Fast Searchable Encryption with Tunable Locality

Ioannis Demertzis, Charalampos Papamanthou (University of Maryland)

Problem: Privacy Preserving Querying via Searchable Encryption.

Our Searchable Encryption scheme has:

1. Formal proofs based on CRYPTO security definitions
2. Improved Efficiency
   a. Up to \textbf{580x} for external disk
   b. Up to \textbf{12x} in-memory
3. Different trade-offs tuning
   a. Space
   b. False Positives
   c. Locality
   d. Parallelism
   e. Communication overhead
Cryptanalysis of Comparable Encryption in SIGMOD’16
Caleb Horst (UW Tacoma) & Ryo Kikuchi, Keita Xagawa (NTT Secure Platform Laboratories)

Problem: Can a cloud break comparable encryption in [Karras et al. SIGMOD’16]?

A token allows division

!!! Sortable w/ two tokens

!!! Decryptable w/ two plaintexts
Problem:
- Understanding and comparing existing blockchain systems, for data processing workloads

Challenges:
- Vast design space, many platforms, lack of data processing workloads

BLOCKBENCH:
- 4 layers of abstraction, extensible framework, with macro- and micro-benchmark workloads
- Used to analyze Ethereum, Hyperledger Fabric and Parity
Living in Parallel Realities —

Co-Existing Schema Versions with a Bidirectional Database Evolution Language

Kai Herrmann, Hannes Voigt, Andreas Behrend, Jonas Rausch, Wolfgang Lehner (TU Dresden)
Synthesizing Mapping Relationship Using Table Corpus

Yue Wang (U. Massachusetts Amherst); Yeye He (Microsoft Research)

**Input:** 100M+ web tables

**Output:** Synthesized Mappings

---

**Application 1: Auto-Join**

<table>
<thead>
<tr>
<th>Company</th>
<th># Employer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Corp.</td>
<td>...</td>
</tr>
<tr>
<td>Walmart</td>
<td>...</td>
</tr>
<tr>
<td>Oracle</td>
<td>...</td>
</tr>
<tr>
<td>General Electric</td>
<td>...</td>
</tr>
<tr>
<td>AT&amp;T Inc.</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

---

**Application 2: Auto-Correction**

<table>
<thead>
<tr>
<th>Ticker</th>
<th>Market Cap</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSFT</td>
<td>...</td>
</tr>
<tr>
<td>WMT</td>
<td>...</td>
</tr>
<tr>
<td>GE</td>
<td>...</td>
</tr>
<tr>
<td>ORCL</td>
<td>...</td>
</tr>
<tr>
<td>UPS</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

---

**Cleaning**

Why Synthesis?
- Better coverage, e.g., synonyms.
- Easy to curate...
Waldo: An Adaptive Human Interface for Crowd Entity Resolution

Vasilis Verroios, Hector Garcia-Molina (Stanford)
& Yannis Papakonstantinou (UC San Diego)

Entity Resolution

Computer Algorithms

Pr(Same Entity) 20%

Human Tasks

Pairwise

Multi-Item

Same Entity?
○ YES
○ NO
Interactive graph serving

- **Social networks**
  - FB, Twitter, LinkedIn
- **Graphs are huge**
  - E.g., FB: ~billion nodes, ~trillion edges, rich attributes → 1.5 PB of data
- **Graph queries: complex**
  - Exhibit little or no locality
  - E.g., “Friends of my friends in Chicago”
- **Interactivity requirements**
  - Low latency, high throughput

ZipG, a memory-efficient graph store

- Executes queries directly on compressed graph representation
  - No decompressions or scans
- **Rich functionality**
  - Queries from several industrial workloads; Regular path queries & graph traversals
- **New log-structured graph storage**
  - Efficiency for both read & write queries
All-in-One: Graph Processing in RDBMSs Revisited

Kangfei Zhao & Jeffrey Xu Yu (CUHK)

Graph Analytics
- PageRank
- Shortest Distance
- Weakly Connected Component
- Hyperlink-Induced Topic Search
- Label Propagation
- Topological Sort, etc.

