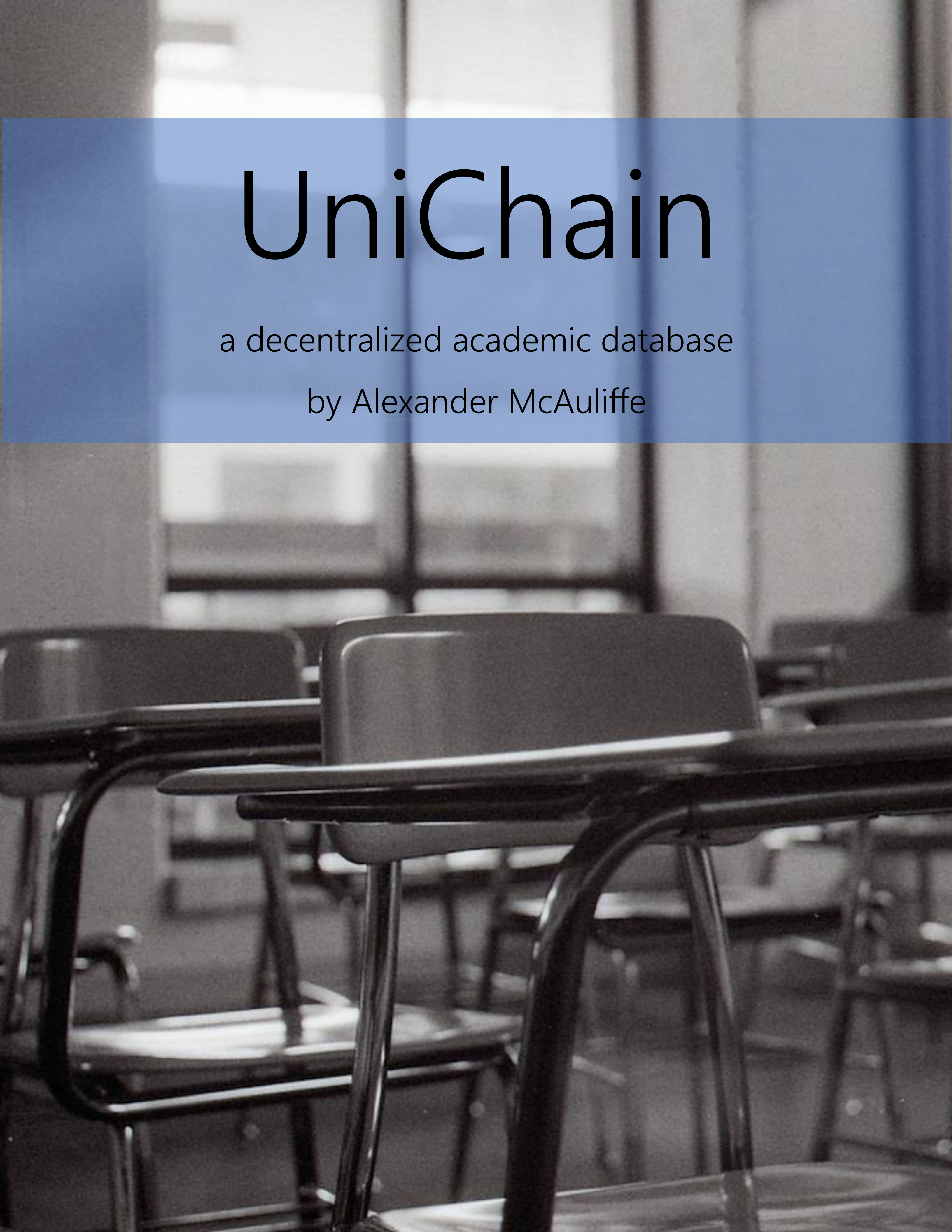
UniChain

a decentralized academic database

by Alexander McAuliffe



INTRODUCTION

The educational sector is in the midst of a cybersecurity crisis. Servers are becoming playgrounds for cyber-espionage and attacks on personal identifying information. Not only has this been a problem ever since educational institutions moved to digital data storage, but the number of incidents per year is also trending *upwards* in recent years. From 2005-2014, 727 recorded data breaches exposed over 14 million individual educational records[1]. Furthermore, 455 incidents with 73 confirmed data disclosures have occurred in the first half of 2017 alone[2]. It is clear from this data that university students face the exposure of their personal information and the grades which they legally have a right to retain every day. Higher education is vulnerable to these attacks due to its lack of innovation in cybersecurity and its outdated methods of data storage. It is time for the education sector to be secured in the same way that Bitcoin revolutionized the financial industry. Rather than storing data on a centralized server with insecure and vulnerable hard drives, the data must be decentralized and stored on the machines of the people who want to access it. Secondly, this data needs to be invulnerable to hackers hoping to change records in their favor. Finally, this data must remain confidential to those who are provided to access the personal, private information of students and faculty. Enter UniChain, a distrusted database system which utilizes the same blockchain technology as Bitcoin to store data on thousands of students' computers while still keeping it completely unchangeable, historical, and confidential.

