Apply Trust Computing and Privacy Preserving Smart Contracts to Manage, Share, and Analyze Multi-site Clinical Trial Data

Yusen Wu[1,2(✉)], Chao Liu[2], Lawrence Sebald[2], Phuong Nguyen[1,2], and Yelena Yesha[1]

[1] University of Miami, Coral Gables, FL 33146, USA
yxy806@miami.edu
[2] University of Maryland, Baltimore County, Halethorpe, MD 21227, USA
{ywu5,chaoliu717,lsebald1,phuong3}@umbc.edu

Abstract. Multi-site clinical trial systems face security challenges when streamlining *information sharing* while protecting patient privacy. In addition, patient enrollment, transparency, traceability, data integrity, and reporting in clinical trial systems are all critical aspects of maintaining data compliance. A Blockchain-based clinical trial framework has been proposed by lots of researchers and industrial companies recently, but its limitations of lack of data governance, limited confidentiality, and high communication overhead made data-sharing systems insecure and not efficient.

We propose Soteria, a privacy-preserving smart contracts framework, to manage, share and analyze clinical trial data on fabric private chaincode (FPC). Compared to public Blockchain, fabric has fewer participants with an efficient consensus protocol. Soteria consists of several modules: patient consent and clinical trial approval management chaincode, secure execution for confidential data sharing, API Gateway, and decentralized data governance with adaptive threshold signature (ATS). We implemented two versions of Soteria with non-SGX deploys on AWS blockchain and SGX-based on a local data center. We evaluated the response time for all of the access endpoints on AWS Managed Blockchain, and demonstrated the utilization of SGX-based smart contracts for data sharing and analysis.

AQ1

Keywords: Permissioned Blockchain · Healthcare · Smart contracts · Clinical trials · Patient consent

1 Introduction

Clinical trials are experiments done in clinical research (e.g., to determine the safety or effectiveness of drugs) that involve human subjects. Centralized clinical trial systems are commonly used but insecure and inefficient when managing and

I. Awan et al. (Eds.): Deep-BDB 2022, LNNS 541, pp. 1–12, 2023.
https://doi.org/10.1007/978-3-031-16035-6_1

sharing data across multiple disparate organizations, and it is difficult without compromising patient and data privacy. In addition, patient enrollment, data confidentiality, and privacy, traceability, data integrity, and reporting in centralized systems are all critical aspects to maintain data compliance. Traditional solutions use informed consent [7] or electronic consents (E-consents) to create a process of communication between patients and health care providers that often generates agreement or permission for care, treatment, or services. As every patient owns the right to ask questions or get all sensitive information before treatment, current electronic documents, such as E-consents, are just electronic paperwork where the centralized signatures can lead to a lack of traceability and trustworthiness. Furthermore, multiple parties, such as hospitals, can not audit consents stored in electronic medical records (e.g., EMRs [13]), and also the clinical research generally is managed in local systems, such as REDCap [8].

Permissioned Blockchain, such as Hyperledger Fabric [12], is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a decentralized network. Its advantages of immutability, visibility, and traceability bring unprecedented trust to sensitive data, for example, recording medical transactions in a multi-copy, immutable ledger shared with different organizations. In fact, Blockchain has seen adoption by a wide range of applications and uses in healthcare recently, such as Akiri [1], BurstIQ [3], Factom [5], etc. These applications have implemented different types of functions in healthcare with both public and permissioned Blockchain, for example, keeping a decentralized and transparent log of all patient data in permissioned Blockchain to share information quickly and safely or verify the sources and destinations of data in real-time.

Leveraging Blockchain technology for informed consent processes and patient engagement in a clinical trial pilot is a new research field that is recently being proposed [23,28,29]. The rationale for the use of Blockchain technologies is to give patients control over who can access their data and when the consent expires. The innovation here is to surface data ownership, increase data confidence and prevent the leakage of sensitive information. Current work, however, has at least the following limitations:

(L1): **Heavy communication overhead**. Some patient consents and clinical trials are stored in a public Blockchain, such as Ethereum. The patient consents in a public Blockchain are shared with all the organizations or users of that Blockchain transparently; sensitive data must be encrypted via cryptographic functions and only the user who gets the private key can access the *ciphertext*. These public Blockchains could have the most negative impact on data sharing, their limited scalability and speed are core limitations. A public Blockchain network typically requires all the nodes to validate transactions; the consensus and validation of all the nodes in a network increase the usage of storage, bandwidth, and communication costs.

