Did You Break the Glass Properly? A Policy Compliance Framework for Protected Health Information (PHI) Emergency Access

Md Al Amin[a], Rushabh Shah[b], Hemanth Tummala[c] and Indrajit Ray[d]
Computer Science Department, Colorado State University, Fort Collins, Colorado, U.S.A.

Keywords: Emergency Access, Patient Consent, Break Glass Protocol, Policy Compliance, Blockchain, Smart Contract.

Abstract: HIPAA, HITECH, GDPR, and other data protection laws and regulations mandate patients' consent to access and share their data. They also impose compliance requirements for healthcare organizations. Non-compliance cases or failure to comply come with financial, reputational, business, and other penalties. In emergency medical situations, accessing a patient's protected health information or records can be critical to saving lives, especially when the patient is unconscious or unable to consent. This paper addresses the need for a secure, compliant, auditable system for emergency PHI access. We propose a blockchain and smart contract-based policy compliance framework where the emergency duty doctor requests access and must obtain approval from the senior in charge, which is recorded through multi-signature transactions. Once access is granted, the patient or their emergency contact is notified. To prevent unauthorized modifications, all actions are captured as immutable audit logs within a private blockchain network. The compliance check uses a novel Proof of Compliance (PoC) consensus mechanism, ensuring all access requests adhere to defined policies. This framework offers transparency, accountability, and security for emergency PHI access requirements.

1 INTRODUCTION

The digitization of healthcare data brings numerous benefits, including improved access to information and enabling real-time and remote care, sophisticated services, etc (King et al., 2014). It enhances patient outcomes by providing healthcare professionals with a comprehensive medical history and supporting coordinated care. Efficiency increases as administrative processes are streamlined, reducing errors and paperwork (Menachemi and Collum, 2011). As healthcare data becomes increasingly digitized, distributed, and interactive, concerns about the patient privacy and security of healthcare information and systems are growing within the healthcare ecosystem (Fernández-Alemán et al., 2013). Various security and privacy regulations are imposed worldwide to protect patient privacy and data security. *HIPAA & HITECH* (USA), *GDPR* (EU, UK), *APPs* (Australia), *PIPEDA* (Canada), *APPI* (Japan), and others are adequate data security and privacy laws. These privacy and data protection laws and regulations commonly dictate that data subjects, particularly patients in the healthcare industry, must provide consent to process their data as required for the intended purposes. Without permission, data should not be collected, processed, used, or shared beyond the mentioned purposes while collecting data to avoid security and privacy violations and lawsuits.

Healthcare providers and other users mainly access patients' healthcare data in three different circumstances: *(i)* accessed by the treatment team members for providing treatment and services and performing business operations; *(ii)* shared with others beyond the treatment team, including enhancing diagnosis and treatment plans through consultations with specialists, research and marketing endeavors, and others; *(iii)* emergency access when a patient is unconscious or insured and admitted in an emergency room in a life-and-death situation. Healthcare providers usually take consent for treatment and sharing purposes. Due to the uncertainty of the emergency, permission is not taken in advance. Also, an emergency may be far from the home or primary care provider. However, getting consent from the admitted or injured patient is impossible during an emergency as the patient is unconscious or incapacitated. It is

[a] https://orcid.org/0000-0003-1700-7201
[b] https://orcid.org/0009-0005-5658-0950
[c] https://orcid.org/0009-0007-7778-5845
[d] https://orcid.org/0000-0002-3612-7738

Al Amin, M., Shah, R., Tummala, H. and Ray, I.
Did You Break the Glass Properly? A Policy Compliance Framework for Protected Health Information (PHI) Emergency Access.
DOI: 10.5220/0013527000003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 195-208
ISBN: 978-989-758-760-3; ISSN: 2184-7711

a life-and-death situation. Healthcare providers may need to bypass traditional consent processes to access PHI for life-saving treatment. The "*break glass*" protocol or emergency access control is used (Ferreira et al., 2006). However, this access must comply with strict policy and regulatory requirements to protect health records, patients' privacy, and accountability.

Security and privacy policy compliance requirements for emergency access include, but are not limited to *(A)* patient must be experiencing a medical emergency and unconscious or unable to give consents to access PHI; *(B)* provider (hence known as *Requester*) must get approval from seniors (hence known as *Approver*) in charge to access PHI, *(C)* seniors in charge must determine the emergency and give approval; *(D)* PHI access must be done from the emergency room or patient carrying ambulance; *(E)* PHI access activities (audit logs) must be stored and not modified once recorded under any situations; *(F)* compliance review or audit must be done after treatment has been done without any delay according to the applicable policies; *(G)* patient or emergency contact person must be notified about PHI access; *(H)* separation-of-duty must be maintained and enforced strictly to keep functionalities of the requester, approver, audit log unit, and auditors; *(I)* Last but not least, least privileges and need-to-know must be maintained to make sure that the requester can access no less-no more health records to provide treatment and services to contain the situation and make the patient stable. In addition to these requirements, others might be based on the organization's business nature, regulations, legal jurisdictions, contractual obligations, etc.

Current research and practice focus on ensuring compliance requirements in an isolated and not timely manner. The following issues must be addressed for compliance assurance: *(a)* requester and approver must be accountable; *(b)* audit logs must be captured as they happened and protected from modifications under any situations by any users; *(c)* enforcing separation of duty to ensure that not a single entity can manipulate every step; *(d)* maintaining least privilege and need-to-know for protecting healthcare data and patient privacy by not disclosing some PHI locked by the patient; *(e)* assuring that after accessing PHI compliance review must be done quickly to check the compliance status and inform patient or emergency contact personnel without any delays.

This paper proposes a policy compliance framework for emergency PHI access to overcome the abovementioned issues and ensure streamlined policy compliance assurance. The proposed approach captures required information, stores it, and performs compliance reviews. A provider or requester submits an emergency access request for an admitted patient. Then, the senior in charge or approver evaluates the patient's condition and determines the criticality of the situation. If it is an absolute emergency, then the approver endorses the request. At this point, both the requester and approver sign the request as a multi-signature transaction using their corresponding private keys. A signed transaction is submitted to the blockchain network. Multi-signature-based blockchain transactions ensure that no single entity can submit transactions in the network. Emergency PHI access activities are captured and stored in a private audit blockchain to provide an immutable access history for compliance review. Finally, a compliance review process is proposed using a blockchain consensus mechanism called Proof of Compliance. Where a set of independent, decentralized, and distributed audit nodes perform compliance checking using provenance information.

Blockchain technology has inherent properties: security, transparency, and immutability (Conte de Leon et al., 2017). At its core, it is a distributed ledger technology that records transactions across multiple nodes so that the registered transactions cannot be altered. This feature ensures the integrity of data once it has been committed to the blockchain and significantly increases the system's fault tolerance and reliability. Integrating multi-signature transactions (Aitzhan and Svetinovic, 2016) at the core of the proposed approach is essential for establishing a decentralized and immutable record of interactions.

To the best of our knowledge, this work is the first to capture and enforce a multi-signature-based emergency PHI access policy compliance assurance framework. This paper makes the following contributions: *(i)* Integrating patient consent into the patient-provider agreement (PPA) and enforcing it while making an emergency PHI access decision. *(ii)* Leveraging *Approver* to evaluate and determine the PHI and access level for the submitted request by the *Requester* to ensure the least privilege and need-to-know basis emergency PHI access. *(iii)* Smart contract-based separation-of-duties enforcement to ensure that *Requester*, *Approver*, *Provenance Unit*, and *Compliance Reviewer* are separate and independent entities. *(iv)* Storing approval request information in the public blockchain using a multi-signature transaction scheme. So, the *Requester* and *Approver* cannot deny their actions, making them accountable for proving compliance assurance. *(v)* Implementing audit log provenance using a private blockchain to provide immutable PHI emergency access activity data. *(vi)* Performing compliance review using a decentralized

and distributed consensus mechanism called *Proof of Compliance* to determine the compliance status of every emergency PHI access. *(vii)* Conducting extensive experimental evaluations for the proposed approach on required smart contract deployment, PPA integrity, and informed consent storage and retrieval. *(viii)* Last but not least, performing and analyzing the gas costs, in token and USD, for informed consent and other required smart contract deployment, storing PPA integrity, and informed consent. Also, analyzing the time requirements for writing and reading data to/from the blockchain network.

