IMPROVING AVIATION DATA COMMUNICATION AND STORAGE SECURITY USING BLOCKCHAIN BASED APPROACH

By

Yusuf Zakir, MS

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

In Computer Information Systems

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

MAY, 2020

IMPROVING AVIATION DATA COMMUNICATION AND STORAGE SECURITY USING BLOCKCHAIN BASED APPROACH

By

Yusuf Zakir

APPROVED BY

_____

Khondker Shajadul Hasan, PhD, Chair

_____

Sadegh Davari, Ph.D., Committee Member

_____

Kwok-Bun Yue, Ph.D., Committee Member

_____

Amlan Chatterjee, Ph.D., Committee Member

RECEIVED/APPROVED BY THE COLLEGE OF SCIENCE AND ENGINEERING:

_____

David Garrison, Ph.D., Interim Associate Dean

_____

Miguel Gonzalez, Ph.D., Dean

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor, Dr. Khondker Shajadul Hasan, for the continuous support on my thesis and related research, for his patience, motivation, and immense knowledge. I had a good learning experience, and the credit goes to Dr. Hasan. His valuable guidance and support made all this possible.

I am very grateful to Dr. Kwok-Bun Yue, Dr. Sadegh Davari and Amlan Chatterjee for serving on my thesis committee and giving me valuable suggestions, their time and input. I would also want to extend my gratitude to the Office of Sponsored Programs and College of Science and Engineering at the University of Houston - Clear Lake for providing me with the graduate research assistantship opportunity for my master's education.

Last but not the least, my deepest gratitude goes to my family, friends, and well-wishers who always had my back, supported me and encouraged me to be the person I am today.

ABSTRACT

IMPROVING AVIATION DATA COMMUNICATION AND STORAGE SECURITY USING BLOCKCHAIN BASED APPROACH

Yusuf Zakir
University of Houston-Clear Lake, 2020

Thesis Chair: Khondker Shajadul Hasan, PhD

Data security and integrity remains a major challenge for the Aviation industry data storage. Many data breaches have happened in the past which has directly caused financial loss to airlines and indirect losses to their customers. Blockchain and its related approaches provide decentralized security and privacy, however yet they involve significant energy, delay, and computational overhead. Security and privacy remain a major challenge for data communication between an Air Controller Tower (ACT) and Flights mainly due to the immense scale of data transfer and distributed nature of this Internet of Things (IoT) network.

The thesis is divided into 2 parts. The first part is Aviation Data communication Security wherein the model consists of three main components: Customized Digital Certificate (CC), Validator (Smart Contract), and Storage (Blockchain Network). A set of new algorithms are proposed to outline various phases during communication and embedding message security. Each ACT and Flight need to be equipped with IoT devices

and Validator (SC) that can send and receive validated messages and store them in the Blockchain network at both ends. This paper explores the communication realm and outlines Cryptography and Blockchain-based approaches. Extensive empirical work has been done to demonstrate that the proposed models are viable concerning confidentiality, integrity, and availability. The second part is demonstrated in paper two, wherein the model consists of 4 components: channel CA and Identity Manager, Smart contract (Validator), Channels, and Distributed Ledger. Different methods are proposed to demonstrate that the proposed models are viable concerning confidentiality, Authentication, Authorization and data integrity.

TABLE OF CONTENTS

LIST OF FIGURES

1. SECTION I

1.1. INTRODUCTION

Data communication in the Aviation industry can also take place through IOT devices using IOT based algorithms for the distributed location of the flights [2]. IoT devices are used by the flights and ACT towers to perceive location. The IoT framework for commercial aviation consists of various sensing and networking devices on the aircraft that can communicate with each other and the ACT towers. Data can be exchanged between aircraft that are within the communication range of each other and the ACT towers [3]. The IoT devices used in aviation demand scalable and distributed security that can ensure data integrity while the message is in transit. Blockchain and smart contracts can provide immutability for data in storage, sender and origin integrity, confidentiality, and non-repudiation [10].

Cybersecurity is a major area of concern for IoT devices. Failure to uphold it can result in disastrous consequences. For example, the data communication between a flight and ACT could be intercepted and modified. This can result in the flight receiving an incorrect location message, and consequently, deviating from its planned course, flying into turbulence, or worse. Hence, the pilot of the aircraft and ACT tower must be certain that data is generated from authenticated sources and sender and origin integrity is maintained. If a pilot receives an illegitimate location message from a malicious individual, the example could certainly become a costly reality. Data confidentiality is also important for VIP flights as they are more frequently targeted by malevolent parties. Furthermore, in case of an accident, the integrity of the data stored on the Black box and at the ACT should be maintained. Non-repudiation should be considered because if the ACT is responsible for the incorrect information, then the guilty person should be identified.

In the Bitcoin Blockchain network, once a block is full, it is appended to the Blockchain by performing a mining process. The mining process requires a few particular nodes called miners to solve a resource-consuming cryptographic puzzle. The miner that solves the puzzle first gets to create a new block. This consensus mechanism is called Proof of Work (POW) [4]. We have eliminated the POW protocol from our blockchain application to reduce the amount of high computational resources required in the system as IoT devices have a limited capacity for resource-intensive tasks [5]. Our proposed framework instead relies on a hierarchical structure and distributed trust mechanism to maintain aviation security and privacy also catering to the specific requirements of IoT. Our design includes: A customized digital certificate is used by the Flight and ACT to authenticate and verify each other. Storage (blockchain Network) at the flight and ACT is where the data is stored after undergoing validation by the Validator. Validator (Smart Contract) is used to validate the data communication messages between the Flight and ACT.

This paper's primary contribution is to present a secure communication model and provide comprehensive empirical work illustrating the enhancement of security in Flight-ACT data communication using blockchain technology. The steps for proposing a secure communication model are as follows. First, a customized digital certificate is created by both the Flight and ACT to authenticate and verify each other. The customization of the digital certificate is based on the details of the Flight and ACT. Second, asymmetric cryptography and hashing are used to transfer the data messages. Third, a Validator (Smart Contract) is used at each end to validate the data communication messages, acting as a firewall. Finally, Storage (blockchain Network) is used to store messages, consequently providing immutability to the data. We provide qualitative arguments to demonstrate that

blockchain and smart contracts in aviation data communication achieve confidentiality, integrity, non-repudiation and sender-origin integrity.

Cathay Pacific lost nearly 9.4 million of its customer data. The airline claims that a major data breach caused loss of their customer information including passport numbers, identity card numbers, travel history, email address and expired credit card details. Air Canada noticed a major breach when an attacker tried to access its 1.7 million customer database. Though, the airlines locked its database still 20,000 records were lost in the data breach. Other information that was at risk was a name, email addresses, phone numbers, and identity information and credit card details [30].

A major data breached happened from August 21 to September 5, 2018, wherein 380,000 passengers of British Airlines' personal and financial details were stolen by hackers which contained names, credit card details and email addresses. Heathrow airport was fined with $120,000 for failing to secure the loss of data after one of its employees because of negligence lost the memory stick [30].

In recent years, we have seen an increase in Cybercriminals targeting the airline database because the airline industry is a rich and time-sensitive industry. Cybercriminals understand that the airline industry has rich and valuable customers who will have a valuable amount of cash in their credit cards or payment gateways. Also, airline companies have one important information that no other industry has in its database, and that is passport information. Passport information can aid fraudsters in phishing attacks and identity theft. The risks of security breaches intensify as several third-party vendors get involved in the process. Airline Companies work with many third parties such as credit card companies, banks, and other organizations. There can be chances of hackers infiltrating through third-party vendors. A strong access control mechanism should be in place to safeguard the data [31].

The bitcoin blockchain is a public blockchain network and during the creation of a new block, a computationally highly intensive task or mining is performed to form a new block. The mining process requires a few nodes called miners to solve a resource-consuming cryptographic puzzle. The miner that solves the puzzle first gets to create a new block. This consensus mechanism is called Proof of Work (POW) [4]. The POW consensus protocol has been eliminated from our model because it is computationally resource intensives. Since this is a permissioned blockchain network, a trusted, mutual consensus method is used to create a new block. Our design includes *Channel CA and Identities* for assigning identifiers and Digital certificates to the Actors for authorization and authentication. A *smart contract* (validator) validates and verifies the requests/queries from the employees and outside world. *Channel* is used for logical grouping of all organizations into the single subnet, it has one or multiple ledgers that can be accessed by actors based on the permissions assigned to them. A *Distributed Ledger* in our model acts like a Blockchain network database.

The research proposes the following methods using the above components. Please note that the order is not compulsory, any method can execute when the need arises. *The addition and deletion of Actors* demonstrate how identity manager and channel CA assign an identifier, smart contract, and Digital certificate to the actor for authorization and authentication. *Access control for actors using the smart contrac*t method demonstrates how the identities assigned in the Identifier activates corresponding functions and classes in the smart contract attached to the actor node. *Storing Data on the blockchain storage network* depicts a straightforward process of storing data on a blockchain storage network. Its data integrity aspects are explained in the blockchain model architecture. Public blockchain network uses POW, POS and various other computational resources intensive consensus protocols. Our design proposes the use of trusted consensus voting by

blockchain Committee which is explained in the *Creation of blockchain by consensus voting by Blockchain admins. Finally, the evaluation and analysis have been done on Authentication, authorization, confidentiality and data integrity.*

2. SECTION II:

2.1. PREVIOUS WORK / LITERATURE REVIEW

In paper [6], An IOT based model has been designed by the author for communication between flight to flight and between a flight to ACT tower. The paper claims that detecting flight location and data over land can be done by radar but detecting flight's path and location over water bodies is a major challenge. Recently, there have been disasters in commercial aviation, where aircraft have gone missing. The paper proposes techniques based on the Internet-of-things model for aircraft, where the aircraft can communicate with each other within a certain range and aircraft and ACT can communicate with each other within a certain range. The main issue in the paper is the data communication security between aircraft to ACT. Data communication security is a major issue in the world.