(L2): **Limited confidentiality for public smart contract**. The advantages of using a permissioned Blockchain to store patient data are explicit. For example, as a new member needs to be invited and approved by a plu-

rality of participants, and typically there are fewer participants in the permissioned Blockchain, the communication overhead is more efficient than on a public Blockchain. A permissioned Blockchain, however, is less resistant to malicious attacks, abusive behaviors, and arbitrary faults. For instance, a smart contract on a permissioned Blockchain can not keep a secret because its data is replicated on all peer nodes. A trusted member, though a majority of the participants accepted the invitations, can easily get access to the smart contract and distribute sensitive data to a third party.

(L3): **Lack of data governance**. Most of the applications and research papers in Blockchain with healthcare lack clear *data governance*. Data governance in our platform is a way that investigators[1] (e.g., attending doctors, patients, or directors) in the system can decide whether a sensitive record can be stored in the ledger with a valid signature or not, or grant permission to other users for access.

To remedy the current limitations, we first propose to use Fabric Private Chaincode (FPC) and Trusted Execution Environments (TEEs), in particular Intel Software Guard Extensions (SGX), to protect the privacy of chaincode data and computation from potentially curious peers in the same Blockchain network. We also propose adaptive threshold signature (ATS) to strengthen data governance. We list the advantages as follows:

Confidential and integrity-protected chaincodes. Fabric Private Chaincode has designed a secure solution for a smart contract executing on a permissioned Blockchain using Intel SGX. The outputs of a consensus algorithm are always final, which avoids the protocol-inherent rollback attack [15,24]. In addition, FPC extends Hyperledger Fabric Blockchain (Fabric) to execute a chaincode in an enclave and isolate the execution even from system applications, the OS, and hypervisor.

Confidential ledger data. FPC clients can send encrypted data to chaincodes inside an SGX enclave, then these chaincodes commit encrypted data as key-value pairs to the ledger. Enclaves can be programmed (and verified) to process and release data following regulatory compliance procedures[2].

Trusted channels for access control. FPC can establish secure channels for access control based on hardware attestation. Authorized members can be invited into different channels and members can only access the ledger of their own channels.

Reducing delegated privileges. FPC chaincode is an active actor that manages data compliance. It can prevent sharing and using data without prior consent, and it can prevent using data that does not belong to registered patients. Moreover, investigators for data governance and experimenters' actions are more constrained; investigators can not authorize data sharing for unapproved trials; experimenters cannot run arbitrary experiments; experimenters can not use

[1] An individual who conducts a clinical investigation.

[2] Regulatory compliance is an organization's adherence to laws, regulations, guidelines, and specifications relevant to its business processes.

arbitrary data. As we can see that FPC chaincode uses real-time compliance to reduce *trust-but-verify approach*[3] and delegated privileges.

Decentralized data governance. As we mentioned in limitation (L3), a decentralized governance system can guarantee that even if some investigators are faulty or offline, the transactions can still be delivered correctly, and the trials can be stored in the immutable ledger shared with different organizations only after a plurality of the investigators signed the clinical trials, that is to say, a trial needs to be certified by a majority of the investigators.

Contributions. We list the main contributions of this research here:

- We propose Soteria, an FPC-based, clinical trials sharing platform, using SGX-based chaincodes and private ledgers.
- We implemented an API to verify the FPC client's requests, and only the verified requests can be committed to the enclave chaincodes and be stored in the ledger. The API can wire up all the functions and components between the front-end and back-end (a Blockchain network).
- In order to eliminate centralized data governance, we use an adaptive threshold signature to strengthen decentralized data governance in clinical trials to tolerate arbitrary faults between different investigators.
- We finally evaluated Soteria including the latency of SGX-based endorsement, the response time (GET/POST) of clinical trials on AWS cloud, and our local Intel clusters.

Organization. Section 2 introduces related work. We introduce detailed privacy-preserving patient consent and IRB chaincodes for better explaining the role of FPC and the framework of the entire system in Sect. 3. A detailed IRB clinical trial example will be discussed in Sect. 4. We introduce the implementations in Sect. 5, evaluation of Soteria in Sect. 6.1, discussion in Sect. 6.2, and conclusion in Sect. 7.