2 PHI ACCESS CLASSIFICATION

This section outlines different access scenarios for healthcare data, including (i) treatment team access, (ii) sharing beyond the team, and (iii) emergency access, as depicted in Figure 1. Figure 2 shows sample health records with PHI ID, name, and description.

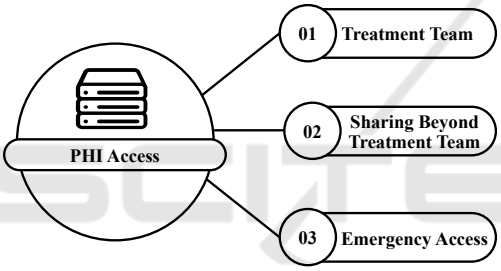

Figure 1: PHI Access Classification.

2.1 Treatment Team Access

Authorized treatment team members access healthcare data within healthcare systems to provide required medical care and services and perform healthcare operations. This includes doctors, nurses, and specialists collaborating to make informed decisions about diagnosis, treatment plans, and ongoing care. In addition to direct patient care, health records are used to perform essential business operations such as billing, insurance claims, scheduling, and quality assurance processes. Ensuring seamless access for healthcare providers while maintaining data privacy and security is critical. Robust access controls and encryption protocols are essential to safeguard sensitive information from unauthorized access or potential data breaches. The authors propose a consent-based PHI access compliance approach (Al Amin et al., 2023) for this group.

PHI ID	PHI Name	PHI Description
PHI-1001	Demographic Information	Basic personal information like name, date of birth, gender, contact
PHI-1002	Previous Medical History	Old medical records from another hospitals and providers
PHI-1003	Immunizations, Vaccinations	Immunization records that are administered over time
PHI-1004	Allergies	Various allergies sources, triggering condition, remediation
PHI-1005	Visit Notes	Physiological data, advises, follow-up, visit details
PHI-1006	Medications, Prescription	Pharmacy information, prescribed medications like name, dosage
PHI-1007	Pathology Lab Works	Biological samples analysis like blood, tissue, other substances
PHI-1008	Radiology Lab Works	Imaging results such as X-rays, CT, MRI, Ultrasound, PET scans
PHI-1009	Billing, Insurance	Bank account, credit/debit card, and insurance policy information
PHI-1010	Payer Transactions	Bills of doctor visit, lab works, and medications

Figure 2: Sample Protected Health Information (PHI).

2.2 Sharing Beyond Treatment Team

Healthcare data is often shared with others beyond direct care providers to enhance patient outcomes and drive broader healthcare initiatives. Consultations with specialists, for instance, allow for more accurate diagnoses and more effective treatment plans. Healthcare data is also leveraged in research to identify trends, develop new treatments, and improve overall healthcare quality. Furthermore, anonymized patient information may be used for marketing purposes, such as promoting relevant health services. However, these practices require strict compliance with data protection regulations to maintain patient privacy and consent. The authors in (Al Amin et al., 2024) proposed a policy compliance assurance framework using patient consent for PHI sharing.

2.3 Emergency Access

In life-and-death situations, such as when a patient is unconscious or critically injured and admitted to the emergency room, immediate access to their healthcare information becomes crucial for treatment. Under normal circumstances, healthcare providers seek consent from patients before accessing their medical data or sharing it with other specialists. However, in emergencies, consent cannot be obtained in advance due to the unpredictable nature of the situation. Additionally, emergencies may occur far from a patient's home or primary care provider, further complicating access to their medical history. In these scenarios, obtaining consent from the injured or incapacitated patient is impossible, as they may be unconscious or unable to communicate. This creates a unique challenge for healthcare professionals, who must act swiftly to provide life-saving care.

Emergency access protocols, such as the *Break-Glass Protocol*, allow healthcare providers to bypass consent temporarily, ensuring they can access essential information while maintaining compliance with privacy regulations and audit controls. If a patient is admitted to the same hospital, which is the primary care provider. Transferring data is unnecessary since doctors would access it from the same EHR system. Data access can be done from the emergency room

while treating the patient or in the ambulance while transferring a patient from the home or accident place to the hospital. When a patient gets regular treatment and medical services from one hospital but is admitted to another hospital for emergency treatment. The patient's health data must be shared between the primary and current providers. Providers must satisfy additional data protection and patient privacy requirements for transferring data. We assume data is transferred from the primary provider to the emergency provider through the proper channel.

This paper does not focus on policy compliance related to treatment team access and sharing of PHI. Instead, it addresses emergency access policy compliance, proposing a blockchain and smart contract-based multi-signature approval system, with audit logs stored on a private blockchain and compliance status verified through a *PoC* consensus mechanism.

3 RELATED WORKS

Yang et al. (Yang et al., 2017) introduced a novel lightweight break-glass access control (*LiBAC*) system designed for the Healthcare IoT, enhancing the security and accessibility of medical data. The system employs a dual access method: attribute-based for regular use and break-glass for emergencies, ensuring timely access to patient information by authorized personnel. The *LiBAC* is rigorously proven secure under the standard model, with formal proof provided to substantiate its resilience against potential cyber threats. Despite its efficiencies, the model relies heavily on a predefined set of emergency contacts, potentially limiting its effectiveness in unexpected situations where those contacts may not be available or when new, unforeseen stakeholders need access.

Loos et al. (Loos, 2020) investigated the tension between emergency accessibility and security in medical devices, highlighting the absence of comprehensive break-glass systems tailored for such devices. They categorized break-glass mechanisms into patient records and medical devices. The authors explore emergency access solutions such as proximity-based access, biometric authentication, UV tattoos, RFID chips, and passive radiopaque markers. Despite proposing innovative mechanisms, they underscore challenges like balancing usability and security, patient acceptance, and lack of standardization. The paper urges further research into unified security protocols that reconcile emergency access needs with robust patient data protection.

Aski et al. (Aski et al., 2021) proposed integrating break-glass mechanisms with attribute-based access control (*ABAC*) to address emergencies in healthcare *IoT* systems. In addition to authorizing users in normal situations, they introduced a break-glass mechanism allowing emergency situation handlers (*ESH*) to handle emergencies. The *ESH* bypasses standard authentication and swiftly accesses critical patient data when immediate medical action is required. Security measures include data encryption and key management, with *ESH* verification through pre-distributed passwords to prevent misuse. Experimental analysis indicates the scheme's efficiency compared to existing access control systems.

Schefer-Wenzl et al. (Schefer-Wenzl et al., 2013) surveyed to investigate the delegation and break-glass-based emergency access control where the standard access policies are insufficient. In delegation models, a user is allowed to transfer access rights or roles to another, discussing role-based and permission-based approaches while considering constraints like separation of duty (*SoD*) and binding of duty (*BoD*). The break-glass models are designed for emergencies, enabling temporary bypass of standard access controls with actions logged to prevent misuse. Analyzing 329 articles and detailing 35 key approaches, the authors compare models based on policy enforcement, support for entailment constraints, and integration with business processes.

Van Bael et al. (Bael et al., 2020) described a new access control system that uses *IoT* sensors to gather contextual data, making break-glass mechanisms more flexible in an emergency. It includes non-repudiation features by logging all actions during a break-glass event, ensuring accountability through evidence like biometric data or badge scans. A fail-safe mechanism is also incorporated to cancel emergency access if activated erroneously. However, the prototype shows it is possible and has reasonable response times. The proposed approach relies on the availability and dependability of *IoT* sensors. It is vulnerable if the integrity of contextual data is compromised, and it is hard to set up complete access policies for all emergencies.

The papers above summarized the application of emergency access control mechanisms like the break-glass protocol. However, they failed to address the security and privacy compliance requirements mandated by various laws and regulatory agencies, such as *HIPAA* and *GDPR*. This paper proposes a policy compliance framework for emergency PHI access to ensure that applicable security and privacy policies are followed while accessing PHI and saving patient life in a critical moment.

