Privacy-preserving and Authenticated Data Cleaning on Outsourced Databases

Thesis Defense

Boxiang Dong

THESIS COMMITTEE:
Advisor: Prof. Wendy Hui Wang
    Prof. Yingying Chen
    Prof. David Naumann
    Prof. Antonio Nicolosi

Department of Computer Science
Stevens Institute of Technology

December 1, 2016
Dirty Data

Real-world datasets, particularly those from multiple sources, tend to be *dirty*.

**Inaccuracy**  Multiple records that refer to the same entity

**Inconsistency**  Violation of integrity constraints

**Incompleteness**  Missing data values

<table>
<thead>
<tr>
<th>Name</th>
<th>Street</th>
<th>City</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Leonard</td>
<td>NY</td>
<td>518-457-5181</td>
</tr>
<tr>
<td>John</td>
<td>Lenard</td>
<td>NY</td>
<td>518-457-5181</td>
</tr>
<tr>
<td>Kevin</td>
<td>LA</td>
<td></td>
<td>213-974-3211</td>
</tr>
<tr>
<td>Mike</td>
<td>Main</td>
<td>Phil</td>
<td>518-457-5181</td>
</tr>
</tbody>
</table>

The ubiquitous dirty data: 40% of companies have suffered losses, problems, or costs due to data of poor quality [Eck02].
Data Cleaning

Data cleaning aims at detecting and removing errors, duplications, missing values, and inconsistencies to improve data quality.

- Data deduplication
- Data inconsistency repair
- Data imputation

Data cleaning is a labor-intensive and complex process. It can be NP-complete [BFFR05].
Data-Cleaning-as-a-Service

Outsourcing the data to a third-party data cleaning service provider provides a cost-effective way. E.g., Google’s OpenRefine, Melissa Data.

Client with limited computational resources
Server computationally powerful
Security Concerns

The third-party server is untrusted.

**Result integrity** The server may return incorrect data cleaning result.

- Software bugs
- Intention to save computational cost

**Data privacy** The outsourced data may include sensitive personal information.

- Medical information
- Financial record
**Thesis topic:** Privacy-preserving and authenticated data cleaning on outsourced databases
**Thesis topic:** Privacy-preserving and authenticated data cleaning on outsourced databases

- **Security & Privacy**
  - Privacy
  - Authentication

- **Data Cleaning**
  - Inconsistency Repair
  - Deduplication

- [BigDataSecurity’16]
- [ICDE’17] (Under Review)
**Thesis topic:** Privacy-preserving and authenticated data cleaning on outsourced databases

Diagram:
- Security & Privacy
  - Privacy
  - Authentication
- Data Cleaning
  - Inconsistency Repair
  - Deduplication

[CIKM’14]
**Thesis topic:** Privacy-preserving and authenticated data cleaning on outsourced databases
Thesis topic: Privacy-preserving and authenticated data cleaning on outsourced databases
Related Work

Data cleaning
- Data deduplication [GIJ+01, SAA10, YLKG07]
- Data inconsistency repair [PEM+15, BFG+07, BFFR05]

Privacy-preserving outsourced computation
- Encryption [SV10, PRZB12]
- Encoding [EAMY+13, CC04]
- Secure multiparty computation [TOEY11, LZL+15]
- Differential privacy [CMF+11, AHMP15]

Verifiable computing
- General-purpose verifiable computing [SVP+12, PHGR13]
- Function-specific verifiable computing [DLW13, LWM+12]
Outline

1 Introduction

2 Research Results
   • Authentication of Outsourced Data Deduplication
     • Verification of Similarity Search Approach ($VS^2$)
     • Embedding-based Verification of Similarity Search Approach ($E-VS^2$)
     • Experiments
     • Privacy-preserving Outsourced Data Deduplication
     • Privacy-preserving Outsourced Data Inconsistency Repair

3 Research beyond the Thesis

4 Future Plan

5 Conclusion
Authentication of Outsourced Data Deduplication

Boxiang Dong, Wendy Hui Wang.
(Acceptance rate = 25%)
Data Deduplication

Data deduplication Eliminate near-duplicate copies.

- Record matching: Detect near-duplicate copies.

\[ \{ s \mid s \in D, DST(s, s_q) \leq \theta \} \]

\( \theta \): similarity threshold

\( DST \): edit distance
Data Deduplication

Data deduplication Eliminate near-duplicate copies.

- Record matching: Detect near-duplicate copies.

