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 ε-differential privacy if for all data sets D1 and D2 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
    - D2 = 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

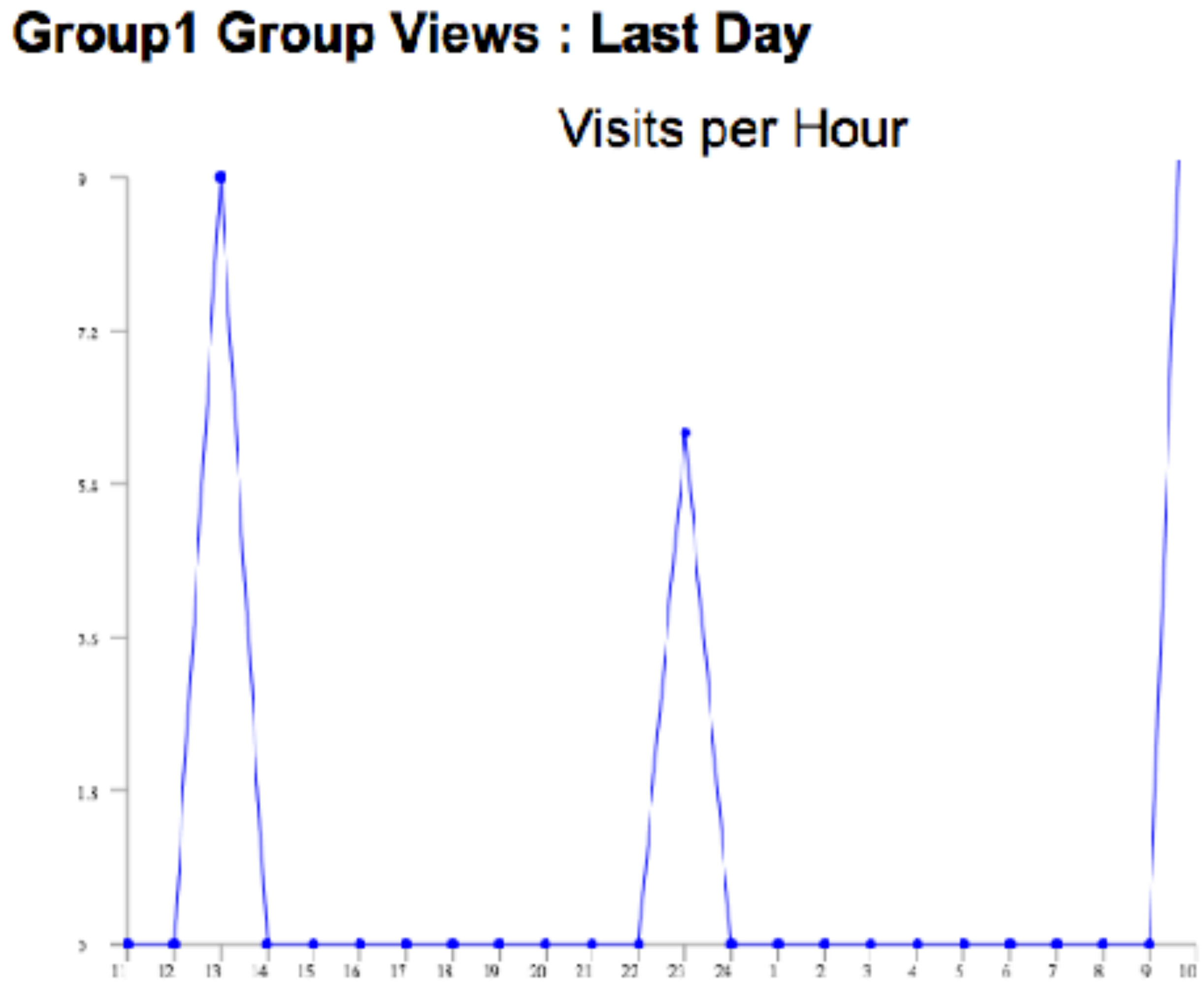
In order to implement differential privacy, we needed to show the statistics:

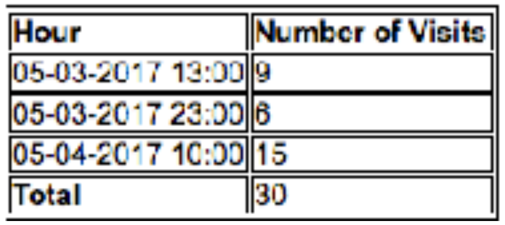
- Extended feature of Yioop in

the statistics of discussion board

system by adding graphical

view of the statistics





Hour	Number of Visits
05-03-2017 13:00	9
05-03-2017 23:00	6
05-04-2017 10:00	15
Total	30