4 PROPOSED APPROACH

The primary goal is to enforce necessary consents and policies for emergency access, capturing essential PHI access activity to verify compliance with security and privacy requirements. Proper policy enforcement is crucial to ensuring compliance with preserving provenance records and conducting timely compliance reviews to maintain a secure and compliant system. For enforcement, this paper considers patient-informed consent, where the patient locks any PHI to keep it restricted from access during an emergency. This work leverages multi-signature-based access request approval to ensure that PHI is not accessed unnecessarily. The emergency PHI access activities are captured and recorded in a private blockchain network as audit logs to provide provenance and reconstruct events that reflect their actual occurrence. Finally, a blockchain consensus mechanism (PoC) is approached to examine the enforcement actions against the applied policy and informed consent, using the provenance data to verify and certify the compliance status.

4.1 Patient-Provider Agreement (PPA)

The patient-provider agreement (PPA) defines the responsibilities of each party in a treatment scenario (Albrecht et al., 2015). It is established when a patient visits a hospital and is documented to facilitate healthcare services. The specifics of a PPA vary by organization and are tailored to match the treatment needs and responsibilities required. The components and format of the PPA also differ depending on the hospital or clinic. Figure 3 shows the structure of a PPA. The central concept of PPA is adopted from (Al Amin et al., 2023; Al Amin et al., 2024). The authors focused on consent management for medical treatment and diagnosis purposes, mainly for the treatment team members and health data sharing beyond the treatment team. They did not include patient consent for emergency access. This paper extends the PPA structure to analyze the requirements and formalize the consent components for emergency PHI access.

A PPA is formally composed of six (6) tuples:

$$PPA = (PC, PrC, TIC, SIC, EIC, ROC)$$

satisfying the following requirements:

(A) *PC* is a finite set of patient components containing the patient's personal information, contact information, mailing information, pharmacy information, billing and insurance information, emergency contact, and others. The patient is responsible for providing and maintaining these components' valid, accurate, and updated information.

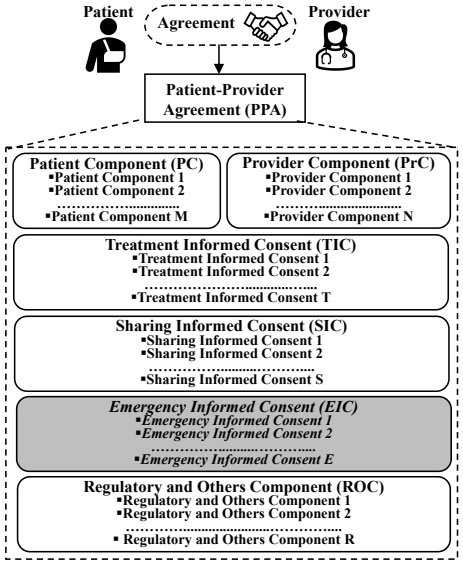

Figure 3: Patient-Provider Agreement (PPA) Components.

(B) *PrC* is a finite set of provider components, including the treatment team, prescription, and others. The provider is responsible for creating an effective team to provide appropriate care. Everything from treatment to insurance coverage and billing is considered during the patient treatment period.

(C) *TIC* is a finite set of treatment-informed consent components. It denotes that the patient has permitted the designated treatment team to access medical records. Treatment team members include doctors, nurses, support staff, lab technicians, billing officers, emergency contact persons, and others assigned by the authority. Some outsider members are insurance agents, pharmacists, pharmacy technicians, doctors, lab technicians from another hospital, etc.

(D) *SIC* is a finite set of sharing informed consent components for sharing PHI beyond the treatment team to get better services. It denotes the patient's consent to sharing medical data for specific purposes: treatment, diagnosis, marketing, and research. Both the sender and the receiver must have permission to share data.

(E) *EIC* is a finite set of emergency informed consent components. It denotes that the patient has permitted the designated treatment team to access medical records. The primary purpose of this work is *EIC*, including (i) identifying, capturing, and storing consent components; (ii) enforcing consents with other applicable security policies and industry best practices to ensure policy

compliance while making emergency PHI access decisions; (iii) defining and capturing provenance information with the enforced consents to maintain audit logs; (iv) performing compliance checking using consensus mechanisms; (v) providing services for both given and executed consents, etc. It does not consider other components: *PC*, *PrC*, *TIC*, *SIC*, and *ROC*.

(F) *ROC* represents a finite set of regulatory components and other relevant elements. It encompasses applicable security and privacy policies required to meet the compliance standards of various governmental levels—local, state, federal, and international—as well as regulatory bodies such as HIPAA and GDPR. Additionally, it may incorporate contractual obligations where applicable.

4.2 Emergency Informed Consent (EIC)

Before approval, patients must know about the emergency informed consent, particularly which PHI must be locked from access. Figure 4 shows the EIC conceptual structure. Emergency informed consent is formally composed of two tuples:

$$EIC = (PHI, LS)$$

satisfying the following requirements:

(a) *PHI* is a finite set, d number, of health records denoted by $\{phi_1, phi_2, phi_3, ...phi_d\}$. It is a digital version of a patient's medical history maintained by healthcare providers over time. Classified as protected health information (PHI), it contains sensitive patient details that must be safeguarded against unauthorized access, disclosure, and sharing. Figure 2 illustrates ten types of PHI considered for each patient, including PHI ID, name, and description. In emergencies, healthcare providers access these records to deliver life-saving treatments.

(b) *LS* is the lock status of the intended PHI with two values: *Locked* and *Unlocked*. A finite set of lock statuses, a d number, can be denoted as $\{ls_1, ls_2, ls_3, ...ls_d\}$. The *Locked* status indicates the PHI cannot be accessed at any moment under any circumstances. The providers cannot access *Locked* PHI during an emergency. While PHI can be accessed during an emergency if the lock status is *Unlocked*. The patient must consult with the corresponding providers to review before locking PHI. It should not create any burden for giving life-saving treatment during an emergency.

There is a one-to-one mapping between each PHI and its lock status: $(phi_1, ls_1), \ldots, (phi_d, ls_d)$. This mapping ensures that patient privacy is respected and health records security is maintained during emergency access.

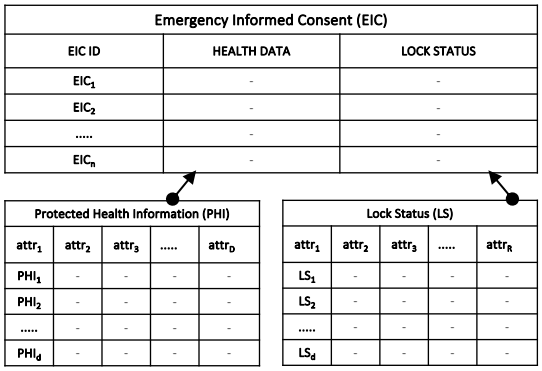

Figure 4: Emergency Informed Consent (EIC) Structure.

4.3 EIC Smart Contract Deployment

Once a *PPA* is established and stored in the repository, all components of the emergency informed consent are deployed as smart contracts within the blockchain network. Figure 5 illustrates the deployment process of the *EIC* smart contracts. The Smart Contract Deployment Unit (*SCDU*) first collects all components of the informed consent from the *PPA* described in *Step 4*. It then verifies the integrity of these components in *Step 5* to confirm that no deliberate or accidental modifications have occurred. Operating as a secure entity, the *SCDU* ensures that any alterations would invalidate the consent. If the consent components are confirmed to be unaltered, the *SCDU* creates and deploys the corresponding smart contracts on the blockchain network in *Step 6*. Subsequently, it updates the patient's profile and the hospital system in *Step 7*. In *Step 8*, users with the appropriate credentials can query and receive responses regarding informed consent directly from the blockchain network. This smart contract-based approach offers an automated system that ensures the integrity and accountability of deployed consents. Once consents are integrated into the smart contract, they become immutable, preventing alterations. The authorization module interacts with these smart contracts, utilizing them alongside other components to make emergency PHI access decisions.