<table>
<thead>
<tr>
<th>RID</th>
<th>Name</th>
<th>Street</th>
<th>City</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>John</td>
<td>Leonard</td>
<td>NY</td>
<td>45</td>
</tr>
<tr>
<td>r2</td>
<td>Kevin</td>
<td>Wicks</td>
<td>LA</td>
<td>31</td>
</tr>
<tr>
<td>r3</td>
<td>Mike</td>
<td>Main</td>
<td>Phil</td>
<td>22</td>
</tr>
</tbody>
</table>

$s_q = (\text{John, Lenard, NY, 45})$

$\theta = 2$

$\{r_1\}$
The client (data owner) outsources the record matching service to the untrusted server.

Assumption: The client is aware of the edit distance metric. We want to make sure that $R^S$ is both sound and complete.

**Soundness**  $\forall s \in R^S, s \in D$ and $DST(s, s_q) \leq \theta$.

**Completeness**  $\forall s \in D$ s.t. $DST(s, s_q) \leq \theta, s \in R^S$. 

\[
R^S = \{s | s \in D, DST(s, s_q) \leq \theta\}
\]
We aim at an authentication framework that satisfies the following objectives.

- Soundness violation
  \[ \exists s \in R^S, \text{ but } s \notin D \]
  \[ \exists s \in R^S, \text{ but } DST(s, s_q) > \theta \]

- Completeness violation
  \[ \exists s \in D \text{ s.t. } DST(s, s_q) \leq \theta \]
  \[ \text{but } s \notin R^S \]

- Supports efficient verification

- Scales well with big data
Merkle tree is a generalization of hash lists and hash chains.

It allows efficient and secure verification of the contents of large data structures.

Hash is computationally more efficient than edit distance calculation.
**Preliminary - $B^{ed}$-Tree**

$B^{ed}$-Tree [ZHOS10] is a string indexing structure.

- Sort the strings in dictionary order.
- Store the longest common prefix (LCP) of the enclosed strings in every node.
$B^{ed}$-Tree [ZHOS10] is a string indexing structure.

- $\forall N$, calculate $MIN\_DST(s_q, N.LCP)$. 
Preliminary - $B^{ed}$-Tree

$B^{ed}$-Tree [ZHOS10] is a string indexing structure.

![Diagram of $B^{ed}$-Tree]

- If $MIN\_DST(s_q, N.LCP) > \theta$, then $N$ is a MF-node.
- All strings covered by a MF-node must be dissimilar to $s_q$.
- Avoid the edit distance calculation for NC-strings.
- Perform well with memory constraints.
Embedding maps strings into Euclidean points in a similarity-preserving way.

- Euclidean distance calculation is much more efficient than edit distance computing, i.e., $O(dst(p_i, p_j)) \ll O(DST(s_i, s_j))$.
- SparseMap[HS] is a contractive embedding approach, i.e., $dst(p_i, p_j) \leq DST(s_i, s_j)$.
- The complexity is $O(cn^2)$, where $c$ is a small constant, and $n$ is the number of strings.
Solution in a Nutshell

We require the server to construct *verification object (VO)* to demonstrate the soundness and completeness of the result.

\[
\sigma \leftarrow \text{setup}(D)
\]

\[
(R^S, VO) \leftarrow \text{search}(D, s_q, \theta)
\]

\[
(R^S / \bot) \leftarrow \text{verify}(R^S, VO, \sigma)
\]

The client is able to efficiently detect any unsound or incomplete result returned by the server by checking the *VO*. 
Outline

1. Introduction
2. Research Results
   - Authentication of Outsourced Data Deduplication
   - Verification of Similarity Search Approach ($VS^2$)
   - Embedding-based Verification of Similarity Search Approach ($E-VS^2$)
   - Experiments
     - Privacy-preserving Outsourced Data Deduplication
     - Privacy-preserving Outsourced Data Inconsistency Repair
3. Research beyond the Thesis
4. Future Plan
5. Conclusion
VS² - Setup

We propose an authenticated string indexing structure, named \( MB \)-tree (Merkle \( B^{ed} \)-tree).