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

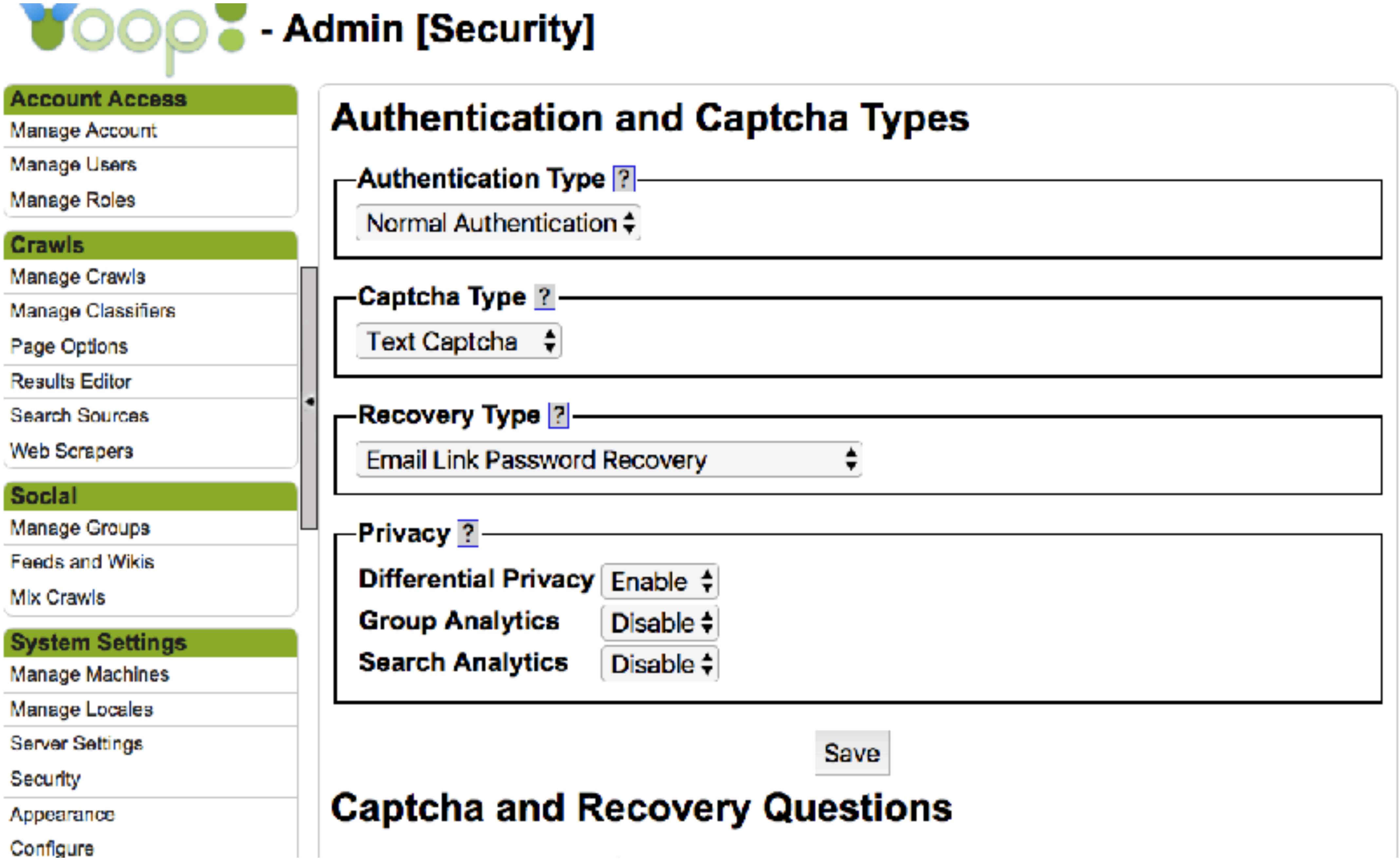
- Different types of contents in Yioop: groups, threads, wikis, search
- Identify data sets that require higher level of privacy. Mostly statistics computed by:
    - Group Analytics
    - Search Analytics

