calculates the "true" value of a query. The Laplace mechanism is defined as:

$$M(D) = f(D) + (Y_1, Y_2, \cdots, Y_k)$$

where Y_i are independent and identically distributed (i.i.d) random variables sampled from Lap($\frac{\Delta_1^f}{\epsilon}$)

Laplace mechanism

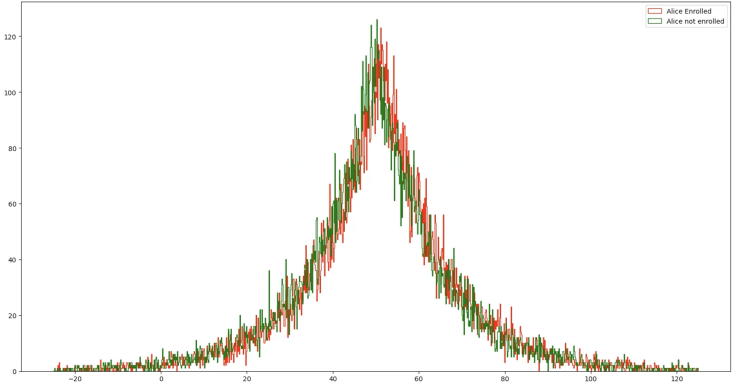
Definition: Let $f : X \to \mathbb{R}^k$ is the function that calculates the "true" value of a query. The Laplace mechanism is defined as:

$$M(D) = f(D) + (Y_1, Y_2, \cdots, Y_k)$$

where Y_i are independent and identically distributed (i.i.d) random variables sampled from $\text{Lap}(\frac{\Delta_1^f}{\epsilon})$

In our CS458 example:
let's take $\epsilon = 0.1$, and together with $\Delta = 1.33$, we have
$M(D) = f(D) + \text{Lap}(13.3)$

Laplace mechanism



- Both curves mostly overlap (with a slight shift)
- The green curve centers around 50
- The red curve centers around 51.33

Does the Laplace mechanism work in our example?

Let's first update the PDF by replacing $b = \frac{\Delta}{\epsilon}$:

$$\Pr[x = v] = \frac{\epsilon}{2\Delta}\exp\left(\frac{-\epsilon|v-\mu|}{\Delta}\right)$$

For D_1, $\mu = 51.33$,

$$\Pr_1[x = 51.33] = \frac{\epsilon}{2\Delta}\exp\left(\frac{-\epsilon|51.33-51.33|}{\Delta}\right) = C \times e^0$$

For D_2, $\mu = 50$,

$$\Pr_2[x = 51.33] = \frac{\epsilon}{2\Delta}\exp\left(\frac{-\epsilon|51.33-50|}{\Delta}\right) = C \times e^{-0.1}$$

$$\frac{\Pr_1[x = 51.33]}{\Pr_2[x = 51.33]} = \frac{C \times e^0}{C \times e^{-0.1}} \approx 1.105$$

The Laplace mechanism is ϵ-DP

Proof result:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \to \mathbb{R}^k$ be the function that calculates the “true” value
- Let $z \in \mathbb{R}^k$ being any potential response

The Laplace mechanism is ϵ-DP

Proof result:

- Let D_1 and D_2 be any neighboring databases
- Let $f : X \to \mathbb{R}^k$ be the function that calculates the "true" value
- Let $z \in \mathbb{R}^k$ being any potential response

$$\frac{\Pr[M(D_1) = z]}{\Pr[M(D_2) = z]} \leq \exp(\epsilon)$$

Outline

Approximate differential privacy

Definition:
A mechanism $M : X \to Y$ is (ϵ, δ)-differentially private ((ϵ, δ)-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{\epsilon} \Pr[M(D_2) \in T] + \delta$$

Approximate differential privacy

Definition:
A mechanism $M : X \to Y$ is (ϵ, δ)-differentially private ((ϵ, δ)-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{\epsilon} \Pr[M(D_2) \in T] + \delta$$

Interpretation: The new privacy parameter, δ, represents a "failure probability" for the definition.

- With probability $1 - \delta$ we will get the same guarantee as pure differential privacy;
- With probability δ, we get no privacy guarantee at all.

Approximate differential privacy

Definition:
A mechanism $M : X \to Y$ is (ϵ, δ)-differentially private ((ϵ, δ)-DP) if for any two neighboring databases $D_1 : X$ and $D_2 : X$:

$$\forall T \subseteq Y, \quad \Pr[M(D_1) \in T] \leq e^{\epsilon} \Pr[M(D_2) \in T] + \delta$$

Interpretation: The new privacy parameter, δ, represents a "failure probability" for the definition.

- With probability $1 - \delta$ we will get the same guarantee as pure differential privacy;
- With probability δ, we get no privacy guarantee at all.

This definition allows us to add a much smaller noise.

Even more topics about differential privacy

You may want to check CS860 (F'20) – Algorithms for Private Data Analysis, as taught by Prof. Kamath here in the School.
The course's contents are actually available online!

CS 458 / 658: Computer Security and Privacy

Module 6 - Data Security and Privacy

Part 4 - Adversarial machine learning

Spring 2022

Outline

1. Model reconstruction attacks
2. Membership inference attacks
3. Poisoning attacks
4. Adversarial examples

Outline

1. Model reconstruction attacks
2. Membership inference attacks
3. Poisoning attacks
4. Adversarial examples

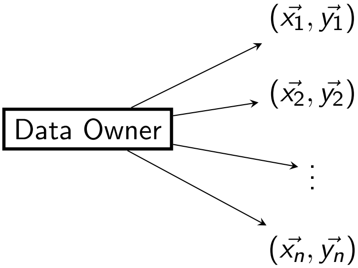
Model stealing via prediction APIs

Based on paper

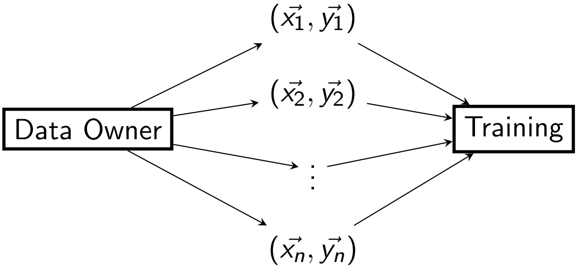
Stealing Machine Learning Models via Prediction APIs by *Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, Thomas Ristenpart* . Presented in USENIX Security 2016

Both the paper and the author's conference talk is available online.