[1] – Grama, "Just in Time Research: Data Breaches in Higher Education." – pg. 1
[2] – Dingan, "Ransomware incidents surge, education a hot bed for data breaches, according to Verizon."

THE PROBLEM

Through the incredibly high number of data breaches and security incidents reported in higher education, it is clear that the educational sector does not share the same level of innovative security and technological advancements as other industries. Not only is compromised data extremely costly for universities to recover—now the average cost has reached $245 *per breached record*[3]—it is also particularly revealing in an educational context, as this data includes students' personal information such as phone numbers, home addresses, and social security numbers in addition to their academic records. Furthermore, 35% of *all* recorded security breaches take place in higher education[4]. That figure alone is significant enough to take a serious look at the methods schools use to store student data and deduce why these systems are vulnerable to attack.

Firstly, there is a huge motivation for cyber-espionage in the education sector. Academic databases are big targets for a number of reasons. Firstly, the nature of the records being stored at universities are of particular interest to attackers. Obviously, the personal information of students at faculty is at risk, but beyond that, valuable confidential research is conducted at universities which attackers would likely target. Secondly, the databases at universities are often massive. There's no telling exactly how much data is compromised when a breach occurs, and with such a big target at play, there is tons of information to potentially steal. Essentially, universities store large amounts of confidential and valuable information, which makes them vulnerable to hacker targeting.

Secondly, universities use outdated methods of data storage and access which are particularly vulnerable to attack. An overreliance on local, physical storage mediums has led to problems. For example, Washington State University had to send an email to over one million people explaining that their personal data had been compromised, simply because someone had broken into a safe containing a backup hard drive[5]. The central servers run by universities are also very vulnerable to denial-of-service (DoS) attacks. These attacks, while not damaging to data, can prevent faculty and students from accessing or adding information to databases, which can be damaging to protocols over a prolonged period of time, and require significant IT manpower to thwart when they are manifested.

Evidently, action needs to be taken to prevent these kinds of attacks from having an effect. Universities are big targets for good reason, and the outdated security and data storage methods currently employed by universities are not able to resist attacks in any meaningful way.

3 – Schaffhauser, "Average Cost Per Record of US Data Breach in Ed: $245."
4 – Barker, "35 Percent of All Security Breaches Take Place in Higher Education."
5 – Long, "Did you get the letter? WSU sends warning to 1 million people after hard drive with personal info is stolen."

UNICHAIN – THE SOLUTION

While many measures could be taken to increase the relative security of centralized educational databases, attackers can and will take additional steps to outpace those developments. The solution is not to spend money and time overdeveloping a dying, inefficient model for data storage and retrieval, but rather to come up with a better one that replaces its shortcomings. It is necessary to abandon the concept of central access points and data storage, like the financial industry is slowly doing with cryptocurrency and Bitcoin. UniChain is a decentralized, distributed academic record database that accomplishes those goals.

The primary way in which UniChain achieves these feats is by distributing the record database to all students and faculty. There are several advantages to this system, firstly that there is absolutely no reliance on a central server to access information. All data is accessed from a local copy of the database, updated from other clients running the software. This makes denial-of-service attacks impossible, since there is no central server to target. The second advantage is that there is no chance for data loss through any means. Since there are so many copies of the information, anything but an extremely minor loss of recent information is implausible. The data itself is also unable to be changed, only overridden, meaning that no one can hack into the database and change records that already exist. If a hacker were to attempt to overwrite any information, as soon as their access is removed, one could look into the historical data and determine what information was correct before the attack and restore it. Finally, secure cryptographic methods make data breaches impossible by "hiding in plain sight". Though the blockchain is inherently public (since it is distributed to so many users), the users control keys which gives them unique access to their data. Without that key, no one can access that data. This means that data breaches can only occur by a user mishandling of keys—and even then, only a small subset of data is compromised.