\[
\text{Sig}(T) = \text{sign}(h_{N_1}) \\
N_1 \\
p_{N_2} \mid p_{N_3} \mid LCP_{N_1} \mid h_{N_1} \\
h_{N_1} = h(h_{N_2} \mid h_{N_3} \mid h(LCP_{N_1})) \\
N_2 \\
p_{N_4} \mid p_{N_5} \mid LCP_{N_2} \mid h_{N_2} \\
h_{N_2} = h(h_{N_4} \mid h_{N_5} \mid h(LCP_{N_2})) \\
N_4 \mid \begin{array}{c} s_1 \mid s_2 \mid s_3 \mid LCP_{N_4} \mid h_{N_4} \\ s_4 \mid s_5 \mid s_6 \mid LCP_{N_5} \mid h_{N_5} \end{array} \\
N_5 \mid \begin{array}{c} s_7 \mid s_8 \mid s_9 \mid LCP_{N_6} \mid h_{N_6} \\ s_{10} \mid s_{11} \mid s_{12} \mid LCP_{N_7} \mid h_{N_7} \end{array} \\
N_6 \mid N_7 \\
N_3 \mid \begin{array}{c} p_{N_6} \mid p_{N_7} \mid LCP_{N_3} \mid h_{N_3} \end{array}
\]

- The client signs the hash value in the root, and only keeps the signature of the \( MB \)-tree locally.
- The hash function is more efficient than edit distance calculation.
VS²-VO Construction

The server searches for the similar strings and constructs VO by traversing the MB-tree.

- Include all the C-strings and similar strings in VO.
- Substitute the large amount of NC-strings with the MF-nodes.
The client checks the soundness of completeness of $R^S$ by verifying the VO.

- soundness violation:
  - $\exists s \in R^S$, but $s \not\in D$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$

- completeness violation:
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$
  - but $s \not\in R^S$

\[
\begin{align*}
R^S &= \{s_1, s_2\} \\
VO &= \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7}))\}
\end{align*}
\]
The client checks the soundness and completeness of \( R^S \) by verifying the VO. 

\[
\begin{align*}
\text{soundness violation} & \quad \begin{cases} 
\exists s \in R^S, \text{ but } s \not\in D \\
\exists s \in R^S, \text{ but } DST(s, s_q) > \theta
\end{cases} \\
\text{completeness violation} & \quad \begin{cases} 
\exists s \in D \text{ s.t. } DST(s, s_q) \leq \theta \\
\text{but } s \not\in R^S
\end{cases}
\end{align*}
\]

\[\text{Catch if } \text{Sig}(T) \text{ matches the local copy}\]

\[\text{Sig}(T) = \text{sign}(h_{N_1})\]

\[\text{Compute } \text{Sig}(T) \text{ from VO}\]

\[R^S = \{s_1, s_2\}\]

\[VO = \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7})))\}\]
The client checks the soundness and completeness of $R^S$ by verifying the $VO$.

- **Soundness violation**
  - $\exists s \in R^S$, but $s \not\in D$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$
- **Completeness violation**
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$
  - $s \not\in R^S$

**Compute $Sig(T)$ from $VO$**

\[ \forall s \in R^S, \text{ check if } DST(s, s_q) \leq \theta \]

\[ \forall C\text{-string } s, \text{ check if } DST(s, s_q) > \theta \]

\[ \forall \text{MF-node } N, \text{ check if } MIN\_DST(N.LCP, s_q) > \theta \]

$s_q = \text{"Celestine"}$

\[
\begin{align*}
R^S &= \{s_1, s_2\} \\
VO &= \{((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7}))\}\end{align*}
\]

- **For similar strings**
  \[
  \begin{align*}
  DST(s_1, s_q) &= 4 \\
  DST(s_2, s_q) &= 3 < 4
  \end{align*}
  \]

- **For C-strings**
  \[
  \begin{align*}
  DST(s_3, s_q) &= 5 > 4 \\
  DST(s_4, s_q) &= 9 > 4 \\
  DST(s_5, s_q) &= 9 > 4 \\
  DST(s_6, s_q) &= 8 > 4 \\
  DST(s_7, s_q) &= 8 > 4 \\
  DST(s_8, s_q) &= 8 > 4 \\
  DST(s_9, s_q) &= 8 > 4
  \end{align*}
  \]

- **For MF-node**
  \[
  MIN\_DST(LCP_{N_7}, s_q) = 6 > 4
  \]

**10 DST calculations**
Outline

1 Introduction

2 Research Results
   • Authentication of Outsourced Data Deduplication
     • Verification of Similarity Search Approach ($VS^2$)
     • Embedding-based Verification of Similarity Search Approach ($E-VS^2$)
     • Experiments
       • Privacy-preserving Outsourced Data Deduplication
       • Privacy-preserving Outsourced Data Inconsistency Repair

3 Research beyond the Thesis

4 Future Plan

5 Conclusion
E-VS² - Setup

- The client constructs the MB-tree.
- The client applies SparseMap to embed strings into Euclidean points.

