ClaimChain: Secure Blockchain Platform for Handling Insurance Claims Processing

Naga Ramya Bhamidipati*, Varsha Vakkavanthula*, George Stafford*, Masrik Dahir*, Roshan Neupane*,
Ernest Bonnah, Songjie Wang, J. V. R. Murthy, Khaza Anuarul Hoque, Prasad Calyam
University of Missouri-Columbia, USA; Jawaharlal Nehru Technological University, India.
{nbny6, vvcnm, ghstgf, eb7z3}@mail.missouri.edu, {neupaner, wangso, hoquek, calyamp}@missouri.edu,
dahirma@vcu.edu, director.ic@jntuk.edu.in

***Abstract*—Insurance claims processing involves multi-domain entities and multi-source data, along with a number of human-agent interactions. Consequently, this processing is traditionally manually-intensive and time-consuming. Blockchain technology-based platforms for intelligent automation can significantly improve the scale and response time of claims processing. However, there is a need to secure such platforms against fraud (e.g., duplicate claims) and the loss of data integrity caused due to cyber-attacks (e.g., Sybil attack). In this paper, we propose a novel "ClaimChain", a consortium Blockchain platform that transforms the state-of-the-art NICB/ISO database architecture approach through increased shared intelligence and participation of insurance companies. ClaimChain features include: (a) automation of insurance claim processing via implementation of a Blockchain infrastructure, (b) infrastructure-level threat modeling via attack tree formalism for data integrity attacks, and (c) application-level fraud modeling for identified prominent red flags through machine learning models and risk scoring on the basis of risk severity. We evaluate the scalability of ClaimChain by simulating realistically large number of Blockchain transactions of claim processing. Further, we show that data integrity attacks at the infrastructure-level can be mitigated (as seen in reduction of 24% probability in loss) through implementation of security design principles. We also perform fraud-detection over an open dataset in ClaimChain to show how machine learning models can detect fraudulent activity with 98% accuracy.**

***Index Terms*—Insurance claims processing, Blockchain, Attack trees, Fraud detection, Statistical model checking**

I. Introduction

The insurance industry manages auto-insurance claim processing through information from multi-domain entities such as e.g., police, county administrators, insurance agents and healthcare professionals [1]. These entities collaborate to share multi-source information that is critical for insurance companies to properly adjudicate policy holder claims. However, most of the current claim handling processes are manually handled and time-consuming due to lack of automation mechanisms for data collection/analysis, as well as technologies to perform trustworthy decision making. Thus, there is a need for integration of intelligent automation and trust management frameworks *at the application-level* to improve the efficiency and scalability of insurance claims processing.

* These authors contributed equally to this work.

This material is based upon work supported by the National Science Foundation under award number CNS-1659134, and the US Department of State under the US-India Partnership 2020 award number 00071769. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the US Department of State.

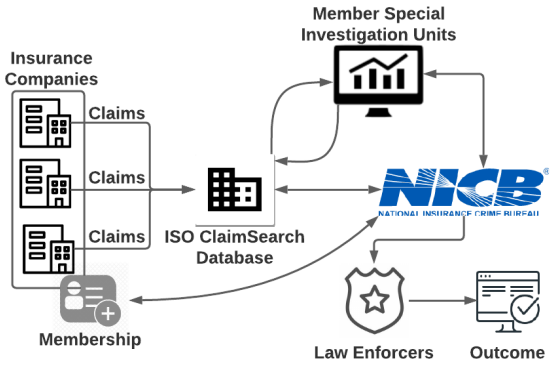

Fig. 1: NICB/ISO database used by multi-domain entities (Special Investigation Units, Law enforcers) to perform fraudulent claims investigations.

Online insurance claim handling systems hosted in local/cloud storage platforms are also prone to cyber attacks. These attacks could result in loss of data confidentiality (e.g., through stealing of sensitive personal information such as social security numbers of policy holders, their personal property details), and more importantly lead to loss of data integrity that can result in fradulent claims. Consequently, there is a need to ensure there are relevant threat models and security mechanisms *at the infrastructure-level* in insurance claim handling systems to mitigate the impact of cyber attacks.

Fig. 1 shows the state-of-the-art approach used in the auto-insurance industry to handle the automation, security and trust issues. The method involves a national-level consortium viz., National Insurance Crime Bureau (NICB) [2], which operates as a non-profit membership organization in order to maintain a database of several major insurance industry memberships. NICB also collaborates with the Insurance Services Office (ISO) [3], which audits all the claims that are processed by the participating NICB members. The overall collaboration thus helps the insurance industry members of NICB in identifying fraudulent activities and for pursuing corrective measures (e.g., fraud mitigation efforts, prosecution of insurance fraud) in questionable claims [4].

There are a number of obvious problems within the current database architecture approach used in the NICB/ISO system. Firstly, only the participating members have access to the database, and need to pay significant annual member fees that smaller insurance companies cannot afford. Thereby, a fraudster could initiate two insurance policies, one with an NICB member, and another with a non-member - in order

to submit duplicate claims and obtain double compensation. Secondly, there are security and privacy issues in accessing the database that may lead to unauthorized access that could lead to loss of data integrity (e.g., data manipulation through an injection attack). These problems motivate the idea of using a consortium Blockchain approach that can transform the state-of-the-art NICB/ISO database approach in order to perform intelligent automation with multi-source data by involving multi-domain entities in a manner that is trustworthy and secure at the infrastructure and application levels.

In this paper, we propose "ClaimChain", which is a consortium Blockchain platform for auto-insurance claims processing with increased shared intelligence and participation of insurance companies of all sizes. Our ClaimChain utilizes the benefits of Blockchain technology [5] and thus presents a superior/secure solution than the NICB/ISO database architecture approach. Specifically, our Blockchain-based solution approach provides benefits such as e.g., decentralized architecture to manage trust, data transparency, immutability as well as auditability. Realizing that a Blockchain-based solution is also prone to have attack surfaces [6], we devise schemes to improve both infrastructure-level as well as application-level security. In order to provide infrastructure-level security in ClaimChain, we present a novel threat model based on attack tree formalism [7] and employ security design principles to counter data integrity attacks. In addition, we present an application-level fraud model to identify prominent NICB-identified red flags through the use of machine learning and a risk scoring scheme for accurately detecting fraudulent claims in the data hosted via ClaimChain.

We implement and evaluate our ClaimChain system using a realistic testbed built using a Hyperledger Composer instance hosted in Amazon Web Services. We first evaluate the scalability of ClaimChain through simulation of realistically large number of Blockchain transactions of claim processing, and compare the performance of ClaimChain with a state-of-the-art CioSy system [8] (which uses Ethereum) for different operations of issuance, approval and cancellation pertaining to insurance policies. Next, we perform a qualitative analysis of ClaimChain with other existing Blockchain-based solutions for different insurance industry applications. Following this, we use a formal verification tool called UPPAAL [9] to show how our ClaimChain approach allows for mitigation of data integrity attacks at the infrastructure-level through the implementation of careful mix of security design principles (i.e., hardening, sufficient documentation, principle of privilege attenuation) [10] [11]. Lastly, we show how our ClaimChain approach enables fraud-detection over an experimental dataset [12] through use of machine learning. Specifically, we experiment with XGBoost, KNN, RCF and LR machine learning models and show how we can detect fraudulent activity with high accuracy.

