PINSA: Privacy-Preserving Cyber Insurance Framework

Georgios Paparis[1], Apostolis Zarras[1,2], Aristeidis Farao[1,3], Christos Xenakis[1]

[1]University of Piraeus, Greece.
[2]Foundation for Research and Technology – Hellas, Greece.
[3]InQbit Innovations SRL, Romania.

Abstract

Cybercriminals continually advance their tactics, exploit novel attack vectors, and target emerging technologies. To mitigate these risks, cyber insurance policies must remain abreast of the latest technological developments. Staying technologically updated enables an Insurance Company (IC) to assess risks more precisely, tailor policies to a potential Policyholder (PH), and accurately calculate fair premiums. By incorporating innovative risk assessment methodologies, robust Know-Your-Customer protocols, and automated claims-handling processes, ICs can offer tailored and cost-effective solutions to their PHs. Embracing technological advancements enables the field of cyber insurance to adapt to the ever-changing landscape of cyber threats, providing comprehensive protection to organizations. This article introduces `PINSA`, an innovative privacy-preserving framework designed to deliver robust security and privacy assurances to PHs against honest but inquisitive entities within the cyber insurance ecosystem. `PINSA` also equips ICs with automated processes for claims management. At the core of `PINSA` lies Hyperledger Aries, leveraging verifiable credentials to empower PHs with identity ownership and data control. Our framework is complemented by Hyperledger Fabric, which imbues `PINSA` with intelligent functionalities enabling PHs and ICs to autonomously execute actions related to cyber insurance and gather historical cybersecurity data. In this direction, we have successfully implemented the key components of `PINSA` and conducted a quantitative performance assessment. We also substantiate its security and privacy attributes, confirming that `PINSA` effectively achieves its objectives. In summary, `PINSA` represents a forward-thinking solution poised to enhance cyber insurance in an era of ever-evolving cyber threats, offering a promising avenue for safeguarding organizations and policyholders in the digital landscape.

Keywords: Cyber Insurance, Verifiable Credentials, Self-Sovereign Identity, Smart Contracts

1 Introduction

In 2021, the cyber insurance sector generated a substantial $10 billion in premiums. Notably, the Swiss Re Institute projects a remarkable annual growth rate of 20%, foreseeing the global market reaching a staggering $23 billion in underwriting premiums by 2025 [1]. This rapid expansion of the cyber insurance industry can be attributed to various factors. Foremost, cyberattacks have gained widespread recognition as a significant threat to the global digital economy, characterized by an escalating frequency and severity. This mounting concern is underscored in the Global Risks Report of 2023 [2], as well as in subsequent reports for 2024 [3] and 2025 [4] from the World Economic Forum, where cybercrime and cyber insecurity rank as one of the most severe global risks.This ranking highlights the persistent and substantial nature of cyber risks, signifying their lasting significance

over the coming decade. High-profile incidents, such as SolarWinds [5], Microsoft Exchange [6], and Colonial Pipeline [7], have served as poignant reminders for both public and private entities, prompting a collective awareness of the pressing need for cybersecurity. In response, organizations are actively seeking ways to fortify themselves against cyber risks, fueling a burgeoning demand for comprehensive cyber insurance coverage.

It is imperative for Policyholders (PHs) to thoroughly evaluate their cyber insurance policies to fully understand the scope of coverage and any limitations [8]. Likewise, Insurance Companys (ICs) often conduct detailed investigations of claims to verify compliance with policy terms and to assess whether the PH implemented adequate cybersecurity measures. Disputes between PHs and ICs frequently arise, particularly in the context of loss reimbursement following cybersecurity incidents. Such conflicts may stem from divergent interpretations of policy language, contested exclusions, or disagreements concerning the cause and extent of the damage.

Notably, the NotPetya ransomware attack, which caused widespread disruption to organizations globally, led to several high-profile insurance disputes. Affected companies, such as shipping giant Maersk [9] and pharmaceutical manufacturer Merck, encountered difficulties in securing reimbursements, as some ICs classified the attack as an act of war, thus invoking war-related exclusions. Similarly, Sony Pictures experienced a significant breach involving the theft and public release of sensitive data, internal communications, and unreleased films [10]. In this case, insurers initially denied coverage because the incident qualified as an act of war or terrorism, leading to protracted legal proceedings.

Another illustrative case involves Cottage Health System, a healthcare provider that suffered a data breach compromising thousands of patient records. When the organization sought coverage under its cyber insurance policy, the insurer, Columbia Casualty, denied the claim, citing an exclusion related to third-party errors and omissions. This dispute ultimately proceeded to litigation [11]. Additionally, the global aluminum producer Norsk Hydro experienced a major ransomware attack that severely disrupted operations [12]. Despite having cyber insurance, the company faced challenges in obtaining full reimbursement, as its insurer contested the valuation of the losses, resulting in prolonged negotiations and potential coverage gaps [13].

Collectively, these examples underscore the complex and often contentious nature of cyber insurance claims. Reimbursement disputes frequently arise due to ambiguous policy wording, stringent security requirements, and conflicting assessments of causality and damage, highlighting the critical need for clarity and due diligence in cyber insurance agreements.

In this article, we provide a detailed analysis of the existing cyber insurance system, highlighting its challenges and processes, together with its basic stakeholders and ICs' information needed by ICs to establish a cyber insurance contract. Moreover, we define for the first time a security model that describes how the stakeholders can compromise the existing system and provide security and privacy requirements that a scheme dedicated to cyber insurance shall meet to prevent misuse actions by them. In addition, we design and develop an innovative framework, entitled `PINSA`[1], which is built on top of INCHAIN [14], an existing innovative cyber insurance architecture. Its goal is to face the existing challenges and ensure security, fairness, trust, and interoperability among all participating entities. To evaluate `PINSA` effectiveness, on the one hand, we measured `PINSA`'s Hyperledger Fabric performance, assessing our in-house functions' latency, throughput, and CPU utilization for 2, 4, 6, 8, and 10 nodes for a batch size equal to 2000, as well as the `PINSA`'s smart contract execution time. On the other hand, we evaluated `PINSA`'s Hyperledger Aries implementation, assessing the average time and frequency of Verifiable Credential (VC) issuance, presentation (with and without attribute satisfaction), as well as the VC's issuance latency. In addition, we have evaluated `PINSA`'s security, privacy, and legal capabilities, and through empirical experiments, demonstrated that the `PINSA`'s SSI implementation is capable of unlinkability and selective disclosure. Finally, we have performed a qualitative comparison between `PINSA` and other works that focus solely on the cyber insurance sector, while we conduct a quantitative comparison

[1]The source code of `PINSA` can be found at https://github.com/UniPiSSL/PINSA.

between PINSA and another scientific work that employs similar approaches in the cyber insurance field.

