Securing Outsourced Database: Architecture for Protected Web Resource

KIRILL SHATILOV, SERGEY KRENDELEV, DIANA ANISUTINA, ARTEM SUMANEEV AND EVGENY OGURTSOV

2015 CEE-SEC(R) Software Engineering Conference in Russia
Current research is performed in Laboratory NSU-Parallels in Novosibirsk State University under financial support of the Ministry of education and science of Russia (contract № 02.G25.310054). Science leader - Krendelev Sergey Fedorovich
IN GENERAL

- OPE & FHE encryptions
- SQL- & DBMS response parser
- Protected Web Resource
OUTLINE

• Motivation

• Methodology and Design
  - encryptions
  - syntax processor
  - components configuration

• Achieved results

• Future challenges

• Summary
MOTIVATION

SCENARIO

- SQL DBMS backed Web resource

THREAT 1

- Insider

THREAT 2

- Adversary
SOLUTION

POINT 1 · Encrypted database
POINT 2 · Intermediate processing components
POINT 3 · Trust zones
POINT 1. ENCRYPTION LIBRARY

**Deterministic and Probabilistic encryptions**
- Strong security
- Text Data

**Order Preserving Encryption**
- Order operations over ciphertexts
- Secure indexes, dates

**Fully Homomorphic Encryption**
- Multiplication & addition over ciphertexts
- Math & commerce
• **FHE**

*Fully Homomorphic Encryption for Secure Computations in Protected Database*
Darya Chechulina, Kirill Shatilov, Sergey Krendelev,
Position Papers of the 2015 Federated Conference on Computer Science and Information Systems, pp. 125-131

• **OPE**

*Order-preserving encryption schemes based on arithmetic coding and matrices*
Maria Usoltseva, Sergey Krendelev, Mikhail Yakovlev,
Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, pp 891-899
POINT 2. PROCESSING COMPONENTS

SQL processing:
1. parsing
2. encrypting
3. reconstructing

Response processing (decrypting)

Encryption LIB
META & KEYS

APPLICATION

Encrypted DATABASE
POINT 2. SYNTAX PROCESSING

Create statement processing:

1. Encryption’s keys are generated or chosen.
2. Determination of number, names, types and constraints of output columns.
3. Correct SQL string is created according to determined information.
4. Anonimisation of columns' and tables' names.
5. Modified statement is sent to DBMS.

DML statements processing:

1. Data's extraction
2. Data's encryption
3. Columns' names synchronization
4. Math correction (in some cases)
5. Decryption of response (if needed)
METAFILE STORAGE

- In-memory database
- Constant backups
- Encrypted on HDD
- Storing:
  - Encryption keys
  - Initial column info
  - Output column format
  - JOIN groups info
POINT 2. PROCESSING CHALLENGES

- Multiple output columns

\[
\text{Encryption}(value) = (a, b, c, \ldots)
\]

\[
\text{SELECT } value \text{ FROM } table\_name \quad \Rightarrow \quad \text{SELECT } a, b, c, \ldots \text{ FROM } table\_name
\]
POINT 2. PROCESSING CHALLENGES

- Encryption specific math

\[ \text{FHEncryption}(\text{value}) = (a)(b) \]

\[ \text{ciphertext + ciphertext} = \text{UDF (a1, b1, a2, b2, Multiplication Table)} \]

**Multiplication Table \(~ 5000 \text{ values} \)**

\[ \text{SELECT \text{SUM} (values) FROM table_name} \quad \Rightarrow \quad \text{SELECT \text{UDF\_SUM} (...) FROM table_name} \]
POINT 3. ZONING & CONFIGURATION

1. Centralized

- Unmodified
- Encryption Aware Site Engine

Crypto Environment

Trusted zone

Encrypted DATABASE

Untrusted zone
POINT 3. ZONING & CONFIGURATION

2. Distributed

- Client0
  - Site Engine
  - Crypto Environment
  - MetaFile Local Copy
  - ...

- ClientN
  - Site Engine
  - Crypto Environment
  - MetaFile Local Copy

MetaFile Sync

Encrypted DATABASE

Trusted zone

Untrusted zone
RESULTS. PRACTICAL IMPLEMENTATION

Client
login
password

WordPress CMS

nginx +
PHP FPM

Hosting

MySQL
Proxy

.LUA
SQL
processing

Response
processing

Encryption
LIB

META &
KEYS

Crypto Environment

MySQL
DBMS
## RESULTS. APPLIED ENCRYPTION

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>tags, headers</td>
<td>text</td>
<td>deterministic</td>
</tr>
<tr>
<td>post, comments text</td>
<td>long text</td>
<td>probabilistic</td>
</tr>
<tr>
<td>post, comments, events date</td>
<td>date</td>
<td>OPE</td>
</tr>
<tr>
<td>user email, name</td>
<td>text</td>
<td>deterministic</td>
</tr>
<tr>
<td>user password</td>
<td>text</td>
<td>deterministic</td>
</tr>
<tr>
<td>ratings, order terms</td>
<td>integer</td>
<td>OPE</td>
</tr>
</tbody>
</table>
mysql> SELECT * FROM wp_posts;
RESULTS. EVALUATION

Initialization: +50%
Uploading: +20%
Retrieving: +15%

Average Performance Overhead

Database size increase: +60%
FUTURE CHALLENGES

- Cross-platform build
- PostgreSQL
- Multithreading environment
- Memory optimization
SUMMARY

WHAT? • Real-time application's data protection

WHY? • Outsourced data's privacy

HOW? • OPE & FHE

AND..? • Real life applications and development goals
THANK YOU