Blockchain Security for IoT

Yousef Hashem
Department of Computer and Information Science
Linköping University
Linköping, Sweden
youha847@student.liu.se

Elmedin Zildzic
Department of Computer and Information Science
Linköping University
Linköping, Sweden
elmzi904@student.liu.se

Abstract—**Internet of Things (IoT) devices such as drones are becoming more prevalent, especially in fields such as medicine, law enforcement and military. The IoT networks still have blatant security issues, some of which could be solved using a blockchain. The decentralization, auditability and persistency of blockchain makes it a great candidate as a security mechanism for IoT networks. The aim of this paper is to explore the scalability and performance of Hyperledger Iroha as a security mechanism for IoT. A small-scale Iroha network is set up and security, performance and scalability are empirically evaluated. The test results indicate that blockchain technology shows promise as a security mechanism for IoT despite being very hardware reliant.**

I. Introduction

The prevalence of IoT (Internet of Things) devices is increasing rapidly in society and they are being used more than ever in both business and commercial fields. Despite this fact, IoT devices on public networks can still be discovered, and possibly even exploited, by malicious outside entities relatively easily. This increases the reluctance to use IoT devices in sensitive solutions, such as solutions in the medical field. As the amount of IoT devices in both business and commercial use keep increasing, so too does the need for a scalabe security framework that can secure communications for an IoT network.

One area of interest for the usage of IoT devices are drones. Drones today are being widely adopted by several government agencies to assist in their work. Drones can, for example, be used to transport medical supplies to a critical site. In this instance, security against external entities with malicious intent is of critical importance. Blockchain is a technology that could potentially address this issue of security, as transactions in blockchain have to be verified and can't be tampered with once they are verified. Blockchain does, however, have some weaknesses, such as slow consensus speed for block verification. Slow consensus speed could lead to very high latency, which would not be ideal for drones used for medical purposes. Nonetheless, they could be acceptable for other applications of drone usage, such as package transportation.

Blockchain is a promising concept that could be used to enforce better security in IoT devices by making it harder for outside entities to inject misinformation and manipulate or gather data from devices. In this study, we present how blockchain can be used in UAS networks to provide security using Hyperledger Iroha as our blockchain framework. More specifically, we present the idea of having nodes as separate entities from UAVs, where UAVs only send data and don't participate in the consensus operation due to the limited processing and storage capabilities of UAVs and IoT in general. We also discuss what blockchain vulnerabilities exist, and how they can affect UAVs connected to a UAS network that uses blockchain. Lastly, the performance, scalability and storage of blockchain when used in a UAS context is analyzed and discussed, i.e. from when a transaction is created to when it is stored as a block on the blockchain, using the European UAS Digital Remote ID Technical Standard[5] as basis for our evaluation.

Generally, there exist some studies on blockchain-integrated UAV networks. Xueping Liang et al. [11] presents their permission-less blockchain architecture *DroneChain* that includes some performance and overhead evaluations on their cloud server and controller setup. They use the drones as nodes in their network. Their results are presented in detail in section III.

A couple of studies specific to Hyperledger Iroha that utilize it for different purposes exist. One study presents a mobile application for secure user identities using blockchain technology [16]. The other study proposes a framework for blockchain usage in IoT using Hyperledger Iroha, but doesn't provide any concrete solutions [1] or performance evaluations. They do, however, conclude that it is feasible to use Hyperledger Iroha in IoT. Nevertheless, this study will explore and experiment with more defined theoretical use cases for blockchain in IoT, than previously performed.

The remainder of this paper presents some background and theory needed to understand the study, the methodology used to conduct the study, as well as the results and a discussion of the results. All of this is then concluded by the end of the paper.

II. Background

Will describe certain aspects and theories needed to get a better understanding of the subject in order for the reader to understand the study.

A. *Blockchain*

A blockchain consists of a sequence of blocks, which together make up a complete list of transaction records. It can be compared to a public ledger, where each transaction is digitally signed with a unique private key. These transactions can then be verified using a public key which matches the private key. Each block also contains a timestamp, usually as seconds in universal time since January 1, 1970. [17]

Blockchain has a few key characteristics that are very useful in enforcing security. These characteristics are the following: [17]

- *Anonymity* – Users can interact with the blockchain using a generated address that is different from the real identity of the user.
- *Auditability* – Any valid transaction needs to refer to previous unspent transactions. Once a new valid transaction is recorded into the blockchain, the unspent balance of those involved in the new transaction is modified. This makes transactions easier to track and easier to verify.
- *Persistency* – Due to the verification nature of blockchain, transactions can be validated rather quickly. Rolling back or deleting a transaction after it has been recorded is very difficult. Blocks that contain "fake" transactions are much easier to detect using this system.
- *Decentralization* – A blockchain verifies transactions made in a decentralized manner, where consensus algorithms are used to ensure data consistency. This eliminates central servers, which is a typical performance bottleneck in centralized transaction systems.