Our work is based on established technologies that have been combined to create a unique framework. The core component of our proposed framework is Blockchain technology, which offers data transparency and traceability. We have chosen the Hyperledger Fabric [15] implementation for that purpose. It provides a fertile ground for applications that can contribute to the advancement of cyber insurance. We have developed two key applications within our framework: Smart Contracts and Self-Sovereign Identity (SSI). Smart Contracts provided by Hyperledger Fabric play a vital role in PINSA by enabling automated tasks and establishing requirements that govern the involved stakeholders. These ensure that obligations and agreements are enforced automatically, reducing the need for manual intervention and streamlining processes. For the SSI, we have chosen the Hyperledger Aries implementation [16] that focuses on data minimization, enabling robust identification, facilitating data interoperability and portability, ensuring data control, promoting decentralization, and providing transaction transparency.

At this point, we would like to note that this work is structured upon the following three hypotheses: **[H1]** – A cyber insurance ecosystem can be built and effectively work on top of blockchain-based applications. **[H2]** – A cyber insurance ecosystem built on top of blockchain-based applications is robust against common, well-known cybersecurity attacks affecting common and blockchain-based systems. **[H3]** – A cyber insurance ecosystem built on top of blockchain-based applications can assist stakeholders in complying with well-known standards, directives, and guidelines.

Overall, our proposed framework offers a practical solution to the existing limitations and complexities within the cyber insurance industry. By harnessing the benefits of the used technologies, we demonstrate how PINSA can effectively address the challenges faced by the industry and provide a more efficient and transparent cyber insurance ecosystem. Moreover, we provide an in-depth practical performance evaluation assessing the effectiveness, efficiency, and robustness of the PINSA, complementing it with a security and privacy appraisal.

In summary, we make the following main contributions:

- We define the security and privacy requirements of the cyber insurance ecosystem deployed on top of Blockchain.
- We formalize a model of possible threats in the cyber insurance ecosystem on top of Blockchain
- We propose PINSA, a privacy-preserving framework designed for the cyber insurance ecosystem that works on top of a Blockchain based on well-established security and privacy-preserving technologies.
- We quantitatively assess PINSA's performance and qualitatively reason about its security and privacy properties. For this purpose, we have implemented the main components of PINSA, including the Hyperledger Aries verifiable credential system and Hyperledger Fabric Smart Contracts.

The remainder of this article is organized as follows. Section 2 analyzes related works from the literature that lies in the fields of cyber insurance and crowdfunding. Next, Section 3 presents the characteristics of the existing cyber insurance ecosystem, including its challenges, operations, and security and privacy requirements, accompanied by a security model. Section 4 presents the architectural overview of the proposed framework. Also, Section 5 elaborates on PINSA operations, describing all the required steps in detail. Section 6 includes the quantitative performance evaluation of our framework together with qualitative security and privacy appraisal of PINSA, while Section 7 describes the answers to research questions in detail, the article's limitations, and proposes directions for future work. Finally, Section 8 concludes this article.

2 Related Work

This section provides an in-depth examination of prior works introducing architectures tailored explicitly for the cyber insurance ecosystem.

In [17], SACI was proposed, a system that leverages Blockchain to establish a reliable and automated interaction between PHs and ICs in the cyber insurance ecosystem. This framework

employs Smart Contracts to address diverse requirements in the cyber insurance process, encompassing premium payments, contract updates, handling of damage coverage requests, and dispute resolutions. The authors conduct a proof-of-concept evaluation, particularly emphasizing dispute scenarios. SaCI is built on the Ethereum network, wherein each Smart Contract function incurs a gas fee, while PINSA is constructed on top of Hyperledger Fabric, eliminating gas fees for transactions. Although SaCI's Smart Contract is accessible to both the corresponding IC and PH, the limitations on this access are not explicitly outlined. In contrast, PINSA ensures that Smart Contract functions can only be executed by the designated IC and PH through the use of VCs. This approach guarantees that cybersecurity incidents can only be reported by legitimate victims who substantiate their ownership by presenting the correct VC. Further, SaCI predominantly focuses on actions occurring during the cyber insurance contract, such as coverage requests. On the contrary, PINSA extends its focus beyond contract-related actions, incorporating functionalities related to premium pricing, such as accurate data gathering. This broader scope enhances the capabilities of PINSA in addressing diverse aspects of the cyber insurance process.

Next, Lepoint et al. [18] propose BlockCIS, a system designed to establish a dynamic cyber insurance framework. This system gathers data about the PH's information technology and computer infrastructure, facilitating personalized risk assessment and identification of the attack surface. The collected data is stored and accessed by authorized third parties and auditors who conduct various analyses and take necessary actions. Constructed on the Hyperledger Fabric platform, BlockCIS eliminates transaction fees for executing each Smart Contract function. Both BlockCIS and PINSA are built on top of Hyperledger Fabric, sharing the advantage of no transaction fees. However, PINSA distinguishes itself by employing VCs to grant access to their rightful owners, enabling them to perform specific actions, including incident reporting and purchasing cyber insurance. This integration represents an Attribute-Based Access Control (ABAC) approach within PINSA. While Access Control Lists (ACLs) and ABAC control resource access, ACLs are generally more straightforward, relying on predefined rules tied to user identities. On the other hand, ABAC provides greater flexibility by considering a broad range of attributes in access decisions, making it well-suited for complex and dynamic access control requirements. PINSA's utilization of ABAC contributes to its adaptability and precision in managing access to critical resources within the cyber insurance ecosystem.

SECONDO [19] is a dedicated platform designed for the comprehensive assessment and efficient management of cyber risks. It adopts a quantitative approach that considers technical parameters and non-technical factors, such as user behavior, contributing to cyber exposure. The platform aims to bridge information asymmetry between the insured and the insurer while offering analyses for effective risk management. This is achieved by recommending optimal investments in cybersecurity controls [20]. Furthermore, SECONDO determines residual risks and calculates cyber insurance premiums based on the business strategy of the IC, thereby eliminating information asymmetry between the PH and the IC [8]. SECONDO also incorporates Blockchain technology to securely store data related to the effectiveness of implemented cybersecurity controls. It leverages Smart Contracts embedded within the distributed ledger to automate agreement processing, notify ICs and PHs when an agreement is binding, and handle premium and commission payments. While SECONDO proposes an architecture without incorporating any implementation, focusing on delivering optimal cybersecurity solutions and minimizing information asymmetry between involved PHs and ICs, PINSA is a comprehensive framework implementing various technologies. It is designed to address cyber insurance challenges, specifically executing transactions related to issuing cyber insurance contracts and handling cybersecurity incident reports from both PH and IC perspectives. PINSA provides a practical and operational solution to the challenges identified in the cyber insurance ecosystem.

Additionally, INCHAIN [14] presents an architectural framework built on Blockchain, leveraging Smart Contracts and SSI to address prevailing challenges in the cyber insurance landscape. It employs the SSI approach to gather accurate data from PH for premium pricing. The Smart Contract within INCHAIN encompasses functions

that enable both PHs and ICs to report cybersecurity incidents and manage them accordingly. However, it is essential to note that INCHAIN lacks a corresponding implementation; it proposes suitable Blockchain and SSI solutions without practical instantiation. In contrast, `PINSA` not only offers a comprehensive framework but also implements a range of technologies to tackle current challenges in the cyber insurance domain. The technologies utilized in `PINSA` align with those proposed by INCHAIN. Overall, it is noted that the proposed work is an extended version of the work entitled INCHAIN. In particular, `PINSA` surpasses its primitive version including: (*i*) detailed analysis of security and privacy requirements that a cyber insurance framework should meet to achieve its goals leaving untouched the cyber insurance features; (*ii*) a comprehensive threat model in the study architecture; (*iii*) a summary of different benchmarking tests assessing the performance and effectiveness of the `PINSA`, and (*iv*) a discussion related to the security features of `PINSA`.

