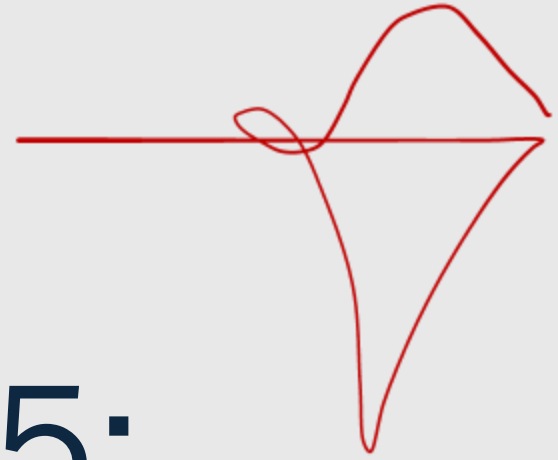


COMP435: *SECURITY CONCEPTS!*

Lecture 28: Differential Privacy, Hardware Security!

tinyurl.com/comp435-fa25

Final Exam Logistics!

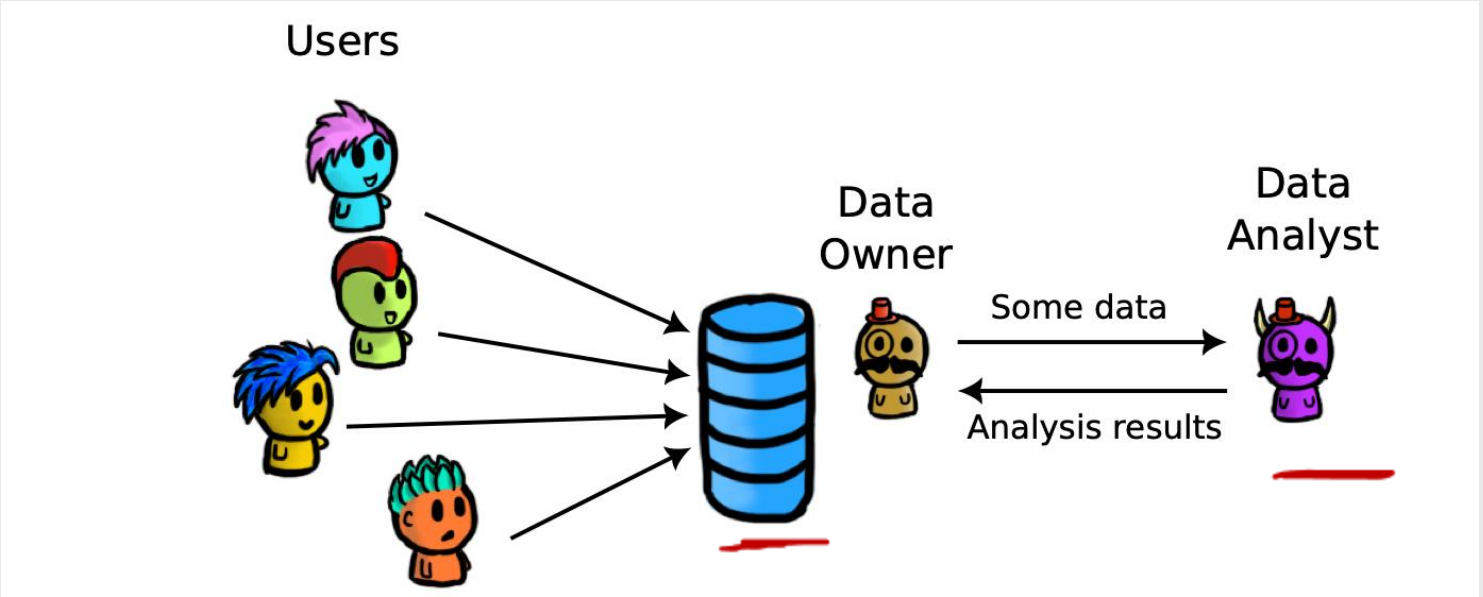
- In this room! **Friday 12/5 8-11am**
- **Assigned seats** will be posted on Canvas the day before
- Test format will be the same as the quizzes
  - *Expected length: ~2.5x quizzes*
- Exam is **cumulative**
  - *Equally distributed across all topics. Including today!*
- Final review session held by TA 12/4. 5pm in SN011
- Going over in-class practice & written assignments will be helpful!
  - *Reading the textbook too! The readings are not long.*

DATA SECURITY & PRIVACY

Adapted from:
https://cs.uwaterloo.ca/~m285xu/courses/cs458-w23/assets/modules/intro/slides.pdf

System Model: Privacy + Utility Tradeoff..

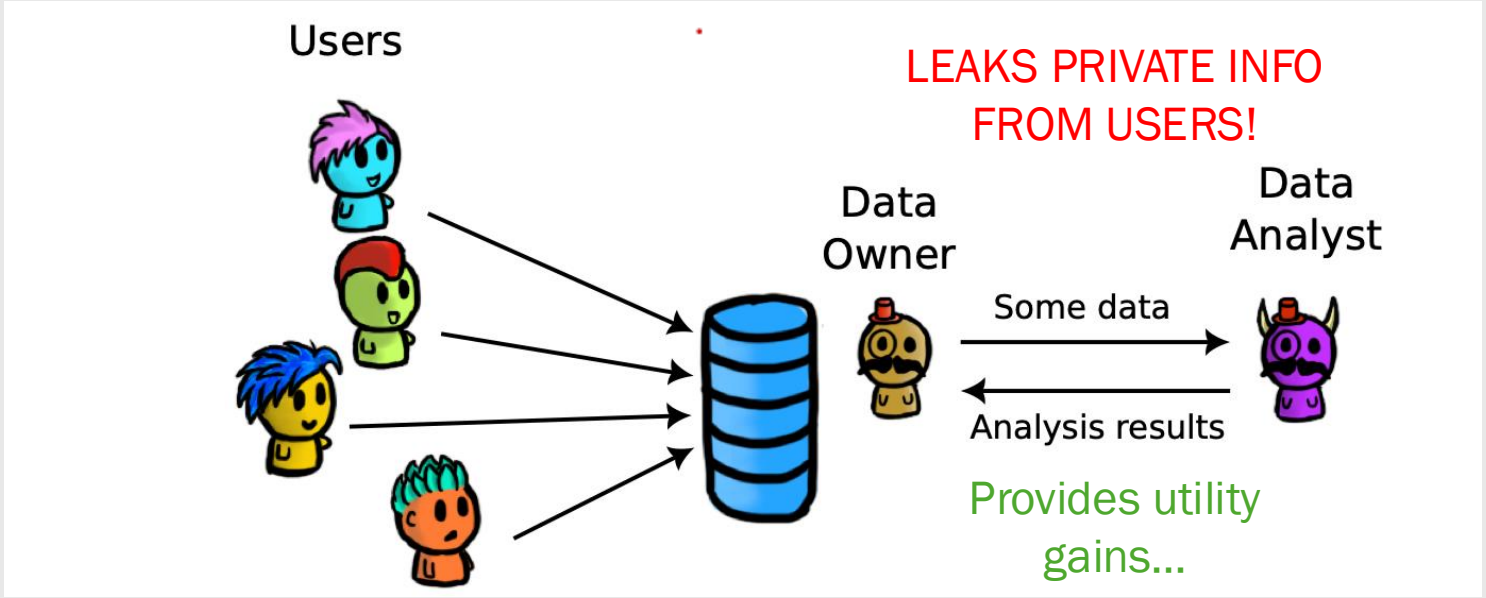
Users provide their data to a data pool.
Admin shares ***a slice of data*** in the pool with a data analyst



https://cs.uwaterloo.ca/~m285xu/courses/cs458-w23/assets/modules/data-sec/part-2-inference-and-syntactic-notions.pdf

System Model: Privacy + Utility Tradeoff..

