

Project Thor

Team Members:

Adonay Pichardo

Jared Blanco

Josh Temel

Luke Boneburger

Faculty Advisor:

Dr. Sid Bhattacharyya

Client:

Dr. Amitabh Nag

[Google Slides](#)

Milestone 3 Task Matrix

Task	Adonay	Jared	Josh	Luke
1. Update demos	25%	25%	25%	25%
2. Update Documentation	Read & Review	Read & Review	100%	Read & Review
3. Add Content to the Web App (About, Generate Key, Learn more)	Read & Review	Read & Review	75%	25%
4. Full functionality to Generate Key button (Strike info, md5)	50%	Offer help / troubleshoot	Offer help / troubleshoot	50%
5. Fix Webhook bug	Offer help / troubleshoot	Offer help / troubleshoot	100%	Offer help / troubleshoot
6. Create website domain name	Offer help / troubleshoot	Offer help / troubleshoot	Offer help / troubleshoot	100%
7. Create LinkedIn Profiles and link to Web App Team page	25%	25%	25%	25%

Milestone 3 Task Matrix

8. Automation that stores all generated numbers in database	50%	50%	Offer help / troubleshoot	Offer help / troubleshoot
9. Create documentation explaining the generation of key	Offer help / troubleshoot	50%	50%	Offer help / troubleshoot
10. Generate key from database, insert key into database, MD5 hash, display MD5 hash on website	50%	Offer help / troubleshoot	Offer help / troubleshoot	50%