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

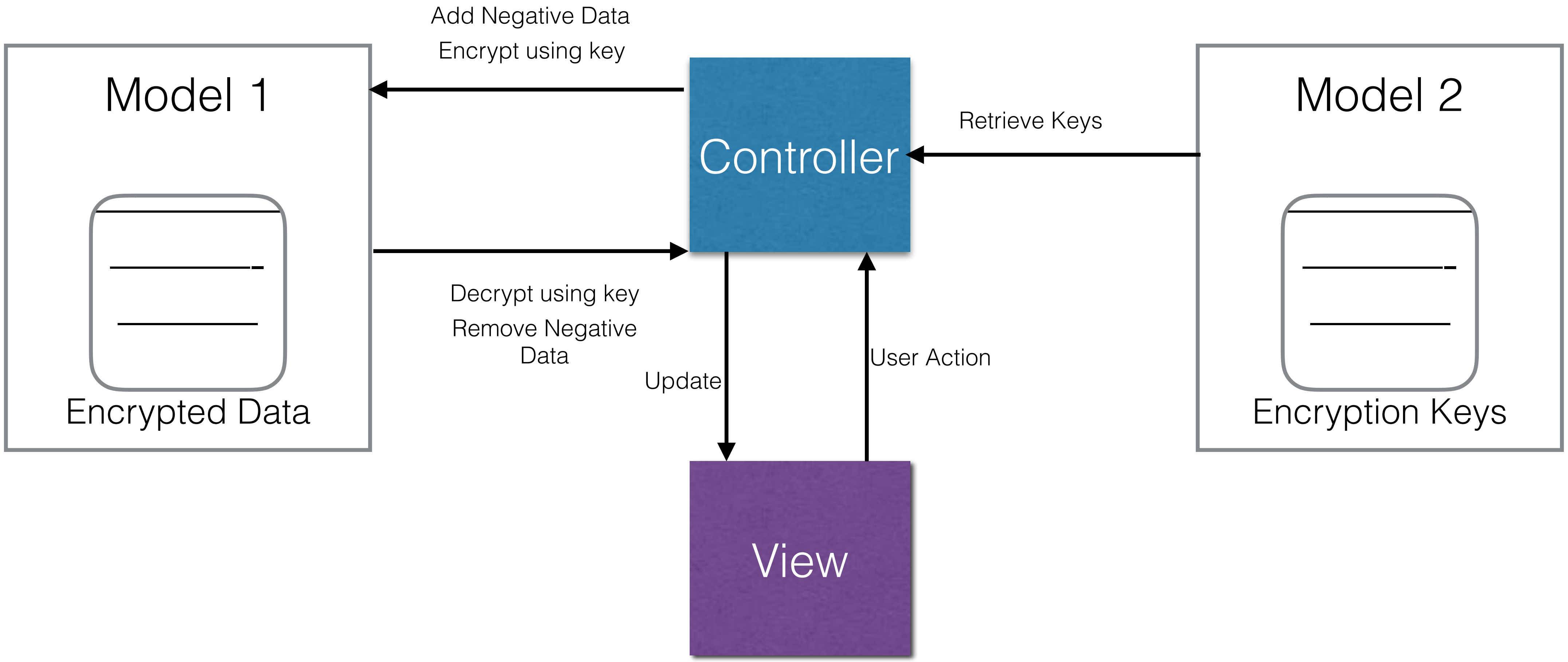
- Type of encryption
  - Not entire database needs to be encrypted
  - Use application level encryption.
  - Use column level encryption

