Blockchain Based Authentication and Authorization Framework for Remote Collaboration Systems

Logan Widick, Ishan Ranasinghe, Ram Dantu, Srikanth Jonnada
Department of Computer Science and Engineering
University of North Texas
Denton, Texas, USA
loganwidick@my.unt.edu, ishanranasinghearachchilage@my.unt.edu, ram.dantu@unt.edu, srikanthjonnada@my.unt.edu

***Abstract*— Due to the advantages of blockchain technologies, including decentralization, immutability, transparency and security, people try to replace existing problematic architectures /frameworks with blockchain based ones. In this paper we propose a novel authentication and authorization framework based on blockchain technologies to control access to the resources of an IoT device. In this paper, we focus on devices such as the Cyber Handyman used in remote collaboration applications to develop our framework. We tested our smart contracts on the Ropsten test network. Our results showed that it can handle 25 service requests simultaneously.**

Keywords— Blockchain, Access Control, Authorization, Digital Certificate, Security, Remote Collaboration, IoT

I. INTRODUCTION

With the help of high-performance computers and artificial intelligence, inventors, manufacturers, and developers have built numerous Internet of Things (IoT) devices once believed impossible. With these technological advancements, manufacturers started to build smart devices targeting regular people to ease day to day activities. But many of these IoT devices do not have good security systems to protect against intruders. Many centralized IoT platforms do not even provide methods to authenticate users [1]. The IoT security market is expected to grow from USD 6.62 billion in 2017 to USD 29.02 billion by 2022 [2].

Despite the security issues with IoT devices, people tend to use these devices due to their enormous benefits. Most of the consumers who use these devices are not aware of the underlying technologies, and failure of any of these devices can cause enough disruption to a user's daily routine. Although this is the case, people must often use these devices in order to keep up with society. We proposed a low-cost remote collaboration system called a "Cyber Handyman" that can assist people at these difficult times [3] [4] [5]. We used an IoT device in this system to provide necessary information to the remote assistant. In our work [4] we proposed a security framework to enforce security for the Cyber Handyman. In this paper we propose a novel blockchain-based framework to provide authentication and authorization for our Cyber Handyman device.

The paper is organized as follows: In Section II we discuss the limitations of existing authentication and authorization frameworks and how blockchain technology can provide solutions to those problems. In Section III we show how our proposed framework differs from existing solutions. Section IV, we give the details of our blockchain-based authentication and authorization framework. In section V, we describe results we obtained. Finally, in Section VI we provide our conclusions and the direction of future work for this research study.

II. PROBLEM DEFINITION

Although online videos and documents provide a lots of information on how to fix and maintain appliances and other devices, many people lack the skills and experience required to follow the given instructions. Due to the increasing demand for available experts, long wait times, and enormous service charges, we must look for other solutions to help people maintain these devices. Our proposed remote collaboration system is an apt solution for this problem. Since we have mounted many sensors (high-definition webcam, LED lights, wheels, speaker and microphone) to our device to improve the helper's situational awareness of the worker's environment, and provide remote gesture capabilities, our Cyber Handyman device is open to security attacks [1][4][36]. The remote helper

Fig. 1. Cyber Handyman device with adjustable height

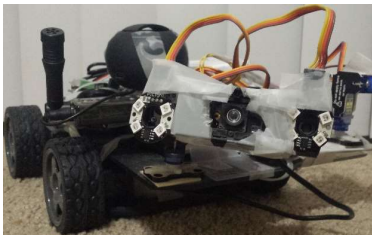

Fig. 2. Cyber Handyman device with minimal height

978-1-7281-0270-2/19/$31.00 ©2019 IEEE

should have access to all these resources via the internet to effectively guide the worker to accomplish the tasks at hand. Unauthorized access to these resources can violate the worker's privacy. Even an authenticated user can misuse Cyber Handyman resources that are not relevant to the task at hand. Hence, both authentication and authorization should be enforced in this system.

Many systems today use digital certificates to verify the users' identities. Digital certificates can be used to verify helpers in our Cyber Handyman system as well. These certificates bind a subject's cryptographic keypair to some details about the subject's identity and are signed (issued) by trusted third parties (TTPs) [14]. Typically, the trusted third parties can be peers (e.g. OpenPGP [15]), or centralized certificate authorities (CAs) (e.g. Verisign for X.509 certificates) [37]. However, these TTPs may make mistakes and misuse certificates. For example, TTP may issue a certificate to an imposter. When TTP misuses a certificate, few people may discover the mistake and even fewer could report the mistake or otherwise take action. If the misissuance is discovered, the logs of all actions that went into the misissuance, which might valuable in investigating why the incident happened and what can be done to prevent similar incidents in the future, are typically only stored by the TTP. In short, the lack of audit trails for these events makes it easier for imposters to take control of and misuse Cyber Handyman devices in ways that can violate workers' privacy and cause injury or damage. In this paper we propose a blockchain based methodology to address the above issues surrounding digital certificates.