The remainder of the paper is organized as follows: Section II discusses related works. Section III presents our ClaimChain approach for insurance claims processing. Section IV presents our schemes for improving infrastructure-level and application-level security. Section V describes the performance evaluation of ClaimChain's threat model for securing at the infrastructure-level, and the fraud model for securing at the application-level. Section VI concludes the paper.

II. Related Works

A. Blockchain in Insurance Claims Processing

Blockchain offers transparency and auditability enabling distributed trust amongst participating peers. Smart contracts in Blockchain also reduce operation and maintenance cost, and improve processing time [13]. Authors in [14] propose a Blockchain framework for insurance claims handling by exchanging documents as part of collective knowledge sharing across multi-domain entities involved, thus increasing accessibility and reducing discrepancies. Similarly, the authors of [15] explored the possibility of Blockchain integration in the insurance industry from a regulation point of view. They showed how auditability enables transparency and compliance with regulation. WISChain [16] introduces a Blockchain framework for insurance companies and claimants. They briefly address security issues via their design of a browser extension for web identity security through password protection. Authors in [13] showed how a shared ledger approach helps insurance companies and third parties to calculate premiums with a low risk and high trustworthiness.

The above review of related works strengthen our argument that Blockchain can be beneficial in the context of insurance claims processing. In addition, our review clearly shows that there is a dearth of works that address threat modeling and security design principles at both the infrastructure and application levels for Blockchain-based insurance claims processing. The novelty of our work is in our approach to add a security layer on top of a Blockchain solution for insurance claims processing that enables us to analyze impact of various attack scenarios and employ security design principles in an effort to reduce their probability of occurrence.

B. Attack Modeling in Blockchain Solutions

Despite many advantages, the use of Blockchain opens new attack surfaces [6]. Though secure than traditional database systems, Blockchain-based solutions are vulnerable to various attacks that could lead to loss of data integrity. Hence identifying attack scenarios and performing pertinent threat modeling is essential. Security issues on cyber-physical systems have been extensively studied in prior works. However, there are relatively few works on categorization of attacks on Blockchain platforms. For instance, the work in [17] categorized common network attacks on Blockchain-based solutions and suggested countermeasures to reduce vulnerabilities. Risk engineering techniques are detailed in [18] in the context of a Blockchain-based system to model different threats.

A few prior works adapted formal modeling of threats for cyber-enabled systems. For instance, authors in [19] discussed different threats on an electric cyber-physical system using the attack tree formalism. They calculate overall risk assessment of attacks based on their probabilities and impact through an Analytic Hierarchy Process (AHP). Authors in [7] used attack trees for modeling of security and privacy concerns in

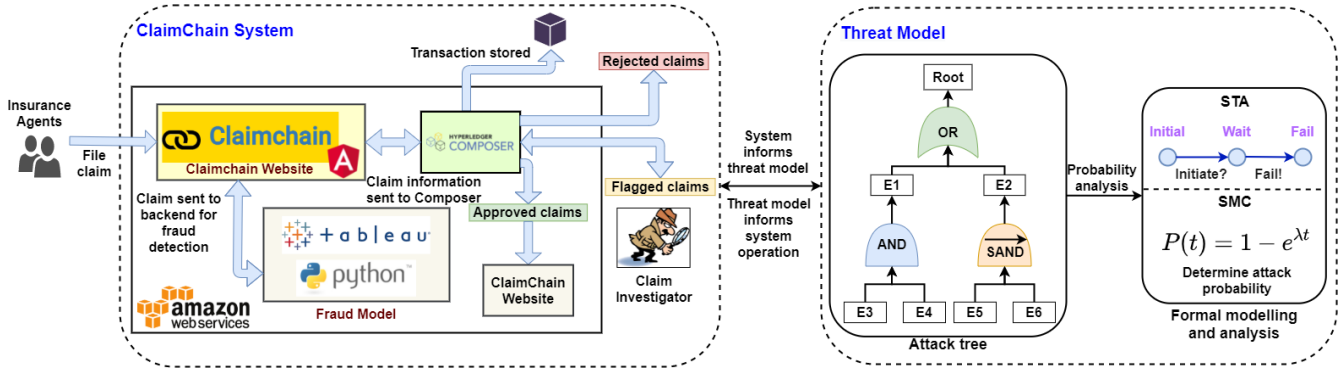

Fig. 2: ClaimChain system architecture for improving security and trust in insurance claims processing, featuring: infrastructure-level threat modeling based on attack trees and application-level fraud modeling using ML models.

social virtual reality learning environments. They performed quantitative analysis on the threat scenarios through stochastic timed automata representations allowing them to perform rigorous statistical model testing. Our work borrows the ideas of risk assessment calculation and use of attack trees from above works, however - we are the first (to the best of our knowledge) to extend these concepts to a Blockchain-based solution for insurance claims processing.

Although formal threat modeling is key to understand different attack vectors on Blockchain-based solutions, above works only focus on specific attacks and do not provide a comprehensive threat modeling strategy as done in this ClaimChain work. Our novelty is in the use of attack tree formalism for threat modeling of various attack scenarios in ClaimChain, and in the method to create a mix of security design principles to reduce the probability of attacks.

C. Fraud Detection in Insurance Claim Processing

Insurance industry faces several types of insurance fraud on a routine basis. The seriousness of insurance fraud ranges from simple exaggeration of claims, to staging of accidents, and all the way to intentionally create damage on insured assets. As the insurance claim processing could involve multiple fraud scenarios, there is a need for effective fraud detection models that have high accuracy to help in identification of fraudulent activities and for pursuing corrective measures toward their mitigation. Several prior works focus on detection of fraud in insurance claim processing using machine learning techniques. In one of the exemplar works [20], authors utilized an available insurance claims dataset and proposed an expert detection system to understand fraud practices by tracking the trends. Authors in [21] and [22] described various types of financial fraud and proposed machine learning and data mining techniques to overcome the hurdles to prevent monetary losses. The works in [23] and [24] proposed secure and automated insurance systems that help in reduction of human involvement in detection of fraudulent claims, thus reducing financial costs for an insurance provider.

Above works lack the consideration of major NICB-identified red flag scenarios that can effectively categorize fraud activities based on their risk levels as done in this work of ClaimChain. Our review of above works also strengthens our argument that an effective fraud detection model that is standards-based is essential in detecting fraud and reducing losses for insurance providers. The novelty of our approach is in performing fraud detection considering various red flag scenarios identified by NICB by assigning risk scores to influence corrective actions based on severity of fraudulence in insurance claims.