If an unauthorized or unknown entity hack's the communication channel between the ACT and aircraft and purposely gives misleading information about the aircraft future path, it can lead the aircraft into dangerous consequences. Our paper proposes the solution for protecting and maintaining data communication security using blockchain-based approaches.

In paper [7], titled Blockchain for large-scale Internet of Things Data Storage and Protection, a distributed data storage scheme via Blockchain is used. Certificate-less cryptography is used in preference to traditional cloud-based IoT structures that consume high computation power and storage. The centralized storage server model characteristically invites distrust and uncertainty. The scheme eliminates the need for traditional centralized servers by instead using "miner" blocks to perform transaction verification and records audit with certificate-less cryptography. The composition of a non-crypto currency transaction and how it is processed will be illustrated.

Paper [8] addressed IOT-based traceability and provenance systems for Agro-Food supply chains built on top of centralized infrastructures. Due to its nature, major concerns such as data integrity violations, tampering, and single points of failure persist. The paper uses a Blockchain-based traceability solution for Agro-Food supply chain management to address the previously listed issues.

In Paper [9], blockchain-based approaches are used to secure IOT communication. The model uses three tiers namely, cloud storage, overlay, and smart home. It also uses miners to validate internal and external messages to the home. Blockchain-based cloud storage is used to store the data messages and improve security.

In [32], the authors propose a new blockchain-based algorithmic approach to achieve secure communications between aircraft and ground stations (GST). The scheme relies on three algorithms, each one defining the interactions between aircraft and ground stations for a specific context. The first algorithm defines the registration procedure and the storage of registration details in the distributed ledger. The second describes the first authentication negotiation between both entities. And finally, the third one determines how entities communicate after this first exchange, i.e. once authenticated. While the solution ensures private communication for registered parties (if the registration details, including the public and private keys, are not stored in the ledger, there is no chance to encrypt the data transferred between parties), there are some limitations. The first one concerns the storing of the private keys inside the ground stations which could cause severe privacy leakages in case of compromising. Then, the ledger that records the identification data is indeed distributed (among the ground station nodes) but there is no mention to any consensus algorithm used to add the data to the chain, nor a choice of a technology (e.g. « bitcoin-like blockchain », « consortium blockchain », etc.).

In [33] the author stresses once again the security and privacy issues related to the adoption of ADS-B systems, and the defiance from the military aviation community. His paper is a contribution to the cryptographically « secure broadcast authorization » by presenting a novel blockchain-based PKI implementation. To do so, he presents an « Aviation Blockchain Infrastructure » that leverages the Hyper ledger Fabric (HLF) software, as it is argued to be more suitable to meet enterprise-like requirements. Indeed, the prototype described defines different types of « organizations » according to the nodes that compose the aeronautical network (e.g. military, corporate and civilian aircraft are simulated as well as airline companies and Air Traffic Management Services, ATMS). For each organization, a ledger is generated with associated access rights (e.g. military-type ledgers are only accessible by the aircraft at stake and ATMS; airline companies create one ledger for each of their aircraft in-flight). While this paper presents an exhaustive description of the roles and ledgers composing the prototype and proposes to use a well-known private blockchain like HLF, it lacks performance evaluations as well as security analysis.

3. SECTION III: AVIATION COMMUNICATION SECURITY

3.1. Core Components of the Proposed Model

3.1.1. Customized Digital Certificate (CC)

Customized certificates are used by both ACT and Flight to authenticate and verify each other before starting the data communication. The *Issuer* field in the certificate is the entity that generates the certificate while *to* field indicates to whom the certificate is sent [11].

$$\text{ACT}_{\text{certificate}} = (\text{ACT}_{\text{info}}, \text{ACT}_{\text{public key}}, \text{issuer} = \text{ACT}, \underline{\text{TO} = (\text{flight details (can vary from flight to flight))}} + (\text{Encryption}_{(\text{with ACT private Key})} \text{Hash}(\text{ACT [package]}) + \text{root CA [11]}$$

$$\text{Flight}_{\text{certificate}} = (\text{Flight}_{\text{info}}, \text{Flight}_{\text{public key}}, \text{issuer} = \text{AP}, \underline{\text{TO} = (\text{ACT details (can vary across towers))}} + (\text{Encryption}_{(\text{with AP private Key})} \text{Hash}(\text{AP [package]}) + \text{root CA [11]}$$

For instance, when the Houston ACT wants to start communication with United Airlines 53 Fight, it will send a customized digital certificate to the Flight recording "United Airlines 53" into the *TO* field. When a customized digital certificate is created for a particular flight, a corresponding block is created in the storage at the ACT tower. In the previously mentioned instance, a block will be created for United Airlines 53 in the ACT storage Blockchain network. Alternatively, when a flight wants to initiate data communication with an ACT, it will first authenticate itself to the ACT. The Flight will also send a customized certificate for that ACT wherein the *TO* field is modified. In this case, it will be changed to "Houston ACT" and subsequently, a corresponding new block

is created for that ACT in its storage. All the data messages between the Flight and ACT are stored in that block.

3.1.2. Validator (Smart Contract (SC))

The Validator validates all the messages per the security algorithms. Both the IoT devices on the Flight and ACT have Validators (Smart Contracts) that can send and receive messages [12]. When the Houston ACT sends a message to United Airlines 53 Flight (UA53), the Validator (SC) at UA53 proceeds to validate all the incoming messages as per the policies or algorithms. Once the message is validated, it is then stored in the Blockchain Storage.

3.1.3. Storage (Blockchain Network)

As per Figure 1, the Flight and ACT use Blockchain storage. The data message is stored in the Block after being validated [13]. In our blockchain network design, when an ACT and Flight initialize data communication, a new block is formed at both ends with the time and date. For example, after the Houston ACT and UA53 Flight start data communication and authenticate and verify each other, a new block is created at the Houston ACT for UA53. This block might have a format similar to Block (UA53 (11/27/18- 09:45:32)). The same process occurs at the other end and the resulting block in the Flight storage might be akin to Block (HOU-ACT (11/27/18-09:45:32)). Once the Blocks are created at each end, further data communication messages are stored within them in the aforementioned manner.

3.2. Phases of the Proposed Model

3.2.1. Phase1: Customized Certificate Creation by ACT and Flight

Before communication can be initiated, both the Flight and ACT must be authenticated to one another. Otherwise, the unfortunate scenarios such as those mentioned earlier regarding VIP Flights and hackers could take place. We use customized certificates to achieve authentication in our proposed model. As explained earlier, when a flight comes into the communication range of an ACT, that ACT sends a customized certificate (CC) for that particular flight by entering its details into the *TO* field of the certificate. The timestamp is also added to the certificate.

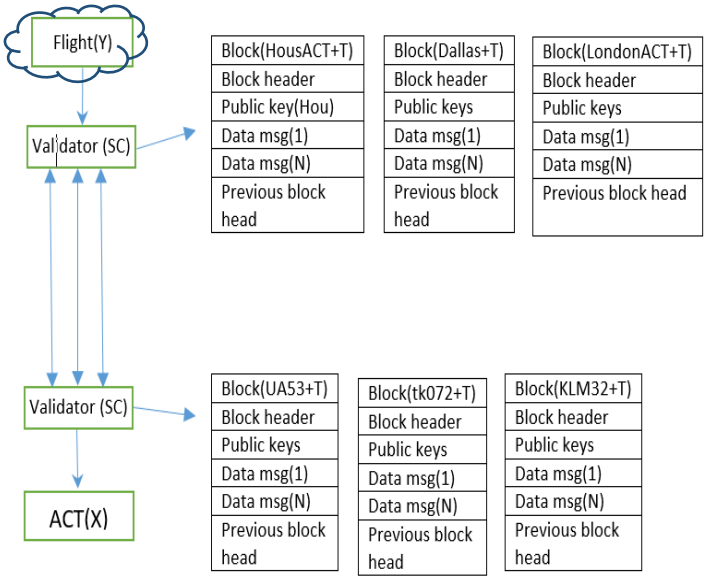

Figure 1. Overview of the Blockchain-based model using Smart contracts and Blockchain storage.

When the flight receives the CC, it attempts to validate the certificate by checking its timestamp, name, and other fields. After authenticating the ACT, the Flight retrieves ACT's public key from the CC. A new block is also generated for that instance of ACT-

Flight communication in storage. For example, after ACT(X) is authenticated by Flight(Y), Flight(Y) will create a block in its storage in the form of "ACT(X)-Timestamp". After creating the block in its storage, the ACT(X) public key is retrieved from the CC as shown in Fig. 1.

Now, if Flight(Y) wants to communicate with ACT(X), it will send a customized certificate (CC) by altering the *TO* field in the CC and adding a timestamp. When ACT(X) receives the package, it checks the timestamp, TO field, and others to authenticate Flight(Y). After this validation, ACT(X) creates a block for Flight(Y) in its storage blockchain network.

3.2.2. Phase 2: Initialize Authentication/Transfer of CC between ACT and Flight

Figure 2 illustrates how the customized certificate is transferred in addition to how the Flight and ACT authenticate each other and store each other's public keys in their respective storage (Blockchain network).

1) A customized certificate (CC) is created, Flight(Y) creates a CC for ACT(X), and ACT(X) creates a CC for Flight(Y).
2) The timestamp is added along with CC as shown in Fig. 2. The entire entity is called a Package (Pkg).
3) Pkg is sent to both parties. For instance, Flight(Y) sends the Pkg to ACT(Y) and ACT(X) sends Pkg to Flight(Y).
4) Pkg is received at each end. Now, Validator (SC) at each end validates the Pkg. Validator first checks the timestamp then, if the timestamp is valid, checks the CC.

5) Validator verifies the CC according to the predetermined algorithm. For Instance, if Flight(Y) sends the CC to ACT(X), the Validator (SC) at ACT(X) verifies the CC of Flight(Y).

