AegisBlock: A Privacy-Preserving Medical Research Framework using Blockchain

Calkin Garg*
Dept. of Computer Science
Georgia Institute of Technology
cgarg35@gatech.edu

Omar Rios Cruz*
Dept. of Computer Science
California State University, Stanislaus
orioscruz@csustan.edu

Tessa E. Andersen
Dept. of Computer Science
Brigham Young University
tessa343@byu.edu

Gaby G. Dagher
Dept. of Computer Science
Boise State University
gabydagher@boisestate.edu

Donald Winiecki
Dept. of OPWL
Boise State University
dwiniecki@boisestate.edu

Min Long
Dept. of Computer Science
Boise State University
minlong@boisestate.edu

***Abstract*—Due to HIPAA and other privacy regulations, it is imperative to maintain patient privacy while conducting research on patient health records. In this paper, we propose AegisBlock, a patient-centric access controlled framework to share medical records with researchers such that the anonymity of the patient is maintained while ensuring the trustworthiness of the data provided to researchers. AegisBlock allows for patients to provide access to their medical data, verified by miners. A researcher submits a time-based range query to request access to records from a certain patient, and upon patient approval, access will be granted. Our experimental evaluation results show that AegisBlock is scalable with respect to the number of patients and hospitals in the system, and efficient with up to 50% of malicious miners.**

***Index Terms*—Blockchain; Medical Research; Data Sharing;**

I. INTRODUCTION

Across all aspects of research, there are often complications related to the access of data. This is especially the case when the data comes from "human subjects" [1]. Protection of privacy related to personal health information (PHI) is not only ethically virtuous and a common good when accomplished for all people; it is also mandated in the USA with HIPAA law [2]. HIPAA was explicitly intended as a strong precedent for handling patient privacy and security to prevent mismanagement of data. HIPAA provides patients with the confidence that their PHI can only be used and shared with individuals who must have it to assist with their medical care.

HIPAA also allows patients to expand or limit access to components of their PHI to groups such as researchers who might find those data valuable in aggregate [3]. This means that even if researchers obtain permission to access data by complying with HIPAA regulations, the representativeness of data for demographic groups may be reduced as a result of selection bias when patients decline to share their PHI with researchers [4]. We focus on observational researchers conducting cohort studies that discern the change of multiple patients with similar characteristics over a set period of time ranging from a few years to decades [5].

*These authors contributed equally.

These aforementioned HIPAA guidelines point to a dilemma for medical researchers, who must maintain the ethical necessity of these protections while still seeking large sets of data that are representative of populations of interest [6]. This dilemma impacts the utility of data for researchers and may also have an adverse impact on the validity of research, which could have a follow-up negative influence on the value of applications of their research [7].

A clear solution to this problem would allow data from a patient's personal health record (PHR) to be available in an environment that maintains HIPAA regulations and follows ethical values in a safe and secure manner. For our purposes, we consider "safe and secure" to mean a system where only patients, hospitals, and researchers are allowed access to only portions of the data they require for their specific purposes. In order to preserve patient privacy through anonymity, we will require multiple methods of encryption and verification while also masking the identities of individuals. However, this system would only work when patients have previously agreed to share their data with researchers in addition to the necessary medical providers. One technology that can assist with the delivery of data while also providing measures to make sure only authorized individuals have access is a Permissioned Blockchain [8]. Blockchain has emerged as a viable solution for managing access to patient data. For one, the system requires consensus before data can be shared [9]. Blockchain is also an immutable and decentralized technology that incorporates various protocols to facilitate transactions [10], which creates a safe system in which patients can input their data and researchers can request it with the consent of the appropriate patient.

In this paper, we utilize blockchain technology to introduce AegisBlock, patient-focused system for sharing data with researchers. AegisBlock behaves as an *access control system*, allowing patients to have full authority to decide who has access to their data, while also allowing researchers to submit time-based range queries to support observational cohort studies. AegisBlock is a permissioned blockchain, where we

assume all users in our system are a valid member of their respective party, validated by a trusted third party. We rely on the immutability property of blockchain to allow researchers to verify the integrity and accuracy of the information they receive.

A. Contribution

The contributions of this paper are as follows:

1) We propose AegisBlock, a proof-of-concept patient-centric permissioned blockchain framework which serves as an access mechanism for sharing medical data with researchers.
2) We incorporate methods to reference patients' previous blocks while still protecting patients' privacy.
3) AegisBlock facilitates researchers' requests for patients' personal health records to conduct cohort studies, while ensuring the trustworthiness of the shared data.
4) We conducted experiment evaluation on AegisBlock. The results demonstrate its scalability with respect to the number of users (patients, hospitals, and researchers) in the system. They also show that AegisBlock is robust against up to 50% of malicious miners (hospitals).

II. Related Work

Prior research has attempted to address the delivery of data from a patient to a vast multitude of parties. However, a majority of these papers only establish a "group" as the end user that collects data from patients.