Key idea  For any C-string $s$, if $\text{dst}(p, p_q) > \theta$, it must be true that $\text{DST}(s, s_q) > \theta$. 

$\text{Sig}(T) = \text{sign}(h_{N_1})$

$N_1$

$h_{N_1} = h(h_{N_2} | h_{N_3} | h(LCP_{N_1}))$

$N_2$

$h_{N_2} = h(h(s_1) | h(s_2) | h(s_3) | h(LCP_{N_2}))$

$N_3$

$h_{N_3} = h(h(s_4) | h(s_5) | h(s_6) | h(LCP_{N_3}))$

$N_4$

$h_{N_4} = h(h(s_1) | h(s_2) | h(s_3) | h(LCP_{N_4}))$

$N_5$

$h_{N_5} = h(h(s_4) | h(s_5) | h(s_6) | h(LCP_{N_5}))$

$N_6$

$h_{N_6} = h(h(s_7) | h(s_8) | h(s_9) | h(LCP_{N_6}))$

$N_7$

$h_{N_7} = h(h(s_{10}) | h(s_{11}) | h(s_{12}) | h(LCP_{N_7}))$
**E-VS$^2$ - VO Construction**

**Distant Bounding Hyper-rectangle (DBH)** A hyper-rectangle $R$ in the Euclidean space is a DBH if $\min_{\text{dst}}(p_q, R) > \theta$.

**DBH-String** For any C-string $s$, if $\text{dst}(p, p_q) > \theta$, we call it a DBH-string.

**FP-String** For any C-string $s$, if $\text{dst}(p, p_q) \leq \theta$, we call it a FP-string.

**Key idea**
- To save the verification cost at the client side, the server should organize the set of DBH-strings into a small number of DBHs.
- By only checking the Euclidean distance between the query point $p_q$ and the DBHs, the client assures that all DBH-strings are dis-similar to $s_q$. 
\( E-VS^2 - VO \) Construction

\[
\begin{align*}
\text{Similar Strings} & \quad \text{DBH-Strings} & \quad \text{C-Strings} \quad \text{FP-Strings} & \quad \text{NC-Strings} \\
\text{MF-Node} \\
\end{align*}
\]

\[ s_q = \text{“ Celestine ”} \]
\[ \theta = 4 \]

\[
\begin{align*}
\text{Sig}(T) &= \text{sign}(h_{N_1}) \\
N_1 &\begin{bmatrix}
p_{N_2} & p_{N_3} & LCP_{N_1} & h_{N_1} 
\end{bmatrix} & 0 \\
N_2 &\begin{bmatrix}
p_{N_4} & p_{N_5} & LCP_{N_2} & h_{N_2} 
\end{bmatrix} & 0 \\
N_3 &\begin{bmatrix}
p_{N_6} & p_{N_7} & LCP_{N_3} & h_{N_3} 
\end{bmatrix} & 0 \\
N_4 &\begin{bmatrix}
s_1 & s_2 & s_3 & LCP_{N_4} & h_{N_4} 
\end{bmatrix} & 3 \\
N_5 &\begin{bmatrix}
s_4 & s_5 & s_6 & LCP_{N_5} & h_{N_5} 
\end{bmatrix} & 0 \\
N_6 &\begin{bmatrix}
s_7 & s_8 & s_9 & LCP_{N_6} & h_{N_6} 
\end{bmatrix} & 6 \\
N_7 &\begin{bmatrix}
s_{10} & s_{11} & s_{12} & LCP_{N_7} & h_{N_7} 
\end{bmatrix} & 1 \\
\end{align*}
\]
**E-VS² - VO Construction**

\[ \text{Sig}(T) = \text{sign}(h_{N_1}) \]

\[ \begin{array}{c}
s_q = \text{"Celestine"} \\
\theta = 4
\end{array} \]

\[
\begin{array}{c|c|c|c|c}
\text{N}_1 & \text{N}_2 & \text{N}_3 & \text{N}_4 & \text{N}_5 \\
\text{p}_{N_2} & \text{p}_{N_3} \text{LCP}_{N_1} & \text{h}_{N_1} & \text{p}_{N_4} \text{p}_{N_5} \text{LCP}_{N_2} & \text{h}_{N_2} \\
\end{array}
\]