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



Design/Implementation

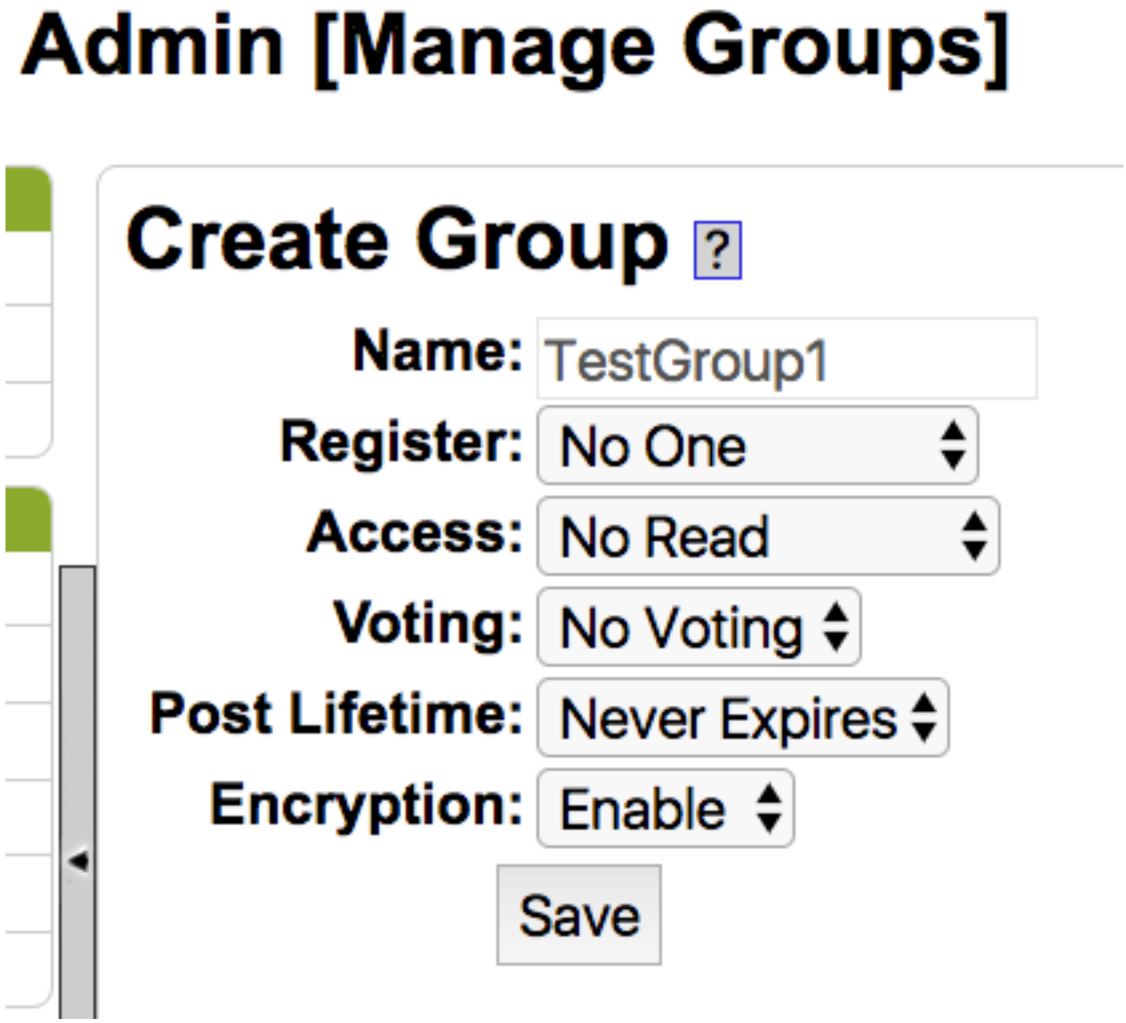
Added Database Encryption to discussion board system

- 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



Admin [Manage Groups]

