**Goal:** Increase efficiency of cryptographic techniques

### Existing Problem

Cryptographic techniques are:

1. Not efficient
2. Prone to various attacks

### Partition Computation

- Partition the data into sensitive and non-sensitive
- Sensitive data is cryptographically secure
- Non-sensitive data is in cleartext

### Query Binning

**Partition Data Security**

- No linking of sensitive and non-sensitive values
- Not revealing #tuples with each sensitive value

**Idea of Query Binning**

- Distribute “values” into a matrix
- Rows become sensitive bins
- Columns become non-sensitive bins

### Performance

\[
\eta = \frac{\text{Cost}_{\text{crypt}}(\|SB\|,S)}{\text{Cost}_{\text{Crypt}}(1,D)} + \frac{\text{Cost}_{\text{plain}}(\|NSB\|,NS)}{\text{Cost}_{\text{Crypt}}(1,D)}
\]

### Inference Attacks due to Data Partitioning

**Adversarial view**

**what the adversary observes**

<table>
<thead>
<tr>
<th>Query value</th>
<th>Returned tuples</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>From sensitive data</td>
</tr>
<tr>
<td>Adam</td>
<td>From non-sensitive data</td>
</tr>
</tbody>
</table>

### Interesting Facts

- Works for any number of sensitive and nonsensitive values
- Improves an underlying cryptographic technique by preventing output-size and frequency-count attacks
- Supports conjunctive selection, join, and range queries

### Reference