Opaque: An Oblivious and Encrypted Distributed Analytics Platform
[NSDI’17]

Raluca Ada Popa

Joint work with: Wenting Zheng, Ankur Dave, Jethro Beekman, Joseph Gonzalez, and Ion Stoica

UC Berkeley
Complex analytics run on sensitive data.
Complex analytics run on sensitive data

client

cloud provider

sensitive data
Complex analytics run on sensitive data

client

cloud provider

- Spark
- SQL
- MLLib
- GraphX
- Spark Streaming

sensitive data
NEWS ANALYSIS

No, your data isn't secure in the cloud

In 2012, Google alone received 21,389 government requests for information affecting 33,634 user accounts

By Lucas Mearian
Senior Reporter, Computerworld | AUG 13, 2013 7:00 AM PT

How to Tell If Your Cloud Provider Can Read Your Data
by Rich Mogull

With the tremendous popularity of services like Dropbox and iCloud there is, rightfully, an incredible amount of interest in cloud data security. Once we start hosting our most sensitive data with cloud services (or any third-party provider) it’s only natural to wonder how secure our data is when it’s in the hands of others. But sometimes it’s hard to figure out exactly who can look at our information, especially since buzzwords like “secure” and “encrypted” don’t necessarily mean you

Why You Shouldn't Trust Your Data: An Employee's Perspective
by Ryan Tate

The world’s biggest data breaches 2015 - 888 incidents, 246 million records, uncounted misery
Cloud attackers

client

cloud provider

sensitive data
Cloud attackers

client

cloud provider

sensitive data
Cloud attackers

client

cloud provider

sensitive data
Cloud attackers

client

cloud provider

sensitive data
Cloud attackers

client

sensitive data

cloud provider
Threat model

Attacker has full access to all cloud software
How to protect data and computation while preserving functionality?
How to protect data and computation while preserving functionality?

relational algebra
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM
  [RAD’78, Gentry’09]
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM
  [RAD’78, Gentry’09]

  too slow
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM [RAD’78,Gentry’09] 
  too slow

- Specialized solutions: CryptDB, Cipherbase, Monomi, [...], BlindSeer, [FJK+15], Arx
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM [RAD’78, Gentry’09]
  
  too slow

- Specialized solutions: CryptDB, Cipherbase, Monomi, […], BlindSeer, [FJK+15], Arx

restricted functionality
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM
  [RAD’78, Gentry’09]
  too slow

- Specialized solutions: CryptDB, Cipherbase, Monomi, [...], BlindSeer, [FJK+15], Arx

restricted functionality

Alternative: hardware enclaves
Hardware enclaves 101
Hardware enclaves (Intel SGX)

- Hardware-enforced isolated execution environment
Hardware enclaves  (Intel SGX)

- Hardware-enforced isolated execution environment
Hardware enclaves (Intel SGX)

- Hardware-enforced isolated execution environment
Hardware enclaves (Intel SGX)

- Hardware-enforced isolated execution environment
- Data decrypted only on the processor
Hardware enclaves (Intel SGX)

- Hardware-enforced isolated execution environment
- Data decrypted only on the processor
- Protect against an attacker who has root access or compromised OS
Remote attestation
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange.
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange.
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange.
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Attacker is restricted to software attacks only, and does not exploit timing. **Attacker controls the software stack.**
Attacker is restricted to software attacks only, and does not exploit timing. **Attacker controls the software stack.**
Attacker is restricted to software attacks only, and does not exploit timing. **Attacker controls the software stack.**

- for the implementation, attacker does not see accesses to a small memory region (due to T-SGX, page pinning, super pages, ... )
Enclave-based systems

• Systems supporting relational algebra: Haven, Scone
Enclave-based systems

- Systems supporting relational algebra: Haven, Scone
  - not distributed
  - data access pattern leakage [XCP ’15, OCFGKS ’15]
Enclave-based systems

- Systems supporting relational algebra: Haven, Scone
  - not distributed
  - data access pattern leakage [XCP ’15, OCFGKS ’15]

