SAND2020-2956C

Designing a Decentralized Communication Platform Using Blockchain

ASHLEY MAYLE, University of New Mexico
JACLYNN STUBBS, Sandia National Laboratories
GABRIEL BIRCH, Sandia National Laboratories
MARIE VASEK, University College of London
SHUANG (SEAN) LUAN, University of New Mexico

Existing communication protocols in security networks are highly centralized. While this naively makes the controls easier to physically secure, external actors require fewer resources to disrupt the system. We present a solution to this problem using a proof-of-work-based blockchain implementation built on Multichain. We construct a testbed network containing two types of data input: visual imagers and microwave sensor information. These data types are ubiquitous in perimeter intrusion detection security systems and allow a realistic representation of a real-world network architecture. The cameras in this system use an object detection algorithm to find important targets in the scene. The raw data from the camera and the outputs from the detection algorithm are then placed in a transaction on the distributed ledger. Similarly, microwave data is used to detect relevant events and are placed in a transaction. These transactions are then bundled into blocks and broadcast to the rest of the network using the Bitcoin-based Multichain protocol. We develop four tests to examine the security metrics of our network. We find that when compared to a centralized architecture our implementation provides a resiliency increase that is expected from a blockchain-based protocol without slowing the system so much that a human operator would notice. Furthermore, our approach is able to detect tampering in real time. Based on these results, we theorize that security networks in general could use a blockchain-based approach in a meaningful way.

ACM Reference Format:
Ashley Mayle, Jaclynn Stubbs, Gabriel Birch, Marie Vasek, and Shuang (Sean) Luan. 2020. Designing a Decentralized Communication Platform Using Blockchain. 1, 1 (March 2020), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Blockchains are often thought to be synonymous with cryptocurrencies such as Bitcoin and Ethereum. However, the applications for blockchains are much larger than implementing currencies and other financial applications. Many companies are currently using private blockchains for tracking their supply chains in a more accurate and secure manner [8]. Others are using blockchain to track the movements of employees in secure areas [8]. These use cases point to broader uses of private blockchains for increasing transparency, resiliency, and providing operational value difficult to achieve via traditional means. In this analysis, we evaluate the feasibility of a private blockchain communication network applied to physical security systems.

This work explores the application of a private blockchain built on MultiChain to collect and distribute information to multiple actors without sharing private data from any one actor. Currently an entity could share pieces of their data with others using something like a shared database. This does not protect the data from being changed or tampered with by a second entity. Using a blockchain prevents unauthorized modifications of the data while allowing multiple untrusted parties to view historical data from multiple entities.

MultiChain is a private blockchain built on Bitcoin core using proof of work and round robin consensus [6]. We use MultiChain to create a decentralized network of cameras and sensors that are capable of communicating data with increased resiliency and prioritization to multiple actors at once. Prioritization implies that human interfaces into the network can easily differentiate between relevant and irrelevant information. This allows the different organizations to see the most relevant data first while still being able to view data deemed irrelevant by the algorithm if necessary. The camera data added to the blockchain includes outputs from the You Only Look Once (YOLO) real time object detection algorithm [10]. By using YOLO, images can be sorted by a computer depending on whether an object of importance was in the scene or not.

This approach uses smart filters, admin nodes, and member nodes on multiple blockchains that are loosely connected with each other. MultiChain implements smart filters which check if transactions meet certain rules before adding them to the block. These smart filters can facilitate novel things like segmentation of the network based on data type without causing a large overlap of different data types, causing problems with reviewing the data. They also allow the system to perform checks on aspects of a transaction autonomously without human involvement, adding additional resiliency without human interaction with the system. The admin nodes add permissions to the network as well as make and approve changes to smart filters to control the network. The member nodes act as validators for the system as well as being the nodes where data is generated and added to the system. Both of these nodes are important: admin nodes control the structure of the system and member nodes add more bulk. This allows us to strategically design the distrubted network where nodes with less privileges can be on the edges of the system completely unprotected and each organization can control an admin node to ensure agreement of what data should go where in the network.

Authors' addresses: Ashley Mayle University of New Mexico, amayle@unm.edu; Jaclynn Stubbs Sandia National Laboratories, jstubbs@sandia.gov; Gabriel Birch Sandia National Laboratories, gcbirch@sandia.gov; Marie Vasek University College of London, m.vasek@ucl.ac.uk; Shuang (Sean) Luan University of New Mexico, sluan@cs.unm.edu.

XXXX-XXXX/2020/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

We measure the security and resiliency of the network using four tests: a Admin Resiliency Test, Member Resiliency Test, Tamper Detection Test, and Bandwidth Comparison Test. The Admin and Member Resiliency Tests determine whether the network could continue working if some nodes were taken offline, both admin and member nodes. This was accomplished by turning off nodes systematically and trying to perform vital functions in the system such as adding new nodes and making transactions. We then measured how the network performed based on the ability to accept new nodes and new transactions. Our Tamper Detection Test used the tamper value on transactions (indicating that a camera was moved or something else about the node was changed) and sent transactions flagged as tampered to the network. We then measured if the network picked up the transaction. The Bandwidth Comparison Test measured the time it took to send multiple transactions through both the blockchain network and measured the amount of time it took for all transactions in the set to be added. After extensive testing in a variety of configurations, for the resiliency tests, we found our network architecture is able to remain online even when multiple nodes are shut down.