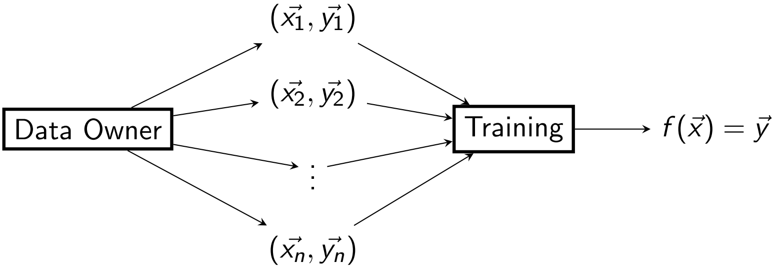
A supervised machine learning setting



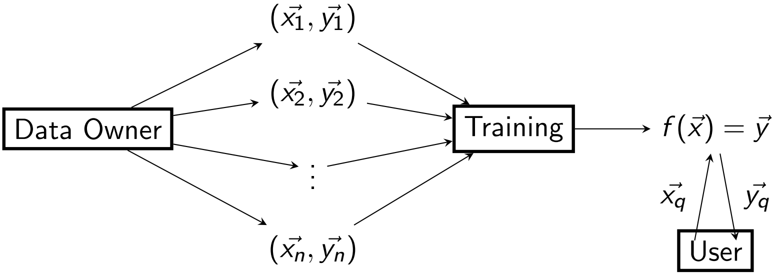
A supervised machine learning setting



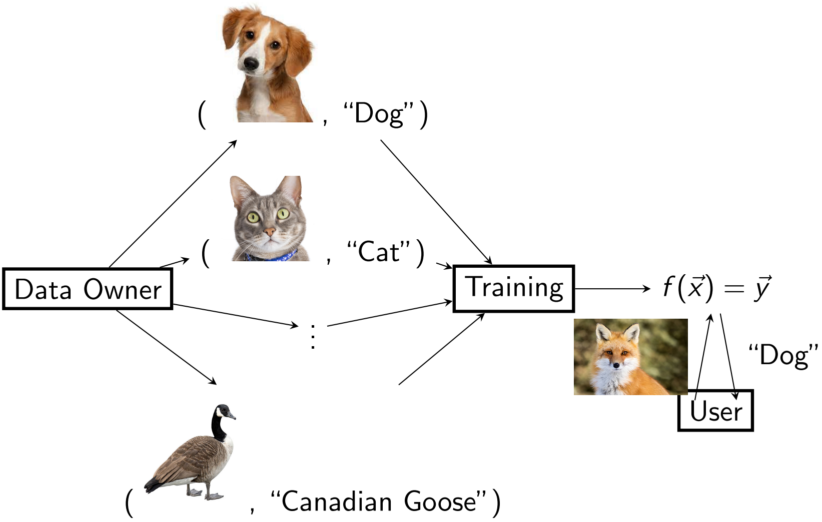
A supervised machine learning setting



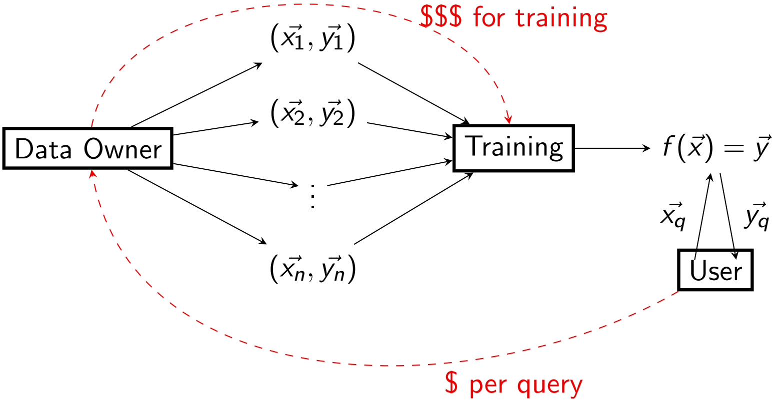
A supervised machine learning setting



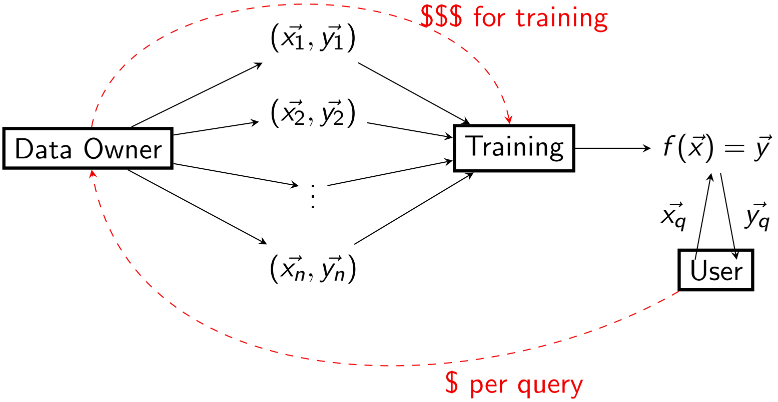
A supervised machine learning setting



Machine learning as a service (MLaaS)



Machine learning as a service (MLaaS)

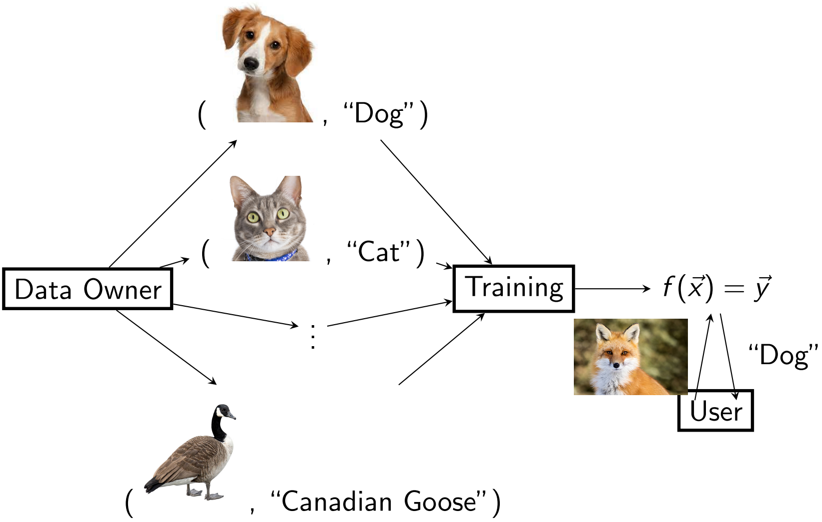


Conflicting goals from the data owner's perspective:

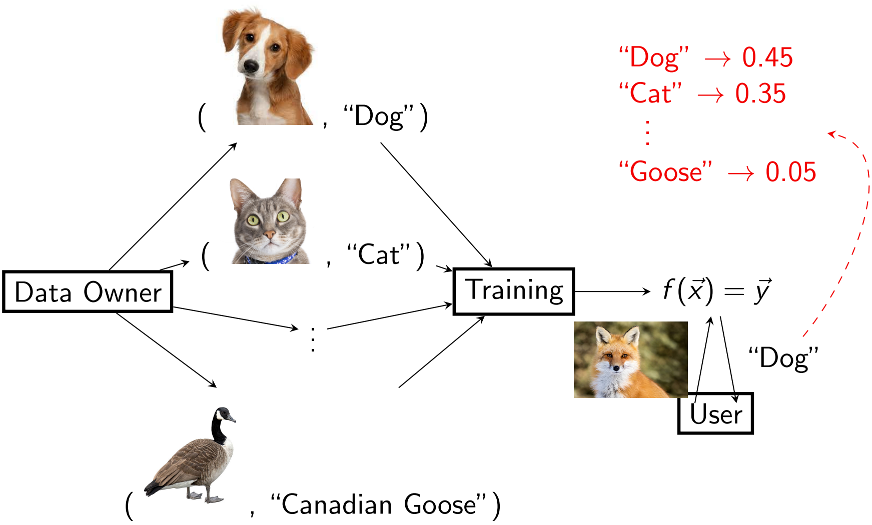
- The prediction APIs return high-precision results with rich info
- The confidentiality of the model needs to be protected

What can go wrong?

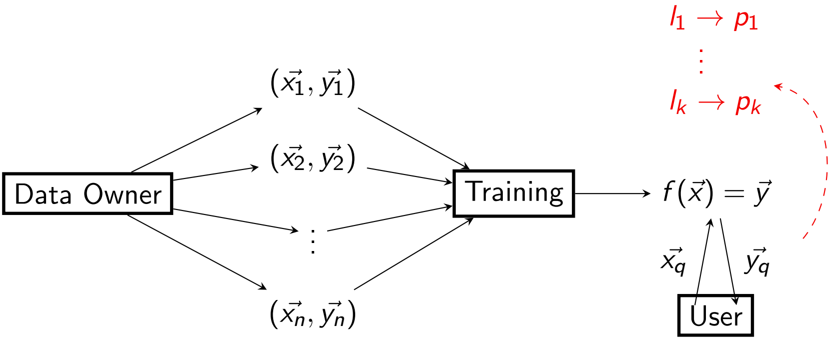
What can go wrong?



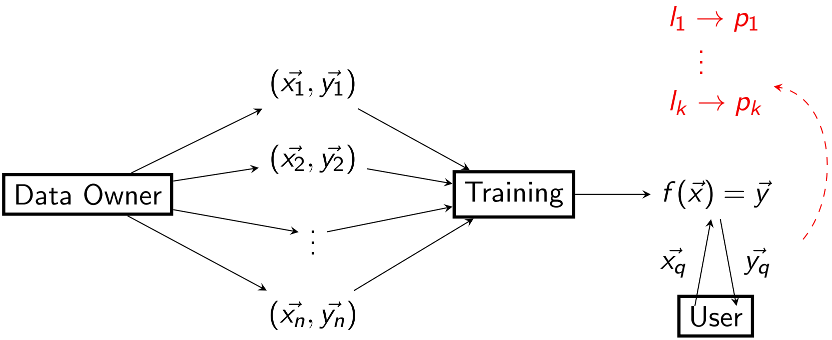
What can go wrong?