Users provide their data to a data pool.
Admin shares ***a slice of data*** in the pool with a data analyst

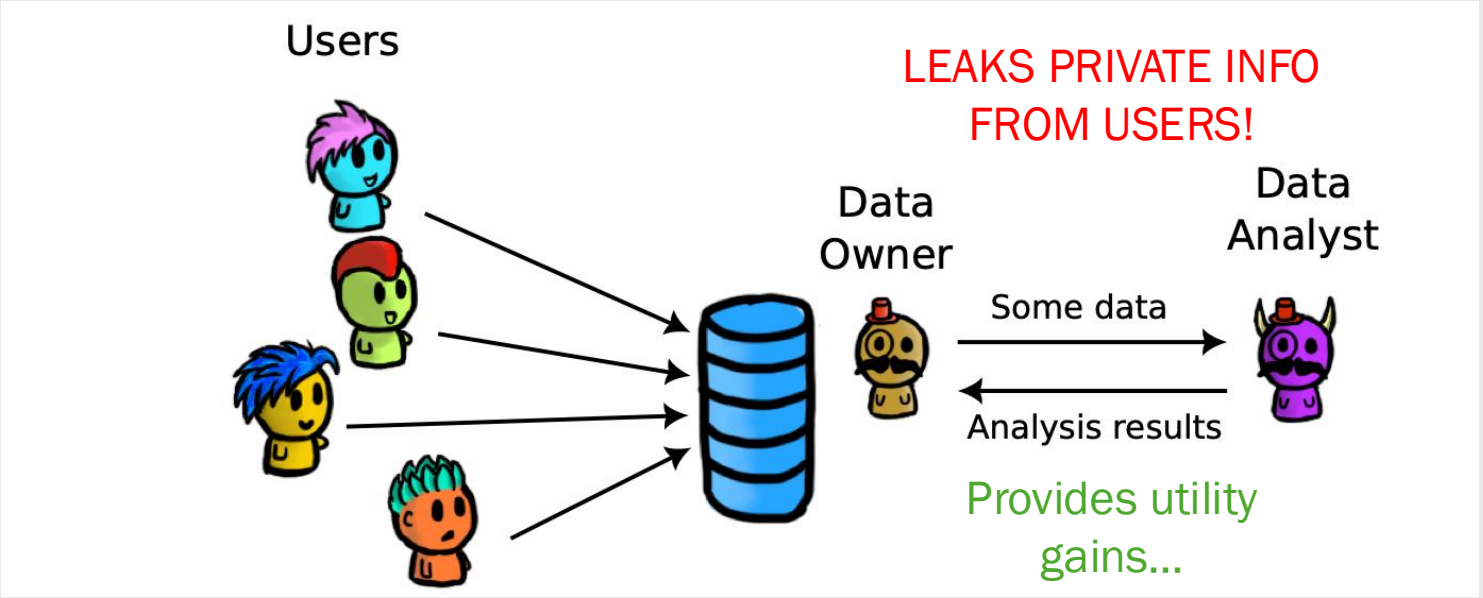


Utility: benefits for both users or the data owner/service provider

Privacy: important for users, it's their data + their right to privacy!

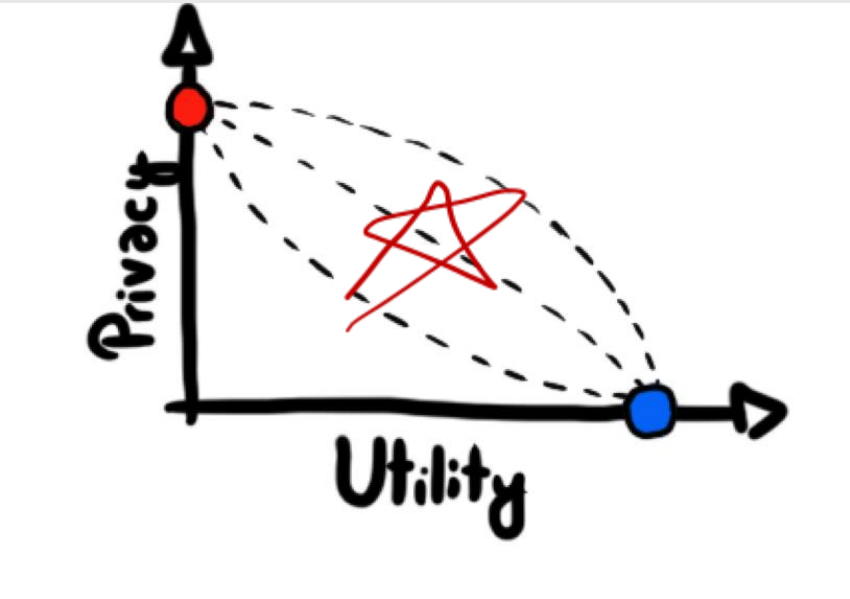
System Model: Privacy + Utility Tradeoff..

Users provide their data to a data pool.
Admin shares ***a slice of data*** in the pool with a data analyst



Q: Any concrete examples that fit this model?

System Model: Privacy + Utility Tradeoff..



Q: easy approaches that fit the red/blue points?

Finding good mechanisms in the middle is hard!!

Some metrics

- There is no perfect metric for privacy and utility that works for every scenario
- **Syntactic notions of privacy** (properties that the published data must follow)
  - k-anonymity
  - ℓ-diversity
  - t-closeness

Some metrics

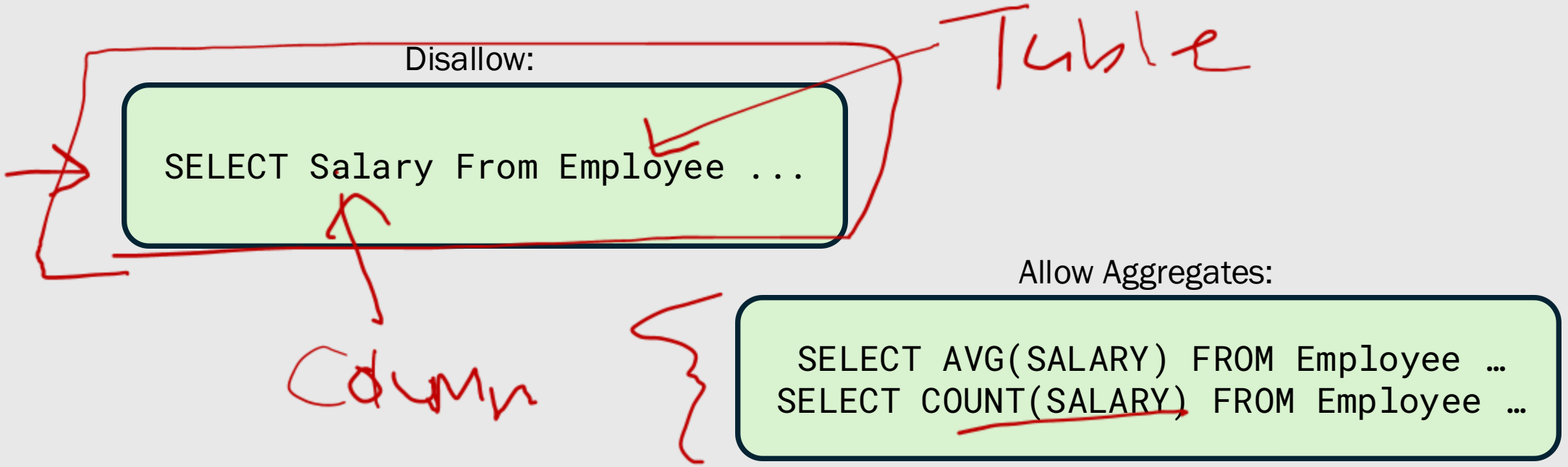
- There is no perfect metric for privacy and utility that works for every scenario
- **Syntactic notions of privacy** (properties that the published data must follow)
  - k-anonymity
  - ℓ-diversity
  - t-closeness
- **Semantic notions of privacy** (properties that the data release mechanism must follow / this is independent of the data that is actually published)
  - **Differential privacy!**

An example: SQL queries

- Scenario: large db with some sensitive attributes
- Utility: we want to allow certain queries
  - Ex: get avg salary of everyone in the company
- Privacy: We also want to protect the privacy of people whose data is in the db
  - Ex: compute avg without revealing each individual's salary

An example: SQL queries

- Scenario: large db with some sensitive attributes
- Utility: we want to allow certain queries
  - Ex: get avg salary of everyone in the company
- Privacy: We also want to protect the privacy of people whose data is in the db
  - Ex: compute avg without revealing each individual's salary



Data inference problem

- Data analysts could **infer** sensitive data, through the output of allowed aggregate queries
  - Inference doesn't have to be a full or accurate recovery either!
  - Ex: determining the range of a specific employee's salary would be considered a leak

- **Goal:** minimize (unintentional) leaks of sensitive data to the data analysts through the allowed queries!

