# Assignment: 3-Achieving Usable and Privacy-assured Similarity Search over Outsourced Cloud Data

Xinyi Li

Dalian University of Technology
*Assignment of System Security*

March 27, 2015

# Overview

# Introduction of the Paper

C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *INFOCOM, 2012 Proceedings IEEE*, pp. 451–459, IEEE, 2012.

# Introduction of the Paper

## Purpose

Solve the problem of secure and efficient fuzzy search over encrypted outsourced cloud data

## Measures

- Suppressing technique
- Building a private trie-traverse searching index

## Performance

Correctly achieves the defined similarity search functionality with **constant** searching time!

# System and Threat Model

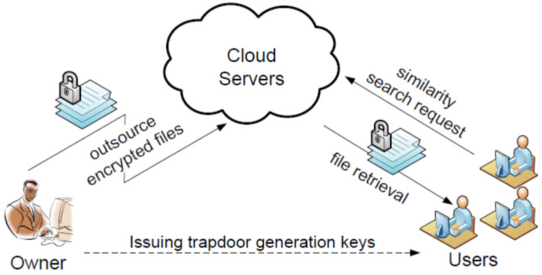


Figure: Architecture of similarity keyword search over outsourced cloud data

- data owner: the individual/enterprise customer,who has a collection of $n$ data files $C = (F_1, F_2, \ldots, F_n)$ to be stored in the cloud server.
- $W = \{w_i, w_w, \ldots, w_p\}$ is denoted as a predefined set of distinct keywords in $C$

# System and Threat Model

- Files are encrypted before outsourced
- The data owner will distribute search request(trapdoor) generation keys *sk* to authorized users.(Assume that the authorization will be done appropriately)
- An authorized user uses trapdoor generation key to generate a search request via some one-way function to search word $w$, and submit it to the cloud.
- The cloud then performs the search over the data collection $C$ without decryption and sends back all encrypted files containing the specific keyword $w$, denoted as $FID_w$.
- The similarity keyword search scheme returns the closest possible results based on aforementioned measures.
- At last, the user decrypts files they received from the cloud.

# Assumption: Honest-but-curious clound server

To ensure the securely similarity searching schema:

**Honest**

Correctly follows the designated protocol specification

**Curious**

Infer and analyze the message flow received during the protocol so as to learn additional information

We follow the security definition deployed in the traditional searchable symmetric encryption(SSE)

# Notations

- $C$ the file collection to be outsourced, denoted as a set of $n$ data files $C = (F_1, F_2, \dots, F_n)$.
- $W$ the distinct keywords extracted from file collection $C$, denoted as a set of $m$ words $W = \{w_i, w_w, \dots, w_p\}$.
- $\mathcal{I}$ the index built for privacy-assured similarity search.
- $T_w$ the trapdoor generated by a user as a search request of input keyword $w$ via some one-way transformation.
- $S_{w,d}$ similarity keyword set of $w$, where $d$ is the similarity threshold according to a certain similarity metrics.
- $FID_{w_i}$ the set of identifiers of files in $C$ that contain keyword $w_i$.
- $f(key, \cdot), g(key, \cdot)$ pseudorandom function (PRF), defined as: $\{0,1\}^* \times key \rightarrow \{0,1\}^\ell$.
- $Enc(key, \cdot), Dec(key, \cdot)$ symmetric key based semantic secure encryption/decryption function.

# Edit Distance

### Quantitative measurement

The edit distance $ed(w_1, w_2)$ between two words $w_1$ and $w_2$ is the mininum number of primitive operations, including character insertion, deletion and substitution, necessary to transform one of them into the other.

### Similarity keyword set

Given a keyword $w$, we let $S_{w,d}$ denote its similarity set of words, such that any $w' \in S_{w,d}$ satisfies $ed(w, w') \leq d$ for a certain integer $d$.

### Example

Consider the keyword $w_0 = CENSOR$
a words set W $= \{CESOR, CENSER, CEANSOR\}$
for any $w' \in W, ed(w_0, w') \leq 1$ holds,
i.e. $w' \in S_{w_0,1}$ and $W \subseteq S_{w_0,1}$

# Building Similarity Keyword Sets

## Straightforward approach

Simply enumerating all possible words $w_i'$ satisfying the similarity criteria $ed(w_i, w_i') \leq d$

For the keyword $w_0 = CENSOR$, consider just one substitution operation with charaters on first character.

There are 26 items $\{AENSOR, BENSOR, \ldots, YENSOR, ZENSOR\}$

So $S_{w_0,1}$ will be $[6 + (6 + 1)] \times 26 + 1$

## Suppression technique

Consider only the positions of the primitive edit operations.Specifically, we use a wildcard * to denote all three operations of character insertion, deletion and substitution at any position.

Now,

$S_{SENSOR,1} = \{SENSOR, *SENSOR, *ENSOR, S*ENSOR, S*NSOR, \ldots, SENSO*R, SENSO*, SENSOR*\}$.