III. ClaimChain: Consortium Blockchain Solution

In this section, we first describe the ClaimChain architecture and compare it with the traditional NICB/ISO database architecture. Following this, we detail the ClaimChain system functioning and identify security as well as trust issues.

A. ClaimChain Architecture

Fig. 2 depicts our overall ClaimChain system architecture. ClaimChain uses the Hyperledger Composer, a Linux Foundation project for developing Blockchain platforms. Insurance agents today use user-interfaces that allow them to instantiate and cancel auto-insurance policies, as well as issue, query, or approve insurance claims. We have developed a similar user-interface using Angular for ClaimChain. All the peers i.e., participating insurance companies are connected to a Hyperledger Composer through this user-interface for initiating a transaction i.e., a claim. The user transactions are validated and inserted into a block and dispersed within a shared Blockchain.

Two key components of our ClaimChain architecture are the threat model for enhancing the infrastructure-level security, and the fraud model for enhancing the application-level security. Our threat modeling uses the attack tree formalism [7] to identify different data integrity attack scenarios. The probability of occurrence of the attacks at the infrastructure-level is quantified by analyzing the ClaimChain related attack tree using Statistical Model Checking tools such as UPPAAL [9]. Details on our threat modeling at the infrastructure-level are presented in Section IV-A. We also use a fraud model that identifies fraudulent claims by monitoring data obtained while handling user queries in application-level operations. Our fraud modeling utilizes supervised machine learning to check for fraudulent activities based on the NICB-identified red flags to accurately detect fraud incidents. Details on our fraud modeling at the application-level are presented in Section IV-B.

TABLE I: Comparison of ClaimChain and NICB/ISO database architecture approach for insurance claims processing.

Attributes	NICB/ISO Database Architecture	ClaimChain Architecture
Architecture	Client/server architecture	Peer-to-peer architecture
Authority	The database is centralized in nature	ClaimChain uses Blockchain which is decentralized
Transparency	NICB administrators only can decide what data to be made public	ClaimChain offers transparency in data
Privacy	Requires authorized privileges for data access	Permissioned ledger for consortium members
Integrity	NICB uses database that can be altered by malicious actors and can lose data integrity	ClaimChain supports integrity in data as any update made is validated through consensus algorithm
Data Handling	The data can be erased or replaced as databases utilize CRUD (Create, Read, Update, Delete)	ClaimChain offers immutability meaning no data tampering is possible within the network

B. ClaimChain vs. NICB/ISO Database Architecture

Automobile insurance is a multi-billion dollar industry with millions of claims being processed every year [25]. Table I shows the detailed comparison of our proposed ClaimChain architecture with the state-of-the-art approach used in the auto-insurance industry that is based on a NICB/ISO database architecture. Recall, the structure and major drawbacks of the NICB/ISO database approach were discussed earlier in Section I. Our comparison is based on a number of critical attributes such as: Architecture, Authority, Transparency, Privacy, Integrity and Data Handling.

It is obvious from the advantages seen in the ClaimChain architecture that there is an imminent transformation in employment of a consortium Blockchain-based system that needs to happen to the traditional database architecture being used in the auto-insurance industry. More specifically, due to the qualities of transparency, immutability, and distributed trust, a Blockchain-based solution such as ClaimChain is a definite alternative to the NICB/ISO database architecture. Further, a consortium Blockchain platform can increase participation of insurance companies without prohibitive membership fees. Thus, it can eliminate the barrier-for-entry for smaller insurance companies that can lead to collective benefits for the insurance industry (e.g., duplicate compensation issue detailed in Section I). Further, the security layering in ClaimChain at the infrastructure and application levels allows for increasing trustworthiness of the Blockchain-based system. The increased trustworthiness can be achieved even when different multi-domain entities are involved in the transactions made within the Blockchain that increase the overall attack vectors.

C. ClaimChain System for Insurance Claims Processing

ClaimChain uses Blockchain transactions to implement insurance claim processing tasks. These transactions include actions such as: issue claim, approve claim, and cancel policy. ClaimChain includes a **smart contract** in the form of a code segment that is pre-approved by consortium peers for manipulating an **asset** (i.e., an automobile policy/claim) in the Blockchain. The execution of a smart contract in ClaimChain is recorded as a **transaction**. **Endorsing peers** in ClaimChain are responsible for validating a participant's Blockchain transactions. We also consider a **World state** that represents the actual record of assets tracked by the Blockchain. ClaimChain keeps records of all the issued claims and policies from peers. Each participant in ClaimChain has a **Certificate Authority (CA)** for defining which contracts they have access to within a peer, and for signing off on those performed for future auditability. Lastly, we use the concept of an **Orderer** in ClaimChain that is responsible for creating a block and readying it for other peers to perform commit actions. Orderers ensure the validity of a transaction by checking the attached endorsement information.

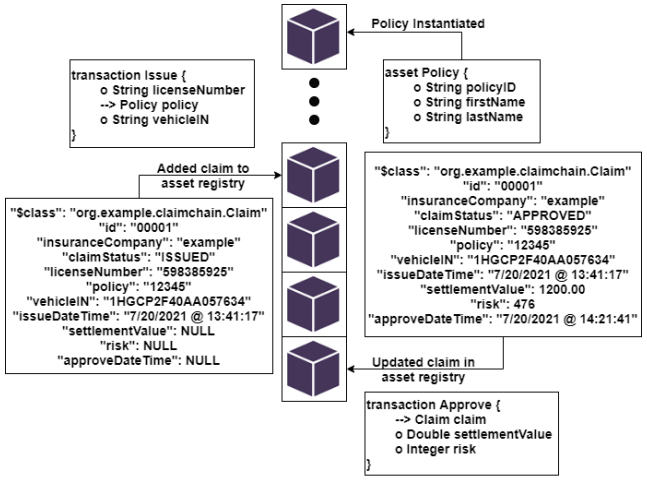

Fig. 3: Transaction issuance and approval processes within the functions involved in the Hyperledger Chaincode.

Fig. 3 depicts the lifecycle of a ClaimChain claim asset. An insurance agent receives a claim from a policy holder, initiates the 'issue claim' smart contract, and appends their CA. Claim information such as the policy holder's license number, policy id, and vehicle identification number are inserted into a claim asset. The peer validates the transaction through simulation and endorses it by appending their CA. The client application continues to collect signatures from Endorsing peers in the channel until endorsement is achieved as specified by the endorsement policy. Then the peer makes a request to the orderer for recording the transaction on the Blockchain. Once the transaction's order is determined, it is packaged into a block and distributed to peers on the network for addition to their record. Only after the transaction is securely recorded is the claim asset marked as issued and inserted into the World state. During fraud detection, the fraud model queries our World state to check e.g., if duplicate claims are already present. In addition, it checks for various NICB-identified red flags in the claim information that might signal fraud.