Our work makes the following contributions:

- We discuss whether applications need blockchains and evaluate other academic blockchain based approaches in different fields in Section 2.
- Section 3 thoroughly describes the architecture of our test network. We discuss other blockchain architectures that systems choose and motivate our use of MultiChain.
- The four resiliency and robustness tests of our system are proposed in Section 4 and then evaluated in Section 5. We find that our network is relatively resilient to attack, especially in contrast to a centralized system.
- We conclude with a discussion in future work in Section 6 where we deal with the real world implications of our model and look toward the additional testing. We find that our approach, while narrowly presented in this paper, can be broadly applied.

2 BACKGROUND

2.1 Applications of Blockchain Technology

Blockchain technology has been the technological backbone of cryptocurrencies since 2009. However, applications have expanded past cryptocurrency into a multitude of domains which can benefit from the immutable features of blockchain technology. Applications currently being researched include supply chain management, immutable audit logs, production systems, as well as many others [8]. Each type of application can give us interesting insights as to the proper use case for blockchain in a system.

However, one major problem with the rise of blockchain applications is that often a blockchain simply is not needed for the use case detailed by designers. Wüst and Gervais detail a model for how to decide if a blockchain is necessary [15]. The model is built upon characteristics of different blockchains and characterizes the needs that each type of blockchain could support. The main questions they suggest asking before using a blockchain are:

- Does state need to be stored?
- Are there multiple writers in the system?
- Can online Trusted Third Party always be used?
- Are all writers known and are they trusted?

Based on this framework, we will discuss other research on applying blockchains to cyberphysical systems.

The application MedRec [2] uses an Ethereum blockchain to communicate data ownership and viewership of medical records. The system uses regular databases to store the data off chain. The writers of the blockchain do not necessarily need to be trusted for the blockchain half of the system, but do need to be trusted for the regular DB half (where the medical records are stored). The need to trust the writers in this system for the whole system to work indicates that a blockchain might not be necessary for an idea like MedRec to be implemented.

Sikorski et al. [12] created a MultiChain blockchain to create a peer-to-peer electricity market. This market application uses existing architectures to make a cross-country currency to exchange electricity. This application did have the correct needs to use a blockchain as there were multiple untrusted actors where the state needed to be saved, as well as fulfilling other criteria.

Zakhary et al. [17] propose a system that employs numerous permissionless blockchains and a permissioned blockchain to facilitate global asset management. The permissionless blockchain part of the system would be suitable to use a blockchain as the writers are not known or trusted, there are multiple writers, and state needs to be stored. The permissioned portion of this system, however, does not necessitate a blockchain because the permissioned blockchain is supposed to have trusted writers.

Saberi et al. [11] discuss how blockchain could affect supply chain management. The authors do not discuss a specific application of blockchain, rather, they specify why a blockchain could be utilized. Based on the needs of the supply chain detailed in their paper, a blockchain would be appropriately used. This is because their system has multiple writers communicating internationally that cannot be trusted. The system also needs to have a true record of what occurred as the item moved through the chain.

Zyskind et al. [18] propose a system to manage personal privacy utilizing blockchain. This application used a blockchain appropriately as it needed to store state and there were multiple writers that were known but not all of which could be trusted. Karafiloski [7] discusses many applications of blockchain to big data problems including use cases similar to ones already discussed. The paper discusses proponents for each type of blockchain and why one would utilize them for a big data application.

Based on past work and Wüst and Gervais' [15] model for needing a blockchain, this work needs to ask do we need a blockchain? A physical security system does need to store state, it cannot use an always online trusted third party, there are multiple writers which are all known but are cannot necessarily be trusted. Because of this information about our system we know that a blockchain could be useful for our needs. Since this system could benefit from a blockchain based on the research, it is valuable to test if the blockchain architecture can support this type of system

2.2 Background of MultiChain

MultiChain is a private blockchain architecture based on the bitcoin core version 0.10 and was created in 2016[6]. This blockchain extends the Bitcoin core and so has integration with the Bitcoin core APIs. MultiChain is written in C++ and can be run on 64 bit architectures. The source code was a fork of the source code of bitcoin. MultiChain developers integrated updates to the bitcoin core which are necessary for private blockchain development. These updates include the creation of two types of nodes, an admin and member node. The admin nodes have permission to add new nodes to the blockchain

MultiChain adds functionality to the bitcoin core for private blockchains. This implementation extends the cryptographic handshake used in bitcoin to ensure that only permitted nodes are able to make peer-to-peer connections. Each node will share their public address that's on the permitted list; each node will verify all other nodes addresses on their version of the permitted list. Each node will then send a challenge message to every other node in the network. After the challenge message is sent each node will send back a signature of the challenge message showing they own the private key which corresponds to the public address. If at any of these steps a node fails, the peer-to-peer (p2p) connection will be terminated. This allows the network to only contain nodes that are known to at least one administrator. This allows nodes to still be mostly unknown without allowing an arbitrary node onto the network.

MultiChain uses a proof-of-work consensus like bitcoin to regulate and randomize the nodes rate of block production but enforces a round robin schedule to promote mining diversity so that one miner cannot monopolize the mining process. This is important in private blockchains because there are not as many participants (nodes or miners) as found in public blockchains. MultiChain allows an institution to create multiple private blockchains on the same machines meaning that there can be data flowing through different blockchains on the same network allowing nodes to pass data between different blockchains. This is useful in the case of having two blockchains where an input occurring on the first affects the behavior of the second. Finally, MultiChain supports multiple tokens on the same blockchain unlike the Bitcoin blockchain this allows an institution to only allow transactions on the blockchain that meet the value requirement for multiple tokens.