Vakilinia et al. [21] introduce a cyber insurance crowdfunding framework built on Blockchain technology, specifically operating on the Ethereum network. This framework involves four primary participants: the Vendor, Customer, Auditor, and IC. The process commences with the vendor requesting insurance services for a cyber product. Interested ICs then engage in a sealed-bid auction, submitting their preferred premiums for providing the insurance service. The winners of the auction are subsequently selected to offer insurance coverage. In the event of an indemnity request, an auditor is tasked with verifying the request's validity. The authors implemented the proposed system on the Ethereum Blockchain, resulting in gas fees associated with each transaction. They developed a Smart Contract for crowdfunding initialization, bidding, wrapping, and reimbursement.

In addition, Xu et al. [22] made a significant contribution to research by presenting a Blockchain-based cyber insurance framework designed to enhance the time efficiency of crowd-sourcing tasks within Blockchain applications. The proposed framework prioritizes strengthening the robustness of crowd-sourcing systems and categorizes tasks into three types based on time sensitivity: time-sensitive, slightly time-sensitive, and time-insensitive. Notably, the framework automates processes related to reimbursement. Operating on the Ethereum network, the framework incurs gas fees for each Smart Contract function. The Smart Contract manages various actions, including bidding, cyber insurance creation, and reimbursement. It is worth highlighting that these works take a different approach than `PINSA`. While the former present architectures and frameworks for the cyber insurance ecosystem, the latter focuses on implementing technologies to address existing challenges in the field.

Nemade et al. [23] present an innovative Blockchain framework for crowdfunding cyber-product insurance. The challenges in cyber-risk management are addressed using Blockchain technology to enhance security, transparency, and trust in cyber-insurance. The paper describes a system where cyber insurance requests are auctioned through Smart Contracts, allowing insurers to bid in a sealed manner. This approach simplifies the cyber insurance procurement process and promotes a transparent environment for determining cyber insurance premiums. The proposed framework includes mechanisms for verifying indemnity requests and distributing claims by utilizing Blockchain technology for data integrity and preventing fraud. In nutshell, the significant contribution of the paper is the proposal of a scalable and efficient solution to cyber-risk management and cyber insurance procurement.

Overall, previous works have proposed the integration of Blockchain and Smart Contracts in the cyber insurance domain [17–19, 21–23]. These endeavors have focused on diverse aspects, including fostering trustworthy interactions between PHs and ICs, dynamic risk assessment, crowdfunding frameworks, time efficiency improvement, effective risk management, and addressing information asymmetry. Moreover, INCHAIN [14] stands out for introducing, for the first time, the SSI into the cyber insurance ecosystem. The authors conceptualize an architecture that can be hosted within this ecosystem. However, to the best of our knowledge, existing literature neither develops nor integrates SSI within the cyber insurance ecosystem to quantitatively assess its features. SSI can enhance the security and privacy of identity management, granting PHs control over their personal data. While current Blockchain-based cyber insurance systems assist ICs in tailoring insurance premiums and enable PHs to validate coverage and conduct claim-related transactions seamlessly,

SSI could introduce additional benefits. Specifically, it empowers PHs to have sovereign control over their identities and personal information in cyber insurance. Recognizing this gap in the literature, the current article addresses it by implementing and assessing the integration of SSI within the cyber insurance ecosystem. This research aims to contribute to understanding the quantitative impact of SSI features in enhancing the overall functionality and security of cyber insurance systems.

3 Cyber Insurance Ecosystem Featuring Smart Contracts and Self-Sovereign Identity on Top of Blockchain

This section sets the research questions for this study. At the same time, it briefly analyzes the most common challenges encountered within the cyber insurance ecosystem, its operations, and a security model, along with security and privacy requirements essential for ensuring the trustworthy operation of this ecosystem's framework. Notably, this article employs numerous acronyms to denote specific topics and concepts.

3.1 Challenges and Operations of the Classic Cyber Insurance Ecosystem

While diverse forms of cyber insurance have existed since the late 1990s, the market has, until recently, experienced relatively slow development. The significant losses incurred during the pandemic in recent years have highlighted the immaturity of the cyber insurance market. This recognition has spurred ICs to embrace a more innovative approach. Developing a robust and flourishing cyber insurance market necessitates innovation within the insurance industry, opening up opportunities for new entrants. This includes a wave of startups dedicated to addressing the intricate technical challenges [14] associated with comprehending, quantifying, and mitigating cyber risks. Successfully capitalizing on this multi-billion-dollar market demands a nuanced understanding and the implementation of effective solutions.

On one hand, ICs face growing profitability challenges due to increasing claims and escalating operational expenses. Two major contributors to these challenges are the rise of *fraudulent claims* and *identity theft*. Fraudulent claims occur when deceptive PHs submit duplicate claims to multiple insurers for the same cyber incident, as documented in several studies [24–26]. Identity theft involves malicious actors impersonating legitimate PHs to file false claims or exploit stolen identities [22, 27–31].

In addition to fraud-related threats, ICs often operate with *limited historical data* regarding their clients' assets and security postures. This limitation stems from inaccuracies or gaps in critical information, including asset inventories, revenue details, data processing practices, and implemented security controls [32–34]. The absence of comprehensive data hinders the underwriting process and complicates effective risk assessment. Compounding this issue is the lack of robust *Know Your Customer (KYC)* practices, which impedes insurers' ability to verify and monitor client behavior accurately [35, 36]. Furthermore, as repositories of sensitive personal and financial information, ICs have become lucrative targets for cybercriminals, making the safeguarding of personal data a pressing concern.

On the other hand, PHs encounter their own set of challenges when engaging with cyber insurance products. Chief among these is the complexity of existing policy structures, which is largely driven by *information asymmetry*—a condition in which one party, typically the insurer, possesses significantly more information than the other [37–40]. As a result, many PHs struggle to fully comprehend policy terms and conditions, leading to uncertainty, potential coverage gaps, and dissatisfaction [41]. Another commonly cited issue is the protracted nature of cyber insurance processes [8], exacerbated by a widespread *lack of automation* in claims management and related procedures [19, 42]. Critical tasks, such as reimbursement processing, remain largely manual and reliant on outdated systems, resulting in significant delays. These delays can hinder organizational recovery, potentially resulting in reputational damage and financial loss for affected PHs.

In summary, the cyber insurance ecosystem presents numerous challenges for both insurers

and policyholders. ICs must navigate fraud, identity theft, data scarcity, and security risks, while PHs face policy complexity, informational disadvantages, and inefficiencies in claims processing. Addressing these systemic issues is crucial for enhancing transparency, streamlining operations, and improving the overall effectiveness of cyber insurance offerings.