- Distributed systems: VC3, M2R, Ohrimenko et al.’16: do not enable relational algebra and query planning
Access patterns leakage
Access patterns

machine 0

processor → memory

Locked
Access patterns

machine 0

processor

addresses

memory

locked
Access patterns

machine 0

processor
addresses

memory

network messages

machine 1
Example: network access pattern leakage
Example: network access pattern leakage

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>
Example: network access pattern leakage

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>

```sql
SELECT count(*) FROM medical
GROUP BY disease
```
Example: network access pattern leakage
Example: network access pattern leakage

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Diabetes</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>29489</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>13744</td>
<td>...</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>18740</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>98329</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Example: network access pattern leakage
Example: network access pattern leakage

<table>
<thead>
<tr>
<th></th>
<th>Diabetes</th>
<th>Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>...</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
</tr>
<tr>
<td>32591</td>
<td>...</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>
Example: network access pattern leakage

<table>
<thead>
<tr>
<th>ID</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Cancer</td>
</tr>
<tr>
<td>32591</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>
Example: network access pattern leakage

12809 … Diabetes
129489 … Diabetes
18740 … Diabetes
32591 … Diabetes

13744 … Cancer
98329 … Cancer
Example: network access pattern leakage

Public information:
Diabetes twice as common as cancer
Example: network access pattern leakage

Public information:
Diabetes twice as common as cancer
Example: network access pattern leakage

Public information:
Diabetes twice as common as cancer
Example: network access pattern leakage

Public information:
Diabetes twice as common as cancer
Example: network access pattern leakage
Example: network access pattern leakage
Example: network access pattern leakage
Example: network access pattern leakage

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>...</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>18740</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>...</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>