B. *Proof-of-Work (PoW)*

Proof-of-Work (PoW) is a consensus algorithm used by many blockchain cryptocurrencies, but most prominently by Bitcoin. PoW uses miners to validate new blocks before they are stored on the blockchain. They do this by calculating the nonce value of a block through many mathematically intense trial-and-errors. Only blocks with a valid nonce are added to the chain. Miners are rewarded with currency as incentives for good behavior, and compete with other miners to validate blocks. [15].

C. *Proof-of-Stake (PoS)*

Proof-of-Stake (PoS) is a consensus algorithm where validators are users that have enough stake in the blockchain, e.g. in Ethereum a user needs to stake 32 ETH to become a validator. Validators are chosen randomly to create new blocks, and are responsible for checking and validating (attesting) blocks whenever they are not creating blocks. They are rewarded with a larger stake as incentive for good behavior. In Ethereum, a validator can lose its stake for behaving maliciously. A big benefit of using PoS instead of PoW is that it requires very little in terms of computational power to create blocks [14].

D. *Smart contracts*

A smart contract is essentially a program that runs on a blockchain. They are often used to represent a contract or agreement between different entities. Similar to a program, a smart contract will contain a set of functions that executes some code. When a smart contract is deployed to the blockchain, users can interact with it by sending transactions that contain the function to be executed in the smart contract [8].

E. *Hyperledger Iroha*

Hyperledger Iroha [6] is a general purpose permissioned blockchain framework that focuses on reliability, performance and usability. It can be used to create and manage digital assets, identity, and serialized data. Iroha uses built-in commands that can be used instead of building smart contracts, which supports robustness and usability for sending transactions to the blockchain network. All nodes and accounts that are active in the blockchain need to be known by each node, i.e. the public key of each account and node is stored in all nodes of the network. This is because Iroha (and other blockchains) use asymmetric key cryptography to sign and verify transactions.

Iroha is permissioned in the sense that an Iroha blockchain network contains registered accounts with different permissions. These accounts can then interact with the network by sending transactions or queries that can include several commands. Upon each transaction or query, the receiving blockchain node will performs stateless validation to ensure the sender has used the correct cryptographic private key to sign the transaction or query, as well as ensure the sender has the required permissions. If the transaction passes stateless validation, it will be sent to the Ordering Service.

The Ordering Service is responsible for combining several transactions in the correct order into a proposal. The proposal is then sent to other peers where it is passed through stateful validation. Stateful validation entails validation against World State View; a snapshot of the current view of the system (e.g. how much bitcoin a user has). Note that queries only need stateless validation as they don't modify the state of the blockchain. Transactions that don't pass the stateful validation are dropped from the proposal, and what's left of the (now verified) proposal will be made into a block. At this point, the peer sends a vote containing the proposal hash (this should be the same across all peers), a block hash generated from the block (does not have to be the same for all peers), some metadata and the signature of the peer, and sends it to other peers. If a peer receives a super-majority (at least 2/3) of all votes, the block will be committed and a commit message will be broadcasted.

1) YAC Consensus algorithm: Yet Another Consensus (YAC) [12] is a Byzantine Fault Tolerance consensus algorithm that was made to be crash fault tolerant, performant and scalable. YAC guarantees safety and liveness as long as no more than n nodes are faulty out of at least $3n+1$ nodes. The creators of YAC wanted it to be asynchronous where peers did not have to be entirely reliant on other peers, and the performance scaling to be linear with respect to the number of nodes [13]. Another important aspect of YAC is that proposal size should not affect agreement time.

2) Permissions: The Iroha framework includes a lot of permissions that can be granted for different users. The permissions include getting or setting information on an account, creating and sending assets, querying the blockchain etc. Most permissions are very coarse-grained, i.e. you have permissions for querying or modifying your own account, and permissions that allow querying or modifying any account. Some permissions are grantable, which allows for a more fine-grained distribution of permissions. Accounts with grantable permissions can grant permissions to modify their own account to other accounts.

3) Genesis block: A genesis block is the first block in a blockchain, and contains a list of initial transactions. A genesis block is where one would want to create accounts with specific permissions, add peers, create a number of assets, set initial account details etc.

III. Related work

O. Almotery [2] has conducted a study in which blockchain is reviewed as a solutuion to drone cybersecurity. The reason behind this study was to determine whether blockchain can offer enhanced security for drones or not. A quantative strategy was applied and some of the prominent UAV security stakeholders in the industry were surveyed. Amongst the surveyed people, there were some Cybersecurity experts, some Drone security engineers, some CEOs (chief executive officers, and some CISOs (chief information security officers).

The survey included questions about blockchain adoption in the survee's organization, as well as questions about trust and security enhancement of blockchain. A majority of answers favored the adoption of blockchain as a solution for drone cybersecurity.

I. J. Jensen et al. [9] explores what security improvements blockchain could provide to a system. They reach the conclusion that implementing blockchain into a system would result in improvements in all CIA properties (confidentiality, integrity and availability).

Jensen et al. state that depending on the blockchain implementation, participation in the network can be vastly limited. Blockchain would also improve confidentiality through the usage of Public Key Infrastructures, since they utilize assymetric encryption. Encryption of each data block would maximize confidentiality since only those with the matching private key would be able to access the relevant information inside the blocks.