2.3 Current Security Systems

Physical security systems often contain many types of components serving many different functions including; delay, detection, assessment, and neutralization. Many of these functions now contain cyber components in connection to the physical components of the system[16]. These cyber components pose new risks to physical security systems such as the ability to remotely disabled physical components and deleting data. Current security systems handle these problems by determining the most important cyber pieces and placing more protections on those components. This work evaluates decentralizing the data sharing of the cyber components to prevent single points of failure.

Current physical security systems pose complex problems for system designers for many reasons. The first type of complexity that designers of security systems face is the diversity and adaptability of threats to a physical protection system [9]. This means that the components of the system must also be able to adapt and be diverse to counter threats. A second problem that designers of physical protection systems face is that the components of the system have varying interdependence with other components. This makes it difficult for designers to add in new components to the system because the relationships between all components is not straight forward. These two needs together show that blockchain is an effective addition to the security architecture because it is adaptable and can support diverse types of data in the same data log. The network will have two levels of dependence with the nodes connected to it. Either there will be an admin level of dependence meaning that the component will have more rights than the second level of dependence or there will be the member level of dependence which will have rights to connect and send and receive transactions.

3 METHODOLOGY

3.1 Selecting a Blockchain Implementation

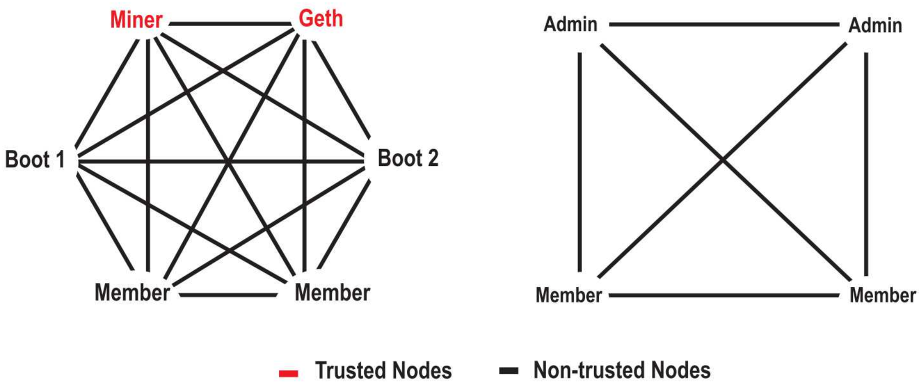
We chose to build out a large model for this system using MultiChain. The process for choosing MultiChain as our basis for this project is detailed in this section. In choosing a blockchain implementation with which one can build an application, there are many options depending on system needs. This work limits itself to blockchains mature enough to build upon as of implementation time in early 2019. We also decided against building our own blockchain from scratch, to leverage the existing extensive testing done on blockchains. One feature that needed for this system was the ability to support writing code to affect blockchain transactions. These criteria limited the blockchains under consideration to Ethereum[13] and the Bitcoin core-based MultiChain[6].

Since MultiChain and Ethereum have similar capabilities small test networks were developed using both networks to determine which implementation would work best for this use case.

The implementation of Ethereum used to test this section is the private version of Go Ethereum [13]. The Ethereum Blockchain has several types of nodes that can be used.

- A **geth node** is the beginning node that sets up the blockchain with the genesis block. This is where all of the parameters on the blockchain are described such as the gas limit, the mining difficulty, and the nonce which is what is used to distinguish this network as private. The nonce is a randomly generated hex value so that an attacker would have difficulty guessing it to connect to the network.
- A **miner node** is a node that can mine Ethereum for the network.
- A **boot node** is a node that is used to connect to the private network. This adds decentralization to how nodes connect to the network.
- A **member node** is node that is connected to the network and can send transactions.

The test setup used to determine if Ethereum should be used contained a geth node, a miner node, two boot nodes, and two



(a) Ethereum Network

(b) MultiChain Network (Note all nodes in this network are non-trusted)

Fig. 1. Small Test Networks using Ethereum and MultiChain

member nodes. This design assumes that the geth and miner node are trusted since this design does not have multiple geth nodes and the miner nodes need to be trusted in this setup since the miner will be the source of Ethereum so it needs to be trusted to distribute it in the correct manner. This design was chosen because it takes advantage of the native protection of having boot nodes while still being small enough to easily run tests on the implementation. The testing setup is shown in Figure 1(a).

The Multichain setup in Figure 1(b) starts with a single admin node and is built from there. Multichain has two types of nodes that can be used.

- A **member node** can validate, send, and receive transactions. They can also mine if given permission from an admin node.
- An **admin node** has all the abilities of a member node well as other abilities, like adding new nodes to the network.

The MultiChain network for this initial test is set up to have two admin nodes and two member nodes, as seen in Figure 1(b). This design allows the network to have enough nodes to turn some nodes off and still have the network be functional. Unlike our Ethereum-based network, the MultiChain network does not assume that any nodes are trusted because every node has other nodes which can perform its functions.

In both networks, we test the functionality of the network by designing two resiliancy tests for our example networks. One resiliancy test took down the connection node (boot node for Ethereum and admin node for MultiChain) and attempted to connect another node to the network. Another resiliancy test sent transactions to the both networks that should not have been accepted and measured the spread of this transaction. Both networks passed both simple tests. Based on these tests and the creation of both networks, we find that they both could perform the tasks required of the network. However, after using both networks, we found that the private Ethereum network has two drawbacks:

(1) it requires more resources to maintain than the MultiChain network
(2) it requires a more complex network structure

The system that we design requires ease of use and minimal addition of resources due to the embedded nature. With these restrictions in mind, we decided to build out our Multichain Network further.