4.4 Emergency Access Authorization

Consent enforcement ensures that related consents are executed while making decisions for the emergency PHI access requests. All consents are stored on the public blockchain network as smart contracts and can-

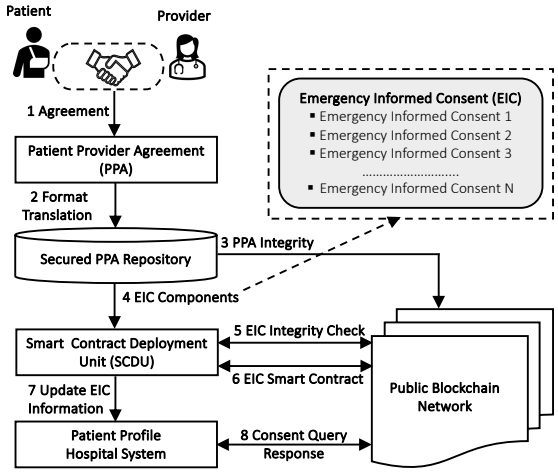

Figure 5: EIC Smart Contract Deployment Process.

not be enforced until they are called. The authorization module (AM), like *Break-Glass Protocol*, considers emergency informed consent with applicable policy and required attributes while making decisions. The attributes may be subject, object, operation, and environmental attributes. The *Requester* must provide the necessary credentials for identification and authentication. Figure 6 shows the informed consent enforcement for the emergency PHI access authorization and policy compliance assurance framework.

The *Requester* submits an emergency PHI access request to the *Approver* in *Step 1*. The *Approver* evaluates and determines the urgency of the admitted patient. Then, the *Approver* approves the access requests through the *Multi-Signature Approval System (MSAS)* in *Step 2*. Both *Approver* and *Requester* use their private keys to sign the transaction. The signed request is submitted to the public blockchain networks like *Ethereum* in *Step 3* to be added to the distributed ledger. Later, this deployed transaction is a source of truth to hold the signers accountable. In *Step 4*, the approved request is forwarded to an emergency authorization module (AM) like *Break-Glass Protocol* for PHI access authorization decision. The AM queries the blockchain network through the corresponding smart contract to get emergency informed consent information and signed request approval transactions for the submitted access request in *Step 6a*. It also makes queries for requests related to applicable policies and required attributes in *Steps 6b* and *6c*.

After evaluating, it makes an authorization decision and sends it to the *EHR* in *Step 7* and notification to the patient's emergency contact in *Step 8*. If the access request is approved, the intended PHI is delivered to the *Requester* in *Step 11*. The audit logs recording unit, *ALRU*, collects logs from the MSAS in *Step 5* and from the AM in *Step 9*. It combines logs and stores them as audit logs in *Step 6c* in Private Audit Blockchain. Section 5 discusses block structure and others. The compliance review is done in *Steps 12a, 12b*, and *12c* by the Proof of Compliance consensus mechanism. Compliance status reports are produced in *Step 12d*. Section 6 discusses the required mechanism. For this study, it is considered that the authorization module is not compromised or tampered with. It is the reference monitor for making access decisions and must be tamper-proof (Mulamba and Ray, 2017). Also, the communication channel between *AM* and the smart contract access points or apps is secured from malicious users.

4.5 Separation-of-Duty Enforcement

There are four significant actors in the proposed approach: (i) the *Requester* who submits the request to access patient data; (ii) the *Approver* who evaluates the situation and determines the level of access required by the *Requester*; (iii) the *Provenance Unit* who maintains all audit logs and applied policies; and (iv) the *Compliance Reviewer* who performs compliance checking to determine the compliance status for every emergency access. These four actors must be different entities from each other. No one entity should perform more than one task. Figure 7 depicts the *SoD* requirements for emergency PHI access compliance. This proposed approach delegates smart contracts to enforce separation-of-duty among those entities to avoid conflicts of interest.

Figure 8 shows the *SoD* enforcement approach for the entities that must be separated for various phases. In *Phase 1*, the *Requster* and *Approver* must be different users. The *MSAS* checks and enforces this condition during the request approval process by the *Approver*, as shown in Figure 6. In the next *Phase 2*, it is ensured that the *Provenance Unit* is different from the *Requster* and *Approver*. The *ALRU* ensures that while collecting and storing audit logs in the private audit blockchain. Finally, it is ensured that the *Compliance Reviewer* is a separate entity from the *Requster*, *Approver*, and *Provenance Unit* (*Phase 3)*. The proposed *Proof of Compliance* maintains the *Phase 3* conditions while performing the compliance review.

5 PHI ACCESS PROVENANCE

Enforcing an applicable set of policies is crucial, but preserving data provenance to show adherence to these policies is also essential. Nevertheless, policy compliance cannot be quantified or confirmed in iso-

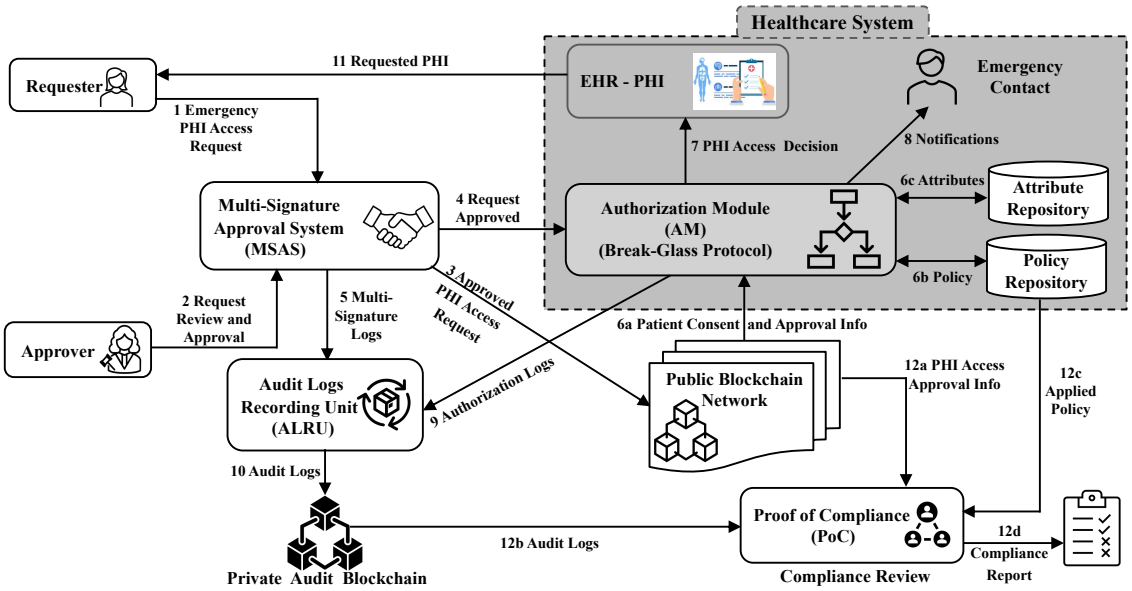

Figure 6: Proposed Emergency PHI Access Policy Compliance Assurance Framework.

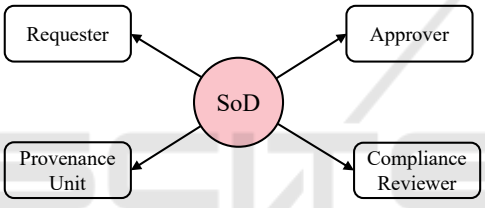

Figure 7: Separation-of-Duty (SoD) Requirements.

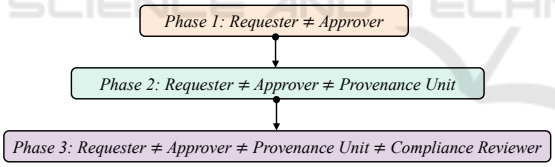

Figure 8: Proposed Separation-of-Duty (SoD) Enforcement.

lation. An independent auditor conducts a thorough policy audit to verify compliance with the policy, utilizing the available provenance data to ascertain and certify the policy's compliance status. For an accurate policy compliance assessment, two critical elements must be diligently maintained: *(i) emergency PHI access request approval* and *(ii) emergency PHI access audit logs*. This section contains detailed provenance mechanisms dedicated to preserving the integrity of emergency PHI access request approvals and ensuring the audit logs' authenticity.