If the claim is approved, the agent initiates the 'approve claim' smart contract and appends their CA. The 'approve claim' contract uses a passed claim identifier to select a claim from the World state, attaches risk score and settlement value, and declares it approved by the organization. Again, the peer validates the transaction through simulation and endorses it by appending their CA. The client application continues to collect signatures from the Endorsing peers in the channel until endorsement is achieved as specified by the endorsement policy. Then the peer makes a request to the Orderer for recording the transaction on the Blockchain. Once the transaction's order is determined, it is packaged into a block and distributed to peers on the network for addition to their record. Only after the transaction is securely recorded is the claim marked as approved and updated in the World state. After a claim asset has been approved, it can no longer be edited but persists in the World state for future reference. The 'cancel policy' transaction is invoked to close out a policy with the organization. Similarly, after a policy has been canceled, it cannot be interacted with but persists in the World state.

D. Security and Trust Issues in ClaimChain

Even though insurance data and related information are better secured on the Blockchain, adoption of a Blockchain-based solution engenders new attack vectors. Hence, we need to enhance ClaimChain security by considering loss of data integrity (LoI) attacks at the infrastructure-level and fraudulent claims at the application-level. LoI attacks can modify or destroy critical data in the system, compromising the veracity and efficacy of the claim processing. In the ClaimChain context, we consider the following notable LoI attacks:

1) Sybil Attack: Sybil attack [26] is an obvious threat to Blockchain given its peer-to-peer architecture. Attacker can undermine the consensus protocol by controlling a disproportional share of consenting nodes. This can be achieved by creating new peers or usurping existing ones. With a smaller share, attacker can influence network decisions. With a controlling share, attacker can effectively control the network. Attack at this scale is also known as 51% attack [27].

2) Injection Attack: Injection attacks leverage exposed input fields to insert malicious data into the backend. When properly executed, malicious data can pose a serious threat to data integrity of an insurance claim.

3) Fraudulent Claims: Fraud actions can arise from both the users side and the multi-domain entities side. A user can file duplicate claims or provide false information in the claim form for increased compensation. The multi-domain entities involved in the claim processing may perform insider-attack during the processing of claims, thus causing data corruption.

4) Malware: In a malware attack, attacker gains access to the backend and inserts malicious software into critical cloud resources. Malware can result in modification of system parameters toward non system-critical functions such as crypto-mining services or data exfiltration.

5) Timestamp Manipulation: Although Blockchain is resistant to data manipulation, attackers can stall efforts to identify fraudulent claims by modifying transaction timestamps. This can occur when an attacker logs into ClaimChain system and gains access to Hyperledger Composer files. With this access, attacker can edit the timestamp of a fraudulent transaction to a time in the past such that the application cannot identify that block. By hiding claim transactions, these attacks compromise the integral trait of data transparency in the Blockchain.

IV. Securing ClaimChain against LoI Issues

In this section, we describe the infrastructure and application level security schemes used in ClaimChain through a threat model as well as a fraud model, respectively.

A. Infrastructure-level Threat Modeling

We evaluate infrastructure-level vulnerabilities of the ClaimChain system using the attack tree formalism [28]. By creation of an attack tree for ClaimChain, we categorize LoI attacks into several categories to understand various potential threat scenarios. In addition, we use the attack tree to quantitatively evaluate the attack probabilities using the UPPAAL Statistical model checking (SMC) tool [9]. Based on this analysis, we recommend security design principles to reduce the impact of the threats.

1) Threat Modeling using an Attack Tree: We categorize different attack scenarios (explained in Section III-D) in the attack tree we created in Fig. 4 based on their risk to cause loss of ClaimChain data integrity in terms of tree leaves at the top-level: system compromise, data modification and application compromise. We use attack tree formalism versus using traditional threat models such as STRIDE [29] because it provides the ability to perform a quantitative analysis of cause-and-effect relationships pertaining to the threats, and also because of its popularity for reliability engineering in numerous industry domains. As described in [7], attack trees are hierarchical models that show the attacker goals that can be further divided into smaller nodes connected through gates such as AND, OR, and SAND (Sequential AND). The gates representation in an attack tree can be understood as follows: (a) AND gate: It is activated when all of the child nodes are activated; (b) OR gate: It is activated when at least one child node is activated; and (c) SAND gate: It is activated as the child nodes are activated from left to right based on the success of previous stage and later determines the activation of the next child node.

Under the top-level tree leaves of threat scenarios, we list the relevant types of attacks (also known as 'basic attack steps') as leaf nodes in the first lower level, and the causal steps in the second lower level. We assume that all the basic attack steps have a duration that is exponentially distributed, and we represent this through an equation given by -

$$P(t) = 1 - e^{\lambda t} \tag{1}$$

The λ is the rate of the exponential distribution. The λ values are chosen based on the concept of weighted probabilities used in prior works such as [30] (see Section V-B1 for details).

2) Quantitative Analysis of Attack trees: To evaluate the probability of LoI threats that can occur in the ClaimChain system, we use the UPPAAL SMC tool [9]. Following the methodology in [7], we calculate the attack probabilities using the following steps: Initially we analyze different threats in

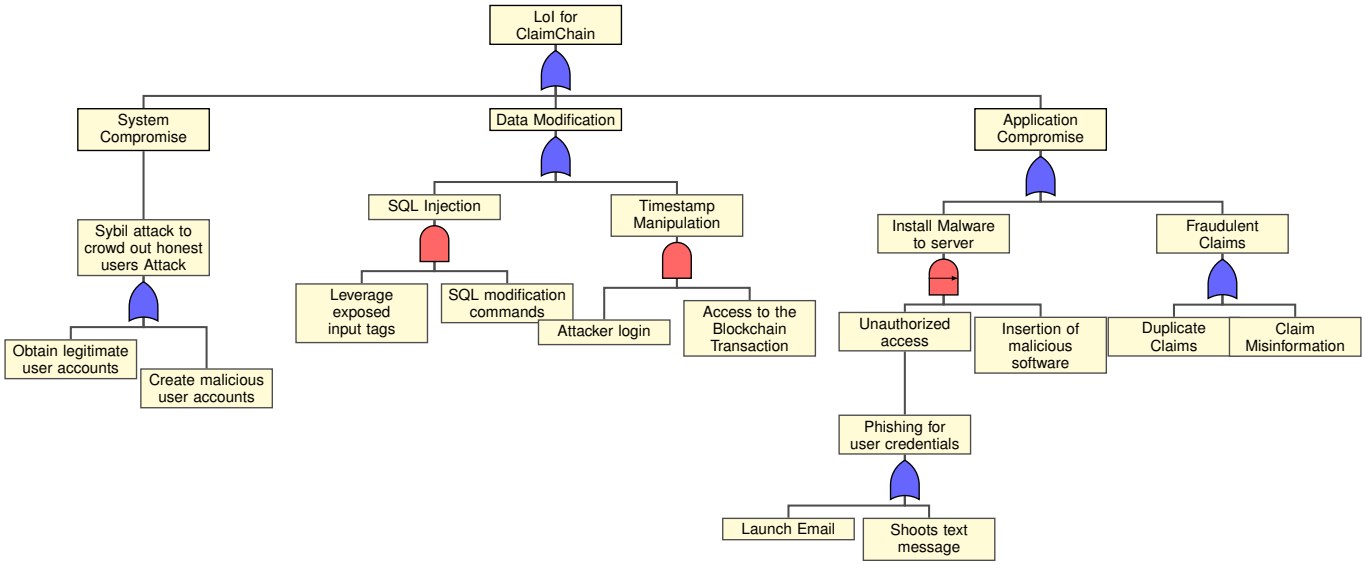

Fig. 4: Security Attack tree for Loss of Integrity issues in ClaimChain.

the attack tree (shown in Fig. 4) by converting them into their equivalent stochastic timed automata (STA) representations [31]. The converted STA is formed into a network of stochastic timed automata using the parallel composition [32] technique and is consequently provided as an input to the UPPAAL SMC tool.