Authorization is just as important as authentication. Due to numerous limitations of existing authorization frameworks [6] - [13] we proposed a novel OAuth based authorization framework for remote collaboration systems [4] such as Cyber Handyman. In the OAuth based framework we proposed, like in many other OAuth based frameworks, people must depend on the centralized authorization server to validate users and issue and validate access tokens. Any attack to the centralized server can compromise the whole system. Even a client making multiple requests can overload the server. In this work we replace the centralized authorization server with two smart contracts that perform the above tasks.

The transparent, tamper-evident, and decentralized nature of blockchains can help provide solutions to issues in existing authentication and authorization frameworks. In this paper we propose a novel framework to authenticate users and authorize access to resources of remote collaboration systems.

III. Related Work

A. Digital Certificates (Non-Blockchain)

Certificate Transparency (CT) [20] is a protocol designed by Google to help people detect misissued X.509 certificates. CT log servers maintain append-only logs of issued digital certificates and their certificate chains. Anyone can add a digital certificate and its chain to a CT log. CT auditors ensure that the CT log servers are functioning correctly and may check to see if certain certificates appear in the logs. CT monitors monitor the certificate chains that are added to the CT logs for signs of misissuance. Revocation Transparency (RT) [22] is a similar effort designed to provide transparency and auditability of revocation lists. However, the transparency and auditability of processes leading to updates to CT logs (e.g. submitting requests for certificates, gathering and analyzing evidence) or revocation lists (e.g. submitting requests for revocation, justifying and analyzing the justification of revocation requests) are out-of-scope of CT and RT. Reporting or otherwise acting on issues detected using CT and RT are out-of-scope [20] [22]. These limitations of CT and RT may allow unauthorized helpers that trick the TTPs to continue to control and misuse the workers' Cyber Handyman devices.

The Domain Name System (DNS) has a Certification Authority Authorization (CAA) resource record type [23]. A domain name owner would add a CAA DNS record that states which CAs can issue certificates for that domain, and how to submit Incident Object Description Exchange Format (IODEF) [24] reports to the domain name owner. Although industry guidelines such as [18] indicate that CAs should check for CAA records and report incidents accordingly, the complexity of the IODEF specifications likely disincentivizes full participation from both CAs and domain name owners. Also, CAA only works for domain names.

B. Digital Certificates (Blockchain)

The Instant Karma Protocol (IKP) [26][27] is a protocol that uses smart contracts on the Ethereum blockchain to not only replace CT, but also to specify what constitutes misissuance (via Domain Certificate Policies), and allow detectors (akin to CT monitors) to report imposters. IKP also allows domain owners to purchase single-use insurance policies (Reaction Policies) that provide payouts for reported misissued certificates to encourage reporting and to help domain owners recover after incidents. This system works well for use cases in which all attributes can easily be stored and processed on-chain (e.g. domain names). However, that is not the case for a helper's identity and qualifications. Also, the lack of audit trails for events leading up to the issuance and revocation may inhibit efforts to find out how to prevent incidents from reoccurring, which may threaten workers' safety, security, and privacy.

DeCert [39] is an Ethereum smart contract that allows users to add digital certificates to the blockchain, vote on those digital certificates, and search for digital certificates. A fork of Boulder's Let's Encrypt CA [38] pushes certificates issued to the blockchain. Other frontends and backends are included for user interaction and querying. This smart contract seems to be one of the few existing works that provide machine readable ways of reporting problems with logged certificates. However, DeCert only logs the resulting digital certificates and peoples' votes concerning the validity of said certificates. It does not provide any sort of audit trail for further investigation of incidents. In addition, on DeCert, the value of a vote is equal to the number of DeCert tokens spent on the vote. Thus, if a group of imposters acquires a lot of DeCert tokens, the imposters can vote for each other heavily to compromise the system.

Blockcerts [43] is an open standard for blockchain-based certificates, based on the W3C Verifiable Claims [44] and IMS Global Open Badge [45] standards. Each Blockcerts issuer would issue separate certificates for each thing about the subject that the issuer wanted to certify. For example, an online

instructor might issue one Blockcert to a student for each course the student took. Blockcerts does not attempt to "certify the mapping of public keys to individuals or organizations" or otherwise establish a subject's identity in any way [43]. Also, Blockcerts doesn't appear to be compatible with pre-existing certificate formats or log events other than issuance and revocation.