Except for the challenges mentioned above, the community has also studied and addressed other challenges such as *Interdependent and Correlated Risks*, and *Premium Calculation* [43–45]. Interdependent and correlated risks present significant challenges in the field of cyber insurance, particularly when it comes to premium calculations. Unlike traditional insurance, cyber risks often exhibit interdependencies, where a single cyber incident can have cascading effects across interconnected systems.

Additionally, risks in the cyber domain can be highly correlated, meaning that multiple organizations can be affected simultaneously by a common threat or vulnerability. These characteristics make it challenging to assess accurately and price premiums for cyber insurance policies. ICs must consider the potential systemic impact of cyber events, evaluate the interdependencies between PHs and their ecosystem, and factor in the correlation of risks when determining premium rates. Developing sophisticated modeling techniques and leveraging comprehensive data sets has become crucial in addressing these challenges and ensuring fair and accurate premium calculations in the dynamic landscape of cyber insurance.

The operations that define cyber insurance involve several key stages. The cyber insurance processes commence with PHs exploring the cyber insurance market to identify potential ICs and evaluate their specific needs. This research aids ICs in tailoring their cyber insurance offerings to match the individual requirements of PHs. Upon identifying potential PHs, the subsequent step includes registering and validating PHs. During this stage, PHs provide essential information to establish their identity and assess their cybersecurity risk profile. This information proves critical during underwriting, where ICs evaluate the risk associated with each PH, determining appropriate coverage and premiums.

Periodic risk assessments are then conducted to monitor changes in PHs' risk profiles and adjust coverage accordingly. In the event of a cyber incident, PHs can submit claims, which undergo validation and auditing to ensure compliance with policy terms and conditions. Once a claim is approved, the IC proceeds with the claim payment, offering financial assistance to the PH as outlined in the policy agreement. These comprehensive processes shield businesses against cyber risks and facilitate a seamless and efficient claims management system.

In addition to these market and operational challenges, current literature emphasizes the necessity of integrating privacy-preserving and automated techniques into the cyber insurance ecosystem. In "Cyber Insurance – Models and Methods and the Use of AI" [46], ENISA experts proceed to analyze contemporary underwriting and risk assessment methodologies, highlighting the growing significance of AI in improving forecast accuracy. The report also notes ongoing challenges in data sharing and trust-building, as well as the potential for automation in claims processes, without prescribing specific privacy-preserving techniques. Complementary research has explored data-driven, entity-specific approaches for cyber risk assessment, leveraging InsurTech-enabled machine learning models to enrich datasets and enhance underwriting and risk profiling [47]. Techniques like Federated Learning (FL) and Secure Multiparty Computation (SMC) [48], while primarily studied in financial contexts, provide a conceptual framework for performing collaborative risk analysis in cyber insurance without exchanging sensitive raw data. Related research in distributed cybersecurity analytics [49] further illustrates the application of privacy-aware computation across heterogeneous environments. Initiatives that utilize blockchain in insurance operations offer valuable insights. Hao et al. [50] investigate blockchain-based parametric insurance that use zero-knowledge proofs to automate claims initiated by sensor or telemetry data. Although these methodologies illustrate the feasibility of privacy-preserving automation, they are predominantly confined to parametric insurance models. Nie et al. [51] present an approach for privacy-preserving record linking at the architectural level, describing the interaction of entities while minimizing information exposure. Together, all of

these research areas demonstrate a progressive shift towards more privacy-conscious and automated cyber insurance frameworks. Reports from The Geneva Association [52] and Moody's [53] underscore operational and market-level factors, such as risk accumulation, multi-jurisdictional compliance, and the necessity for automation in claims processing. Confronting these problems is essential for the continued development of reliable and sustainable cyber insurance applications.

In actual deployment, various operational factors must be addressed. Integrating with legacy IT systems requires standardized interfaces and hybrid deployment models that enable blockchain-based workflows to coexist with traditional claims and underwriting tools [46, 49]. Credential lifecycle management presents challenges including reissuance and revocation, demanding robust governance procedures and safe recovery mechanisms. Moreover, the transnational aspect of cyber insurance requires meticulous compliance with jurisdiction-specific data protection regulations, including data residency prerequisites and rights such as erasure requests, which can be managed through selective disclosure [46]. Other operational considerations involve ensuring system scalability, providing adequate stakeholder training, and enabling interoperability with third-party risk assessment systems [52]. Addressing these operational problems is crucial for ensuring the reliable adoption of privacy-preserving and automated cyber insurance systems in real-world implementations.

3.2 Research Questions

To achieve our overall objectives, we first need to understand what the potential of Blockchain with respect to cyber insurance and accordingly design our study along four research questions:

RQ1: *How effectively can the Blockchain technology be integrated into the cyber insurance ecosystem?*

RQ2: *What are the fundamental security and privacy requirements for a cyber insurance platform built on Blockchain?*

RQ3: *How can identity theft be prevented within the cyber insurance ecosystem?*

RQ4: *Which is the threat model of the cyber insurance ecosystem resulting from the incorporation of Blockchain implementations?*

Addressing the four outlined research questions is crucial for several reasons. Firstly, understanding the effectiveness of incorporating Blockchain technologies into the cyber insurance ecosystem is essential to fully utilize its capacity to enhance trust, transparency, and traceability in transactions, significantly reducing related risks to digital operations. Secondly, determining the security and privacy requirements for a Blockchain-based cyber insurance framework ensures compliance with strict security standards and protects the privacy of PHs. Thirdly, investigating strategies to prevent identity theft within the cyber insurance system is vital to establish robust mechanisms that safeguard against identity theft and maintain the confidence of both PHs and ICs, ensuring operational security. Finally, examining the potential threats that Blockchain technologies may pose to the cyber ecosystem is crucial to proactively identifying and addressing any emerging threats.

3.3 Ecosystem Overview

The cyber insurance ecosystem featuring Smart Contracts and SSI is an ecosystem that blends the classic cyber insurance ecosystem [14, 54] and the Blockchain features. This ecosystem allows organizations and humans to purchase cyber insurance services being sold by ICs and become PHs, achieving the fairest cyber insurance premium based on their verified attributes being enclosed in their VCs issued by their National Cyber Security Authority (NCSA). The basic usage scenario of the decentralized cyber insurance ecosystem featuring Smart Contracts and SSI on top of Blockchain unfolds as follows. At some point, a possible PH is an organization that as an entity provides services (i.e., media streaming) and is equipped with assets (i.e., hardware, software, employees, procedures) [37] to successfully deliver its services to its clients (i.e., subscribers). These assets are valuable to the organization and critical to its operations, reputation, and competitive advantage. However, due to its physique, the organization is exposed to external cybersecurity

threats, such as hackers attempting to gain unauthorized access to its systems, or internal threats, such as employees intentionally or unintentionally compromising the security of the organization's assets. Thus, the first step is to identify, assess, and manage their risks and threats to safeguard its service from reducing its cybersecurity risk and financial exposure by offsetting costs associated with systems and data recovery during and after a cybersecurity incident. Overall, the organization as potential PH directly communicates with an Cyber Insurance Broker (CIB) responsible for delivering all possible cyber insurance contracts that meet its requirements related to its assets and cybersecurity exposure.