The UPPAAL SMC tool helps in determining the likelihood of occurrence (P_r) for different attack scenarios. For each of the attacks involved, we generate different probabilities relevant to our ClaimChain system. For our ClaimChain system, we assign λ (rate of exponential) values for STAs at leaf nodes in the LoI attack tree, and the likelihood of occurrence is estimated based on the λ values. The likelihood of occurrence value corresponding to individual leaf nodes is propagated upward in the attack tree to determine the overall likelihood of LoI for ClaimChain (top most node).

3) Recommended Security Design Principles: As the various attacks occurrence is disruptive to the ClaimChain system, we next discuss how we consider security design principles that could strengthen the vulnerable components of the system and reduce the attack impact. Based on the guidelines in NIST SP800-160 document [10] [11] and empirical evaluation (as detailed in Section V-B2), we employ three security design principles to address the ClaimChain LoI issues: (i) *Hardening* – helps in strengthening certain components in order to make them more difficult to compromise or damage, (ii) *Sufficient Documentation* – documentation and other information should be supplied to users who have a responsibility to interact with the system in a way that contributes to system security, and (iii) *Principle of Privilege Attenuation (POPA)* – prevents unprivileged users from cooperating with one another to acquire access.

B. Application-level Fraud Modeling

We evaluate fraud detection at the application-level in data hosted via ClaimChain using various red flags identified by NICB [33]. Using machine learning, we accurately identify patterns of red flags within ClaimChain data and feed our fraud model with the identified patterns with various severity levels of red flags to calculate risk scores for a given claim asset. The risk scores can be used by Special Investigation Units, Law enforcers to pursue corrective measures (e.g., fraud mitigation efforts, prosecution of insurance fraud) in questionable claims.

1) Detecting Red Flags in Data using Machine Learning: There are over 200 NICB-identified red flag conditions that we consider in our fraud model for ClaimChain. Examples of salient red flags that are commonly used by Special Investigation Units, Law enforcers include: (i) If a claimant takes long time to report an accident, (ii) If incident happens within 10 days of holding the policy, (iii) If someone reports claim few days after a holiday i.e., Christmas, thanksgiving, (iv) If no police report is filed, and (v) If there is no witness.

Our fraud model uses machine learning to learn from any given ClaimChain data, and identifies patterns to help make decisions with minimal manual intervention. We use machine learning to avoid the risk of flagging legitimate claims and rejecting them. For instance, when a claim is registered, we can not directly reject it in the event we find a red flag condition in it. Red flag conditions may exist in legitimate claims due to genuine reasons such as unintentional misentry of data by policy holder or lack of necessary information (e.g., police report for minor accidents). Use of machine learning models such as KNN, RCF, LR and XGBoost helps us to identify anomaly patterns in the user claim data considering a holistic analysis based on NICB-identified red flags in ClaimChain (see details in Section V-C).

2) Risk Scoring for Pursuing Corrective Measures: Fig. 5 shows how we take the output of machine learning models featuring detected red flags with various severity levels to assess the risk of a new claim being fraudulent. We devise a risk scoring scheme that compares the new claim‘s entities with an estimate of the mean of existing claims based on red flag conditions. We calculate mean scores for different conditions such as e.g., frequency, reporting delays by organizing the data that is contained in the peer database

for analysis with an analytical engine called Tableau. We assign different risk score weights for the red flags (e.g., w = 50 for the case - 'If a claimant takes long time to report an accident'); we remark that the actual weights can be customized by a Special Investigation Unit team depending on their business preferences in terms of a tolerable number/scale of investigations for a given set of claims.

We categorize the risk scores in ClaimChain depending on their severity as High (H), Medium (M) or Low (L). Based on the claim risk score range as shown in Equation 2, a claim decision is made through either auto-approval or manual approval (with additional evidence verification) or sent for fraud investigation. We use a risk score range from 0-900 following a similar range used in the insurance industry.

Thus, the risk range for a claim and corresponding followup is given as follows:

$$R_s = \begin{cases} 0-600, \ (C_r = L); \ auto-approved \ claim \\ 600-700, \ (C_r = M); \ manually \ approved \\ 700-900, \ (C_r = H); \ needs \ investigation \end{cases} \quad (2)$$

After the calculation of the risk, the claim object along with its risk score is sent into the Hyperledger Composer through API calls. The chaincode helps in the decision making process of a particular claim. When a claim has a certain risk level, the claim could be rejected or assigned for further review/investigation through the chaincode and sent to the attention of a claim adjudicator or a fraud analyst.

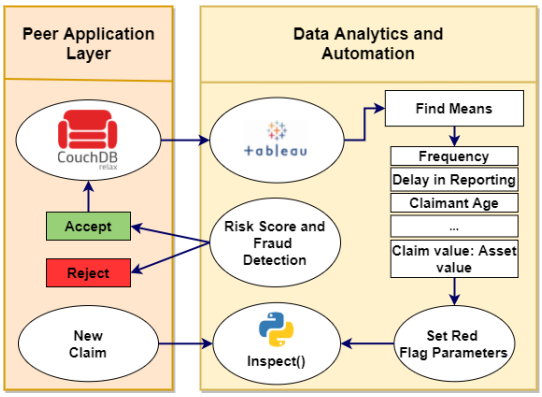

Fig. 5: Fraud detection by risk scoring of a new claim's entities based on severity of red flag conditions.

V. PERFORMANCE EVALUATION

In this section, we first evaluate ClaimChain platform quantitatively using a scalability experiment, and then qualitatively compare it with existing Blockchain-based solutions for different insurance industry applications. Subsequently, we validate our threat model for infrastructure-level security as well as our fraud model for application-level security in ClaimChain.