2 Related Work

A number of researchers have highlighted the potential of using Blockchain technology to address existing challenges in healthcare. For instance, Mettler [26] aims to illustrate possible influences, goals, and potentials connected to Blockchain technology with healthcare, he implemented a smart health management system with Blockchain to fight counterfeit drugs in the pharmaceutical industry. McGhin [25] listed some security challenges in healthcare, such as access control, authentication, and non-repudiation of records, and proposed using the Blockchain network as the underlying approach to manage data securely. J. Gordon [20] proposed to use Blockchain for facilitating the transition to patient-driven interoperability through the benefit of data management mechanisms of Blockchain. Dwivedi [17] proposed a decentralized privacy-preserving healthcare Blockchain system for IoT, in which he eliminated the concept of PoW to

[3] https://en.wikipedia.org/wiki/Trust_but_verify.

make it suitable for smart IoT devices. Yesha proposed Chios [16], a lightweight permissioned publish/subscribe system, to securely collect HL7 format data in healthcare or other formats of medical records. One sub-module of the Chios system can also tolerate Byzantine faults in distributed machine learning [30] when training a model shared with different organizations or hospitals. In addition, several papers have proposed to store patient consent in a Blockchain to improve the security of patient records, surface data ownership and increase data confidence [14,18,19,23,27]. Two recent papers this year [11,22] propose to use Blockchain and IoT to manage healthcare data.

The papers above are all mentioned data security in Blockchain, the limitations of data security and communication overhead still made the sensitive patient data vulnerable to malicious attacks. As a result, applying privacy-preserving smart contract to manage, share, and analyze sensitive data become necessary.

3 The Soteria System

Soteria framework provides a modular framework allowing trade-offs between functionality, security, and efficiency. Soteria currently supports three main modules, including patient consent and clinical trial chaincode, API gateway, and decentralized data governance, as shown in Fig. 1. We describe the detailed functions of these modules as follows.

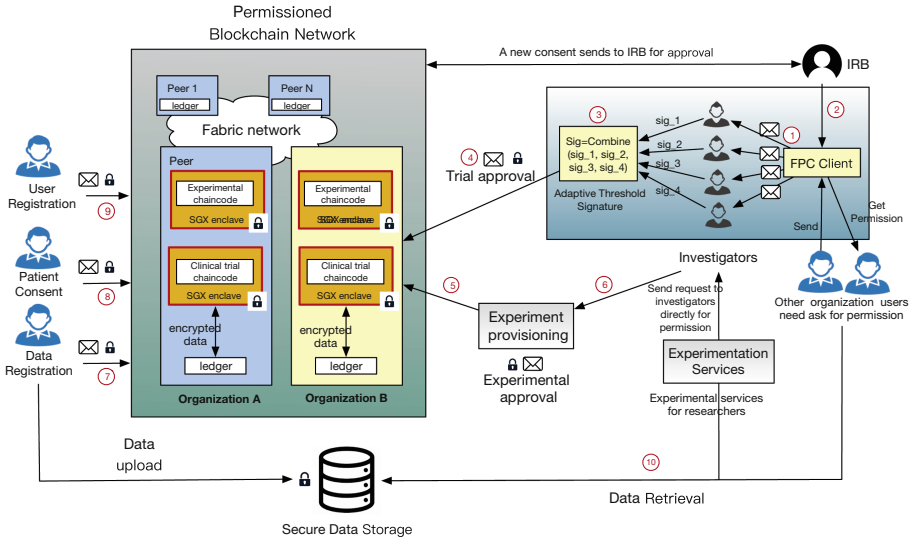

Fig. 1. Soteria framework and workflow. A detailed IRB demo for secure data sharing is given in Sect. 4.

3.1 Patient Consent and Clinical Trials Chaincode

Our patient consent chaincode includes eight main functions for interacting with the fabric client and Blockchain ledgers, such as CreateRecord or QueryConsentByID. Patient consent can be stored in the ledger and queried by a patient identity or record ID from the ledger. All the consents have a start time and an end time for legal access. Patient consent can be revoked by the admins or the patient, but it needs to be approved by half of the investigators (data governance).

The clinical trials chaincode includes nine main functions for interacting with the fabric client and Blockchain ledger. A trial needs to be registered and signed before being in the ledger; it can be queried by institution id or patient id; investigators can change a trial's status to Approved, Pending, Completed, and Revoked after being verified.