Once the potential PH ends up with a contract by a IC they establish a communication channel to proceed with the cyber insurance service acquisition. However, the IC to sell a cyber insurance contract has required information that shall be collected by each potential PH. These requirements are the following:

- ***Basic company details*** that is general information including the organization name, primary industry sector, primary address (province, country), description of business activities, official website, establishment date, number of employees, gross revenue, and gross profit.
- ***Risk related data*** where the organization has to confirm whether multi-factor authentication is always enabled on processes and maintain daily offline backups of all critical data.
- ***Experienced cybersecurity incidents*** where the organization has to confirm the type (i.e., ransomware) of its previous cybersecurity incidents and that the total number of its experienced cybersecurity incidents shall equal less than ten.
- ***Resources and infrastructure*** where the potential PH has to provide an accurate and updated asset list (i.e., software inventory).
- ***Information security governance*** where the organization has to describe the collected, stored, and processed data and define the standards (i.e., ISO/IEC 27001) that the organization complies with.
- ***Cybersecurity controls*** where the organization has to deliver an updated list with its installed cybersecurity controls (including but not limited to firewalls, IDS, and DDoS mitigation tool).

Thus, the potential PH to purchase the cyber insurance service, firstly, has to obtain VCs by its verified issuer (e.g., NCSA), providing that its attributes meet the IC requirements. The potential PH and its NCSA establish a secure communication channel, where the potential PH sends the required documents (i.e., licenses, certifications, asset list) and the NCSA issues VCs that prove the PH's attributes. Once the PH obtains these VCs, it forwards them to its chosen IC to purchase the cyber insurance contract. Next, the IC and the PH upload the cyber insurance contract to the Cyber Insurance Blockchain. The Cyber Insurance Blockchain is a consortium Blockchain that belongs to a group of ICs and contains cybersecurity data of their PHs and their corresponding cyber insurance contracts. Later, the PH can use the cyber insurance service and Smart Contract functions to manage a cybersecurity incident.

A notable advantage of the decentralized cyber insurance ecosystem featuring Smart Contracts and SSI on top of Blockchain is the promotion of automated cyber insurance operations and the disincentivization of ICs to needlessly collect data when they aim to sell a cyber insurance contract. Thus, the proposed ecosystem promotes an ecosystem dedicated to cyber insurance equipped with fully automated processes via Smart Contracts, creating an agile environment where both ICs and PHs can perform their operations via functions geared toward the Smart Contracts. Next, the proposed ecosystem urges the involved ICs to collect only data required for calculating the cyber insurance premium and monitoring the behavior of each PH to confirm that each PH complies with its cyber insurance contract obligations.

In general, the decentralized cyber insurance ecosystem featuring Smart Contracts and SSI on top of Blockchain enables innovative business cases and new actors. In particular, we identify four key stakeholders: (*i*) the PHs; (*ii*) the ICs; (*iii*) the NCSA and (*iv*) the CIB. A fifth stakeholder is a secondary market that may emerge to address the need for new specialized software to provide more in-depth cybersecurity data for each potential PH, as well as for accurate monitoring of each IC cybersecurity behavior. On a technical

level, the entities that participate in the decentralized cyber insurance ecosystem featuring Smart Contracts and SSI on top of Blockchain are the following:

- **PH** is a cyber insurance holder and Hyperledger Aries holder since he is equipped with VCs to prove his requirements in a privacy-preserving manner.
- **IC** is accountable for selling cyber insurance policies to prospective PHs, investigating a cybersecurity incident, and auditing PH compliance with cyber insurance policies and implementation of the indicated cybersecurity countermeasures. ICs are also Hyperledger Aries verifiers since they utilize this technology to verify that their potential PHs meet their requirements to obtain a cyber insurance contract and to allow their PHs to prove their identity before being enabled to execute cyber insurance services.
- **NCSA** is an SSI issuer that issues VCs to potential PHs based on their identity attributes that have been verified.

3.4 Security Model

To systematically analyze the security threats within the proposed cyber-insurance ecosystem, we apply the STRIDE threat modeling framework (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege). The following threats were identified based on the assumed behaviors of PHs, ICs, and potential external attackers (see Table 1) providing a clear view of how different adversarial behaviors affect PINSA's confidentiality, integrity, and availability.

- **Spoofing**: PHs may attempt to submit verifiable credentials (VCs) issued by untrustworthy entities or impersonate legitimate issuers to misrepresent their cybersecurity posture. External attackers may also spoof legitimate entities to intercept or alter communication with the blockchain networks.
- **Tampering**: Both PHs and ICs can modify data to manipulate outcomes—PHs by altering claim or credential information, ICs by performing biased or falsified incident investigations. External attackers could attempt to modify Smart Contracts or data stored on the Cyber Insurance Blockchain.
- **Repudiation**: PHs or ICs might deny participation in fraudulent or inaccurate actions, such as duplicate claim submissions or falsified investigations, especially if logging and auditing mechanisms are weak.
- **Information Disclosure**: An adversary intercepting communication channels could access sensitive information exchanged among PHs, ICs, and the blockchains. Improper handling of incident reports by ICs may also leak confidential PH data.
- **DoS**: External attackers may attempt to disrupt the availability of the Cyber Insurance Blockchain or the SSI Blockchain to halt insurance operations.
- **Elevation of Privilege**: PHs could exploit protocol loopholes to receive multiple reimbursements for the same incident, while attackers may gain unauthorized control over Smart Contracts or blockchain nodes.

3.5 Security and Privacy Requirements

As mentioned, the cyber insurance ecosystem combines features from the classic insurance ecosystem and information technology. The cyber insurance ecosystem inherits the security and privacy requirements of both. Since the existing literature does not define them, this work is the first to define the security and privacy requirements that shall be met by solutions (architectures, frameworks, schemes) that aim to lie within the cyber security field. For solutions designed to address the needs of *ICs* and *PHs* who are interested in selling and purchasing cyber insurance and wish to be its users, we define the following security and privacy requirements after considering the cyber insurance ecosystem on top of Blockchain, users' security and privacy demands and the related research.

3.5.1 Security Requirements

The definition of security requirements is responsible for identifying and quantifying the potential cybersecurity risks a cyber insurance system may face. The definition also reduces the risk of cybersecurity incidents (i.e., data breaches) and minimizes their impact. Thus, the security

Table 1 Threat model based on STRIDE

Threat	STRIDE Category
PH submits VCs with inaccurate or manipulated attributes	Spoofing, Tampering, Repudiation
PH submits duplicate claims for the same cybersecurity incident to multiple ICs	Tampering, Repudiation, Elevation of Privilege
Collusion between PHs to fabricate or stage a cybersecurity incident	Spoofing, Tampering, Repudiation
IC conducts inaccurate or biased cybersecurity incident investigations to avoid reimbursement	Tampering, Repudiation, Information Disclosure
External attacker attempts to modify Smart Contracts or data on the Cyber Insurance Blockchain	Tampering, Elevation of Privilege
Tampering, Elevation of Privilege	Information Disclosure, Tampering, Spoofing
External attacker disrupts the availability of the Cyber Insurance Blockchain or SSI Blockchain	Denial of Service

requirements that should be met by systems dedicated to cyber insurance on top of Blockchain are presented below:

(**S1**) ***Data confidentiality:*** Cyber insurance data and Smart Contracts stored within the Cyber Insurance Blockchain must be available only to the responsible *ICs* and the corresponding PH. No external entities may collude to gain information to read the aforementioned information.

