Distributed Application to Provably Log User Activity Centralized Databases

Mentor: Dr. Sandeep Shukla
By: Shubham Sharma and Rahul Gupta
Department of Computer Science and Engineering
IIT Kanpur

Outline

Abstract

- Normally several users can have read/write access on conventional database systems.
- If someone deletes database logs, there is no way to figure out who made changes to the database.
- This means that database is open to ”insider threats”
- Also, normally it is not possible to trace damage if database is modified in a malicious event.
- This is the problem of non-repudiation.

Problems with current system

OARS SYSTEM

Professor	Course	Grade
1	ESC101	A*
2	CS698	A
3	CS425	B
4	CS771	A

Problems with current system

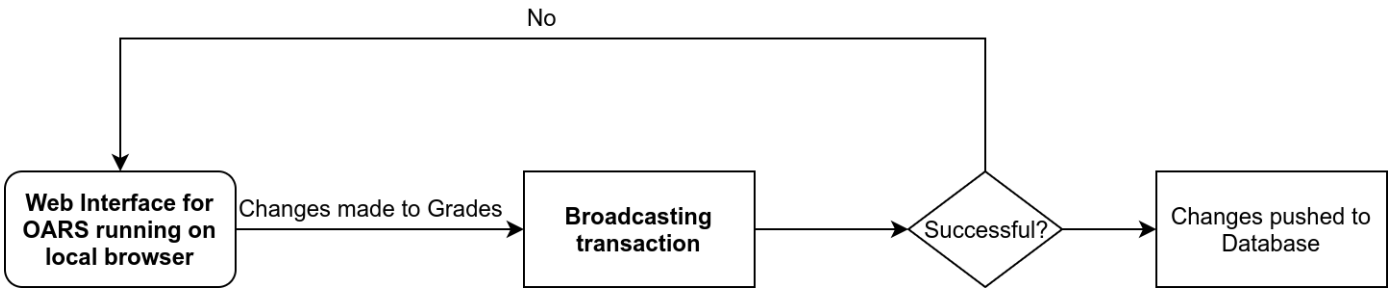
Professor	Course	Grade
1	ESC101	A*
2	CS698	A
3	CS425	B
4	CS771	A


```
UPDATE student_grades
SET ESC101=D,...
WHERE
student=some_value
```

Professor	Course	Grade
1	ESC101	D
2	CS698	D
3	CS425	D
4	CS771	D

Logs Removed

Brief Control Flow of our Solution



- We log the state of data by signing it with the authorized person's private key who made change leading to current state.
- We employ the block chain technology which allows for cryptographic signature schemes, and 'append-only' logging mechanism.
- Changes made on this permissioned and distributed block chain (multichain stream) are immutable and visible to everyone.

MySQL

- MySQL database is hosted on centralized server, each user is given username and password through which they can authenticate themselves and connect to the server.
- MySQL views and privileges are used to maintain the confidentiality of data.
- SELECT privilege is given to students on their respective view.
- SELECT, UPDATE, INSERT privileges are given to professor on their respective view.

Multichain

- Multichain streams provide a natural abstraction to blockchain use cases which can be used as a key-value database in a NoSQL style in which entries are classified according to their authors.
- Multichain stream viz logstream is used to store the logs of MySQL which is signed by the professor making that change.
- The (CourseID, StudentID, ProfessorID) is used as a key and the hash of the change is used as a value.
- Hashing is used to maintain the confidentiality of the data which is present on the multichain stream in distributed manner.

Django

- User Client is hosted on localhost using Django on Python. This enables us to get graphical functionality up and running quickly.
- Created views for students and professors to view and change grades respectively.
- Python communicates with multichain stream using JSON RPC calls provided by Savoir library.
- A seperate Django based server runs to grant write permission to professors on blockchain stream using their username and passwords.

Explaining the solution

- Non-repudiability is solved because final state of database corresponding to any change made by the professor is logged on the stream signed by his private key.
- Can't Adversary just change the grade and log nothing on multichain? No, because then the final state on chain wouldn't match with database entries.
- Who maintains the private and public keys? In our solution, we are simply storing the public keys against professors id on database. However, a public key infrastructure would be required to verify professor's public key. This aspect needs to be taken care of during deployment.