Disallow:

```
SELECT Salary From Employee ...
```

Allow Aggregates:

```
SELECT AVG(SALARY) FROM Employee …
SELECT COUNT(SALARY) FROM Employee …
```

Inference Attack: Single Query

- One single query directly outputs the sensitive data

```
SELECT SUM(Salary) FROM Employee
  WHERE Name = "Alice"
  AND (Gender = "M" OR Gender = "F" OR Gender = "X");
```

Inference Attack: Single Query

- One single query directly outputs the sensitive data

```
SELECT SUM(Salary) FROM Employee
    WHERE Name = "Alice"
    AND (Gender = "M" OR Gender = "F" OR Gender = ”X");
```

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

Inference Attack: Multiple Queries

- How can you infer Alice's Salary in this case?

Name (PK)	Age	Zip	Salary
Alice	32	N2L 0G7	55 000 CAD
Bob	34	N2L 3E4	65 000 CAD
Carol	26	N2L 0E1	35 000 CAD
Dave	24	N2L 2W4	40 000 CAD
...			

Table: Employee (example only)

Inference Attack: Multiple Queries

- How can you infer Alice's Salary in this case?

Name (PK)	Age	Zip	Salary
Alice	32	N2L 0G7	55 000 CAD
Bob	34	N2L 3E4	65 000 CAD
Carol	26	N2L 0E1	35 000 CAD
Dave	24	N2L 2W4	40 000 CAD
...			

Table: Employee (example only)

We can use set theory!

Indirect attack

Q_1: `SELECT SUM(Salary) FROM Employee;` (outputs s)

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

$s - r$ reveals the secret salary.

Inference Attack: Multiple Queries

- How can you infer Alice's Salary in this case?

Name (PK)	Age	Zip	Salary
Alice	32	N2L 0G7	55 000 CAD
Bob	34	N2L 3E4	65 000 CAD
Carol	26	N2L 0E1	35 000 CAD
Dave	24	N2L 2W4	40 000 CAD
...			

Table: Employee (example only)

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?

Name (PK)	Age	Zip	Salary
Alice	?	?	???
⋮	⋮	⋮	
⋮	⋮	⋮	
⋮	⋮	⋮	

Assumptions: Alice is in the dataset, but we don't know anything else. Also, the median age in the company is 30

Template

Q_1: `SELECT SUM(Salary) FROM Employee WHERE ;`
Q_2: `SELECT SUM(Salary) FROM Employee WHERE ;`
Q_3: `SELECT SUM(Salary) FROM Employee WHERE ;`

Inference Attack: Tracker Attack

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

Name (PK)	Age	Zip	Salary
Alice	?	?	???
⋮	⋮	⋮	
⋮	⋮	⋮	
⋮	⋮	⋮	

Assumptions: Alice is in the dataset, but we don't know anything else. Also, the median age in the company is 30

Template

Q_1: `SELECT SUM(Salary) FROM Employee WHERE ;`
Q_2: `SELECT SUM(Salary) FROM Employee WHERE ;`
Q_3: `SELECT SUM(Salary) FROM Employee WHERE ;`

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:

```
SELECT SUM(Salary) FROM Employee WHERE Age < 30;
```

For genericity, we use **C** to represent the (Age < 30) 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:

```
SELECT SUM(Salary) FROM Employee WHERE Age < 30;
```

For genericity, we use **C** to represent the (Age < 30) 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 = "Alice" OR` *C*`;`

Q_2: `SELECT SUM(Salary) FROM Employee WHERE Name = "Alice" OR NOT` *C*`;`

Q_3: `SELECT SUM(Salary) FROM Employee;`

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

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:

```
SELECT SUM(Salary) FROM Employee WHERE Age < 30;
```

For genericity, we use **C** to represent the (Age < 30) constraint that makes **T** to include a proper number of records.

And this query T is called a tracker.

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

Tracker attack

Q_1: `SELECT SUM(Salary) FROM Employee WHERE Name = "Alice" OR C;`

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

Q_3: `SELECT SUM(Salary) FROM Employee;`

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

Inference Across Multiple Sources

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! ☹

More severe b/c access controls rarely apply across data sources

Inference Across Multiple Sources

Where does an adversary get external data sources?

- Publicly available data (ex. Census, regional records)
- Purchasing it from a data broker
- Governments might share dossiers with each other
- Large companies may collect info about their customers

- If these contain identifiers that are persistent pseudonyms that can link records across datasets… we can learn more about individuals & entities!

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

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]

How could you uncover their identities?

Anonymity failure: Massachusetts Health Insurance 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!

Privacy vs. Utility Tradeoff

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. ⇒ 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)

k-anonymity

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

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

Removing a quasi-identifier

- e.g., removing the gender attribute

Reducing the granularity

- e.g., hiding the last characters of a ZIP code or the day from a DOB

Grouping quasi-identifiers

- e.g., reporting age ranges, instead of actual ages

k-anonymity

3-anonymity

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

ZIP	Party affiliation
N1CFFA	Green Party
G0ANFA	Liberal Party
N1C5YN	Green Party
N2J0HJ	Conservative Party
N1C4KH	Green Party
G0A3G4	Conservative Party
G0A3GN	Liberal Party
N2JWBV	New Democratic Party
N2JWBV	Liberal Party

ntifiers suc

he day fro

s

ZIP	Party affiliation
N1C***	Green Party
G0A***	Liberal Party
N1C***	Green Party
N2J***	Conservative Party
N1C***	Green Party
G0A***	Conservative Party
G0A***	Liberal Party
N2J***	New Democratic Party
N2J***	Liberal Party

k-anonymity: issues!

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***	196*-**-**	Liberal Party
G0A***	196*-**-**	Liberal Party
G0A***	196*-**-**	Conservative Party
N1C***	199*-**-**	Conservative Party
N1C***	199*-**-**	New Democratic Party
N1C***	199*-**-**	Liberal Party

k-anonymity: issues!

f you know Bob (G0A***, 196*-**-**) 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***	196*-**-**	Liberal Party
G0A***	196*-**-**	Liberal Party
G0A***	196*-**-**	Conservative Party
N1C***	199*-**-**	Conservative Party
N1C***	199*-**-**	New Democratic Party
N1C***	199*-**-**	Liberal Party

ℓ-diversity

ℓ-diversity: For any quasi-identifier value, there should be at least ℓ distinct values of the sensitive fields

This table is 2-diversified

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

ℓ-diversity: issues!

ℓ-diversity: For any quasi-identifier value, there should be at least ℓ distinct values of the sensitive fields

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

ℓ-diversity: issues!

ℓ-diversity: For any quasi-identifier value, there should be at least ℓ distinct values of the sensitive fields

If you know David who who is in his twenties is in this table, what will you learn?

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

What went wrong?

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

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 does it mean for something to be “unexpected”?

The “unexpected” feeling comes from the distribution of
sensitive values of the whole dataset being different than the
distribution of the sensitive values per class.

i.e., 5% of positive rate
overall vs 98% of positive rate in the first group.

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

What does it mean for something to be “unexpected”?

The “unexpected” feeling comes from the distribution of
sensitive values of the whole dataset being
different than the
distribution of the
class.

i.e., 5% of positive
overall vs 98% of p
group.

ZIP	DOB	Virus X Test
N3P***	199*-**-**	Positive
N3P***	199*-**-**	Positive

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

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.

Privacy is measured by the information gain of an observer

This gain is the difference between some prior belief (before seeing the data) and a posterior belief (after seeing the data)

Limitations of Syntactic Measures

- 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

DIFFERENTIAL PRIVACY

Adapted from:
https://cs.uwaterloo.ca/~m285xu/courses/cs458-w23/assets/modules/intro/slides.pdf

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.

