ADDING DIFFERENTIAL PRIVACY TO AN OPEN SOURCE DISCUSSION BOARD

Pragya Rana

Advisor: Dr. Chris Pollett
Committee members: Dr. Melody Moh, Mr. Mahesh Subedi

Department of Computer Science
San Jose State University
Agenda

- Introduction
- Background
- Preliminary Work
- Design/Implementation
- Experiment
- Conclusion
Introduction

• Various online platforms created for users: social network, e-commerce, video streaming, etc.

• These platforms collect personal information for statistical analysis. E.g., Amazon recommends the products to users based on browsing history
Introduction

• Numerous attacks on database systems on a frequent basis

• Relying on older ways of authentication and access control are not enough

• Typical approaches when releasing statistics/synopses:
  • Sanitization/Anonymization: remove well-known identifiers such as names, dob, son
Introduction

Cases where releasing anonymized data failed to preserve the privacy

• Identification of medical records of MA governor in public “anonymized” medical database

• Identification of search history of Thelma Arnold in public “anonymized” AOL query records
Introduction

So how can we protect a user’s privacy who is participating in the statistical analysis?

*If we can ensure a user about the chance that the released statistics would be nearly the same, whether or not he/she submitted his/her information.*
Introduction

• **Goal**: Implement some privacy techniques to a statistical database

• We are using Yioop system to implement privacy techniques

• **Yioop** is an open source search engine developed by Dr. Chris Pollett

• Techniques implemented in Yioop:
  
  • **Differential Privacy**
  
  • **Database Encryption**
Background

• **What is Differential Privacy?**

  “a randomized function $K$ gives $\varepsilon$-differential privacy if for all data sets $D_1$ and $D_2$ differing on at most one element, and all $S \subseteq \text{Range}(K)$, 
  \[ \Pr[K(D_1) \in S] \leq \exp(\varepsilon) \times \Pr[K(D_2) \in S] \]” \[1\]

  *a mechanism $K$ that satisfies above definition ensures the user that any responses to queries is equally likely to occur even if the user decides to remove his/her data from the data set* \[1\]
Example

Statistical study to show that smoking causes cancer:

• If a user Mary is a smoker, then there two harms to Mary from the study:
  • Her insurance will go up if the insurance provider consults the database
  • She learns that smoking causes cancer (which can be helpful to her and also helps the medical research)

• Can we ensure Mary that the impact on her insurance remains the same whether or not she opts in or out of the database
  • $D_1$ = Data set when Mary is in the database
  • $D_2$ = Data set when Mary is not in the database
  • $S$ = Query result set
  • $P(K(D_1) \in S) \sim P(K(D_2) \in S)$
Two models of privacy mechanism

1. Non-Interactive Setting: data collector publishes a sanitized version of the collected data (de-identification, anonymization)

2. Interactive Setting: data collector provides an interface through which users present queries about the data to get some answers with some added noise
Privacy Mechanism in Differential Privacy

- An interactive privacy mechanism is used for achieving differential privacy.
  
  - The mechanism works by adding appropriately chosen random noise to the answer $a = f(X)$, where $f$ is the query function and $X$ is the database. [1]
Database Encryption

• Previous works done to secure the database. One of them is Negative Database [2]
  
  • A negative database contains data that includes real data as well as negative data.
  
  • We have applied this concept for our database.

• Different database encryption methods such as Symmetric/Asymmetric, Field Level, Column Level, External database encryption, etc.
  
  • We have used application level encryption
Preliminary Work

In order to implement differential privacy, we needed to show the statistics:

- Extended feature of Yioop in the statistics of discussion board system by adding graphical view of the statistics
Preliminary Work

• Developed test suite of statistical attacks against query and discussion board statistics.

• Implemented differential privacy algorithm in the group’s thread view.