Demonstration

Shortcomings

- Since only 6 grades can be assigned to a particular course, confidentiality of data can be breached by using brute force method. Salting can be used to prevent this breach in which salt is separately stored on stream encrypted with student's public key.
- For a large number of students, syncing changes on stream takes time. Further as time increases, log size increases which would mean more disk space is needed for storing the changes.
- Current multichain implementation mines blocks continuously irrespective of any new transactions, this leads to unnecessary increase in size of blockchain.
- If one node turns rogue it can publish garbage data on blockchain since smart contracts are not available in asset-less multichain streams. This scenario can be detected by running a central validation server.

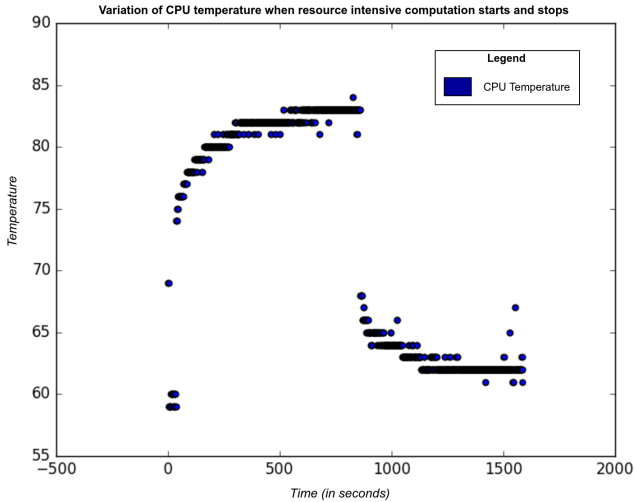
Future Work

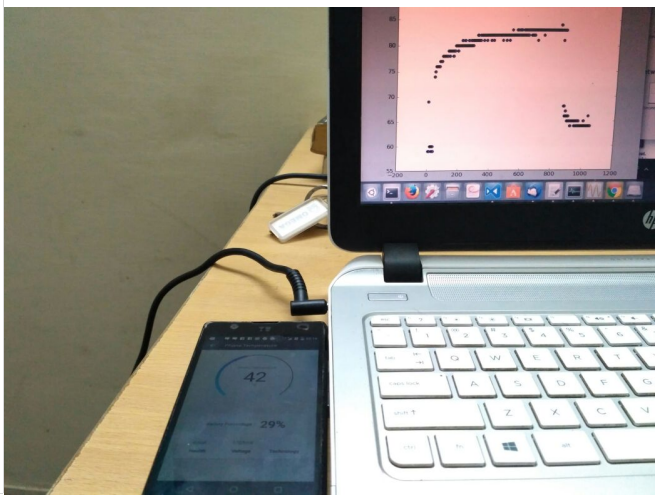
- More enhanced and robust Public Key Infrastructure.
- Auto recovery server for recovery from maliciously changed records in an attack. This would be possible because a student can always report to OARS with his salt and message digest. So, brute force over grades can be done to find out the last consistent grade with database.

Containerisesd Web Browser

- Docker container image containing firefox browser is created.
- The image uses X11 server and Pulse-audio unix domain sockets on the host to enable audio/video support in the web browsers.
- The X11 socket is used to display the user interface the host, while the pulse-audio socket is used to render the audio output on the host.
- This dockerised browser can be set as a default browser so that any hyperlinks opened from emails, etc open within container.

Thermal Eavesdropping





- We observed that it takes about 5 minutes for the temperature to rise on mobile while the temperature got down in 10 minutes. This suggests a bit rate of about 4 bits/hr using manchester encoding.

Key Learnings

- Learned to make simple apps which utilise the power and security of distributed ledgers like bitcoin.
- Learned to maintain and host a central MySQL server for working with many clients.
- Learned to assign MySQL privileges to different users.
- Learned the concept of using public private cryptographic keys to enhance security in real world problems.
- Learned to use docker and run GUI applications inside docker container.
- Learned techniques of eavesdropping(Thermal, Acoustic and Electromagnetic).

References

- https://www.multichain.com/
- https://www.coursera.org/learn/cryptocurrency
- https://www.djangoproject.com/
- https://www.mysql.com/
- https://www.docker.com/

Links to Project Work

- https://github.com/smsharma1/Logging-Dapp
  Distributed Application to Log Activity on Database
- https://github.com/smsharma1/Secure-Browser
  Containerised Browser using Docker
- https://github.com/rahulguptakota/thermal_eavesdropping
  Thermal_Eavesdropping

Thank You