3.2 Network

The notional network based on MultiChain is shown in Figure 2. Our larger test network uses 12 virtual machines running Ubuntu 16.04 created using MultiChain 2.0 community edition [5].

The MultiChain network in Figure 2 has two types of nodes; admin nodes and member nodes. The nodes are defined by what permissions they have. Member nodes have permissions to send, validate, and receive transactions; admin nodes have those permissions as well as the ability to create and approve filters and add new nodes to the chain. These filters check if transactions meet certain rules before adding them to the block. Our network also has two types of data input: microwave sensors and cameras. The MultiChain network for this project was setup to have five admin nodes and eight member nodes: nodes 0, 2, 5, 8, and 11 are admin nodes, nodes 3 and 4 are member nodes which add in imager data to the system, nodes 9 and 10 are member nodes which add in microwave data to the system, and all other nodes are member nodes with no extra functionalities.

Each color in Figure 2 refers to a different set of data that is communicated between that sub-network. The pink sub-network contains all data that originates from a camera and has no classification data attached to it. The green sub-network contains all data that originates from a camera which has a classification for an object in the scene. These types of data are on two different sub-networks so that people processing this data can focus their attention on the most relevant information.

The blue and purple sub-networks handle microwave data; the blue sub-network handles the data from the microwave that has

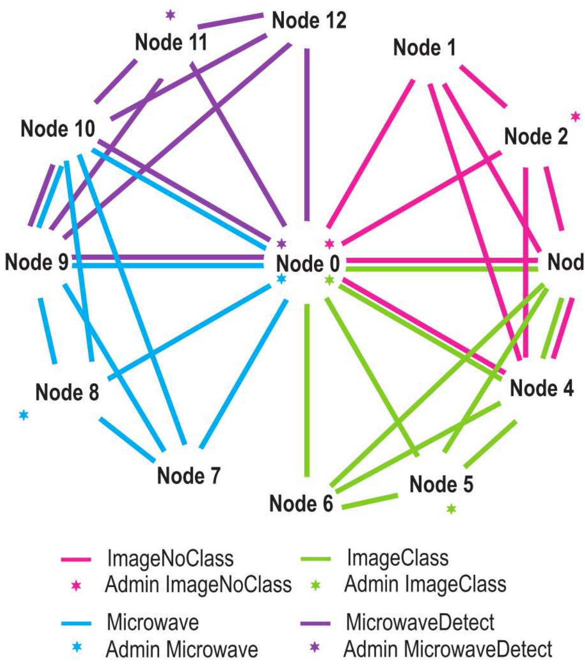


Fig. 2. Network Diagram for 12 nodes with two types of data input

pertinent information while purple sub-network handles the data from the microwave when there is not important information i.e. the rule set is not met for the microwave. These sub-networks are connected so that data can flow to either network depending on whether the input had relevant data or not.

This network can be expanded upon either by adding more inputs for the current types of data or by adding new types of data to be stored in the distributed ledger e.g. one could add data about employee movements, movement of materials, or data from other types of sensors. Adding this information would make the system more complex, but could add even more decentralization therefore requiring more resources to harm the system.

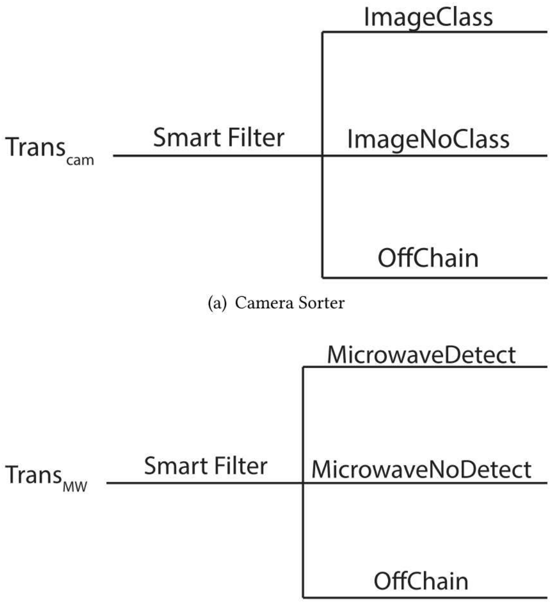
This setup was designed to take advantage of having a decentralized architecture; new nodes can still connect to the network and new filters can still be approved as long as some subset of admin node(s) are present. MultiChain has the advantage that no nodes in the system need to be trusted for the network to continue operations. As we will demonstrate later in Section 5, if an admin node is taken over, the other admin or admins can start a stream or even a new chain with the non-compromised nodes. This means that it would be difficult to perform an attack on this system by shutting off nodes without spending a large amount of resources.

3.3 Data Processing

Two types of sensing devices were used in this analysis: cameras and microwaves. Each camera sensor takes images of the object related to the microwave sensor and processes them using the You Only Look Once (YOLO) algorithm to detect classes of interest (e.g., people, vehicles, etc.). YOLO was chosen because it can process images in real time and give regional data for where a target is located within a scene. [10]. Microwaves are sensors which use switch closures to indicate that an object has passed through their line of detection.

3.4 Data Sorting

Figure 3 shows how the data from the sensors is sorted in this network using smart filters. A smart filter in MultiChain is code that is added to the chain and approved by the admin nodes. This code affects which transactions will be accepted by the chain. The implementation of smart filters in our system allows the data to be sorted to the appropriate channels depending on the values in the transaction. The smart filters also allow the system to ignore data that has been tampered with instead of letting it flood the system. An example of a transaction is listed below. Other types of transactions are listed in Appendix A.