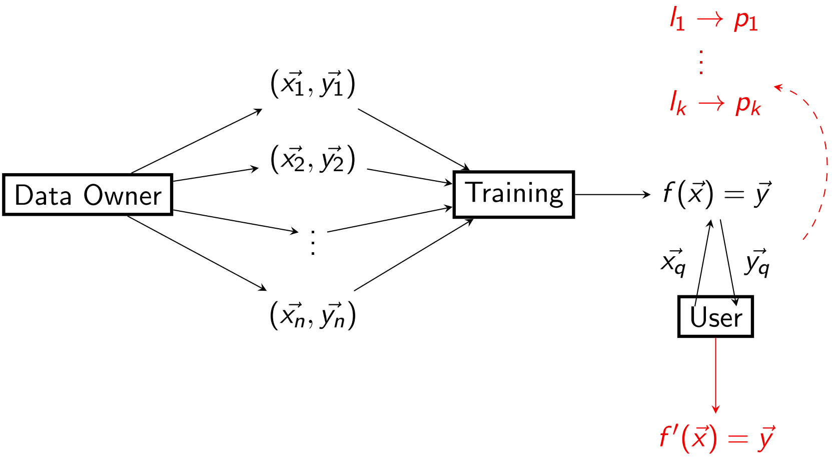
What can go wrong?



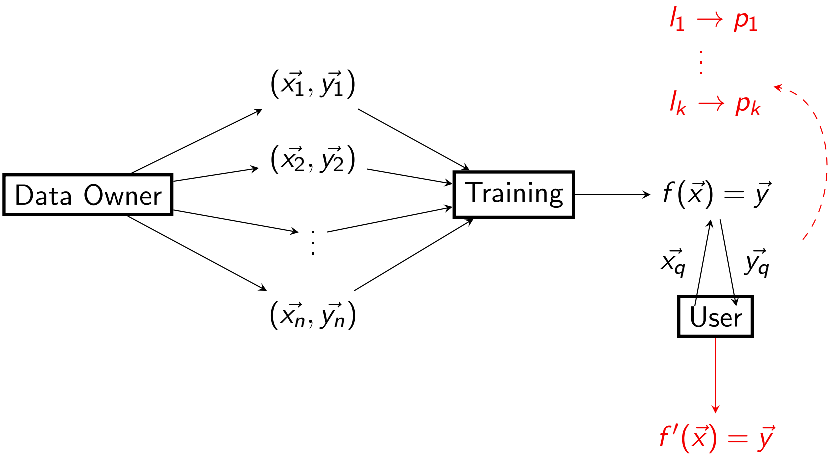
Model extraction attack



Model extraction attack



Model extraction attack



Goal: reconstruct a close approximate of f using as few queries as possible, i.e., $f'(\vec{x}) = f(\vec{x})$ for 99.9% of inputs.

Binary logistic regression

Binary logistic regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.

Binary logistic regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.

Example: Students spends between 0 and 5 hours studying for CS458 final exam. How does the number of hours spent studying affect the probability of the student passing the exam?

Hours (x)	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Pass (y)	0	0	0	1	0	1	0	1	1	1	1

Binary logistic regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.

Example: Students spends between 0 and 5 hours studying for CS458 final exam. How does the number of hours spent studying affect the probability of the student passing the exam?

Hours (x)	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Pass (y)	0	0	0	1	0	1	0	1	1	1	1

The logistic function (i.e., the model) is of the form:

$$f(x) = \frac{1}{1 + e^{-(ax+b)}}$$

Binary logistic regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.

Example: Students spends between 0 and 5 hours studying for CS458 final exam. How does the number of hours spent studying affect the probability of the student passing the exam?

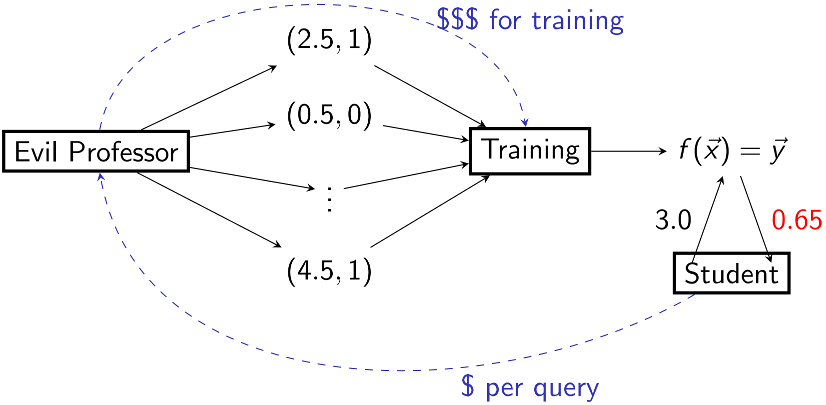
Hours (x)	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Pass (y)	0	0	0	1	0	1	0	1	1	1	1

The logistic function (i.e., the model) is of the form:

$$f(x) = \frac{1}{1 + e^{-(ax+b)}}$$

Training $\implies$ finding the value of a and b that minimizes the classification loss (or maximize the accuracy).

Final exam prediction as a service



Recovery of logistic regression model

Transform

$$f(x) = \frac{1}{1 + e^{-(ax+b)}}$$

into

$$\ln\left(\frac{f(X)}{1 - f(X)}\right) = ax + b$$

Recovery of logistic regression model

Transform

$$f(x) = \frac{1}{1 + e^{-(ax+b)}}$$

into

$$\ln\left(\frac{f(X)}{1 - f(X)}\right) = ax + b$$

Given two data points $(x_1, f(x_1))$ and $(x_2, f(x_2))$, we can fully recover the parameters a and b

Recovery of logistic regression model

Transform

$$f(x) = \frac{1}{1 + e^{-(ax+b)}}$$

into

$$\ln\left(\frac{f(X)}{1 - f(X)}\right) = ax + b$$

Given two data points $(x_1, f(x_1))$ and $(x_2, f(x_2))$, we can fully recover the parameters a and b

This means that you can reconstruct a local model f' which behaves exactly the same as f on all inputs.

This idea generalizes to other ML models

- Logistic regression
- Decision trees
- Support vector machines
- Neural networks

This idea generalizes to other ML models

- Logistic regression
- Decision trees
- Support vector machines
- Neural networks

Successful attacks against cloud MLaaS providers including

- Amazon web services
- BigML

Outline

1. Model reconstruction attacks
2. Membership inference attacks
3. Poisoning attacks
4. Adversarial examples

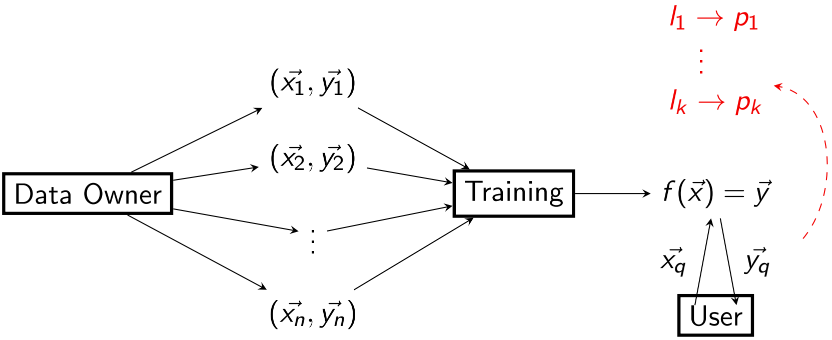
Membership inference via prediction APIs

Based on paper

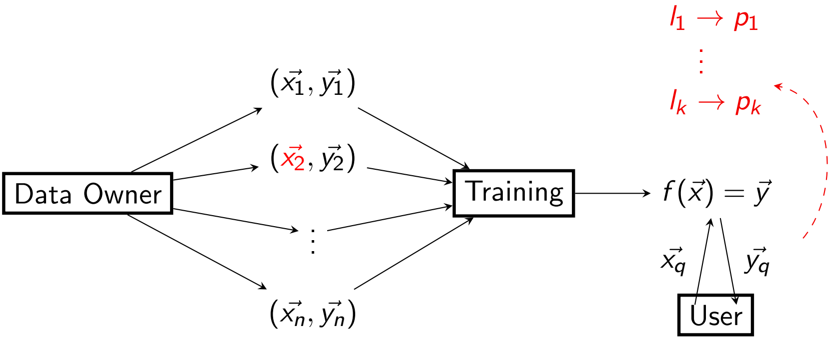
Membership Inference Attacks against Machine Learning Models by *Reza Shokri, Marco Stronati, Congzheng Song, Vitaly Shmatikov* . Presented in IEEE S&P 2017

Both the paper and the author's conference talk is available online.