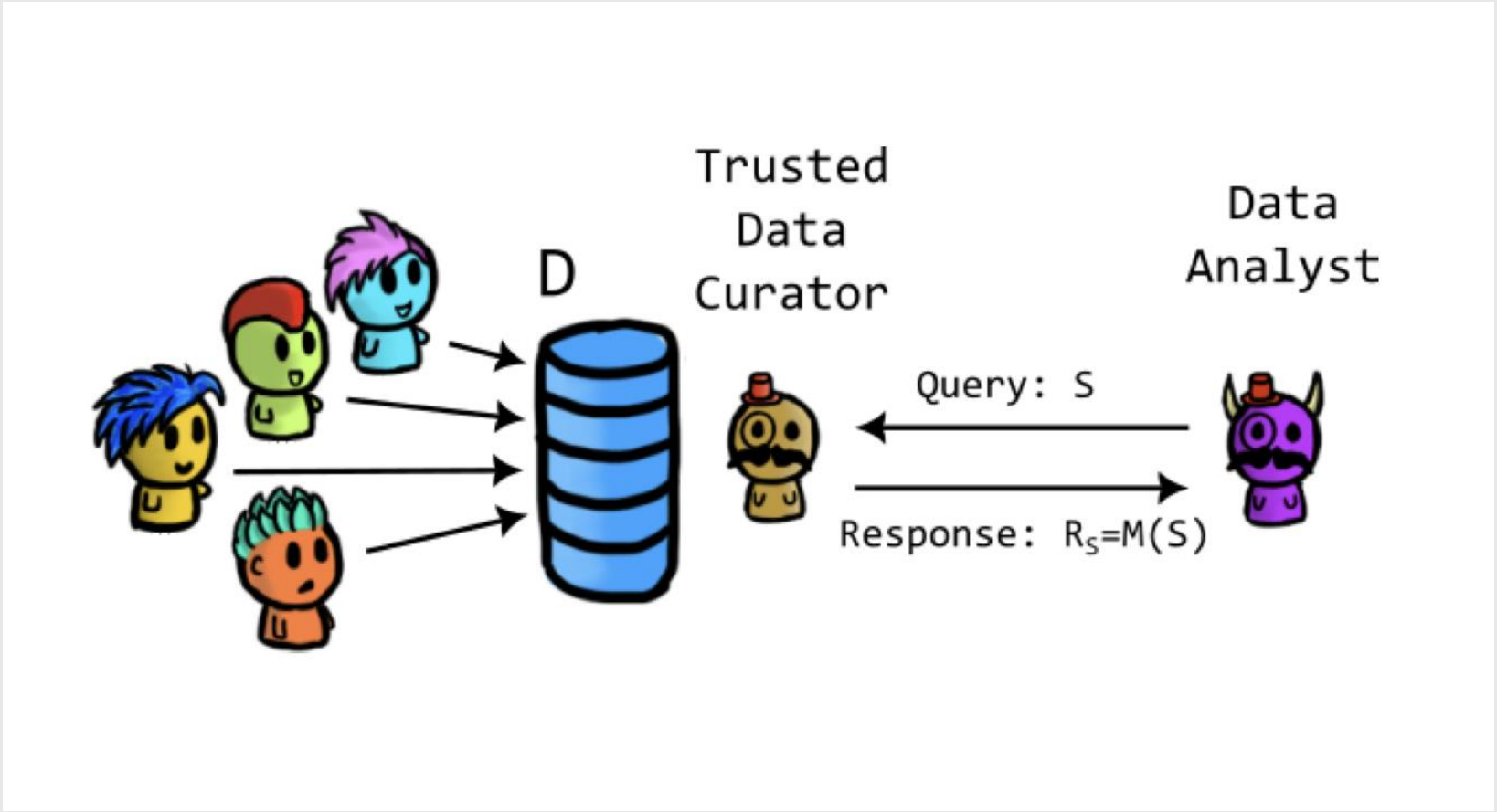
The data analyst consumes the data by asking a series of queries to the curator. Each query is denoted as S and the curator provides a response to query S with R_S.

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 mechnism that

- gives statistically useful responses but
- avoids leaking sensitive information about individuals.

Our setup



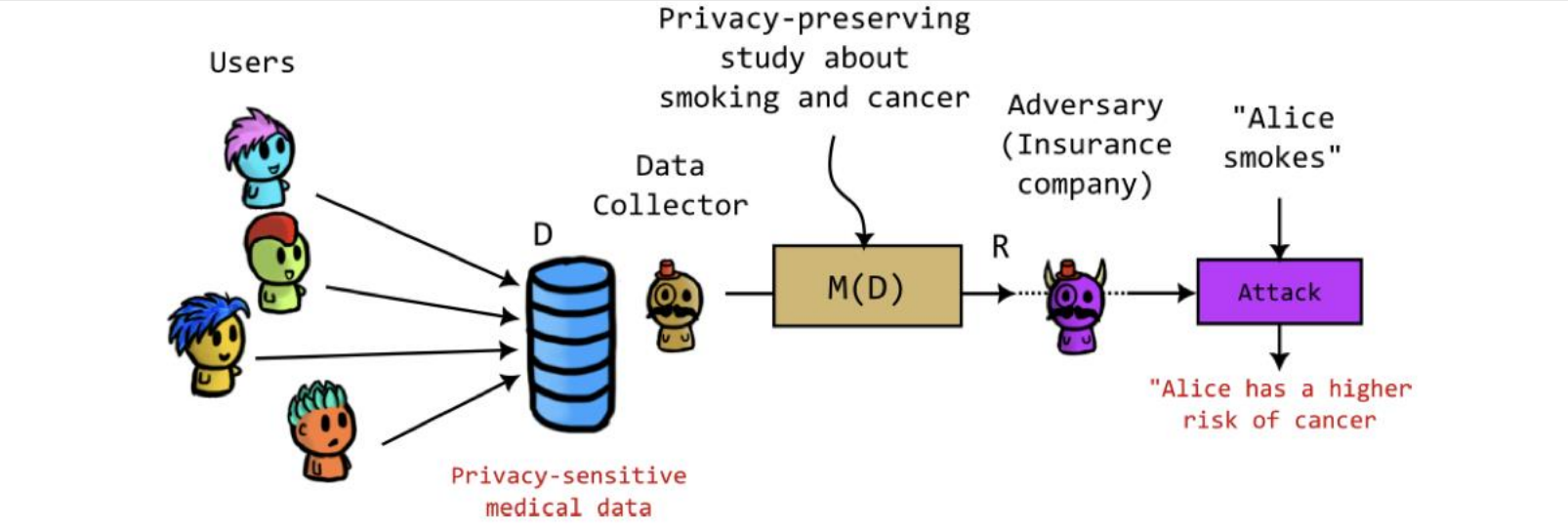
Bad News: Adding Noise is Tricky!

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.

Such a mechanism is called 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.