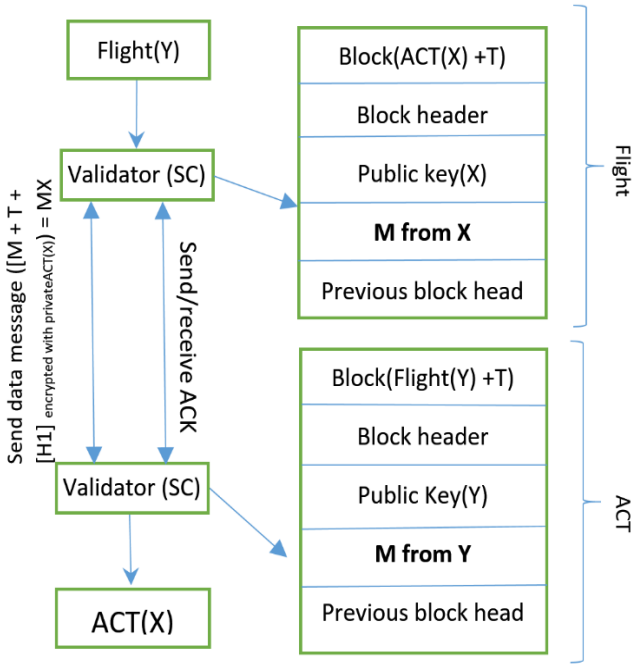

Figure 2. Transfer of messages for authentication between participating ends.

6) If the CC is valid, the public keys are retrieved at each end from CC and stored in the block as the first message.
7) If Validator (SC) at Flight(Y) successfully authenticates ACT(X), it will create a new block for ACT(X) like“BlockACT(X)-Timestamp” in its storage.
8) The same events take place if ACT(X) successfully validates Flight(Y) resulting in the creation and storage of new block “Block Flight(Y) +Timestamp” on its end.
9) Once the CC authentication is complete, the Flight and ACT send an ACK message to each other.

A Timestamp is added to the block name for logging purposes. It shows when the block was created, it's associated Flight/ACT and other information. After authenticating each other, Flight(Y) and ACT(X) have each other's public keys. They can now use them to encrypt the messages sent between them.

A set of new algorithms are proposed to model the smart contract validating customized certificates, validating messages, and termination of communication. The purpose of Algorithm 1 is to authenticate the CC at each end. After verification, it stores the public keys at their respective ends. This algorithm is implemented by both the Flight and ACT Validators (SC).

Algorithm 1: Smart Contract (SC) in Flight and ACT for validating Customized Certificate (CC).

Input: PKG: Customized Certificate (CC), Timestamp (T)

Get values for CC and T

Set the values of Timestamp Flag and PKG to False

For *values* in PKG

If (T→ PKG satisfies the threshold value) **then**

PKG is accepted for processing.

Send CC for verification.

Else If (T→PKG doesn't satisfy the threshold value)

Timestamp Flag value is set to FALSE.

Negative ACK is sent to the sender.

If (CC→ verified TRUE by the Validator (SC)) **then**

{

Retrieve public key from CC.

Else (CC→ verified false by the validator (SC)) **then**

Discard the CC.

Send Negative ACK to the sender.

Output: Stores Public key of the sender in the block created as the first Data message.

When the PKG = {CC + Timestamp (T)} is received, the algorithm will first check the Timestamp against the set threshold value. If the Timestamp satisfies the threshold value, the CC verification process continues. If the Timestamp does not satisfy the threshold value, the PKG is discarded and the Timestamp Flag value is set to FALSE. After this Timestamp verification process, the Customized Digitał Certificate (CC) is verified by the algorithm. If the CC is successfully verified, the public key is retrieved from the CC and used for further encryption and an ACK is sent to the sender. This is the process by which the Flight and ACT authenticate each other. It prevents malicious and fraudulent parties from communicating with the Flight or ACT. The Validator (SC) will always prevent this because the *TO* field in the CC will not match the sender.

3.2.3. Phase 3: Transfer of messages after Authentication

After *initialize authentication* phase, as shown in Fig. 3, data communication between Flight(Y) and ACT(X) starts. The communication involves the following steps:

1. If *Flight(Y)* wants to send M (location update message) to *ACT(X)*, it will first collect the information. In this case, the location of Flight(Y) is the M
2. *Flight(Y)* generates a Hash Value of the M, H_1.
3. *Flight(Y)* generates the Timestamp T
4. Flight(Y) combines aforementioned data into package X = [M + T + H_1]
5. *Flight(Y)* encrypts M (message) X using ACT(X)'s public key → [M + T + H_1] $_{\text{encryptwithact(X)}}$ = X
6. *Package X* is sent to ACT(X) and its Validator (SC) validates the incoming MX.
7. *Validator (SC)* decrypts the M X using *its private key*.

8. *Validator (SC)* checks the timestamp to prevent a replay attack. If the timestamp is proper, it proceeds forward.
9. *Validator (SC)* calculates a new Hash of M → H_2
10. The values of H_1 and H_2 are compared. If H_1 equals H_2, the integrity of the message has been upheld. If H_1 does not equal H_2, the M is discarded.
11. M is stored in block *Flight(Y)* in the Storage of *ACT(X)* as the message.
12. *ACT(X)* sends an ACK message back to *Flight(Y).*

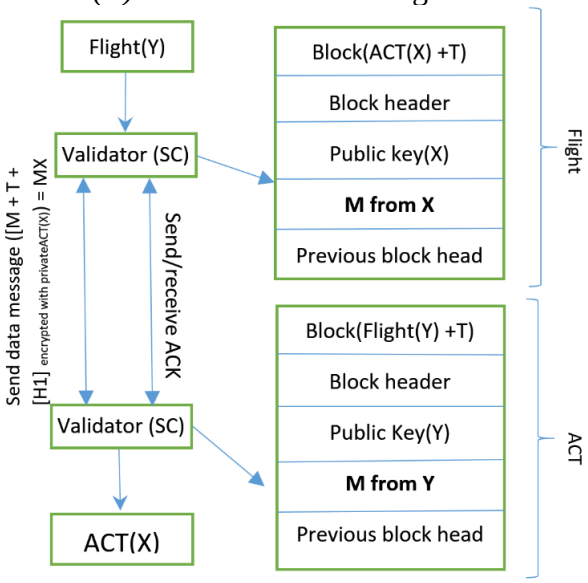

Figure 3. Data Transfer and storage of Data Msg in Blockchain storage.

The same process takes place when ACT(X) wants to send a message M to Flight(Y). After steps 1 through 10, Flight(Y) stores M in the Block ACT(X) in its storage (BC). After storing the M, Flight(Y) will send ACK back to ACT(X).

Algorithm 2: Smart contract at ACT/Flight for validating messages

Input: The Set X: (Data Message (M), Timestamp (T), Message Hash Value (M_H))

Get values for M, T, and M_H

Set the values of Timestamp Flag and X_{flag} to False

For values in X

Decrypt → ({[M+ T + H1] $_{\text{encrypt with public}}$} decrypt (private key))

If ({[MT+T+H1] $_{\text{encrypt with public}}$} decrypt (private key) = [MT+T+H1]**)**

[M+T+H1] is accepted for processing

Send the Timestamp (T) for verification

Else If (({{M+T+M_H} $_{\text{public key}}$} decrypt private key) not decrypted to [M+T+M_H]

The packet ({{M+T+M_H} $_{\text{encrypt with public key}}$} $_{\text{decrypt with private key)}}$ is discarded

If (T in [M+T+M_H]→SC threshold value) = =TRUE**) then**

[M+H1] is accepted for processing

Carry on checking the Message Hash Value

Else If (T in [M_T+T+M_H] →SC threshold value) = = FALSE)

X is discarded from the message queue.

Else If (New Hash (M) = M_H (M)) **then**

M is stored at the respective block.

Send Ack to the sender.

Else If (New Hash (M) ≠ M_H (M)) **then**

Set the value of the Y $_{\text{flag}}$ to False

Negative ACK is sent to the sender

Output: M is stored at the respective block, ACK is sent to the sender.

Algorithm 2 uses *Validator* (Smart Contracts) at both ends to authenticate the Data Communication Packet(X) (aka Data Message M). After M is verified, it is stored in the block. The inputs provided to the algorithm are a set of values X containing: Termination message (M_t), Timestamp (T), and MsgHashValue (M_H)). M is the original data communication message sent from either end. A timestamp will be added using a Date Time function for that packet. The Timestamp is added as a defense against replay and DOS attacks. MsgHashValue (M_H) is the hash value calculated by the sender and is

attached with the M + T. The Data Packet X= {M + T + M_H} is encrypted using the public key of the recipient → X = {MT + T + M_H} $_{\text{public receiver key}}$

On receiving X, the algorithm first uses the private key of the receiver to decrypt X. If Packet(X) cannot be decrypted with the receiver's private key, it is discarded. This protects the receiver from fraudulent messages M and ensures Origin and Sender Integrity. Thus, the receiver is protected from receiving an M from an unauthorized source. If Packet(X) is successfully decrypted, Timestamp (T) is verified against a threshold. If Timestamp (T) is invalid, a replay or DOS attack may be in progress and Packet(X) is discarded. If the Timestamp is successfully verified, a new hash value of the data message (M) is calculated. If the new hash value calculated is not equal to the hash value in the Packet(X), message M's integrity may have been compromised. This would be the case if an attacker did manage to modify message M. Thus, it is the block with the altered message whose data integrity has been violated. If the new Hash value is equivalent to the hash value in the packet, M is stored in the block. To conclude, the above algorithm authenticates and verifies the Data message packet (X).

3.2.4. Phase 4: Termination of Data Communication phase

When either the ACT or Flight wants to break the data communication channel, they send a *Termination Message* (MT). A timestamp is added to the MT and a Hash Value of MT is calculated resulting in H1. All three pieces of data are combined into Package(Y) and sent. At the destination end, the Validator (SC) checks the integrity of the MT using Algorithm 3, defined below. Once the MT is verified, both the Flight and ACT become aware that the data communication channel needs to be terminated. At this point, the private consensus protocol that was used to create public-private key pairs voids the current public-private key pair that has been used for communication between Flight(Y) and

ACT(X) making them now unusable. Consequently, after the Termination of Data Communication Phase, the Validators (SC) of Flight(Y) and ACT(X) will no longer accept any further packets.

