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

Name	Street	City	Phone
John	Leonard	NY	518-457-5181
John	Lenard	NY	518-457-5181
Kevin		LA	213-974-3211
Mike	Main	Phil	518-457-5181

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.

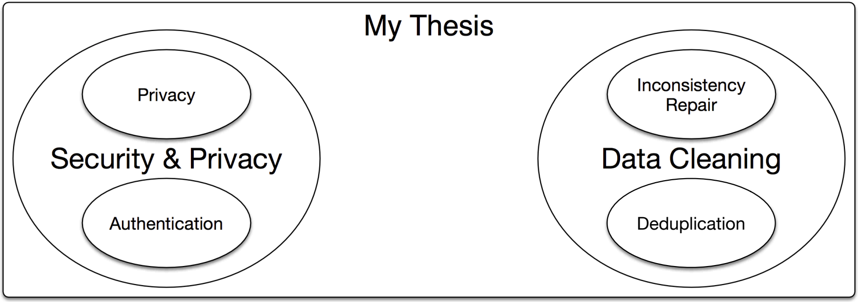
- Software bugs
- Intention to save computational cost

Data privacy The outsourced data may include sensitive personal information.

- Medical information
- Financial record

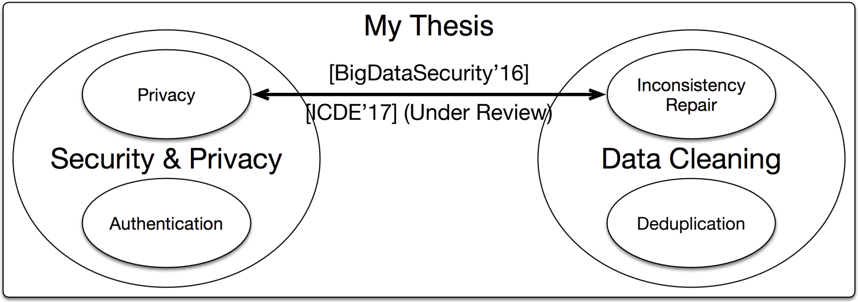
My Thesis

Thesis topic: Privacy-preserving and authenticated data cleaning on outsourced databases



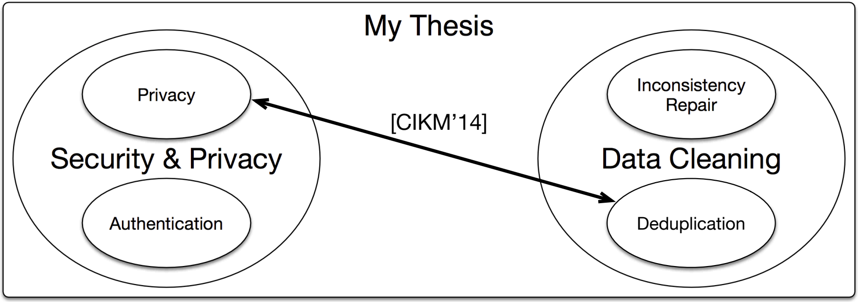
My Thesis

Thesis topic: Privacy-preserving and authenticated data cleaning on outsourced databases



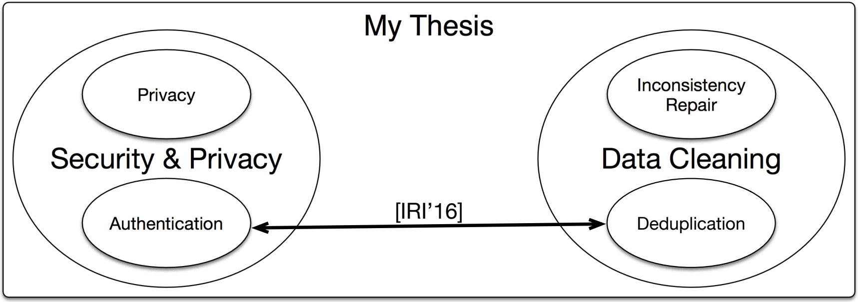
My Thesis

Thesis topic: Privacy-preserving and authenticated data cleaning on outsourced databases



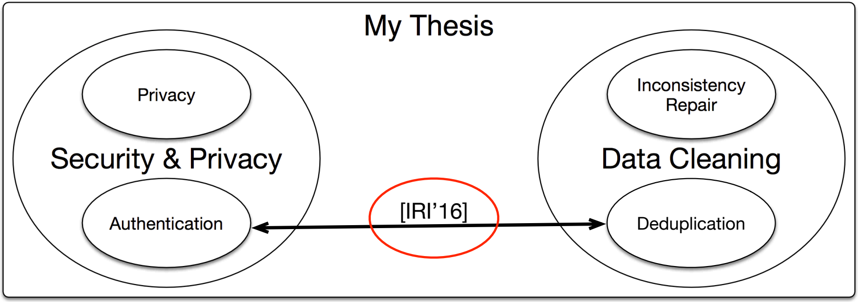
My Thesis

Thesis topic: Privacy-preserving and authenticated data cleaning on outsourced databases



My Thesis

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

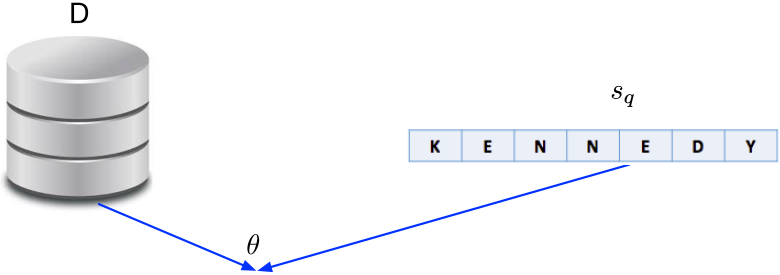
Authentication of Outsourced Data Deduplication

Boxiang Dong, Wendy Hui Wang.
IEEE International Conference on Information Reuse and Integration (IRI), Pittsburgh, PA. July 2016.
(Acceptance rate = 25%)

Data Deduplication

Data deduplication Eliminate near-duplicate copies.