A. Evaluation of ClaimChain as a Blockchain Solution

1) Quantitative Evaluation: The goal of our quantitative evaluation of ClaimChain is to show that our system implementation is scalable in terms of the transactions' processing rate. For the purpose of conducting this scalability experiment, we setup a ClaimChain testbed on a public cloud infrastructure i.e., Amazon Web Services (AWS). The testbed features our Hyperledger Composer Blockchain network hosted on an AWS EC2 t2.micro instance with 16 GB storage. In the experiment, we stress-test the three smart contracts: *Issue Claim*, *Approve Claim* and *Cancel Policy*. We extensively simulate each contract to account for varying processing times in their execution. For a fair comparison with a state-of-the-art system, we compare our ClaimChain system with the "CioSy" [8] system. The CioSy system utilizes Ethereum and has similar mechanism and functions defined through smart contract methods. We use the same number of insurance policies for our experiments with ClaimChain and CioSy, and create a total of 1000 distinct insured accounts (the same 1000 number of accounts was used in [8] evaluations) and follow the general use case outlined in 3 and end by canceling each of the policies.

Fig. 6 shows the scalability results in terms of the time taken to invoke the ClaimChain chaincode. As the number of policies increases, the processing time increases from 0 to 240 seconds. We can see that the total time necessary for chaincode interactions in ClaimChain is directly proportional to the number of insurance policies, and that the number of insured policies vastly influences the total time required. Fig. 7 shows the results for the processing time for the ClaimChain (uses a permissioned Blockchain platform) and CioSy (uses a permissionless Blockchain platform) systems. We can see that the ClaimChain system processing time is comparable if not slightly better than that of CioSy system for 'Claim Creation', 'Claim Decision' and 'Cancel Policy' cases. Thus, we conclude that our ClaimChain meets the insurance claims processing objectives in terms of processing speed, while also reducing administrative and operational costs by minimizing manual interactions, and by recording the insurance transactions in a tamper-proof manner.

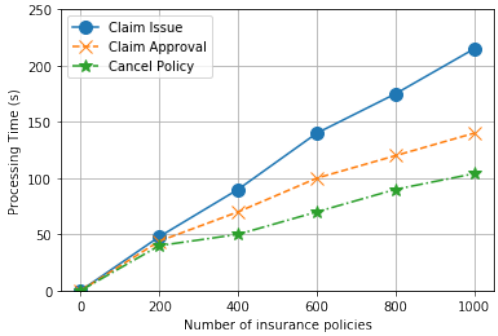

Fig. 6: ClaimChain scalability when processing different operations of issuance, approval and cancellation with different number of insurance policies.

2) Qualitative Evaluation: Table II distinguishes ClaimChain from other existing Blockchain-based solutions for different insurance industry applications. Inspeer [34] is a Blockchain-based insurance company that focuses on process transparency. On the other hand, Friendsurance [35] focuses on offering online contract management with their fully featured digital bancassurance platform. Etherisc's [36] generic insurance framework includes core application specific smart contracts and microservices based on which, users can invest

TABLE II: Comparison of ClaimChain with existing Blockchain-based solutions for different insurance industry applications.

Blockchain Solution	Basic Methods	Exemplar Tools	Scalability	Adding Peers	Fraud Detection	Other Advantages
Inspeer [34]	Proprietary	Robo advisor	Medium	Public	No	Ability to increase and insure deductible
Friendsurance [35]	Digital Brokerage	Bancassurance	High	Private	No	Offers rewards for staying claims-free
Etherisc [36]	Application-specific smart contracts	Risk pool keeper, Relayer	High	Public	No	Helps earn interest in cryptocurrency
B-FICA [37]	Dynamic block	Protocol validator	Medium	Private	No	Builds resilience to Sybil attack
WISChain [16]	Smart contract	DengLu	Low	Private	No	Rewards Insurers for data packing
ClaimChain (This work)	Smart contract	Tableau, CouchDB	High	Private	Yes	Analyzes and mitigates LOI attack impact

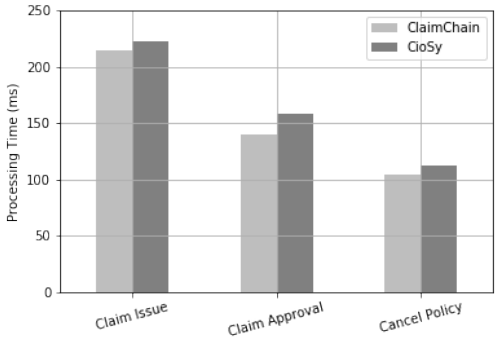

Fig. 7: Comparison of processing times between ClaimChain system and CioSy system.

and earn interest from a tokenized risk pool. B-FICA [37] targets detection and prevention of Sybil attacks. WISChain [16] provides insurance for websites that incentivize claimants for data-packing to maintain the system lifecycle.

In comparison to prior related works, the intelligent automation and security layers at infrastructure and application levels in ClaimChain make it more highly scalable in terms of performance, and relatively more resilient to LoI attacks and fraudulent claims. Specifically, ClaimChain borrows best practices of threat modeling based on attack trees, and fraud modeling based on NICB-identified red flags to protect against LOI attacks. None of the existing Blockchain-based solutions provide such a set of capabilities. Consequently, an insurer using existing solutions needs to manually check the validity of each claim, which delays the claim approval process, and subjects the process to human error. In addition, ClaimChain is open-source, whereas both Inspeer and Friendsurance use proprietary methods and tools.

B. Evaluation of Infrastructure-level Threat Model

As discussed in Section III-D, we assign λ values for the calculation of probability of different attacks. Our λ value assignment considers the fact that multiple attack scenarios can occur concurrently in real-world systems. Consequently, we assigned a λ value to a specific leaf node in the attack tree and utilized a small positive constant (K) of $\approx$0.0002 for all the remaining nodes for calculating the likelihood of a particular LoI attack scenario.

TABLE III: λ values for different LoI threat scenarios.

Type	Threat Events	λ
S1	Maliciously obtain legitimate user accounts	0.01
S2	Create malicious user accounts	0.03
S3	Exploit exposed input tags	0.03
S4	Unauthorized SQL modification commands	0.01
S5	Unauthorized attacker login	0.03
S6	Unauthorized access to the Blockchain transaction	0.007
S7	Malicious software injection	0.03
S8	Duplicate claims to receive double compensation	0.04
S9	Claim misinformation to increase compensation	0.05
S10	Phishing through Email to obtain sensitive information	0.03
S11	Phishing through Instant message to obtain sensitive information	0.02

1) Probability Analysis of LoI Attack Tree: As shown in Table III, we allocate constant λ values based on [30] using the concept of weighted probabilities. Subsequently, to evaluate the attack vulnerability, we examine each leaf node individually to determine the likelihood of LoI attacks.

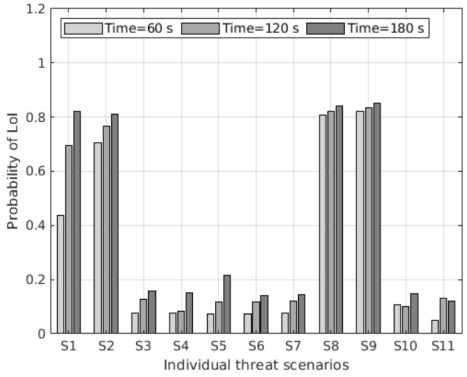

Fig. 8: Probability of LoI in different threat scenarios shown in Table III.