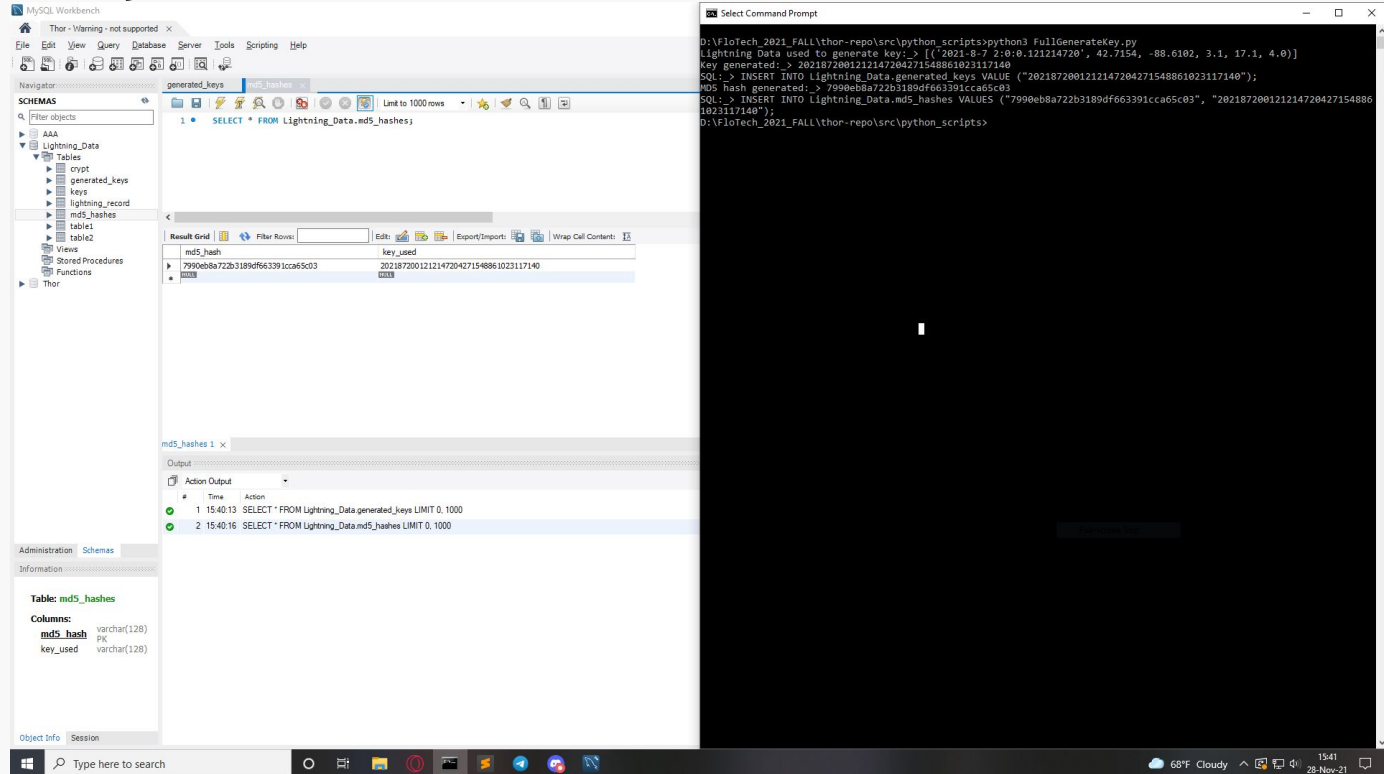
Demos

1. [Live Web Application](#)

2. Full Key Generation

3. Current Data Entropy

Demo 2: Full Key Generation



The screenshot displays the MySQL Workbench interface on the left and a Windows Command Prompt on the right, illustrating the process of generating full keys for a database.

MySQL Workbench Interface:

- Navigator:** Shows the database structure with schemas like AAA, Lightning_Data, and Thor. The **Lightning_Data** schema is expanded, showing tables like **md5_hashes**.
- Query Editor:** Contains the SQL query: `SELECT * FROM Lightning_Data.md5_hashes;`
- Result Grid:** Displays the output of the query, showing columns **md5_hash** and **key_used**. The data includes a long MD5 hash and a corresponding key used.
- Table: md5_hashes:** Shows the table structure with columns **md5_hash** (varchar(128)) and **key_used** (varchar(128)).

Command Prompt Output:

```
D:\FlotTech_2021_FALL\thor-repo\src\python_scripts\python3 FullGenerateKey.py
Lightning Data used to generate key-> [{"2021-8-7 2:0:0.121214720", 42.7154, -88.6102, 3.1, 17.1, 4.0}]
Key generated: > 2021872001212147204271548861023117140
SQL-> INSERT INTO Lightning_Data.generated_keys VALUE ("2021872001212147204271548861023117140");
MD5 hash generated: > 7990eb8a722b3189df663391cca65c03
SQL-> INSERT INTO Lightning_Data.md5_hashes VALUES ("7990eb8a722b3189df663391cca65c03", "2021872001212147204271548861023117140");
D:\FlotTech_2021_FALL\thor-repo\src\python_scripts>
```

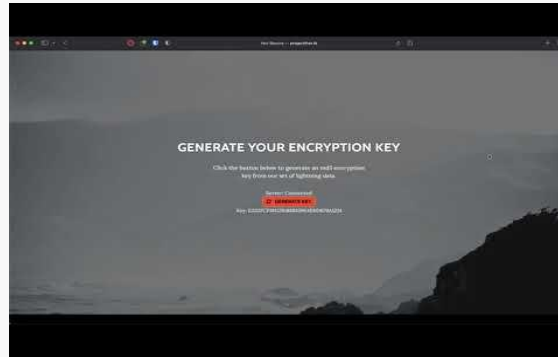
The Command Prompt shows the execution of a Python script that generates a key based on Lightning Data, inserts it into the **generated_keys** table, and then inserts the corresponding MD5 hash into the **md5_hashes** table.

Current Technical Challenges

- Removing lightning data used to generate a key so as to avoid regenerating the same key.
- Measuring entropy of dataset
 - Understanding & Selecting Dieharder tests for our data set
 - Exporting our data in a format acceptable to the test suite
- Creating interactive features of website for key attributes
- Fixing Webhooks bug
- Displaying data used for our encryption so it is visually engaging to the user