- Record matching: Detect near-duplicate copies.



$$\{s | s \in D, DST(s, s_q) \leq \theta\}$$

θ: similarity threshold
DST: edit distance

Data Deduplication

Data deduplication Eliminate near-duplicate copies.

- Record matching: Detect near-duplicate copies.

RID	Name	Street	City	Age
r_1	John	Leonard	NY	45
r_2	Kevin	Wicks	LA	31
r_3	Mike	Main	Phil	22

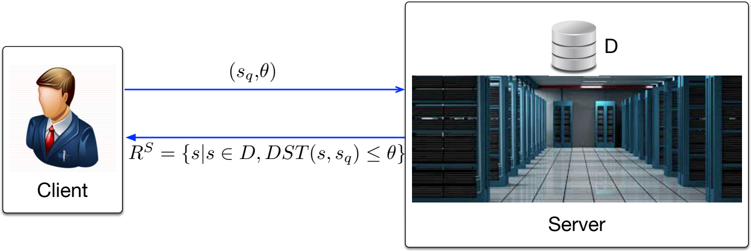
$s_q = $ (John, Lenard, NY, 45)

$\theta = 2$

$\{r_1\}$

Outsourcing Framework

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. $DST(s, s_q) \leq \theta$, $s \in R^S$.

Authentication

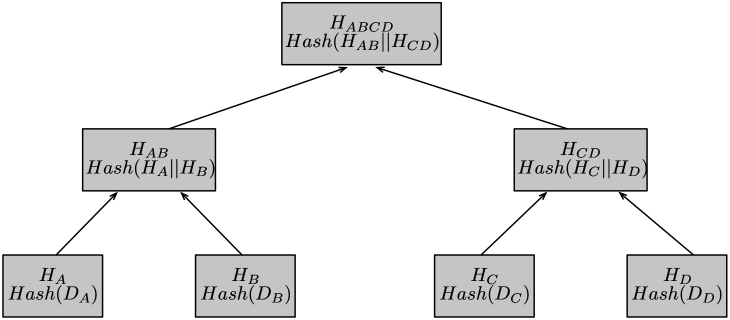
We aim at an authentication framework that satisfies the following objectives.

Authentication Objective

- catches
  - soundness violation
    - $\exists s \in R^S$, but $s \notin D$
    - $\exists s \in R^S$, but $DST(s, s_q) > \theta$
  - completeness violation
    - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$
- supports efficient verification
- scales well with big data

Preliminary - Merkle Tree

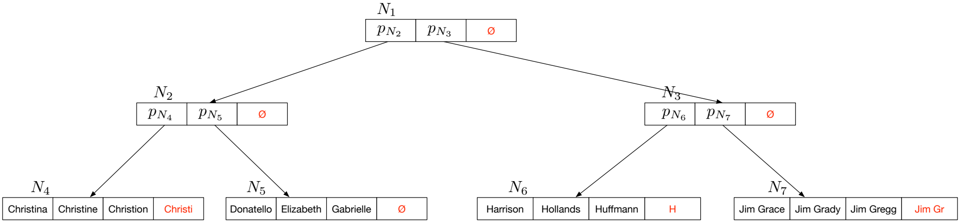
Merkle tree is a generalization of hash lists and hash chains.



- 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

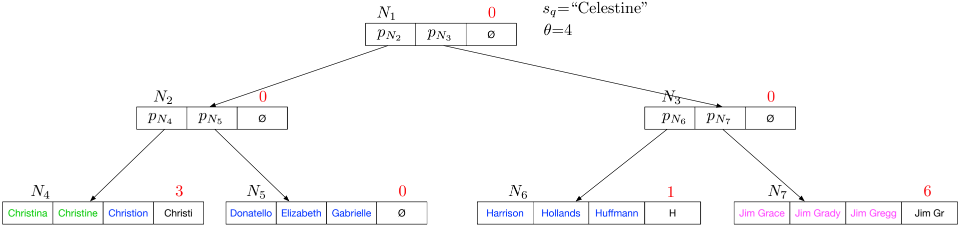
B^{ed}-Tree [ZHOS10] is a string indexing structure.



- Sort the strings in dictionary order.
- Store the longest common prefix (LCP) of the enclosed strings in every node.

Preliminary - B^{ed}-Tree

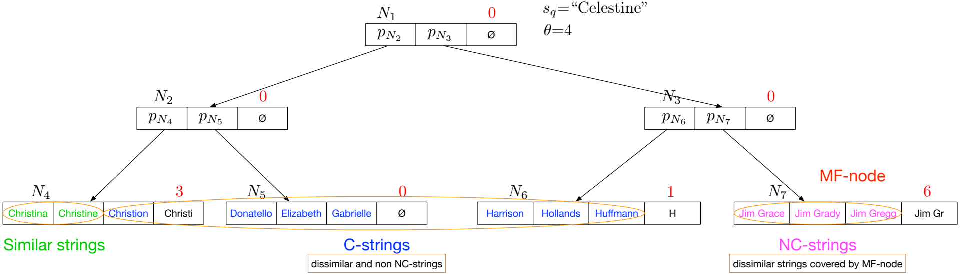
B^{ed}-Tree [ZHOS10] is a string indexing structure.