• Made necessary changes to the database needed for adding differential privacy
Design/Implementation

Defining policy based on which differential privacy is targeted on the specific data set

- Different types of contents in Yioop: groups, threads, wikis, search

- Identify data sets that require higher level of privacy. Mostly statistics computed by:
  
  - Group Analytics
  
  - Search Analytics
Design/Implementation

Controlling Security Feature from the UI level

• Added an option to enable/disable Differential Privacy under Security section
Security Feature

- Admin [Security]

**Authentication and Captcha Types**

- **Authentication Type**
  - Normal Authentication

- **Captcha Type**
  - Text Captcha

- **Recovery Type**
  - Email Link Password Recovery

- **Privacy**
  - Differential Privacy: Enable
  - Group Analytics: Disable
  - Search Analytics: Disable

**Captcha and Recovery Questions**
Design/Implementation

Database encryption at an application level

- Identify which data is more sensitive and requires higher privacy
- Perform encryption only in those data
- Type of encryption
  - Not entire database needs to be encrypted
  - Use application level encryption.
  - Use column level encryption
Design/Implementation

Additional level of security

• Symmetric keys stored in an external database.

• Concept of Negative Database [2] has been applied
  • Before encrypting data, add some negative data to the real data
  • When decrypting data, remove those negative data and display the real data

• So even if intruder gets access to the main database, won’t be able decrypt without having access to external database
Data encryption/decryption process

- **Model 1**: Encrypted Data
  - Encrypt using key
  - Decrypt using key
  - Remove Negative Data

- **Controller**: Update
  - User Action
  - Retrieve Keys

- **View**:

- **Model 2**: Encryption Keys
Design/Implementation

Added Database Encryption to discussion board system

• Current Discussion Board System has:
  • Different groups: each group has a list of users
  • Users can post different threads, add/edit/delete comments
  • vote +/- for each thread
  • Identify data that requires additional level of privacy
  • Threads posted by all users and it’s replies/comments
Design/Implementation

• Database Encryption added as an option when creating a new group

• Under Manage Group section, when you create a new group, there is a drop down menu for Encryption field

• Two options: Enable/Disable
Design/Implementation
Design/Implementation

• If encryption is enabled for a group, all posts in that group are encrypted before storing to the database

• When displaying the posts of a group, key which is stored in an external database is accessed first in order to decrypt data before displaying
Encrypted/Decrypted data

Final exam on May 23!

Comment

Final exam on May 23! (+0/-0). - 16 m 27 s ago TestGroup1
The final exam will be held on May 23!
Vote: [+ -]

user1

Final exam on May 23! (+0/-0). - 0 m 0 s ago TestGroup1
What are the chapters that will be included in Final?
Vote: [+ -]

[Edit] [X]

<table>
<thead>
<tr>
<th>TITLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>p/d. _K OF p0000000000ZUuFFdD...</td>
<td>M%KK K K p&lt; 0000...</td>
</tr>
</tbody>
</table>
Differential Privacy

Privacy mechanism, $K_f$ for a query function $f$, computes $f(x)$ and adds noise with a scaled symmetric exponential distribution with variance $\sigma$ in each component. [1]

Pr[$K_f(X) = a] \propto \exp(-\|f(X) - a\|/\sigma)$
Design/Implementation

Existing Groups Statistics Page

• Current analytics job uses raw data accumulated from each group’s activities

• Aggregates those data into different time periods giving statistics hourly, daily, monthly, yearly, all time

• These statistics gives information on how frequently certain group or thread or wiki is visited
Group Statistics View