Algorithm 3 is used by the Validator (Smart contracts) at both ends to authenticate the Termination Packet(Y) (aka Termination Message M_t). After M_t is verified, it is stored in the block. The inputs provided to the algorithm are a set of values Y containing: Termination Message (M_t), Timestamp

Algorithm 3: Termination of Data Communication

Input: The Set Y: (Termination Message (M_T), Timestamp (T), MsgHashValue (M_H)

Get values for M_T, T, and M_H

Set the values of Timestamp Flag and Y_{flag} to False

For values in Y

Decrypt → ({[MT+ T + H1] encrypt with public} decrypt (private key))

If ({[M_T+T+H_1] encrypt with public-key} decrypt (private key) = [MT+T+ H_1]**) then**

[M_T+T+H1] is accepted for processing

Send the Timestamp (T) for verification

Else If (({{M_T+T+M_H} public key} decrypt private key) Not decrypted to [MT+T+M_H])

The packet ({{M_T+T+M_H} encrypt with public key} decrypt With the private key,) is discarded

If (T in [M_T+T+M_H] →satisfy the SC threshold value) **then**

[MT+H1] is accepted for processing

Carry on to check the Message Hash Value

Else If (T in [M_T+T+M_H] →doesn't satisfy the value) **then**

Y is discarded from the message queue

If (New Hash (M_T) = M_H (M_T)) **then**

M_T is stored in the same block

The communication channel is terminated

Invalidate both Public and Private Keys

The Blockchain is saved & copied (headquarter storage)

Else If (New Hash (M_T) ≠ M_H (M_T)) **then**

Set the value of the Y_{flag} to False

Negative ACK is sent to the sender

Output: M_T is stored in the block, Communication channel terminated, both keys are invalidated, and the chain is copied to headquarter storage.

Value (T) and MsgHashValue (M_H). M_t is the original data communication message sent from either end. A timestamp will be added using a Date Time function for that packet. The Timestamp is again added as a defense against replay and DOS attacks. MsgHashValue (M_H) is the hash value calculated by the sender and is attached with the MT + T. The Data packet Y= {M_T + T + M_H} is encrypted using the public key of the receiver → Y = {M_T+ T + M_H}$_{\text{public receiver key.}}$

On receiving Y, the algorithm decrypts the X using its private key. If Packet(X) cannot be decrypted with its private key, it is discarded. This protects the receiver from receiving a false M_t and ensures Origin and Sender Integrity. For instance, if the private key of the receiver was unable to decrypt a Packet(Y), it can be concluded that Packet(Y) was not sent by sender A. Thus, the receiver is protected from receiving an M_t from an unauthorized source. If Packet(Y) is successfully decrypted, Timestamp (T) is verified against a threshold. If Timestamp (T) is invalid, a replay or DOS attack may be in progress,

thus Packet(Y) is discarded. If the Timestamp is successfully verified, a new hash value of the termination message (M_t) is calculated. If the new hash value calculated is not equal to the hash value in the Packet(Y), then the M_t integrity may have been compromised. Thus, it is the block with the altered message. The algorithm blocks altered message whose data integrity has been violated. If the new Hash value is equivalent to the hash value in the packet, $_{Mt}$ is stored in the block and the data communication channel is terminated, i.e., The Flight and ACT can no longer communicate with one another. The public keys stored in blocks at both ends get invalidated such that when the same sender and/or receiver try to use them to communicate, the algorithm does not accept the messages (M or M_t).

3.3. The Proposed Blockchain Architecture

The Blockchain Architecture encompasses a chain of blocks connected such that every block is connected to the previous block by the previous block's header. For instance, Block 5 is connected to Block 6 by Block 6's header and Block 4 is connected to Block 5 by Block 5's header. A Block structure is comprised of a Block Name, followed by a Block Header, Data Messages (N), and the previous Block's Header as shown in Fig. 4.

The block header is generated by calculating the Hash Value of all the fields in the Block including the Block Name, Data Messages, and previous Block Header. A Merkel Root Hash is used to calculate the Hash Value. When a new block is formed, it creates a hash value that is stored in the previous block. This is how a new block gets added to the Blockchain Network. The Hash Value ensures immutability for the Blockchain Network. If a hacker changes a Data Message in any block, the hash value of that block then changes, causing it not to match its old hash value stored in the previous block. As a result, a flag is generated, and that block will be frozen or deleted. However, Data will not be deleted because the Blockchain uses distributed ledger technology.

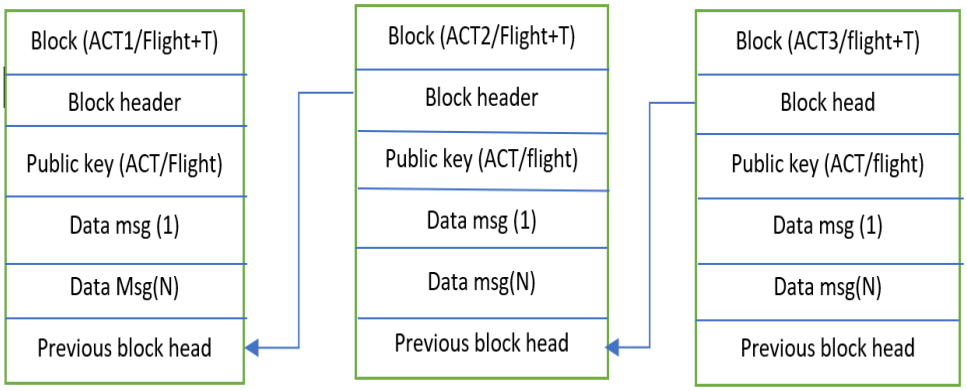

Figure 4. Proposed Blockchain data storage architecture.

Blockchain is an electronic cryptographic ledger that follows a decentralized network model—instead of storing all information in one database such as in conventional cloud-based applications, the information is distributed and synchronized across all nodes in the network. A consensus algorithm is deployed within the network to mitigate the issue of transaction duplication (or double-spending) by allowing nodes to verify true information. Once verified, information is then added to the hash value of a previous block, and the new sequence (i.e., previous hash + newly verified information) is hashed to form a new block using a cryptographic (i.e., one-way) hash function.

A cryptographic hash value is a string of non-readable letters and numbers of consistent length that represent information that was subjected to a hash algorithm. Each hash value is unique to the information from which it was derived. These characteristics, in addition to the network forcing continuous synchrony across all nodes, make blockchain immutable and tamper-resistant. Although cryptographic hashing is one-way, the decrypted information can be rehashed and compared with the stored hash value in the ledger. Furthermore, the network can persist amidst node failure. The threshold for the number of nonfunctional nodes before network failure is a function of the number of nodes connected to the network. The more the nodes in the network, the less likely it is to fail [4].

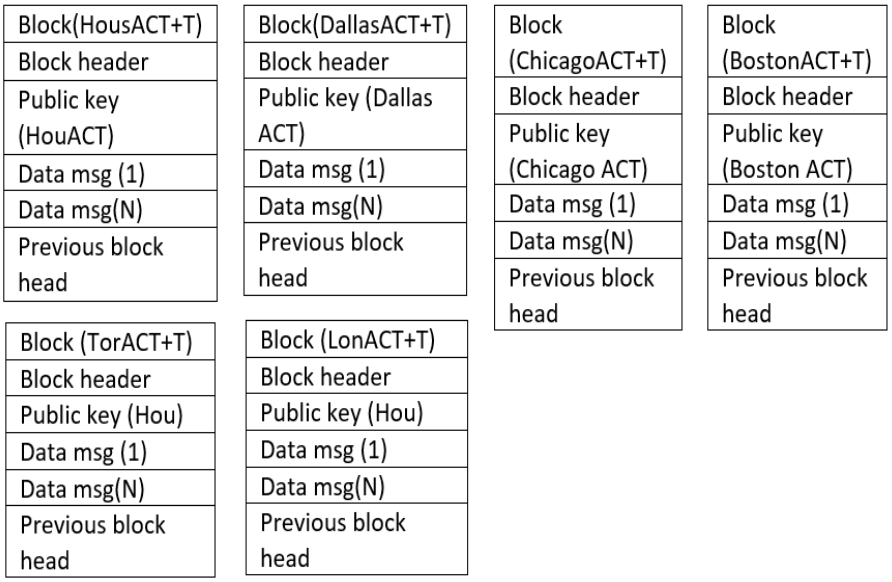

Figure 5. Proposed Blockchain Storage Network at ACT and Flight.

In the proposed model, when a Flight is airborne, it creates a block when it comes into the communication range of an ACT tower after the *Initialize authentication/transfer of CC between the ACT and flight phase.* For Instance, If United Airlines (UA53) has a flight route from Houston to London, it will first communicate with the Houston ACT and form a Block called "BlockHOU+TimeStamp". While flying to London, it will come within the communication range of several ACTs, such as the Dallas ACT, Chicago ACT, Boston ACT, Toronto ACT and finally the London ACT where it lands. Flight UA53 creates separate blocks for the respective ACTs along with the Timestamp for every ACT it comes into contact with as illustrated in Fig. 5.

When Flight UA53 lands at London airport, the entire blockchain network created as illustrated in the above Fig 5. Gets copied to the Aviation's (United Airlines) main headquarter storage. Again, when flight UA53, starts its journey from London to Houston, it creates a new chain of blocks of the respective ACTs with which it communicates during its route to Houston. After landing at the Houston airport, it transfers the network to the main Aviation's (united Airlines) storage. A common distributed ledger is created for

UA53’s flight route from Houston to London. After landing, the whole chain of blocks is copied to the airline's headquarters database. The Block structure is as follows: each block has a Block Name, followed by a Block Header, and Data messages. Next, there are the Data communication messages, the Merkel root hash of all the Data packets is computed at the end of the Block, then finally the previous block header is present. To generate a Public-Private key pair for asymmetric key cryptography, Elliptic Curve Cryptography (ECC) is used as opposed to RSA. ECC is stronger than RSA for a given number of bits [11]. The 256-bit ECC key pair is equal in strength to about 3072 bits of RSA key pair for hashing purposes. This paper utilizes SHA-256 to improve the security of the proposed model.