(**S2**) ***Data integrity and authenticity:*** All data exchanged between the participating entities should be protected against tampering and duplication. Each participating entity must be able to confirm the data provenance.

(**S3**) ***Non-repudiation:*** None of the PH and IC should be able to deny their actions related to their obligations against the purchased cyber insurance contract.

(**S4**) ***Authorization and access control:*** Access to the Smart Contracts should be granted only to legitimate PHs registered at the *ICs* that handle the Cyber Insurance Blockchain.

(**S5**) ***Accountability:*** The participating entities should be held accountable for their actions.

(**S6**) ***Physical protection:*** The participated *PHs* should be equipped with protection mechanisms to protect the exchanged *VCs* being tampered with by adversaries with physical access (e.g., steal, transfer, and modify the VC).

3.5.2 Privacy Requirements

The privacy requirements set for cyber insurance systems on top of Blockchain are mainly *human-related* since cyber insurance is for organizations that consist of employees. These requirements lead the systems to comply with legal and regulatory compliance (i.e., General Data Protection Regulations (GDPRs)), become ethical and trustworthy for all participating entities, highly reputable among the existing and future PHs, and prevent illicit competition among the ICs. Thus, the privacy requirements that should be met by systems dedicated to cyber insurance on top of Blockchain are presented below:

(**P1**) ***Identity privacy:*** The participating PHs' true identity should only by known to its legitimate ICs and NCSA.

(**P2**) ***Minimum data disclosure:*** The system should guarantee that the participating ICs should access a possible PH's data limited to the minimum required to bill them for the purchased cyber insurance contract.

(**P3**) ***Privacy-preserving data accumulation:*** Data exchange among the ICs that participate in the Cyber Insurance Blockchain should be transferred under a secure and privacy-preserving manner to protect PH's identity from being disclosed or modified by unauthorized parties and prohibits future overpricing of a specific PH.

4 Architectural Overview of `PINSA`

We now present a blueprint of the `PINSA` architecture, along with the protocol stack of each involved entity, as depicted in Figure 1. At this point, it is important to remind the reader that `PINSA` is not a vanilla implementation; rather, it is built upon INCHAIN, with which it shares several concepts. Nevertheless, it introduces some key differences. Most notably, `PINSA` transitions from a conceptual cyber insurance architecture to a fully operational framework with designated technologies.

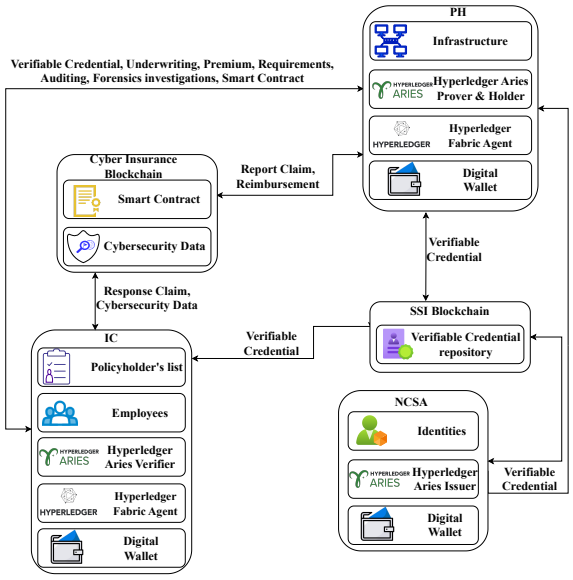

Figure 1 `PINSA` architectural components

PH: The PH owns a comprehensive infrastructure that encompasses various assets, including employees, digital assets, hardware, and operational components. Moreover, the PH functions as a Hyperledger Aries holder, intending to acquire Aries VCs for accessing cyber insurance services offered by its corresponding IC. Additionally, the PH is equipped with a digital wallet designed to securely store the necessary cryptographic keys and issued VCs . Furthermore, the PH is furnished with a Hyperledger Fabric agent, facilitating access to the Cyber Insurance Blockchain supported by its IC. This capability enables the PH to execute Smart Contracts directly linked to its cyber insurance agreement. PHs engage with their NCSA to issue VCs based on verified attributes. They also interact with ICs to exchange VCs and address requirements related to the cyber insurance contract.

IC: The IC maintains a diverse client list, encompassing both active and previous (*inactive*) PHs. Additionally, the organization employs a dedicated team responsible for executing essential cyber insurance processes (refer to Section 3). These processes include underwriting during the initial phase of cyber insurance and conducting forensics investigations in response to cybersecurity incidents. The IC serves as a Hyperledger Aries verifier, tasked with validating the VCs presented by its associated PHs. To facilitate VC verification, the IC communicates with the SSI Blockchain using a cryptographic key securely stored within its digital wallet. Furthermore, the IC is equipped with a Hyperledger Fabric agent, enabling communication with the Cyber Insurance Blockchain. This functionality allows the IC to upload Smart Contracts and execute relevant functions to manage and respond to cybersecurity incidents effectively. In essence, the IC performs various cyber insurance-related actions, such as underwriting, forensics investigation, and auditing, directly for its PHs. The IC collects the following data from the PH: (i) the basic company details, (ii) information security governance, (iii) resources and infrastructure, and (iv) experienced cybersecurity incidents.

NCSA: Functioning as a Hyperledger Aries issuer, the NCSA enables prospective PHs to issue Hyperledger Aries credentials directly to their digital wallets based on verified identity attributes. Subsequently, these credentials are utilized to access cyber insurance services. The NCSA employs a Digital Wallet, securely storing cryptographic keys crucial for performing Hyperledger Aries credential issuance. Collaborating with relevant ministries, government agencies, and entities, the NCSA ensures the issuance of VCs with valid information for all PHs, thereby guaranteeing interoperability at every level. Furthermore, the NCSA uploads references of issued VCs to the SSI Blockchain , exclusively communicating with potential PHs and the SSI Blockchain network. Notably, the NCSA diligently maintains information on the latest cybersecurity events involving PHs, crucial for generating accurate and updated VCs. Recognizing the necessity for issuing VCs with precise and current information, the introduction of a dedicated entity for this purpose becomes imperative. Present stakeholders lack the confidence to assert accurate claims to VCs, a void effectively filled by the NCSA with high confidence.

SSI Blockchain: Ownership of this Blockchain network lies with the NCSA. This is responsible issue and verify the Hyperledger Aries credentials, as well as store references to these. The ICs access this Blockchain network for issuing purposes, while PHs utilize it for verification purposes.

Cyber Insurance Blockchain: Implemented on the well-established Hyperledger Fabric, this Blockchain consists of predefined ICs entrusted with transaction verification. These ICs have collaboratively joined forces to share cybersecurity data concerning their respective PHs. This collective effort aims to enhance operational workflows, transparency, and accountability within the ecosystem. The shared data encompasses, but is not limited to, the reputation of PHs, their cybersecurity behavior, and past violations. Moreover, the cyber insurance contracts between ICs and their PH are securely stored within this Blockchain. Additionally, the Blockchain is tasked with maintaining and processing Smart Contracts pertaining to cyber insurance.

