On the security of NoSQL cloud database services

Problem:

Confidential Data leaks from cloud database services

Sensitive personal information of 143 million was exposed in a data breach at Equifax (2017).

A cyber attack to JP Morgan Chase, compromised personally identifiable information records of 76 million households and 7 million small businesses (2014).

Protecting sensitive information by processing on encrypted data.

Agenda

1. Introduction
2. Searchable Security Scheme
   • 2.1 Design principles
   • 2.2 Query re-writing
3. Information Leakage Prevention
   • 3.1 Classification
   • 3.2 Joint size estimation
4. Conclusion
   • Discussion

Selected references
2. Searchable Security Scheme for Cloud NoSQL

- Design principles
- Threat model
- Searchable encryption
- Query re-writing
- Experiment

2. Searchable Security Scheme

**Design principles**

1. Unmodified server and Applications
2. Database Server never gets decryption key only gets ciphered data
3. Modest overhead 20% throughput loss (number of query per second)
4. JavaScript based descriptive language to describe security parameters
**Threat model**

1) External attacker
2) Cloud malicious insiders

**Attack on database server:**

1) Passive database server attacks (honest but curious)
2) Active attack on server (The adversary gains complete control over database servers)

---

**Searchable encryption**

1- Deterministic

\[ C_j = E_k(P_j); \quad P_j = D_k(C_j) \] (1)

2- Random

\[ C_j = E_k(P_j \oplus IV); \quad P_j = IV \oplus D_k(C_j) \]
\[ C_j = E_k(P_j \oplus C_{j-1}); \quad P_j = C_{j-1} \oplus D_k(C_j) \] (2)

3- Order Preserving Encryption

\[ \forall x, y \in \text{Data Domain}, \quad x < y \implies OPE_k(x) < OPE_k(y) \] (3)

4- Additive Homomorphic Encryption

\[ D_k(E_k(m_1, n) \times E_k(m_2, n) \mod n^2) = m_1 + m_2 \mod n \] (4)

---

**2. Searchable Security Scheme**

Client side

- Security Layer
- Data Integrity
- Leakage prevention

SecureNoSQL Proxy

Response

Query

Server side

- Query Language
- Data Model
- Storage Engine

Data store

Replica set

The organization of SecureNoSQL

---

**Query re-writing**

**Applications of DET encryption**

DET encryption was used to support equality check and full-word search operations, therefore, a group of operations such as Count (\$count), Group by (\$group), Equality match (\$eq), Not equal (\$ne), and Selects if value specified is in the array (\$in).

**RND-Application**

Support indistinguishability under an adaptive chosen-plaintext attack (IND-CPA)

**OPE-Application**

For operations that are dealing with the order of data such as Greater than (\$ge), Less than (\$lt), $orderby, $sort, $max, and $min as well as range queries on encrypted data.

**AHOM-Application**

Summation and multiplication of encrypted numerical values.
Query re-writing

3- Cross-correlation Information Leakage

3-Cross-correlation information leakage
a) Cloud database service cross-correlation model
b) DBaaS Leakage Management
c) Disinformation, Sensitivity Analysis, and Approximate Query Processing
d) Warehouse Information Leakage

Performance

What is information leakage

“Information leakage is inadvertent disclosure of sensitive information, a cloud insider attacker can infer sensitive information either through multiple database searches, or cross-correlations among databases.”
Attribute cross-correlation (Example)

Example 1

Using this technique M. Naveed et. al. Were able to retrieve 80% of patient information from encrypted columns.

L_1 = \{CCN, Account, SSN, Balance\}

L_2 = \{CCN, Account, SSN, Balance, DOB, Disease\}

Leakage function

The cross-correlation information leakage model

\[ \Psi_{\text{CC}}(C_1, C_2) : \]

\[ \forall d \in C_1 \land \forall d' \in C_2 \]

\[ \mu(d, d') = \text{True} \implies L = \{\text{Att} \mid \forall \text{Att} \in d' \land \text{Att} \not\in d\} \]

The feasibility function \( \mu \):

\[ \mu(d, d') : \]

\[ \begin{cases} 
\text{True} & \text{if } \exists \text{Att}_i \in d \land \exists \text{Att}_j \in d' \land \left( (\text{Att}_i, \text{key}) = (\text{Att}_j, \text{key}) \land (\text{Att}_i, \text{value}) = (\text{Att}_j, \text{value}) \right) \\
\text{False} & \text{Otherwise} 
\end{cases} \]