**Similar Strings**
- Similar Strings
- DBH-Strings
- FP-Strings
- NC-Strings

**MF-Node**

\[
\begin{array}{c}
p_2 \\
p_3 \\
p_4 \\
p_5 \\
p_6 \\
p_7 \\
p_8 \\
p_9 \\
\theta
\end{array}
\]

\[
\begin{array}{c}
p_{10} \\
p_{11} \\
p_{12}
\end{array}
\]
Theorem (NP-Completeness of DBH Construction)

Given a query string $s_q$, and a set of DBH-strings $\{s_1, \ldots, s_t\}$, let $\{p_1, \ldots, p_t\}$ be their Euclidean points. It is a NP-complete problem to construct a minimum number of rectangles $\mathcal{R} = \{R_1, \ldots, R_k\}$ s.t.

1. $\forall i \neq j$, $R_i$ and $R_j$ do not overlap; and
2. $\forall p_i$, there exists a $R_j$ s.t. $p_i$ is included in $R_j$.

- We design an efficient heuristic algorithm for the server to construct a small amount of DBHs.
- The complexity is cubic to the number of DBH-strings.
The server includes the **DBHs in the VO**.

$$s_q = \text{“ Celestine ”}$$

$$\theta = 4$$

$$\text{Sig}(T) = \text{sign}(h_{N_1})$$

$$N_1 \xrightarrow{p_{N_2}} p_{N_3} LCP_{N_1} h_{N_1}$$

$$N_2 \xrightarrow{p_{N_4}} p_{N_5} LCP_{N_2} h_{N_2}$$

$$N_3 \xrightarrow{p_{N_6}} p_{N_7} LCP_{N_3} h_{N_3}$$

$$N_4 \xrightarrow{s_1} s_2 \xrightarrow{LCP_{N_4}} h_{N_4} \xrightarrow{s_3} N_5$$

$$N_5 \xrightarrow{s_4} s_5 \xrightarrow{s_6} LCP_{N_5} h_{N_5}$$

$$N_6 \xrightarrow{s_7} s_8 \xrightarrow{s_9} LCP_{N_6} h_{N_6}$$

$$N_7 \xrightarrow{s_{10}} s_{11} \xrightarrow{s_{12}} LCP_{N_7} h_{N_7}$$

$$R^S = \{s_1, s_2\}$$

$$VO = \{(((s_1, s_2, (s_3, p_{R_1})), ((s_4, p_{R_2}), (s_5, p_{R_1}), (s_6, p_{R_1}))), ((s_7, p_{R_2}), (s_8, p_{R_1}), s_9), (LCP_{N_7}, h_{N_7})), \{R_1, R_2\}\}$$
**E-VS² - VO Verification**

The client checks the soundness and completeness of \( R^S \) by verifying the VO.

- **Soundness violation**
  - \( \exists s \in R^S, \text{ but } s \notin D \)

- **Completeness violation**
  - \( \exists s \in D \text{ s.t. } DST(s, s_q) \leq \theta \)
  - \( \text{ but } s \notin R^S \)

\[
s_q = \text{"Celestine"}
\]
\[
\theta = 4
\]

\[
Sig(T) = \text{sign}(h_{N_1})
\]

[Check if \( Sig(T) \) matches the local copy]

\[
R^S = \{s_1, s_2\}
\]

\[
VO = \{((s_1, s_2), (s_3, p_{R_1})), ((s_4, p_{R_2}), (s_5, p_{R_1}), (s_6, p_{R_1})), ((s_7, p_{R_2}), (s_8, p_{R_1}, s_9), (LCP_{N_7}, h_{N_7})), \{R_1, R_2\}\}
\]
E-VS² - VO Verification

The client checks the soundness and completeness of \( R^S \) by verifying the VO.