(a) Camera Sorter

(b) Microwave Sorter

Fig. 3. Diagram of how data is sorted in the system

3.4.1 *Transactions.*

The transaction in listing 1 is sent through the camera sub-network. This transaction contains an image with an object detected in it. It is being sent to another address, in this case 1WfiQNZirYMwVqVgdKvaboupEpz3FWHzcAEzec, which is the location of another node on this blockchain. The smart filters are based off of assets which are spent to make transactions.

```
sendwithdata 1WfiQNZirYMwVqVgdKvaboupEpz3FWHzcAEzec
'{"tamperAsset":10, "classifyAsset":10}'
'{"json":
    {
    "filename":"/media/TA-V_FFMPEG_DATA/210/
      2019-05-07_14-53-09.png",
    "classify":"person",
    "percent":0.5402294993400574,
    "center x":1024.7264404296875,
    "center y":180.95614624023438,
    "width":16.506134033203125,
    "height":33.79193878173828
    }
}'
```

Listing 1. Transaction for camera with object detected

The assets being sent for this transaction are:

- the **tamper asset** which represents the value of whether the system has been tampered with or not. When a sensor is tampered with, a switch will change the value being sent to the network. A transaction that has not been tampered with will have a value of 10. If a transaction has been tampered with the value would be outside of 10.
- the **classify asset** is the asset that transactions with images must be sent with. This allows the system to determine which smart filter to apply to the transaction. In this case the smart filter that will be applied is the Classify filter shown in listing 2.

These values are both set to 10: the tamper asset is 10 so that the tamper smart filter will approve this as a valid transaction.

The data passed through in this transaction is represented by a JSON object. This contains a link to the image (stored in a video management system) being passed as well as the classification made by YOLO along with the percentage of certainty and location values. This is useful information for the operator to have as it will allow them to find objects in the image being referenced more easily making for a quicker assessment of the data. This transaction will be added to the ImageClass sub-network because the JSON object contains a value for classification; if it did not it would be sent to the ImageNoClass chain. JSON objects are an easy way to send multiple types of data making them ideal for large complex systems like the one described in this paper; an example of a JSON object being used for other types of data is shown in Appendix A.

3.4.2 Smart Filters. Each transaction is added to a specific blockchain based on the code written in the smart filters for that chain. Each smart filter affects transactions made on the blockchain that has the smart filter approved. A transaction can be sent to multiple blockchains which will sort the data accordingly. An example of a smart filter is shown below; all other smart filters for this system are in Appendix B.

```
Classify Filter function filtertransaction()
{
    var meta=getfiltertransaction();
    s=String(meta.vout[1].data[0].json.classify);
    if (s.valueOf() == " ".valueOf())
    {
        return "Classify not found"
    }
}
```

Listing 2. Smart filter to determine if an object is in an image

The Smart Filter in listing 2 checks to see if there is an object detected in an image. This code is written in JavaScript and checks that the classify value in the JSON object of the transaction contains a value from YOLO. This filter is placed on the Image Class sub-network. If the transaction fails this code check, the image will not be added to the Image Class chain and will need to go through another smart filter to be added on a different sub-network matching the data type and detection status.

With the smart filters and transactions set up the network ran with data being placed on the chain correctly and securely. This shows that the network can support thousands of transactions being placed on it at a rate that is fast enough that humans in the loop wouldn't be able to detect the difference in the time it takes for the data to be uploaded. Therefore it will be valuable to make some initial measurements of the system for quantitative analysis. Allowing us to determine if there are actually added security measures from the use of blockchain in the system.

4 SECURITY METRICS

This system utilizes the MultiChain network framework which contains admin nodes as well as member nodes. These nodes were connected together based on the framework described in Section 3.2. This framework allowed the system to communicate data effectively and securely as the data was sorted properly and quickly. It is important to have quantitative analysis for new distributed network architectures so that there can be proof that the system will work under specific conditions. This ensures that the system will perform as we expect it to during the worst circumstances. To make preliminary steps toward providing quantitative measurements for a blockchain-based data sharing network, we perform four tests on the system described above.

Admin Resiliency Test: We take down different combinations of admin nodes and measure whether new nodes can connect to the network. Admin nodes provide a multitude of important features for the system: admins control connection to the network, the control of the creation of code, the approval of code, and the permissions of each node in the system. If taking an admin node offline could take away any of these features, then the network would be susceptible to an easy attack. Being able to successfully run with fewer admins increases the robustness of the system. We perform this test by shutting down the virtual machines that host the admin nodes for each sub-network. This simulates the plug being pulled which would turn the node off and disconnect it from the network architecture. We test this for all admin nodes including turning all admins off.

Member Resiliency Test: We take down different combinations of member nodes and measure whether transactions can still be sent through the system. This test was important to show that a partial outage of the system would not remove vital capabilities from the entire system. For example if one half of the system shown in Figure 2 were to be taken off, the consequences for the other half of the network should be kept to a minimum. We test this by taking down a node one at a time and checking each time that the system continues to add transactions to the blockchain.

Tamper Detection Test: We detect tampering of the physical system by evaluating each transaction for a tamper value. These values can come from any node on any transaction. The tests involve sending a transaction with a tamper from admin nodes and member nodes. Rather than simply turning a node off, some threats can merely change a node's camera's direction or change how the sensor communicates the data to the network. We perform this test by changing the tamper values for the transactions to represent the change in value from the sensor.