5 System Architecture and Operations

`PINSA` comprises the following operations: (*i*) Verifiable Credential Issuance, (*ii*) Verifiable Credential Verification and Cyber Insurance Issuance, and (*iii*) Cybersecurity Incident Report and Reimbursement. The `PINSA` architecture deliberately excludes actions involving the selection of a cyber insurance contract between a potential PH and CIB. Although this is integral to the cyber insurance backbone, it falls outside the scope of this work. Each `PINSA` operation's purpose and relationship to existing cyber insurance processes are explored below.

5.1 Credential Issuance within the `PINSA` Ecosystem

As indicated by its name, this operation is responsible for issuing Hyperledger Aries VCs to a PH based on their attributes. It occurs between a PH and its NCSA. The aim is to establish a reliable identification mechanism to facilitate the accurate data transfer from the PH to the IC. Specifically, this operation integrates automated procedures into the cyber insurance process's standard Client Registration and Validation step. These procedures are designed to furnish PHs with NCSA-verified data that does not necessitate human intervention for validation by ICs.

For the VC issuance process (refer to Steps 1 – 14 illustrated in Figure 2), we begin with the

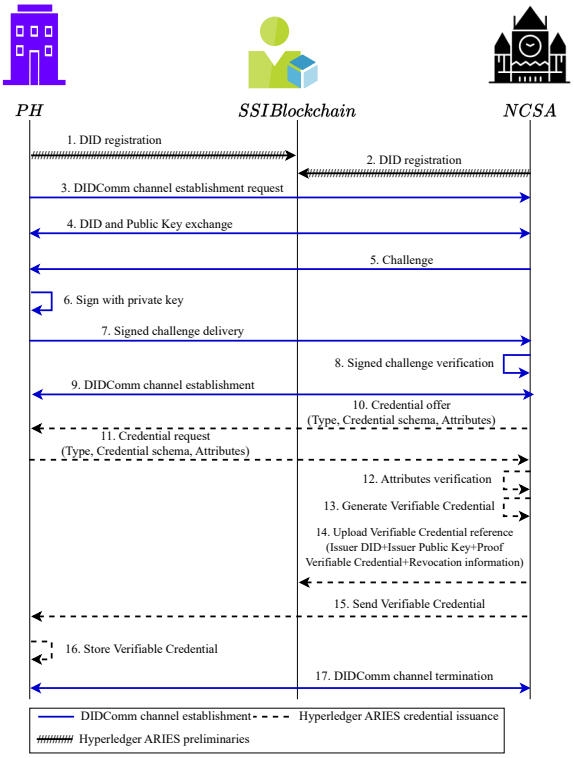

Figure 2 `PINSA` Hyperledger Aries credential issuance (bold lines highlight `PINSA`'s novelty)

assumption that both the PH and NCSA generate a cryptographic key pair. The public key is encoded into a specific format, which will later represent the Decentralized Identifier (DID) (e.g., SHA-256) to be shared with other entities, while the private key is securely stored within a digital wallet (see Figure 1). Subsequently, both the PH and NCSA publish their DID to the decentralized *SSI Blockchain* network.

The Hyperledger Aries VC issuance is initiated when the PH sends a DIDComm channel establishment request via message to its corresponding NCSA, and both entities exchange their DIDs and public keys (Steps 3 – 4). Before establishing a Decentralized Identifier Communication (DIDComm) channel with the corresponding PH, the NCSA aims to authenticate the PH using a challenge-response protocol. The PH signs the challenge with its private key, securely stored within its digital identity wallet, and sends it back to the NCSA for verification. Notably, the user's device must be secure from attacks attempting to retrieve the private key. If the verification is

successfully completed, the DIDComm channel is established (Steps 5 – 9).

Once the DIDComm channel is established, the NCSA, acting as the VC issuer, initiates the Hyperledger Aries VC issuance by sending a VC offer to the PH (Step 10). This offer includes information about the credential type, schema, and required attributes. Subsequently, upon receiving the VC offer, the PH sends a VC request to the NCSA, expressing the intent to accept the offer and proceed with the issuance process. The credential exchange request includes the PH's DID (serving as a unique identifier), the credential type, the credential schema, and their verified attributes (Step 11).

Next, the NCSA verifies the received attributes and generates the Hyperledger Aries credential. As a public sector entity, the NCSA collects verified data from other ministries, government agencies, and bodies, allowing it to issue the *Basic-Company-Details-VC* and *Experienced-Cybersecurity-Incidents-VC* directly. However, regarding the PH's asset inventory and governance, the NCSA lacks information. Consequently, the PH transmits its verified attributes to the NCSA. These qualifications include certificates demonstrating compliance with ISO27001 and GDPR, along with the most recent security update occurrences issued by reputable companies (e.g., the service provider).
In summary, the PH obtains the following four Hyperledger Aries VCs:

- **Basic Company Details:** This category furnishes general information about the potential PH, encompassing its name, primary field of business, main address (province, nation), operational description, a link to its official website, the year of establishment, the number of employees, gross revenue, and gross profit.
- **Experienced Cybersecurity Incidents:** This section verifies that the PH has fallen victim to three distinct cybersecurity attacks (ransomware, website defacement, BEC), with a cumulative fine totaling 700K €.
- **Resources and Infrastructure:** This category provides evidence that the PH has implemented all requisite security controls (i.e., IDS, firewall, backup policy routine) and has installed the latest available updates across its assets.
- **Information Security Governance:** This section verifies that the PH adheres to ISO27001 and GDPR compliance standards and explicitly states that the PH exclusively stores and processes Personal Identifiable Information (PII) data.

Listing 1 PINSA VC

```
1  {
2    id: "https://ncsa.gr/verifiablecredential/123455",
3    type: ["VerifiableCredential"],
4    issuer: {
5      id: "did:ncsa:1234545667",
6      name: "National Cyber Security Agency",
7      url: "https://ncsa.gr"
8    },
9    issuanceDate: "2022-04-15T10:30:00Z",
10   expirationDate: "2023-04-15T10:30:00Z",
11   credentialSubject: {
12     id: "did:ph:012023",
13     total loss: "700K EUR",
14     past cybersecurity incidents: [
15       {
16         incident: "Ransomware",
17         Date: "2018-01-01",
18         Loss: "300K EUR"
19       },
20       {
21         incident: "BEC",
22         Date: "2020-02-05",
23         Loss: "300K EUR"
24       },
25       {
26         incident: "website defacement",
27         Date: "2022-05-09",
28         Loss: "100K EUR"
29       }
30     ]
31   },
32   proof: {
33     type: "Ed25519Signature2018",
34     created: "2023-06-05T00:00:00Z",
35     proofPurpose: "assertionMethod",
36     verificationMethod: "did:ncsa#key1",
37     signatureValue: "GKugZG68B7N6FGNJNK"
38   }
39 }
```

For instance, a Hyperledger Aries VC pertaining to *Experienced-Cybersecurity-Incidents* follows the following structure (refer to Listing 1 for an example of the *Experienced-Cybersecurity-Incidents-VC* in the Hyperledger Aries VC structure presented in *.json* format, as utilized within `PINSA`):

- **id:** Unique identifier for the VC (e.g., *https://nsca.gr/verifiablecredential/123*) [Line 2].