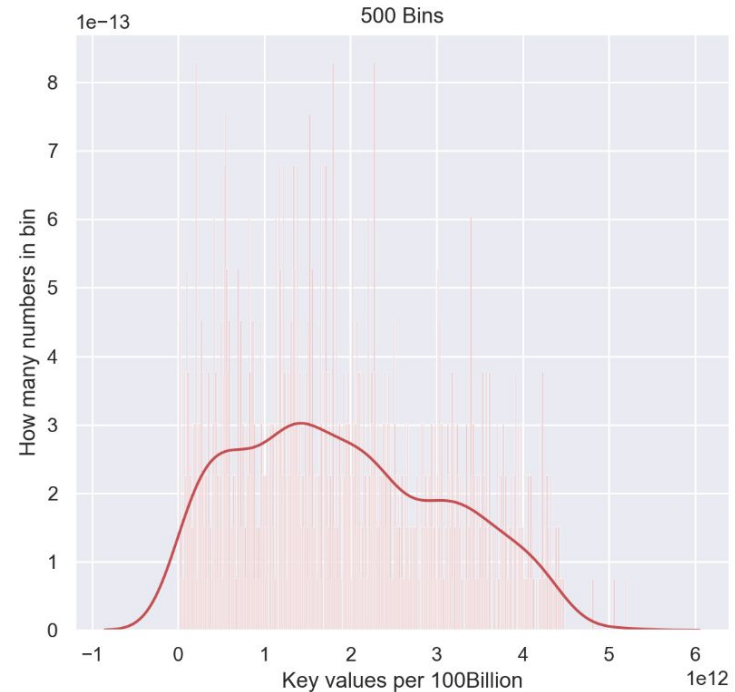
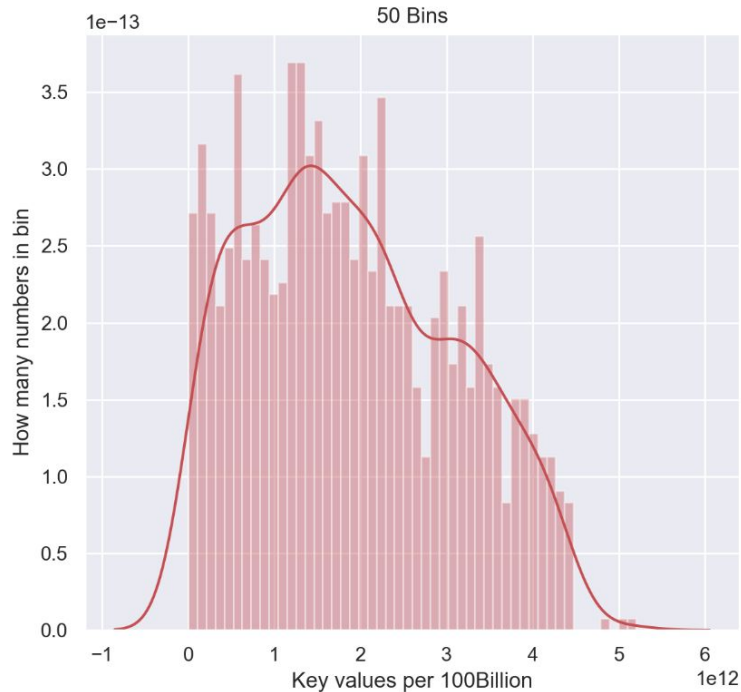
Demo 1: Live Web Application

1. Live Web Application



Demo 1 video

Demo 3- Milestone 2 vs Milestone 3



Demo 3: Data Analysis

Problems To Solve:

- Need more data
- Importing data file
- Understand test results

```
parallels@ubuntu-linux-20-04-desktop:~$ dieharder -a -g 201 -f binaryNum.bin
#####
#               dieharder version 3.31.1 Copyright 2003 Robert G. Brown               #
#####
# rng_name | filename | rands/second |
# file_input_raw | binaryNum.bin | 7.43e+07 |
#####
# test_name | ntup | tsamples | psamples | p-value | Assessment |
#####
# The file file_input_raw was rewound 11 times
# diehard_birthdays | 0 | 100 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 91 times
# diehard_operm5 | 0 | 1000000 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 195 times
# diehard_rank_32x32 | 0 | 40000 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 243 times
# diehard_rank_6x8 | 0 | 100000 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 265 times
# diehard_bitstream | 0 | 2097152 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 434 times
# diehard_opso | 0 | 2097152 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 547 times
# diehard_oqso | 0 | 2097152 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 600 times
# diehard_dna | 0 | 2097152 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 605 times
# diehard_count_1s_str | 0 | 256000 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 709 times
# diehard_count_1s_byt | 0 | 256000 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 711 times
# diehard_parking_lot | 0 | 12000 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 712 times
# diehard_2dsphere | 2 | 8000 | 100 | 0.00000000 | FAILED
# The file file_input_raw was rewound 713 times
# diehard_3dsphere | 3 | 4000 | 100 | 0.00000000 | FAILED
^C
parallels@ubuntu-linux-20-04-desktop:~$
```

Questions