98329 | ... | Cancer
Example: network access pattern leakage

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
```

![Diagram of network access pattern leakage](image)
Example: network access pattern leakage

<table>
<thead>
<tr>
<th></th>
<th>Diabetes</th>
<th></th>
<th>Diabetes</th>
<th></th>
<th>Cancer</th>
<th></th>
<th>Diabetes</th>
<th></th>
<th>Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>...</td>
<td>29489</td>
<td>...</td>
<td>13744</td>
<td>...</td>
<td>18740</td>
<td>...</td>
<td>32591</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>????</td>
<td></td>
<td>????</td>
<td></td>
<td>????</td>
<td></td>
<td>????</td>
<td></td>
<td>????</td>
</tr>
<tr>
<td></td>
<td>Diabetes</td>
<td></td>
<td>Diabetes</td>
<td></td>
<td>Cancer</td>
<td></td>
<td>Diabetes</td>
<td></td>
<td>Cancer</td>
</tr>
<tr>
<td></td>
<td>???</td>
<td></td>
<td>???</td>
<td></td>
<td>???</td>
<td></td>
<td>???</td>
<td></td>
<td>???</td>
</tr>
</tbody>
</table>
Example: network access pattern leakage

Learns that Alice has cancer
Leakage from prior work

- Memory access patterns attacks [XCP15] extracted complete text documents and photo outlines
- Network access patterns [OCF+15] extracted age, gender, address of individuals
Goal: oblivious distributed analytics
Goal: oblivious distributed analytics

access patterns are independent of data content
Opaque*: oblivious and encrypted distributed analytics platform

* Oblivious Platform for Analytic QUEries
Security guarantees (informal)
Security guarantees (informal)

- Data encryption and authentication
Security guarantees (informal)

- Data encryption and authentication
Security guarantees (informal)

- Data encryption and authentication

- Computation integrity: the client can check that the computation result was not affected by an attacker
Security guarantees (informal)

- **Data encryption and authentication**

- **Computation integrity**: the client can check that the computation result was not affected by an attacker
Security guarantees (informal)

- **Data encryption and authentication**
- **Computation integrity:** the client can check that the computation result was not affected by an attacker
- **Obliviousness:** The memory and network accesses of a query is the same for any two inputs with the same size characteristics (input/outputs)
Security guarantees (informal)

- **Data encryption and authentication**

- **Computation integrity:** the client can check that the computation result was not affected by an attacker

- **Obliviousness:** The memory and network accesses of a query is the *same* for *any* two inputs with the same size characteristics (input/outputs)
  - When enabling padding, Opaque hides output sizes as well
Achieving practical obliviousness is not easy
Achieving practical obliviousness is not easy

Obliviousness typically comes with high overheads

- For example, the state-of-the-art system, ObliVM, is six orders of magnitude slower than regular computation
Opaque components
Opaque components

Data encryption and authentication
Opaque components

- Computation verification
- Data encryption and authentication
Opaque components

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Query execution

Client

Scheduler

Database

Server
Query execution

Client

Spark Driver

Opaque

Catalyst

Server

Scheduler

Database

1
2
3
query = SELECT sum(*) FROM table
query = SELECT sum(*) FROM table

Client

Spark Driver

Opaque

Catalyst

Scheduler

1 2 3

Database

Server
query = SELECT sum(*)
FROM table
Query execution

query = SELECT sum(*)
FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*) FROM table
Query execution

query = SELECT sum(*) FROM table
query = SELECT sum(*)
FROM table
Query execution

query = SELECT sum(*)
FROM table

Client

Server

Database

Scheduler

Opaque
Catalyst
Spark Driver
query = SELECT sum(*)
FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*) FROM table
Problem: cloud can alter distributed computation
Problem: cloud can alter distributed computation

- Drop data
Problem: cloud can alter distributed computation

- Drop data
- Modify data
Problem: cloud can alter distributed computation

- Drop data
- Modify data
- Skip task
Problem: cloud can alter distributed computation

- Drop data
- Modify data
- Skip task
- Replay old state
Example: drop data

query = SELECT sum(*)
FROM table
Example: drop data

query = SELECT sum(*)
FROM table
Example: drop data

query = SELECT sum(*)
FROM table
Example: drop data

query = SELECT sum(*)
FROM table
Example: drop data

query = SELECT sum(*)
FROM table

Client

Scheduler

Opaque
Catalyst
Spark Driver

23
Database
Server
Example: drop data

query = SELECT sum(*)
FROM table
Self-verifying computation

Invariant: if computation does not abort, the execution completed so far is correct
Self-verifying computation

Invariant: if computation does not abort, the execution completed so far is correct
Self-verifying computation

Invariant: if computation does not abort, the execution completed so far is correct

If the computation is complete, then the entire query was executed correctly
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

Task 13

Task 14

Task 15

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

1 12809 ...
   ... Diabetes

1 29489 ...
   ... Diabetes

1 13744 ...
   ... Cancer

2 18740 ...
   ... Diabetes

2 98329 ...
   ... Cancer

2 32591 ...
   ... Diabetes
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

There can be many partitions
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Map

Sort

Diabetes

Cancer

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton ‘85]

Map

Sort

12809 ... Diabetes
98329 ... Cancer
13744 ... Diabetes
29489 ... Cancer
18740 ... Diabetes
32591 ... Diabetes
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Map

Sort

Oblivious sort

[CLRS, Leighton ‘85]

13744 ... Cancer
98329 ... Cancer
12809 ... Diabetes
29489 ... Diabetes
18740 ... Diabetes
32591 ... Diabetes
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

```
SELECT count(*) FROM medical GROUP BY disease
```
Oblivious aggregation

```sql
SELECT count(*) FROM medical GROUP BY disease
```

The “Diabetes” group is split!
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

The “Diabetes” group is split!

How to aggregate obliviously and in parallel?
Oblivious aggregation

```
SELECT count(*) FROM medical GROUP BY disease
```

The “Diabetes” group is split!

How to aggregate obliviously and in parallel? It can span over many partitions.
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan
Oblivious aggregation

```
SELECT count(*) FROM medical GROUP BY disease
```
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan  Boundary processing
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan Boundary processing

Partial agg.

Cancer;Diabetes:1

Diabetes;Diabetes:3
Oblivious aggregation