5.1 Emergency PHI Access Approval

After submitting the emergency access request, the *Approver* evaluates and determines the situation to make the decision. If conditions demand, the submitted request is approved and forwarded to the authorization module for the final PHI access decision. Both request and approval make a transaction signed by the *Requester* and *Approver* using their private keys or wallets. The signed transaction is submitted and recorded in the public blockchain to provide an unaltered source of truth regarding emergency PHI access compliance review. This is done through the multi-signature scheme of blockchain technology (Aitzhan and Svetinovic, 2016). Due to the cryptographic properties, both *Requester* and *Approver* cannot deny their actions regarding PHI access.

5.2 Emergency PHI Access Audit Logs

Integrity in policy enforcement ensures that events are documented faithfully, reflecting the sequence and nature of actions taken. This authenticity is crucial for transparency and accountability. Provenance plays a key role by offering a detailed and unalterable history of policy enforcement actions as they are carried out, safeguarding against any tampering of records. The alteration of audit trails or unauthorized access to healthcare data is strictly prohibited to maintain the sanctity of the process. Maintaining the integrity of the audit trail is essential for policy compliance assurance. If integrity is compromised, checking compliance status to find compliance and non-compliance cases is questionable. The blockchain provides these requirements as ledger properties. This work adopts a private blockchain as an audit log storage system.

Figure 9 illustrates the private audit blockchain's

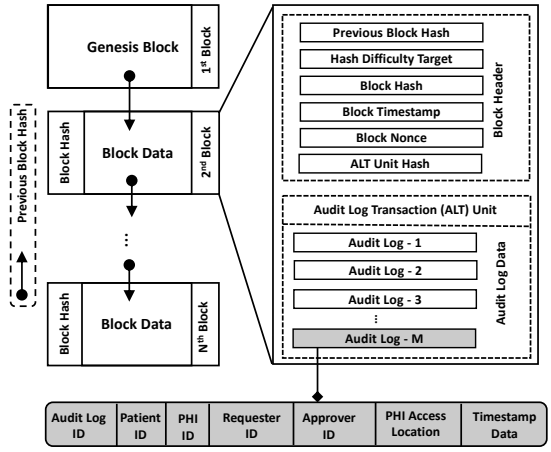

Figure 9: Audit Blockchain Block Structure.

block components and structure. Each block has a block header part that contains block metadata and a data part that stores the audit trail data. Each audit log has seven components: *(i) audit log ID; (ii) patient ID; (iii) PHI ID; (iv) Requester ID; (v) Approver ID; (vi) PHI access location;* and *(vii) timestamp data.*

The audit log ID uniquely identifies each access log, while the patient ID refers to the patient receiving emergency life-saving treatment. The PHI ID indicates the specific health records accessed during treatment, as depicted in Figure 2. Patients can lock any particular health record in *EIC*. The Requester ID identifies the healthcare provider treating the admitted patient who needs access to the patient data. The Approver ID belongs to the person responsible for evaluating and endorsing the access request based on the current situation for authorization. These access requests and endorsements are securely recorded on a public blockchain network like *Ethereum* through a multi-signature process, ensuring non-repudiation by involved parties. The PHI access location identifies the physical setting, such as an emergency room or an ambulance, from which healthcare records are accessed. Finally, the timestamp means the time when the access authorization is done. Steps *5* and *9* in *Figure 6* show the process of capturing audit logs from the *MSAS* and *AM*. The *ALRU* stores audit logs in a private audit blockchain in *Step 10*.

6 COMPLIANCE REVIEW

Simply maintaining audit logs and enforcing informed consent and policies does not guarantee compliance. A mechanism that can verify compliance status using these elements is crucial. This paper introduces a blockchain-based Proof of Compliance (*PoC*) consensus mechanism designed to validate compliance through the utilization of audit logs, informed consent (*EIC*), and other relevant policies. The *PoC* is governed by independent, distributed auditor nodes, which operate separately from the units, enforcing policies and managing provenance, ensuring unbiased compliance verification. Figure 10 shows the decision mechanism of *PoC*.

6.1 Decision Counting Threshold

Assume s auditor nodes are in the *PoC* network. A batch of transactions is processed to assess compliance status, but it is not guaranteed to receive responses from all s nodes. Responses may be missing due to various reasons such as connectivity issues, power failures, intentional non-submission, or auditor nodes going offline unexpectedly due to system errors (Haeberlen et al., 2007). Now, consider that m is the number of responses from the auditors out of s. A required threshold, η, must be satisfied to make the compliance decision for an audit log. The following conditions must be satisfied to make the compliance decision:

$$(i) s \geq m \quad \text{and} \quad (ii) s \geq m \geq \eta \quad \textbf{or} \quad s \geq D_m \geq \eta$$

Where D_m is the number of received decisions from the m number of auditors (A), and η is the minimum number of decisions that must be present to make the decision. If there is no loss, this $s = m$ is ideal. Then the conditions became:

$$(i) m \geq \eta \quad \textbf{or} \quad D_m \geq \eta$$

In the ideal case, all auditors receive the required information and return results after the compliance evaluation. The value of the η is determined and influenced by the design decision, the organization's business nature, legal requirements, contractual obligations, and others. If $m < \eta$ or $D_m < \eta$, the compliance status is assigned as *"Not-Determined"* to avoid any policy violation, it must be further investigated to check the reasons.

6.2 Auditors and Decisions

Let m be the total number of auditor nodes; the following information is given. The final compliance decision is derived based on a majority rule among decisions.

- Let $A = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m\}$ is defined as a set of auditors, where each α_i represents an individual auditor node. These nodes are responsible for checking the compliance requirements. Auditor nodes can be hospitals, local governments,

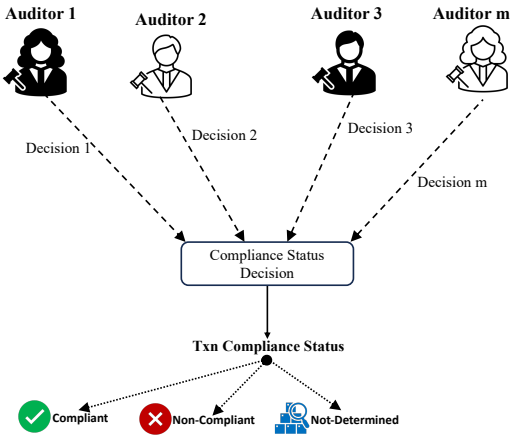

Figure 10: Proof of Compliance Decision Mechanism.

state governments, the federal government, regulatory agencies, insurance companies, business associates, accreditation bodies, independent auditors, and others from contractual obligations.

- Let D be a set of decisions corresponding to each auditor in A, defined as $D = \{\delta_1, \delta_2, \delta_3, \ldots, \delta_m\}$, where δ_i is the decision made by the α_i auditor node for a given transaction, where $\delta_i \in \{Compliant, Non-Compliant, Not-Determined\}$
- There is a one-to-one mapping between each auditor node and its decision: $(\alpha_1, \delta_1), \ldots, (\alpha_m, \delta_m)$ since each auditor node α_i in set A makes a compliance decision δ_i in set D. Therefore,. This mapping allows us to analyze the decisions collectively and apply the *PoC* decision combining algorithm to determine the compliance status.

6.3 Decision Counting Process

The total counts for each type of decision are calculated as follows, where $F(.)$ is an indicator function that equals *1* if the inside condition is true and *0* otherwise.

$$(a) \quad \mathbb{C} = \sum_{i=1}^{m} F(D_i = Complaint)$$

$$(b) \quad \mathbb{N} = \sum_{i=1}^{m} F(D_i = Non-Complaint)$$

$$(c) \quad \mathbb{U} = \sum_{i=1}^{m} F(D_i = Not-Determined)$$

6.4 Decision Combining Process

After counting, the final decision is made, and the distinct combinations are given in Table 1. The *Not-Determined* dictates to others if they are equal to it.

Table 1: PoC Decision Combining Scope.