Bandwidth Comparison Test: We measure the time it takes to send a multitude of data points to both the blockchain network and a basic SQL database. The database is set up on the same architecture as the blockchain network (Ubuntu 16.04 Virtual Machine). The tests involve using data sets from the same source changing the data to follow the format that is required by MySQL for the database and the JSON objects for the blockchain network. The data sets are sent through both the network and the database and they are timed using the time function from the kernel.

5 RESULTS

After performing four metric tests in a variety of different contexts (Table 1), we are able to see a path forward for scaling this to a full scale system.

The admin resiliency test shows that the network would continue to allow new regular and admin nodes to be created as well as the creation of new smart filters and the approval and falsification of older smart filters. As we can see in Table 1, this is true for each sub-network when any one admin node was taken down in the network and true for the whole network when up to four of the five admin nodes were taken down (depending on which set of nodes were taken down simultaneously). In a larger system, more admin nodes would be able to be taken down since the system would be distributed across more actors that could maintain functionality.

The member resiliency test shows that taking regular nodes offline still yields a network that is able to support the transactions required for the system to communicate data. We find that as long as two nodes in the sub network remained on, the network could still communicate the data as expected. This is important because if only one node needed to be taken down to cut communication, then the failure of a single sensor would compromise the operation of the entire network – an obvious failure for a secure system. Similarly, in a larger system, more nodes could be taken offline and this functionality would still work because there is more decentralization and simply more nodes to take offline in general.

The goal of the tamper detection test is to evaluate if the system could detect a tamper in the physical space. When a device is physically tampered with it will send a signal that it has been opened. This signal is represented in the system as the tamper asset. Code was written in this architecture to detect a change in this value from the expected value of 10 seen in Appendix B. If there is a change in the tamper asset for a transaction that transaction will be ignored by the system to prevent misinformation from making its way to a human for assessment.

This test revealed that the transactions could correctly identify tampering and ignore the data being sent. This simulated tamper assessment is similar to a real life tamper but would need to be tested further on a physical test bed to confirm the results given by the simulation.

The goal of the bandwidth comparison test is to evaluate if a blockchain decreases the speed of data flow enough for a human in the loop to notice from a database. We can see from the table that using the blockchain does slow the speed of data flow compared to the database but we also see that it is fast enough for human participants to receive multiple transactions per second (i.e 17000/any time value in figure 4) which means there will be no noticeable difference for the human in the loop while having the added security from a decentralized immutable data log. This test was setup by running thousands of transactions through the blockchain and the centralized database at different bandwidths and seeing how long the blockchain and centralized database took to store all of the data.

Overall the results from this work indicate that a blockchain-based communication network would be a reasonable step forward for a distributed data application. It brings many important advancements such as decentralization and immutable data logs to this test system. There are many things that can be tested in the future of this project. Some that are of particular interest include more resiliency metrics including tests on the absorptive, adaptive and restorative capacity. These metrics are discussed further in section 6.

6 CONCLUSION AND FUTURE WORK

Our work develops a blockchain-based approach to secure and promulgate data from different entities that wish to share data but do not trust the other entities with their entire sets of data. Our main result was showing that this was possible. After performing four tests on the network, we are able to verify that this architecture behaves similarly to public permissionless blockchains like Bitcoin.

The results discussed above show that this type of network has promise to be a future communication network for data sharing. We have shown that our network is resistant to power failures and attacks which take down parts of the system as one would need to take down many nodes to shut off communication to each entity. The results for taking down admin nodes show promise for the network being able to adapt to an attack with more work put into the system, as an admin node can add new nodes and change code such that an attacker may not be able to maintain control of enough of the network to perform an attack.

Another aspect of the system which was tested was the ability to detect tampers. This test showed that the system was able to correctly detect when a tamper was happening and disregard the data being sent in as it might be false. This will allow a human to assess the components of the system which has been tampered with before adding that information back into the network. The last aspect of the system that was tested was the comparison of blockchains and databases on speed of adding information to the system. This test showed that databases were faster but that blockchains still added multiple transactions per second meaning that the human in the loop would not be able to detect the difference in the amount of information shown to them on a second by second basis.

There are other benefits to the system which we believe to be true but have not been tested yet. Including that it will be hard to destroy or change historical data in the network so that when an audit occurs, accurate data is represented and so that if an insider tries to attack the system. It will be recorded accurately without being changed by said insider. A final benefit that this network structure appears to provide is that entities no longer need to rely on trust of the components and of each other. This will allow the system to be secure and resilient because components could break

Test	# of Attempts	# of Admins	# of Members	Can Connect	Can Send Transactions	Tamper Detected
Admin	5	5	7	✓		
	5	4	7	✓		
	5	3	7	✓	-	-
	5	2	7	✓	–	
	5	1	7	✓	-	-
	5	0	7	✗	-	-
Member	30	5	7	-	✓	-
	30	5	6	-	✓	-
	30	5	5	-	✓	-
	30	5	4	-	✓	-
	30	5	3	-	✓	-
	30	5	2	-	✓	-
	30	5	1	-	✓	-
	30	5	0	-	✓	-
	30	4	0	-	some	-
	30	3	0	-	some	-
	30	2	0	-	some	-
	30	1	0	-	✗	-
Tamper	50	5	7	-	-	✓

Table 1. Results after testing the first three tests in a variety of node combinations.