```sql
SELECT count(*) FROM medical GROUP BY disease
```

Scan  Boundary processing
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan

Boundary processing
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

```
SELECT count(*) FROM medical GROUP BY disease
```
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan

Boundary processing

Scan
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan

Boundary processing

Scan
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan          Boundary processing          Scan

13744 ... Cancer
98329 ... Cancer
12809 ... Diabetes

129489 ... Diabetes
18740 ... Diabetes
32591 ... Diabetes

13744 ... Cancer
98329 ... Cancer
12809 ... Diabetes

129489 ... Diabetes
18740 ... Diabetes
32591 ... Diabetes

Diabetes: 1
Cancer: 2
Diabetes: 4
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan | Boundary processing | Scan

| 13744 | ... | Cancer |
| 98329 | ... | Cancer |
| 12809 | ... | Diabetes |

| 29489 | ... | Diabetes |
| 18740 | ... | Diabetes |
| 32591 | ... | Diabetes |

| 13744 | ... | Cancer |
| 98329 | ... | Cancer |
| 12809 | ... | Diabetes |

Diabetes: 4

Cancer: 2

DUMMY

DUMMY

DUMMY

DUMMY

DUMMY

DUMMY

Diabetes: 1
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

DUMMY
Cancer: 2
DUMMY

DUMMY
DUMMY
DUMMY
Diabetes: 4
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Sort

DUMMY

Cancer: 2

DUMMY

DUMMY

DUMMY

Diabetes: 4

Oblivious sort
[CLRS, Leighton '85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton '85]

Cancer: 2
Diabetes: 4

Sort
Final result
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton ‘85]

Sort

Cancer: 2
Diabetes: 4

Final result

Aggregation has two sorts…
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Aggregation has two sorts…

Can we do better?

Sort

Final result

Oblivious sort
[CLRS, Leighton ‘85]
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Rule-based optimization
SELECT count(*)
FROM medical
WHERE age > 30
GROUP BY disease
Rule-based optimization

SELECT count(*)
FROM medical
WHERE age > 30
GROUP BY disease
Insight 1
Insight 1

1. Split each logical operator into smaller Opaque operators
Insight 1

1. Split each logical operator into smaller Opaque operators

2. Take a global view across the plan to remove some Opaque operators
Rule-based optimization

Logical op.

- Aggregation
- Filter
- medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter
- medical

- medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

medical
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter
- medical

- Scan
- medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation
  ↑
Filter
  ↑
medical
  ↑
Filter
  ↑
O-sort
  ↑
Project
  ↑
Scan
  ↑
medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

Scan

medical

Filter

O-sort

Project

medical

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Filter

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter
- O-sort
- Project
- Scan
- medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical

Filter

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical

Filter

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th></th>
<th>Amanda D. Edwards</th>
<th>40</th>
<th>Diabetes</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>122809</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Robert R. McGowan</th>
<th>56</th>
<th>Diabetes</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>129489</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Kimberly R. Seay</th>
<th>51</th>
<th>Cancer</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Dennis G. Bates</th>
<th>32</th>
<th>Diabetes</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>18740</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Donna R. Bridges</th>
<th>26</th>
<th>Diabetes</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>132591</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Ronald S. Ogden</th>
<th>53</th>
<th>Cancer</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>98329</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Medical Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Filter

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

12809 Amanda D. Edwards 40 Diabetes 0
129489 Robert R. McGowan 56 Diabetes 0
13744 Kimberly R. Seay 51 Cancer 0
18740 Dennis G. Bates 32 Diabetes 0
98329 Ronald S. Ogden 53 Cancer 0
32591 Donna R. Bridges 26 Diabetes 1
Rule-based optimization

Opaque op.

Logical op.

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Filter