3.2 API Gateway

API Gateway is an independent module deployed outside the Blockchain network as a middleware application written in NodeJS. We deploy it through the AWS Serverless architecture. The API can help enroll a user, assign secret keys for a specific user, and load the Fabric client via a connection profile. Then, the front end can send GET and POST requests to the chaincode through different endpoints to register users, query patient records, create patient consent, and so on.

3.3 Decentralized Governance with (t, n) Adaptive Threshold Signature

We use the (t, n) adaptive threshold signature scheme as the core function for governing the clinical trials as the trials need to be signed before storing. The detailed steps are shown as follows.

Step 1: Parameters generation phase. A group of investigators generates two key pairs, one is *yes* key and another for *no* key. The *yes* key pair is $(pri_i^{yes}, vk_i^{yes})$ where a investigator keeps the pri_i^{yes} secret and publish its public verification key vk_i^{yes}. Similarly, the group of investigators generates the *no* key pairs (pri_i^{no}, vk_i^{no}). Every investigator in this step generates two key pairs *yes/no*, the *yes* key pair is for confirming that the message is valid and investigators can use the key pri_i^{yes} to sign the message.

Step 2: Transaction submitting phase. A fabric client wants to submit a message m to the Blockchain network. The investigator group consists of multiple n members $c1$, $c2$, ..., cn. We take four members as an example here. Fabric client calculates the hash code of the message $hash(m)$ or $h(m)$. Finally, the fabric client sends $< m, h(m) >$ to each of the investigators in the group.

Step 3: Sign a signature. All the investigators received the message m and its hash value $h(m)$. First, they will check the content of the message m (e.g., recalculate hash code of m, $h'(m)$). If a member c_i confirms that this message is

valid ($h(m) == h'(m)$), then c_i uses its private *yes* key to sign the message and generates the share signature $sig_i = sign(pri_i^{yes}, m)$. After that, the c_i send this share signature sig_i to the consensus node to vote.

Step 4: Make a consensus and deliver the result. A consensus node (an investigator) will receive the message m, $h(m)$, and share signature sig_i from the Fabric client after signing the signature. The node can first verify this share signature sig_i with the c_i's two public verification keys vk_i^{yes} and vk_i^{no}.

When the consensus node receives more than t thresholds (t, n) from the all the peer investigators, that is the *yes* or *no* number in an array, the consensus node can run the combination algorithm to recover the final signature $fsig = combine(sig_1, sig_2, ..., sig_t)$. Finally, the consensus node can verify the final signature with the public key pairs pk^{yes} or pk^{no}.

After this final signature is verified by the *yes* or *no* public key $true/false = verify(m, fsig, pk^{yes})$ (pk^{no}), the consensus node can determine if this message m (a patient trial) can be submitted to the Blockchain ledger or not. Figure 1 displays the message flow from the client sending the message to the Blockchain ledger. We give a detailed IRB example for data sharing and analysis in Sect. 4 for better understanding the Soteria framework.

4 A Detailed IRB Use Case for Data Sharing

We introduce a detailed IRB use case and its workflow in this section for a better understanding of Soteria architecture.

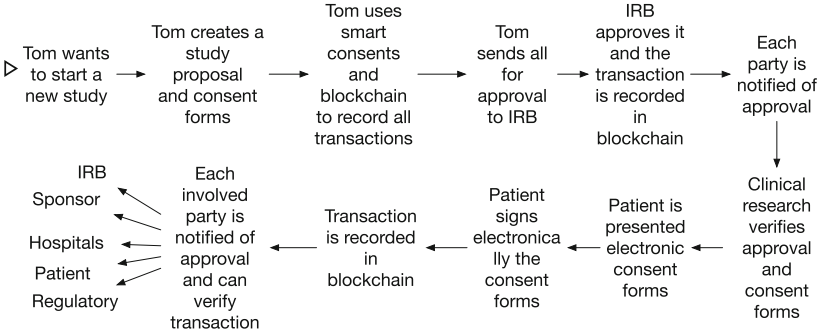

Fig. 2. Workflow overview.

Workflow Overview. We show an example of how a user Tom creates his clinical study, other involved parties get notified and verify transactions in Fig. 2. In this example, the role of Institutional Review Boards (IRBs) is for clinical trial approval, investigators conduct the clinical trials, and researchers typically monitor subjects and assess changes.

