**DBMask: Encrypted Query Processing over an Encrypted Database**

Mohamed Nabeel, Jianpeng Cao, Mohamed Sarfraz, Elisa Bertino
Dept. of Computer Science, Purdue University

From Private to Public Relational Databases in the Cloud

**Problem Statement**

1. Upload encrypted relational data
2. Retrieve encrypted data
3. Transparent
4. Result Set
5. Encrypted Result Set
6. Untrusted Cloud

**Challenges**

1. How can users query and retrieve encrypted data without the cloud decrypting the data or query?
2. How to provide fine-grained access to the data over encrypted relational data?

How to support (1) and (2) while allowing relational database operations?

**Our Approach: SQL Aware Encryption**

- Value
- BE Value¹
- Blinded Value²

For fine grained (attribute based) access control
For encrypted SQL queries

1. Nabeel et al., Privacy preserving policy based content sharing in the cloud, TKDE 2012
2. Nabeel et al., Efficient privacy preserving publish subscribe systems, SACMAT 2012

**Example**

<table>
<thead>
<tr>
<th>Patient</th>
<th>Age</th>
<th>Doctor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>15</td>
<td>Sam</td>
</tr>
<tr>
<td>Bob</td>
<td>14</td>
<td>Sam</td>
</tr>
<tr>
<td>Troy</td>
<td>19</td>
<td>Pat</td>
</tr>
</tbody>
</table>

Access Control Policies:
- A doctor can access only its patients’ records
- A patient can access only its record

Sam: SELECT * from Patient WHERE Age > 14;

<table>
<thead>
<tr>
<th>Patient</th>
<th>BE_Age</th>
<th>B_Age</th>
<th>Doctor</th>
<th>PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_{a}(Alice)</td>
<td>E_{a}(15)</td>
<td>B(15)</td>
<td>E_{a}(Sam)</td>
<td>1</td>
</tr>
<tr>
<td>E_{b}(Bob)</td>
<td>E_{b}(14)</td>
<td>B(14)</td>
<td>E_{b}(Sam)</td>
<td>2</td>
</tr>
<tr>
<td>E_{t}(Troy)</td>
<td>E_{t}(19)</td>
<td>B(19)</td>
<td>E_{t}(Pat)</td>
<td>3</td>
</tr>
</tbody>
</table>

Proxy: SELECT * from Patient WHERE UDF(B_Age, Trapdoor(14), '>') = 1;