There are countless other advantages to this system as well. Since all personal information is protected and only accessible by those with the keys, students have greater control over their information and subsequently more privacy. Secondly, this system severely cuts costs for universities who incorporate it. The system maintains itself, and requires no additional cost to host a constantly online server. Running webpages and servers to access and save information is extremely costly, but a system like this only requires the initial setup, and then is completely self-maintaining. Data redundancy is also ensured in a system like this. Rather than having to take regular backups and archives of information, so many copies of the database exist already that as long as one copy exists, the database is completely safe. Storing data in a central location is completely unnecessary; the data is stored on every user's personal hard drive, creating an indestructible network. In addition, faculty participate in consensus operations, ensuring that every client has the same, authoritative copy of the database.

It is clear that a distributed database is superior to a centralized server filled with information. There are so many advantages to this self-maintaining system that the only remaining question is how to implement it.

METHODOLOGY

Blockchain Fundamentals

It is first necessary to understand the fundamentals of a blockchain database. In a blockchain, users who have a copy can see and make transactions. These transactions are compiled by users into "blocks", which are simply groups of transactions. These blocks build on top of each other in a linear fashion, with each block including the "hash" summary of the previous block's contents. If someone were to change the data inside a single block, the hash summary of that block would subsequently change, meaning that the summary of the next block would change, so on and so forth. This means that the person changing that data would have to rebuild the entire chain, a feat that becomes infeasible with the round-robin mining scheme described in detail in the Consensus Model section. Each transaction is also timestamped, creating an unchangeable, historical database of which there are thousands of copies.

Client Communication

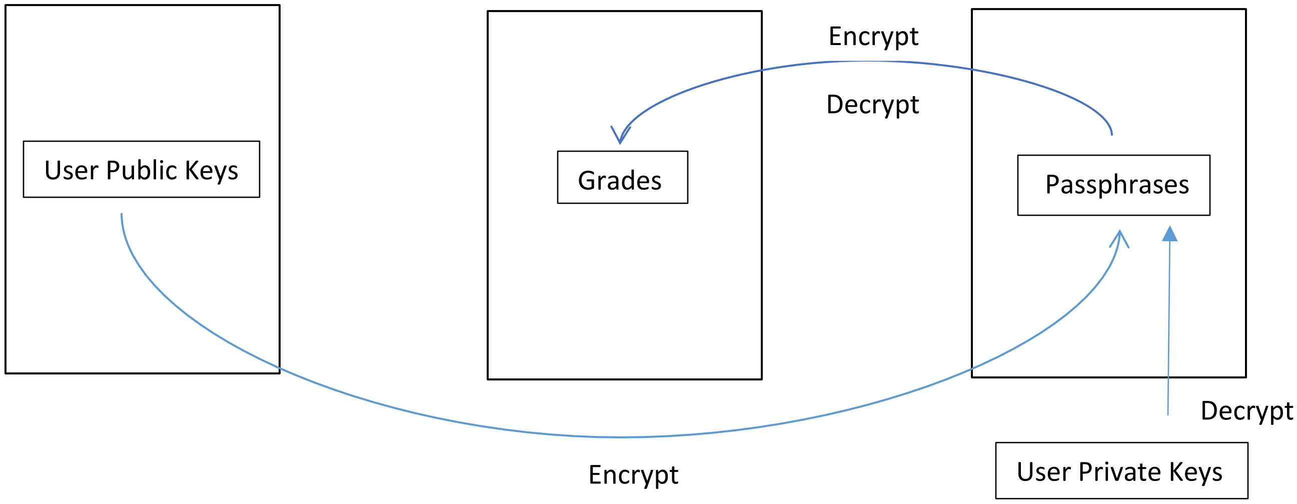
Every client running this software are able to "see" each other and their copy of the database. When this occurs, the clients check the length (size) of their respective copies of the blockchain. The longer (bigger) chain is always accepted as the authoritative copy. If a client determines that a different client has a longer chain whose blocks abide by all the rules, it copies that chain and makes it its own. Clients are then able to go share this data with other clients as well.

Consensus Model

Whenever a new block of data is generated, the faculty clients must approve that it follows all the rules before it is able to be added to the blockchain. To ensure that every user is on the same page in terms of the blockchain's rules and the kind of data they will accept, the same faculty or small group of faculty cannot approve multiple blocks within a certain time period. What this entails is that a large portion of the faculty are agreeing on the kind of data which is allowed to enter the database. Since multiple faculty are required to "mine" blocks in this way, it is definite that the longest chain is the one with the most total faculty approval at any given moment in time. Therefore, it is logical that clients should accept the longest chain as authoritative. If one or a small group of faculty members were to start mining incorrect data by accident or otherwise, this round-robin mining style would prevent them from doing so since the majority of faculty are approving a different kind of data. The chain would "split" as it were, with the small offshoot mining different data. However, since the majority of faculty are mining the correct data, their chain will progress faster, and since clients accept the longest chain, the attackers will be ignored.