A. Patient-Centric Blockchain System

Roehrs et al. proposed a system titled OmniPHR [11], which handles PHR with respect to data maintenance. Similarly to OmniPHR, Dubovitskaya et al. also proposed their own system, ACTION-EHR [12]. What separates ACTION-EHR from their related work is the use of smart contracts through Hyperledger Fabric. ACTION-EHR also mentions the use PBFT as a consensus mechanism used within permissioned blockchains. In both works, patients benefit from being in their systems by having all their records stored within a secure and private blockchain. Hospitals benefit while in the system because they can collect the same patient's data across multiple visits at different locations, allowing them to perform better evaluations when prescribing care. However, the sharing of patient data among differing medical institutions does raise concerns about anonymity, especially regarding access to patient data from hospitals the patient has never been to. AegisBlock avoids this problem by having the patient choose who their data goes to.

Azaria et al. proposed MedRec [13], a proof-of-work patient-centric blockchain system in which stakeholders can query through a string that lists specific patient data. In MedRec, smart contracts are utilized through medical records and their associated viewing rights. The main purpose of MedRec was to create a system where patients are able to maintain their own data, such as OmniPHR.

Ranaweera et al. proposed a system that revolves around the use of ZKPs to secure patient anonymity [14]. Our system relies on such technologies; however, their usage is merely meant to create a new account on a web application. AegisBlock uses ZKPs to validate patients and hospitals without having to reveal their identities when their blocks are being mined. This way, any block that has already been created can be trusted that all the information inside is valid.

Hylock and Zeng describe a proof-of-concept, patient-centric, blockchain system in which patients, through the use of smart contracts, can share their data [15]. HealthChain utilizes proxy re-encryption, which is a tool we do not need to use, which simplifies our process.

Other papers such as [18], [19], and [20] are all for the sake of maintaining and preserving anonymity with Patient Health Records (PHR). These papers differ with complex smart contract usage, incremental updates, and the ability to query through the blockchain.

B. Patient-Centric Blockchain System with a Focus on Researchers

Cardoso et al. created a proof-of-concept, patient-centric blockchain design that has researchers as the end user [16]. However, they use computationally expensive mechanisms such as Fully Homomorphic Encryption, while also leaving out specific information such as the mining process or key generation. Our work focuses on offering these protocols within a patient-centric, researcher-focused blockchain.

Aldamaeen et al. also created a framework that focuses on researchers, but not as the primary end user [21]. Their implementation of an application along with patients deciding how they can deal with researcher requests does make it similar to our paper. However, after a researcher has been granted access to view a PHR, they will be required to pay. This raises multiple ethical concerns, which we are not comfortable having within our system since we would have to explore the ethics of selling personal medical data, even if the patient provided consent.

Zaghloul et al. had proposed a management system to allow a patient to pick who their information goes to [17], however, the systems do not mention anything about the preservation of anonymity. This system incorporates the smart contracts, split between staff member registration and access verification. AegisBlock preserves anonymity through every step of the blockchain. Furthermore, [17] does not mention how a block is created, what is stored in the block, or who the miners are, all of which are discussed in AegisBlock. Table I provides a comparative evaluation of closely related works.

III. Problem Formulation

We identified many problems within the current world of clinical research, which we attempt to solve with AegisBlock.

First, a researcher would want to identify the group that they want to study. This selection process is usually performed with certain characteristics in mind [22], be it diseases, infections, etc. After identifying a specific group, researchers would

Table I: Comparative evaluation of main features in closely related work including our proposed work AegisBlock.

Papers	Protection of Patient		Researcher Focused	Smart Contracts	Consensus Mechanism
	Data	Anonymity			
Roehrs et al. [11]	✓				
Dubovitskaya et al. [12]	✓	✓	✓	✓	PBFT
Ranaweera et al. [14]	✓	✓			
Hylock and Zeng [15]	✓	✓		✓	PoC
Cardoso et al. [16]	✓		✓		PoC
Zaghloul et al. [17]	✓			✓	
Azaria et al. [13]	✓			✓	PoW
This Paper: AegisBlock	✓	✓	✓		PoC

usually have to go through a hospital. After gathering the consent of the participants, the researcher would proceed to conduct their studies.

Before a researcher attempts to conduct a cohort study, they would first need to go through a hospital's governing body. Researchers would need to obtain approval from an Institutional Review Board, then they would need consent from patients [23]. From this point, the researcher would collect as much data as they need. One of the problems with observational studies is the amount of time it takes. This problem can take up to several years at a time [24]. By allowing a researcher to access data from any range in time, we seek to cut down on the amount of time it takes to conduct research. As we are focused on improving the process for observational studies, we will not discuss the requirement of clinical trials in this paper.

From the perspective of a patient, AegisBlock focuses on giving them full authority when it comes to handling their data. One problem with a cohort study is that one needs to inherently trust that the hospital or other covered entity handling personal data takes the appropriate measures to ensure that it has gone through a sufficient amount of differential privacy or has been de-identified entirely [25]. Another problem patients face is what data they can choose to give away. AegisBlock creates a system in which researchers request data from a patient without accessing the patient's entire health record, similar to how actual studies are conducted. With other studies, there is an agreement to allow active monitoring over a set period of time. AegisBlock creates a system which allows a researcher to access data from a certain time frame, without the need for active monitoring.