UPort [47] is attempting to establish a decentralized identity management solution using open standards on the Ethereum blockchain. For off-chain messaging, uPort uses JSON Web Tokens (JWTs) [46]. However, uPort doesn't appear to be compatible with pre-existing certificate formats or log events other than attestation and revocation.

Several startups are working on blockchain identity management solutions, including ShoCard [28] and Civic [29]. However, many of these startups appear to be creating their own credential formats that are not compatible with existing systems. Also, events leading up to the creation (or revocation, if necessary and supported) of attestations do not appear to be logged except by the attesters.

C. Blockchain-Based Access Control Framework

The Access Control Framework introduced by O. Novo [30] is closely related to our access control mechanism. The author introduced a blockchain-based access control framework for distributed sensor networks. The main difference of this work with respect to others [31] [32] [33] is that the author excluded the IoT devices from the blockchain and used an interface called a "Management Hub" to communicate to the IoT device. This is a great solution for conserving the limited resources of IoT devices. This architecture has a single Ethereum smart contract. However, our work is different from [30] in several ways. In [30], each IoT device is controlled by a single "manager" role and all managers use the same functions in smart contract. By contrast, our Cyber Handyman system has two roles ("worker" and "helper") with different functionalities. We implemented a role-based access control system to restrict access to the smart contract's functions. Moreover, in our system the owner of the device (the "worker") grants a remote expert (the "helper") access to certain resources of the device for a limited time. Furthermore, in [30] the managers are validated by their public keys. In our scenario the identity of the remote expert should be thoroughly verified, since an unlicensed professional can provide incorrect guidance to a worker, which may cause serious injury or property damage. Therefore, we used a novel digital certificate framework to verify the helpers.

D. Hwang et. al. [34] extended the study done by O. Novo [30] by introducing a dynamic access control scheme. One of the limitations of [30] is that the policies need to be generated in advance. The authors of [34] introduce a scheme that allows people to change the policies after requesting data. Although this is a great feature, the work done by D. Hwang at. el. is not sufficient to provide access control for our Cyber Handyman and similar remote collaboration systems. A. Z. Ourad at. el. [31] used a different approach to create their access control and authentication framework. In their architecture, upon an authentication request from a user, the smart contract validates the user's public key and broadcasts an access token to both the user and the IoT device. G. Papadodimas [35] also proposed a framework to share an IoT device's resources using access tokens. Both [31] and [35] included the IoT device on the blockchain. Moreover, [31] and [35] do not provide much user verification other than checking public keys and IP addresses.

IV. Methodology

In this section, we describe a novel framework for authentication and authorization for remote collaboration systems such as Cyber Handyman. This is inspired by the O. Novo's [30] decentralized access control framework.

A. Architecture

The architecture of our proposed framework is shown in Fig. 3. The components of our framework are as follows:

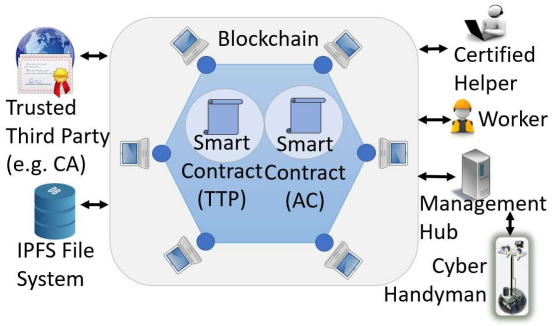

Fig. 3: Access Control Architecture

- **Worker**: Workers are people that need assistance. The public key of the worker's Ethereum wallet account is used to identify the worker. Each worker can only register one Cyber Handyman device.
- **Helper**: Helpers are experts in their fields. Only *certified helpers* (helpers that have digital certificates certifying their identities and qualifications logged in our TTP smart contract) can register as helpers. Helpers are also identified by the public keys of their Ethereum wallet accounts.
- **Management Hub**: The Management Hub is an interface between the blockchain network and the Cyber Handyman.
- **Cyber Handyman**: The Cyber Handyman is an IoT device with multiple resources that authorized helpers can use to obtain better situational awareness and provide instructions to the worker more efficiently. The Cyber Handyman talks to the blockchain through the Management Hub.
- **Trusted Third Party (TTP)**: Issues digital certificates that verify helpers' identities and qualifications.
- **InterPlanetary File System (IPFS)**[40]: Due to the transparent nature of the Ethereum blockchain and the high cost of on-chain storage, we only store metadata on-chain. Most data are stored in an off-chain distributed file system such as the InterPlanetary File System (IPFS)[40].
- **Reviewer**: Reviewers evaluate evidence supplied by helpers during the registration process.
- **Smart Contract**: Our blockchain based authentication and authorization framework consists of two smart

contracts. One smart contract handles digital certificate operations, while the other handles access control.