Mb/s	10	10	45	45	100	100
	DB (s)	BC(s)	DB (s)	BC (s)	DB (s)	BC (s)
Run 1	58	1533	58	455	56	411
Run 2	59	447	63	423	61	438
Run 3	59	452	62	424	60	439
Run 4	58	443	62	421	59	424
Run 5	59	436	61	422	58	429
Run 6	58	453	58	426	55	425
Run 7	58	453	60	428	58s	429
Run 8	57	455	60s	429	60	435
Run 9	58	441	57	433	59	423
Run 10	57	447	57	430	58	411
Average	58.1	556	59.8s	429.1	58.4	426.4
σ	0.73	343.33	2.201	9.87	1.84	9.86

Fig. 4. Run, average and standard deviation from comparison test on bandwidth

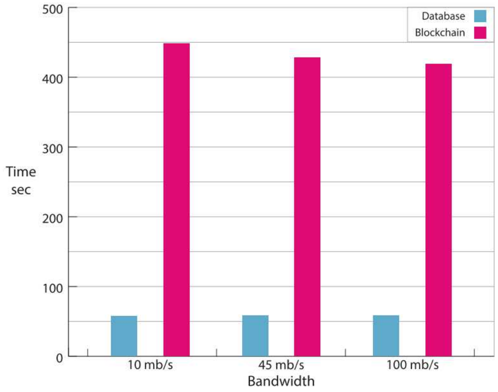


Fig. 5. Bandwidth Comparison test medians

or fault and it would not affect the system as a whole. These could all be tested using a physical test bed in future work.

There are many avenues to expand on this work, all of which would be valuable to proving that blockchain can be used as a data sharing platform between large scale entities. First, it would be useful to build a physical test bed to more accurately test the physical threats to the system such as tampering. This would help to show that the network works outside of the digital space and can truly meet the needs of a physical security system. Second, would be testing different types of private blockchain implementation such as private Ethereum[13], Hyperledger fabric [1], R3 Corda[3], Polkadot[14], etc. to see if these results hold true across multiple types of blockchain implementations or if there is added benefits from other implementations that have not been seen in MultiChain to this point. While we have shown in Section 3.1 that in a very small case that MultiChain appeared to be a more optimal framework than private Ethereum, there has been a vast expansion close to publication time of the variety as well as the quality of blockchain-based frameworks.

A final avenue to expand on is testing more resiliency metrics. While this work shows that it would be feasible to use blockchain as the communication network of the system, it is important to take a solid quantitative approach when making a new data sharing platform. The current metrics in preliminary testing are focused heavily on how a blockchain would operate in the architecture. The metrics that will be created should focus on creating a system with absorptive, adaptive, and recoverable capacity [4]. If these metrics can be met by the system, then the network architecture will prove suitable to be deployed as a communication network for a physical security system.

7 ACKNOWLEDGMENT

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

8 OUTLINE

- Introduction
  - This paper seeks to show that a blockchain based security system is a feasible alternative to the current communication architecture
  - Explain the difference between private/permissioned blockchains (Mutlichain)and public blockchains (Bitcoin)
  - Give outline of talking points ie applications of blockchain, architecture walk through, security metrics, results, ending with conclusion and future work
- Background
  - Past applications of blockchain
    - Do you need a blockchain?
    - Medrec: EMR on a blockchain
    - Electricity Market
    - Global Asset Management
    - Supply Chain
    - Privacy Management
  - Background of multichain
    - Similarities to Bitcoin
    - Differences from Bitcoin
    - Features that make it suitable for this application
  - Discussion of current security systems
    - Common practices in security systems
    - Problems that exist i.e. insider threats, centralized nodes, etc.
- Security Metrics
  - Admin Resiliency Test
    - Taking down each admin node systematically and determining if the blockchain loses functionality
  - Member Resiliency Test
    - Taking down each member node systematically and determining if the blockchain loses functionality
  - Tamper Detection Test
    - Send transactions with tamper value flagged and determine if the transaction was handled correctly by the rules set in place
  - Bandwidth Comparison Test
    - Send many transactions at once to both a blockchain and centralized database and measure the amount of time for each to process the data sent
- Results
  - Admin Resiliency Test
    - Blockchain network can handle 4 of the 5 admin nodes being taken offline before losing all functionality related to admin nodes
  - Member Resiliency Test
    - Blockchain network can handle 8 nodes being taken offline with no loss in functionality and another 3 can be taken offline before all functionality is lost
  - Tamper Detection Test
    - The network correctly handled all tamper flags that passed through the system
  - Bandwidth Comparison Test
    - The blockchain network was slower than a centralized database but still was able to store multiple transactions per second meaning that the blockchain network is not slow enough to have a noticeable impact on the human reading the data
- Conclusion and Discussion
  - Results show that a private/permissioned blockchain network has similar security features to public blockchains like bitcoin
  - Results show that a blockchain network like the toy model here is capable of processing large amounts of data like in a security system
  - Future work includes expanding the network, developing more metrics, and creating a physical test bed to test our results on a real system

A TRANSACTIONS

```
sendwithdata 1FdqTN7c8XXkSbh9s262kKyLRMNZHgz92svNYL
'{"tamperAsset":10,"detectAsset":10}'
'{"json":
   {
       "Microwave":"2.5"
   }
}'
```

Listing 3. Transaction to show microwave in alarm state

The transaction in listing 3 is a microwave transaction that shows a microwave in an alarm state. This transaction is valid because the tamper value in the system was set to 10. This means that it will be added to the Microwave Detect chain.

```
sendwithdata 1FdqTN7c8XXkSbh9s262kKyLRMNZHgz92svNYL
'{"tamperAsset":10,"detectAsset":10}'
'{"json":
    {
        "Microwave":"5"
    }
}'
```