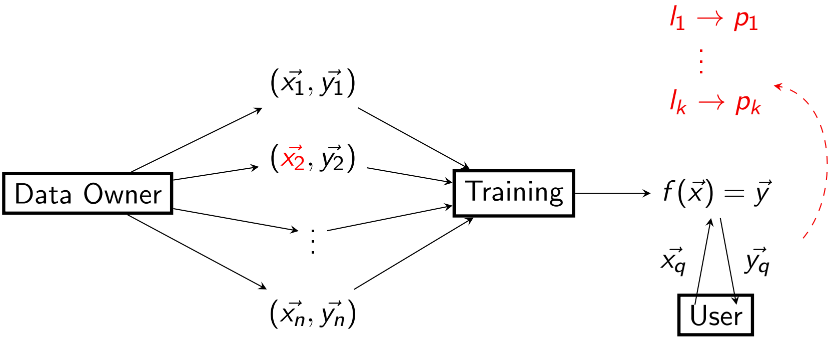
What can go wrong?



Member inference attack



Member inference attack



The model remains in the cloud as a black-box, i.e., the user

- does not have direct access to the model
- does not know the type and architecture of the model
- does not know the parameters of the model
- does not know anything about the trainig data
- has no access to the intermediate steps of the prediction

The main insight

Machine learning models tend to react differently with respect to its training data vs data it has never seen before.

Q: What do you call this phenomenon?

The main insight

Machine learning models tend to react differently with respect to its training data vs data it has never seen before.

Q: What do you call this phenomenon?
A: Overfitting!

The main insight

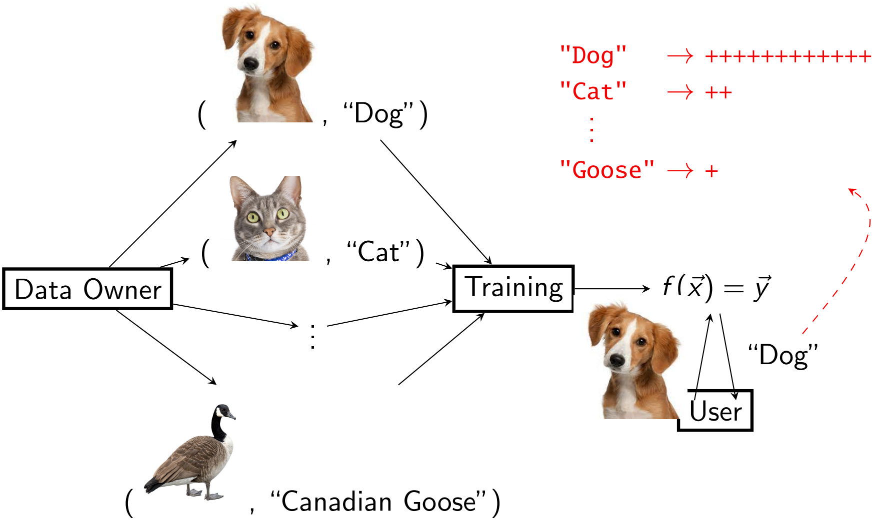
Machine learning models tend to react differently with respect to its training data vs data it has never seen before.

Q: What do you call this phenomenon?
A: Overfitting!

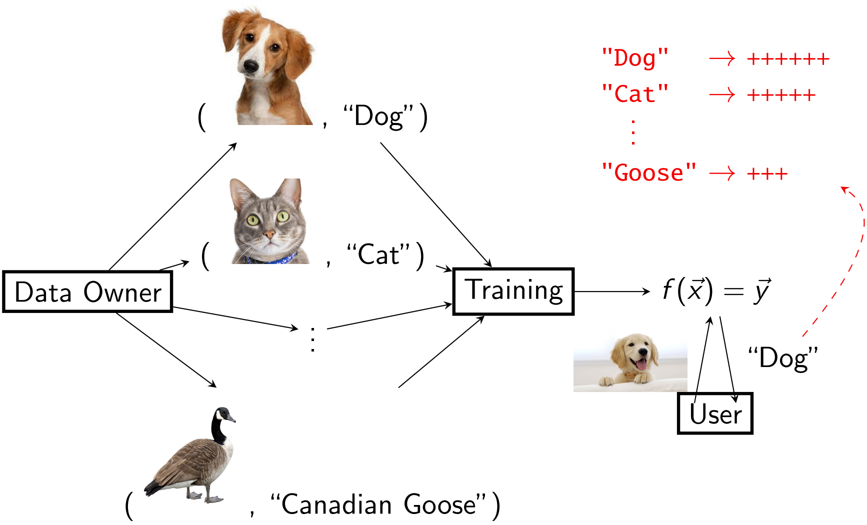
The accuracy of the training data is much higher than the prediction accuracy of the test data.

The distribution of classification results

The distribution of classification results



The distribution of classification results



The distribution of classification results

Query $\in$ the training set:

l_1	+++++++++
l_2	+++++++++++
l_3	+++++++
l_4	+++
$\vdots$	$\vdots$
l_n	+++++++

Query $\notin$ the training set:

l_1	+++++++
l_2	+++
l_3	+++++++
l_4	++++++++++
$\vdots$	$\vdots$
l_n	+++++++++

The distribution of classification results

Query $\in$ the training set:

l_1	$++++++++$
l_2	$++++++++++$
l_3	$++++++$
l_4	$+++$
$\vdots$	$\vdots$
l_n	$++++++$

Query $\notin$ the training set:

l_1	$++++++$
l_2	$+++$
l_3	$++++++$
l_4	$+++++++++$
$\vdots$	$\vdots$
l_n	$++++++++$

Q: How to recognize the difference between these distributions?

The distribution of classification results

Query $\in$ the training set:

l_1	++++++++
l_2	++++++++++
l_3	++++++
l_4	+++
$\vdots$	$\vdots$
l_n	++++++

Query $\notin$ the training set:

l_1	++++++
l_2	+++
l_3	++++++
l_4	+++++++++
$\vdots$	$\vdots$
l_n	++++++++

Q: How to recognize the difference between these distributions?

A: This is a classification problem...

The distribution of classification results

Query $\in$ the training set:

l_1	+++++++++
l_2	+++++++++++
l_3	+++++++
l_4	+++
$\vdots$	$\vdots$
l_n	+++++++

Query $\notin$ the training set:

l_1	+++++++
l_2	+++
l_3	+++++++
l_4	++++++++++
$\vdots$	$\vdots$
l_n	+++++++++

Q: How to recognize the difference between these distributions?

A: This is a classification problem... and... let's throw machine learning to solve it! *... only magic can defeat magic ...*

How to train the attacker's ML model?

Recall that the attacker knows nothing about the training data nor the internal details of the target ML model.

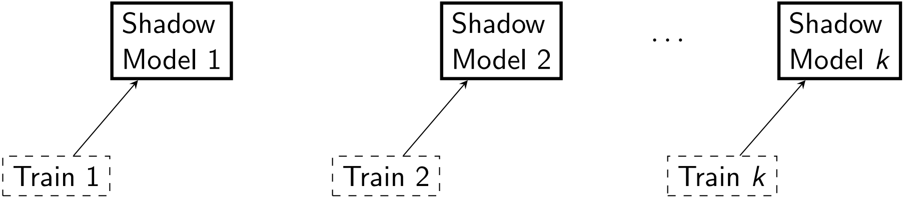
How to train the attacker's ML model?

Recall that the attacker knows nothing about the training data nor the internal details of the target ML model.

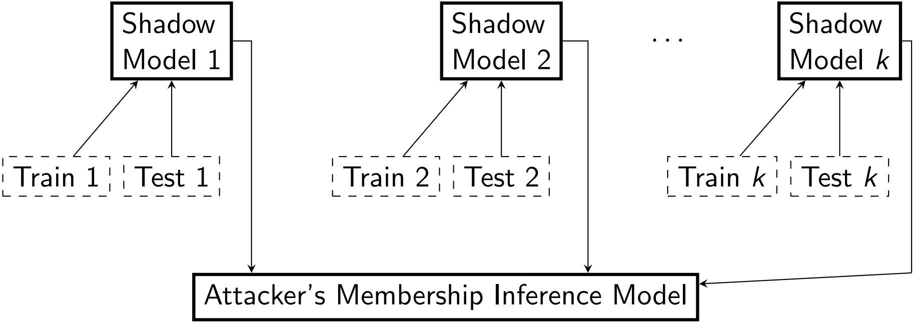
The solution: use shadow models that are controllable by the attacker. Shadow models should ideally

- share the type and architecture with the target model, and
- might differ in parameters (e.g., weights in neural networks).