<table>
<thead>
<tr>
<th>Group Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Hour: No Activity</td>
</tr>
<tr>
<td>Last Day: No Activity</td>
</tr>
<tr>
<td>Last Month: No Activity</td>
</tr>
<tr>
<td>Last Year: No Activity</td>
</tr>
<tr>
<td>All Time: No Activity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Thread Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Hour: No Activity</td>
</tr>
<tr>
<td>Last Day: No Activity</td>
</tr>
<tr>
<td>Last Month: No Activity</td>
</tr>
<tr>
<td>Last Year: No Activity</td>
</tr>
<tr>
<td>All Time:</td>
</tr>
<tr>
<td>404 Wiki Page Created: No Activity</td>
</tr>
<tr>
<td>409 Wiki Page Created: 2</td>
</tr>
<tr>
<td>Syntax Wiki Page Created: 1</td>
</tr>
<tr>
<td>ac_program_terms Wiki Page Created: No Activity</td>
</tr>
<tr>
<td>advertise Wiki Page Created: No Activity</td>
</tr>
<tr>
<td>bot Wiki Page Created: 2</td>
</tr>
<tr>
<td>captcha_time_out Wiki Page Created: 2</td>
</tr>
<tr>
<td>presentation Wiki Page Created: No Activity</td>
</tr>
<tr>
<td>privacy Wiki Page Created: 1</td>
</tr>
<tr>
<td>register_time_out Wiki Page Created: No Activity</td>
</tr>
<tr>
<td>suggest_day_exceeded Wiki Page Created: No Activity</td>
</tr>
<tr>
<td>terms Wiki Page Created: 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wiki Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Hour: No Activity</td>
</tr>
<tr>
<td>Last Day: No Activity</td>
</tr>
<tr>
<td>Last Month: No Activity</td>
</tr>
<tr>
<td>Last Year: No Activity</td>
</tr>
<tr>
<td>All Time:</td>
</tr>
<tr>
<td>404: 1</td>
</tr>
<tr>
<td>409: 3</td>
</tr>
<tr>
<td>Syntax: 2</td>
</tr>
<tr>
<td>ac_program_terms: 2</td>
</tr>
<tr>
<td>advertise: 2</td>
</tr>
<tr>
<td>bot: 2</td>
</tr>
<tr>
<td>captcha_time_out: 1</td>
</tr>
<tr>
<td>presentation: 4</td>
</tr>
<tr>
<td>privacy: 3</td>
</tr>
<tr>
<td>register_time_out: 4</td>
</tr>
<tr>
<td>suggest_day_exceeded: No Activity</td>
</tr>
<tr>
<td>terms: No Activity</td>
</tr>
</tbody>
</table>
Design/Implementation

Adding Differential Privacy to Groups Statistics Page

For each time period under group, thread and wiki, calculate the views using Differential Privacy Algorithm and display the fuzzified value.
Design/Implementation

Adding Differential Privacy to Query Statistics Page

• Query Statistics page displays statistics about each query entered by user in the search box

• Sensitive information about the user

• Critical to ensure the privacy of the user
Design/Implementation

Search Query Statistics

Last Hour: No Activity
Last Day: No Activity
Last Month: No Activity
Last Year: No Activity
All Time:
san jose: 2
costco: 1
san francisco: 1
jazz: 1
Design/Implementation

• Once Differential Privacy has been enabled, the actual count for each search query is fuzzified

• Makes it incomprehensible for anyone to extract the exact information
Testing/Experiment

• Basic Set up
  • Create 100 users, 50 groups
  • Add 20 threads to Group1
  • Generate statistics by simulating users visiting 20 threads randomly
Testing/Experiment

- Statistics displayed by differential privacy does not reveal exact count
- Makes it difficult for an adversary to perform statistical attacks

Table: Statistics of Group’s views

<table>
<thead>
<tr>
<th>Differential Privacy vs. Non-DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-DP</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>200</td>
</tr>
<tr>
<td>400</td>
</tr>
<tr>
<td>800</td>
</tr>
<tr>
<td>1600</td>
</tr>
<tr>
<td>3200</td>
</tr>
<tr>
<td>6400</td>
</tr>
<tr>
<td>12800</td>
</tr>
<tr>
<td>25600</td>
</tr>
<tr>
<td>51200</td>
</tr>
<tr>
<td>102400</td>
</tr>
<tr>
<td>204800</td>
</tr>
<tr>
<td>409600</td>
</tr>
<tr>
<td>819200</td>
</tr>
<tr>
<td>1638400</td>
</tr>
<tr>
<td>3276800</td>
</tr>
<tr>
<td>1638400</td>
</tr>
</tbody>
</table>

Conclusion

• Data privacy issues are becoming important in database systems

• Database serves many useful goals.

• Better participation -> Better results

• Differential privacy encourages participation

• Already used in various real-life applications
  - Google -> historical traffic statistics
  - U.S Census Bureau -> commuting patterns
References