- $\forall N$, calculate $MIN\_DST(s_q, N.LCP)$.

Preliminary - B^{ed}-Tree

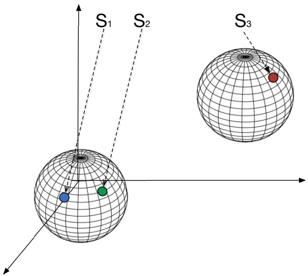
B^{ed}-Tree [ZHOS10] is a string indexing structure.



- 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.

Preliminary - Embedding

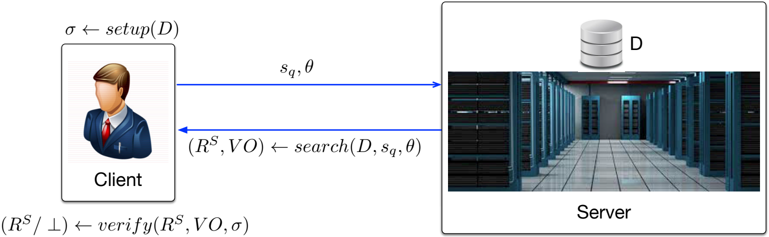
Embedding maps strings into Euclidean points in a similarity-preserving way.



- Euclidean distance calculation is much more efficient than edit distance computing, i.e., $O(dst(p_i, p_j)) << O(DST(s_i, s_j))$.
- *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.



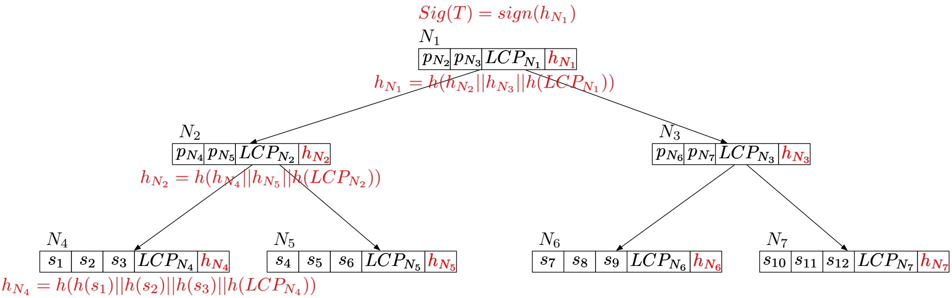
The client is able to efficiently detect any unsound or incomplete result returned by the server by checking the *VO*.

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 ($E\text{-}VS^2$)
        - Experiments
    - Privacy-preserving Outsourced Data Deduplication
    - Privacy-preserving Outsourced Data Inconsistency Repair
3. Research beyond the Thesis
4. Future Plan
5. Conclusion

VS^2 - Setup

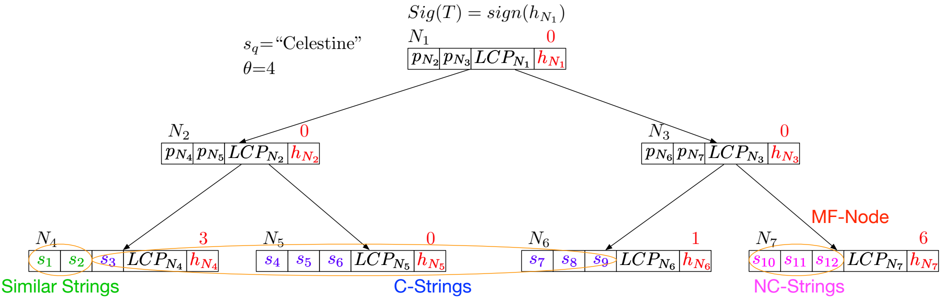
We propose an authenticated string indexing structure, named *MB*-tree (Merkle B^{ed}-tree).



- 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.

VS^2-VO Construction

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



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

$VO = \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7})))\}$

- Include all the C-strings and similar strings in *VO*.
- Substitute the large amount of NC-strings with the MF-nodes.

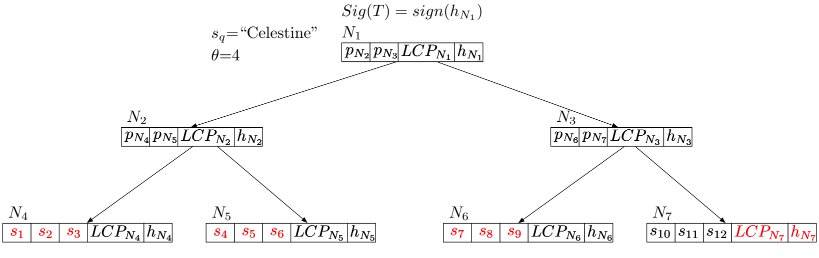
VS^2 - VO Verification

The client checks the soundness of completeness of R^S by verifying the VO.

catches:

- soundness violation
  - $\exists s \in R^S$, but $s \notin D$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$
- completeness violation
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$