3.4. Evaluation and Analysis

Security analysis: our goal is to achieve the following security requirements for the data communication between ACT and aircraft, such as confidentiality, integrity, and availability. Confidentiality is defined as the authorization of a user to read/accept the data message. The aircraft and ACT are authorized to send and receive the data messages to each other. Integrity makes sure that the sent message received at the destination is not altered. Integrity is also divided into origin integrity and sender integrity. Origin integrity is defining the source that produced or constructed the data message. Sender integrity is defined as a sender who is authorized to send the data communication packet. Availability means that the authorized participants receive the data messages on time. To make data communication between aircraft and ACT confidential, private common distributed ledger technology, such as permissioned blockchain, will be used to store the data messages at both ends.

Unlike public blockchain wherein the transactions or records the transparent, in permissioned blockchain only the authorized participants can access the data messages. In our case only authorized participants such as aircraft and ACTs will be allowed to see/access each other's data communication messages. Transfer of messages is secured by customized certificates and generation of new public-private key pairs. Hence, only the receiver will have the current public key of the sender extracted from the customized certificate. The Customize certificate creation by ACT and flight phase and Initialize authentication/transfer of CC between ACT and flight authenticate each other and authorize each other to send and receive the data communication messages. Because of this, a malicious attacker will not be able to send or receive messages from the aircraft or the ACT.

The integrity of the data messages is assured by immutability factors provided by the blockchain network. Blocks in the chain are connected via the prior block hash. For instance, block 4 will have a hash value of block 5. A hash value of the block is generated from all the fields in that block and previous block hash. If a hacker tries to alter a message in any block, its hash value changes and is no longer equal to its old hash value in the previous block. Consequently, the network will detect the altered block and freeze it. The altered data message is unaffected because it is duplicated across all the blocks in the common distributed ledger. Hashing provides data integrity while the data packet is in transit from the sender to the receiver. The *Initialize authentication/transfer of CC between ACT and flight* phase provides origin and sender integrity.

Due to Customized Certificates (CC) only the aircraft and ACT have the one-time generated public-private key pairs of one another. If the receiver (aircraft/ACT) can decrypt the data packet using the private key, it is assured of sender and origin integrity. Blockchain storage networks also provide a provenance/non-repudiation factor. The blocks are created

with the respective (ACT/aircraft) name and timestamp (date, time). So, if the airline wants the data of a flight, for a particular day and time, they can create a search query in their main storage database where the data is stored. For instance, if United Airlines wants to investigate the crash of flight UA87, they can find out the data communication that had taken place because of the timestamp that is attached when blocks are created. The availability of the messages is provided by the validator (smart contracts) at each end. The validator will only allow authentic messages to pass. Following we will analyze the effectiveness of our solution in the prevention of security attacks, such as Distributed Denial of Service (DDOS) attacks and replay attacks.

Distributed denial of service (DDOS) is an attack that can be done by an attacker using infected devices by sending duplicate messages at a high rate of speed to the target [15]. In our solution, the validator (SC) at each end will not allow unauthorized data packets because the receiver will only have the authorized sender's public key extracted from CC. Therefore, the validator will not be able to decrypt packets from other sources. Even if the attacker captures an authorized packet(s) in transit and sends the same packet(s) as a DOS attack, the validator will discard duplicate authorized packets because it is using the timestamp to verify uniqueness.

4. SECTION IV: AVIATION DATA STORAGE SECURITY

4.1 Components

4.1.1 Channel CA and Identity Manager

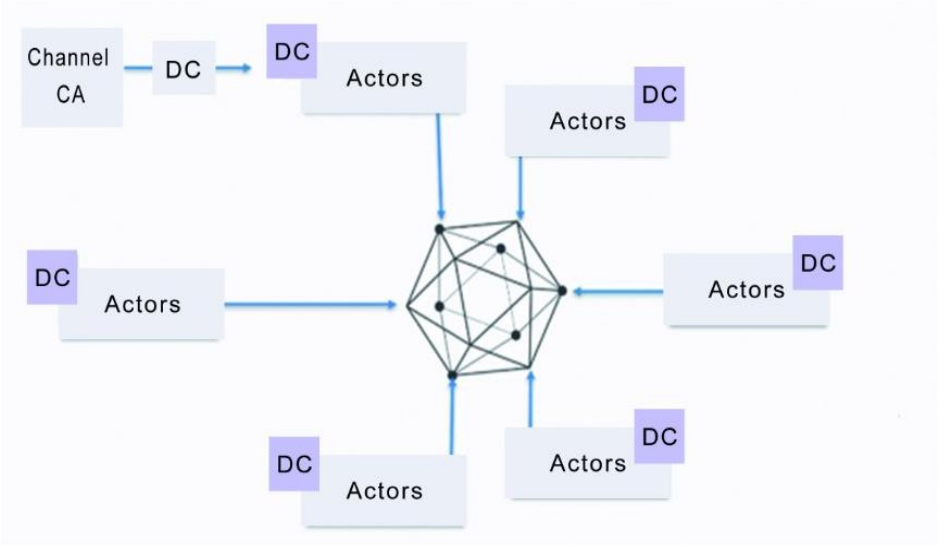

Figure 6. Assigning of Digital Certificate (DC) by channel CA to Actors as per their identifiers.

Every channel will have its Identity Manager and root/channel CA. The Identity Manager is created by the company or industry having the blockchain network. The main purpose of Identity Manager is to assign Identities to Actors in the Blockchain network as per their roles, responsibilities and permissions demanded. Next, Channel CA has to assign Digital certificates to the Actors. The digital certificates are based on the identities assigned to the actors. The combined Identities and Digital certificates work as an authorization protocol for the actors. The digital certificate given to an actor depends on the Identities assigned to the Actors. For Instance, if a new employee is added to the company and if

he/she is the part of the Blockchain network and given the position of System admins, then he is assigned an Identifier. Based on the role of that category, the actor may have some set of rights and unauthorized for other work. So, a digital certificate assigned to him will be based on that identifier. If he has been given rights to access the blockchain data in his identifier, then he can use his digital cert as an authorization to access the data but if he tries to access something that is denied in his identifier- then his digital cert will block him from doing that particular task.

4.1.2 Smart Contract (Validator)

Smart contracts in this model act as a validator. It is a set of Algorithms running against the transactions and data before being recorded or queried from the Blockchain database storage. Whenever data is requested to be stored in the blockchain storage, the requested transactions/ data passes against the validator. If the validator validates the data, then it is stored in the Blockchain network. The aviation data is huge and its streaming in and out at a huge scale, hence, the validator will validate the data at a constant rate.

A validator can also be set for giving access and querying the Distributed ledger in the channel. The Actors in the channels are assigned Identities and Digital certificate/Signature by the channel CA and Identity Manager based on their roles, responsibilities, and permissions. Now, whenever an actor needs to access or query the Blockchain storage, their request will pass through the validator, the validator will check the digital certificate/signature. If the cert has the permissions for that actor, then the validator will allow the actor's request to query the blockchain storage. If the cert has denial for that actor's permission, then the validator will deny access to the blockchain storage.

4.1.3. Channel

A Channel is a private subnet of the Communication between one or many network members, to conduct private, confidential transactions. A channel is defined by members of the organizations that are added in the channel, actors, shared ledger, and validator. Each request/transaction on the channel that is coming in or going out is validated on the channel. The channel has a group of organizations. For Instance, in our model, there is an airline known as Bogo Airline. It has 3 branches, we can call these 3 branches in Blockchain world as 3 organizations, such as HousOrg, DallasOrg, and ChicagoOrg. These 3 organizations will form a channel. Each organization has members/Actors, a smart contract (validator) for that channel. The smart contract is customized and made for that particular channel and all the algorithms are fed into it. The copy of the smart contract (validator) is distributed across all the members in the channel, attached to the distributed ledger.

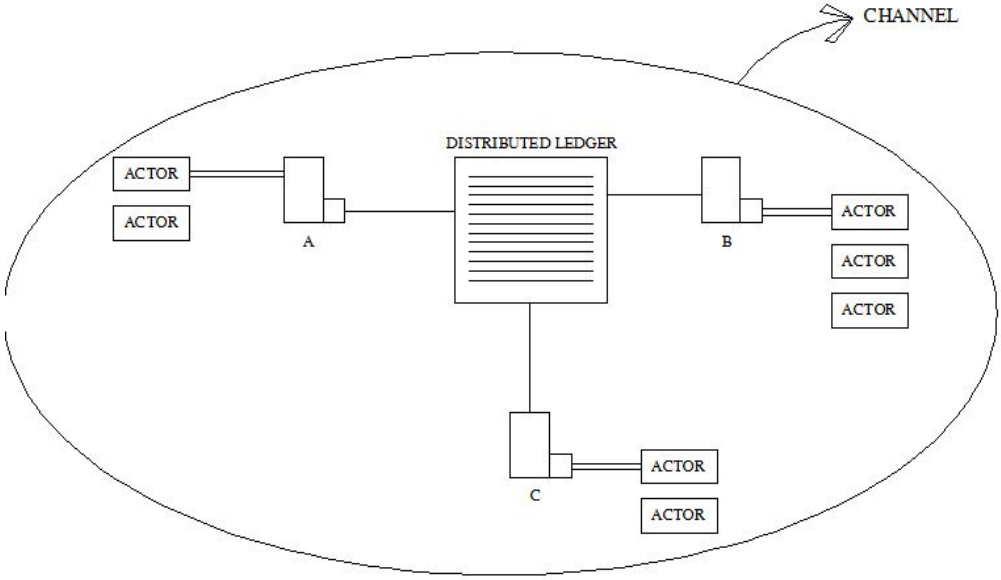

Figure 7. The architecture of Airline Blockchain model

Theoretically, many Blockchain system implementations grant members the power to conduct private and confidential transactions while coexisting with restricted members on the same blockchain network. Controlling members define one or more channels to

isolate peers into subnets and create private ledgers. Each channel's ledger is only accessible to its member's nodes. The channel's organizations (entities) must approve each member's membership to the channel. Client requests are routed to a specified channel to run a smart contract that is deployed on that channel. The results are endorsed and verified, and then updated in that channel's ledger.

