PhD Dissertation Defense

Mohammad Ahmadian
Department of Computer Science
Advisor: Professor Dan C. Marinescu
10/17/2017

On the security of NoSQL cloud database services

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
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.
1. Introduction

- Relational
  - MySQL
  - Oracle
  - IBM DB2
  - Sybase
  - MS SQL Server
  - MariaDB
  - PostgreSQL

- NoSQL

<table>
<thead>
<tr>
<th>Key-Value</th>
<th>Redis, Memcached, Hazelcast, Cosmos DB, Riak, KV, Couchbase, BoltDB, LevelDB, Berkeley DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
<td>Bigtable, Cassandra, Hbase, Vertica, Druid, Accumulo, Hypertable</td>
</tr>
<tr>
<td>Document</td>
<td>MongoDB, DynamoDB, Table storage, CouchDB, Couchbase, Q1px, RethinkDB, RavenDB, Terrastore</td>
</tr>
<tr>
<td>Graph</td>
<td>Neo4j, OrientDB, InfiniteGraph, AllegroGraph, SAP HANA</td>
</tr>
</tbody>
</table>

Problems of relational databases

1-Object-relational impedance mismatch

2-Scalability of database
2-Searchable Security Scheme for Cloud NoSQL

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

2. Searchable Security Scheme

Current settings

Application ➔ DBaaS

With SecureNoSQL

Application ➔ SecureNoSQL ➔ DBaaS
2. Searchable Security Scheme

Client

Proxy

DbaaS server

db.customers.find({
  balance: { $gte: 500000 },
  balance: { $lte: 900000 }
}).

{ "_id" : 1, "balance" : 1124442, "age" : 25,
  "eyeColor" : "blue", "name" : "White Roy",
  "sex" : "male", "company" : "ANGACCO",
  "email" : "white@bananco.com", "phone" : "(848) 466-2489", "address" : "313 Bath Avenue, Brookfield, Illinois, 7284", "son" : 11111111, "son" : "1978-03-30",
  "salary" : 7426712 }
**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)

---

**2. Searchable Security Scheme**

The organization of SecureNoSQL
## Searchable encryption

1- **Deterministic**

**Deterministic Encryption**

\[
\text{for } j = 1 \ldots n; \quad C_j = E_k(P_j); \quad P_j = D_k(C_j) \quad (1)
\]

2- **Random**

**Random Encryption**

\[
\text{for } j = 2 \ldots n; \quad C_j = E_k(P_j \oplus IV); \quad P_j = IV \oplus D_k(C_j) \quad (2)
\]

3- **Order Preserving Encryption**

**Order-Preserving Encryption**

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

4- **Additive Homomorphic**

**Additive Homomorphic Encryption**

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

## 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** ($\text{Count}$), **Group by** ($\text{group}$), **Equality match** ($\text{eq}$), **Not equal** ($\text{neq}$), and **Selects if value specified is in the array** ($\text{in}$).

### 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** ($\text{ge}$), **Less than** ($\text{lt}$), **Sort by** ($\text{sort}$), **Max** ($\text{max}$), and **Min** ($\text{min}$) as well as range queries on encrypted data.

### AHOM-Application

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

```javascript
db.customers.find({salary: {$gte: 5000}, balance: {$lt: 2000}});  
```

Performance

![Performance Graphs](image_url)
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

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)

Leakage function

The cross-correlation information leakage model

\[ \Psi_T(C_i, C_j) : \]
\[ \forall d \in C_i \land \forall d' \in C_j \]
\[ \text{if}(\mu(d, d') == \text{True}) \implies L = \{\text{Att} \mid \forall \text{Att} \in d' \land \text{Att} \notin d\} \]

The feasibility function \( \mu \)

\[ \mu(d, d') : \]
\[ \text{True} \text{ iff } \exists \text{Att}_i \in d \land \exists \text{Att}_j \in d' | \]
\[ [(\text{Att}_i, \text{key} == \text{Att}_j, \text{key}) \land (\text{Att}_i, \text{value} == \text{Att}_j, \text{value})] \]
\[ \text{False} \text{ Otherwise} \]
Example 1

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

Counter-measure

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

Drawbacks

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

S. Kenny et al. cracked encrypted database
Counter-measure

Filtering & Integrity verification

Performance of hash functions

Indexes over encrypted data

Index over encrypted data
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.

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 $\sigma = \frac{|C|}{|S|}$.
AQP based classification - experiment

1. \( S = \{s_1, s_2, \ldots, s_j\}; \quad C = \sum_{i=0}^{c} c_i \) 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**

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 $f(v) > T_i$, add it to the sample set otherwise with probability $P_v = f(v)/T_i$ it will be included.
- To balance between sample size and accuracy a tunable threshold $T_i$ for each collection $C_i$ is defined.
- The higher value of $T_i$ result in smaller sample size.

**Cross-correlation Approximation**

$$c_v : f_L(v)f_R(v)$$

- if $f_L(v) \geq T_L$ and $f_R(v) \geq T_R$
- if $f_L(v) < T_L$ and $f_R(v) \geq T_R$
- if $f_L(v) \geq T_L$ and $f_R(v) < T_R$
- if $f_L(v) < T_L$ and $f_R(v) < T_R$

$$f_L(v)f_R(v) \max\left(\frac{T_L}{f_L(v)}; \frac{T_R}{f_R(v)}\right)$$

- if $f_L(v) < T_L$ and $f_R(v) < T_R$
DBaaS warehouse joint size approximation

Experimental results:

- Social network
  - Name
  - Email
  - Gender
  - UserID
- Health
  - Patient ID
  - SSN
  - DOB
  - Disease
  - Address
  - Phone
- Phone directory
  - Phone#
  - Email
  - Name
  - Zip

56.34% 41.7%

Error percentage

Conclusion

4-Conclusion
Conclusion

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 (CCIs).

Selected References


Selected References


Discussion: Questions and Answers

Q&A

THANK YOU!