$\boxed{\text{Compute } Sig(T) \text{ from } VO}$



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

$VO = \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7})))\}$

VS^2 - VO Verification

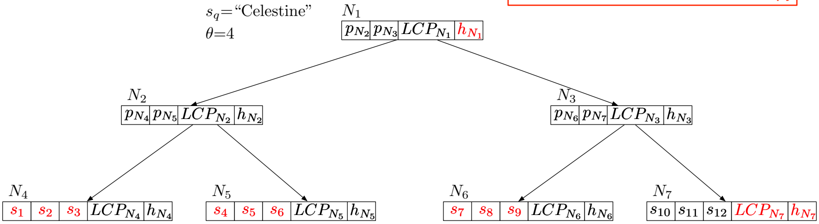
The client checks the soundness and completeness of R^S by verifying the VO.

$$\text{catches} \begin{cases} \text{soundness violation} \begin{cases} \exists s \in R^S, \text{ but } s \notin D \\ \exists s \in R^S, \text{ but } DST(s, s_q) > \theta \end{cases} \\ \text{completeness violation} \quad \exists s \in D \text{ s.t. } DST(s, s_q) \le \theta \text{ but } s \notin R^S \end{cases}$$

$\boxed{\text{Compute } Sig(T) \text{ from } VO}$

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

$\boxed{\text{Check if } Sig(T) \text{ matches the local copy}}$



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

$VO = \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7})))\}$

VS^2 - VO Verification

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

catches
- soundness violation
  - $\exists s \in R^S$, but $s \notin D$ — $\boxed{\text{Compute } Sig(T) \text{ from } VO}$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$ — $\boxed{\forall s \in R^S, \text{ check if } DST(s, s_q) \leq \theta}$
- completeness violation
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$
    - $\boxed{\forall \text{C-string } s, \text{ check if } DST(s, s_q) > \theta}$
    - $\boxed{\forall \text{MF-node } N, \text{ check if } MIN\_DST(N.LCP, s_q) > \theta}$

s_q= " Celestine "

θ=4

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

$VO = \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7})))\}$

for similar strings
$$\begin{cases} DST(s_1, s_q) = 4 \\ DST(s_2, s_q) = 3 < 4 \end{cases}$$

for C-strings
$$\begin{cases} DST(s_3, s_q) = 5 > 4 \\ DST(s_4, s_q) = 9 > 4 \\ DST(s_5, s_q) = 9 > 4 \\ DST(s_6, s_q) = 8 > 4 \\ DST(s_7, s_q) = 8 > 4 \\ DST(s_8, s_q) = 8 > 4 \\ DST(s_9, s_q) = 8 > 4 \end{cases}$$

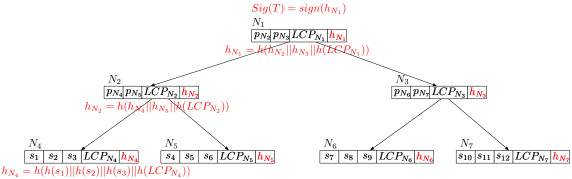
for MF-node
$$MIN\_DST(LCP_{N_7}, s_q) = 6 > 4$$

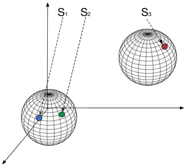
10 DST calculations

Outline

E-VS^2 - 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 $dst(p, p_q) > \theta$, it must be true that $DST(s, s_q) > \theta$.

E-VS^2 - VO Construction

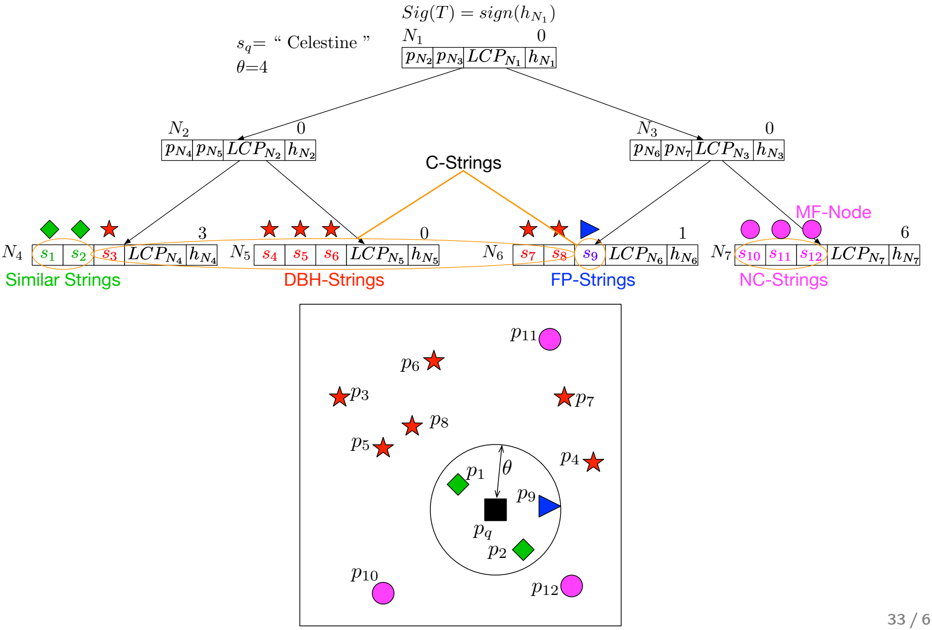
Distant Bounding Hyper-rectangle (DBH) A hyper-rectangle R in the Euclidean space is a DBH if $min\_dst(p_q, R) > \theta$.