B. Event Flow

Our framework has four main phases. This section explains the event flow of each of these phases.

1) Deploying Smart Contracts

Our architecture includes two smart contracts. Both smart contracts (TTP – Trusted Third Party) and (AC - Access Control) are deployed by a single node called the "agent node".

2) Obtaining Digital Certificates

In our system we only allow certified helpers to assist the workers. Helpers are verified during the registration process. The digital certificate phase of the process is inspired by existing frameworks such as the Automated Certificate Management Environment (ACME)[41] but is designed to be digital certificate format-agnostic, use the Ethereum blockchain to provide a tamper-evident, auditable log of all steps (not just certificate issuance and revocation), and decentralize some processes (e.g. evidence review).

The steps in this phase are discussed below. If a certificate is misissued, people can report the misissuance using a similar procedure.

a) Create Request

First, the helper requests a certificate. The helper uploads the contents of the request (CertRequest) to IPFS, and puts the metadata and IPFS hash on the blockchain as shown in Fig. 4.

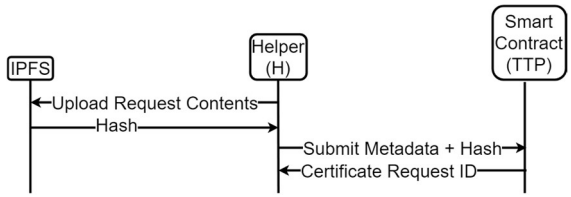

Fig. 4: Creating Certificate Request (CertRequest)

b) Add and Link Requirements

After the TTP retrieves the certificate request, the TTP decides what evidence requirements (EvidReqts) the helper must be met. For example, a TTP may require a helper to submit scans or video of existing identity or qualification documents for evaluation. The TTP then uploads the details of these requirements to IPFS and stores applicable metadata and IPFS hashes on the blockchain. Then, the TTP links these requirements to the certificate request on-chain. This is shown in Fig. 5.

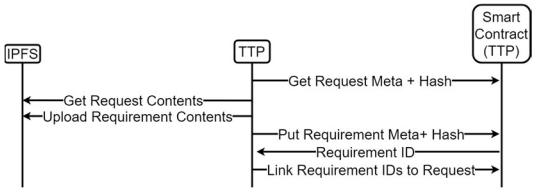

Fig. 5: Linking and Adding EvidReqts

c) Submit Evidence

In the next phase, the helper retrieves the updated request metadata from the blockchain, which now includes links to the EvidReqts. Upon retrieving the metadata from the blockchain the helper downloads the EvidReqt details from IPFS. The helper collects the required evidence, uploads the contents to IPFS, and stores the metadata and IPFS hash on the blockchain. This is shown in Fig. 6.

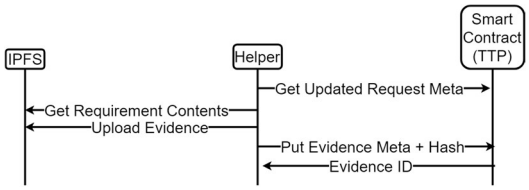

Fig. 6: Submit Evidence (Evid)

d) Review Evidence

After the helper has submitted evidence, the reviewers will retrieve the metadata (from blockchain) and evidence (from IPFS). Reviewers will then review the submitted evidence. For example, a reviewer familiar with the format of a state or province's occupational licenses may check submitted scans or videos of said licenses for signs of fraud. Or, such a reviewer may query databases from the applicable authorities.

After a reviewer has evaluated the evidence, the reviewer will write a review (EvidReview). This review will include a (positive or negative) confidence score for the evidence, and an explanation of the steps taken to reach that conclusion. Then, the reviewer will upload the review contents to IPFS and post the metadata and IPFS hash to the blockchain. After receiving the review metadata, the smart contract adds the product of the reviewer's confidence and the reviewer's trustworthiness to the score for that piece of evidence, and updates the status of that piece of evidence as appropriate. This is shown in Fig. 7.