\[
\begin{align*}
\text{soundness violation} & \quad \exists s \in R^S, \text{ but } s \notin D \\
\text{completeness violation} & \quad \exists s \in D \text{ s.t. } DST(s, s_q) \leq \theta \text{ but } s \notin R^S
\end{align*}
\]

\[
R^S = \{ s_1, s_2 \}
\]

\[
VO = \{ (((s_1, s_2), (s_3, p_{R_1})), ((s_4, p_{R_2}), (s_5, p_{R_1}), (s_6, p_{R_1}))), ((s_7, p_{R_2}), (s_8, p_{R_1}), s_9), (\text{LCP}_{N_7}, h_{N_7})) \}, \{ R_1, R_2 \}
\]

\[
\begin{align*}
&DST(s_1, s_q) = 4 \\
&DST(s_2, s_q) = 3 < 4
\end{align*}
\]

\[
\begin{align*}
MIN\_DST(\text{LCP}_{N_7}, s_q) = 6 > 4
\end{align*}
\]

\[
\begin{align*}
&\text{for similar strings} \\
&\text{for MF-node} \\
&\text{for DBH-strings} \\
&\text{for FP-string}
\end{align*}
\]

\[
\begin{align*}
&MIN\_DST(\text{LCP}_{N_7}, s_q) = 6 > 4 \\
&\forall \text{MF-node } N, \text{ check if } MIN\_DST(N, \text{LCP}, s_q) > \theta \\
&\forall \text{DBH-string } (s, p_R), \text{ check if } p \in R, \text{ and if } min\_dst(p_q, R) > \theta \\
&\forall \text{FP-string } s, \text{ check if } DST(s_q, s) > \theta \\
&\exists s \in R^S, \text{ check if } DST(s, s_q) \leq \theta \\
&\end{align*}
\]

\[
\text{Naive approach: } 12 \text{ DST calculations}
\]

\[
\text{VS²: } 10 \text{ DST calculations}
\]
### Complexity Analysis

<table>
<thead>
<tr>
<th>Phase</th>
<th>Measurement</th>
<th>VS²</th>
<th>E-VS²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup</td>
<td>Time</td>
<td>$O(n)$</td>
<td>$O(cdn^2)$</td>
</tr>
<tr>
<td></td>
<td>Space</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>VO Construction</td>
<td>Time</td>
<td>$O(n)$</td>
<td>$O(n + n_{DS}^3)$</td>
</tr>
<tr>
<td></td>
<td>VO Size</td>
<td>$(n_R + n_C)σ_S + n_{MF}σ_M$</td>
<td>$(n_R + n_C)σ_S + n_{MF}σ_M + n_{DBH}σ_D$</td>
</tr>
<tr>
<td>VO Verification</td>
<td>Time</td>
<td>$O((n_R + n_{MF} + n_C)C_{Ed})$</td>
<td>$O((n_R + n_{MF} + n_{FP})C_{Ed} + n_{DBH}C_{El})$</td>
</tr>
</tbody>
</table>

(n: # of strings in $D$; c: a constant in [0, 1]; d: # of dimensions of Euclidean space; $σ_S$: the average length of the string; $σ_M$: Avg. size of a MB-tree node; $σ_D$: Avg. size of a DBH; $n_R$: # of strings in $M^S$; $n_C$: # of C-strings; $n_{FP}$: # of FP-strings; $n_{DS}$: # of DBH-strings; $n_{DBH}$: # of DBHs; $n_{MF}$: # of MF nodes; $C_{Ed}$: the complexity of an edit distance computation; $C_{El}$: the complexity of Euclidean distance calculation.)

- **E-VS²** results in higher VO construction complexity at the server side.
- **E-VS²** dramatically saves the VO verification cost at the client side.
Outline

1 Introduction
2 Research Results
   • Authentication of Outsourced Data Deduplication
     • Verification of Similarity Search Approach ($VS^2$)
     • Embedding-based Verification of Similarity Search Approach ($E-VS^2$)
   • Experiments
     • Privacy-preserving Outsourced Data Deduplication
     • Privacy-preserving Outsourced Data Inconsistency Repair
3 Research beyond the Thesis
4 Future Plan
5 Conclusion
Experiments - Setup

- Environment
  - **Language**: C++
  - **Testbed**: A Linux machine with 2.4 GHz CPU and 48 GB RAM

- Datasets
  - **Actors**¹ 260,000 lastnames
  - **Authors**² 1,000,000 full names

- Evaluation metric
  - VO construction time
  - VO verification time

¹ [http://www.imdb.com/interfaces](http://www.imdb.com/interfaces)
² [http://dblp.uni-trier.de/xml/](http://dblp.uni-trier.de/xml/)
Experiments - VO Construction Time

**Time Performance of VO Construction**

- **E-VS²** takes more time at the server side to construct VO, especially when θ is small.