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Condition</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

Rule-based optimization

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

O-sort

Filter

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

O-sort

Filter

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

O-sort
Agg.
O-sort
Filter
O-sort
Project
Scan
medical

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>
Rule-based optimization

Can we remove any sort?
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

O-sort

Filter

O-sort

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

O-sort

Filter

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter
- medical

- O-sort
- Agg.
- O-sort
- Filter
- O-sort
- Project
- Scan
- medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Sort on 0/1 column

O-sort

Agg.

O-sort

Filter

O-sort

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter

Scan

- O-sort
- Agg.
- O-sort
- Filter
- Project

Sort on 0/1 column
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

medical

Scan

Project

O-sort

Agg.

Sort on Disease

Sort on 0/1 column
Rule-based optimization

Opaque op.

Logical op.

Medical

Sort on Disease

Sort on 0/1 column

Scan

Project

O-sort

Agg.

O-sort

Filter

Aggregation

Filter

Medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

medical

Scan

Project

O-sort

O-sort

Agg.

Sort on Disease

+  

Sort on 0/1 column

=  

medical
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter
- medical

- O-sort
- Agg.
- O-sort
- Filter
- O-sort
- Project
- Scan
- medical

Sort on Disease + Sort on 0/1 column = Sort on (0/1, Disease)
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
  - Filter
    - medical

- O-sort
  - Agg.
  - Filter
  - O-sort
  - Project
  - Scan
    - medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

medical

↑

medical

↑

Scan

↑

Project

↑

O-sort

↑

Agg.

↑

O-sort

↑

Filter

↑

O-sort

↑

medical
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter
- medical

- Scan
- Project
- O-sort
- Agg.
- O-sort
- medical
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
  - Filter
    - medical

Scan

Project

Filter

Agg.

O-sort

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sorted

Agg.

Filter

O-sorted

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

12809 Amanda D. Edwards 40 Diabetes
129489 Robert R. McGowan 56 Diabetes
13744 Kimberly R. Seay 51 Cancer
18740 Dennis G. Bates 32 Diabetes
32591 Donna R. Bridges 26 Diabetes
98329 Ronald S. Ogden 53 Cancer
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

Project

Scan

medical

medical

O-sort

Agg.

Filter

O-sort

Amanda D. Edwards 40  Diabetes

Robert R. McGowan 56  Diabetes

Kimberly R. Seay 51  Cancer

Dennis G. Bates 32  Diabetes

Donna R. Bridges 26  Diabetes

Ronald S. Ogden 53  Cancer
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

12809 Amanda D. Edwards 40 Diabetes
129489 Robert R. McGowan 56 Diabetes
13744 Kimberly R. Seay 51 Cancer
18740 Dennis G. Bates 32 Diabetes
32591 Donna R. Bridges 26 Diabetes
98329 Ronald S. Ogden 53 Cancer
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Symptom</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

multi-column sort

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Filter

Aggregation

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

multi-column sort

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Logical op.

Opaque op.

Aggregation

Filter

medical

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

multi-column sort

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Diabetes</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

- O-sort
- Agg.
- Filter
- O-sort
- Project
- Scan
- medical

multi-column sort

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Logical op.

Opaque op.

Aggregation

Filter

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

multi-column sort

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
### Rule-based optimization

The schema tree visualizes the logical and opaque operations involved in processing medical data. Here are the operations in order from top to bottom:

- **Filter**
- **Project**
- **Scan**
- **O-sort**
- **Agg.**
- **O-sort**

#### Logical op.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Opaque op.

- **Filter**
- **Aggregation**

- **Filter**

- **Multi-column sort**
Rule-based optimization

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Rule-based optimization

Opaque op.

Logical op.

Filter

Aggregation

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