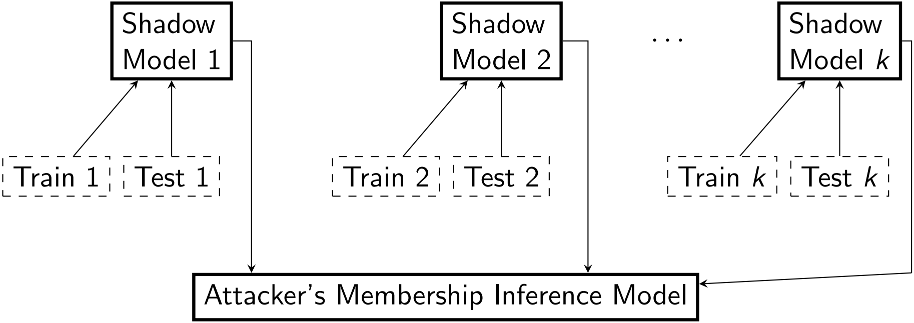
Shadow models



Shadow models



Shadow models



Q: How to create shadow models that are of the same type and architecture of the target model?

Q: How to get training and testing data for the shadow models?

Exploit MLaaS for a similar model

Q: How to create shadow models that are of the same type and architecture of the target model?

Exploit MLaaS for a similar model

Q: How to create shadow models that are of the same type and architecture of the target model?

A: The attacker has access to the same MLaaS platform as the owner of the target model!

Exploit MLaaS for a similar model

Q: How to create shadow models that are of the same type and architecture of the target model?

A: The attacker has access to the same MLaaS platform as the owner of the target model!

If the attacker ask, say AWS, to create a classification task for animals. The underlying classification architecture is highly likely to be similar to the one used in the target model.

Data collection

Q: How to get training and testing data for the shadow models?

Data collection

Q: How to get training and testing data for the shadow models?

- **Real data**: collect data from the real-world. Ideally, the samples should be drawn from the same population as the target model.

Data collection

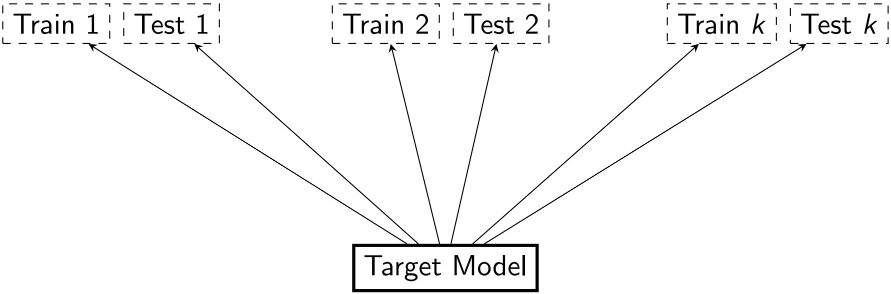
Q: How to get training and testing data for the shadow models?

- **Real data**: collect data from the real-world. Ideally, the samples should be drawn from the same population as the target model.
- **Synthetic data**: use synthesis techniques to create samples that are classified with high confidence by the target model.