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

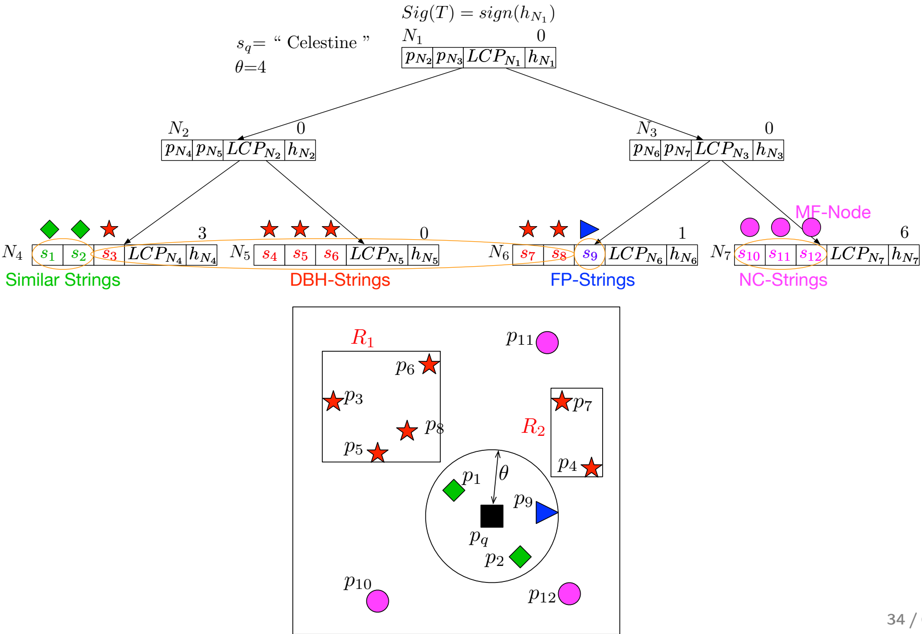
FP-String For any C-string s, if $dst(p, p_q) \leq \theta$, we call it a FP-string.

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



E-VS^2 - VO Construction



E-VS^2 - VO Construction

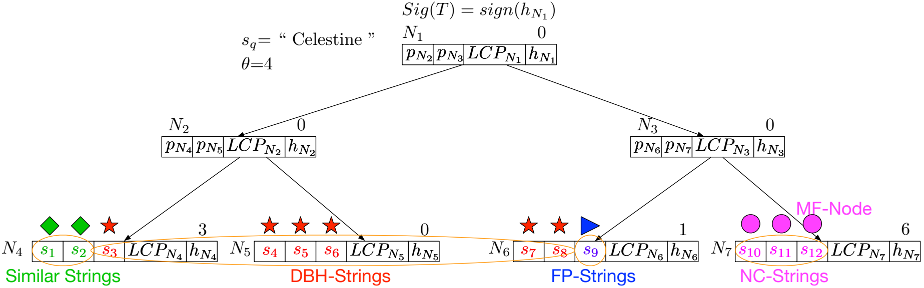
Theorem (NP-Completeness of DBH Construction)

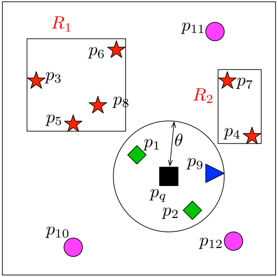
Given a query string s_q, and a set of DBH-strings $\{s_1, \ldots, s_t\}$, let $\{p_1, \ldots, p_t\}$ be their Euclidean points. It is a NP-complete problem to construct a mimimum number of rectangles $\mathcal{R} = \{R_1, \ldots, R_k\}$ s.t.
(1) $\forall i \neq j$, R_i and R_j do not overlap; and
(2) $\forall p_i$, 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 *DBH*s.
- The complexity is cubic to the number of DBH-strings.

E-VS^2 - VO Construction

The server includes the *DBHs* in the *VO*.





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

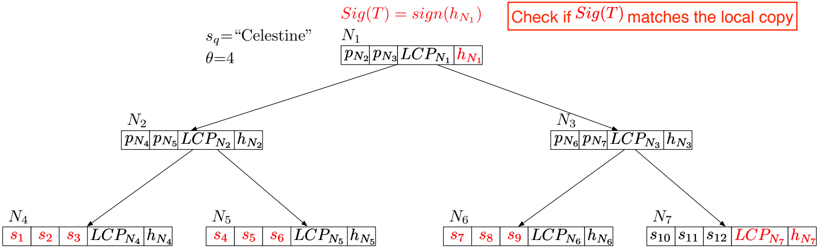
$VO = \{(((s_1, s_2, (s_3, p_{R_1})), ((s_4, p_{R_2}), (s_5, p_{R_1}), (s_6, p_{R_1}))),$
$(((s_7, p_{R_2}), (s_8, p_{R_1}), s_9), (LCP_{N_7}, h_{N_7}))), \{R_1, R_2\}\}$

E-VS^2 - VO Verification

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

$\boxed{\text{Compute } Sig(T) \text{ from } VO}$

$$\text{catches} \begin{cases} \text{soundness violation} \begin{cases} \exists s \in R^S, \text{ but } s \notin D \\ \exists s \in R^S, \text{ but } DST(s, s_q) > \theta \end{cases} \\ \text{completeness violation} \quad \exists s \in D \text{ s.t. } DST(s, s_q) \le \theta \text{ but } s \notin R^S \end{cases}$$