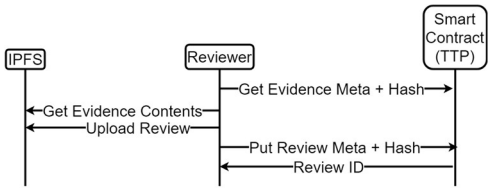

Fig. 7: Review Evidence (EvidReview)

In our system, we assume reviewers performed the steps and obtained the results described in their reviews. However, the log of all steps leading to certificate issuance (including evidence review) can allow investigators to discover dishonest reviewers. The dishonest reviewers' trustworthiness can then be reduced or set to zero as needed.

e) Submit Additional Data

After the reviews have been submitted, the helper can download updated evidence metadata from the blockchain. When the evidence has received enough positive reviews to be considered valid, the helper can submit additional data. For example, if the worker may need to interact with a Cyber Handyman that uses traditional technologies like X.509 digital certificates [37], this additional data may come in the form of a Certificate Signing Request (CSR)[19] that includes the Subject Public Key Info. The helper stores the additional data on IPFS

and sends a Certificate Request Update (CertRequestUpdate) to the blockchain with the metadata and IPFS hashes. This is

Fig. 8: Submit Additional Data

shown in Fig. 8.

f) Issue Certificate

After the subject updates the certificate request, the TTP may decide to post additional announcements to the certificate request in the form of other CertRequestUpdates. For example, if compatibility with CT [20] is desired, the TTP can post a precertificate. When the TTP is ready to issue a certificate, it can post a CertRequestUpdates for that as well. Regardless, the actual data for all CertRequestUpdates goes on IPFS, and metadata and hashes go on-chain. Fig. 9 depicts the TTP issuing a certificate, and the helper downloading the certificate.

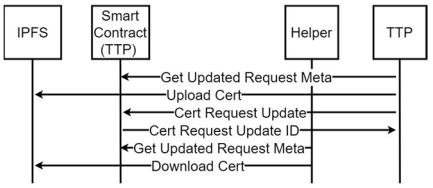

Fig. 9: Issue and Download Certificate

3) Worker, Helper, and IoT Device Registration

Anyone who has an Ethereum wallet can register. However, a helper must also have a digital certificate from our TTP smart contract and will provide the ID of this certificate when registering. The event flow for the registration process is shown in Fig. 10.

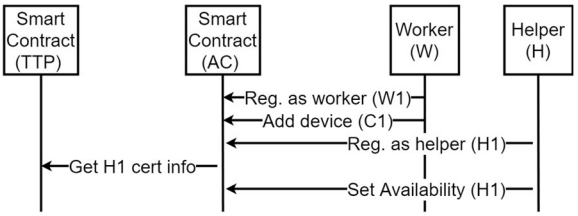

Fig. 10: Worker, helper and IoT device registration

4) Handling Service Requests

To request services from remote helpers, a worker and the worker's Cyber Handyman device should be registered in the system. After a worker submits a service request to the smart contract, the smart contract finds the next available remote helper. The worker can retrieve the helper's details stored on the blockchain (metadata and hashes) and IPFS (contents) as appropriate. This procedure is shown in Fig. 11.

After reviewing this information, the worker can choose whether to accept the allocated helper or chose another one. If the worker doesn't like the allocated helper, the system allocates

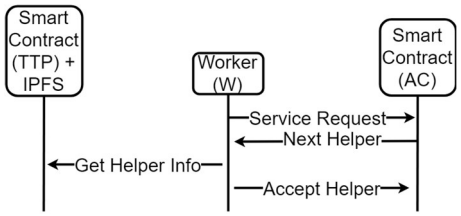

Fig. 11: Service Request and Helper Selection

the next available helper and sends the information to the worker. If the worker accepts the allocated helper, the smart contract generates an access token and broadcasts the token to the helper and the worker's Cyber Handyman device (through the management hub).

An access token includes the details of the helper, the duration of the task, and the resources that are required to perform the task. Upon receiving the access token, the Cyber Handyman device allows the allocated helper to access the device's resources as specified in the token. This procedure is shown in Fig. 12.

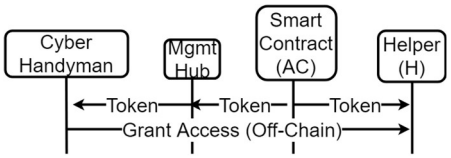

Fig. 12: Access Token Creation, Broadcasting, and Use

V. Results

We have tested our two Ethereum smart contracts separately. Both were written in the Solidity language [42].