A. Notation

Table II shows all the symbols we will use for the rest of the paper.

Symbol	Description
p	Patient
r	Researcher
h	Hospital
b	Block ID
n^b	Nonce
$(x_p^{\text{ID}}, y_p^{\text{ID}})$	Patient's identity key pair
(x_p^b, y_p^b)	Patient's block key pair
s_p^b	Patient-block-specific symmetric key
$(x_h^{\text{ID}}, y_h^{\text{ID}})$	Hospital's identity key pair
(x_h^b, y_h^b)	Hospital's block key pair
y_H	Hospitals' Group Public Key
$(x_r^{\text{ID}}, y_r^{\text{ID}})$	Researcher Identity Key Pair

Table II: Notations

B. Adversarial Model

Within AegisBlock, there are three types of parties that interact with our system. These are *patients*, *hospitals*, and *researchers*. A patient, along with their hospital, collaborates to upload a specific PHR information for a patient onto a block on AegisBlock. The hospitals will also be part of the mining group that validates whether or not a block should be uploaded onto AegisBlock. Since AegisBlock is a permissioned blockchain, a trusted third party is utilized to authenticate patients, hospitals, and researchers. We describe below the possible malicious behavior of each party in the system.

Patients. A malicious patient may attempt to sign or upload a block with their set of keys while being an invalid user.

Hospitals. A malicious hospital may attempt to sign a block with their set of keys while being an invalid user. Additionally, while a hospital is participating in the consensus protocol of a block, they may immediately attempt to decline signing a block, even though it is valid. A malicious hospital may also attempt to upload a patient block without the consent of the patient.

Researchers. A researcher may attempt to use the information the patient provides to access blocks outside the allowed range, or attempt to identify which specific blocks they are receiving data from.

We assume that the number of malicious miners (hospitals) never exceeds 50%. We also assume that there is no collusion between parties in the system.

IV. Solution: AegisBlock

Our proposed system offers a novel framework for access control of medical data between researchers and patients. AegisBlock incorporates a private, permissioned blockchain with ZKPs to ensure patient privacy while giving researchers access to data they need for cohort studies. Within AegisBlock, we include two protocols, with one focused on patients sharing their data on the blockchain and the other allowing researchers access to patient data.

AegisBlock utilizes a private, permissioned blockchain, which means that patients and hospitals will be registered

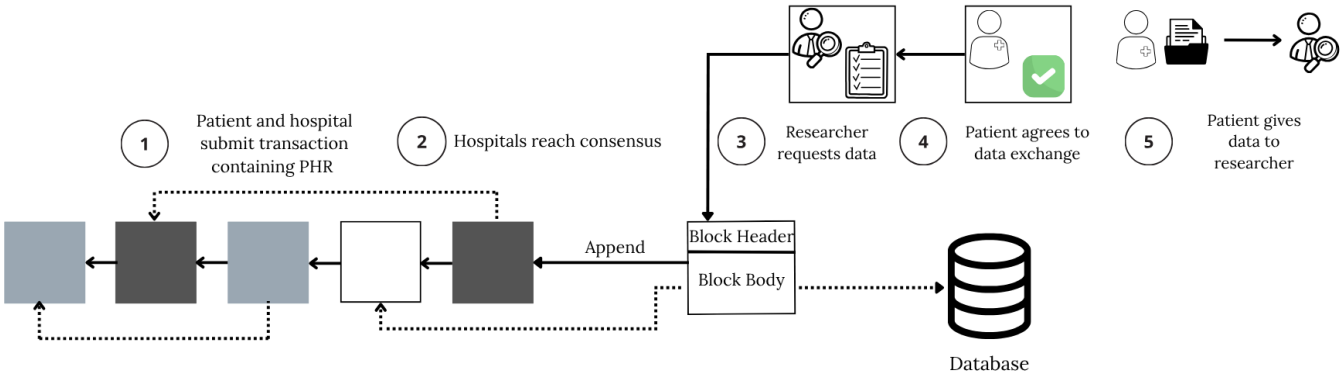


Figure 1: AegisBlock Overview. Each set of blocks with the same color represents different medical records for the same patient.

through a third party that verifies their identities before they are allowed to access AegisBlock. This increases security as only verified users can add medical data to AegisBlock, therefore decreasing malicious attacks where fake medical data is added. In addition to the verification of patients and hospitals, the researchers' identities are also verified before being added to the blockchain. This reduces the chance for malicious users to request access to medical data and increases trust between the researcher and the patients within AegisBlock.

The first protocol is for patient block creation. Patients will submit their PHR onto the chain, which is verified by their corresponding hospital. Other hospitals will act as miners on the blockchain and reach consensus on the data based on the validity of the data, the patient, and the hospital. Our second protocol is used to allow researchers to request and access PHR on the chain for their research. This protocol starts with a researcher requesting access to data within a specified range. The patient will then sign off on this request and give the data to the researcher. Our AegisBlock framework can be seen in Figure 1.