SN	Decision Counting Combination	Final Decision ($\mathbb{D}_{final}$)
1	$\mathbb{C} > \mathbb{N} > \mathbb{U}$	$\mathbb{C}$
2	$\mathbb{C} > \mathbb{U} > \mathbb{N}$	$\mathbb{C}$
3	$\mathbb{C} > \mathbb{N} = \mathbb{U}$	$\mathbb{C}$
4	$\mathbb{N} > \mathbb{C} > \mathbb{U}$	$\mathbb{N}$
5	$\mathbb{N} > \mathbb{U} > \mathbb{C}$	$\mathbb{N}$
6	$\mathbb{N} > \mathbb{C} = \mathbb{U}$	$\mathbb{N}$
7	$\mathbb{N} = \mathbb{C} > \mathbb{U}$	$\mathbb{N}$
8	$\mathbb{U} > \mathbb{C} > \mathbb{N}$	$\mathbb{U}$
9	$\mathbb{U} > \mathbb{N} > \mathbb{C}$	$\mathbb{U}$
10	$\mathbb{U} = \mathbb{C} > \mathbb{N}$	$\mathbb{U}$
11	$\mathbb{U} > \mathbb{C} = \mathbb{N}$	$\mathbb{U}$
12	$\mathbb{U} = \mathbb{N} > \mathbb{C}$	$\mathbb{U}$
13	$\mathbb{C} = \mathbb{N} = \mathbb{U}$	$\mathbb{U}$

The final decision $\mathbb{D}_{final}$ can then be set based on predefined majority rules, such as:

- *Compliant Majority:* This decision is made when the majority decision is *Complaint* or $\mathbb{C} > \mathbb{N}$ and $\mathbb{C} > \mathbb{U}$ out of *m* decisions made by the auditors regardless $\mathbb{N} > \mathbb{U}$ or $\mathbb{U} > \mathbb{N}$ or $\mathbb{U} = \mathbb{N}$.
- *Non-Compliant Majority:* This decision is made when the majority decision is *Non-Complaint* or $\mathbb{N} > \mathbb{C}$ and $\mathbb{N} > \mathbb{U}$ out of *m* decisions made by the auditors regardless $\mathbb{C} > \mathbb{U}$ or $\mathbb{U} > \mathbb{C}$ or $\mathbb{C} = \mathbb{U}$.
- *Not-Determined Majority:* This decision is made when the majority decision is *Not-Determined* or (i) $\mathbb{U} > \mathbb{C}$ and $\mathbb{U} > \mathbb{N}$, or (ii) $\mathbb{U} = \mathbb{C} = \mathbb{U}$, or (iii) $\mathbb{U} = \mathbb{C} > \mathbb{U}$, or (iv) $\mathbb{U} = \mathbb{N} > \mathbb{C}$ out of *m* decisions made by the auditors regardless $\mathbb{C} > \mathbb{N}$ or $\mathbb{N} > \mathbb{C}$ or $\mathbb{C} = \mathbb{N}$.

6.5 PoC Compliance Report

After determining the compliance status, it is stored in a private blockchain to ensure transparency, immutability, and accountability. Figure 11 shows the compliance block structure. Each compliance block includes unique audit log IDs and corresponding compliance statuses, categorized as *compliant*, *non-compliant*, or *not-determined*. These blocks are then stored within the private compliance blockchain, providing an immutable record of all verified compliance checks.

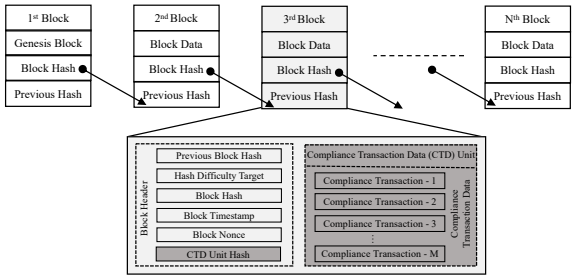

Figure 11: PoC Compliance Blockchain Block Structure

7 EXPERIMENTAL EVALUATION

The *Ethereum Virtual Machine (EVM)* based blockchains are chosen for the proposed approach experiments. It offers a Turing-complete smart contract language, *Solidity*, which enables the implementation of our model's logic. We developed smart contracts for storing and retrieving informed consent, testing them on test networks: *Ethereum* and *Optimism* to ensure reliability before deployment. Since smart contracts, once deployed, are immutable and errors can incur financial and reputational costs, rigorous testing on these networks is crucial. Ethereum's *Remote Procedure Call (RPC) API* services are employed for deploying smart contracts on these test networks (Kim and Hwang, 2023). Utilizing public *RPC* eliminates the need to maintain a blockchain node for contract interaction, assuming minimal resource usage (*CPU, HDD, Bandwidth*) on the local machine. Faucet *ETH* serves as gas to authorize transactions using the Metamask wallet (Lee, 2023).

7.1 Private Audit Blockchain

We have chosen a private blockchain infrastructure to manage the provenance of audit logs, specifically utilizing an Ethereum private network deployed via the Go Ethereum (geth) client. This approach enhances data security and provides centralized control over policy-provenance activities. The private network employs the *Proof of Authority (PoA)* consensus algorithm, specifically the *Clique* protocol, to mine and validate the audit trails. Additionally, as the *Proof of Compliance* algorithm evolves, modifications to the *Clique* algorithm can be implemented to adapt the block structure to meet specific requirements.

Figure 12 shows the miner node responsible for the end-to-end transaction handling process. Beginning with the submission of transactions. Once a transaction is submitted, the miner node includes it in a block and uses the mining process to validate it. Furthermore, the miner node actively publishes this mined data to all other nodes within the network, ensuring a synchronized and updated ledger.

7.2 Block Integrity Writing Cost

In the proposed approach, audit logs are stored in the audit blockchain, and compliance status is stored in the compliance blockchain. Both are private blockchain networks, where participants are limited to organizations. This doesn't provide the public with trust. To avoid this, block ID and hash as integrity are stored in a public network like *Ethereum*. Figure 13 shows the block integrity storage cost in tokens for two public blockchain networks: *Ethereum and Optimism*. The *USD* costs are depicted in Figure 13. Ethereum is Layer 1, and the optimism is *Layer 2* (Gangwal et al., 2023; Gudgeon et al., 2020). *Layer 1* is the core blockchain framework for implement-

```
INFO [11-30|00:04:45.006] Successfully sealed new block      number=2003 sealhash=ab913e..f93552 hash=0def93..687eda elapsed=5.001s
INFO [11-30|00:04:45.009] Commit new sealing work            number=2004 sealhash=b28694..b15e07 txs=0 gas=0 fees=0 elapsed=2.191ms
INFO [11-30|00:04:49.408] Looking for peers                  peercount=1 tried=0 static=0
INFO [11-30|00:04:50.005] Successfully sealed new block      number=2004 sealhash=b28694..b15e07 hash=446467..f174f0 elapsed=4.996s
INFO [11-30|00:04:50.008] Commit new sealing work            number=2005 sealhash=d5135d..dbb4bf txs=0 gas=0 fees=0 elapsed=1.807ms
WARN [11-30|00:04:53.154] Caller gas above allowance, capping requested=112,915,065 cap=50,000,000
INFO [11-30|00:04:53.188] Setting new local account          address=0x9Ee65e7127516E08AdE560eb63fb8c26016348d7
INFO [11-30|00:04:53.188] Submitted contract creation        hash=0x4117d77443e685b70d9eef36d54cfe495e53641eb124a51c283101199f9f5b57 from=0x9Ee65e7127516E
8AdE560eb63fb8c26016348d7 nonce=67 contract=0x58707DABaC5Cd7AA5d12AB721a1bf003099436c6 value=0
INFO [11-30|00:04:54.014] Commit new sealing work            number=2005 sealhash=3d1180..e80f66 txs=1 gas=105,687 fees=0.000105687 elapsed=2.270ms
INFO [11-30|00:04:55.003] Successfully sealed new block      number=2005 sealhash=3d1180..e80f66 hash=e9b97c..598d65 elapsed=989.129ms
INFO [11-30|00:04:55.005] Commit new sealing work            number=2006 sealhash=6f9ba7..42cb72 txs=0 gas=0       fees=0           elapsed=1.643ms
```

Figure 12: Miner Node Operations.