Furthermore, they reach the conclusion that a certain level of integrity is ensured because of the immutability of blockchain. Transactions on the ledger can be assumed to be valid after they have been added to the ledger. Smart contracts, which allow two parties to establish rules between one another, also increase the integrity.

As for availability, Jensen et al. reach the conclusion that it is increased due to the decentralization and peer-to-peer nature of blockchain. The network is able to continue operation as normal even if some of the nodes are under attack, since those nodes can be excluded. This makes it harder to DoS/DDoS a blockchain network, further increasing availability.

Xueping Liang et al. [11] presents their own architecture for a permission-less blockchain integrated UAS network called *DroneChain*. In their blockchain network, they use drones as nodes in the network. The drones utilize a duplex communication link with a controller where drones send data to the controller, and the controller can send commands to the drones. The controller forwards all data to the blockchain network, and all records are saved on the cloud. The authors also did performance and overhead measurements on their cloud service with various number of drones and data size. Results include:

- Average response time for data size of 64 bytes shows linear growth up to a 1000 nodes, with a max of around 550-600 ms.
- Average response time for data sizes of 1-1024 KBs using 100 drones, with response times ranging from 2000 to 8000 ms.
- Average response latency is generally stable under 1000 ms for 100 drones.

The authors of the study also did a security analysis of their blockchain network. Through their architecture they are able to achieve secure communications using an intermediary controller between drones and the blockchain network as well as their cloud service. By using both a cloud service to store all records and the blockchain network itself, they are able to also offer data assurance, resilience and accountability. However, this is only assuming the data stored on the cloud is encrypted. Furthermore, they mention that the network is not suitable for drones that require different permissions, and that such features require a private blockchain network.

IV. Methodology

The overall setup for the performance, scalability, and storage testing is presented in Fig. 1. The load test provided in the Iroha git repository was modified and used to spawn multiple workers that would fire one request per second each to the blockchain network residing on the cloud. The load test was launched on two separate computers with multiple workers active in order to simulate a UAS network. In order to take measurements, a separate load test with only one worker active was launched on a Raspberry Pi device. The Pi would act as a UAV and send one request per second for a total of 100 requests. The response times and the block sizes would be saved as files, which would later be processed to produce readable data. Response time is the amount of time it takes to commit a block (the block has reached consensus and is being saved to the blockchain) for any given request.

A. Requirements per standard

As per the standard [5], the following variables were mandatory to store for a UAV:

1) Basic ID Message
2) Location Message
3) System Message

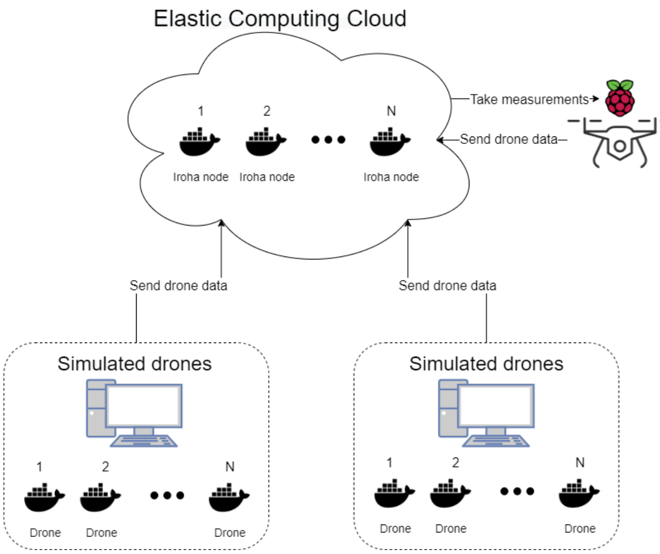

Fig. 1. Testing infrastructure

4) Operator ID
5) Message Pack

Out of these, *Location message* is the only data that has to be updated frequently. Other data can also be stored on the blockchain without difficulties, and drones can be registered as accounts with their ID as the account name. But, as this data did not have to be updated as frequently as *Location message*, it was not included in the performance evaluation.

1) Location Message: Each request was made to contain data that is mandatory for a UAV to send. These included:

- Latitude
- Longitude
- Altitude
- Direction
- Speed
- Timestamp

The data was sent each second as per the standard requirement [5], which states at least one update per second. Each request was sent as a transaction to store the data on the blockchain, where it could then be retrieved by querying the blockchain.

B. Hardware

The hardware setup used for this project was very basic. It consisted of two computers, one Raspberry PI device and an instance hosted on Amazon's Elastic Computing Cloud (EC2) service.