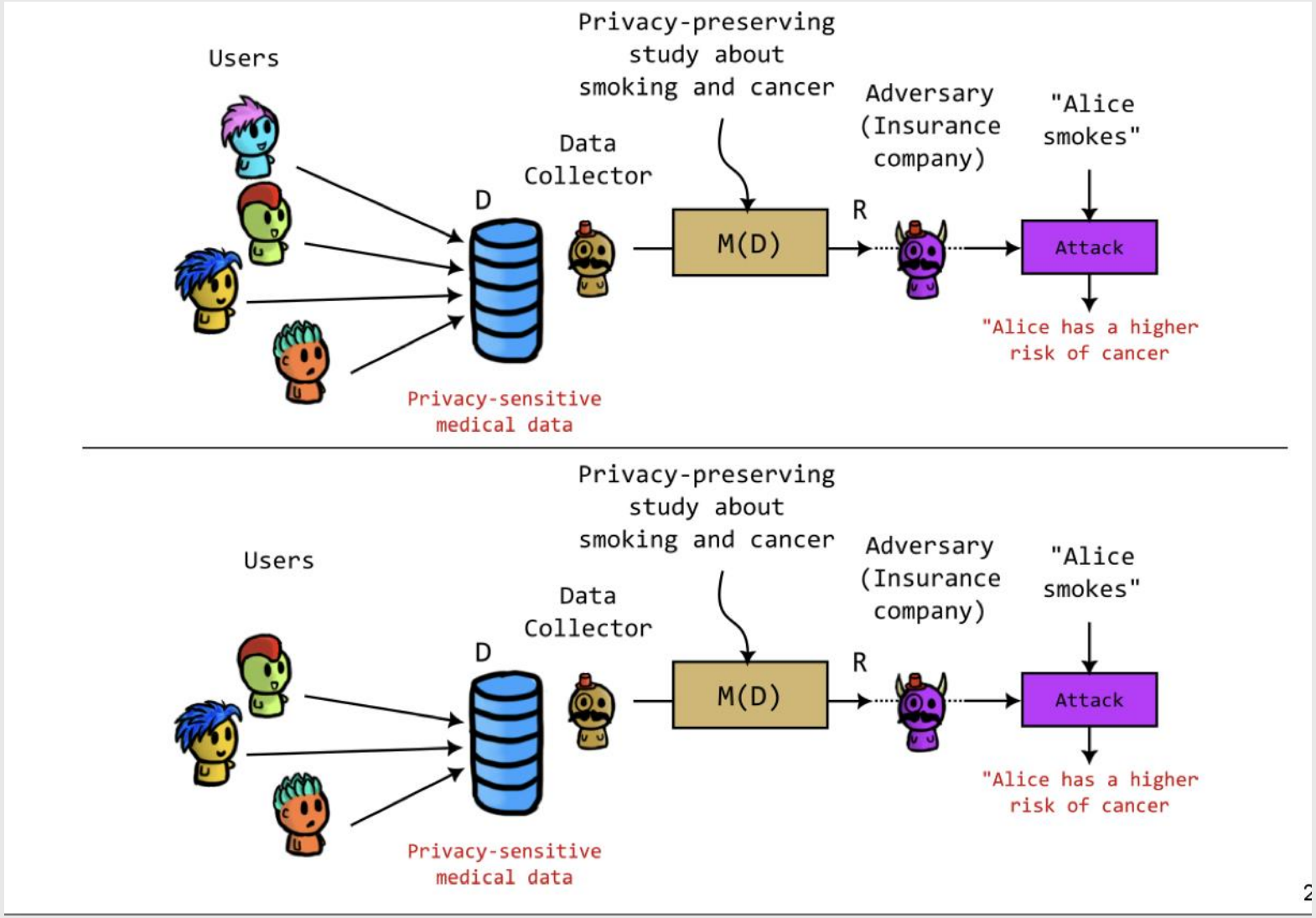
Example: strong auxiliary information



A study proved that smoking and cancer are correlated. Thanks to the study, the adversary learns that Alice has higher risk of cancer.

Q: Is this a violation of Alice's privacy? Is this the study's fault? Should we design an M to prevent this?

Example: strong auxiliary information

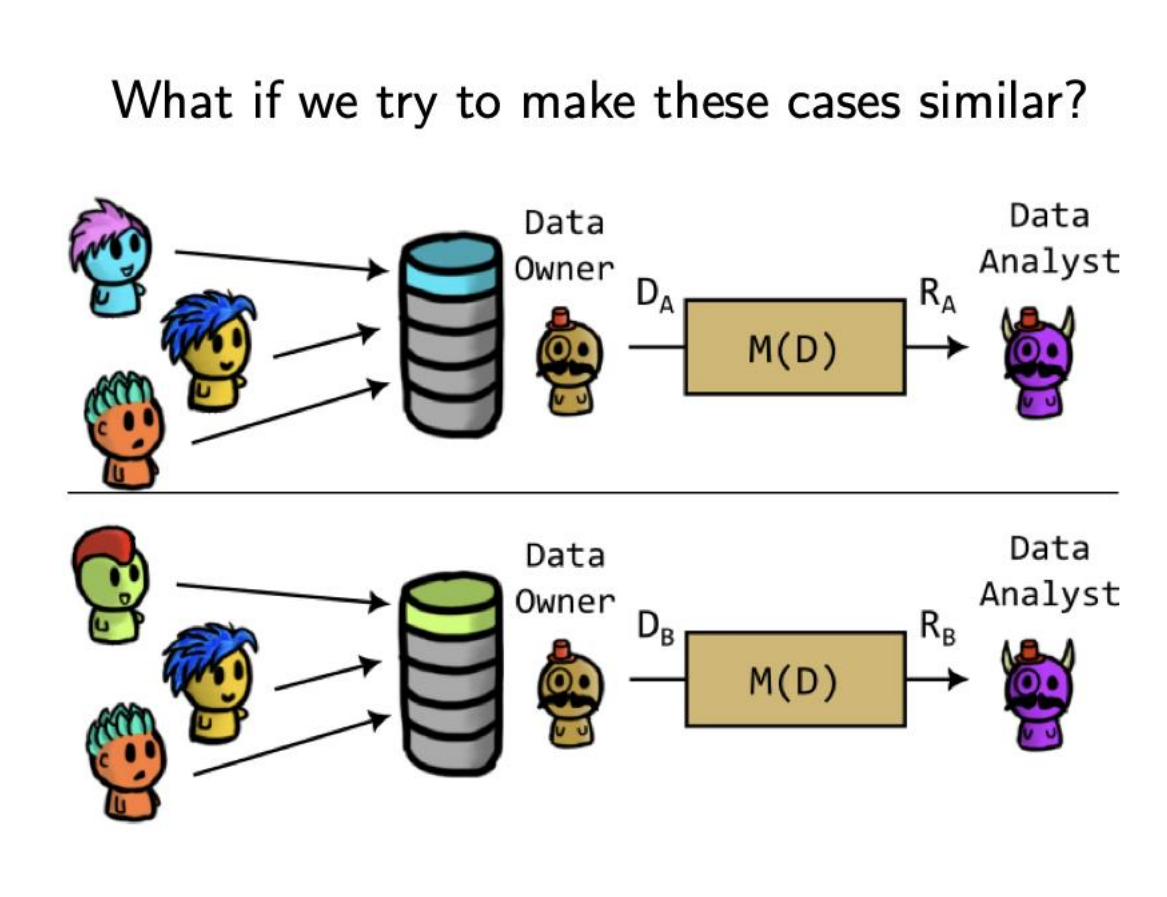


Possible Privacy Goal...

We cannot guarantee absolute privacy — if the adversary has sufficiently strong background information, there is nothing M can do about it!

We should instead ensure that the adversary cannot gain (significantly) new information from R (i.e., we want a “differential” and not an “absolute” privacy)

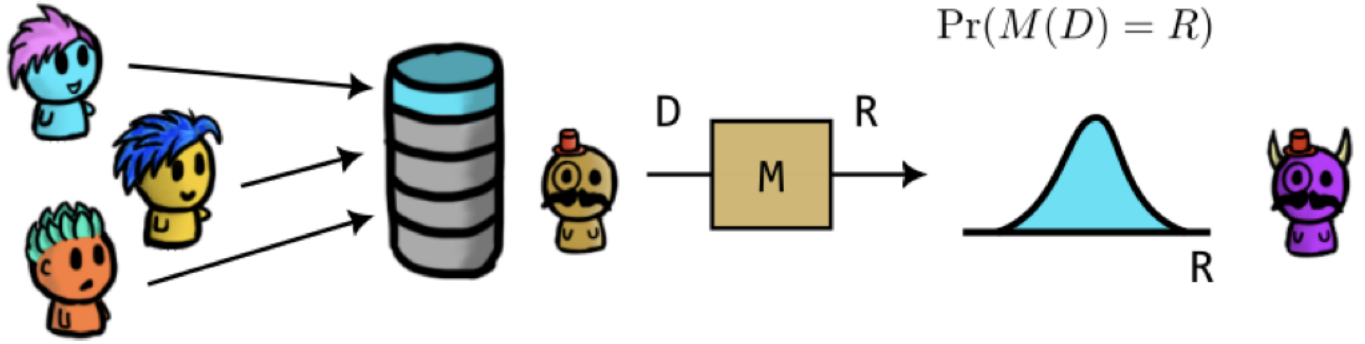
Possible Privacy Goal...

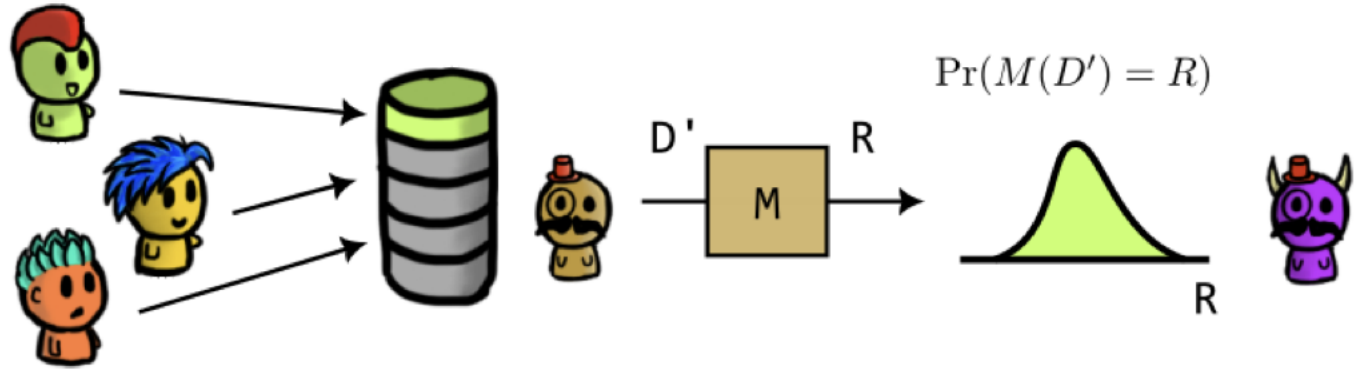


R_a and R_b would be similar. Mechanism doesn't depend "too much" on any single user!

