**Operon**: An Encrypted Database for Ownership-Preserving Data Management

Sheng Wang  Yiran Li  Huorong Li  Feifei Li  Chengjin Tian  Le Su  Yanshan Zhang  Yubing Ma  Lie Yan  Yuanyuan Sun  Xuntao Cheng  Xiaolong Xie  Yu Zou
Databases run in private domain, **system owners** inherently have full access to data

- **Change 1**: data flow to other processing components (entities) are out of control
- **Change 2**: insiders could compromise the outsourced computing infrastructure
Existing Solutions Fail to Protect Data in Entire Business Process

- Existing encrypted databases protect the **confidentiality** of outsourced data
  - **Approach**: using cryptographic primitives or trusted execution environments (TEE)
  - **Assumption**: the entity directly interacting with database is trusted and can touch sensitive data
  - **Limitation**: cannot protect application subsystems hosted or controlled by other entities

**Operon**: An Encrypted Database for Ownership-Preserving Data Management
**Operon**: TEE-Based Ownership-Preserving Data Management

- Re-establish the private domain assumption
  - Decoupling data ownership and system ownership by granting only necessary operations
  - Providing both confidential data computation and policy enforcement with TEE

Data remains in ciphertext except the two secure endpoints

---

**Operon**: An Encrypted Database for Ownership-Preserving Data Management
Paradigm for Exclusive Data Accessibility and Behavior Control

- **Ownership-preserving database (OPDB) paradigm**
  - **Principle 1**: An entity can not access the sensitive data content without the data owner’s authorization
  - **Principle 2**: An entity can only conduct authorized operations on sensitive data without knowing its content
  - **Principle 3**: An entity can only use authorized operations to learn properties of sensitive data

- **OPDB operations**
  - **Operator**: ops. leak nothing
  - **Measure**: ops. return specific properties

- **OPDB roles and responsibilities**
  - **Data Owner** *(DO, Factory)*
    - Who exclusively controls data accessibility and behaviors
  - **Data Manipulator** *(DM, Software Vendor)*
    - Who determines (alone or joint with DO) the purposes and means of data processing
  - **Data Processor** *(DP, Cloud Service)*
    - Who processes data on behalf of DM

---

**Operon** Default Primitive Configuration Example

<table>
<thead>
<tr>
<th>Operand Type</th>
<th>Primitive</th>
<th>Default</th>
<th>Leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>enc_{int,float}</code></td>
<td>+ - × ÷ % EXP</td>
<td>Operator None</td>
<td></td>
</tr>
<tr>
<td><code>enc_text</code></td>
<td>SUBSTRING</td>
<td>Operator None</td>
<td></td>
</tr>
<tr>
<td><code>enc_text</code></td>
<td>LIKE</td>
<td>Measure Matching result</td>
<td></td>
</tr>
<tr>
<td><code>enc_{int,float,text}</code></td>
<td>= ≠ &gt; &lt; ≥ ≤</td>
<td>Measure Operands order</td>
<td></td>
</tr>
</tbody>
</table>
Implementing OPDB Using Fine-Grained Behavior Control

- **Operon** proposes behavior control list (BCL) to control the behavior of data
  - **Content:** issuer & subject IDs, data key IDs, operation, preprocessing, postprocessing, *etc.*
  - **Security:** using TEE to **validate** BCL authenticity and **enforce** the defined data behaviors

**Example:** granting DBA **equality** and **aggregation** ops. for outsourced diagnosis

- DBA can perform SQL queries and locate problems **like on plaintext** database
- Data owner clearly **knows** (from BCL) what the DBA might learn from the data
- Data owner can specify proper desensitization rules based on responsibilities of DBA

**BCL supports format-preserving preprocessing and postprocessing actions**

<table>
<thead>
<tr>
<th># omitted fields</th>
<th>i_id: &lt;user-id&gt;</th>
<th>prep: MASK_TAIL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_id: &lt;dba-id&gt;</td>
<td>postp: NULL</td>
<td></td>
</tr>
<tr>
<td>ops: [EQUAL_P, NE_P]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Granted Equality Measure BCL**

<table>
<thead>
<tr>
<th># omitted fields</th>
<th>i_id: &lt;user-id&gt;</th>
<th>prep: NULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_id: &lt;dba-id&gt;</td>
<td>postp: MASK_TAIL2</td>
<td></td>
</tr>
<tr>
<td>ops: [SUM_C, AVG_C]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Granted Agg. Measure BCL**
Operon Architecture: Flexibility, Stateless & Functionality

- Flexible arch. for easy integration: only computation module resides in enclave

- Stateless computation module enables SQL to operate ciphers as plaintexts
  - **Failing-fast**: throws error message (e.g., key miss, buffer overflow), retry with new parameter
  - **Caching soft-states**: improves efficiency and can be discarded or generated as will (e.g., key)

- Stateless computation enables **connection pool** and **parallel processing**

```
SELECT AVG(salary) FROM EM_Employee WHERE los > 'xA2E3'
```
Two-layer key hierarchy: data encryption key (DEK) & master encryption key (MEK)

- **Benefit**: reduces key management cost for DOs & enables fine-grained cipher management
- **Cipher context (CC)**: shorthand for the full cryptographic metadata (e.g., DEK ID, algorithm)