A. Digital Certificate

We have tested this smart contract using the Ropsten testnet. For the first set of tests, we examined how the gas used by each type of operation changes based on changes in the input sizes. For example, we measured how the number of EvidReqts in an EvidReqtGroup influences the gas used to create and update the EvidReqtGroup, and how the number of EvidReqtGroups in a CertRequest influences the gas used to create and update the CertRequest. Some results from this set of tests are shown in the table below. From Table I. , the gas consumption of our smart contract operations scales well.

TABLE I. Gas Use of TTP Contract Operations

Operation	Average Gas Used
Submit Evidence Requirement (EvidReqt)	431825
Submit Evidence Requirement Group (EvidReqtGroup)	451179, 488267, 525338, 562410 for 1, 2, 3, and 4 EvidReqt instances per group, respectively
Submit Certificate Request (CertRequest) (including linking to EvidReqtGroups)	514216, 514301, 560492, and 606685 for 1, 2, 3, and 4 EvidReqtGroups per CertRequest, respectively
Submit Evidence (Evid)	363788
Submit Evidence Review (EvidReview)	383264
Submit Certificate Request Update (CertRequestUpdate)	406306, 417926, 442782, and 467738 for 1, 2, 3, and 4 EvidReqtGroups per CertRequest, respectively

For the next set of tests, we attempted to determine how well the costs of using our contract would scale as the number of simultaneous helpers increased. The cost of a transaction is the product of the gas used and the gas price. From the first set of tests, we determined that the gas used scaled well. Thus, in this set of tests, we sought to determine how the gas price changed as the number of simultaneous helpers increased. For this set of tests, we started with 2 helpers and took these users through each step of our processes, with a starting gas price of 1 gwei. If the helpers were able to complete our processes (get all transactions on the chain within 50 blocks of the submission time), we added 2 more helpers. If the helpers were not able to complete our processes, we increased the gas price by 1 gwei and repeated the experiment. From Fig. 13, the application can only handle about 25 helpers before the gas prices that the helpers must pay start to exponentially increase.

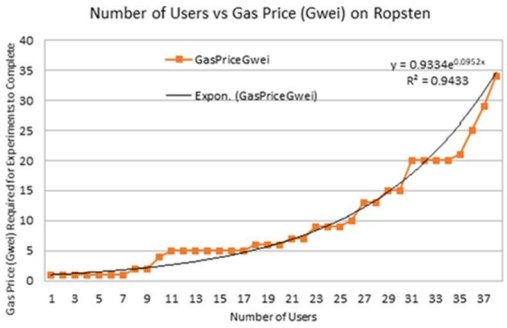

Fig. 13: TTP Contract Number of Helpers vs Gas Price (Gwei) on

B. *Access Control*

We created 106 accounts on the Ropsten testnet: 1 account for the agent node, 35 accounts for workers, 35 accounts for helpers and 35 accounts for Cyber Handyman devices. By incrementing number of users, we measure the gas used for each service request. The gas consumed ranged from 327887 to 937578. The gas consumption for each service request is shown in Fig. 14. We observed that the number of users causes gas consumption to increase linearly to quadratically.

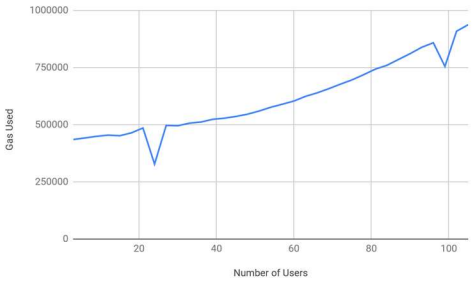

Fig. 14: Gas Consumption for Handling Service Requests

The performance of our system can be measured by latency [30]. We measured the time it takes to issue an access token to the remote helper and the device. The results we obtained are shown in Fig. 15. The time taken to issue a token ranged from 8.5 seconds to 106 seconds. The number of users may linearly affect the time taken to process service requests. However, there are additional variations likely due to other network issues, such as the total number of pending transactions and the competitiveness of the gas price offered by the service requester (worker) [35]. These additional variations lead to the sudden spikes in the figure as well. The average time taken to issue an access token was 42 seconds.