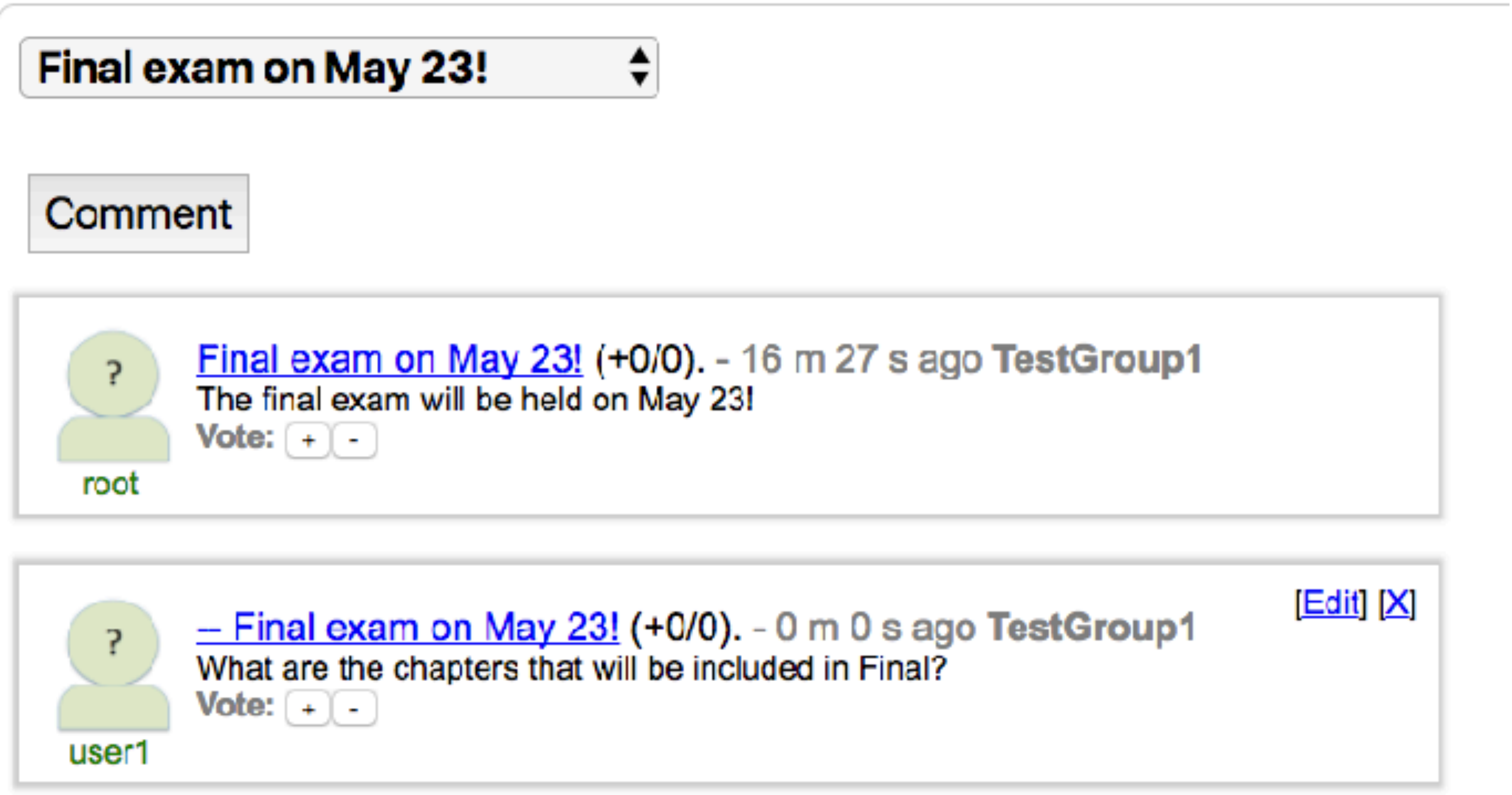
Create Group

Name: TestGroup1
Register: No One
Access: No Read
Voting: No Voting
Post Lifetime: Never Expires
Encryption: Enable

Design/Implementation

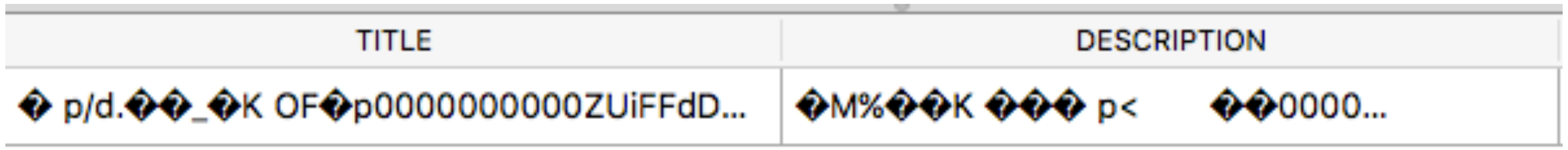
- If encryption is enabled for a group, all posts in that group are encrypted before storing to the database
- When displaying the posts of a group, key which is stored in an external database is accessed first in order to decrypt data before displaying

Encrypted/Decrypted data



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

[Edit] [X]
– Final exam on May 23! (+0/0). - 0 m 0 s ago **TestGroup1**
What are the chapters that will be included in Final?
Vote: + -
user1



TITLE	DESCRIPTION
� p/d.��_�K OF�p000000000ZUiFFdD...	�M%��K ��� p< ��0000...