13744 Kimberly R. Seay 51 Cancer 0
98329 Ronald S. Ogden 53 Cancer 0
12809 Amanda D. Edwards 40 Diabetes 0
18740 Dennis G. Bates 32 Diabetes 0
29489 Robert R. McGowan 56 Diabetes 0

multi-column sort

Eliminated one oblivious sort!
Opaque components

- Oblivious query planning
  - Cost model
  - Rule-based opt.
  - Cost-based opt.

- Distributed oblivious operators
  - Oblivious Filter
  - Oblivious Aggregation
  - Oblivious Join

- Computation verification

- Data encryption and authentication
Observation: not all tables are sensitive
Observation: not all tables are sensitive

Hospitalized patients
- P_ID
- D_ID
- Name
- Age

Disease
- D_ID
- Name
- G_ID

Medication
- M_ID
- D_ID
- Name
- Cost
Observation: not all tables are sensitive
Observation: not all tables are sensitive

Opaque can operate in *mixed sensitivity*: sensitive tables are run with oblivious operators.
Observation: not all tables are sensitive
Observation: not all tables are sensitive
Observation: not all tables are sensitive

Not oblivious!
Observation: not all tables are sensitive
Observation: not all tables are sensitive

Sensitivity propagation: propagate obliviousness from leaf to root
Observation: not all tables are sensitive

Sensitivity propagation: propagate obliviousness from leaf to root
Observation: not all tables are sensitive

Sensitivity propagation:
propagate obliviousness from leaf to root
Insight 2

Sensitivity propagation introduces a new dimension to query optimization
Cost-based optimization

Find the least costly medication for each patient
Cost-based optimization

Find the least costly medication for each patient

Assumption: |P| < |D| < |M|
Cost-based optimization

Find the least costly medication for each patient
Assumption: |P| < |D| < |M|

SELECT p_name, d_name, med_cost
FROM patient, disease,
    (SELECT d_id, min(cost) AS med_cost
     FROM medication
     GROUP BY d_id) AS med
WHERE disease.d_id = patient.d_id
    AND disease.d_id = med.d_id
Cost-based optimization

Find the least costly medication for each patient

Assumption: |P| < |D| < |M|

SELECT p_name, d_name, med_cost
FROM patient, disease,
     (SELECT d_id, min(cost) AS med_cost
      FROM medication
      GROUP BY d_id) AS med
WHERE disease.d_id = patient.d_id
    AND disease.d_id = med.d_id
Cost-based optimization

Find the least costly medication for each patient

Assumption: |P| < |D| < |M|

3-way join

SELECT p_name, d_name, med_cost
FROM patient, disease,
    (SELECT d_id, min(cost) AS med_cost
     FROM medication
     GROUP BY d_id) AS med
WHERE disease.d_id = patient.d_id
    AND disease.d_id = med.d_id
Cost-based optimization

SQL optimizer with new cost:
Cost-based optimization

SQL optimizer with new cost:

More selective non-oblivious join
Cost-based optimization

SQL optimizer with new cost:

More selective non-oblivious join
Cost-based optimization

SQL optimizer with new cost and sensitivity propagation:
Cost-based optimization

SQL optimizer with new cost and sensitivity propagation:

Fewer oblivious joins
Cost-based optimization

SQL optimizer with new cost and sensitivity propagation:

Fewer oblivious joins
Evaluation setup
Evaluation setup

- Single machine experiments:
  - Intel Xeon E3-1280 v5, 4 cores, 64 GB RAM
  - Intel SGX: 128 MB of enclave page cache (EPC)
Evaluation setup

• Single machine experiments:
  • Intel Xeon E3-1280 v5, 4 cores, 64 GB RAM
  • Intel SGX: 128 MB of enclave page cache (EPC)

• Distributed experiments
  • A cluster of 5 SGX machines
Evaluation
Evaluation

• How does Opaque compare to Spark SQL?
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
Evaluation

- How does Opaque compare to Spark SQL?
  - Big Data Benchmark (BDB); 4 queries total
    - Queries 1, 2, 3: filter, aggregation, join
    - 1 million records
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
    • 1 million records

• How does Opaque compare to state-of-the-art oblivious systems?
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
    • 1 million records

• How does Opaque compare to state-of-the-art oblivious systems?
  • GraphSC (oblivious graph analytics)
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
    • 1 million records

• How does Opaque compare to state-of-the-art oblivious systems?
  • GraphSC (oblivious graph analytics)
    • PageRank
Big Data Benchmark (distributed)
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

<table>
<thead>
<tr>
<th>Runtime (s)</th>
<th>Query number</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Spark SQL</td>
</tr>
<tr>
<td></td>