Fig. 8 shows the Likelihood of LoI for each leaf node of the attack tree for the threat scenarios shown in Table III at various time intervals $\in$ [60s, 120s, 180s]. We can observe

that the leaf nodes $S1$ (Obtain legitimate user accounts), $S2$ (Create malicious user accounts), $S8$ (Duplicate Claims) and $S9$ (Claim Misinformation) are the most dominant attacks that are likely to disrupt the ClaimChain system.

2) Attack Probability Reduction: To achieve attack probability reduction in ClaimChain, we employ security design principles as discussed in Section IV-A3. Specifically, we apply the security design principles on the earlier identified vulnerable nodes (i.e., Sybil attack, unauthorized access and claim misinformation) of the LoI attack tree. We incorporate the *hardening* design principle on the unauthorized access leaf nodes by adding an extra node in the attack i.e., a firewall to prevent access to the ClaimChain system. Note, as per the POPA principle, the original account of the attacker and the accounts they create lack the privilege to access the target account i.e., they need the approval of other users to obtain access. We apply the POPA principle on one of the vulnerable nodes i.e., Sybil attack by adding a new node approval of access. Similarly, we add a security design principle on the claim misinformation node by adding an extra node on the attack tree. Subsequently, we re-evaluate the modified attack trees after incorporating the three design principles using the UPPAAL tool. With the stated attacker profile, the obtained results are shown in Fig. 9. We observe that the probability of an LoI disruption is lowered from 0.85 to 0.646 (24% reduction) through the use of a mixture of recommended security design principles in our ClaimChain system.

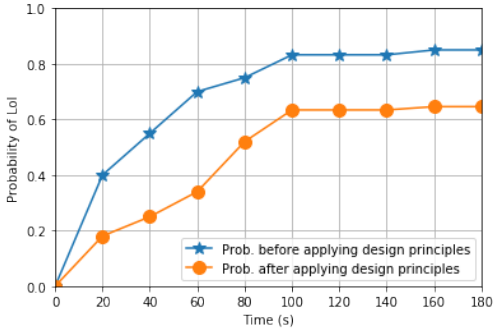

Fig. 9: Probability after application of design principles to mitigate impact of LoI in ClaimChain.

C. Evaluation of Application-level Fraud Model

To evaluate our fraud model, we host a publicly available dataset [12] on the ClaimChain testbed and perform machine learning model experiments. The dataset describes insurance vehicle incident claims for an undisclosed insurance company. It contains 15,530 claims, out of which 924 claims are labeled as fraudulent (ground truth), and each claim comprises of 31 attributes describing the following components: (a) Customer demographic details (Age, Sex, Marital Status), (b) Purchased policy (Policy Type, Vehicle Category, No: of supplements, Agent Type), (c) Claim circumstances (day/month/week claimed, policy report filed, witness present, past days between incident-policy report and incident-claim, and (d) Other customer data (number of cars, previous claims, Driver Rating).

TABLE IV: Machine learning models accuracy in fraudulent claims analysis.

S.No	Model	Accuracy	Precision	Recall	F-Score
i	KNN	96%	0.96	0.96	0.96
ii	RCF	82%	0.96	0.67	0.78
iii	LR	76%	0.70	0.74	0.72
iv	XGBoost	98%	0.98	0.98	0.98

For detecting fraudulent activities based on the NICB-identified red flags, we use four machine learning models: Random Cut Forest, K-Nearest Neighbor, Logistic Regression and XGBoost. The choice of our machine learning models is influenced by their unique significance and their performance in identifying patterns in comparison with other machine learning models such as e.g., Support Vector Machine, Decision Tree. We used 80% of the dataset for training and 20% for testing. Table IV shows the machine learning models accuracy results along with precision, recall and F-score values for the test data set. We conclude that the XGBoost is ideal for use in ClaimChain because it has the highest accuracy of 98% in detecting fraudulent activities when compared to all other machine learning models.

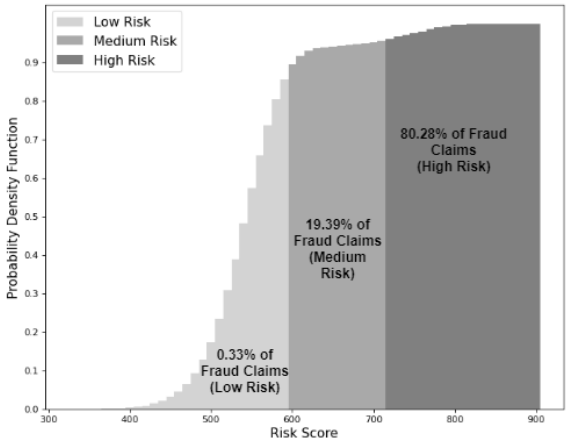

Fig. 10: Fraudulent claims PDF based on risk scores (High, Medium, Low) obtained from the open dataset analysis.

Fig. 10 shows the fraudulent claims probability density function (PDF) based on risk scores (High, Medium, Low) considering a representative sample of 15,530 claims in the open dataset. We observe that 80.28% of the claims fall in the high-risk score area in terms of fraudulence, 19.39% of the claims fall in the medium-risk area, and 0.33% of the claims fall in the low-risk area.

In an additional set of experiments, we perform data analysis on the open dataset of claims hosted on the ClaimChain testbed. In this case, we borrowed the idea of calculating Performance Metrics & Statistical Significance from the work in [38]. Thereby, we are able to understand the various data features and their inter-relation. At a descriptive level, we first summarize a macro-profile for 924 fraud cases in the dataset. We draw the following few key conclusions from the analysis of the dataset: (i) 88.6% of the fraudsters were male, (ii) 67.2% were married, (iii) average age was 38.2 years, (iv) 51.7% have rating greater than 2 i.e., 3, 4, (v) 98.2% do not have police

reports, and (vi) 99% do not have a witness. Additionally, we conclude that most of the fraudulent claims identified have no police report and nor witnesses.

VI. Conclusion and Future Work

In this paper, we proposed an intelligent insurance claim processing system viz., "ClaimChain" that is built upon a consortium Blockchain platform in order to replace the traditional NICB/ISO database architecture used in the auto-insurance industry. Our ClaimChain featured novel schemes to: (i) improve security at the *infrastructure-level* through threat modeling via the use of attack trees, and (ii) improve security at the *application-level* through fraud modeling using machine learning and NICB-identified red flags. Our evaluation results showed that our ClaimChain solution is clearly a futuristic alternative to the current NICB/ISO database architecture, and can help in achieving greater participation, processing efficiency, and trust amongst the insurance provider organizations. In addition, we showed that ClaimChain can be equipped with a mix of security design principles that are effective in protecting insurance claims processing as seen from the results that show a reduction of the probability of LoI by up to 24% before and after application of the security design principles. Lastly, we showed that our fraud detection approach featuring the XGBoost machine learning model in ClaimChain is effective in detecting NICB-identified red flags with an accuracy of 98%.