Counter-measure

Disinformation Padding (H. Garcia-Molina et. al.): Replication of collection documents with altered sensitive fields

\[ \text{name: "Alex Johns", CCN: "234967891", Account: 432, SSN:234967431, Balance:200k, DOB: "5/10/1990"} \]

\[ \text{Disease: "Diabetes"} \]

\[ \text{C_1} \]

\[ \text{C_2} \]

\[ \text{C_3} \]

\[ \text{L_1 = \{CCN, Account, SSN, Balance\}} \]

\[ \text{L_2 = \{CCN, Account, SSN, Balance, DOB, Disease\}} \]

\[ \text{\Psi_{\text{Acc}}(C_1, C_2, C_3) = \Psi_{\text{Acc}}(C_2, C_3)} \]

\[ \text{Drawbacks} \]

1. Filtering mechanism
2. Performance degradation
3. Cost
4. CRUD operations

Immediately Lazy fashion
**Counter-measure**

<table>
<thead>
<tr>
<th>Filtering &amp; Integrity verification</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Graph showing performance of hash functions" /></td>
</tr>
</tbody>
</table>

**Classification method**

We proposed:

A sensitivity analysis method by leveraging the Approximate Query Processing (AQP) technique.

- Cloud databases include data with various degrees of sensitivity. We need to identify the most valuable information that must be protected.
- This approach compromises between orders of magnitude processing time improvement to the limited inaccuracy in response.

**Indexes over encrypted data**

<table>
<thead>
<tr>
<th>Index over encrypted data</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Graph showing latency" /></td>
</tr>
</tbody>
</table>

**AQP based classification method**

(i) Establish sensitivity levels
(ii) Approximate the number of documents in each sensitivity class by:
  a) Uniform random sampling
  b) Processing aggregate function over samples.
  c) Calculate confidence intervals.
  d) Scale the results to database size by factor of \( \sigma = \frac{|C|}{|S|} \).
1. \( S = \{s_1, s_2, ..., s_n\} \); 
\[ C = \sum_{i=1}^{n} c_i \] Query \( \Theta \) Transfer to \( \hat{\Theta} \)

2. We create 100 sample sets
3. The result of \( \hat{\Theta} \) with cardinality of each class are calculated
4. CLT-based closed form is used to generate error bounds

**Note:** Key element to guarantee accuracy of answers with error bounds (Confidence Intervals)

---

### Warehouse Leakage approximation

<table>
<thead>
<tr>
<th>Class ((s_i))</th>
<th>Without replacement</th>
<th>With replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Secret</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>Secret</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>Information</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>Official</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>Unclassified</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>Clearance</td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>Confidential</td>
<td>0.9</td>
<td>0.8</td>
</tr>
<tr>
<td>Restricted</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>0.3</td>
<td>0.2</td>
</tr>
</tbody>
</table>

The sample sizes needed to achieve different levels of relative error (averaged over 100 node from experiment cluster).

---

### DBaaS warehouse join size approximation

**Problem:** Random samples have larger error
**Solution:** Heterogeneous biased sampling method

- Sampling with respect to the repetition frequency of attribute of interest.
- For any value \( v \) if \( f_l(v) \geq T_l \), add it to the sample set otherwise with probability \( P_v = f_l(v) / T_l \), it will be included.
- To balance between sample size and accuracy a tunable threshold \( T_v \) for each collection \( C_v \) is defined.
- The higher value of \( T_v \) result in smaller sample size

**Cross-correlation Approximation**

\[
C_v = \begin{cases} 
\frac{f_l(v) \cdot f_R(v)}{T_l \cdot T_R} & \text{if } f_l(v) \geq T_l \text{ and } f_R(v) \geq T_R \\
\frac{f_l(v)}{T_l} & \text{if } f_l(v) < T_l \text{ and } f_R(v) > T_R \\
\frac{f_R(v)}{T_R} & \text{if } f_l(v) \geq T_l \text{ and } f_R(v) < T_R \\
\frac{f_l(v) \cdot f_R(v) \cdot \max(T_l, f_l(v), T_R, f_R(v))}{T_l f_l(v)} & \text{if } f_l(v) < T_l \text{ and } f_R(v) < T_R 
\end{cases}
\]
1. First practical Proxy for query processing on encrypted NoSQL.
2. Modest overhead (proportional to the desired security level).
3. No change to server and no change to applications.
4. Computation on the sample set (limited access to the original data set).
5. 1400X speed up in trade of less than 5% inaccuracy in response.
6. Leakage free secure DBaaS in a public cloud.
7. Fast attribute cross-correlation analysis in DBaaS warehouse level.

**Future work:** Sensitivity and cross-correlation analysis at cloud warehouse level besides individual Service Level Agreements enables CSPs to periodically compute Cross-Correlation Indexes (CCI).

---

**Conclusion**

**Selected References**


Selected References