Possible Privacy Goal...

In addition, note that M is randomized (e.g., adds noise). Thus, instead of ensuring $R_A \approx R_B$, we ensure their probability distributions are "similar".





i.e., for all R, the chance of producing R by D and D' are close:
$\Pr(M(D) = R) \approx \Pr(M(D') = R)$

Formalizing this a bit

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: 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.

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: COMP435 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).

How can you infer if Alice is in the class or not?

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

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.

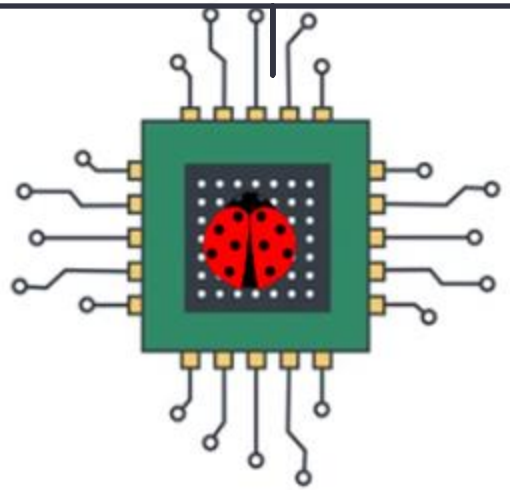
Background knowledge 2: COMP435 is challenging and historical records show that most students score in the range of [45, 55].

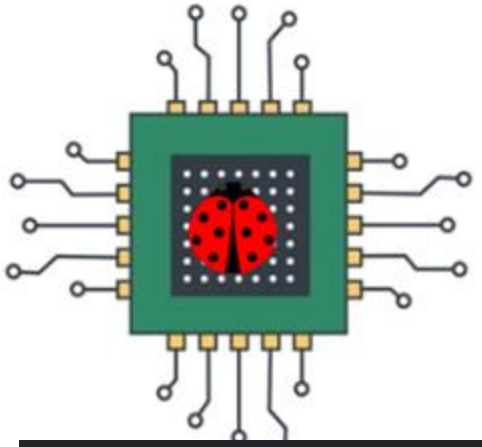
Algorithm: You are given an algorithm that
allows you to make 5 queries, each query returns the average score of 3 randomly selected students (out of 30 scores in total) plus a random value ☺

HARDWARE SECURITY ☺

Our lives depend on secure systems!

Hardware is the root of trust in all systems!





Hardware Backdoor Discovered in RFID Cards Used in Hotels and Offices Worldwide

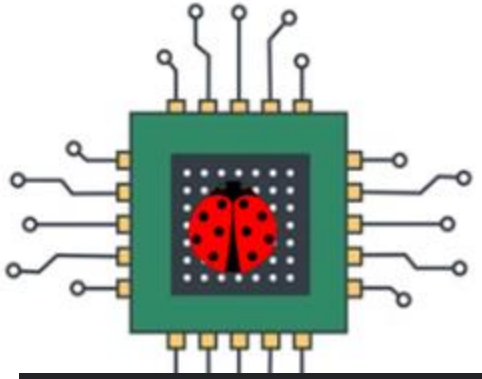
Aug 22, 2024 Ravie Lakshmanan

Unpatchable vulnerability in Apple chip leaks secret encryption keys

Fixing newly discovered side channel will likely take a major toll on performance.

Hardware vulnerabilities in Hitachi Energy, ABB, B&R ICS devices pose critical infrastructure threat

APRIL 07, 2025



Hardware Backdoor Discovered in RFID Cards Used in Hotels and Offices Worldwide

Aug 22, 2024 Ravie Lakshmanan

Unpatchable vulnerability in Apple chip leaks secret encryption keys

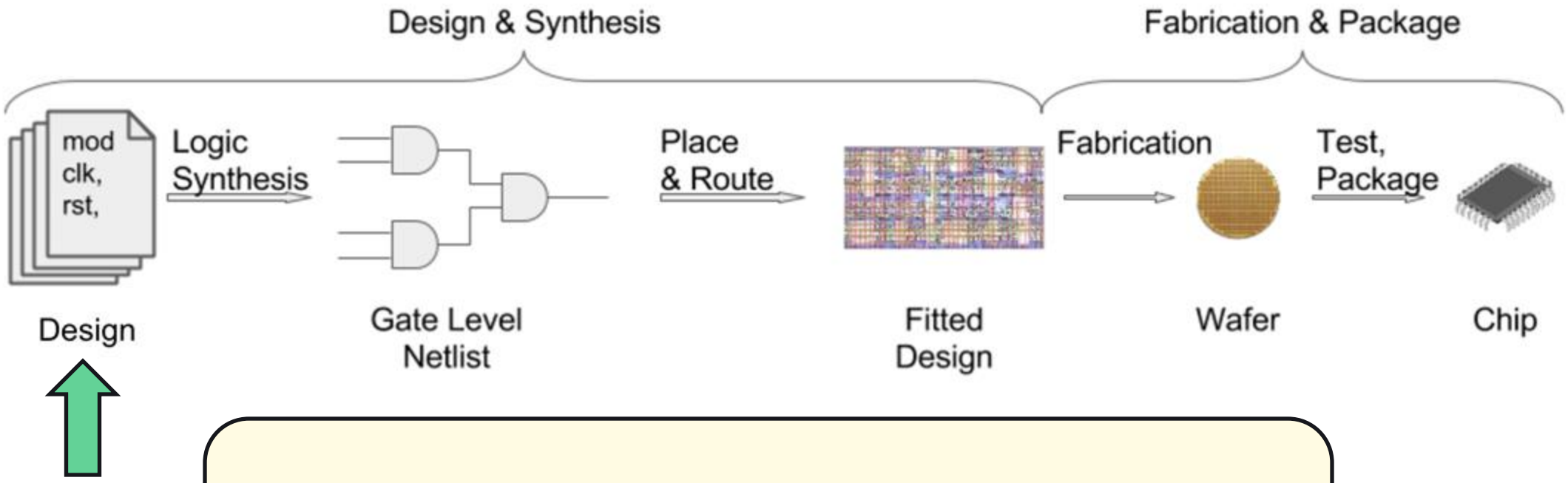
Fixing newly discove

High-assurance verification is critical!

Hardware vulnerabilities in Hitachi Energy, ABB, B&R ICS devices pose critical infrastructure threat

APRIL 07, 2025

The Problem(s)…



Hardware bugs are hard to patch post-deployment!

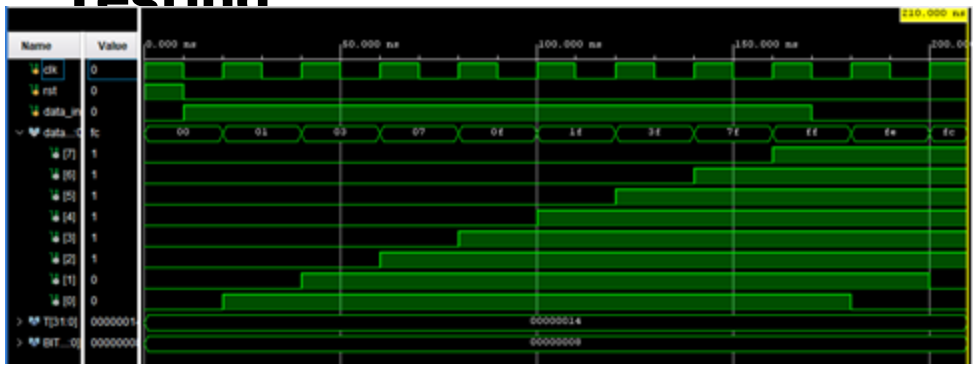
The Problem(s)…

Increasing design complexity!