Design/Implementation

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 σ in each component. [1]

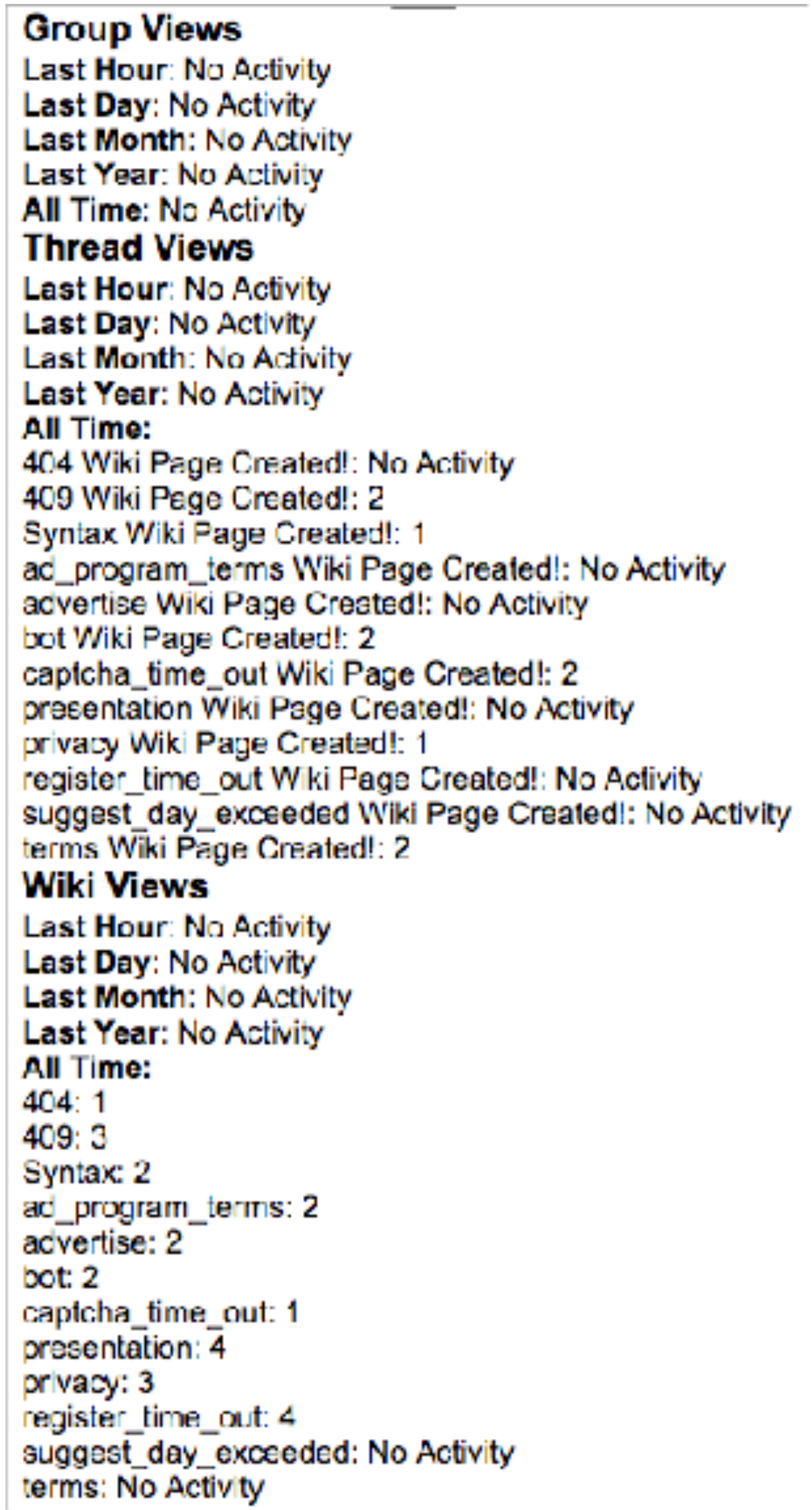
$Pr[K_f(X) = a] \propto \exp(-\|f(X) - a\|/\sigma)$

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



Group Views

Last Hour: No Activity
Last Day: No Activity
Last Month: No Activity
Last Year: No Activity
All Time: No Activity

Thread Views

Last Hour: No Activity
Last Day: No Activity
Last Month: No Activity
Last Year: No Activity
All Time:
404 Wiki Page Created!: No Activity
409 Wiki Page Created!: 2
Syntax Wiki Page Created!: 1
ad_program_terms Wiki Page Created!: No Activity
advertise Wiki Page Created!: No Activity
bot Wiki Page Created!: 2
captcha_time_out Wiki Page Created!: 2
presentation Wiki Page Created!: No Activity
privacy Wiki Page Created!: 1
register_time_out Wiki Page Created!: No Activity
suggest_day_exceeded Wiki Page Created!: No Activity
terms Wiki Page Created!: 2