A. Phase I: Patient Health Record (PHR) Submission

The first phase of AegisBlock includes a patient submitting their PHR to be incorporated into the blockchain. A patient and their corresponding hospital will work together to create a block. The block will contain a header and body with differing information. Before submitting a block, the patient will generate a new private/public key pair for the block they are submitting to, $(x_p^b,\ y_p^b)$, along with a new symmetric key, s_p^b, for the same block. The hospital will also create a new private/public key pair, $(x_h^b,\ y_h^b)$, unique to this block. The patient will then encrypt their data using the symmetric key and store it in an off-chain database.

1) PHR Block Creation: Within the PHR Block Creation section, we allow patients and hospitals to submit data on the blockchain for researcher access. After the patient and the related hospital have registered in AegisBlock, the patient can agree to submit their PHR to be incorporated into the blockchain. In this process, both the patient and the hospital will submit data to verify their identity and the integrity of the data. As seen in Figure 2, the header will contain information that is used to verify the authenticity of both parties. The body of the block will contain information that is primarily relevant to a PHR.

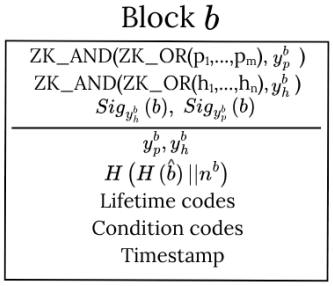


Figure 2: Patient Block Structure

a) Block Body: Protocol 1 details the process for creating a new block. In Steps 1-2, both the patient and hospital create new public keys. In Step 5, the patient will submit a sequence of bits that reveal their lifetime and visit specific conditions. A 0 in a specific position indicates that they do not have that condition, and a 1 indicates they do have the condition. The patient submits the bits in plaintext so that researchers can scan through all the blocks on the blockchain and only fork those of interest to them.

Next, in Step 6, the patient will submit a hash, $H(H(\hat{b})||n^b)$, where $H(\hat{b}) = H(s_p^b\,||\,Ptr(D_b)\,||\,H(D_b)\,||\,H(\hat{b}-1))$. We have the patient submit this hash to prevent researchers from looking beyond within their record range. The inclusion of the nonce, n^b, ensures that it is only necessary to reveal the very last block within a contiguous range of blocks while still allowing the researcher to verify that nothing in the range was tampered with. It also prevents against malicious researchers reconstructing these hashes and attempting to go through AegisBlock to find a match, which would inform them as to where the data they are looking at is coming from. If they

Block Creation Protocol
Input: Patient data recorded this visit: D_b, hash of Patient's most recent block: $H(\hat{b}-1)$
Output: New block, b

The patient, p, constructs a new block for their data D_b, signed by both them and the hospital, h, such that access to encrypted D_b can be granted to interested researchers.

1) p creates a new block specific public key, y_p^b.
2) h creates a new block specific public key, y_h^b.
3) p submits a ZK_AND(ZK_OR(p_1, ..., p_m), y_p^b)
4) h submits a ZK_AND(ZK_OR(h_1, ..., h_n), y_h^b)
5) p submits their lifetime and visit specific condition codes.
6) p submits a hash to the block: $H(H(\hat{b})||n^b)$, where n^b is a 256 bit nonce specific to b.
   a) $H(\hat{b})$ is equal to: $H(s_p^b \,||\, Ptr(D_b) \,||\, H(D_b) \,||\, H(\hat{b}-1))$.
7) p uploads their public key to the block: y_p^b.
8) h uploads their public key to the block: y_h^b.
9) If all information is correct, p will sign off on the block: $Sig_{y_p^b}$
10) If all information is correct, h will sign off on the block: $Sig_{y_h^b}$
11) p submits new block, b, to be confirmed.

Protocol 1: Block Creation Protocol

are able to find these blocks, they could learn patient aliases and other information that is not required for what they want to do.

Then, in Step 7- 8, the patient and hospital will submit their newly generated public keys, y_p^b, and y_h^b. We do this so the miners can use it to verify the ZKP in the block header for the patient.

b) Block Header: Contained inside of the header are non-interactive ZKP transcripts of both the hospitals and patients. In Steps 3-4, both the patient and hospital submit ZKPs to prove that these public keys are owned by a valid patient and a valid hospital, respectively. The patient will run a ZKP on their block-specific public key, y_p^b. This shows the miners that whoever submitted the block knows the associated private key. The patient will also run a ZK-OR on all the public keys within the patient registry. This shows the miners that the patient knows a private key corresponding to a public key that belongs to a valid patient in the registry. Finally, the patient will run a ZK-AND where they provide their ZKP for this new public key and their ZK-OR as inputs. This combination proves to the miners that whoever controls this new public key is also a valid patient within the registry. We repeat the same steps for the hospital to ensure the hospital creating this block is valid. This combination is used by the miners to verify the identities of the patient and hospital without revealing who they each are. By honoring the privacy of the patient and the hospital, we remain ethically virtuous as required by HIPAA guidelines, while also providing a service to researchers.