C. Elastic Computing Cloud

Amazon's EC2 service provides a scalable computing capacity, and eliminated the need for us to invest in hardware ourselves. An instance was set up and hosted on Amazon's EC2 service, using the AWS Free Tier. The AWS Free Tier only offered the instance type *t3.micro*, which according to Amazon [3] is a burstable general-purpose instance type, meaning the CPU has a consistent baseline performance that can also burst if enough CPU credits are present. Amazon does not have any concrete specification regarding any of the hardware, but states that (for t3) each vCPU is a thread of either an Intel Xeon core or an AMD EPYC core. The complete specification:

- 2 vCPU Intel Scalable Processor @2.5 GHz with 6 CPU credits/hour
- 1 GB RAM
- 30 GB EBS Storage
- Up to 5 Gbps network speed

```
config_docker = {
  "block_store_path" : "/tmp/block_store/",
  "torii_port" : 50051,
  "internal_port" : 10001,
  "max_proposal_size" : 10,
  "proposal_delay" : 10,
  "vote_delay" : 10,
  "mst_enable" : False,
  "mst_expiration_time" : 1440,
  "max_rounds_delay": 50,
  "stale_stream_max_rounds": 2
}
```

Listing 1. The configuration for each node

The RAM was the biggest bottleneck during the study, which severely limited the amount of nodes that could be launched simultaneously and the amount of connections that could be made to the server. As such, only four Iroha nodes were created and tested, with up to 32 drones sending requests.

D. Docker

Docker containers were used to create a blockchain network and a UAS network. One great benefit of using Docker is that multiple containers can be created on the same machine.

E. Iroha configuration

Iroha allows the configuration of each node through various parameters [7]. Of use for this study were the various *delay* parameters, which can be tweaked to gain better performance. The same configuration was used for all nodes, the details of it can be viewed in listing 1. The values for different parameters were derived from testing. Low delays can often increase CPU utilization, but the reasoning behind using such values is that nodes are going to receive a lot of transactions in under one second. Note that the values are set in milliseconds.

F. Use case experiments

The Iroha permissions served as a basis to determine use cases. There is some flexibility involved with how a blockchain network can be structured. Iroha has some commands that allows clients to add additional peers and nodes, and these commands in turn require the correct permissions as well. Such permissions need to be included in the genesis block of the blockchain if such flexibility is needed. Some additional things that were looked at were query responses and how they support confidentiality and non-repudiation, how the

```
[Account]:
-Account Id:- drone1@coniks
-Domain- coniks
-Roles-:
user
-Data-: {
    "admin@coniks": {"status": "grounded"},
    "drone1@coniks": {"status": "airborne"}
}
```

Listing 2. Account details of drone1

blockchain itself is stored, and how transactions and queries are sent to and from nodes.

V. Results

The results of the performance, scalability and storage tests, as well as the use case experiments, are presented in this section.

A. Drones in Iroha

Iroha naturally supports the use of drone ID's through the use of accounts, where each account equates to a drone ID. Each account would then have specific drone details attached to them, either through a *set my account detail* command or in the *genesis block*.

The details are set with a key/value pair, similar to a *dictionary* object in Python. Key/value pairs can not be removed once they are set, values can only be modified through the corresponding key. Furthermore, the account that sets a key/value pair is included in the account details as a key/value pair itself, meaning one account can not modify details of another account that are set by that other account. See listing 2 for clarification. Notice that both accounts have set the same key but contain different values. This sort of implementation supports both non-repudiation, but also trust in the sense that a drone (assuming the drone is not compromised) can always trust its own data, and does not have to interpret data coming from other accounts. Similarly, other accounts accessing another account's data can trust that it is set by the corresponding account (e.g. fetching gps data).

If a key/value pair that should not be modified has been modified, the previous values are still stored as blocks in the blockchain, which supports traceability. Unfortunately, Iroha does not have any finer-grained permissions that would allow an account to only change one specific key/value pair and not all of them, or allow other accounts to access only specific information. A workaround would be to have multiple accounts attached to a drone, with different permissions and with different data stored in each. For example, one account can have *set account detail* permissions with gps data, and another account with only *get* permissions with drone data that should never be changed, or only changed by an admin.

B. Use cases

The most relevant permissions for a UAS network are the following:

- Can create account
- Can add/remove peer
- Can get/set my account detail
- Can get my account

Through the use of the permissions listed, there are two main scenarios that can be applied to a UAS network. One is where network utilizes admin accounts to dynamically change the network, and another network where everything is initialized in the genesis block with no admin accounts. Drones would only need the *can get/set my account detail* and *can get my account* permissions in order to update its location. Note that there are many other permissions that can still be useful in a UAS network, but will not be covered here as the use of them are dependent on the needs of the entity setting up a blockchain network.

1) Dynamic network: There are a number of advantages to using a dynamic network with admin accounts:

- New nodes can be added to the network.
- New drones can be added to the network.
- Malicious peers can be removed.

The main drawback to having admin accounts in the network is that, once an admin account is compromised, then the whole blockchain network is compromised.

2) Static network: The main advantage to using a static network is that no admin accounts need to be a part of it. A compromised node or account would not be able modify any other account or the blockchain itself (with the correct permissions set). The drawbacks are that new drones or peers can not be added to the network, and that dishonest peers can not be removed.