Wiki Views

Last Hour: No Activity
Last Day: No Activity
Last Month: No Activity
Last Year: No Activity
All Time:
404: 1
409: 3
Syntax: 2
ad_program_terms: 2
advertise: 2
bot: 2
captcha_time_out: 1
presentation: 4
privacy: 3
register_time_out: 4
suggest_day_exceeded: No Activity
terms: No Activity

Design/Implementation

Adding Differential Privacy to Groups Statistics Page

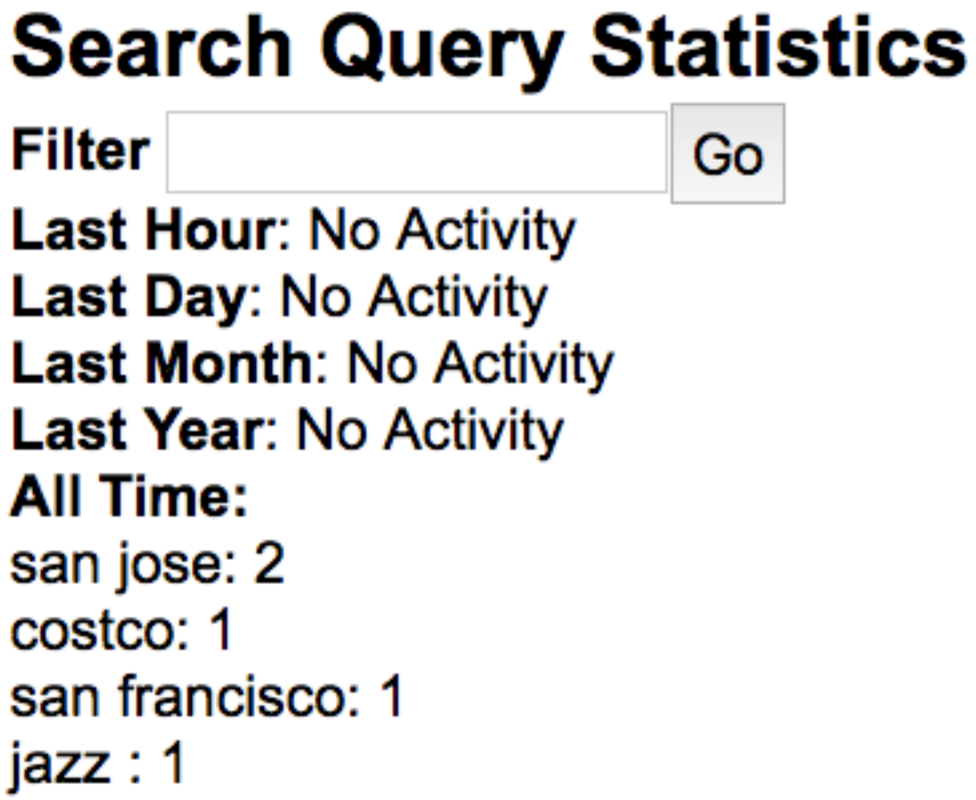
For each time period under group, thread and wiki, calculate the views using Differential Privacy Algorithm and display the fuzzified value.

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

Filter

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

Non-DP	DP
20	17
50	44
100	78
200	258
400	233
800	617
1600	1370
3200	3211
6400	6434
12800	12797
25600	25578
51200	51192
102400	102453
204800	204846
409600	409620
819200	819233
1638400	1638273
3276800	3276834
1638400	1638273

Table: **Statistics of Group's views**

Differential Privacy vs. Non-DP

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

[1] Dwork, C. Differential Privacy, 33rd International Colloquium on Automata, Languages and Programming, part II, 2006

[2] Patel, A., Sharma N., Eirinaki M., Negative Database for Data Security, Proceedings of the 2009 International Conference on Computing, Engineering and Information, p.67-70, April 02-04, 2009

[3] Dwork, C, The Promise of Differential Privacy: A Tutorial on Algorithmic Techniques, Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, p.1-2, October 22-25, 2011 [doi>10.1109/FOCS.2011.88]

[4] Differential Privacy. Retrieved May 17, 2017 from https://en.wikipedia.org/wiki/Differential_privacy