After the ZKPs have been submitted, we proceed to Steps 9 and 10 where both the patient and the hospital sign off on the block, represented as $Sig_{y_p^b}(b)$ and $Sig_{y_h^b}(b)$, respectively. Both signatures are created using the new public key generated for the patient and the hospital. When the patient submits the block, they will sign it to verify that the data they submitted is valid. The associated hospital signs the block to verify that the data is accurate and has not been tampered with. When reaching consensus, miners can use these signatures to verify that the data is accurate and has not been tampered with, while also checking if the public keys associated with the signature exist and have been verified via the ZKP. Finally, in Step 11, new block b is submitted to the miners (hospitals) for confirmation.

2) PHR Block Confirmation: In order to reach consensus, the miners in AegisBlock will need to verify that the information is accurate. The miners will confirm whether the ZKPs provided by the hospital and the patient in the header are valid. We use non-interactive Zero Knowledge Schnorr Proofs through the help of the Fiat-Shamir heuristic [26]. The miners will verify the ZKP transcripts on the block header. Once this has been done, the miners can verify that the signatures provided by the hospital and the patient are valid using the published public keys on the block. Once 50% of the miners approve this information, they upload a new block onto AegisBlock.

B. Phase II: Researcher Access

After the blocks have been created and approved by the mining groups, researchers can comb through the blocks to find a patient that fits their specific area of interest by looking at the lifetime and temporary condition codes of a block. The full researcher process can be seen in Figure 3.

The researchers then find the block that fits their criteria by going through the patient's codes. From then on, the researcher can fork the block, which will act as an initial request. Within the initial fork, the researcher can request a time-based record range of PHRs found within the blockchain. It is important to note, however, that the researcher has no knowledge of what other blocks may or may not exist for that patient within AegisBlock. After requesting a specific range, the researcher includes the pointer to the block that they are forking from, $Ptr(b)$. They would then sign the block they forked with their

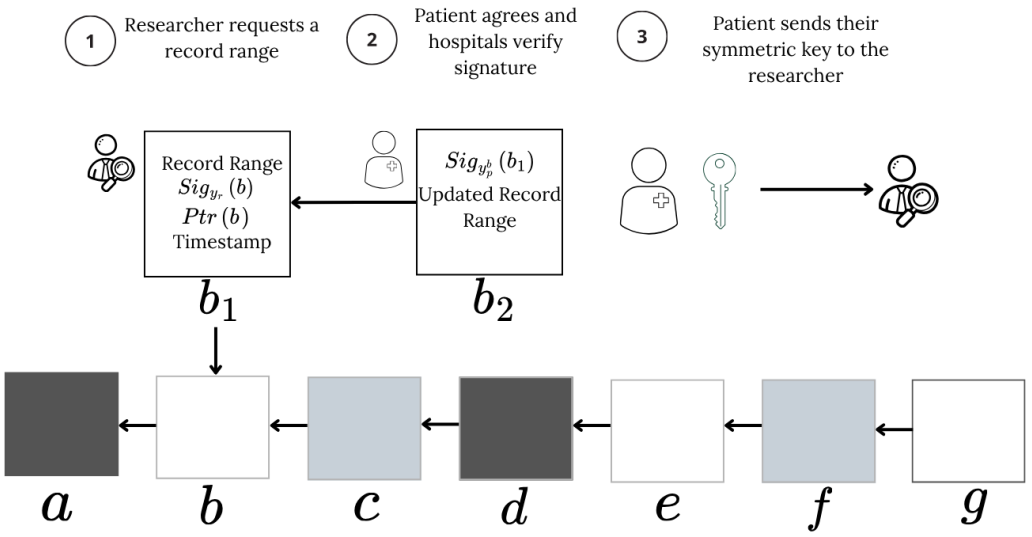


Figure 3: Process for Researcher Access

public key, $Sig_{y_r}(b)$. All of this information gets stored on a new block, b_1, as a fork to the patient block. Once the researcher block has been created, it must be approved by the miners before it is uploaded to AegisBlock.

If the patient agrees to the researcher's request, they can create another block, b_2, where they sign b_1 with their public key, $Sig_{y_p^b}(b_1)$. The patient also writes the record range they wish to grant to the researcher in b_2. If they would like to narrow the researcher's range, they may do so.