Size can be reduced to $S_{w_0,1}$ will be $[6 + (6 + 1)] \times 1 + 1$

# Building Similarity Keyword Sets

**Algorithm 1:** CreateSimilaritySet($w_i$, $d$)

**Data**: keyword $w_i$ and threshold distance $d$
**Result**: similarity keyword set $S_{w_i,d}$
**begin**
  **if** $d > 1$ **then**
    CreateSimilaritySet($w_i$, $d - 1$);
  **if** $d = 0$ **then**
    set $S_{w_i,d} = \{w_i\}$;
  **else**
    **for** $k \leftarrow$ *1 to* $|S_{w_i,d-1}|$ **do**
      **for** $j \leftarrow$ *1 to* $2 \times |S_{w_i,d-1}[k]| + 1$ **do**
        **if** $j$ *is odd* **then**
          Set $variant$ as $S_{w_i,d-1}[k]$;
          Insert $\star$ at position $\lfloor (j+1)/2 \rfloor$;
        **else**
          Set $variant$ as $S_{w_i,d-1}[k]$;
          Replace $\lfloor j/2 \rfloor$-th character with $\star$;
        **if** *variant is not in* $S_{w_i,d-1}$ **then**
          Set $S_{w_i,d} = S_{w_i,d} \cup \{variant\}$;

The size of $S_{w_i,d}$ will be $\mathcal{O}(\ell^d)$,opposing to $\mathcal{O}(\ell^d \times 26^d)$ obtained in the straightforward approach.

### Theorem

*The intersection of the similarity sets $S_{w_i,d}$ and $S_{w,d}$ for keyword $w_i$ and search input $w$ is not empty if and only if $ed(w, w_i) \leq d$.*

### Proof.

- Completeness(i.e. $ed(w, w_i) \leq d \rightarrow S_{w_i,d} \cap S_{w,d} \neq \emptyset$ ):
  - $w \rightarrow w_i$ need at most $d$ primitive operations.
  - $w$ must be in $S_{w_i,d}$
  - $w$ is naturally in $S_{w,d}$

$\square$

# Generating Searching Request

**Proof.**

- Soundness(i.e. $S_{w_i,d} \cap S_{w,d} \neq \emptyset \rightarrow ed(w, w_i) \leq d$)

  $w^*$ the common element in $S_{w_i,d} \cap S_{w,d}$

  1. $w^*$ does not contain any wildcard *,
     then $w^* = w = w'$,and $ed(w, w') = 0 \leq d$
  2. $w^*$ does contain some wildcard *(at most d *'s),
     change * in $w^*$ back to the character in $w$ and $w_i$,
     denote the result as $w'^*$ and $w_i'^*$ with both sharing $d - 1$ different *'s.
     $w'^* \rightarrow w_i'^*$ need at most one primitive operation.
     So, $ed(w'^*, w_i'^*) \leq 1$
     $\Rightarrow ed(w, w_i) \leq d$

$\square$

# Paragraphs of Text

Sed iaculis dapibus gravida. Morbi sed tortor erat, nec interdum arcu. Sed id lorem lectus. Quisque viverra augue id sem ornare non aliquam nibh tristique. Aenean in ligula nisl. Nulla sed tellus ipsum. Donec vestibulum ligula non lorem vulputate fermentum accumsan neque mollis.

Sed diam enim, sagittis nec condimentum sit amet, ullamcorper sit amet libero. Aliquam vel dui orci, a porta odio. Nullam id suscipit ipsum. Aenean lobortis commodo sem, ut commodo leo gravida vitae. Pellentesque vehicula ante iaculis arcu pretium rutrum eget sit amet purus. Integer ornare nulla quis neque ultrices lobortis. Vestibulum ultrices tincidunt libero, quis commodo erat ullamcorper id.

# Bullet Points

- Lorem ipsum dolor sit amet, consectetur adipiscing elit
- Aliquam blandit faucibus nisi, sit amet dapibus enim tempus eu
- Nulla commodo, erat quis gravida posuere, elit lacus lobortis est, quis porttitor odio mauris at libero
- Nam cursus est eget velit posuere pellentesque
- Vestibulum faucibus velit a augue condimentum quis convallis nulla gravida

# Blocks of Highlighted Text

## Block 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

## Block 2

Pellentesque sed tellus purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vestibulum quis magna at risus dictum tempor eu vitae velit.

## Block 3

Suspendisse tincidunt sagittis gravida. Curabitur condimentum, enim sed venenatis rutrum, ipsum neque consectetur orci, sed blandit justo nisi ac lacus.

# Multiple Columns

**Heading**

1. Statement
2. Explanation
3. Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

# Table

| Treatments | Response 1 | Response 2 |
|---|---|---|
| Treatment 1 | 0.0003262 | 0.562 |
| Treatment 2 | 0.0015681 | 0.910 |
| Treatment 3 | 0.0009271 | 0.296 |

Table: Table caption

# Theorem

**Theorem (Mass–energy equivalence)**

$E = mc^2$

# Verbatim

**Example (Theorem Slide Code)**

```
\begin{frame}
\frametitle{Theorem}
\begin{theorem}[Mass--energy equivalence]
$E = mc^2$
\end{theorem}
\end{frame}
```

# Figure

Uncomment the code on this slide to include your own image from the same directory as the template .TeX file.

# Citation

An example of the `\cite` command to cite within the presentation:

This statement requires citation [Smith, 2012].

# References


John Smith (2012)
Title of the publication
*Journal Name* 12(3), 45 – 678.

# The End