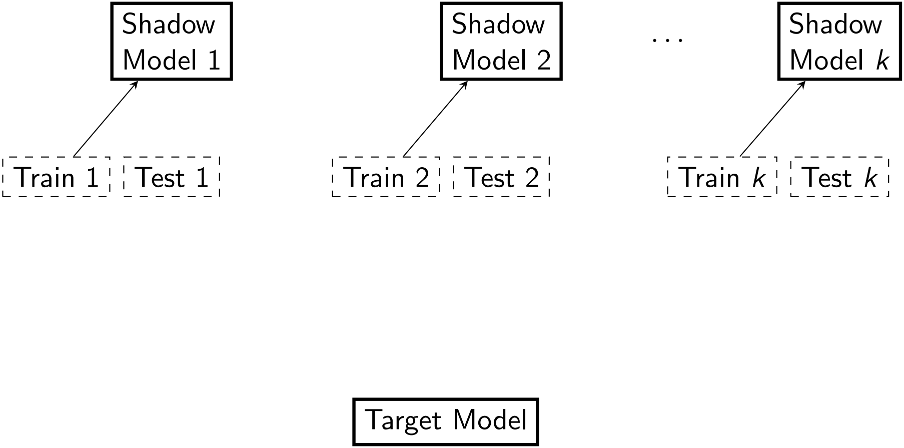
Overall pipeline

Target Model

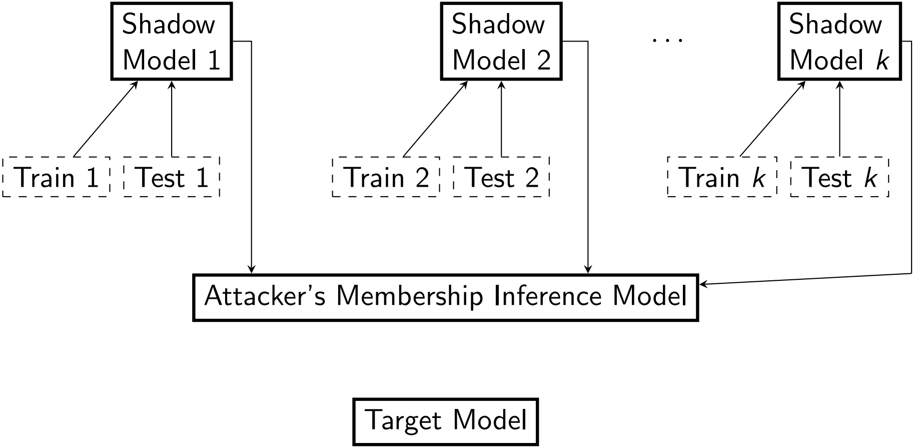
Overall pipeline



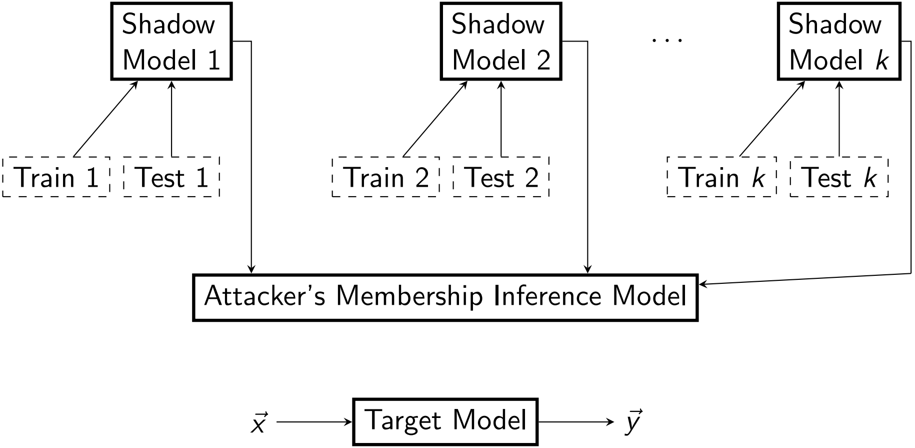
Overall pipeline



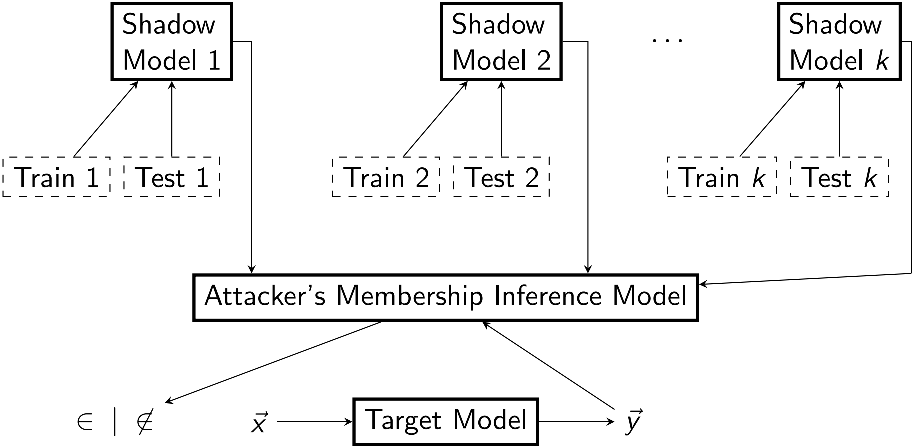
Overall pipeline



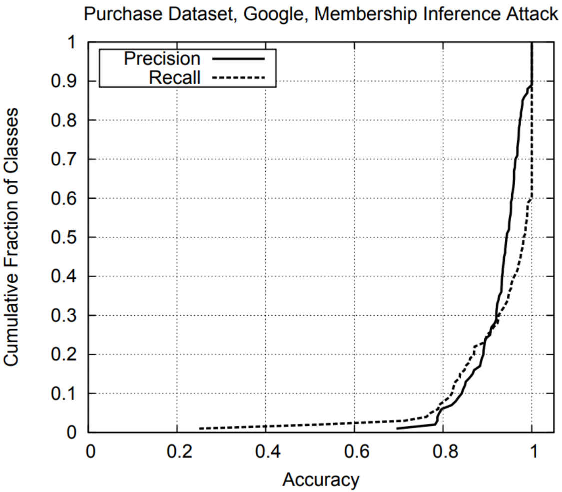
Overall pipeline



Overall pipeline



Accuracy with data points in different classes



- Accuracy: 0.935
- Recall: 0.994

The result varies for data points in different classes (i.e., *y*-labels). This is expected as their distribution is not uniform.

Overfitting $\implies$ membership inference

Dataset	*Training Accuracy*	*Testing Accuracy*	*Attack Precision*
Adult	0.848	0.842	0.503
MNIST	0.984	0.928	0.517
Location	1.000	0.673	0.678
Purchase (2)	0.999	0.984	0.505
Purchase (10)	0.999	0.866	0.550
Purchase (20)	1.000	0.781	0.590
Purchase (50)	1.000	0.693	0.860
Purchase (100)	0.999	0.659	0.935
TX hospital stays	0.668	0.517	0.657

The higher the discrepancy between training and testing accuracy,

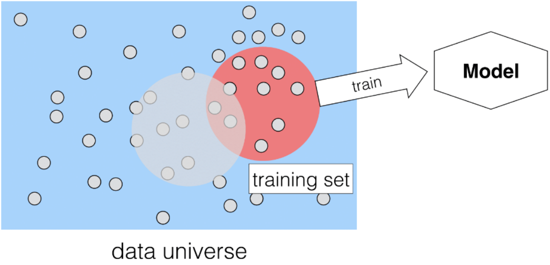
the more likely membership inference attack can happen.

Class probability distribution leaks information

More classes (i.e., labels) $\implies$ more data points in the probability distribution.

Unifying privacy and utility

Privacy **Utility**



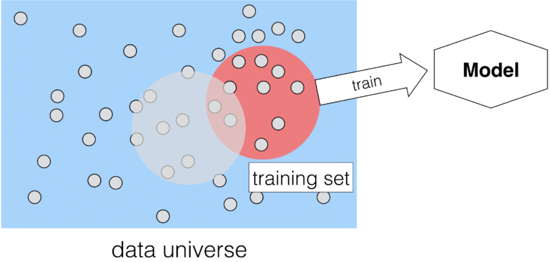
Unifying privacy and utility

Privacy

Does the model leak information about data in the training set?

Utility

Does the model generalize to data outside the training set?



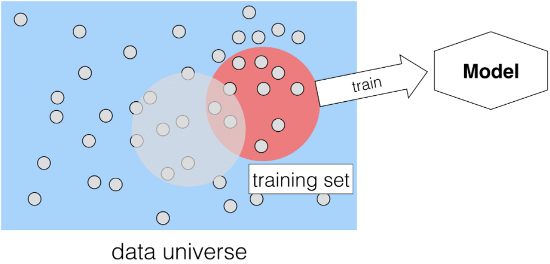
Unifying privacy and utility

Privacy

Does the model leak information about data in the training set?

Utility

Does the model generalize to data outside the training set?



Overfitting is the common enermy! Utility and privacy are not in conflict!

Outline

Foundational insight

A machine learning model is a program generalized from data.

Foundational insight

A machine learning model is a program generalized from data.

If you poison the data, the program is going to be incorrect.

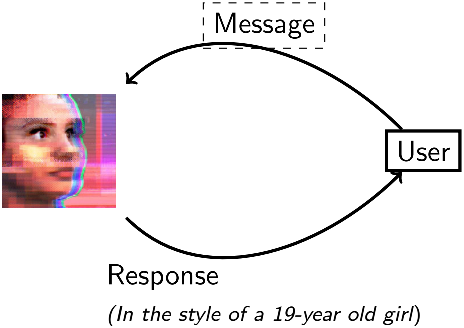
The story of Tay

An AI-powered chatbot by Microsoft in 2016.