Our Enhanced Recursive Query

SQL-99 recursion

Monotonic RA
\[ \sigma, \pi, \times \]

Stratified Program
Least Fixed Point

Non-monotonic RA
- MV-Join
- MV-Join
- Anti-Join
- Union-by-Update

XY-Stratified Program
Iterative Fixed Point

Generated SQL/PSM
Computing A Near-Maximum Independent Set in Linear Time by Reducing-Peeling

*Lijun Chang* (UNSW Sydney), Wei Li, Wenjie Zhang

- Objective: compute large independent set for large graphs in a time-efficient (Subquadratic or more desirable linear to $m$) and space-effective ($2m + O(n)$) manner
  - $m$ is the number of *undirected* edges
Utility-Aware Ridesharing on Road Networks

Peng Cheng, Hao Xin, Lei Chen (HKUST)

Design the schedules for the vehicles to maximize the overall satisfaction of riders under the constraints:

- the deadlines of the riders
- the capacity of the vehicles

Riders’ Satisfaction:

- Vehicle (Driver)-Related Utility
- Rider-Related Utility
- Trajectory-Related Utility
Distance Oracle on Terrain Surface

Victor Junqiu Wei (HKUST), Raymond Chi-Wing Wong (HKUST), Cheng Long (Queen’s University Belfast), David M. Mount (U of Maryland)

Problem

Given two POIs s and t on the terrain surface, estimate the geodesic distance between s and t.

Existing Method

- Computing Geodesic Distance On-The-Fly
  - Very Large Query Time
- Distance Oracle
  - \( \varepsilon \)-approximate (\( \varepsilon \) is a user-specified parameter)
  - Introduces a large amount of Steiner points/edges
  - Large Space and Building Time

Contributions

- We proposed a Distance Oracle, SE.
- Accuracy Guarantee: \( \varepsilon \)-approximate (\( \varepsilon \) is a user-specified parameter)
- Significantly outperforms State-of-the-Art

Building Time: 1-2 orders of magnitude smaller
Oracle Size: 1-3 orders of magnitude smaller
Query Time: 2-3 orders of magnitude smaller
With the same error guarantee \( \varepsilon \)
Efficient Computation of Top-k Frequent Terms over Spatio-Temporal Ranges

Pritom Ahmed, Mahbub Hasan, Abhijith Kashyap, Vagelis Hristidis and Vassilis J Tsotras (UC Riverside)

- **kFST Problem**: Given a spatio-temporal region $R_Q$, find the most frequent terms among the social posts in $R_Q$.
- **Setting**: No predefined region borders, large disk resident data, exact answers
- **Obvious solution**: Use R-tree
- **Our solution**:
  - STL-enhanced indexing and top-k algorithms
  - Theoretical model to optimize STL space requirements
  - Space versus query trade-offs
  - Various indexing options from no STLs to full and/or partial STLs
iceberg query, noun, SQL.
aggregate query with arbitrary, complex joins and having clause

1. Formulate existing problems as iceberg queries
   - Market basket analysis
   - Skyline
2. Completely new framework for combining complementary techniques from existing problems
3. Formal conditions for applicability of techniques
4. Implementation in Postgres
The Dynamic Yannakakis Algorithm: Compact and Efficient Query Processing Under Updates

Muhammad Idris, Stijn Vansummeren and Martín Ugarte

1. Dynamic Query Evaluation
   How to quickly react under database updates
   - Desiderata:
     - In-memory data structure
     - Constant-delay enumeration of results
     - Space linear in the size of the database
     - Efficiently adapt under updates

2. Incremental View Maintenance
   - Keep (sub) results materialized
   - Only change what is necessary

3. Can we avoid the tradeoff?

4. Dynamic Yannakakis
   - A practical algorithm
   - Match two theoretical lower bounds
   - Opt & MainMem
Revisiting Reuse in Main Memory Database Systems

Kayhan Dursun, Carsten Binnig, Ugur Cetintemel, Tim Kraska (Brown)

How reuse is done today?

EXPENSIVE MATERIALIZATION COSTS

THESE MAY NOT PAY OFF IN THE FUTURE

IF YOU WOULD LIKE TO SEE HOW WE GET REUSE FOR FREE, PLEASE COME AND SEE MY TALK 😊
Teaser Talks (Second Part)
Pufferfish Privacy Mechanisms for Correlated Data
Shuang Song, Yizhen Wang, Kamalika Chaudhuri (UCSD)

Sensitive Data with Correlation

Our Contribution

- a general privacy-preserving mechanism for any Pufferfish privacy framework – the Wasserstein Mechanism
- a mechanism when the correlation is described by a Bayesian network – the Markov Quilt Mechanism
- an efficient implementation when the correlation is described by a Markov chain
- experiments on real data-sets

Challenge:
DP does not hide sensitive information about individual records in the presence of correlations.
Better differentially private Stochastic Gradient Descent (SGD).