Once the patient signs off on the block, they will give the researcher all the data they requested. They will need to give $3k + 2$ pieces of information, where k is the number of blocks in the range. For each block b, they will need to give the researcher the symmetric key, s_p^b, a pointer to the data, $Ptr(D_b)$, and the hash of the data $H(D_b)$. For the first block in the range, block b_f, they will also need to give $H(\hat{b_f} - 1)$, so that the researcher can calculate $H(\hat{b_f})$. Additionally, for the last block in the range, b_l, they will need to provide n^{b_l}, the nonce of b_l, to allow the researcher to compare their final hash with the hash on the blockchain. This also means the patient has to reveal b_l in memory, however, b_l is the only block that would need to be revealed. With only this information, the researcher can calculate $H(\hat{b})$ for each block in the range, and can verify that all the information is accurate using the final block. As we have implemented a continuous hashing formula, if even one of these values is tampered with, all subsequent hashes (including the final one stored on the blockchain with the nonce) will not match. This storage method prevents malicious researchers from going outside the allowed record range and makes it impossible to know which blocks they are actually getting information from. This also makes it impossible for the patient to omit any data as the hashes would not match, and it is unreasonable for a patient to compute a different input that hashes to the same output due to the security of the hashing algorithm [27].

V. Experimental Evaluation

All experiments were performed locally on a computer cluster. We requested four tasks per node, two nodes, and 128gbs of memory.

A. Patient Block Creation

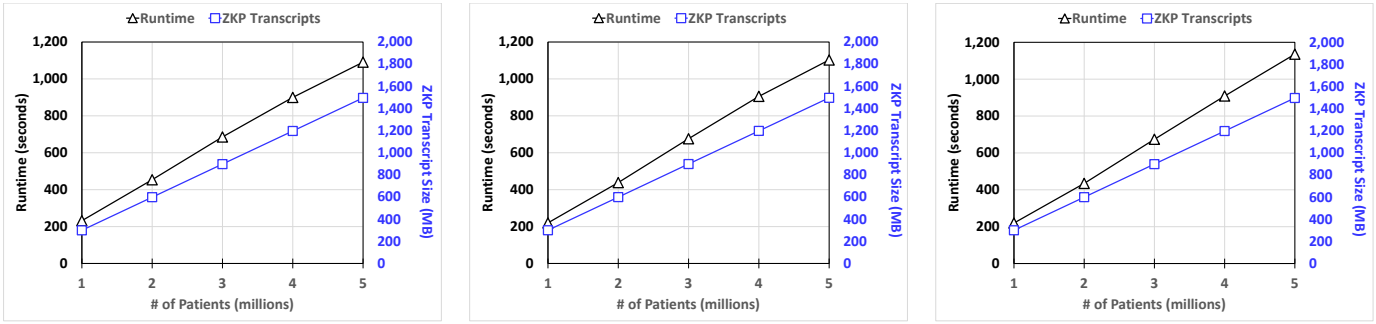
As AegisBlock gains more patients and hospitals in the registry, both the time and space required to create a new block will increase, potentially being infeasible for scalability. This experiment measures both the creation time of the block and the size of transcripts from a ZKP with respect to both the number of patients and the number of hospitals.

We separated our data into three different graphs, shown on Figure 4. Each one contains the number of patients on the x-axis (from one to five million, incremented by one million), the runtime of the protocol on the primary y-axis (measured in seconds), and the size of the ZKP transcripts on the secondary y-axis (measured in megabytes). The first graph shows these ten data points for a set of two thousand hospitals, the second for four thousand, and the third for six thousand.

As the bulk of the runtime is spent doing the ZKP, we would expect to see a runtime that is roughly $O(\#patients + \#hospitals)$. We observe in Figure 4 that the runtime appears to match this prediction. Additionally, as the number of ZKPs being conducted is equal to the sum of number of patients and the number of hospitals, we would expect the ZKP transcript size to also grow at approximately $O(\#patients + \#hospitals)$. Looking at Figure 4, this observation appears to be reflected in the data, with the ZKP transcript size growing at appromately the same rate.

B. Hospital Consensus Process

Miners are essential to AegisBlock, where they are relied upon to verify the integrity of each block. The purpose of this experiment is to measure robustness by determining how



(a) 2,000 Hospitals (b) 4,000 Hospitals (c) 6,000 Hospitals

Figure 4: Patient Block Creation

long it would take for a block to be approved assuming that our mining group has been compromised. This experiment measures robustness with respect to the malicious percentage of miners, as well as the scalability with respect to the number of hospitals. We measures the amount of time it takes to reach consensus on a patient block in AegisBlock with a varying percentage of malicious miners, from 10% to 40%. Block approval happens when half of the miners accept the block.

Our results are shown in Figure 5. On the x-axis, we have the number of hospitals from one thousand to five thousand, incremented by one thousand each time. On the y-axis, we measured the runtime in seconds for ten, twenty, thirty, and forty percent of malicious hospitals. The number of folds for this experiment is four.

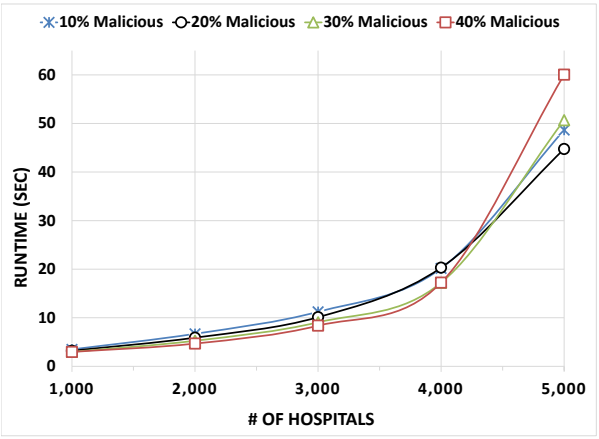


Figure 5: Hospital Consensus Process

We observe in Figure 5 that the runtime increases in $O(n^2)$ as the number of hospitals increases linearly, which is consistent with a model where every node propagates a signature to each other. We also observe that the runtime for different malicious levels with a fixed number of hospitals does not appear to significantly change.