In our IRB demo, a patient needs to be registered before storing the patient consents and clinical trials (⑨). Every patient's sensitive data will be uploaded

to an AWS cloud database and only the experimenters or researchers who get permission and a private key from the Blockchain can retrieve data ((10)) from the AWS data storage. Patients can submit their consents ((8)) through user interfaces (a Web or an App) to grant the permit to experimenters or researchers. When an IRB member sends a clinical trial to FPC client ((2)), the FPC client will broadcast ((1)) the $h(m)$ and m to every investigators, then every investigator verify input trials m, sign a signature and send their shares σ_i to combination, function ((3)). If the output is *true*, this clinical trial will be successfully committed to the Blockchain ledger ((4)). The experimenters can be researchers or students, they need permission to access all the patient trials ((5) and (6)) for research. All of the researchers and experimenters can securely download patient data for their research after they get consent and private keys after getting approved by investigators.

5 Implementations

Soteria consists of a Golang/C++ module (chaincode), a Python module (for data analysis), a NodeJS module (API gateway), and a Terraform automation development tool with about 40,000 lines of code in total. We deploy our Soteria on AWS Managed Hyperledger Blockchain. The IRB/trials chaincodes are written in Golang. We also implemented a consent API in NodeJS with about 1,000 lines of code to interact with the front end. We implemented the front end through AWS Amplify.

We deploy SGX-based FPC locally in a simulation mode to evaluate the IRB. It allows for writing chaincode applications where the data is encrypted on the ledger and can only be accessed in clear by authorized parties. The SGX-based IRB chaincode is written in C++ with 1,000 lines of code because FPB only supports C/C++ language currently. Soteria client is written in Golang and we use gRPC [6] to commit transactions between different languages.

The Blockchain user is registered (created) in the Hyperledger Fabric Certificate Authority, and their enrollment credentials are stored in AWS Secrets Manager [2]. A corresponding user is also created within a Cognito User Pool [4], with a custom attribute, fabricUsername, that identifies this user within the Certificate Authority. Each portal attempts to authenticate the user (via username and password through sign-up or invite participant users) against a Cognito User Pool. Upon successful authentication, Cognito returns an identity token, which is a JSON Web Token (JWT). The client application includes this JWT in requests sent to the API Gateway, which authorizes the user to invoke the API route, as shown in Fig. 3.

API Gateway retrieves the fabricUsername custom attribute from the JWT, and sends this to the Lambda function that will be executing the Blockchain transaction. The Lambda retrieves the Blockchain user's private key from AWS Secrets Manager and retrieves the connection profile for connecting to the Amazon Managed Blockchain network from Amazon Systems Manager (Parameter

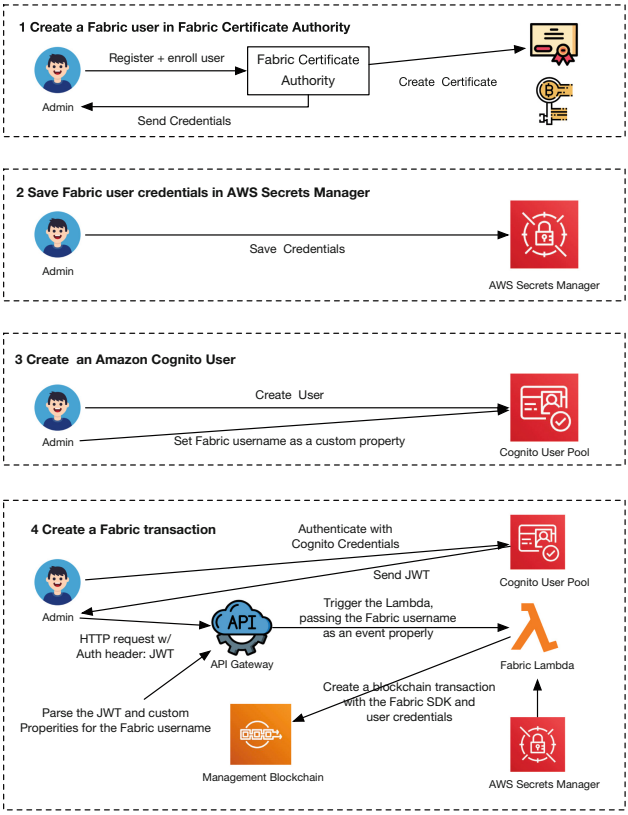

Fig. 3. Sequence diagram. This diagram shows the sequence of events that transpire to authenticate a user and invoke Blockchain transactions on their behalf.