<td>Opaque</td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td></td>
</tr>
</tbody>
</table>

Query 1  Query 2  Query 3
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

Runtime (s)

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>Query 2</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Query 3</td>
<td>10.0</td>
<td></td>
</tr>
</tbody>
</table>
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>Runtime(s)</td>
<td></td>
</tr>
<tr>
<td>Query 2</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Query 3</td>
<td>10</td>
<td>100</td>
</tr>
</tbody>
</table>
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>Query 2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Query 3</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

+ Obliviousness

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

+ Obliviousness

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

+ Obliviousness

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Query 2</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>Query 3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

+ Obliviousness

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Query 2</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>Query 3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Overhead: -0.47x to 2.3x

Overhead: 21x to 45x
PageRank: comparison with GraphSC (single machine)

![Graph showing comparison between PageRank and GraphSC runtime]

- GraphSC
- Opaque

Runtime (s) vs. Graph size (nodes+edges)

2300x improvement for GraphSC over Opaque
How does Opaque fit among practical encrypted databases*?

*single-cloud, relational DBs
Confidentiality is a beast
Confidentiality is a beast

- Sharp tradeoff between performance and confidentiality. No practical encrypted database has perfect confidentiality.
Confidentiality is a fun beast

- Sharp tradeoff between performance and confidentiality. No practical encrypted database has perfect confidentiality.
Confidentiality is a fun beast

- Sharp tradeoff between performance and confidentiality. No practical encrypted database has perfect confidentiality.

- Non-ideal confidentiality is meaningful: it removes classes of attackers, and leaks much less data.
Spectrum of contributions

plaintext → property-preserving encryption → semantic security → hide access patterns → hide result sizes → no leakage

all leaks
Spectrum of contributions

- plaintext
- property-preserving encryption
- semantic security
- hide access patterns
- hide result sizes
- timing, query, structure of input

better performance

all leaks

no leakage
Spectrum of contributions

better performance

plaintext

property-preserving encryption

semantic security

hide access patterns

hide result sizes

timing, query, structure of input

all leaks

CryptDB

Monomi

Cipherbase

no leakage

SAP

skyhigh
Spectrum of contributions

- All leaks
- Plaintext
- Property-preserving encryption
- Semantic security
- Hide access patterns
- Hide result sizes
- Hide timing, query, structure of input

CryptDB, Monomi, Cipherbase, Haven, SCONE, Arx, BlindSeer, VC3, [FJK+15]

Better performance
Spectrum of contributions

- plaintext
- property-preserving encryption
- semantic security
- hide access patterns
- hide result sizes
- timing, query, structure of input
- all leaks
- better performance

CryptDB, Monomi, Cipherbase, Haven, SCONE, VC3, Arx, BlindSeer, [FJK+15]

Opaque, skyhigh

no leakage
Spectrum of contributions

- Better performance
- Plaintext
- All leaks
- Property-preserving encryption
- CryptDB, Monomi, Cipherbase
- Haven, SCONE, VC3, Arx, BlindSeer, [FJK+15]
- Opaque +pad
- No leakage
- Semantic security
- Opaque
- Hide access patterns
- Hide result sizes
- Timing, query, structure of input
Spectrum of contributions

- plaintext
- property-preserving encryption
- semantic security
- hide access patterns
- hide result sizes
- timing, query, structure of input

Each of these systems protects against classes of attackers

better performance

all leaks

Opaque

Opaque + pad

no leakage
Spectrum of contributions

 plaintext

 property-preserving encryption

 semantic security

 hide access patterns

 hide result sizes

 timing, query, structure of input

 better performance

 plaintext

 all leaks

 CryptDB

 Monomi

 Cipherbase

 Haven, SCONE, VC3, Arx, BlindSeer, [FJK+15]

 Opaque

 Opaque +pad

 no leakage

 Each of these systems protects against classes of attackers

 Opaque offers strong security guarantees
Conclusion

Opaque is an oblivious and encrypted distributed analytics platform

Open source: github.com/ucbrise/opaque