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

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

E-VS^2 - VO Verification

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

catches

- soundness violation
  - $\exists s \in R^S$, but $s \notin D$ — $\boxed{\text{Compute } Sig(T) \text{ from } VO}$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$ — $\boxed{\forall s \in R^S, \text{ check if } DST(s, s_q) \le \theta}$
- completeness violation
  - $\exists s \in D$ s.t. $DST(s, s_q) \le \theta$ but $s \notin R^S$
    - $\boxed{\forall \text{MF-node } N, \text{ check if } MIN\_DST(N.LCP, s_q) > \theta}$
    - $\boxed{\forall \text{DBH-string } (s, p_R), \text{ check if } p \in R, \text{ and if } min\_dst(p_q, R) > \theta}$
    - $\boxed{\forall \text{FP-string } s, \text{ check if } DST(s_q, s) > \theta}$

s_q= " Celestine "

θ=4

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

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

for similar strings $\begin{cases} DST(s_1, s_q) = 4 \\ DST(s_2, s_q) = 3 < 4 \end{cases}$

for MF-node $MIN\_DST(LCP_{N_7}, s_q) = 6 > 4$

for DBH-strings $\begin{cases} min\_dst(p_q, R_1) > \theta \\ min\_dst(p_q, R_2) > \theta \end{cases}$

for FP-string $DST(s_9, s_q) = 8 > 4$

4 DST calculations + 2 dst calculations

Naive approach: 12 DST calculations

VS^2: 10 DST calculations

Complexity Analysis

Phase	Measurement	VS^2	$E\text{-}VS^2$
Setup	Time	$O(n)$	$O(cdn^2)$
	Space	$O(n)$	$O(n)$
VO Construction	Time	$O(n)$	$O(n + n_{DS}^3)$
	VO Size	$(n_R + n_C)\sigma_S + n_{MF}\sigma_M$	$(n_R + n_C)\sigma_S + n_{MF}\sigma_M + n_{DBH}\sigma_D$
VO Verification	Time	$O((n_R + n_{MF} + n_C)C_{Ed})$	$O((n_R + n_{MF} + n_{FP})C_{Ed} + n_{DBH}C_{El})$

(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.)

- $E\text{-}VS^2$ results in higher VO construction complexity at the server side.
- $E\text{-}VS^2$ dramatically saves the VO verification cost at the client side.

Outline

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

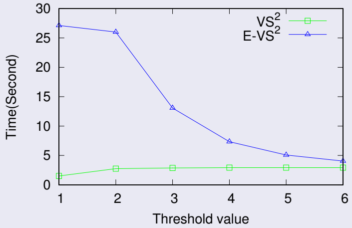
- Evaluation metric
  - VO construction time
  - VO verification time

[1] http://www.imdb.com/interfaces

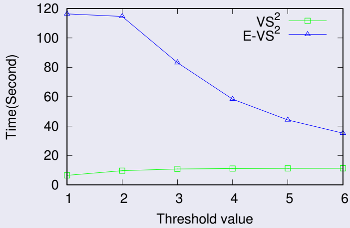
[2] http://dblp.uni-trier.de/xml/

Experiments - VO Construction Time

Time Performance of VO Construction



(a) The *Actors* dataset

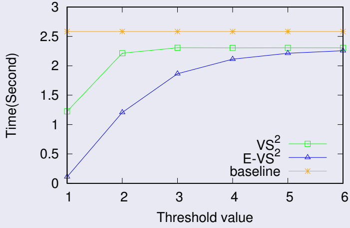


(b) The *Authors* dataset

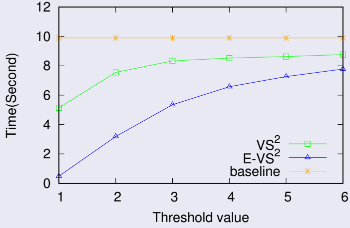
- $E\text{-}VS^2$ takes more time at the server side to construct VO, especially when θ is small.

Experiments - VO Verification Time

Time Performance of VO Verification



(a) The *Actors* dataset ($f = 1,000$)



(b) The *Authors* dataset ($f = 1,000$)

- VS^2 and $E\text{-}VS^2$ are significantly more efficient than the baseline approach in verification cost.
- The advantage of $E\text{-}VS^2$ is large when θ is small.

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 ($E\text{-}VS^2$)
     - Experiments
   - **Privacy-preserving Outsourced Data Deduplication**
   - Privacy-preserving Outsourced Data Inconsistency Repair
3. Research beyond the Thesis
4. Future Plan
5. Conclusion

α-Security against Frequency Analysis (FA) Attack [3]

Define α-security to limit the success probability of frequency analysis attack.

$$\textbf{Experiment } Exp_{A,\Pi}^{FA}()$$
$$p' \leftarrow A^{freq_\epsilon(e), freq(\mathcal{P})}$$
$$\text{Return 1 if } p' = Decrypt(k, e)$$
$$\text{Return 0 otherwise}$$

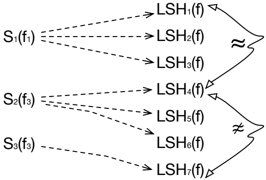
$$\alpha\text{-security against FA attack if } Pr[Exp_{A,\Pi}^{FA}() = 1] \leq \alpha$$

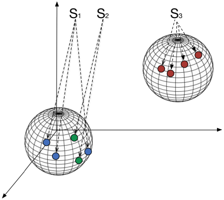
[3] Boxiang Dong, Ruilin Liu, Wendy Hui Wang. *Prada: Privacy-preserving Data-Deduplication-as-a-Service.* International Conference on Information and Knowledge Management, 2014. (Acceptance rate=20%).