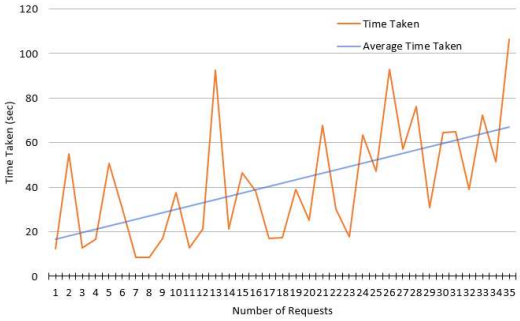

Fig. 15: Time Taken to Process Service Requests

VI. Conclusion and Future Work

Most authentication and authorization frameworks existing today depend on centralized servers. Such a server is a single point of failure. The transparent, tamper-evident, and decentralized nature of blockchains can help provide solutions to issues in existing authentication and authorization frameworks. In this paper we propose a novel framework to authenticate users and control access to remote collaboration systems. For user authentication, our framework can support existing digital certificate formats, and provides a decentralized audit trail for lifecycle events involving said certificates. We used access tokens generated by smart contracts to control access to the resources of our remote collaboration system. We believe our proposed framework can provide solutions to many limitations of existing authentication and authorization systems.

We created two smart contracts to authenticate users and control access to our remote collaboration system. One smart contract handles digital certificates and the other handles the registration, access control, and scheduling processes for the remote collaboration system. We tested our two smart contracts on the Ropsten test network. To evaluate our framework, we varied parameters such as the number of users in our system, the number of fields required in the digital certificates, and the number of types of evidence that helpers can submit for each field. We showed that the gas consumption scales linearly to quadratically at worst with respect to these parameters. However, more network capacity may be required to support more than a couple dozen participants with reasonable gas prices.

We would like to further extend our work to charge the workers for requested services. Moreover, we are planning on conducting a security analysis, and integrating our smart contracts with existing digital certificate management software. For security and privacy reasons, we would also like to investigate the feasibility of off-chain storage that is as decentralized as but more secure than IPFS. Possible approaches for such storage include those described in [16], [17], [21], and [25]. Furthermore, we would like to test our system with real participants on the Ethereum mainnet and collect feedback.

ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under awards 1241768 and 1637291.

REFERENCES

[1] K. Salah, "IOT Access control and Authentication Management via blockchain," in 2018 International Conference on Internet of Things (ICIOT 2018), Seattle, 2018.

[2] M. Rohan, "Internet of Things (IoT) Security Market worth 29.02 Billion USD by 2022," MarketsAndMarkets, [Online]. Available: https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf. [Accessed 03 Sept. 2018].

[3] S. Jonnada, Analysis And Performance Of A Cyber Human System And Protocols For Geographically Separated Collaborators, Ph.D. Dissertation, University of North Texas, August 2017.

[4] S. Jonnada, R. Dantu, P. Shrestha, I. Ranasinghe, and L. Widick, "An OAuth-Based Authorization Framework for Access Control in Remote Collaboration Systems," National Cyber Summit, IEEE Explore, Jun. 2018, pp. 38-44.

[5] S. Jonnada, R. Dantu, and I. Ranasinghe, "Cyber Handyman and Nursing for Humanitarian Services and Disaster Relief," in 2018 IEEE International Symposium on Technologies for Homeland Security (HST), 2018, pp. 53-56.

[6] D. Hardt, The OAuth 2.0 Authorization Framework, October 2012.

[7] Hakki C Cankaya, Access control lists, Encyclopedia of Cryptography and Security,Springer, 2011, pp. 9-12.

[8] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman, Role-based access control models, Computer 29 (1996), no. 2, 38-47.

[9] Vincent C Hu, D Richard Kuhn, and David F Ferraiolo, Attribute-based access control,Computer 48 (2015), no. 2, 85-88.

[10] Ludwig Seitz, Goran Selander, and Christian Gehrmann, Authorization framework for the internet-of-things, World of Wireless, Mobile and Multimedia Networks (WoWMoM),2013 IEEE 14th International Symposium and Workshops on a, IEEE, 2013, pp. 1-6.

[11] Sergio Gusmeroli, Salvatore Piccione, and Domenico Rotondi, A capability-based security approach to manage access control in the internet of things, Mathematical and Computer Modelling 58 (2013), no. 5, 1189-1205.

[12] F. Fernandez, A. Alonso, L. Marco, J. Salvachua, A Model To Enable Application-Scoped Access Control As A Service For IoT Using OAuth 2.0, 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), 2017, pp. 322-324.

[13] M. Noureddine, R. Bashroush, A Provisioning Model Towards OAuth 2.0 Performance Optimization, 10th International Conference on Cybernetic Intelligent Systems (CIS), 2011, pp. 76-80.

[14] N. Ferguson, B. Schneier, and T. Kohno, Cryptography Engineering: Design Principles and Practical Applications. Wiley, 2011.

[15] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, "OpenPGP Message Format," RFC Editor, Nov. 2007.