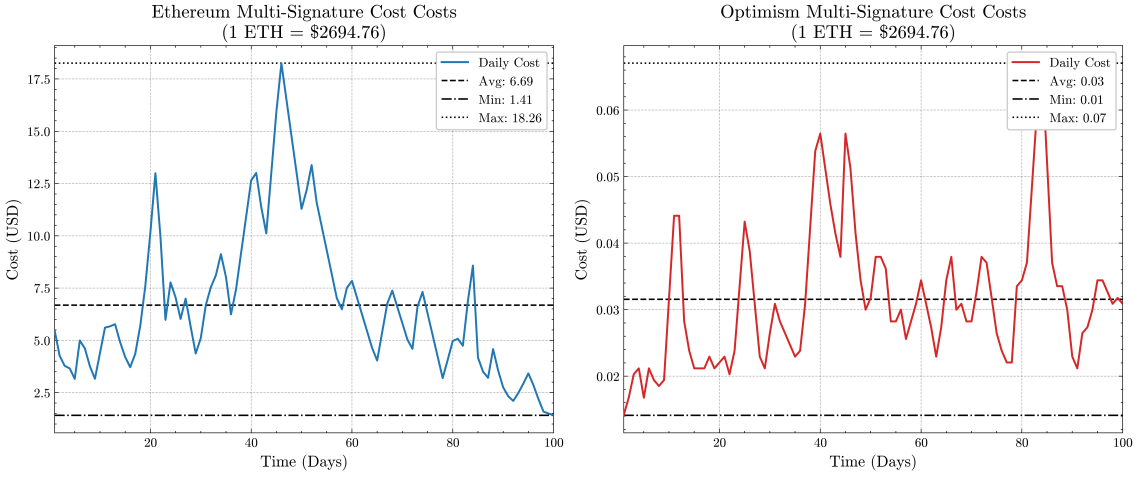

Figure 13: Smart Contract Deployment Cost.

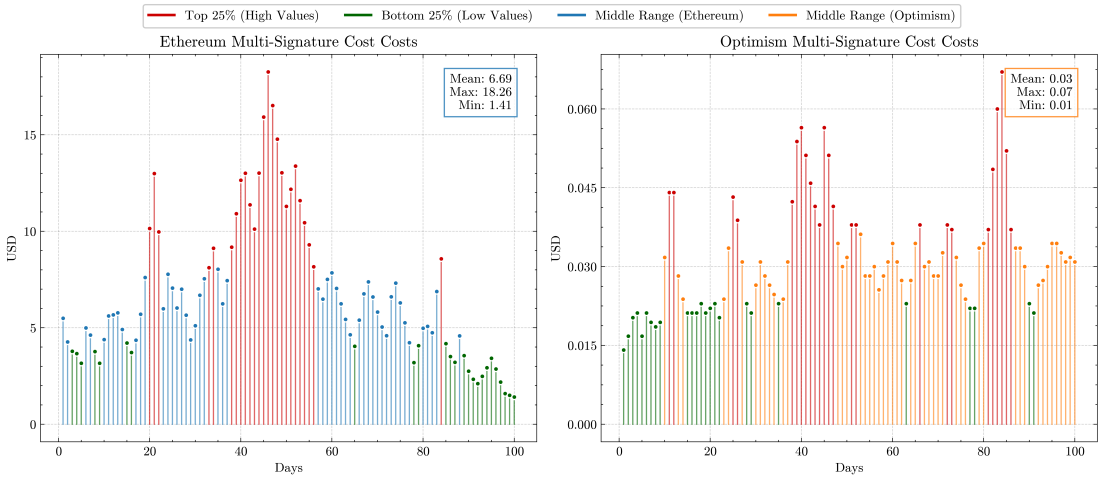

Figure 14: Multi-Signature Cost.

ing the network's consensus mechanism, transaction validation and storage, and native token functionality. *Layer 2* is a secondary framework built on top of an existing *Layer 1* blockchain to enhance the scalability and efficiency of the *Layer 1* blockchain without compromising its security or decentralization. It performs transaction validation and storage outside the *Layer 1* network but stores proof on it. The *Layer 2* solution handles more transactions per second, reducing transaction costs and speeding up confirmation times.

7.3 Multi-Signature Transaction Cost

The two entities must sign every access request. It costs for each multi-signature operation. Figure 14 shows the costs of the Ethereum and Optimism blockchain network. The prices fluctuate significantly, with a maximum of $18.26 and a minimum of $1.41 for Ethereum, as noted in Figure 14a. The average transaction cost over the 100 days is about $6.69. The graph shows several spikes, suggesting periods of high gas prices, possibly due to network congestion. Figure 14b shows the cost for Optimism, which is lower than on Ethereum, with values ranging from $0.068 to $0.013. The average cost is much lower at $0.03.

7.4 Time Requirements

Blockchain-based applications require block data writing and reading time requirements. Writing time includes smart contract deployment and data addition. Table 2 shows the writing time for various consent numbers for the test networks. The reading time indicates the required time to get data from the block of the blockchain ledger. All the read calls of smart contracts are gas-free. Table 3 shows the test network's reading time for various consent numbers. The same smart contracts and consents are used for all test networks. Maintaining a node locally can reduce the reading time from the network, where block data can be accessed in real-time. The system continuously synchronizes with the blockchain network to update the ledger data. The providers can maintain local nodes for faster authorizations.

Table 2: Writing Time to Blockchain Network.

Consents #	Polygon	Arbitrum	Optimism
4	5.256 Sec	4.519 Sec	8.167 Sec
8	6.329 Sec	6.713 Sec	8.926 Sec
12	6.653 Sec	6.907 Sec	7.156 Sec
16	5.923 Sec	4.683 Sec	7.692 Sec
20	7.465 Sec	6.651 Sec	8.426 Sec
24	5.562 Sec	6.098 Sec	7.318 Sec
28	10.927 Sec	2.142 Sec	8.925 Sec
32	10.518 Sec	4.782 Sec	8.145 Sec
36	10.637 Sec	6.872 Sec	7.562 Sec
40	11.268 Sec	4.329 Sec	7.498 Sec
44	12.519 Sec	7.602 Sec	7.387 Sec
48	13.876 Sec	5.274 Sec	8.156 Sec

Table 3: Reading Time from Blockchain Network.

Consents #	Polygon	Arbitrum	Optimism
4	0.357 Sec	0.265 Sec	0.378 Sec
8	0.352 Sec	0.231 Sec	0.329 Sec
12	0.467 Sec	0.276 Sec	0.398 Sec
16	0.394 Sec	0.246 Sec	0.571 Sec
20	0.331 Sec	0.276 Sec	0.603 Sec
24	0.354 Sec	0.215 Sec	0.613 Sec
28	0.329 Sec	0.234 Sec	0.423 Sec
32	0.426 Sec	0.247 Sec	0.612 Sec
36	0.353 Sec	0.265 Sec	0.376 Sec
40	0.436 Sec	0.291 Sec	0.602 Sec
44	0.524 Sec	0.221 Sec	0.421 Sec
48	0.462 Sec	0.237 Sec	0.342 Sec

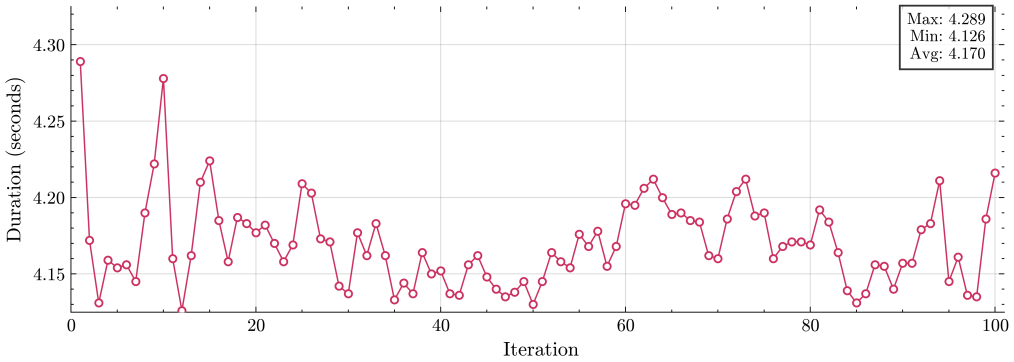

Figure 15: Compliance Block Construction Time.