- SGD is a popular optimization algorithm for machine learning.
- Differential privacy is the de facto standard for formalizing privacy.

Improve private SGD on the following aspects simultaneously:

- Easier to implement: “Bolt on” with an existing implementation.
- Run faster,
- Better convergence/accuracy and
- Support a stronger privacy model.

Essence behind the “all-win” improvements: A novel analysis of the L2-sensitivity of SGD.
Pythia: Data Dependent Differentially Private Algorithm Selection

Ios Kotsogiannis, Ashwin Machanavajjhala, Gerome Miklau, Michael Hay

Algorithm Selection...
- Private evaluation of task $T$
- Algorithms $A_T$ suitable for $T$
- Choose $A^* \in A_T$ to answer $T$

...Without Data Access
- No clear winner in $A_T$ for all instances of $T$
- Running all algorithms **violates privacy**

Pythia
- End-to-end privacy
- Chooses the right algorithm
Utility Cost of Formal Privacy for Releasing National Employer-Employee Statistics

S Haney, A Machanavajjhala, J Abowd, M Graham, M Kutzbach, L Vilhuber

US Law: Title 13 Section 9

≈

Pufferfish Privacy Requirements

DP-like Privacy Definition

Noisy Employer Statistics

Comparable or lower error than current non-private methods
Online Deduplication for Databases

Lianghong Xu (CMU); Andy Pavlo (CMU);
Sudipta Sengupta (Microsoft Research); Gregory Ganger (CMU)
QFix: Diagnosing errors through query histories

Xiaolan Wang, Alexandra Meliou (U. Massachusetts Amherst) & Eugene Wu (Columbia U.)

QFix: Fixing bad queries for dynamic DBMS

Find & fix errors in query histories.

Traditional Data Cleaning

Find & fix errors directly on current db

Queries Change Database

Static Database
UGuide – User-Guided Discovery of FD-Detectable Errors
S. Thirumuruganathan, L. Berti-Equille, M. Ouzzani, J. Quiane-Ruiz, N. Tang (HBKU)

Ideally!

Reality!

What you really need!

Pick best question

Dirty Data

Mine AFDs

Candidate FDs and violations

Update Candidate set
SLiMFast: Guaranteed Results for Data Fusion and Source Reliability

Theo Rekatsinas; Manas Joglekar; Hector Garcia-Molina; Aditya Parameswaran; Christopher Ré

Problem: Clean inaccurate, conflicting data and find hoax sources!

SLiMFast: New ML data fusion framework; subsumes and generalizes most existing models; theoretical guarantees on the quality of its output.

Use features to describe sources and fix inaccurate data twice more accurately!

In most cases, Logistic Regression is enough to solve data fusion!
Crowdsourced Top-k Queries by Confidence-Aware Pairwise Judgments
Ngai Meng KOU¹, Yan LI¹, Hao WANG², Leong Hou U¹, Zhiguo GONG¹
¹University of Macau, ²Nanjing University

Problem: find the top-k items from a set of \textit{computationally challenging} items.
UI of microtask: pairwise comparison.

What’s new?

Previous work: the budget for every pair is constant and the query processing is not confidence-aware.
Ours: the budget for a pair is dynamically decided by the hardness with confidence.

Then, how to design a method that optimizes cost and latency with quality guarantee?
Falcon: Scaling Up Hands-Off Crowdsourced Entity Matching to Build Cloud Services
Sanjib Das*, Paul Suganthan G. C.*, AnHai Doan*, Jeff Naughton*, Ganesh Krishnan+,
Esteban Arcaute*, Rohit Deep*, Vijay Raghavendra*, Youngchoon Park++,
*University of Wisconsin-Madison, *WalmartLabs, ++Johnson Controls

**Table A**

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dave Smith</td>
<td>Madison</td>
<td>WI</td>
</tr>
<tr>
<td>Joe Wilson</td>
<td>San Jose</td>
<td>CA</td>
</tr>
<tr>
<td>Dan Smith</td>
<td>Middleton</td>
<td>WI</td>
</tr>
</tbody>
</table>

**Table B**

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>David D. Smith</td>
<td>Madison</td>
<td>WI</td>
</tr>
<tr>
<td>Daniel W. Smith</td>
<td>Middleton</td>
<td>WI</td>
</tr>
</tbody>
</table>

**Challenge**: Scale up EM workflow
- DAG involving rules, ML, crowdsourcing
- Use crowd time to mask machine time

**Domain scientists @ UW-Madison**

**Results**
Matches tables of 1M - 2.5M tuples, $54-66, 2-14 hours
Deployed as a cloud service at CloudMatcher.io
Used extensively at several organizations
  - e.g., UW Depts., Johnson Controls, WalmartLabs, etc.

