Distributed Application to Provably Log User Activity on Centralized Databases

Shubham Sharma[1] and Rahul Gupta

Indian Institute of Technology, Kanpur, India
smsharma@iitk.ac.in, grahul@iitk.ac.in

Abstract. Applications relying on centralized databases are often vulnerable to insider attacks. Any user with root access to the database system or the hosting server, is capable of modifying the database entries. Furthermore, such an user might modify the corresponding log entries making it virtually impossible to detect such an attack. In this paper, we propose a solution to this problem using blockchains. We have implemented and tested our protocol for the grade distribution system, which is robust to such an insider attack.

Keywords: Blockchain, Centralised Database Systems, Repudiation, Immutability, Insider Attack

1 Introduction

Currently, access control in most of the centralized database systems and applications are based on organizational policies allowing privileged (read/write) access on data to several users. As a result, if a privileged user manipulates entries in the database, and removes the corresponding database log entries, it is difficult to attribute any changes to such users. This is an example of "insider threat". It is also possible that through SQL injection or other vulnerabilities in a web application, the database entries might change. The person whose data is changed has no proof that changes have been made or a way to attribute the change to someone. This non-repudiability is not desirable for databases such as student grade database on which the academic performance of a student is certified.

The remainder of this research study is organized as follows: Section 2 provides a brief overview of the security issues in centralized database systems. Section 3 describes why should we use Blockchain? Section 4 describes our proposed solution and architecture to these issues. Section 5 presents some concluding remarks.

2 Security issues in centralized database systems

Currently most of the applications are built on top of relational database management systems(RDBMS) like MySQL for data organization, storage and access. Ensuring confidentiality, availability and integrity of data is crucial and very

important problem in modern internet world. An *Insider attack* is threat to organizations data that comes from within. These threats are usually attributed to privileged users who can abuse legitimate privileges to masquerade as other users or to maliciously inject data. Insider attacks are very difficult to detect and are very costlier to remediate. According to recent survey[1] from all Electronic Crime events (known or suspected) 28% are caused by insider attacks, out of which 46% are more costlier and damaging to the organization.

It is also possible that, vulnerabilities might be present in web applications, and through SQL-Injection or other techniques an outsider can exploit these and change the database entries. These kind of outsiders attack can be mitigated by using defensive programming techniques[2].

All of these attacks permits malicious manipulation of data and forging the identification of actions by manipulating the logs. Hence non-repudiation cannot be ensured.

3 Why Blockchain?

We leverage the block chain technology for cryptographic signature schemes, and 'append-only' logging mechanism. For logging, it is essential that the data structure is immutable and distributed so that all changes are permanently stored and visible to everyone, yet the confidentiality is maintained. We achieve this by utilizing the asymmetric cryptography scheme and using distributed and permissioned blockchain viz. multichain as storehouse.

4 Proposed Solution on a Case Study of Grade Distribution

Consider a scenario in an academic institution where we have to distribute grades to the students. There are several courses hosted by institution in each semester. Each course is taught by one or more professors in which multiple students are enrolled. When using a traditional centralized SQL based storage, we have the following problem.

A privileged user(professor or admin) can make unjustified changes to some student's grade and delete any logs if present. Now, the student can observe the grade change but cannot find out who made that change. He can possibly complain to the administration but even they cannot find out the real culprit since several possible entities holding the privilege to change grades could have done that. Or even worse, a malicious event (SQL injection or some other attack) could have led to the changes on database.

We propose a system where the database is centralized as usual in institution with privileged access given to limited professors for their respective courses as per the principal of least privileges. But now the logs are explicitly maintained over distributed blockchain. In our example, we have created a publicly readable blockchain. The write permissions on blockchain are given using Professor's address taken from a reliable PKI. For our example, we created a client application

which serves the pages and supports required functionality. To make changes to the grades on central database, it first connects to the blockchain to append the change log and then pushes the changes on central database. Students also run a slightly different version of application which supports viewing their respective grades and verification using the logs available on public blockchain.

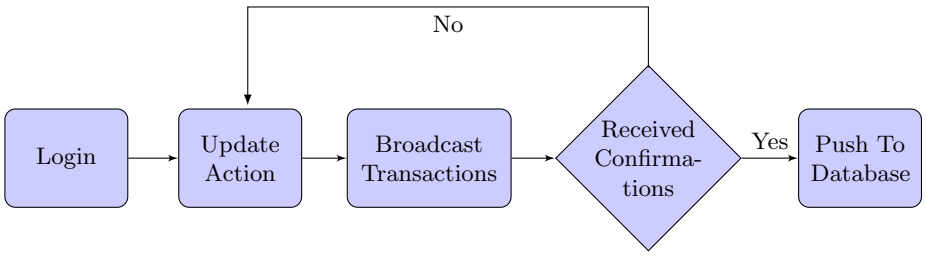

Fig. 1. Flowchart of our DApp

We will dive deeper into our solution now. The professors are expected to run the client application, particularly, the blockchain forever to ensure that there are enough miners to ensure security of blockchain. The students can just connect to the blockchain whenever they wish to look at their grades and verify them from blockchain. They are not expected to mine any blocks. We will now describe how data is stored on blockchain. Whenever professor changes/inserts grade for a student the application digitally signs the hash of resulting final state of the database about the concerned student with a random salt using the private key of professor and puts that signature on the blockchain. The information about salt is also encrypted using student's public key (on a seperate stream or database itself?) and it is then that change would be propogated to the database using our application. When the concerned user checks his data on the database, he/she would verify the signature of data with actual data on the database. This way, he/she would be sure that:

- The change is made by which privileged user.
- The integrity of the change is maintained.

If any of this is violated, the student would find it out immediately when he sees the grades due to verification and would be able to report the incident to the admin of database or the blockchain. Even the client application can automatically send notifications to the concerned authorities. Also, if both properties are true, the student would know who made the changes and can then contact the respective person or some other authority in the organisation to blame the person for making unjustified changes to his data.

5 Conclusion

In this paper we have presented an approach based on blockchain to detect an insider attack or any unauthorized modification of the database entries. The system presented has a low storage overhead, as in addition to the usual data, it also stores a hash value and a public key in each row. It will also involve some latency, as instead of directly modifying the entries of the database, the application waits for some required number of confirmations on the blockchain corresponding to that query.

References

1. 2014 State of Cybercrime Survey Presentation. https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=298318
2. Owasp-sql injection prevention cheat sheet (2008)
3. https://www.multichain.com/download/MultiChain-White-Paper.pdf
4. Satoshi Nakamoto : Bitcoin: A Peer-to-Peer Electronic Cash System https://bitcoin.org/bitcoin.pdf