Confidentiality

The confidentiality model is the most important part of a database like this, and as such it is meticulously designed to be as secure as possible. When a student starts the client for the first time, they will be registered in the database. A set of two keys is generated for each student—one private, one public. The public key is published in the database, while the private key is to remain appropriately private. When a faculty adds a grade for a given student and course, a random passphrase is generated. The grade is then encrypted with that passphrase. Then, the passphrase itself is published to a different area of the database, itself encrypted by the public key of the relevant user. Here's the trick: anyone can encrypt something with a public key, but only the user with the corresponding private key can decrypt it. This means that grades only have to be entered once, and can be encrypted by anyone, yet only decrypted by the owner of the private key. The secret passphrase can be published multiple times encrypted by different users' public keys according to who needs access. In this way, the grades need only be stored once, but can be made accessible independently by multiple people by using their personal keys to encrypt the grade's password.



COUNTERARGUMENTS

Q. Isn't making the database public akin to making all the data inside of it public? Isn't that just creating a massive entry point for hackers?

A. While the database itself is public, the data inside of it is even more secure than the data in a standard centralized database. The strong cryptographic methods described in the Confidentiality section are virtually uncrackable, and users have full control over the keys to access the data. It's like a owning a house full of valuables inside; the house is public, and everyone can see it, but only one person has the keys to get in.

Q. Wouldn't it be inconvenient for students to store an entire copy of the database on their computers?

A. While students are required to store the entire database in order for the consensus model to function, it would be expected of the university to provide storage to make this possible. With the costs saved from running a constantly-maintained server, the administration could purchase, for example, thumb drives for students to store the database on at a price that would likely still undercut the costs of a centralized server. The required storage is also likely to grow at a rate that is close to the natural cheapening of storage over time.

Q. Couldn't someone steal a key and have access to the entire database?

A. Key theft in this system is actually much less damaging than password theft in a standard system. Passwords can be changed, while keys cannot, meaning that if a user kept a backup key, an attacker cannot take over that user's access. Secondly, a single key only gives an attacker the ability to see a small portion of the database. Even if they managed to get their hands on a faculty key, data can only be overwritten, not changed, meaning that any malicious changes can be easily reversed by simply overwriting them again.

Q. Have universities not already implemented similar systems?

A. The only widely-implemented software similar to this is MIT's BlockCerts, which stores academic certificates such as degrees in a blockchain in order to timestamp and verify them as well as submit them to employers. However, this system does not record the grades which may have led to those accomplishments; UniChain deals with the details on a university level, solving practical issues such as data breaches and loss. The two softwares deal with fundamentally different aspects of educational data.

Works Cited

Barker, Ian. “35 Percent of All Security Breaches Take Place in Higher Education.” *BetaNews*, 17 Dec. 2014, www.betanews.com/2014/12/17/35-percent-of-all-security-breaches-take-place-in-higher-education/. Accessed 26 Oct. 2017.

Dignan, Larry. “Ransomware Incidents Surge, Education a Hot Bed for Data Breaches, According to Verizon.” *ZDNet*, 27 Apr. 2017, www.zdnet.com/article/ransomware-incidents-surge-education-a-hot-bed-for-data-breaches-according-to-verizon/. Accessed 26 Oct. 2017.

Grama, Joanna. “Just in Time Research: Data Breaches in Higher Education.” *EDUCAUSE Center for Analysis and Research*, 2014, www.educause.edu/ir/library/pdf/ecp1402.pdf. Accessed 26 Oct. 2017.

Long, Katherine. “Did you get the letter? WSU sends warning to 1 million people after hard drive with personal info is stolen.” *The Seattle Times*, 22 June 2017, www.seattletimes.com/seattle-news/education/did-you-get-letter-wsu-sends-warning-to-1-million-people-after-hard-drive-with-personal-info-is-stolen/. Accessed 26 Oct. 2017.

Schaffhauser, Dian. “Average Cost Per Record of US Data Breach in Ed: $245.” *THE Journal*, 18 July 2017, thejournal.com/articles/2017/07/18/average-cost-per-record-of-us-data-breach-in-ed-245.aspx. Accessed 26 Oct. 2017.