C. Performance and scalability

The performance and scalability of a blockchain application in IoT networks could not be thoroughly tested due to hardware limitations. Only 4 nodes were successfully hosted on the EC2 instance before the 1GB RAM on the instance ran out, which made it rather difficult to test the performance and scalability of the blockchain application. Nonetheless, an attempt to bypass this limitation by hosting the nodes on a personal computer was made. This attempt showed promise, however, the computer lacked the necessary processing power, and the attempt was promptly abandoned. Thus, it can be stated that the scalability and performance of a blockchain application in IoT networks is heavily dependent on the hardware.

The test results can be viewed in Fig. 2 and 3. Fig. 2 shows the response times using 4 Iroha nodes, with 4, 8, 16 and 32 simulated drones. Fig. 3 shows the same response time but with box plots. The full line inside a box represents the median value, while the dashed line represents the mean value. Most response times are within 150-250 ms, with some outliers ranging from 300 ms to more than 700 ms. The most extreme outliers lie with the 32 drones test, which is to be expected when load on the server increases.

D. Storage

The average block size for each of the test configurations used can be seen in Fig. 4. The average block size increases with the amount of simulated drones that are used. This happens most likely due to the fact that, as more transactions are sent to Iroha Ordering Service on the server, more of them will fit into a single proposal before a proposal is passed on to the stateful validation process. The Ordering Service of Iroha waits a set proposal delay for additional transactions to be put into a proposal, see listing 1. The more transactions that are stored in a single proposal, the bigger the block size.

Using the most likely scenario for a UAS network from the test, i.e. 32 drones, a prediction can be made on how big the blockchain can become after 24 hours of constant location updates. For a block size of about 3000 bytes, the blockchain would require approximately 7.72 GB of storage for just location updates (i.e. not counting other drone data) after 24 hours uptime.

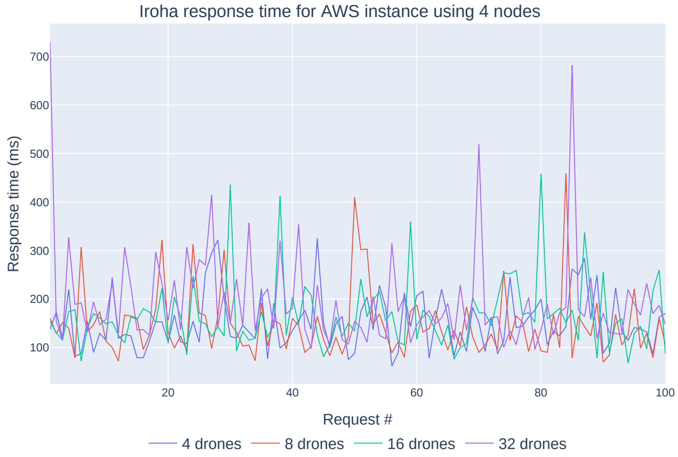

Fig. 2. Response times recorded on AWS EC2 instance

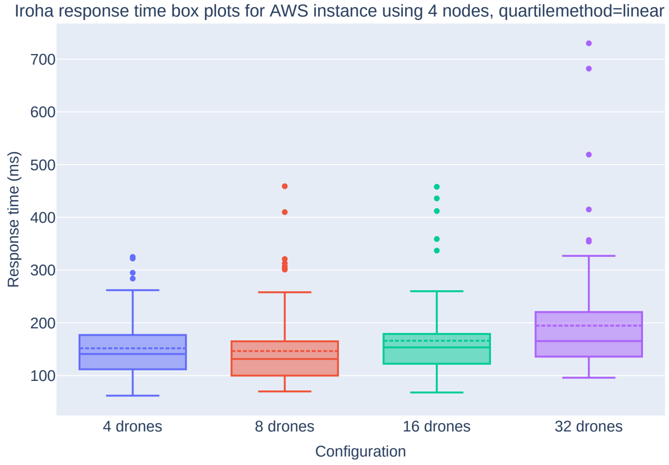

Fig. 3. Box plot response times recorded on AWS EC2 instance

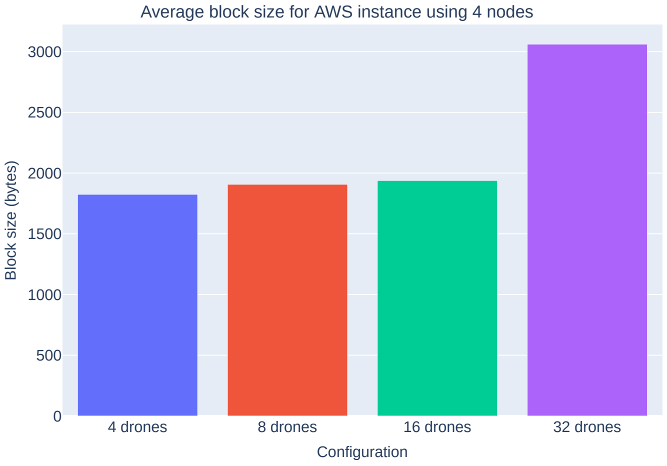

Fig. 4. Average block sizes on AWS EC2 instance

VI. Discussion

A discussion of the results of the study is presented in this section. The discussion includes own thoughts as well as possible ways to bypass certain blockages or problems.