[16] S. Wang, Y. Zhang, and Y. Zhang, "A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems," IEEE Access, vol. 6, pp. 38437–38450, 2018.

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

[18] "Baseline Requirements Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates." CA/Browser Forum, 2019.

[19] M. Nystrom and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7," RFC Editor, Nov. 2000.

[20] B. Laurie, A. Langley, and E. Kasper, "Certificate Transparency," RFC Editor, Jun. 2013.

[21] M. Egorov, M. Wilkison, and D. Nunez, "NuCypher KMS: Decentralized key management system," Jul. 2017.

[22] B. Laurie and E. Kasper, "Revocation transparency," Google Res. Sept., pp. 0–2, 2012.

[23] P. Hallam-Baker and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record," RFC Editor, Jan. 2013.

[24] R. Danyliw, "The Incident Object Description Exchange Format Version 2," RFC Editor, Nov. 2016.

[25] M. Luongo and C. Pon, "The Keep Network: A Privacy Layer for Public Blockchains," 2017

[26] S. Matsumoto and R. M. Reischuk, "IKP: Turning a PKI Around with Blockchains." 2016.

[27] J. Fries and H. Niedermayer, "Using the blockchain to add automated financial incentives to the Public Key Infrastructure," in Proceedings of the Future Internet (FI) and Innovative Internet Technologies and Mobile Communication (IITM) Seminars, 2017.

[28] "Travel Identity of the Future," SITA, 2016.

[29] "Civic Whitepaper," Civic Technologies, 2017.

[30] O. Novo, "Blockchain Meets IoT: An Architecture for Scalable Access Management in Iot," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1184-1195, 2018.

[31] A. Z. Ourad, B. Belgacem and K. Salah, "IOT Access control and Authentication Management via blockchain," in 2018 International Conference on Internet of Things (ICIOT 2018), Seattle, 2018.

[32] S. S. Choi, J. W. Burm, W. Sung and Y. J. Heo, "A Blockchain-based Secure IoT Control Scheme," in 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE-2018), Paris, France, 2018.

[33] G. Papadodimas, G. Palaiokrasas, A. Litke and T. Varvarigou, "Implementation of smart contracts for blockchain based IoT applications," 2018 9th International Conference on the Network of the Future (NOF), Poznan, 2018, pp. 60-67.

[34] D. Hwang, J. Choi and K. Kim, "Dynamic Access Control Scheme for IoT Devices using Blockchain," 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2018, pp. 713-715.

[35] G. Papadodimas, G. Palaiokrasas, A. Litke and T. Varvarigou, "Implementation of smart contracts for blockchain based IoT applications," 2018 9th International Conference on the Network of the Future (NOF), Poznan, 2018, pp. 60-67.

[36] A. Ranjan, J. P. Birnholtz, and R. Balakrishnan, "An exploratory analysis of partner action and camera control in a video-mediated collaborative task," in 2006 20th anniversary conference on Computer supported cooperative work, 2006, pp. 403–412.

[37] IETF Tools, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," [Online]. Available: https://tools.ietf.org/html/rfc5280 [Accessed 26 Feb. 2019].

[38] Github, "Boulder". [Online]. Available: https://github.com/letsencrypt/boulder. [Accessed: 26-Feb-2019].

[39] L. Hentschker, "DeCERT : A Decentralized Certificate Authority," May 2018.

[40] J. Benet, "IPFS -Content Addressed, Versioned, P2P File System," arXiv Prepr. arXiv1407.3561, 2014.

[41] R. Barnes, J. Hoffman-Andrews, and J. Kasten, "Automatic Certificate Management Environment (ACME)," Dec. 2018.

[42] "Solidity — Solidity 0.5.4 documentation," 2019. [Online]. Available: https://solidity.readthedocs.io/en/v0.5.4/. [Accessed: 26-Feb-2019].

[43] "Blockcerts : The Open Initiative for Blockchain Certificates." [Online]. Available: http://www.blockcerts.org/. [Accessed: 22-Jan-2018]

[44] M. Sporny, D. Burnett, G. Kellogg, and D. Longley, "Verifiable Claims Data Model and Representations," Aug. 2017

[45] "Open Badges v2.0." IMS Global, 08-Mar-2017.

[46] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC Editor, May 2015.

[47] "A Complete List of uPort's Protocols, Libraries and Solutions," Medium, 20-Jun-2018. [Online]. Available: https://medium.com/uport/a-complete-list-of-uports-protocols-libraries-and-solutions-63e9b99b9fd6. [Accessed: 01-Mar-2019]