Store). IAM policies [9] are used to restrict access to the Lambda function to only the Secrets Manager and Systems Manager [10]. The query and update functions are written in NodeJS using the Hyperledger Fabric NodeJS API. An AWS IAM user will be needed for provisioning the AWS Blockchain network. We also implemented an IAM policies sample which can be associated with this IAM user. Default IAM associated with users have credentials to bootstrap AWS managed Blockchain and other AWS resources.

6 Evaluations and Discussion

6.1 Evaluations

Experimental setup. For the non-SGX version chaincode as we said before, all of the functions are deployed on the AWS Managed Blockchain. An AWS IAM user will be needed for provisioning the AWS Blockchain network. The AWS Blockchain created a unique ordering service endpoint and VPC endpoint

Table 1. Endpoints response time.

Request	Method	T_W
Register	POST	233 ms
Consent Grant	POST	5.23 s
Consent PatientId	GET	623 ms
Consent Revoke	GET	256 ms
Consent Acknowledge	POST	516 ms
Consent	GET	3.91 s
Consent Validate StudyNumber	GET	4.64 s
IRB Trials (all trials)	GET	4.97 s
IRB Trials Register	POST	4.36 s
IRB Trials Status	POST	153 ms
IRB Trials Institutions	GET	4.51 s
IRB Trials Join	POST	751 ms
IRB Trials StudyNumber Status	POST	45 ms
Hospital Trials StudyNumber Invitation	POST	38 ms

for our access. We create one member and each member has a unique certificate authority endpoint and several peer nodes (peer endpoints). For SGX version chaincode, the code is managed on the Github[4].

Latency of SGX-based endorsement. We referenced the latency with an increasing number of clients from FPC [15]. The best endorsement latency is 8 and 16 clients (around 15 ms) and it starts to increase after 16 clients. The latency breakdown for submitting transactions with 4 clients showing the average response time as follows: the mean of *Decrypt* a transaction is 0.2 ms, *getState* is 0.37 ms, *Ledger enclave* time is 0.68 ms, and *Decryption* and *Verify* state time is 0.06 ms.

Latency of consent and clinical modules on AWS Managed Blockchain. After deploying the API and Blockchain correctly, all the endpoints can be accessed through GET/POST requests. It includes user registration, data confirmation, grant, queries, trial revoke, trial registration, trial validation, query registered institutions, query trials by institution, query all trials, invite participants to trial, update trial status, list participants by study number, acknowledgment, participants invitation, link registration, etc. We tested the main endpoints' response time in Table 1. T_w refers to a response time of each request in the WAN settings. We tested 10 times and take the average value. In addition, some requests' T_w , such as *Query* (GET), are affected by the number of patient trials stored in the ledger. Apparently, the time increases with more trials in store.

6.2 Security Discussion

For permissioned Blockchain and FPC. Though SGX encrypts sections of memory using security instructions native to the CPU, attackers inject malicious

[4] https://github.com/hyperledger/fabric-private-chaincode/tree/main/samples/demos/irb.

data into a running program, and stealing sensitive data and keys is possible. That's the reason we involve permissioned Blockchain and FPC as a private platform only for authorized organizations, and decentralized data governance for other uncertain third parties, then sensitive data can be securely exchanged between different hospitals and organizations.

For SGX. TEEs can not be directly used for *non-final* consensus protocols, such as PoW in Bitcoin or Ethereum, because TEEs generally are *stateless* [21], and it only works for the consensus decisions are *final*. As a consequence, we use TEEs in Fabric Blockchain because it supports *finality*. In each round, the BFT consensus always delivers a result, enclaves do not need to keep a state for the next round of consensus. By running all the ledger and smart contracts within an enclave, the smart contracts maintain confidentiality and secure chaincode execution.

7 Conclusions

We propose Soteria, an SGX-based privacy-preserving smart contracts framework for sensitive clinical trials in healthcare, including three main modules: patient consent and clinical trials chaincode, API gateway, and decentralized data governance. We evaluated the response time of clinical trials through the API endpoints and latency of SGX-based endorsement.