4.1.4. Distributed Ledger:

A channel has a distributed ledger, which is stored in each block of the blockchain. If the blockchain network has 50 nodes/blocks, the DLT is replicated in all the blocks. In our model, the DLT has 2 types of database states. First, is the transaction database wherein all the data is stored. For Instance, Customer ID, Name, frequent flyer type, credit card details, etc. Thousands of customer's details can be stored as rows in the database.

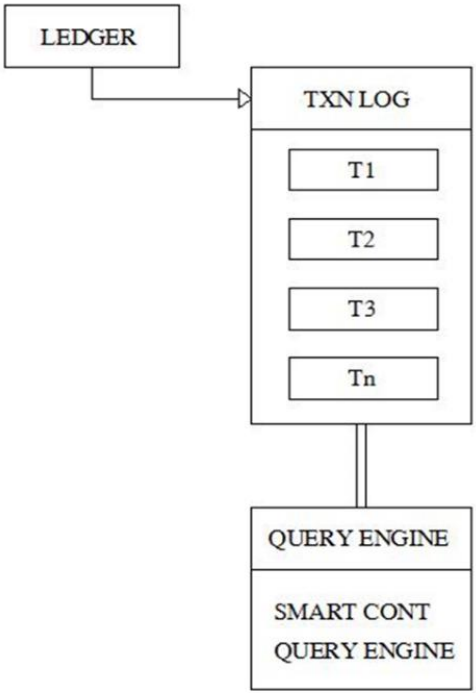

Figure 8. Distributed Ledger diagram

The second type of database state is the Query engine. A copy of the smart contract is attached to the Query engine, the request/queries coming in and out, are first validated

against the smart contract, once validated the query engine, processes the query and shows the output. If the smart contract detects an unauthorized request, the query engine will send a flag signal to the Blockchain admins with details of the unauthorized query. Blockchain admins can then track down the source of the unauthorized request.

4.2 Methods

4.2.1 Addition and Deletion of Actors.

Whenever a new Actor is added in the channel that is in any organization. The actors have to enroll with the Identity Provider. The Identity manager or mangers will provide the identity to the new actor based on the role, responsibilities, and permissions assigned to it by the core committee. The identity manager will also create the node for it and attach the smart contract to it. The Organization programs the roles, permissions and activities for that actor. The algorithms for the actor may be a set of functions. For Instance, If the company hires a project manager, then the identity manager will assign the smart contracts and activate the function () project manager in the validator. If the airline hires a System admin then the Identity manager will activate the function () System, an admin. A set of predefined functions should already be defined in the smart contracts and the incoming actors should be added under that category respective to the functions defined in the smart contracts.

It can also be the case that some actors are not defined in the set of categories. For those, customized functions have to be added for setting up roles, permissions, and responsibilities. For Instance, the airline company was not hiring an intern for last 5 years and this year they decided to hire 10 interns for 4 months, for the different type of algorithms has to be written to assign them permissions, roles, and responsibilities to the blockchain network and the database. After the assignment of the Identities to the actors,

the Channel CA, in this case, is the Fabric CA, which assigns the Digital certificates and PKI keys to the Actors based on their identities. The actors must use the Digital certs and PKI keys to Access, cipher and decipher the database if needed.

Whenever an employee leaves the company. The identity manager freezes its nodes, such that no requests get generates from that actor's node. Also, the second layer of security is added, such that the smart contract functions activated for that actors get void so even if some unauthorized person is successful in sending a request from that actor node, the validator (Smart contract) will not uphold its request. The third layer of security, which is the channel CA, deactivates the PKI keys and Digital certificates of the actor. For Instance, if a System administrator is leaving the firm, the identity manager deactivates the nodes. Secondly, the validator functions for that actor get void and thirdly, Channel CA deactivates the PKI keys and Digital certificates of the actor.

4.2.2. Permissions and Access Control for Actors using Smart Contract:

An Actor is identified by the Identity manager in the Blockchain channel. An Actor can be an individual or a category depending on the assignment of roles, responsibilities, and permissions by the company. Actions are always contained within transactions and requests. A transaction can be one or more atomic requests. Identities attached to the actors are used to authorize and authenticate the requests to and from Blockchain storage. Each identity is attached to the smart contract that contains the respective function that must be satisfied to allow the action associated with the request. The channel CA also provides Digital certificates to the actor. After the smart contract validates the Actor's request, the actor digital certificate is third layer security which acts as an authentication step wherein it verifies that the actor is indeed who he/she claims to be.

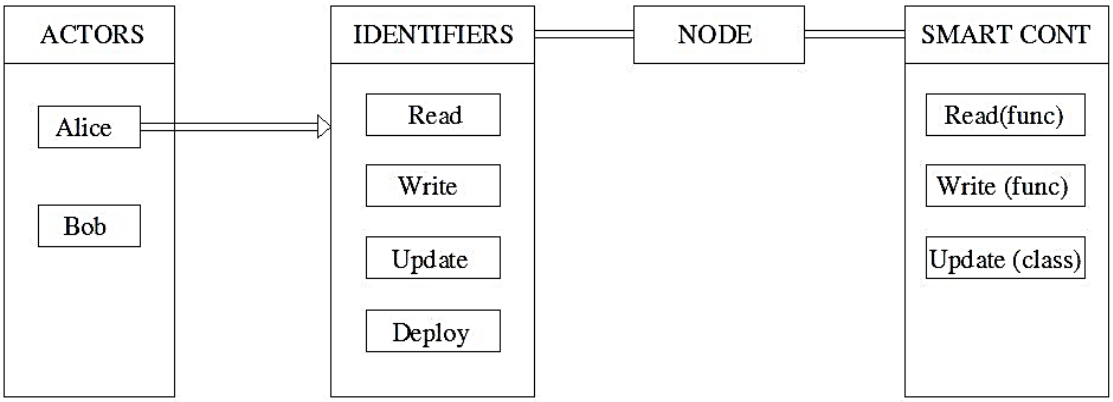

Figure 9. Access control model for each actor

The above example depicts a new employee named Alice that has been hired by Bogo airline. Alice has been hired as a Blockchain administrator. The identity manager based on its post will be assigned an identity to Alice. As we can see in the figure, the identity assigned to Alice has all the requests and permissions specified. For instance, the identifier assigned to Alice shows that she has admin, read, update, delete and program permissions assigned for the channel blockchain network. Now, there are two options to activate the functions in the smart contract corresponding to identities specified. First, to already created functions or class having functions for that category. E.g., blockchain administrator is a category and blockchain programmer programmed functions or class Blockchain Programmer () in the smart contract for that category. Now, when the identity manager assigns the identifier as Blockchain programmer, then the identifier will activate that Class Blockchain Programmer () or functions () corresponding to that category.

The second case is when a new employee, named Bob, is hired as a part-time intern. Then the identity manager will assign an identifier to Bob. Now if there is no such category as an intern, and then programmers will program customize functions or classes for that actor in the smart contract. The identifier will then activate the functions or class corresponding to the requests and permissions written in the identifier.

4.2.3. Storing Airline Data on Blockchain storage Network

Because computational and storage resources are getting cheaper and smaller in dimensions. Instead of creating the hash value of the traditional database and storing the hash value into the blockchain network, our model proposed the idea to store the data in the distributed ledger of the blockchain network itself, thus creating a blockchain storage/database. The streaming data used by the aviation data will be stored on the traditional database such as SQL, spark, etc. The Sensitive data of the customers, employers, and aircraft should be stored in the Blockchain storage. An alternate option would be to create separate ledgers in the channel. First ledger for Customer information, second ledger for employer's information and third ledger aircraft information such as maintenance records and aircraft path after every flight. All three ledgers are independent of each other.

The blocks in the blockchain network are created using the trusted consensus amongst the consensus admins. This will be explained in the brief next. Since we are storing the actual data on the blockchain network, the number of blocks should be less and data intake capacity per block should be high. Lesser the block numbers less is the computational resources used to hash the data in the block.

4.2.4. Creation of Blockchain using consensus by trusted Blockchain Admin.

In public blockchain consensus protocols are formed such as POW, POS, etc. to form new blocks since it is a trustless blockchain network. No committee as such is keeping an eye over the blockchain network. But, in a permissioned blockchain network, the trust-based consensus is the most important factor for the formation of the blocks in the blockchain network. The model proposes the idea of consensus voting to form a new block whenever needed.

A Blockchain consensus team is created whose work is to validate, verify and vote for the creation of the block. For instance, if there are 50 employees in the Blockchain consensus team, and there is a need to form a new block then a voting session is done, and the smart contract will run the algorithm to get the consensus. If the consensus is majority then the new block is formed with the new entries. In case of a tie in the consensus, the Blockchain consensus admin makes the veto vote. The veto vote is only applicable in case of a tie.

How do the Blockchain consensus members get rights? When an employee is hired in the Blockchain consensus team, the identity manager attaches an identifier concerning the Blockchain consensus permissions and rules. The identifier then invokes or executes the smart contract functions/class for the category Blockchain consensus. The smart contract will authorize the members to execute consensus and the digital certificates provided by Channel CA will act as an authentication mechanism.