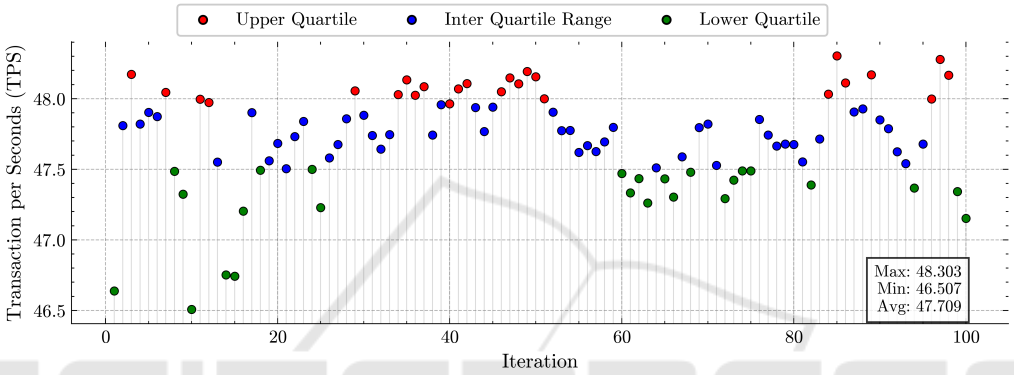

Figure 16: Compliance Checking Throughput.

7.5 Compliance Block Creation Time

This time measurement pertains to the duration required to confirm a compliance block after completing compliance checking and block finalization. The *Auditor* nodes perform compliance verification and make final decisions regarding the compliance status. In contrast, the *Committer* nodes are responsible for finalizing the block by recording it on the compliance blockchain ledger. This metric does not account for the time the *Orderer* nodes need to retrieve audit logs from the private audit blockchain, obtain informed consent from the public blockchain network, or gather relevant policies from the policy repository. Figure 15 illustrates the compliance block construction times, with a maximum of 4.289, a minimum of 4.126, and an average of 4.170 seconds.

7.6 Compliance Checking Throughput

The throughput, measured as the number of transactions per second (*TPS*), reflects the processing capacity after all required operations are completed within the *Proof of Compliance* consensus mechanism. These operations include compliance verification and the finalization of compliance blocks by the *Auditor* and *Committer* nodes. Figure 16 shows that each transaction represents an audit log. The throughput statistics show a maximum of 48.303, a minimum of 46.507, and an average throughput of 47.709 *TPS*.

8 CONCLUSIONS

In conclusion, the proposed blockchain-based policy compliance framework for emergency access to protected health information (PHI) addresses critical challenges in ensuring data security and patient privacy during medical emergencies. By implementing a multi-signature transaction system and maintaining immutable audit logs, the framework enhances accountability and transparency in accessing sensitive data. Furthermore, integrating a Proof of Compliance consensus mechanism ensures adherence to regulatory requirements, safeguarding patient rights while facilitating timely medical interventions.

Future studies could investigate the integration of advanced technologies, such as artificial intelligence and machine learning, into existing blockchain-based compliance frameworks. This research could enhance the efficiency of compliance checking and auditing processes and improve the detection of unauthorized

access attempts in real-time, thereby strengthening the overall security of emergency PHI access.

ACKNOWLEDGEMENTS

This work was partially supported by the U.S. National Science Foundation under Grant No. 1822118 and 2226232, the member partners of the NSF IUCRC Center for Cyber Security Analytics and Automation – Statnett, AMI, NewPush, Cyber Risk Research, NIST, and ARL – the State of Colorado (grant #SB 18-086), and the authors' institutions. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or other organizations and agencies.

REFERENCES

Aitzhan, N. Z. and Svetinovic, D. (2016). Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE transactions on dependable and secure computing*, 15(5):840–852.

Al Amin, M., Altarawneh, A., and Ray., I. (2023). Informed consent as patient driven policy for clinical diagnosis and treatment: A smart contract based approach. In *Proceedings of the 20th International Conference on Security and Cryptography - SECRYPT*, pages 159–170. INSTICC, SciTePress.

Al Amin, M., Tummala, H., Shah, R., and Ray., I. (2024). Balancing patient privacy and health data security: The role of compliance in protected health information (phi) sharing. In *Proceedings of the 21st International Conference on Security and Cryptography - SECRYPT*, pages 211–223. INSTICC, SciTePress.

Albrecht, J. S., Khokhar, B., Pradel, F., Campbell, M., Palmer, J., Harris, I., and Palumbo, F. (2015). Perceptions of patient provider agreements. *Journal of Pharmaceutical Health Services Research*, 6(3):139–144.

Aski, V., Dhaka, V. S., and Parashar, A. (2021). An attribute-based break-glass access control framework for medical emergencies. In *Innovations in Computational Intelligence and Computer Vision: Proceedings of ICICV 2020*, pages 587–595. Springer.

Bael, D. V., Kalantari, S., Put, A., and Decker, B. D. (2020). A context-aware break glass access control system for iot environments. In *7th International Conference on Internet of Things: Systems, Management, and Security (IOTSMS)*, pages 20–27. IEEE.

Conte de Leon, D., Stalick, A. Q., Jillepalli, A. A., Haney, M. A., and Sheldon, F. T. (2017). Blockchain: properties and misconceptions. *Asia Pacific Journal of Innovation and Entrepreneurship*, 11(3):286–300.

Fernández-Alemán, J. L., Señor, I. C., Lozoya, P. Á. O., and Toval, A. (2013). Security and privacy in electronic health records: A systematic literature review. *Journal of biomedical informatics*, 46(3):541–562.

Ferreira, A., Cruz-Correia, R., Antunes, L., Farinha, P., Oliveira-Palhares, E., Chadwick, D. W., and Costa-Pereira, A. (2006). How to break access control in a controlled manner. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pages 847–854. IEEE.

Gangwal, A., Gangavalli, H. R., and Thirupathi, A. (2023). A survey of layer-two blockchain protocols. *Journal of Network and Computer Applications*, 209:103539.

Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., and Gervais, A. (2020). Sok: Layer-two blockchain protocols. In *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*, pages 201–226. Springer.

Haeberlen, A., Kouznetsov, P., and Druschel, P. (2007). Peerreview: Practical accountability for distributed systems. *ACM SIGOPS operating systems review*, 41(6):175–188.

Kim, S. and Hwang, S. (2023). Etherdiffer: Differential testing on rpc services of ethereum nodes. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1333–1344.

King, J., Patel, V., Jamoom, E. W., and Furukawa, M. F. (2014). Clinical benefits of electronic health record use: national findings. *Health services research*, 49(1pt2):392–404.

Lee, W.-M. (2023). Using the metamask crypto-wallet. In *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*, pages 111–144. Springer.

Loos, M. (2020). Break-glass access control systems in medical devices. *RTDS, WS 2020, Institute of Distributed Systems, Ulm University*. This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Menachemi, N. and Collum, T. H. (2011). Benefits and drawbacks of electronic health record systems. *Risk management and healthcare policy*, pages 47–55.

Mulamba, D. and Ray, I. (2017). Resilient reference monitor for distributed access control via moving target defense. In *Data and Applications Security and Privacy XXXI: 31st Annual IFIP WG 11.3 Conference, DBSec 2017, Philadelphia, PA, USA, July 19-21, 2017, Proceedings 31*, pages 20–40. Springer.

Schefer-Wenzl, S., Bukvova, H., and Strembeck, M. (2013). A review of delegation and break-glass models for flexible access control management. In *Proceedings of the International Conference on Security and Trust Management*, pages 1–12. University of Applied Sciences Campus Vienna and WU Vienna, Austria, Springer.

Yang, Y., Liu, X., and Deng, R. H. (2017). Lightweight break-glass access control system for healthcare internet-of-things. *IEEE Transactions on Industrial Informatics*, 14(8):3610–3617.