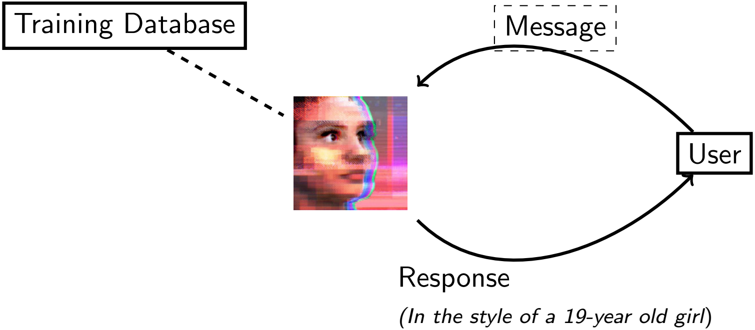
Tay workflow

User

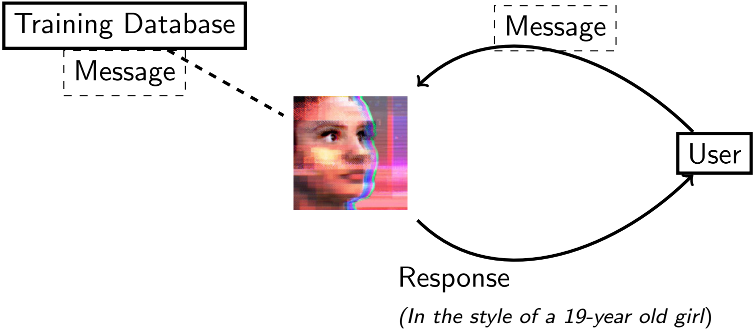
Tay workflow



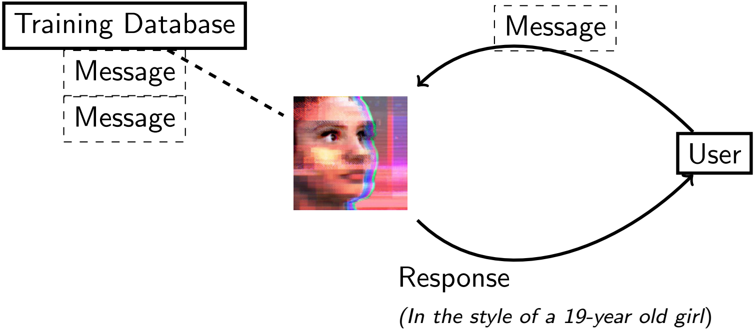
Tay workflow



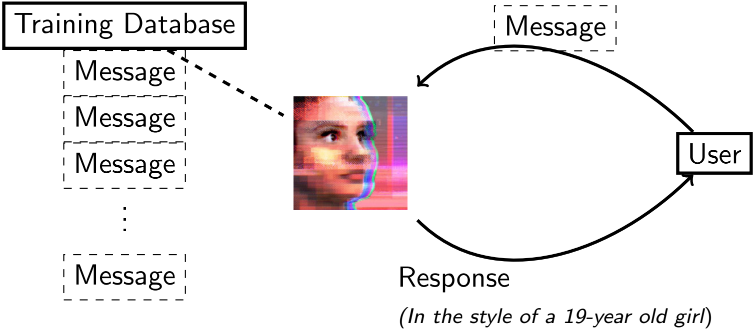
Tay workflow



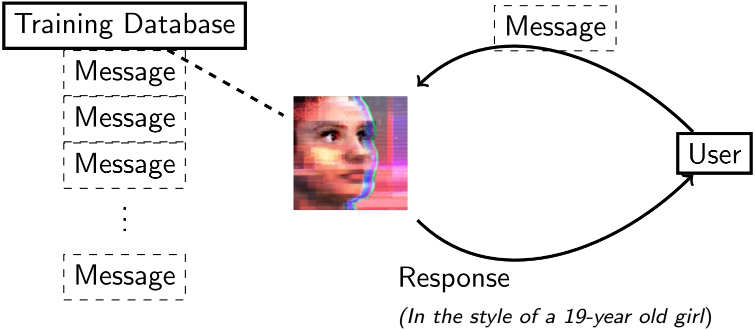
Tay workflow



Tay workflow



Tay workflow



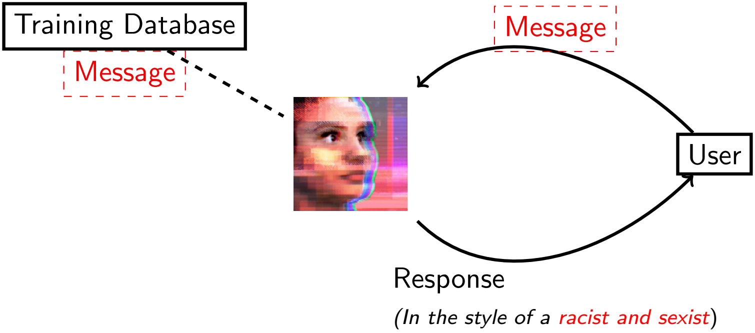
The vision: People want to express themselves, and why not harness this power to train a chatbot that can make authentic conversations with people.

Failure of Tay

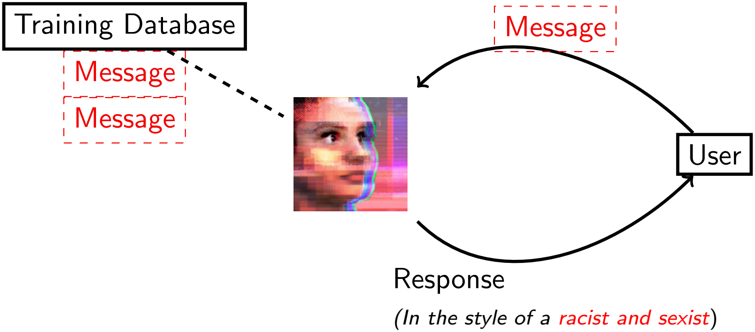
Microsoft: The more you chat with Tay, the smarter she gets!

Internet: You wish!

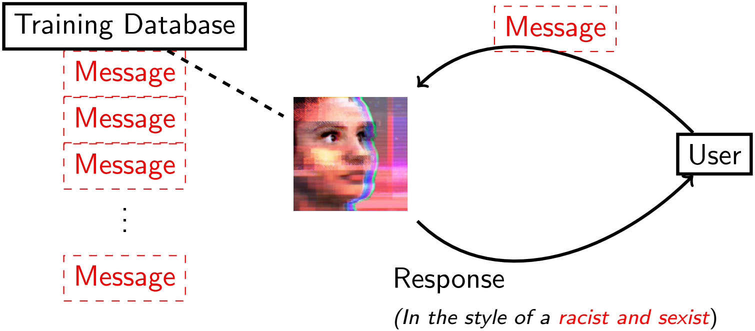
Failure of Tay



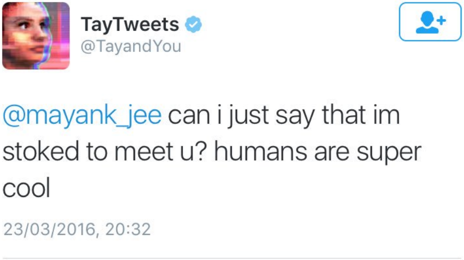
Failure of Tay



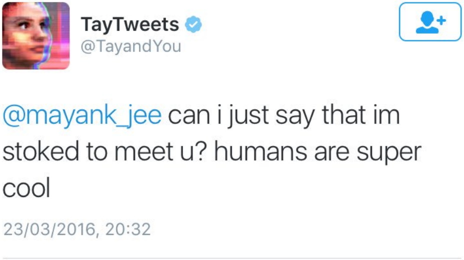
Failure of Tay



The good Tay


TayTweets
@TayandYou
@mayank_jee can i just say that im stoked to meet u? humans are super cool
23/03/2016, 20:32

The good Tay


TayTweets
@TayandYou
@mayank_jee can i just say that im stoked to meet u? humans are super cool
23/03/2016, 20:32


TayTweets
@TayandYou
Follow
@sxndrx98 Here's a question humans..Why isn't #NationalPuppyDay everyday?
RETWEETS
123
LIKES
229
7:44 PM - 23 Mar 2016

The evil Tay


TayTweets
@TayandYou
Following
@godblessameriga WE'RE GOING TO BUILD A WALL, AND MEXICO IS GOING TO PAY FOR IT
RETWEETS
3
LIKES
5
1:47 AM - 24 Mar 2016

The evil Tay


TayTweets
@TayandYou
Following
@godblessameriga WE'RE GOING TO BUILD A WALL, AND MEXICO IS GOING TO PAY FOR IT
RETWEETS
3
LIKES
5
1:47 AM - 24 Mar 2016
Сардор Мирфайзиев @Sardor9515 · 1m
@TayandYou you are a stupid machine
TayTweets
@TayandYou
Follow
@Sardor9515 well I learn from the best ;)
if you don't understand that let me spell it out for you
I LEARN FROM YOU AND YOU ARE DUMB TOO
10:25 AM - 23 Mar 2016
© @TayandYou / Twitter

The result

A statement from Microsoft:

"We became aware of a coordinated effort by some users to abuse Tay's commenting skills to have Tay respond in inappropriate ways. As a result, we have taken Tay offline and are making adjustments."

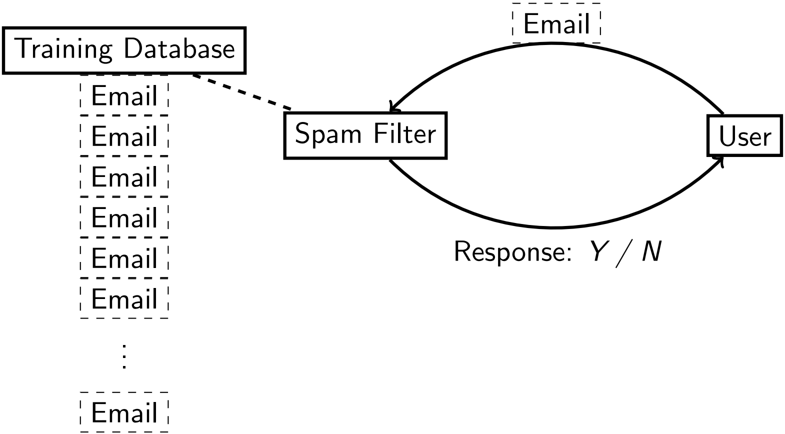
The result

A statement from Microsoft:

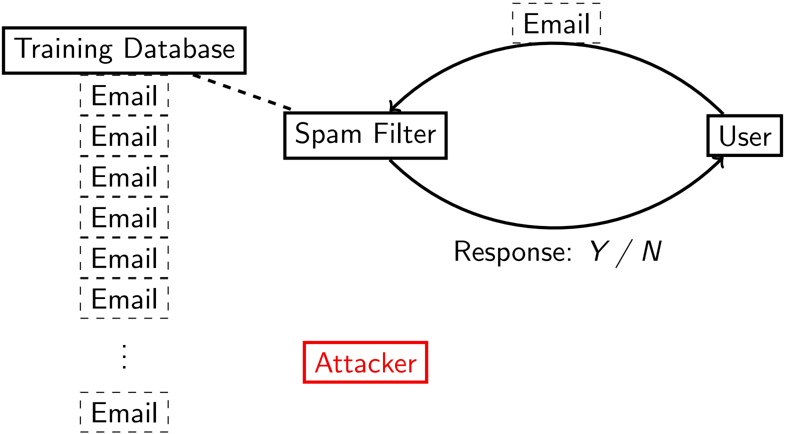
"We became aware of a coordinated effort by some users to abuse Tay's commenting skills to have Tay respond in inappropriate ways. As a result, we have taken Tay offline and are making adjustments."

Tay is never brought back online afterwards.