A. Blockchain usability

Blockchain seems to be promising in a context similar to this one. It increases the integrity of the UAS, while still keeping communications rather simple. The performance and scalability could, however, not be properly tested, and a statement regarding whether blockchain is actually usable in such a context cannot be made. If the response time is less than a second for a blockchain network consisting of 64 nodes and 256 active users, then it is very plausible that the blockchain network is usable in practice as well.

1) Possible issues with Blockchain in IoT networks: Depending on whether the blockchain network is dynamic or static, different problems may come to light. The main issue in a dynamic blockchain network is the increased risk of the system being compromised. If one of the admin accounts in a dynamic network it is equivalent to the whole system being compromised. Having admin accounts in a network could therefore be deemed a security risk.

Setting up a static blockchain network without admin accounts is not problem-free either. Not having admin accounts means the blockchain network cannot be expanded with new accounts, which in this case represent drones.

A solution to this dilemma could be to set up one unique admin account for each action requiring elevated permissions. The problem with this solution is that it would require one account which has the ability to elevate permissions of other accounts. Compromising this account would mean compromising the blockchain network. Another solution could be to set up a semi-dynamic blockchain network, where only one account with elevated permissions exist. This account would require extra security measures to be in place in order to make sure that the account is never compromised, since compromising this account means compromising the whole blockchain network.

B. Iroha Security Concerns

Xiaoqi Li et al. [10] have performed a systematic examination of blockchain risks that exist today and their respective solutions. Evangelos Deirmentzoglou et al. [4] have also performed a survey on existing blockchain attacks and solutions, but provide a more comprehensive list and a unique long-range blockchain attack.

Many of the existing blockchain risks are present in blockchains that use Proof-of-Work (PoW) and Proof-of-Stake (PoS) as consensus algorithms, as well as blockchains that use smart contracts. Due to the fact of how Iroha is constructed and operates, it is able to mitigate a lot of the vulnerabilities and attacks common in public blockchain networks such as Bitcoin or Ethereum. Some key components in Iroha that aid its robustness and security are:

1) It does not use PoW or PoS consensus algorithms, instead it uses YAC [12], which is a Byzantine Fault Tolerant based consensus algorithm.
2) It does not use miners to validate blocks; a super-majority of nodes need to collaborate on the validation.
3) It uses built-in commands to perform transactions. Smart contracts; subject to traditional coding error and mistakes, and targeted vulnerabilities presented by Li et al., are not needed in Iroha. This is not to say that the built-in commands do not have potential flaws that are yet to be discovered.
4) Nodes need to have been added either in the genesis block, or by an account with the correct permission, in order to be part of the blockchain network.

In any case, there are other vulnerabilities that could potentially affect an Iroha blockchain network as presented by Li et al. and Deirmentzoglou et al. Some of them are:

- Private Key Security
- 51% attacks
- Sybil attacks
- Eclipse attacks

1) Private Key Security: Iroha uses an asymmetric key encryption scheme to sign and verify transactions. Anyone that has access to the private key of any entity that is part of the blockchain network can sign and send transactions. It is therefore of vital importance that private keys are well protected, especially in drones that operate in public spaces where anyone can potentially get a hold of one to access its hardware.

If a drone is compromised, the attacker can change any data that belongs to it on the blockchain. Getting hold of an admin account private key would be catastrophic for the entire network (see section VI-A1). If an Iroha node is compromised, it would be able to disrupt communication between all entities in the blockchain.

2) 51% attacks: 51% attacks is when an adversary takes control of the majority of the nodes or miners in a blockchain network. Doing so would give the adversary full control of it. This is most common with blockchains that use PoW or PoS. Iroha, however, uses a Byzantine Fault Tolerance based consensus algorithm. A 51% attack (or more correctly a 34% attack) on Iroha would only require control of at least 1/3 of all nodes in order to be able to reject any new transactions that are sent to any node. This is because this sort of consensus algorithm requires a super-majority (at least 2/3) of all the votes in order for a block to be committed and stored on the blockchain (see section II-E1 for details). Naturally, following that fact, an adversary gains complete control of the network if they can gain control of the super-majority of all nodes (could be considered a 67% attack).

Because Iroha is permission-based, gaining control of 1/3 of all nodes is still very difficult considering that new nodes can not be inserted into the network without permission. An attacker would have to gain access to a node that is already part of the network, as well as its private key. It is therefore of high importance that both of these are well protected, and preferably stored in separate hardware without remote access to the key.

3) Sybil attacks: Sybil attacks entail inserting fake identities (e.g. drones or nodes) into a blockchain network in order to disrupt or misguide the conensus of new blocks.

A sybil attack is only made possible if an adversary gets hold off an account with *Add a peer* or *Add an account* privileges in a dynamic blockchain network setting (see section V-B). With such privileges, an adversary can insert as many peers or create as many accounts as they want. Such privileged accounts need to be well protected. If this is not possible, a static network should be used instead which would prevent this attack entirely.