C. Researcher Access

Researchers wanting to gain access to a patient's data have to go through a multi step process in order to be granted permission, including verification of requests by the miners. This experiment seeks to measure the robustness of the researcher access protocol with respect to malicious miners. Additionally, it seeks to measure the scalability of the protocol with respect to the percentage of malicious nodes.

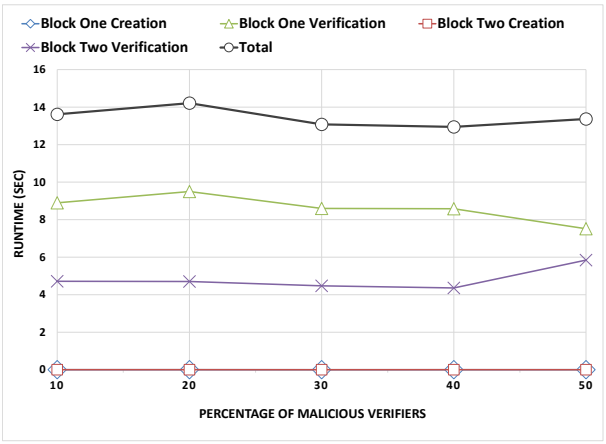


Figure 6: Researcher Access. Block One indicates the researcher request block, while Block Two indicates the patient approval block.

Our results for this experiment are located in Figure 6. On the x-axis, we measure the percentage of malicious miners, ranging from ten percent to fifty percent, incremented by ten percent. On the y-axis, we have the runtime of the whole operation in seconds. We fix the number of hospitals at six thousand for this experiment. We measured the runtime for four phases of the process, which include the creation and verification of the researcher request block (Block One), and the creation and verification of the patient approval block (Block Two). The number of folds for this experiment is four.

We observe that the creation time for both blocks is extremely quick, consistently less than 0.01 seconds, and the creation times for both blocks are independent of the malicious percentage of the mining pool. Our verification runtime is relatively constant even with an increased malicious percentage. The reason for this is our assumption that malicious

nodes immediately reject while honest nodes take their time to verify before approving. The total runtime of this process is approximately $O(1)$, showing that our system is robust for processing researcher requests with up to 50% of malicious miners.

VI. Conclusion and Future Work

In this paper we proposed AegisBlock, a blockchain based system where patients can maintain total anonymity while providing PHR to researchers. AegisBlock implements many different privacy measures to protect patient anonymity. Participation is voluntary, and if a patient decides to enroll, they are given different public keys for each block, making it harder for a malicious user to connect different blocks to the same person. AegisBlock also stores only a hash of several different variables, rather than variable each separately, meaning a malicious user would need to know multiple correct variables to identify a block, including a single use nonce. As all the condition codes are made public on the blockchain, AegisBlock makes it easy for a researcher to identify blocks they may be interested in. By forking the desired block, they can request patients to share their PHR, and they can ensure the validity of the data provided using the hashes on the blockchain. AegisBlock is a robust system with respect to the amount of malicious miners up to fifty percent as demonstrated by the experiments we designed and ran. AegisBlock is also scalable with respect to the number of existing hospitals and patients in the registry. AegisBlock can have important consequences with managing access control and protecting patient privacy. AegisBlock has also been shown to be ethically virtuous when protecting the privacy of all users involved.

As a future work, it would be interesting to investigate how to support other types of queries besides just time-based range queries, such as count and/or conditional range queries. Another future work would be to investigate how a patient can satisfy a researcher query without revealing any block that belongs to that patient. Our proposed solution reveals one block if the range query does not require future blocks' information, and two if it does.

Acknowledgment

This work is supported in part by the National Science Foundation under award number 2349042.

References

[1] J. Metcalf and K. Crawford, "Where are human subjects in big data research? the emerging ethics divide," *Big Data & Society*, vol. 3, no. 1, p. 2053951716650211, 2016.

[2] H. C. Assistance, "Summary of the hipaa privacy rule," *Office for Civil Rights*, 2003.

[3] D. S. Friedman, "Hipaa and research: how have the first two years gone?" *American journal of ophthalmology*, vol. 141, no. 3, pp. 543–546, 2006.

[4] S. H. Woolf, S. F. Rothemich, R. E. Johnson, and D. W. Marsland, "Selection bias from requiring patients to give consent to examine data for health services research," *Archives of family medicine*, vol. 9, no. 10, p. 1111, 2000.