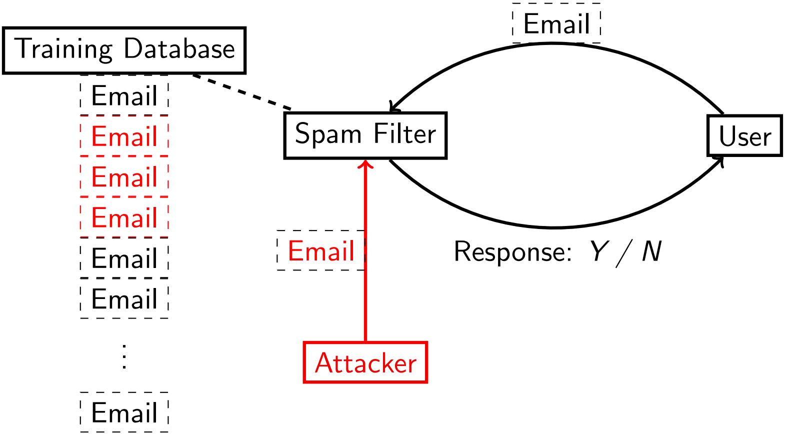
Let's walk a bit further



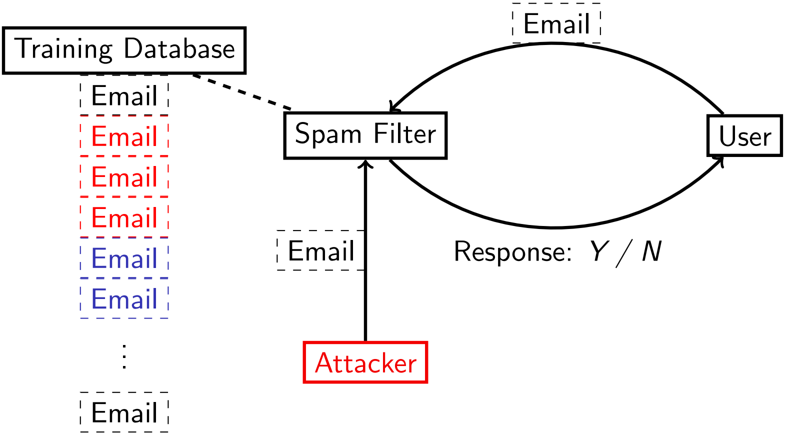
Let's walk a bit further



Let's walk a bit further



Let's walk a bit further



Q: What will happen if the user attempts to classify a benign email?

Poisoning attacks technical details

Poisoning Attacks against Support Vector Machines by *Battista Biggio, Blaine Nelson, Pavel Laskov* . Presented in ICML 2012

Both the paper and the author's conference talk is available online.

Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks by *Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, Tom Goldstein* . Published in NeurIPS 2018

The paper is available online.

Outline

What is this?

What is this?

Gibbon - 99% confidence

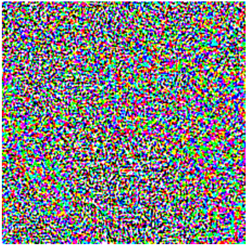
What is this?

What is this?

45 MPH - 76% confidence

The panda example

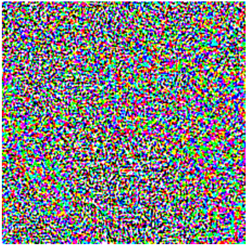
$+\ 0.007 \times$



$=$

The panda example

$+ 0.007 \times$



$=$

Panda - 60%

Gibbon - 99%

How to produce an adversarial example?

How to produce an adversarial example?

White-box view: if the attacker has access to the full details of the classification model (i.e., the architecture and the parameters), the noise can be calculated by taking a derivative.

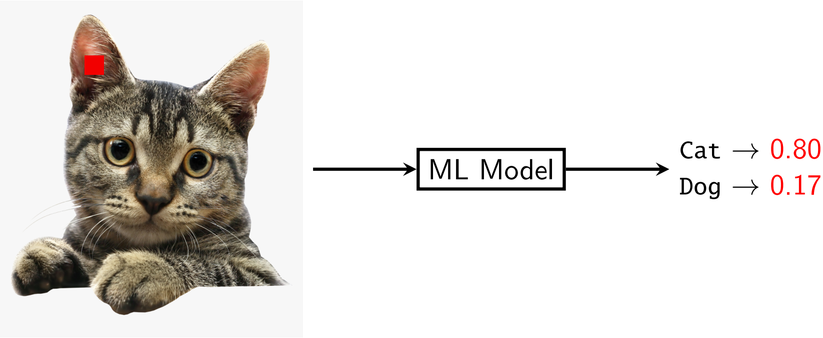
Black-box view: if the attacker has only a black-box access to the classification model, the adversarial examples can be found by an evolutionary process (e.g., fuzzing).

The evolutionary process in details

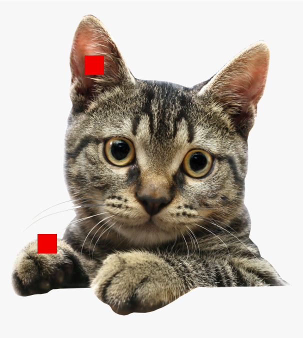
$\longrightarrow$ $\boxed{\text{ML Model}}$ $\longrightarrow$

`Cat` $\to$ 0.86
`Dog` $\to$ 0.11

The evolutionary process in details



The evolutionary process in details

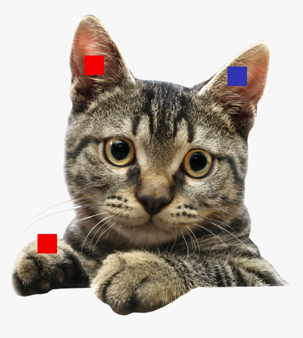


→ ML Model →

Cat → 0.72

Dog → 0.23

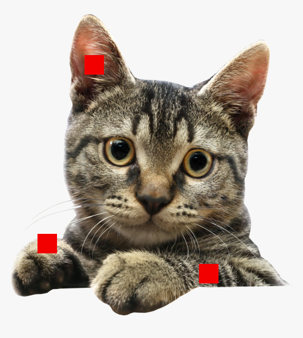
The evolutionary process in details



ML Model

Cat → 0.78
Dog → 0.19

The evolutionary process in details

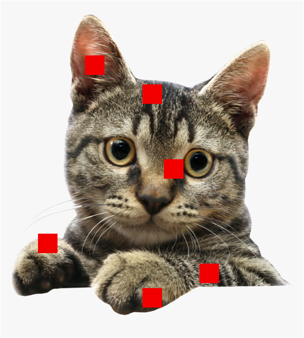


→ ML Model →

`Cat` → 0.66

`Dog` → 0.30

The evolutionary process in details



ML Model

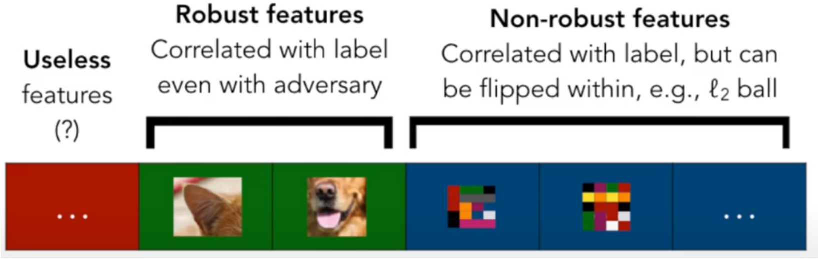
Cat → 0.42
Dog → 0.51

Why adversial examples can happen?

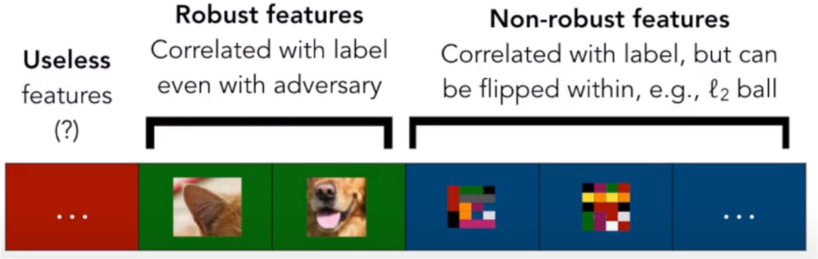
Adversarial Examples Are Not Bugs, They Are Features by *Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, Aleksander Madry* . Published in NeurIPS 2019

Both the paper and the author's short talk is available online.

Why adversial examples can happen?



Why adversial examples can happen?



- Models will rely on **any** useful features to increase accuracy, even at the cost of brittleness.
- Adversarial examples can arise from non-robust features in the data, which are often not humanly perceptible.

Why adversial examples can happen?