(a) The *Actors* dataset

(b) The *Authors* dataset
Experiments - VO Verification Time

Time Performance of VO Verification

- VS^2 and E-VS^2 are significantly more efficient than the baseline approach in verification cost.
- The advantage of E-VS^2 is large when θ is small.
1. Introduction
2. Research Results
   - Authentication of Outsourced Data Deduplication
     - Verification of Similarity Search Approach ($VS^2$)
     - Embedding-based Verification of Similarity Search Approach ($E-VS^2$)
   - Experiments
   - Privacy-preserving Outsourced Data Deduplication
     - Privacy-preserving Outsourced Data Inconsistency Repair
3. Research beyond the Thesis
4. Future Plan
5. Conclusion
\( \alpha \)-Security against Frequency Analysis (FA) Attack

Define \( \alpha \)-security to limit the success probability of frequency analysis attack.

\[
\text{Experiment } Exp_{A, \Pi}^{FA} ()
\]

\[
p' \leftarrow A_{freq}(e), freq(P)
\]

Return 1 if \( p' = Decrypt(k, e) \)

Return 0 otherwise

\( \alpha \)-security against FA attack if \( Pr[Exp_{A, \Pi}^{FA} () = 1] \leq \alpha \)

---

\(^3\)Boxiang Dong, Ruilin Liu, Wendy Hui Wang. 
Prada: Privacy-preserving Data-Deduplication-as-a-Service. 
International Conference on Information and Knowledge Management, 2014. (Acceptance rate=20%).
We design two approaches to enable data deduplication and defend against the frequency analysis attack.

- **Locality-sensitive Hashing Based Approach (LSHB)**
- **Embedding & Homomorphic Substitution Approach (EHS)**

LSHB approach encodes strings into LSH values that preserve the string similarity; and (2) are of the same frequency groupwise.

EHS approach encodes strings into Euclidean points that preserve the string similarity; and (2) are of uniform frequency.

---

Prada: Privacy-preserving Data-Deduplication-as-a-Service. 
International Conference on Information and Knowledge Management, 2014. (Acceptance rate=20%).
Privacy-preserving Outsourced Data Deduplication

Experiment Results

(a) Time performance

(b) Deduplication accuracy
Outline

1. Introduction
2. Research Results
   - Authentication of Outsourced Data Deduplication
   - Verification of Similarity Search Approach ($VS^2$)
   - Embedding-based Verification of Similarity Search Approach ($E-VS^2$)
   - Experiments
   - Privacy-preserving Outsourced Data Deduplication
   - Privacy-preserving Outsourced Data Inconsistency Repair
3. Research beyond the Thesis
4. Future Plan
5. Conclusion
**Functional Dependency (FD)**

A functional dependency (FD) is defined as $X \rightarrow Y$ if $r_1[X] = r_2[X]$, then $r_1[Y] = r_2[Y]$.

FDs play a key role in identifying and fixing data inconsistency.

<table>
<thead>
<tr>
<th>TID</th>
<th>Conference</th>
<th>Year</th>
<th>Country</th>
<th>Capital</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>SIGMOD</td>
<td>2007</td>
<td>China</td>
<td>Beijing</td>
<td>Beijing</td>
</tr>
<tr>
<td>$r_2$</td>
<td>ICDM</td>
<td>2014</td>
<td>China</td>
<td>Shanghai</td>
<td>Shenzhen</td>
</tr>
<tr>
<td>$r_3$</td>
<td>KDD</td>
<td>2014</td>
<td>U.S.</td>
<td>Washington D.C.</td>
<td>New York City</td>
</tr>
<tr>
<td>$r_4$</td>
<td>KDD</td>
<td>2015</td>
<td>Australia</td>
<td>Canberra</td>
<td>Sydney</td>
</tr>
<tr>
<td>$r_5$</td>
<td>ICDM</td>
<td>2015</td>
<td>U.S.</td>
<td>New York City</td>
<td>Atlantic City</td>
</tr>
</tbody>
</table>

**FD** : Country $\rightarrow$ Capital
Indistinguishability against FD-preserving Chosen Plaintext Attack (IND-FCPA)

**Experiment** \( \text{Exp}^{\text{IND-FCPA}}_{A, \Pi}(\lambda) \)

\[ k \leftarrow \text{KeyGen}(\lambda) \]