As part of future work, one can expand ClaimChain to be more resilient to Sybil attacks by developing better detection mechanisms. In addition, multi-domain entities (such as e.g., police and third-party insurance administrators) can be involved in the insurance claims processing towards building an industry-wide claims processing solution to more effectively conduct fraud analytics at large-scale.

References

[1] T. Catlin, J.-T. Lorenz, J. Nandan, S. Sharma, and A. Waschto, "Insurance beyond digital: The rise of ecosystems and platforms," *McKinsey & Company*, 2018.

[2] "National Insurance Crime Bureau (NICB)," *https://www.nicb.org/*.

[3] "Insurance Services Office (ISO)," *https://www.verisk.com/insurance/brands/iso/*.

[4] A. Haskin, "Federal Insurance Office U.S. Department of the Treasury." *https://home.treasury.gov/system/files/311/FACI_NICB_Insurance_Fraud.pdf*, Nov 2016.

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

[6] K. S. Braunwarth, M. Kaiser, and A.-L. Muller, "Economic evaluation and optimization of the degree of automation in insurance processes," *Business & Information Systems Engg*, vol. 2, no. 1, pp. 29–39, 2010.

[7] S. Valluripally, A. Gulhane, R. Mitra, K. A. Hoque, and P. Calyam, "Attack trees for security and privacy in social virtual reality learning environments," in *17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2020, pp. 1–9.

[8] F. Loukil, K. Boukadi, R. Hussain, and M. Abed, "CioSy: A collaborative blockchain-based insurance system," *Electronics*, vol. 10, no. 11, p. 1343, 2021.

[9] A. David, K. G. Larsen, A. Legay, M. Mikučionis, and D. B. Poulsen, "UPPAAL SMC tutorial," *International journal on software tools for technology transfer*, vol. 17, no. 4, pp. 397–415, 2015.

[10] R. Ross, M. McEvilley, and J. Oren, "Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems," 2018-01-03 2018.

[11] M. Bishop, "Computer security: Art and science," *Addison-Wesley; https://books.google.com/books?id=pfdBiJNfWdMC*, 2003.

[12] "Fraud Detection In Insurance Claims (Dataset)," *https://kaggle.com/roshansharma/fraud-detection-in-insurance-claims*.

[13] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: Is the technology mature enough?" *Future internet*, vol. 10, no. 2, pp. 20–, 2018.

[14] J. Gera, A. R. Palakayala, V. K. K. Rejeti, and T. Anusha, "Blockchain technology for fraudulent practices in insurance claim process," in *5th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2020, pp. 1068–1075.

[15] R. Brophy, "Blockchain and insurance: a review for operations and regulation," *Journal of financial regulation and compliance*, vol. 28, no. 2, pp. 215–234, 2019.

[16] Y. Guo, Z. Qi, X. Xian, H. Wu, Z. Yang, J. Zhang, and L. Wenyin, "Wischain: An online insurance system based on blockchain and denglu1 for web identity security," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, 2018, pp. 242–243.

[17] J. Moubarak, E. Filiol, and M. Chamoun, "On blockchain security and relevant attacks," in *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, 2018, pp. 1–6.

[18] M. Huth, C. Vishik, and R. Masucci, "Risk engineering and blockchain: Anticipating and mitigating risks," in *International Conference on Business Information Systems*. Springer, 2018, pp. 381–392.

[19] Y. Ru, Y. Wang, J. Li, J. Liu, G. Yang, K. Yuan, and K. Liu, "Risk assessment of cyber attacks in ecps based on attack tree and ahp," in *12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 2016, pp. 465–470.

[20] X. Liu, J.-B. Yang, D.-L. Xu, K. Derrick, C. Stubbs, and M. Stockdale, "Automobile insurance fraud detection using the evidential reasoning approach and data-driven inferential modelling," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2020, pp. 1–7.

[21] M. Taher, H. Guermah, and M. Nassar, "Mcdm method for financial fraud detection: A review," in *Proceedings of the 4th International Conference on Big Data and Internet of Things*, 2019, pp. 1–8.

[22] R. Bhowmik, "Detecting auto insurance fraud by data mining techniques," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 2, no. 4, pp. 156–162, 2011.

[23] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement," *IEEE Access*, vol. 8, pp. 58 546–58 558, 2020.

[24] ——, "Extreme gradient boosting machine learning algorithm for safe auto insurance operations," in *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2019, pp. 1–5.

[25] L. Thomas, "Auto insurance statistics and facts," *https://www.bankrate.com/insurance/car/auto-insurance-statistics/*.

[26] P. Swathi, C. Modi, and D. Patel, "Preventing sybil attack in blockchain using distributed behavior monitoring of miners," in *ICCCNT*. IEEE, 2019, pp. 1–6.

[27] "51% Attack," *https://www.investopedia.com/terms/1/51-attack.asp*.

[28] V. Saini, Q. Duan, and V. Paruchuri, "Threat modeling using attack trees," *Journal of Computing Sciences in Colleges*, vol. 23, no. 4, pp. 124–131, 2008.

[29] A. Marback, H. Do, K. He, S. Kondamarri, and D. Xu, "A threat model-based approach to security testing," *Software: Practice and Experience*, vol. 43, no. 2, pp. 241–258, 2013.

[30] P. Saripalli and B. Walters, "Quirc: A quantitative impact and risk assessment framework for cloud security," in *3rd international conference on cloud computing*. IEEE, 2010, pp. 280–288.

[31] N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Groesser, and M. Jurdzinski, "Stochastic timed automata," *Logical methods in computer science*, vol. 10, no. 4, 2014.

[32] P. Ballarini, N. Bertrand, A. Horváth, M. Paolieri, and E. Vicario, "Transient analysis of networks of stochastic timed automata using stochastic state classes," in *International Conference on Quantitative Evaluation of Systems*. Springer, 2013, pp. 355–371.

[33] "NICB Red flags," *https://tinyurl.com/3685zxsc*.

[34] H. Terry, "insPeer," *https://www.the-digital-insurer.com/dia/inspeer-1st-peer-to-peer-insurance-service-in-france/*.

[35] "Friendsurance - Pioneer in Digital Insurance," *https://www.friendsurance.com/*, 2019.

[36] "Etherisc," *https://etherisc.com/*, 2021.

[37] C. Oham, R. Jurdak, S. S. Kanhere, A. Dorri, and S. Jha, "B-fica: Blockchain based framework for auto-insurance claim and adjudication," in *iThings*. IEEE, 2018, pp. 1171–1180.

[38] M. K. Severino and Y. Peng, "Machine learning algorithms for fraud prediction in property insurance: Empirical evidence using real-world microdata," *Machine Learning with Applications*, p. 100074, 2021.