Acknowledgement. We gratefully acknowledge the support of the NSF through grant IIP-1919159. We also acknowledge the support of Andrew Weiss, and Mic Bowman from Intel.

References

1. Akiri. https://akiri.com/
2. AWS Secret Manager. https://aws.amazon.com/secrets-manager/
3. Burstiq. https://burstiq.com/
4. Cognito. https://docs.aws.amazon.com/cognito
5. Factom. https://www.factomprotocol.org/
6. grpc. https://grpc.io/
7. https://www.ama-assn.org/delivering-care/ethics/informed-consent
8. https://www.project-redcap.org/
9. Iam policy. https://docs.aws.amazon.com/iam/latest/userguide/access.html
10. Lambda. https://aws.amazon.com/lambda/
11. Adere, E.M.: Blockchain in healthcare and IoT: a systematic literature review. Array 100139 (2022)
12. Androulaki, E., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, pp. 1–15 (2018)
13. Bates, D.W., Ebell, M., Gotlieb, E., Zapp, J., Mullins, H.: A proposal for electronic medical records in US primary care. J. Am. Med. Inform. Assoc. **10**(1), 1–10 (2003)
14. Benchoufi, M., Porcher, R., Ravaud, P.: Blockchain protocols in clinical trials: transparency and traceability of consent. F1000Research 6 (2017)

15. Brandenburger, M., Cachin, C., Kapitza, R., Sorniotti, A.: Blockchain and trusted computing: problems, pitfalls, and a solution for hyperledger fabric. arXiv preprint arXiv:1805.08541 (2018)
16. Duan, S., et al.: Intrusion-tolerant and confidentiality-preserving publish/subscribe messaging. In: 2020 International Symposium on Reliable Distributed Systems (SRDS), pp. 319–328. IEEE (2020)
17. Dwivedi, A.D., Srivastava, G., Dhar, S., Singh, R.: A decentralized privacy-preserving healthcare blockchain for IoT. Sensors **19**(2), 326 (2019)
18. Genestier, P., et al.: Blockchain for consent management in the ehealth environment: a nugget for privacy and security challenges. J. Int. Soc. Telemed. eHealth **5**, GKR-e24 (2017)
19. Gilda, S., Mehrotra, M.: Blockchain for student data privacy and consent. In: 2018 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–5. IEEE (2018)
20. Gordon, W.J., Catalini, C.: Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. Comput. Struct. Biotechnol. J. **16**, 224–230 (2018)
21. Kaptchuk, G., Miers, I., Green, M.: Giving state to the stateless: augmenting trustworthy computation with ledgers. Cryptology ePrint Archive (2017)
22. Mamun, Q.: Blockchain technology in the future of healthcare. Smart Health **23**, 100223 (2022)
23. Mann, S.P., Savulescu, J., Ravaud, P., Benchoufi, M.: Blockchain, consent and prosent for medical research. J. Med. Ethics **47**(4), 244–250 (2021)
24. Matetic, S., et al.: {ROTE}: rollback protection for trusted execution. In: 26th {USENIX} Security Symposium ({USENIX} Security 17), pp. 1289–1306 (2017)
25. McGhin, T., Choo, K.-K.R., Liu, C.Z., He, D.: Blockchain in healthcare applications: research challenges and opportunities. J. Netw. Comput. Appl. **135**, 62–75 (2019)
26. Mettler, M.: Blockchain technology in healthcare: the revolution starts here. In: 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1–3. IEEE (2016)
27. Rantos, K., Drosatos, G., Demertzis, K., Ilioudis, C., Papanikolaou, A., Kritsas, A.: ADvoCATE: a consent management platform for personal data processing in the IoT using blockchain technology. In: Lanet, J.-L., Toma, C. (eds.) SECITC 2018. LNCS, vol. 11359, pp. 300–313. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12942-2_23
28. Rupasinghe, T., Burstein, F., Rudolph, C.: Blockchain based dynamic patient consent: a privacy-preserving data acquisition architecture for clinical data analytics (2019)
29. Tith, D., et al.: Patient consent management by a purpose-based consent model for electronic health record based on blockchain technology. Healthc. Inform. Res. **26**(4), 265–273 (2020)
30. Wu, Y., Chen, H., Wang, X., Liu, C., Nguyen, P., Yesha, Y.: Tolerating adversarial attacks and byzantine faults in distributed machine learning. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 3380–3389. IEEE (2021)