4) Eclipse attacks: Eclipse attacks involve an adversary that takes control over the inbound and outbound traffic of a node. This is done to effectively block off communication from honest peers, and instead force the node to communicate directly with dishonest ones. This is often done to stage other attacks, such as a 51% attack.

Staging a 51% attack by using eclipse could potentially be a much easier way of asserting some control over the blockchain. Now, the attacker only has to disrupt the communication of 1/3 of all nodes, instead of taking over each one. At the very least, an eclipse attack could be effective at lowering the throughput of transactions.

Effective ways of mitigating this sort of attack would be to ensure that adversaries can not get information about the structure of the network, i.e. they can not get hold of IP addresses of the nodes. Good firewall rules and secure communications could be useful defensive measures in that regard. Otherwise, Iroha could already be considered robust against this type of attack in view of the fact that a node needs to communicate with $2/3 - 1$ nodes in order to get a proposal through. Finally, using a dynamic network discussed in section V-B with backup nodes could be a last resort technique, where blocked off nodes are cut off from the network and new ones are added.

5) Communication: The Iroha blockchain framework supports the use of TLS, which would encrypt and secure any communication taking place in the blockchain network.

C. Iroha privacy concerns

Thanks to the fact that Iroha is permission-based, confidential information can only be accessed with the correct permissions. In order to add an extra layer of protection and redundancy to the confidential information, such data can be encrypted before they are sent as transactions and stored on the blockchain. Encryption should be mandatory if there is a high risk of a node being compromised, as that could potentially give access to the entire blockchain storage where plain text could be read.

In the event that a node is compromised, the owner of the node should have taken other preventative measures beforehand to protect confidential information. Such measures include:

- Strong database password.
- Encrypted hard drive.
- Encryption key kept in a separate offline hard drive.

With the use of TLS communication and encryption of blocks, the amount of confidential data that can be leaked should be minimal outside of the node.

Extra steps might need to be implemented depending on the usage of the blockchain. However, for a private drone blockchain network, the stated privacy measures should go a long way in protecting stored private information.

D. Standard compliance

We believe our use-case scenarios would comply with the Direct Remote ID Standard [5] as all the needed information about a drone can be stored on the blockchain. The response times presented in section V-C shows that they are within one second, which fulfills the standard requirement of positional updates at least every second. If any of the drone information is requested with Direct Remote ID, queries can be sent from the drone to the blockchain network on behalf of the requester and then relayed back to the requester. Data stored locally, such as location, can be sent directly without querying the blockchain, which would help minimize load and congestion at blockchain nodes. All unique data that does not require regular updates, such as the registered operator ID, should be stored on the blockchain in order to prevent repudiation and illegal altering of data.

Network remote ID could also be supported with the use of blockchain, by providing accounts with special query permissions in order to query any drone on the network.

E. The reason why drones should not be nodes in Iroha

Aside from the obvious performance, energy and storage requirements that is needed to process several hundred drone location transactions, there is one other major reason why drones should not be taking part in the consensus that is specific to Iroha, and has to do with how a network is set up in it.

In Iroha, a node needs to know all other nodes in the network and their respective public key. This is because a super-majority of votes is needed to reach consensus on a block. If a node goes offline, the peer has to be removed from the network so that other nodes do not needlessly communicate with it and so that it is not counted towards a super-majority. If at least 1/3 of nodes go offline without being removed, no consensus can be made on any block. Since drones fly around in public space, they could easily be shot down (accidentally or maliciously), hijacked and stolen, or brought down by birds or mother nature. In the worst case, a stolen drone could become malicious through the efforts of the hijacker(s).

F. The performance and scalability evaluation

A blockchain hardly ever exists with only 4 nodes handling transactions at any point, as that would limit the security benefits of using a blockchain. Much of the security of blockchain comes from a multitude of nodes that store a copy of the blockchain and the current world state view each, which means it's very hard to alter the blockchain and perform illegal transactions unless the majority of nodes are compromised at the same time. As such, the performance evaluation of this study is not very applicable to the real world. However, permission-based blockchains has the added benefit of whitelisting the nodes that should be a part of the network, meaning such networks can potentially have a vastly lower amount of nodes compared to public networks, as it would be much harder to exploit if dishonest nodes can not be added to the blockchain network without permission. In any case, we believe that 4 nodes would comprise a too small of a blockchain network.

In order to properly test the performance and scalability of an Iroha node, each node needs to be set up on multiple servers or instances, preferably in separate locations too. This can be accomplished using a docker orchestration tool such as Kubernetes or Docker Swarm. Performance also needs to be evaluated both in the context of the entire network, and of the conensus algorithm itself, in order to identify one-way trip time from the sender to the receiver and see which of the two impacts the response time the most: the consensus algorithm, or the trip time. Scalability should also be tested in both dimensions; different amount of nodes with different amount of drones.

G. Limitations

Due to time constraints, limited knowledge of docker orchestration tools, and Amazon Free Tier, only a simple performance evaluation could be made where multiple Iroha docker instances were launched on the same cloud server instance. This evaluation only shows what the performance could look like for very small networks, i.e. 4 nodes on the blockchain network and up to 32 (simulated) drones.