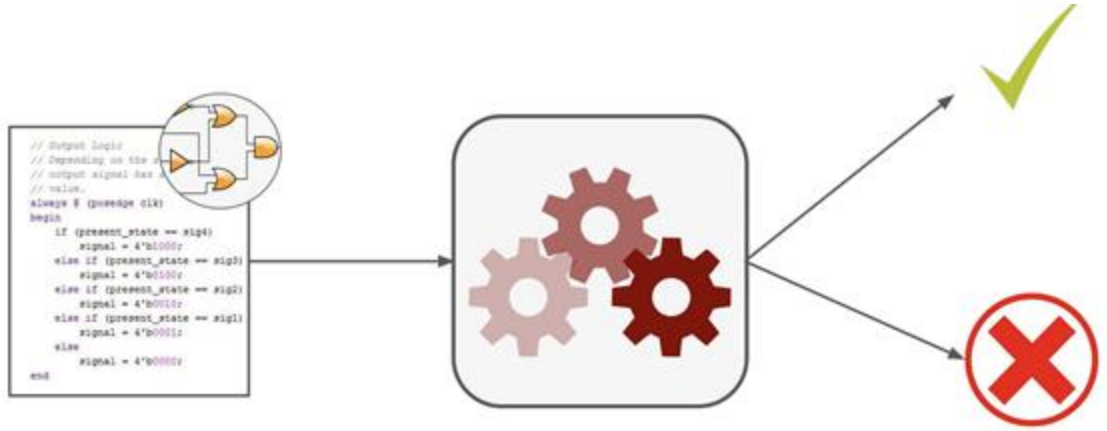
Number of Transistors in CPUs over Time

Current State of the Art in Hardware Verification

Simulation Based Testing



Model Checking



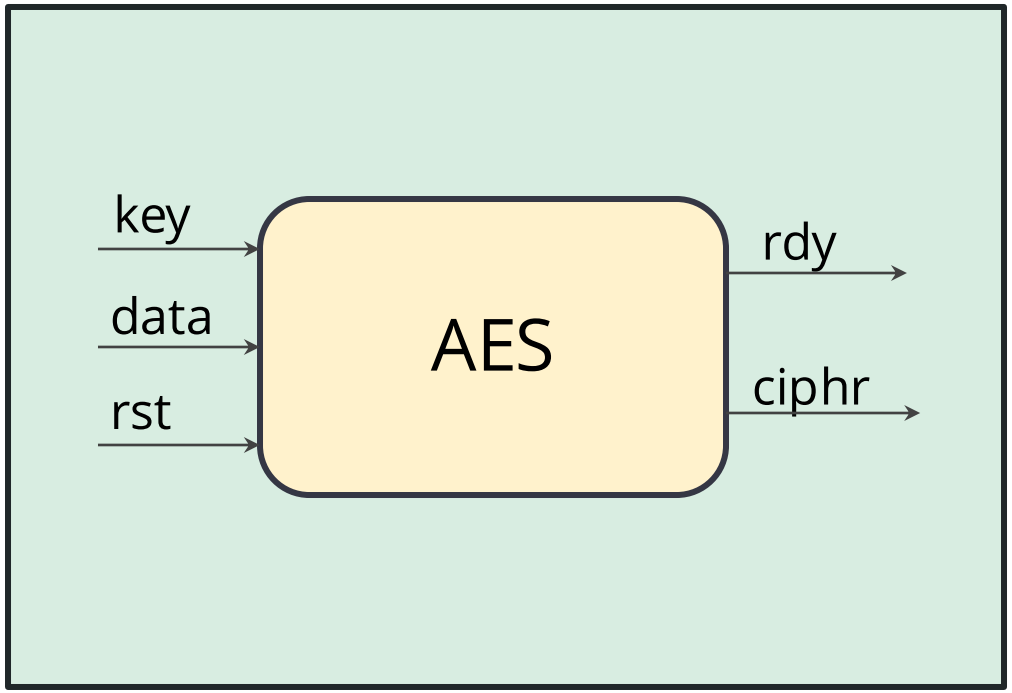
Quick Background: Information Flow

```
1  input  A
2  output E

3  reg B, C, D;

4  assign E = D;

5  if (A)
6    D <= B;
7  else
8    D <= C;
```

Line 6 shows an **explicit flow** from B to D.

Lines 5-6 shows an **implicit flow** from A to D.

