Cryptographically Protected Database Search

Benjamin Fuller, Mayank Varia, Arkady Yerukhimovich, Emily Shen, Ariel Hamlin, Vijay Gadepally, Richard Shay, Darby Mitchell, Robert Cunningham

benjamin.fuller@uconn.edu
“Data is the new oil”
– Shivon Zilis, Bloomberg Beta

“Data will become a currency”
– David Kenny, IBM Watson

“... the fourth industrial revolution is connectivity and data”
– Mukesh Ambani, Reliance

Interesting takeaway No. 1: 61% of respondents “acknowledge that big data is now a driver of revenues in its own right and is becoming as valuable to their businesses as their existing products and services.”
Value implies Risk

The telecommunications company TalkTalk admitted that its data breach last year resulted in criminals using customer information to commit fraud. This was more bad news for a company that had just executed a 100% efficiency improvement. OPM breach: 4.5 million more individuals of personal data stolen sells for 10X price on the black market. Why corporates should care about the Ashley Madison breach. Hackers.com: Massive IRS data breach much bigger than thought.

“Data is a toxic asset”

– Bruce Schneier, 2016

"We're sorry you got hacked": Target's letter to unlucky shoppers.
Lets encrypt data!!

Data owners

Clients

Owner and Client could be at same or partner orgs

Database servers

Backend storage
Let's encrypt data!!

Network attacks can be (mostly) mitigated using standard techniques such as TLS.
Encrypt data at rest, required for some data types due to regulation (HIPAA)

- Capability exists in modern databases (e.g. Accumulo)
Let's encrypt data!!

Data owners

Clients

possible threats?

Database servers

Backend storage

Client restricted using access control at server
Let's encrypt data!!

Data owners

Clients

Mitigation seems difficult, server must be able to process queries, but is not trustworthy

possible threats?

Database servers

Backend storage
Cryptographically Protected Search

Utility of stored data

- Return whole dataset encrypted
- Use homomorphic encryption or multi-party computation

Risk of data compromise

- No server protections (encrypt data at rest)

Homomorphic encryption vector-matrix mult: 30s\(^1\)
Multi-party computation: 200,000 AES\(^2\) blocks/s, does not scale to large data

Databases are expected to answer common queries in milliseconds

5000 range queries takes 1s

---


\(^2\)M. Keller, E. Orsini, D. Rotaru, P. Scholl, E. Soria-Vazquez, S. Vivek,
“Faster Secure Multi-Party Computation of AES and DES Using Lookup Tables,” in ACNS 2017
Cryptographically Protected Search

Includes:
- Symmetric searchable encryption (SSE)
- Property preserving encryption

Return whole dataset encrypted

- Use homomorphic encryption or multi-party computation

- No server protections (encrypt data at rest)
Why systematize?

- No server protections (encrypt data at rest)
- Return whole dataset encrypted
- Use homomorphic encryption or multi-party computation

We evaluated results for IARPA SPAR

Utility of stored data

Risk of data compromise
Outline

• Overview of Protected Search

Leakage Impacts

• Finding a basis for search results
  • Range queries
    • Compatible approach: Order-Preserving Encryption / CryptDB
    • Custom approach: Partial Order-Preserving Encryption
    • Obliv approach: SisoSPIR
  • Combining queries

• Extending to new database paradigms
Common Language for Leakage

Protected search schemes reveal some information about the query, data set, and result set to each party.

Called leakage.

Difficult to compare, phrased to make proofs work, not to compare schemes

Define five types of leakage of increasing impact¹:
1. Structure
2. Identifiers
3. Predicates
4. Equality
5. Order

Some schemes leak:
1. At Initialization on entire DB
2. At Query on relevant records

Hospital Data Set

<table>
<thead>
<tr>
<th>Birth Month</th>
<th>Length of Stay</th>
<th>Gender</th>
<th>Diagnosis</th>
<th>SSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>February</td>
<td>1</td>
<td>M</td>
<td>Flu</td>
<td>000-00-001</td>
</tr>
<tr>
<td>April</td>
<td>30</td>
<td>M</td>
<td>Cancer</td>
<td>000-00-002</td>
</tr>
<tr>
<td>June</td>
<td>3</td>
<td>F</td>
<td>Pneumonia</td>
<td>000-00-003</td>
</tr>
</tbody>
</table>

• Assume:
  • Server sees which field queried
  • Records are identifiable between queries
Statistical Attack Against Hospital Length of Stay

• Suppose:
  • Queries of form:
    SELECT * FROM table
    WHERE
    length_stay=XXXXX;
  • Observe |records|
  • Create unique id for query
Distribution of length of stay is known, attacker can use prior statistical information

Query with highest number of returned records likely represents 4 days
Statistical Attack Against Hospital Length of Stay

What to do if number of records is not identifying enough? Or statistical prior is inaccurate?

Attacks exploit correlation between fields, use techniques from optimization
Why systematize?

- Utility of stored data
- Focus of the remainder of the talk
- No server protections (encrypt data at rest)
- Return whole dataset encrypted
  - Use homomorphic encryption or multi-party computation
- Risk of data compromise
Approaches to Protected Databases

Find three approaches to protected databases:

1. Legacy:
   - Leak at *Initialization*
   - Inherit DB advances

2. Custom:
   - Leak during *Query*

3. Obliv:
   - Leak only structure
   - Require multiple servers to be efficient

Define five types of leakage of increasing impact:
1. Structure
2. Identifiers
3. Predicates
4. Equality
5. Order

Distinguish between schemes that leak this information at *Initialization* and at *Query*.