One-way trip time of the requests was not measured. As such, no conclusion about the actual consensus speed of the blockchain can be made.

VII. Conclusion

IoT networks are quickly becoming more common, and devices like drones are being widely adopted by several government agencies and medical facilities to assist in their work. The IoT security however, is still lacking in some areas. In order to increase the security of IoT networks, blockchain can be used. A blockchain network with many nodes is much harder to compromise than a normal network, since a majority of the nodes have to be hijacked for the network to be compromised. This makes blockchain an excellent candidate when it comes to IoT security.

The aim of this paper was to explore Iroha, a distributed blockchain ledger framework, and its possible applications for an IoT network. Therefore, a small-scale blockchain network was set up using Iroha to see if it was a suitable security mechanism for an IoT network. Results showed that, when using 4 nodes, the response times in the Iroha network are manageable, and the block sizes reasonable. The tests also showed that a blockchain network relies heavily on the hosting hardware, where CPU and RAM are the greatest bottleneck factors. Even so, blockchain as a security mechanism for IoT networks shows great promise.

A possible problem of an Iroha network as a security mechanism was whether the network should be static or dynamic. A static network involves less security risks at the cost of essential functionality, and vice versa with a dynamic network. A possible solution for this problem would be to set up a semi-dynamic network, which reinforces the security flaws of having a dynamic network while retaining the essential functionality.

References

[1] Tanweer Alam. “IoT-Fog: A Communication Framework using Blockchain in the Internet of Things”. In: *International Journal of Recent Technology and Engineering (IJRTE)* 7.6 (July 2019). DOI: 10.5281/zenodo.3871166. URL: https://doi.org/10.5281/zenodo.3871166.

[2] Ossamah Almotery. “Blockchain as a solution to Drone Cybersecurity”. In: *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. 2020, pp. 1–9. DOI: 10.1109/WF-IoT48130.2020.9221466.

[3] *Amazon EC2 Instance Types*. Accessed: 2021-05-11. Amazon Web Services. URL: https://aws.amazon.com/ec2/instance-types/.

[4] Evangelos Deirmentzoglou, Georgios Papakyriakopoulos, and Constantinos Patsakis. “A Survey on Long-Range Attacks for Proof of Stake Protocols”. In: *IEEE Access* 7 (2019), pp. 28712–28725. DOI: 10.1109/ACCESS.2019.2901858.

[5] *Direct Remote ID – Introduction to the European UAS Digital Remote ID Technical Standard*. Standard. prEN 4709-002:2020. Aerospace and Defence Industries Association of Europe - Standardization (ASD-STAN), 2020.

[6] *Hyperledger Iroha*. Accessed: 2021-03-09. 2021. URL: https://readthedocs.org/projects/iroha/.

[7] *Hyperledger Iroha Configure*. Accessed: 2021-05-15. 2021. URL: https://iroha.readthedocs.io/en/main/configure/index.html.

[8] *Introduction to smart contracts*. Accessed: 2021-05-17. Ethereum. Mar. 2021. URL: https://ethereum.org/en/developers/docs/smart-contracts/.

[9] Isaac J. Jensen, Daisy Flora Selvaraj, and Prakash Ranganathan. “Blockchain Technology for Networked Swarms of Unmanned Aerial Vehicles (UAVs)”. In: *2019 IEEE 20th International Symposium on ”A World of Wireless, Mobile and Multimedia Networks” (WoWMoM)*. 2019, pp. 1–7. DOI: 10.1109/WoWMoM.2019.8793027.

[10] Xiaoqi Li et al. “A survey on the security of blockchain systems”. In: *Future Generation Computer Systems* 107 (2020), pp. 841–853. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2017.08.020. URL: https://www.sciencedirect.com/science/article/pii/S0167739X17318332.

[11] Xueping Liang et al. “Towards data assurance and resilience in IoT using blockchain”. In: *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*. 2017, pp. 261–266. DOI: 10.1109/MILCOM.2017.8170858.

[12] Fedor Muratov et al. *YAC: BFT Consensus Algorithm for Blockchain*. 2018. arXiv: 1809.00554 [cs.DC].

[13] Fyodor Muratov. *HL Iroha. Yet Another Consensus*. Accessed: 2021-05-06. Hyperledger Iroha. 2019. URL: https://www.youtube.com/watch?v=mzuAbalxOKo.

[14] *Proof-of-stake (PoS)*. Accessed: 2021-05-17. Ethereum. Apr. 2021. URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/.

[15] *Proof-of-work (PoW)*. Accessed: 2021-05-17. Ethereum. May 2021. URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/.

[16] M. Takemiya and B. Vanieiev. “Sora Identity: Secure, Digital Identity on the Blockchain”. In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 02. 2018, pp. 582–587. DOI: 10.1109/COMPSAC.2018.10299.

[17] Zibin Zheng et al. “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends”. In: *2017 IEEE International Congress on Big Data (BigData Congress)*. 2017, pp. 557–564. DOI: 10.1109/BigDataCongress.2017.85.