[5] C. Andrade, "Research design: cohort studies," *Indian Journal of Psychological Medicine*, vol. 44, no. 2, pp. 189–191, 2022.

[6] K. E. Artnak and M. Benson, "Evaluating hipaa compliance: A guide for researchers, privacy boards, and irbs," *Nursing outlook*, vol. 53, no. 2, pp. 79–87, 2005.

[7] J. D. Miller, "Sharing clinical research data in the united states under the health insurance portability and accountability act and the privacy rule," *Trials*, vol. 11, no. 1, p. 112, 2010.

[8] D. Hossain, Q. Mamun, and R. Islam, "Unleashing the potential of permissioned blockchain: Addressing privacy, security, and interoperability concerns in healthcare data management," *Electronics*, vol. 13, no. 24, p. 5050, 2024.

[9] P. Novotny, Q. Zhang, R. Hull, S. Baset, J. Laredo, R. Vaculin, D. L. Ford, and D. N. Dillenberger, "Permissioned blockchain technologies for academic publishing," *Information Services and Use*, vol. 38, no. 3, pp. 159–171, 2018.

[10] M. P. McBee and C. Wilcox, "Blockchain technology: principles and applications in medical imaging," *Journal of digital imaging*, vol. 33, no. 3, pp. 726–734, 2020.

[11] A. Roehrs, C. A. Da Costa, and R. da Rosa Righi, "Omniphr: A distributed architecture model to integrate personal health records," *Journal of biomedical informatics*, vol. 71, pp. 70–81, 2017.

[12] A. Dubovitskaya, F. Baig, Z. Xu, R. Shukla, P. S. Zambani, A. Swaminathan, M. M. Jahangir, K. Chowdhry, R. Lachhani, N. Idnani *et al.*, "Action-ehr: Patient-centric blockchain-based electronic health record data management for cancer care," *Journal of medical Internet research*, vol. 22, no. 8, p. e13598, 2020.

[13] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd international conference on open and big data (OBD)*. IEEE, 2016, pp. 25–30.

[14] T. Ranaweera, H. Hewage, K. Preethilal *et al.*, "Ensuring electronic health record (ehr) privacy using zero knowledge proofs (zkp) and secure encryption schemes on blockchain," in *2023 5th international conference on advancements in computing (ICAC)*. IEEE, 2023, pp. 792–797.

[15] R. H. Hylock and X. Zeng, "A blockchain framework for patient-centered health records and exchange (healthchain): evaluation and proof-of-concept study," *Journal of medical Internet research*, vol. 21, no. 8, p. e13592, 2019.

[16] D. Cardoso, M. Anastácio, C. M. Antunes, M. Maximiano, R. Gomes, V. Távora, M. Dias, and R. C. Bezerra, "Using homomorphic encryption to create clinical trial

cohorts based on blockchain notarized private patient data," *Procedia Computer Science*, vol. 256, pp. 988–995, 2025.

[17] E. Zaghloul, T. Li, and J. Ren, "Security and privacy of electronic health records: Decentralized and hierarchical data sharing using smart contracts," in *2019 international conference on computing, networking and communications (ICNC)*. IEEE, 2019, pp. 375–379.

[18] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable cities and society*, vol. 39, pp. 283–297, 2018.

[19] L. Wang, X. Liu, W. Shao, C. Guan, Q. Huang, S. Xu, and S. Zhang, "A blockchain-based privacy-preserving healthcare data sharing scheme for incremental updates," *Symmetry*, vol. 16, no. 1, p. 89, 2024.

[20] L. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future generation computer systems*, vol. 95, pp. 420–429, 2019.

[21] O. Aldamaeen, W. Rashideh, and W. J. Obidallah, "Toward patient-centric healthcare systems: key requirements and framework for personal health records based on blockchain technology," *Applied Sciences*, vol. 13, no. 13, p. 7697, 2023.

[22] J. W. Song and K. C. Chung, "Observational studies: cohort and case-control studies," *Plastic and reconstructive surgery*, vol. 126, no. 6, pp. 2234–2242, 2010.

[23] S. J. Nass, L. A. Levit, L. O. Gostin *et al.*, *Effect of the HIPAA privacy rule on health research*. National Academies Press (US), 2009.

[24] K. D. Midkiff, E. B. Andrews, A. W. Gilsenan, D. M. Deapen, D. H. Harris, M. J. Schymura, and F. J. Hornicek, "The experience of accommodating privacy restrictions during implementation of a large-scale surveillance study of an osteoporosis medication," *pharmacoepidemiology and drug safety*, vol. 25, no. 8, pp. 960–968, 2016.

[25] U.S. Department of Health and Human Services, "Covered entities and business associates — hipaa privacy rule," *HIPAA Privacy Rule and Research*, 2007, accessed: 2025-07-24. [Online]. Available: https://privacyruleandresearch.nih.gov/pr_06.asp

[26] Q. Dao, J. Miller, O. Wright, and P. Grubbs, "Weak fiat-shamir attacks on modern proof systems," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 199–216.

[27] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–34, 2019.