Information Flow Property Verification

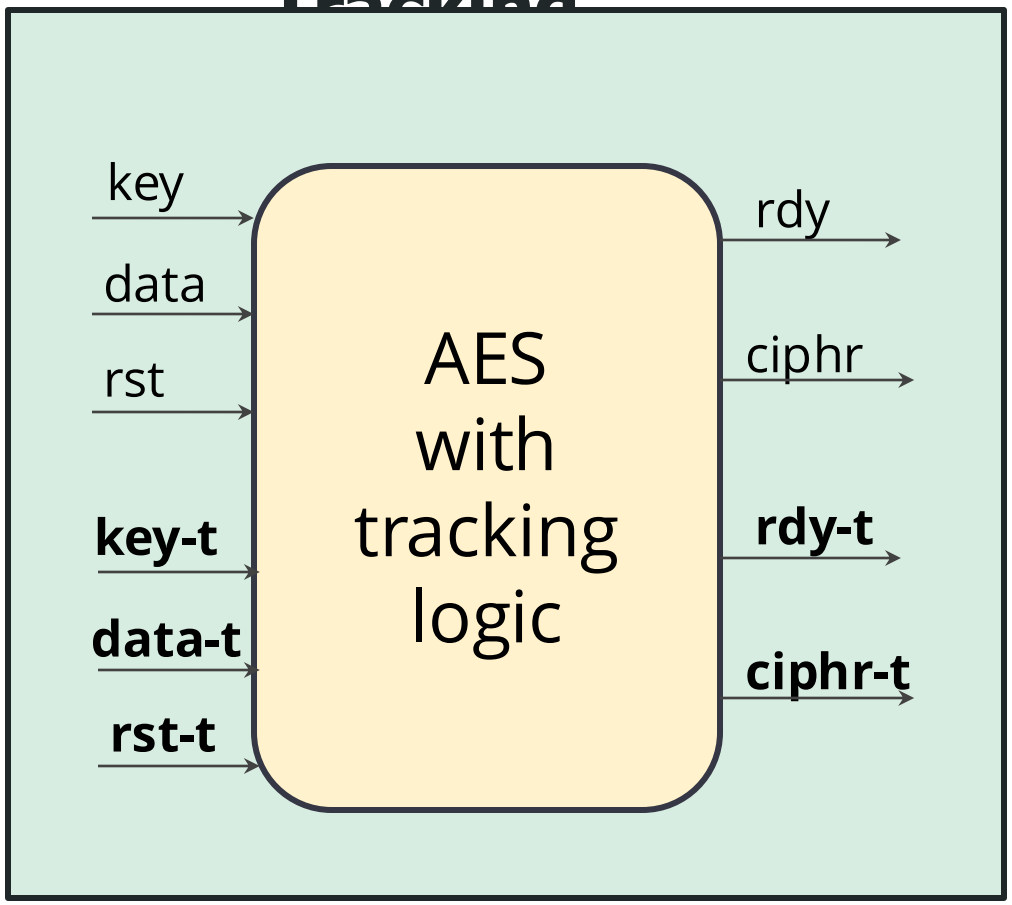


φ: key $\not\rightarrow$ rdy

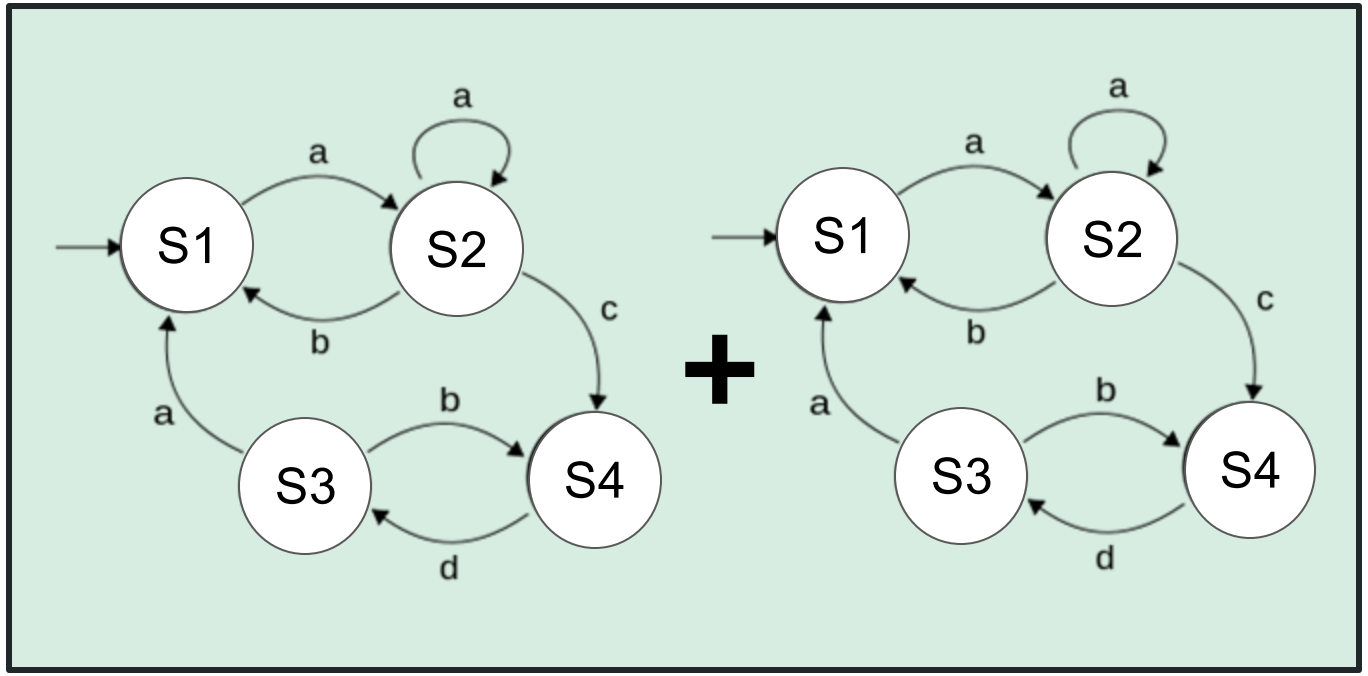
Information Flow Property

Current State of the Art

Taint Tracking

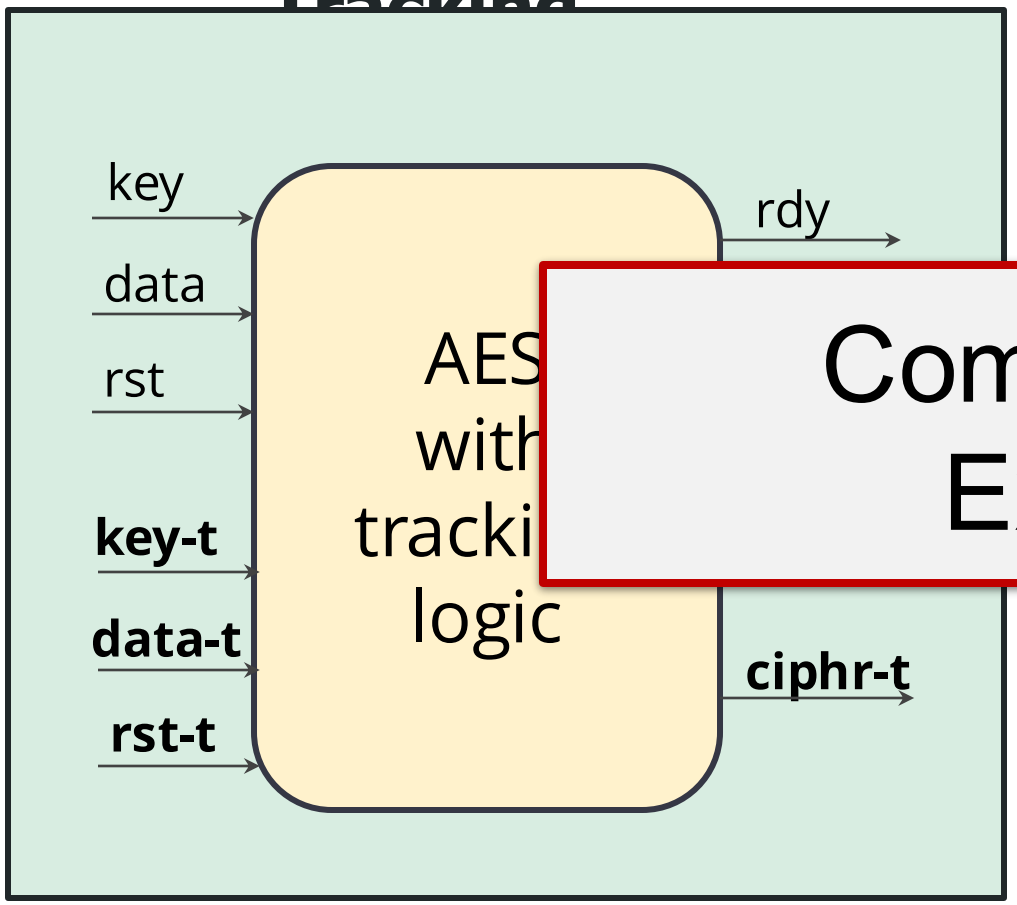


Model Checking

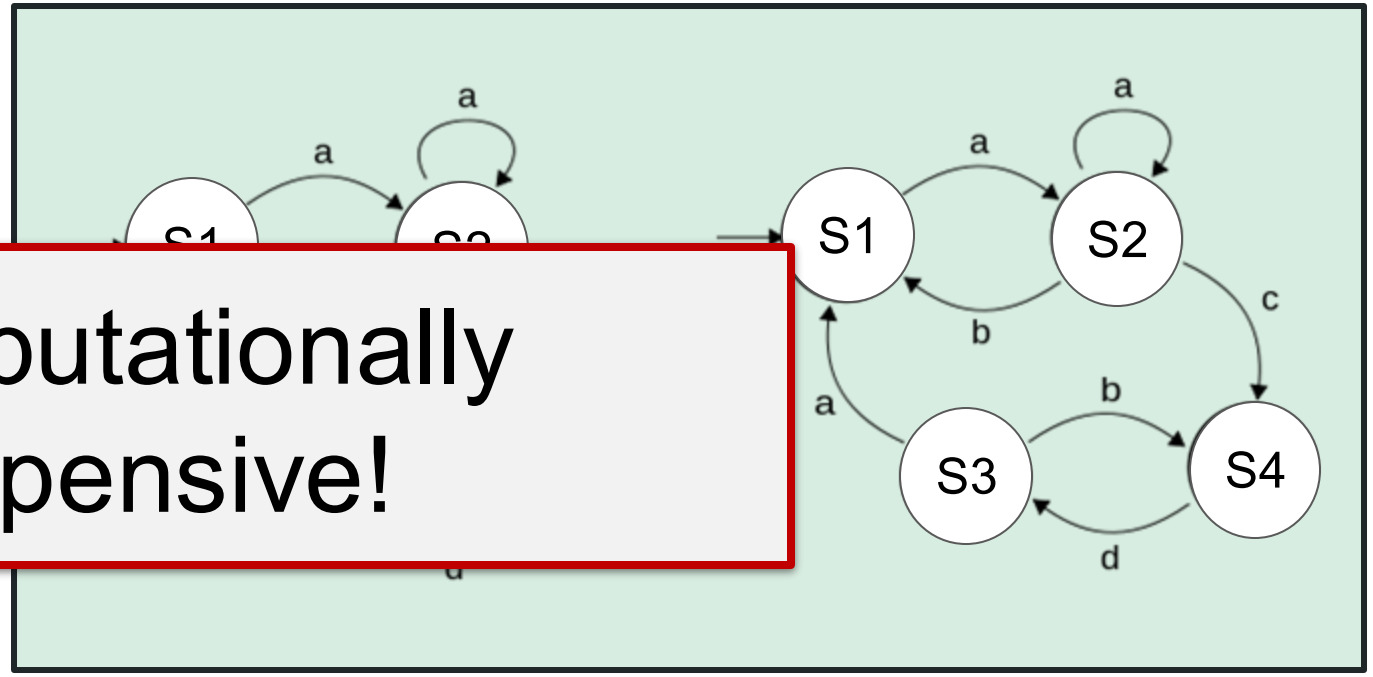


Current State of the Art

Taint Tracking



Model Checking



Computationally Expensive!

CONCLUSION

Security Mindset

- Identify unstated assumptions
  - *what happens when an attacker violates the assumptions?*
- Ask questions
  - *to find weak points, you have to understand the system*

The presence of an adversary changes everything

- Who is the attacker?
- What is the attacker's motivation?
- What are the attacker's resources?

P.S.

- Never roll your own cryptographic algorithms