Key management system utilizes database tables and TEE sealing

---

### Table Examples

**MEK Table**

<table>
<thead>
<tr>
<th>Sealed MEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3e9fba02</td>
</tr>
</tbody>
</table>

**Cipher Context Table**

<table>
<thead>
<tr>
<th>CC ID</th>
<th>DEK ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>452</td>
<td>1039</td>
</tr>
</tbody>
</table>

**DEK Table**

<table>
<thead>
<tr>
<th>DEK ID</th>
<th>Encrypted DEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1039</td>
<td>0x890ae3bb</td>
</tr>
</tbody>
</table>

**MEK Cache**

<table>
<thead>
<tr>
<th>MEK Plaintext</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x98a0b453</td>
</tr>
</tbody>
</table>

**DEK Cache**

<table>
<thead>
<tr>
<th>CC ID</th>
<th>DEK Plaintext</th>
</tr>
</thead>
<tbody>
<tr>
<td>452</td>
<td>0x24a3f101</td>
</tr>
</tbody>
</table>

**Cipher Context**

<table>
<thead>
<tr>
<th>CC ID</th>
<th>DEK Plaintext</th>
</tr>
</thead>
<tbody>
<tr>
<td>452</td>
<td>0x24a3f101</td>
</tr>
</tbody>
</table>

---

**Operon**: An Encrypted Database for Ownership-Preserving Data Management

No additional trust required, except TEE and Operon enclave
**Operon** Provides Full-Featured Encrypted Database Experience

- **Type system:** evaluate **mixed expressions** of different data types
  - **Conventional:** combination of implicit type conversions, default rules, priorities, *etc.*

- **Operon:** lattice of encrypted types, match signature by finding the least upper bound type

- **Indexing:** ECall-less order-revealing encryption (ORE) measure
  - Extending ORE to support decryption and floating data type

- **Client-side:** SDK and **OpeJDBC**
  - **SDK:** key management functionalities, local DEK cache, data encryption and decryption
  - **OpeJDBC:** performs automatic data encryption/decryption by calling the SDK

---

**SELECT** salary  
**FROM** EM_Employee  
**WHERE** los > (SELECT AVG(los)  
**FROM** EM_Employee)
Performance Evaluation Setup

- **Hardware specification**
  - **Product:** ApsaraDB RDS for PostgreSQL
  - **CPU:** 24 vCPU with SGX
  - **Memory:** 192 GB memory
  - **Storage:** 2TB SSD

- **Benchmark configuration**
  - **Sysbench:** index performance
    - 32 tables of $10^6$ records
    - All columns encrypted
  - **TPC-C:** transaction performance
    - Default 4 instances simulating 128 clients
    - Default 256 warehouses
    - Encryption: ID columns are non-sensitive sequence numbers
      - **6-Column:** encrypt name columns and address columns (the same as Always Encrypted)
      - **58-Column:** encrypt all 58 columns except the ID columns

---

**Sysbench SQL Templates**

<table>
<thead>
<tr>
<th>Point Query</th>
<th>Range Query</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SELECT c FROM sbtest WHERE id = ?</code></td>
<td><code>SELECT c FROM sbtest WHERE id BETWEEN ? AND ?</code></td>
</tr>
</tbody>
</table>
ORE-Measure Improves Index Performance

- Performing ORE-measures outside TEE achieves near-plaintext index performance

![Normalized Throughput](#)

Point Query Index Performance

- CMP-measure
- ORE-measure

~84%

Range Query Index Performance

- CMP-measure
- ORE-measure

~96%

Operon: An Encrypted Database for Ownership-Preserving Data Management
**Operon** Preserves Data Ownership with Low Overhead

- Encryption setting: 6 columns as Always Encrypted & all 58 columns except IDs

---

**Operon**

- An Encrypted Database for Ownership-Preserving Data Management

---

**TPC-C Warehouses**

- Normalized Throughput
  - 64, 128, 256, 512, 1024 Warehouses
  - Orange bars: 6-Col-RND
  - Gray bars: 6-Col-DET
  - Blue bars: 58-Col-RND
  - Yellow bars: 58-Col-DET

- Throughput:
  - ~97%
  - ~70%

---

**TPC-C Terminals**

- Normalized Throughput
  - 64, 128, 192, 256 Terminals
  - Orange bars: 6-Col-RND
  - Gray bars: 6-Col-DET
  - Blue bars: 58-Col-RND
  - Yellow bars: 58-Col-DET

- Throughput:
  - ~97%
  - ~70%
Conclusion: Decoupling Data Ownership & System Ownership

- **OPDB**: data owner **exclusively controls** its sensitive data across multiple entities

**Operon**: TEE-based encrypted database that follows the OPDB paradigm

- **Architecture**: adapts to various TEEs and databases, built-in key management
- **BCL**: preserves the data ownership by taking operation behavior into consideration
- **Features**: connection pool, mixed-type expressions, OpeJDBC, ORE indexing, etc.
- **Performance**: 71% - 97% of the plaintext database performance under TPC-C benchmark

---

**Operon**: An Encrypted Database for Ownership-Preserving Data Management

---

Thanks

sh.wang@alibaba-inc.com