\((D_0, D_1) \leftarrow A^{\text{Encrypt}(.)}(k) \text{ s.t. } FD_0 = FD_1 \text{ and } |D_0| = |D_1| \]

\[ b \leftarrow \{0, 1\} \]

\[ b' \leftarrow A^{\text{Encrypt}(.)}(k) \]

Return 1 if \( b = b' \)

Return 0 otherwise

\( \text{IND-FCPA if } Pr[Exp^{\text{IND-FCPA}}_{A}(n) = 1] \leq \frac{1}{2} + \text{negl}(n) \)
We consider two scenarios of the outsourced data inconsistency repair, and design two encryption/encoding approaches to provide robust privacy guarantee.\(^5\)

---

Outline

1 Introduction
2 Research Results
   • Authentication of Outsourced Data Deduplication
     • Verification of Similarity Search Approach ($VS^2$)
     • Embedding-based Verification of Similarity Search Approach ($E-VS^2$)
     • Experiments
   • Privacy-preserving Outsourced Data Deduplication
   • Privacy-preserving Outsourced Data Inconsistency Repair
3 Research beyond the Thesis
4 Future Plan
5 Conclusion
Research beyond the Thesis

- Authentication of outsourced data mining computations
  - Association rule mining [DBSec’13, ICDM’13, TSC’15]
  - Outlier mining (under review)
- Rank aggregation in the crowdsourcing setting (under review)
  - Rank inference
  - Task assignment with data privacy concern
- Data-as-a-commodity (under review)
  - Budget constraint
  - High quality (low inconsistency)
1 Introduction

2 Research Results
   • Authentication of Outsourced Data Deduplication
     • Verification of Similarity Search Approach ($VS^2$)
     • Embedding-based Verification of Similarity Search Approach ($E-VS^2$)
     • Experiments
   • Privacy-preserving Outsourced Data Deduplication
   • Privacy-preserving Outsourced Data Inconsistency Repair

3 Research beyond the Thesis

4 Future Plan

5 Conclusion
Future Plan

- Authenticated outsourced data inconsistency repair
  
  **Challenge**  It is NP-complete to find a repair with the minimum cost.

  **Solution**  
  - Convert the strings into Euclidean space.
  - It is the center of mass that results in the smallest repair cost.

- Authenticated outsourced data imputation
  
  **Challenge**  It demands a similarity matrix between all values.

  **Solution**  Create evidence imputation objects to verify the result in a probabilistic way.
Conclusion

Privacy-preserving and authenticated data cleaning on outsourced databases.

- Define two security notions, namely $\alpha$-security and $IND-FCPA$.
- Authentication of outsourced data deduplication.
- Privacy-preserving outsourced data deduplication.
- Privacy-preserving outsourced data inconsistency repair.
  - Privacy against FD attack.
  - Privacy against frequency analysis attack.

The suit of encryption, encoding, and authentication schemes address the security and privacy concerns in outsourced computing.
## My Publications

<table>
<thead>
<tr>
<th>Conference</th>
<th>Authors</th>
<th>Title</th>
<th>Venue</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICDM'13</td>
<td>Boxiang Dong, Ruilin Liu, Hui (Wendy) Wang</td>
<td>Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee.</td>
<td>IEEE International Conference on Data Mining (ICDM). Dallas, Texas. 2013. (Acceptance rate = 19.7%).</td>
<td></td>
</tr>
<tr>
<td>DBSec'13</td>
<td>Boxiang Dong, Ruilin Liu, Hui (Wendy) Wang</td>
<td>Result Integrity Verification of Outsourced Frequent Itemset Mining.</td>
<td>Annual IFIP WG 11.3 Conference on Data and Application Security and Privacy (DBSec). Newark, NJ. 2013.</td>
<td></td>
</tr>
</tbody>
</table>


Data quality and the bottom line.
The Data Warehouse Institute Report, 2002.

Approximate string joins in a database (almost) for free.

Contractive embedding methods for similarity searching in metric spaces.
Technical report, Computer Science Department, University of Maryland.

Audio: An integrity auditing framework of outlier-mining-as-a-service systems.
In Machine Learning and Knowledge Discovery in Databases, pages 1–18. 2012.

Efficient secure similarity computation on encrypted trajectory data.
In IEEE International Conference on Data Engineering, pages 66–77, 2015.

Functional dependency discovery: An experimental evaluation of seven algorithms.

Pinocchio: Nearly practical verifiable computation.
In IEEE Symposium on Security and Privacy (SP), pages 238–252, 2013.


Thank you!

Questions?