4.2.5 Different Types of Blockchain Storage Networks in a channel.

Different types of blockchain storage mean that if Bogo airlines decide to have 3 separate blockchain storage for the different database then the Blockchain consensus committee will form 3 separate Blockchain storage networks, independent of each other. For instance, the Bogo airline committee decides customer data, employee data and airline data are most important. The blockchain committee will decide the formation of 3 separate Databases E.g., Customer data network, Employee data network, and Aircrafts data network.

It is the blockchain committee's call whether to make 3 separate blockchain consensus teams or a single team or one team for one network and another team for managing the other 2 networks. To allow this dynamic, our model proposes to give the

Blockchain committee the right to choose, but the smart contracts associated with each Blockchain storage network should be separate. For 3 blockchain networks, 3 separate smart contracts are associated with each blockchain network. Depending on the employee's roles, responsibilities and permissions the identifier attach the smart contract/contracts to the employee's node.

For Instance, If Bogo airline hires James as a Blockchain architect for all 3 Blockchain networks, then the identifier will attach 3 smart contracts to the James node and assigns a digital certificate as per those conditions. Similarly, if they hire Michael as a Customer database admin that is related to the Customer data blockchain network, then the identifier will attach smart contracts related to a customer data network to Michael's node.

4.3. The Blockchain Architecture

The Blockchain Architecture encompasses a chain of blocks connected such that every block is connected to the previous block by the previous block's ID. For instance, Block 15 is connected to Block 14 by having block 14's ID and Block 16 is connected to block 15 by having block 15's ID. A Block structure is comprised of a Block ID, Previous block ID, timestamp, data entries (T1, T2, T3, TN) and Merkle Root hash.

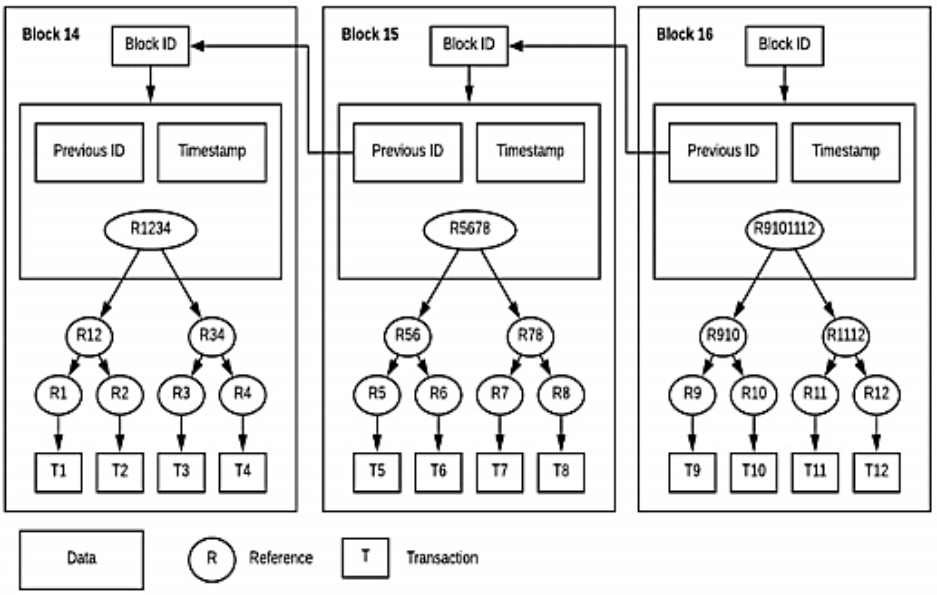

Figure 10. Blockchain architecture model for Storing Data

The block header is generated by calculating the Hash Value of all the fields in the Block including the Block Name, Data Messages, and previous Block Header. A Merkel Root Hash is used to calculate the Hash Value. When a new block is formed, it creates a hash value that is stored in the previous block. This is the way a new block gets added to the Blockchain Network. The Hash Value ensures immutability for the Blockchain Network. If a hacker changes a Data Message in any block, the hash value of that block then changes, causing it not to match its old hash value stored in the previous block. As a result, a flag is generated, and that block will be frozen or deleted. However, Data will not be deleted because the Blockchain uses distributed ledger technology. In our Blockchain architecture, there is a customization wherein a second hash value is also calculated. After, getting the final Merkle root hash, a second hash value is generated from Previous Block ID, Timestamp and Merkle Root Hash. That Hash value is called that block's ID. For Instance, Block 15, calculate a Final Merkle root hash from all the Data entries in it. It also

has the timestamp when that block was created and the previous block ID. The second hash value is created using the above 3 values and that value is Block 15's ID.

An unauthorized change in any data value will result in changing the final Merkle root hash value of that block, change in Merkle root hash will change the second hash value which is block ID which will result in a mismatch with the hash value stored in the previous block. In our model, the data originally stored in the distributed ledger is divided into the block size using a pointer and depends on the block size decided by the admins. For instance, DLT has 1000 rows in the database server and the block size decided by the admins is 100 data (T1, T2…TN), then the 1000 rows will be divided by 100 and 10 blocks will be needed to store the data.

4.4. Evaluation and Analysis

Security analysis: our main purpose is to achieve data security by fulfilling the following parameters such as Confidentiality, Data integrity, authorization, and authentication. Confidentiality is termed as protecting the data from being access by unauthorized entities. Only the authorized entities should be able to access the data. A failure to maintain confidentiality means that an unauthorized entity was able to steal/hack the sensitive data. Such an incident is known as breach of data, typically cannot be remedied. In our model, the breach of confidentiality is when the customer data, employee data, and flight/aircraft data is lost or hacked, resulting in a major leak of data. The customer and employee data contain much sensitive personal information such as credit card details, address, etc., which can be exploited by hackers who gain the data. Our model's whole focus is to maintain the confidentiality of the data. Also, unlike the public blockchain wherein anybody can access the data, this is a permissioned blockchain network, only the authorized entities can access and query the data.

Authorization is termed as a security measure to determine the entity privileges or access rights over the system resources such as data, files or any other sensitive features. Authorization is normally preceded by authentication for user identity verification. For example, System admins are authorized to install, update and copy any data in the company because they have been given the rights and permissions to do so. In this model, the authorization mechanism is achieved by Identifier. The identity manager assigns each actor/employee with an identifier. The identifier has all the roles, rights and permissions assigned to that actor. The permissions assigned in the identifier then invoke/activates the functions/class in the smart contract. Due to the identifier, the actor can only request what has been included in the identifier. By this, the actor is authorized to issue or access the rights and permissions specified in the identifier.

Authentication is a mechanism wherein it indicates that if the entity that it claims to be, is who or what it declares itself to be. Authentication is a type of access control by checking if the user's credentials/digital certificates match with the credentials provided in the database to authorize him. For Example, A company has its backend servers on Azure cloud and to manage the backend environment, it gives the username and password to its system admin so that he can log in (authenticate) and manage the server. Similarly, in our model, the authentication mechanism is taken care of by the channel CA. Once, the identity manager assigns the identifier to the actors, the channel CA, based on the identifier, assigns the digital certificate to the actor. For instance, if a System admin has permission to read the blockchain database while accessing the database, he has to use the digital certificate. The smart contract will validate the digital certificate, to determine whether the actor has the right to read the database. Then the actor can read data from the database. But, if the system admin tries to update the blockchain database using his digital certificate then he

will be denied. The fallback of the digital certificate is to keep it secure, if a hacker gets access to your digital certificate then he can access the data.

The main component of our model is to provide Data integrity and security. Data integrity means that the data is unchanged over time. Data integrity is the process of keeping data intact from hackers and unauthorized entities. The blockchain Storage network ensures data integrity as explained above in the Blockchain architecture model. The integrity of the data messages is assured by immutability factors provided by the blockchain network. Blocks in the chain are connected via the prior block hash. For instance, block 4 will have a hash value of block 5. A hash value of the block is generated from all the fields in that block and previous block hash. If a hacker tries to alter a message in any block, its hash value changes and is no longer equal to its old hash value in the previous block. Consequently, the network will detect the altered block and freeze it. The altered data message is unaffected because it is duplicated across all the blocks in the common distributed ledger.

5. SECTION V

5.1. CONCLUSION

IoT security is gaining a lot of attention in both academia and industry. Existing security solutions are not suited for IoT due to its high energy consumption and processing overhead. We proposed a Blockchain-based approach to implement data communication security and data storage security between a Flight and Air Controller Tower (ACT). The various core mechanisms and phases used to secure the data communication and storage were outlined. Moreover, pertinent security and privacy issues regarding the implementation were extensively analyzed.

Aviation data security is a major issue, not only for the aviation industry but also for major privacy and breach of data for the customers of the airline industry. Important customer data such as passport details, credit card info and other payment details are saved in aviation databases. We proposed a Blockchain-based approach to implement data security and integrity in the aviation industry. Various core components and different methods are outlined to secure the data. Moreover, pertinent security, confidentiality, authenticity, authorization, and Data integrity issues regarding the implementation were extensively analyzed.

The paper one has provided a roadmap of solutions to address the issues concerning data storage and communication using Blockchain-Based approaches by presenting an overview of the different phases of communication between the Flight and ACT. Paper two proposes methods to maintain the authenticity, confidentiality, authorization and Data integrity of customer data stored in the aviation databases using a blockchain-based storage network. It also proposes methods for access control for employees and external entities for the blockchain database.

6. SECTION VI

6.1 FUTURE DIRECTIONS

One of the main fallbacks of Blockchain technology, in general, is high computational resources and processing power. Due to hashing generated constantly, it needs high computational resources and processing power. We have seen the fact that computational resources are getting cheaper, smaller, faster and powerful. Hence, we have assumed that future devices will have stronger and faster computational power as compared to today. That is why we have designed the model based on the fact that high and expensive computational resources will be needed.

Our future research will be to make the model as optimized as possible. Optimizing the model is very challenging and time-consuming hence we need funds and more people to work with us but this will be one of our future goals. Also, my Blockchain idea is to make Blockchain to act as a database with Blockchain features like reliability, traceability and Data integrity to add the security aspect. Currently, we are proposing this model to the Aviation industry but in the future, we want to optimize it as much as possible. Due to a lack of better Software engineering skills, for now, I lack Optimization skills but in the future, this will be the main focus of my research. Also, my second aim will be to integrate Deep learning with my model.

