T-79.514 Special Course on Cryptology

# Seminar 10: Secure Approximate Matching

Matti Järvisalo
Helsinki University of Technology
`http://www.tcs.hut.fi/~mjj/`

# Motivation

- A scenario: Alice wants to compare her DNA against a DNA DB with known genetic diseases $\Rightarrow$ privacy concerns!
- Need for privacy in e.g. e–commerce, banking/health/etc. records
- In many cases exact matching is not possible
- Exact matching well–studied, approximate not so much
- High interest in *efficient* protocols (MPC too general)

# Overview of the Lecture

- Secure Database Access (SDA)
- SDA in Different Models and Metrics
- Overview of Protocols for the Models
- More In-Depth Look at one Protocol

Based on *W. Du, M.J. Atallah. Protocols for Secure Remote Database Access with Approximate Matching*, appeared in ACM CCS 2000.

# Secure Database Access (SDA)

The SDA Problem:

> Alice has a string $q$, and Bob has a database of strings $T = \{t_1, \ldots, t_N\}$. Alice wants to know whether there exists a string $t_i \in T$ that *matches* $q$. Give a protocol that accomplishes this without revealing to Bob neither (i) $q$ nor (ii) the found match.

- The answer depends on whether exact or approximate PM is considered
- Depending on the model, the result can be either the closest match or the distance to the closest match

# Metrics

Let $a = (a_1 \ldots a_n)$, $b = (b_1 \ldots b_n)$ be two strings. Possible metrics are:

- $\sum_{i=1}^{n} |a_i - b_i|$ (e.g. in image processing)
- $\sum_{i=1}^{n} (a_i - b_i)^2$ (e.g. in image processing)
- $\sum_{i=1}^{n} f(a_i, b_i)$ ($f$ a function)
- *edit distance* (e.g. in string matching)
- # of indices in which $a$ and $b$ differ, etc.

# Models: Overview

- Database $T$, possessed by Bob
  - ⋆ Number of entries (strings) $N$
  - ⋆ Each string of length $n$
  - ⋆ Each string over an alphabet of size $m$ (might be infinite)
- Four models, differences in
  - ⋆ whether $T$ is private;
  - ⋆ who owns $T$; and
  - ⋆ who may query $T$.

# Models: PIM

**Private Information Matching model (PIM)**.

- Alice has a query string $q$, and wants to know Match$(q, T)$ without revealing $q$ nor Match$(q, T)$ to Bob.

- Bob, the *sole* possessor of $T$, doesn't want to reveal any $t_i \in T$ to Alice except what can be derived from Match$(q, T)$.

- Alice has to query $T$ through Bob.

# Models: PIMPD

**Private Information Matching from Public Database model (PIMPD)**.

As PIM, but

- $T$ is public
- the privacy concerns is that Alice doesn't want to reveal $q$ nor Match$(q, T)$ to Bob.

# Models: SSO

**Secure Storage Outsourcing model (SSO)**:

- The owner of $T$ is Alice, but $T$ has been outsourced to Bob (e.g. for storage space reasons).

- Alice wants to query $T$ without revealing $T$ nor $q$ to Bob.

# Models: SSCO

**Secure Storage and Computing Outsourcing model (SSCO)**:

SSO with the following extension:

- any individual may query $T$
- Alice should be aware of any such queries.
- The individual making the query should learn the distance of the closest match from the query, while this should be kept secret from Alice.

# Overview of Results

| Model | Metrics | CC | 3rd ? |
|---|---|---|---|
| PIM | $\sum_{i=1}^{n}(a_i - b_i)^2$ | $\mathcal{O}(nN)$ | yes |
| | $\sum_{i=1}^{n}\lvert a_i - b_i\rvert$ | $\mathcal{O}(nWN)$ | yes |
| | $\sum_{i=1}^{n} f(a_i, b_i)$ | $\mathcal{O}(mnN)$ | yes |
| SSO | $\sum_{i=1}^{n}(a_i - b_i)^2$ | $\mathcal{O}(n)$ | no |
| SSCO | $\sum_{i=1}^{n}(a_i - b_i)^2$ | $\mathcal{O}(n^2)$ | yes |

- $W$ an accuracy parameter (in a Monte Carlo – based protocol)
- PIMPD is a special case of PIM $\Rightarrow$ same protocols applicable
- Third party needed for computing scalar products $\mathbf{x} \cdot \mathbf{y}$ of Alice's $\mathbf{x}$ and Bob's $\mathbf{y}$.

# Protocol for SSO: Preliminaries

Idea: pick a random matrix and disguise $T$ before outsourcing. Do the same for $q$.

- Let $\mathbf{Q}$ be an $(n+3) \times (n+3)$ random invertible matrix
- Let $R$, $R_A$ and $R_i$, $i \in \{1, \ldots, N\}$, be random numbers, private to Alice
- For each string $t_i = t_{i,1} \ldots t_{i,n} \in T$, we have a vector $\mathbf{t}_i = (\sum_{k=1}^{n} t_{i,k}^2 + R - R_i, t_{i,1}, \ldots, t_{i,n}, 1, R_i)$ of length $n+3$
- In $T'$, the outsourced version of $T$, we have the entry $\mathbf{t}'_i = \mathbf{Q}\mathbf{t}_i{}^T$

# Protocol for SSO

1. Alice
   - generates $R_A$,
   - constructs $\mathbf{q} = (1, -2q_1, \ldots, -2q_n, R_A, 1)$, and
   - sends $\mathbf{q}\mathbf{Q}^{-1}$ to Bob.

2. Bob
   - computes $\text{score}_i = \mathbf{q} \cdot \mathbf{t}_i{}^T$ for each $\mathbf{t}'_i \in T'$,
   - determines $\arg\min_{i=i}^{N} \text{score}_i$, and
   - sends $\mathbf{t}'_i$ to Alice.

3. Alice determines the closest match $\mathbf{t}_i = \mathbf{Q}^{-1}\mathbf{t}'_i$.

# Notes on the Protocols (1/2)

For SSO and SSCO

- Quite similar solutions
- As Carl may also query, calculating $\mathbf{x} \cdot \mathbf{y}$ between Alice and Carl brings $\mathcal{O}(n)$ to communication complexity
- For SSCO the answer is only the distance to the closest match

# Notes on the Protocols (2/2)

For PIM and PIMPD

- Not reasonable due to high communication complexity
- Similar to computing $\mathbf{x} \cdot \mathbf{y}$ for $\sum_{i=1}^{n}(a_i - b_i)^2$
- A bit obfuscated Monte–carlo based protocol for $\sum_{i=1}^{n} |a_i - b_i|$, answer is only the distance to the closest match . . .
- . . . as well as for $f$
- For $f$, predefined *finite* alphabet is required

# In Addition

- No protocol given for edit distance, although it is said that one exists
- The need for a third party problematic; could this be avoided?
- It is proposed that a sublinear dependency w.r.t. $N$ might be possible