Privacy-preserving Outsourced Data Deduplication [4]

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

(1) preserve the string similarity; and

(2) are of the same frequency groupwise.

EHS approach encodes strings into Euclidean points that

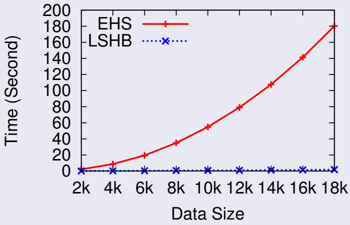
(1) preserve the string similarity; and

(2) are of uniform frequency.

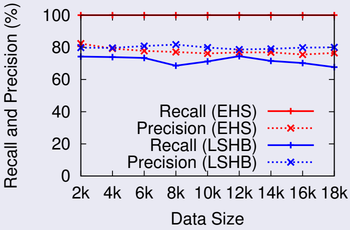
[4] Boxiang Dong, Ruilin Liu, Wendy Hui Wang.
Prada: Privacy-preserving Data-Deduplication-as-a-Service.
International Conference on Information and Knowledge Management, 2014. (Acceptance rate=20%).

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 ($E\text{-}VS^2$)**
        - **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)

Functional dependency (FD) $X \rightarrow Y$ 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.

TID	Conference	Year	Country	Capital	City
r_1	SIGMOD	2007	China	Beijing	Beijing
r_2	ICDM	2014	China	Shanghai	Shenzhen
r_3	KDD	2014	U.S.	Washington D.C.	New York City
r_4	KDD	2015	Australia	Canberra	Sydney
r_5	ICDM	2015	U.S.	New York City	Atlantic City

$FD : Country \rightarrow Capital$

Indistinguishability against FD-preserving Chosen Plaintext Attack (IND-FCPA)

Experiment $Exp_{A,\Pi}^{IND-FCPA}(\lambda)$

$k \leftarrow KeyGen(\lambda)$

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

$b \xleftarrow{\$} \{0, 1\}$

$b' \leftarrow A^{O_{Encrypt(.)}}(k)$

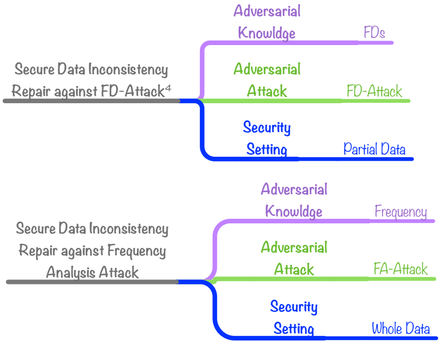
Return 1 if $b = b'$

Return 0 otherwise

$IND - FCPA$ if $Pr[Exp_A^{IND-FCPA}(n) = 1] \leq \frac{1}{2} + negl(n)$

Privacy-preserving Outsourced Data Inconsistency Repair

We consider two scenarios of the outsourced data inconsistency repair, and design two encryption/encoding approaches to provide robust privacy guarantee [5].



[5] Boxiang Dong, Wendy Hui Wang, Jie Yang. *Secure Data Outsourcing with Adversarial Data Dependency Constraints.* International Conference on Big Data Security on Cloud, 2016. (Acceptance rate=23%).

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 ($E\text{-}VS^2$)**
        - **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)

Outline

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 α-*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

IRI'16 Boxiang Dong, Hui (Wendy) Wang.
ARM: Authenticated Approximate Record Matching for Outsourced Databases.
IEEE International Conference on Information Reuse and Integration (IRI). Pittsburgh, PA. 2016. (Acceptance rate = 25%).

BigDataSecurity'16 Boxiang Dong, Hui (Wendy) Wang, Jie Yang.
Secure Data Outsourcing with Adversarial Data Dependency Constraints.
IEEE International Conference on Big Data Security on Cloud (BigDataSecurity). New York. 2016. (Acceptance rate = 23%).

TSC'15 Boxiang Dong, Ruilin Liu, Hui (Wendy) Wang.
Trust-but-Verify: Verifying Result Correctness of Outsourced Frequent Itemset Mining.
IEEE Transactions on Services Computing. 2015.

CIKM'14 Boxiang Dong, Ruilin Liu, Hui (Wendy) Wang.
Prada: Privacy-preserving Data-Deduplication-as-a-Service.
ACM International Conference on Information and Knowledge Management (CIKM). Shanghai, China. 2014. (Acceptance rate = 20%).

ICDM'13 Boxiang Dong, Ruilin Liu, Hui (Wendy) Wang.
Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee.
IEEE International Conference on Data Mining (ICDM). Dallas, Texas. 2013. (Acceptance rate = 19.7%).

DBSec'13 Boxiang Dong, Ruilin Liu, Hui (Wendy) Wang.
Result Integrity Verification of Outsourced Frequent Itemset Mining.
Annual IFIP WG 11.3 Conference on Data and Application Security and Privacy (DBSec). Newark, NJ. 2013.