Listing 4. Transaction to show microwave in secure state

The transaction in listing 4 is similar to the one above, the only difference is that this transaction has a microwave in a secure state so it will be added to the Microwave No Detect chain.

```
sendwithdata 1WfiQNZirYMwVqVgdKvaboupEpz3FWHzcAEzec
'{"tamperAsset":10, "classifyAsset":10}'
'{"json":
   {
    "filename":"/media/TA-V_FFMPEG_DATA/210/2019-05-08_15
         -00-59.png",
    "classify":"", "percent":0
   }
}'
```

Listing 5. Transaction for camera with no object detected

The transaction in listing 5 is a camera transaction that shows an image from a video feed which has not detected a target. This transaction is valid because the tamper value in the system was set to 10. This means it will be added to the Image No Class chain.

B SMART FILTERS

```
No Classify Filter function filtertransaction()
{
  var meta=getfiltertransaction();
  s=String(meta.vout[1].data[0].json.classify);
  if (s.valueOf() != " ".valueOf())
  {
    return "Classify found"
  }
}
```

This is a smart filter that checks if there is no object detected in an image. This filter is run on the Image No Class chain. If a transaction fails this check, the transaction will not be added to the Image No Class chain and will need to go to another chain to check if the transaction follows the rules set by that chain.

```
Tamper Filter function filtertransaction()
{
  var meta=getfiltertransaction();
  s=String(meta.vout[0].assets[0].qty);
  if(s.valueOf() != "10".valueOf())
  {
    return "Tamper Detected"
  }
}
```

This is a smart filter detects tampering in the system by checking the value of a tamper asset and comparing it against a predetermined value. This value can be changed periodically to make sure that an attacker does not have time to test the system enough to figure it out. This smart filter is placed on every chain to detect tampering on all sensors in the system.

```
MicrowaveDetect Filter function filtertransaction()
{
    varmeta=getfiltertransaction();
    s=String(meta.vout[1].data[0].json.Microwave);
    if (s.valueOf() !="2.5".valueOf())
    {
        return "No intrusion found"
    }
}
```

This smart filter checks to see if the data sent by the microwave indicates there was an intrusion. If a transaction fails this check it will not be added to the Microwave Detect chain and will need to be sent on another chain to check if the transaction follows the rules set by that chain.

```
Microwave Filter function filtertransaction()
{
    var meta=getfiltertransaction();
    s=String(meta.vout[1].data[0].json.Microwave);
    if (s.valueOf() !="5".valueOf())
    {
        return "Intrusion found"
    }
}
```

This code checks to see if the data sent by the microwave indicates that the microwave is in secure mode. If a transaction fails this check it will not be added to the Microwave No Detect chain and will need to be sent to another chain and checked by its filters.

REFERENCES

[1] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference* (2018), ACM, p. 30.

[2] Azaria, A., Ekblaw, A., Vieira, T., and Lippman, A. Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)* (2016), IEEE, pp. 25–30.

[3] Brown, R. G., Carlyle, J., Grigg, I., and Hearn, M. Corda: an introduction. *R3 CEV, August 1* (2016), 15.

[4] Francis, R., and Bekera, B. A metric and frameworks for resilience analysis of engineered and infrastructure systems. *Reliability Engineering & System Safety 121* (2014), 90–103.

[5] Greenspan. Creating and connecting to a blockchain.

[6] Greenspan, G. Multichain private blockchain—white paper. *URl: http://www. multichain. com/download/MultiChain-White-Paper. pdf* (2015).

[7] Karafiloski, E., and Mishev, A. Blockchain solutions for big data challenges: A literature review. In *IEEE EUROCON 2017-17th International Conference on Smart Technologies* (2017), IEEE, pp. 763–768.

[8] Makowski, P. Blockchain-security symbiosis. DEFCON, 2019.

[9] Nunes-Vaz, R., and Lord, S. Designing physical security for complex infrastructures. *International Journal of Critical Infrastructure Protection 7*, 3 (2014), 178–192.

[10] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 779–788.

[11] Saberi, S., Kouhizadeh, M., Sarkis, J., and Shen, L. Blockchain technology and its relationships to sustainable supply chain management. *International Journal of Production Research 57*, 7 (2019), 2117–2135.

[12] Sikorski, J. J., Haughton, J., and Kraft, M. Blockchain technology in the chemical industry: Machine-to-machine electricity market. *Applied Energy 195* (2017), 234–246.

[13] Wilcke, J., and Lange, F. Go-ethereum private network. https://github.com/ethereum/go-ethereum/wiki/Private-network, 2017.

[14] Wood, G. Polkadot: Vision for a heterogeneous multi-chain framework. *White Paper* (2016).

[15] Wüst, K., and Gervais, A. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)* (2018), IEEE, pp. 45–54.

[16] Wyss, G. D., Sholander, P. E., Darby, J. L., and Phelan, J. M. Identifying and defeating blended cyber-physical security threats. Tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2007.

[17] Zakhary, V., Amiri, M. J., Maiyya, S., Agrawal, D., and Abbadi, A. E. Towards global asset management in blockchain systems. *BCDL: First Workshop on Blockchain and Distributed Ledger* (2019).

[18] Zyskind, G., Nathan, O., et al. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops* (2015), IEEE, pp. 180–184.