**Talk@Session 28, Buckingham (Thur 14:00-15:40)**
CrowdDQS: Dynamic Question Selection in Crowdsourcing Systems

Asif R. Khan & Hector Garcia-Molina (Stanford)

- CrowdDQS dynamically chooses questions in real time
- Automatically learns worker accuracies and blocks spammers
- Deployed to 1000s of workers on AMT
- Can reduce costs up to 6x
CDB: A Crowd-Powered Database System

Guoliang Li and others (Tsinghua U.)

Graph-based Tuple-level Optimization Model
- Tree Model: 15 questions
- Graph Model: 3 questions

Multi-Goal Optimization (Cost, Quality, Latency)
Scaling Locally Linear Embedding

Yasuhiro Fujiwara and others (NTT Communication Science Laboratories)

LLE reduces the dimensionality of dataset

Step 1: k-NN graph
Step 2: Edge weight by regression
Step 3: Eigen decomposition of \((I-W)^T(I-W)\)

We reduce the computation cost as follows:

1. Used common nearest neighbors
   - Efficiently find k-NN
   - Incrementally compute edge weight

2. LU decomposition
   - Efficiently compute Eigen decomposition
   - Low memory consumption
New Query: A cluster-group-by query is given a set $Q$ of data points, and groups the points of $Q$ by the clusters they belong to.

Contributions:
Data structures with fast update and query time.

For $Q = \{q_1, q_4, q_5\}$, answer: $\{q_1 \cup q_4, q_5\}$.

For $Q = \{q_1, q_2, q_4\}$, answer: $\{q_1, q_4\}, \{q_2\}$.

Lower bounds when such structures do not exist.
Deep Insights has been a sub-branch project of the Auto Insights research framework at Microsoft Research.

Auto Insights has been continuously shipping new techniques (e.g., Quick Insights, Scoped Insights, etc.) to Microsoft Power BI since Dec 2015, as enabling techniques for leading the BI & Analytics market.

Mining Deep Insights against hierarchical meta cube
E.g., Brand F’s rank\textsubscript{3} (across all brands) w.r.t. YOY increase\textsubscript{2} of sales\textsubscript{1} has a rising trend.
Problem:
The optimal curve for the target query pattern

Square Query

Elongate Rect. Query

Z-Curve

C-Curve (Composite Index)

Contributions:
• Cohesion-based Cost Model
  • Measure curve property for query pattern and data distribution

• Curve Design Method
  • Heuristics to design effective curves in terms of the cost model
Leveraging Re-costing for Online Optimization of Parameterized Queries with Guarantees

Anshuman Dutt, Vivek Narasayya and Surajit Chaudhuri (Microsoft Research)

Parameterized query

Select attributes
From relations
Where join predicates and other predicates and i_current_price < @Param1 and cs_sales_price < @Param2

Problem: online version of parametric query optimization (PQO)

Performance Trade-off

Many different query instances may lead to same optimal execution plan

Opportunity: to avoid optimizer overhead
Handling Environments in a Nested Relational Algebra with Combinators and an Implementation in a Verified Query Compiler

Joshua Auerbach and others (IBM Research)

Handling Environments:
- Keep Variables: simple plans, complex rewrites
- Remove Variables: simple rewrites, complex plans

Nested Relational Algebra with Combinators:
- $\text{NRAEnv} = \text{NRA Combinators} + \text{Environment}$
- Definition, Expressivity, Rewrites, Applications

Implementation:
- Written with Coq Proof Assistant
- Algebraic Optimizer Verified Correct
- $\text{Q*cert}$ demo at SIGMOD 2017

Verified Query Compiler:
- $\text{Q*cert}$
  - https://querycert.github.io/
From In-Place Updates to In-Place Appends: Revisiting Out-of-Place Updates on Flash
S. Hardock*, I. Petrov†, R. Gottstein* and A. Buchmann*
(*TU Darmstadt, †Reutlingen University)

Problem:
- Small updates $\rightarrow$ write-amplification $600\times$

Approach:
- Small updates $\rightarrow$ physical in-place appends

IPA: Flash updates without a prior erase