IJIPM'10 Weifeng Sun, Juanyun Wang, Boxiang Dong, Mingchu Li, Zhenquan Qin. *A Mediated RSA-based End Entity Certificates Revocation Mechanism with Secure Concern in Grid.* International Journal of Information Processing and Management (IJIPM). 2010.

IIH-MSP'10 Weifeng Sun, Boxiang Dong, Zhenquan Qin, Juanyun Wang, Mingchu Li. *A Low-Level Security Solving Method in Grid.* International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP). Darmstadt, Germany. 2010.

References I

[AHMP15] Tristan Allard, Georges Hébrail, Florent Masseglia, and Esther Pacitti.
Chiaroscuro: Transparency and privacy for massive personal time-series clustering.
In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 779–794, 2015.

[BFFR05] Philip Bohannon, Wenfei Fan, Michael Flaster, and Rajeev Rastogi.
A cost-based model and effective heuristic for repairing constraints by value modification.
In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 143–154, 2005.

[BFG+07] Philip Bohannon, Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis.
Conditional functional dependencies for data cleaning.
In *IEEE International Conference on Data Engineering*, pages 746–755, 2007.

[CC04] Tim Churches and Peter Christen.
Some methods for blindfolded record linkage.
BMC Medical Informatics and Decision Making, 4(1):9, 2004.

[CMF+11] Rui Chen, Noman Mohammed, Benjamin CM Fung, Bipin C Desai, and Li Xiong.
Publishing set-valued data via differential privacy.
Proceedings of the VLDB Endowment, 4(11):1087–1098, 2011.

[DLW13] Boxiang Dong, Ruilin Liu, and Hui Wendy Wang.
Result integrity verification of outsourced frequent itemset mining.
In *Data and Applications Security and Privacy XXVII*, pages 258–265. 2013.

[EAMY+13] Durham E. Ashley, Kantarcioglu M., Xue Y., Kuzu M., and Malin Bradley.
Composite bloom filters for secure record linkage.
In *IEEE Transactions on Knowledge and Data Engineering*, 2013.

References II

[Eck02] Wayne W Eckerson.
Data quality and the bottom line.
The Data Warehouse Institute Report, 2002.

[GIJ+01] Luis Gravano, Panagiotis G Ipeirotis, Hosagrahar Visvesvaraya Jagadish, Nick Koudas, Shanmugauelayut Muthukrishnan, Divesh Srivastava, et al.
Approximate string joins in a database (almost) for free.
In *Proceedings of the International Conference on Very Large Data Bases*, volume 1, pages 491–500, 2001.

[HS] G Hjaltason and H Samet.
Contractive embedding methods for similarity searching in metric spaces.
Technical report, Computer Science Department, University of Maryland.

[LWM+12] Ruilin Liu, Hui Wendy Wang, Anna Monreale, Dino Pedreschi, Fosca Giannotti, and Wenge Guo.
Audio: An integrity auditing framework of outlier-mining-as-a-service systems.
In *Machine Learning and Knowledge Discovery in Databases*, pages 1–18. 2012.

[LZL+15] An Liu, Kai Zhengy, Lu Liz, Guanfeng Liu, Lei Zhao, and Xiaofang Zhou.
Efficient secure similarity computation on encrypted trajectory data.
In *IEEE International Conference on Data Engineering*, pages 66–77, 2015.

[PEM+15] Thorsten Papenbrock, Jens Ehrlich, Jannik Marten, Tommy Neubert, Jan-Peer Rudolph, Martin Schönberg, Jakob Zwiener, and Felix Naumann.
Functional dependency discovery: An experimental evaluation of seven algorithms.
Proceedings of the VLDB Endowment, 8(10):1082–1093, 2015.

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova.
Pinocchio: Nearly practical verifiable computation.
In *IEEE Symposium on Security and Privacy (SP)*, pages 238–252, 2013.

References III

[PRZB12] Raluca Ada Popa, Catherine Redfield, Nickolai Zeldovich, and Hari Balakrishnan.
Cryptdb: Processing queries on an encrypted database.
Communications of the ACM, 55(9):103–111, 2012.

[SAA10] Yasin N Silva, Walid G Aref, and Mohamed H Ali.
The similarity join database operator.
In *IEEE International Conference on Data Engineering*, volume 10, pages 892–903, 2010.

[SV10] Nigel P Smart and Frederik Vercauteren.
Fully homomorphic encryption with relatively small key and ciphertext sizes.
In *Public Key Cryptography–PKC*, pages 420–443. 2010.

[SVP+12] Srinath Setty, Victor Vu, Nikhil Panpalia, Benjamin Braun, Andrew J Blumberg, and Michael Walfish.
Taking proof-based verified computation a few steps closer to practicality.
In *The USENIX Security Symposium*, pages 253–268, 2012.

[TOEY11] Nilothpal Talukder, Mourad Ouzzani, Ahmed K Elmagarmid, and Mohamed Yakout.
Detecting inconsistencies in private data with secure function evaluation.
Technical report, Computer Science Department, Purdue University, 2011.

[YLKG07] Su Yan, Dongwon Lee, Min-Yen Kan, and Lee C Giles.
Adaptive sorted neighborhood methods for efficient record linkage.
In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 185–194, 2007.

[ZHOS10] Zhenjie Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, and Divesh Srivastava.
Bed-tree: an all-purpose index structure for string similarity search based on edit distance.
In *Proceedings of the International Conference on Management of Data*, 2010.

Q & A

Thank you!

Questions?