Second, the digital certificate can still be hacked by the hacker while in transit. This is one of my main goals is to encrypt the digital certificate. The only problem is that the public key in encrypted in the digital certificate, Hence, I need to find an alternative way to encrypt the Digital certificate. This itself is a wide research domain.

7. REFERENCES

[1] Controller Pilot Data Link Communications (CPDLC), Skybrary, https://www.skybrary.aero/index.php/Controller_Pilot_Data_Link_Communications_(CPDLC), Last edited on 26 January 2019.

[2] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” Computer Networks, vol. 57, no. 10, pp. 2260–2280, 2013.

[3] Amlan Chatterjee, Hugo Flores, Soumya Sen, Khondker S. Hasan, Ashish Mani, "Distributed Location Detection Algorithms using IoT for Commercial Aviation", 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Published by IEEE, December 2017.

[4] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,”, cryptography mailing list at metzdowd.com, October 2008.

[5] A. Dorri, S. S. Kanhere, and R. Jurdak, “Blockchain in the internet of things: Challenges and solutions,” arXiv preprint ar X iv:1608.05187, 2016.

[6] H. G. Do and W. K. Ng, "Blockchain-Based System for Secure Data Storage with Private Keyword Search," *2017 IEEE World Congress on Services (SERVICES)*, Honolulu, HI, 2017, pp. 90-93. DOI: 10.1109/SERVICES.2017.23.

[7] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain For Large-Scale Internet of Things Data Storage and Protection," in *IEEE Transactions on Services Computing*.
DOI: 10.1109/TSC.2018.2853167.

[8] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in Agri-Food supply chain management: A practical implementation," *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*, Tuscany, 2018, pp. 1-4.

[9] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Kona, HI, 2017, pp. 618-623.

[10] Michael Schukat, Pablo Cortijo, "Public key infrastructures and digital certificates for the Internet of things", 2015 26th Irish Signals and Systems Conference (ISSC).

[11] Alharby, Maher & van Moorsel, Aad. (2017). A Systematic Mapping Study on Current Research Topics in Smart Contracts. International Journal of Computer Science and Information Technology. 9. 151-164. 10.5121/ijcsit.2017.9511.

[12] A. Nayak and K. Dutta, "Blockchain: The perfect data protection tool," *2017 International Conference on Intelligent Computing and Control (I2C2)*, Coimbatore, 2017, pp. 1-3.

[13] Qizhi Qiu and Qianxing Xiong, "Research on elliptic curve cryptography," 8th International Conference on Computer Supported Cooperative Work in Design, Xiamen, China, 2004, pp. 698-701 Vol.2.

[14] Y. Chen, L. Wang, and S. Wang, "Stochastic Blockchain for IoT Data Integrity," in IEEE Transactions on Network Science and Engineering.

[15] K. S. Bhosale, M. Nenova, and G. Iliev, "The distributed denial of service attacks (DDoS) prevention mechanisms on the application layer," 2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), Nis, 2017, pp. 136-139.

[16] Restuccia, Francesco & d'oro, Salvatore & Kanhere, Salil & Melodia, Tommaso & Das, Sajal. (2018). Blockchain for the Internet of Things: Present and Future.

[17] Khondker S. Hasan, Hazera Nasreen Razvi, Amlan Chatterjee, “A New Distributed Task Look-ahead Predictive Model for Optimized Job Allocation”, 3rd IEEE International Conference on Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT-2018), Sponsored by IEEE Bangalore, Mysuru, India, December 2018.

[18] Sugita M, Miyakawa M. Economic analysis of the use of counterfeit drugs: health impairment risk of counterfeit phosphodiesterase type 5 inhibitor taken as an example. Environ Health Prev Med. 2010 Jul;15(4):244–51. DOI: 10.1007/s12199-010-0134-5.

[19] trade in fake medicines. Expert Opin Drug Saf.; 16(5):587–602. DOI: 10.1080/14740338.2017.1313227, May 2017.

[20] Amlan Chatterjee, Hugo Flores, Soumya Sen, Khondker Hasan, Ashish Mani, “IoT based algorithms for distributed location detection for flights”, International Journal of Hybrid Intelligence (IJHI), Inderscience Publishers, Onley, UK, June 2018.

[21] Chapron G., The environment needs crypto governance, Nature.;545(7655):403–5. DOI: 10.1038/545403a, May 22, 2017.

[22] E. Bertino, L. R. Khan, R. Sandhu, and B. Thuraisingham, "Secure knowledge management: confidentiality, trust, and privacy," published in IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 36, no. 3, pp. 429-438, May 2006.

[23] Khondker S. Hasan, John K. Antonio, Sridhar Radhakrishnan, "A model-driven approach for predicting and analyzing the execution efficiency of multi-core processing", International Journal of Computational Science and Engineering (IJCSE), INTERSCIENCE Publishers, Vol: 14, No 2, Page 105 – 125, Olney, Bucks, UK, March 2017.

[24] W. Luo and G. Bai, "Ensuring the data integrity in cloud data storage," 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, pp. 240-243, Beijing, 2011.

[25] Y. Zhang, M. Qiu, C. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare Cyber-Physical System Assisted by Cloud and Big Data," in IEEE Systems Journal, vol. 11, no. 1, pp. 88-95, March 2017.

[26] Khondker S. Hasan, Amlan Chatterjee, Sridhar Radhakrishnan, and John K Antonio, "Performance Prediction and Analysis of Compute-intensive Tasks on GPUs", The 11th IFIP International Conference on Network and Parallel Computing (NPC-14), Sept. 2014, Lecture Notes in Computer Science (LNCS), Springer, ISBN: 978-3-662-44917-2, Vol: 8707, pp 612-17, Berlin, Germany, 2014.

[27] F. Dai, Y. Shi, N. Meng, L. Wei, and Z. Ye, "From Bitcoin to cybersecurity: A comparative study of blockchain application and security issues," 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, pp. 975-979, 2017.

[28] Amlan Chatterjee, Hugo Flores, Soumya Sen, Khondker Hasan, Ashish Mani, "IoT based algorithms for distributed location detection for flights", International Journal of Hybrid Intelligence (IJHI), Inderscience Publishers, Onley, UK, June 2018.

[29] Controller Pilot Data Link Communications (CPDLC), An Introduction to CPDLC Operations, Skybrary, https://www.skybrary.aero/index.php/Controller_Pilot_Data_Link_Communications_(CPDLC), Circular 90-117, October 2017.

[30] V. Saraogi, "Five times airports were involved in cyberattacks and data breaches," *Airport Technology*, 30-Jan-2020. [Online]. Available: https://www.airport-technology.com/features/five-times-airports-were-involved-in-cyberattacks-and-data-breaches/. [Accessed: 08-Mar-2020].

[31] Dark Reading. (2020). *Buckle Up: A Closer Look at Airline Security Breaches*. [online] Available at: https://www.darkreading.com/threat-intelligence/buckle-up-a-closer-look-at-airline-security-breaches/d/d-id/1333336 [Accessed 8 Mar. 2020].

[32] A. Arora and S. K. Yadav, "Batman: Blockchain-based aircraft transmission mobile ad hoc network," in Proceedings of 2nd International Conference on Communication, Computing and Networking, C. R. Krishna, M. Dutta, and R. Kumar, eds. (Springer Singapore, Singapore, 2019), pp. 233–240.

[33] R. J. Reisman, "Air traffic management blockchain infrastructure for security, authentication, and privacy," (2019).

8. APPENDIX A:

AUTHORIZATION BY CO-AUTHOR, AMLAN CHATTERJEE

CSUDH

DEPARTMENT OF
COMPUTER SCIENCE

Amlan Chatterjee, PhD
Assistant Professor, Department of Computer Science
California State University Dominguez Hills
NSM E-113, 1000 E. Victoria St., Carson, CA 90747, USA
achatterjee@csudh.edu ☎ (310) 243-3240

February 13, 2020

To whom it may concern:

Subject: Authorization for Yusuf Zakir to use content from the paper titled "Improving Data Security in Message Communication between ACT and Aircraft using Private Blockchain"'

As co-author of the paper "Improving Data Security in Message Communication between ACT and Aircraft using Private Blockchain"', I, Amlan Chatterjee, authorize Yusuf Zakir to use the contents from the aforementioned paper for inclusion in his masters thesis report.

If you have any questions, please feel free to contact me.

Sincerely,

Amlan Chatterjee

Amlan Chatterjee, PhD
Assistant Professor, Department of Computer Science
California State University Dominguez Hills
achatterjee@csudh.edu ☎(310) 243-3240

9. APPENDIX B:

AUTHORIZATION BY CO-AUTHOR, NAOMI WIGGINS

Naomi S. Wiggins
Naomiwiggins08@gmail.com
713-494-8659

February 12, 2020

Subject: Authorization for Yusuf Zakir to use content from "Improving Data Security in Message Communication between ACT and Aircraft using Private Blockchain"

To Whom it may Concern,

As co-author of "Improving Data Security in Message Communication between ACT and Aircraft using Private Blockchain", I, Naomi Wiggins, authorize Yusuf Zakir to take content from the aforementioned paper for the purposes of writing his master thesis report.

Sincerely,

Naomi Wiggins

Naomi S. Wiggins

10. APPENDIX C:

AUTHORIZATION BY ORGANIZING COMMITTEE

Subject: Authorization for **Yusuf Zakir** to use content from "Improving Data Security in Message Communication between ACT and Aircraft using Private Blockchain"

To Whom it may Concern,

As the organizing committee of 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)", We, authorize **Yusuf Zakir** to take content from the aforementioned paper for the purposes of writing his master thesis report.

Sincerely,

Sandra Sendra Compte

Prof. Sandra Sendra Compte

IOTSMS 2019 Conference Chair