---

Approaches to Protected Databases

Find three approaches to protected databases:

1. **Legacy**:
   - Leak at *Initialization*
   - Inherit DB advances

2. **Custom**:
   - Leak during *Query*

3. **Obliv**:
   - Leak only structure
   - Require multiple servers to be efficient

---

Developed\(^9\):
- a database instrumentation platform
- data and query generator
- Used in prior work\(^{10, 11}\)

---

\(^9\)https://github.com/mit-ll/sparta
How to compare functionality?

• Natural approach: what fraction of a unprotected database language is supported?

• Current systems implement base queries using cryptography, extend from these base queries:
  • Keyword Equality
  • Range
  • Boolean Combination
  • Other (graph alg and substring)

Find three approaches to protected databases:
1. Legacy:
   • Leak at Initialization
   • Inherit DB advances
2. Custom:
   • Leak during Query
3. Obliv:
   • Leak only structure
   • Require multiple servers to be efficient
Outline

• Overview of Protected Search
• Leakage Impacts

Finding a basis for search results
  • Range queries
    • Order-Preserving Encryption
    • Partial Order-Preserving Encryption
    • SisoSPIR
  • Combining queries

• Extending to new database paradigms
Order-Preserving Encryption

• Enc that preserves plaintext order:
  • If $m_1 < m_2$ then $\text{Enc}(m_1) < \text{Enc}(m_2)$

1. Encrypt query $\text{Enc}(a), \text{Enc}(b)$
2. Let server use standard search mechanism
3. Return encrypted records

$\text{Enc}(a), \text{Enc}(b)$
$c_1, c_2, c_{10}, c_{4000}$

Client

Database server
Leakage Attacks of OPE

- Data is sorted, does not protect dense data
- Strongest leakage attack applies to OPE
- Technique used in many commercial product

Row corresponding to OPE
Partial Order Preserving Encoding

- Client sends data to server encrypted and unsorted
- Client and Server work together to create partially sorted tree
  - Client performs all comparisons
  - Server is able to build tree based on client comparisons
- Stronger security than Order-Preserving Encryption if tree is only partially built

---

B+ trees are used in many unprotected databases

Variable number of children per node

Idea of approach: use crypto to hide all information in traversing B+ tree

Requires multiple servers for practical efficiency

14Y. Ishai, E. Kushilevitz, S. Lu and R. Ostrovsky, “Private Large-Scale Databases with Distributed Searchable Symmetric Encryption,” CT-RSA 2015
Data is high entropy

Small fraction of DB is returned by queries

Otherwise
Query Combination

• Techniques to combine base queries:
  • Range $\rightarrow$ Equality, search for $[a, a]$
  • Boolean $\rightarrow$ Range, using set covers
  • Range $\rightarrow$ Substring, by inserting each prefix

• Most combination techniques are less efficient and have more leakage than equivalent base query

• Allow for rapid expansion of query functionality
Approaches to Protected Databases

- Natural approach: what fraction of an unprotected database language is supported?

- Systems implement base queries w/ crypto, extend from these base queries:
  - Keyword Equality
  - Range
  - Boolean Combination
  - Other (graph alg and substring)

SQL has a well defined mathematical set-theory basis of operations\(^\text{14}\):
- Union: \(A \cup B\)
- Difference: \(A \setminus B\)
- Join: \(A \times B\)
- Projection: Take some dimensions of results
- Selection: Take rows satisfying some condition

\(^{14}\)E. Codd, “A relational model of data for large shared data banks,” *Communications of the ACM*, 1970
Unprotected DB Development

Introduction of relational model by Codd

MySQL, PostGRES, Oracle

It took 20 years to secure SQL

How can we catch up?

Crypto community starts working on protected search

NoSQL Key-Value

NoSQL Graph DBs

NewSQL

Polystore
Keeping up with database diversification

Common unprotected databases have a mathematical basis of operations:

- For SQL: Union, Difference, Join, Projection, Selection
- For Array-Store: Construct, Find, Array (+, x), Element-wise x
- For Graph: Linear algebra over matrices

Cryptographers and DB designers should work together to:

1. Identify base queries that are likely to be useful across DB paradigms
2. Understand critical functions of emerging databases
3. Quickly fill gaps using combiners

Questions?
https://arxiv.org/abs/1703.02014
Backups
<table>
<thead>
<tr>
<th>DB Paradigm</th>
<th>Basis Operation</th>
<th>Crypto Base Operation?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSQL – Key Value Store</td>
<td>Construct</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Find</td>
<td>Yes – Mature range search with variety of techniques</td>
</tr>
<tr>
<td></td>
<td>Array (+, x)</td>
<td>Some – Addition possible using partially homomorphic techniques</td>
</tr>
<tr>
<td></td>
<td>Element Wise x</td>
<td>Some – Using partially homomorphic techniques</td>
</tr>
</tbody>
</table>

Main gap is support for very high insert rates above 1M records per second
<table>
<thead>
<tr>
<th>DB Paradigm</th>
<th>Basis Operation</th>
<th>Crypto Base Operation?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Databases—Linear Algebra</td>
<td>Construct</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Find</td>
<td>Yes – Mature range search with variety of techniques</td>
</tr>
<tr>
<td></td>
<td>Matrix (+, x)</td>
<td>Some – Have private algorithms for matrix mult./add.</td>
</tr>
<tr>
<td></td>
<td>Element Wise x</td>
<td>Some – Using homomorphic operations</td>
</tr>
</tbody>
</table>

Current matrix operations operate on full structure, need algorithms for sparse matrices (most graph algorithms)
Current systems

- Currently mature systems with peer-reviewed descriptions
- All systems use the basis and combination approach to get rich functionality

Questions?
https://arxiv.org/abs/1703.02014