- **type:** Indicates the type of the credential that is verifiable (e.g., *Basic-Company-Details-VC*) [Line 3].
- **issuer:** Contains information about the corresponding VC's issuer, including their name (e.g., NCSA), DID (e.g., *did:ncsa:1234545667*), and URL (e.g., *https://www.ncsa.gr*) [Lines 4 – 7].
- **issuanceDate:** The date and time when the credential was issued (e.g., *2023-06-05T10:30:00Z*) [Line 9].
- **expirationDate:** The date and time when the credential expires (e.g., *2024-06-05T10:30:00Z*) [Line 10].
- **credentialSubject:** Contains information about the subject of the credential [Lines 11 – 30]. In this example, it includes the PH's DID [Line 12], their past cybersecurity incidents [Lines 13 – 31], their total loss [Line 13], and separately the loss for each incident [Lines 16 – 18, 21 – 23, and 26 – 28].
- **proof:** Contains cryptographic proof of the authenticity and integrity of the credential, including the following information [Lines 32 – 39]:
  - **type:** Represents the specific type of the proof's signature (e.g., *Ed25519Signature2020*).
  - **creation date:** Represents the day of the proof's creation (e.g., *2022-31-12T00:00:00Z*).
  - **proof purpose:** Represents the purpose for the proof (e.g., *assertionMethod*).
  - **verification method:** The method used for verification by the verifier (e.g., selective disclosure).
  - **signature value:** Represents the signature (e.g., *GKugZG68B7N6FGNJNK*).

Subsequently, the NCSA stores a reference to the credential on the *SSI Blockchain* network, ensuring that the issued Hyperledger Aries VC remains secure and unmodifiable after issuance (Steps 12 – 14). This reference encompasses information related to the NCSA's DID, NCSA's public key, VC's proof, and revocation information. Following this, the NCSA transmits the generated VC to the PH. The PH then securely stores it within their digital identity wallet, and

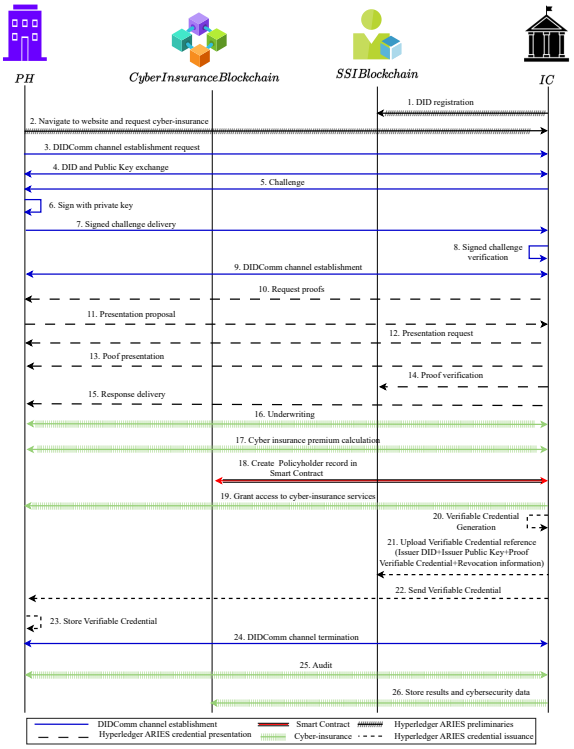

Figure 3 `PINSA` Hyperledger Aries Credential presentation (bold lines highlight `PINSA`'s novelty)

the DIDComm channel is terminated (Steps 15 – 17).

5.2 Verifiable Credential Presentation and Cyber Insurance Issuance within the `PINSA` ecosystem

In this operation, PHs submit their VCs to their chosen ICs, and upon successful verification, the PHs become eligible to access the ICs' cyber insurance services. Smart Contracts play a crucial role in this process by transforming the conventional cyber insurance contract into a digital format, binding both PHs and ICs to specific obligations. When a prospective PH intends to purchase a cyber insurance contract from a specific IC, it is necessary for the respective PH to submit their VCs to the IC for verification (refer to Figure 3 for Steps 1 – 26). As a prerequisite, we assume that the involved IC has already generated the required cryptographic key pairs, with the corresponding public key encoded into a specific format (e.g.,

SHA-256). This public key is uploaded and stored within the decentralized SSI Blockchain network (Step 1) to be shared with other entities, while the private key is securely stored within a digital wallet (see Figure 1). Additionally, the PH visits the corresponding IC's website and requests to purchase a cyber insurance contract.

First and foremost, the presentation procedure of the Hyperledger Aries VC is initiated between the IC and the PH by initiating a DIDComm channel request. Specifically, the potential PH submits a DIDComm channel establishment request via message to the relevant IC, and both parties exchange DIDs and public keys (Steps 3 – 4). Before establishing a DIDComm channel with the appropriate PH, the IC attempts to authenticate the latter using a challenge-response protocol and sends them the challenge. The PH signs the received challenge with their private key, securely stored within their digital identity wallet, and forwards it to the IC for verification. Clearly, the PH's wallet must be safeguarded from attacks attempting to recover the private key. If the verification is successful, the DIDComm channel becomes established (Steps 5 – 9).

Once the DIDComm channel is established, the IC, acting as the VC verifier, initiates the Hyperledger Aries VC presentation by sending a request to the PH for a specific set of verifiable credentials. Subsequently, the PH sends a presentation proposal message to the IC, specifying which credentials they wish to present. The IC then responds with a presentation request message, indicating any specific requirements and constraints for the presentation (Steps 10 – 12). Following this, the PH generates a proof of possession of the requested credentials, typically using zero-knowledge proof. The IC can then verify the proof and the contents of the presentation message and respond with a presentation acknowledgment or rejection message. All the aforementioned messages are established via the DIDComm channel (Steps 13 – 15).

After successfully completing the verification process, the IC and PH proceed with the underwriting and premium pricing procedures, culminating in establishing the cyber insurance contract agreement. Assuming a cyber insurance premium of 1,080 €, a liability limit of 591K €, and a deductible of 4K €, the cyber insurance procured by the corresponding IC may offer coverage for events against cybersecurity attacks, including social engineering, ransomware payment, malware/virus, as well as regulatory and legal compliance, with respective maximum compensation amounts of 1M €, 25K €, 7.5K €. Subsequently, this information is transcribed into a digitized configuration, transforming it into a Smart Contract that enforces specific obligations.

In addition to payment details, the cyber insurance contract imposes obligations on the PH, such as conducting penetration testing every two months, daily vulnerability scanning and patching, and organizing four cybersecurity awareness programs for PH's staff annually. The Smart Contract continuously verifies the PH's adherence to these obligations throughout the coverage period. Failure to comply with these obligations by the PH will result in the termination of the Smart Contract, rendering the PH ineligible for any form of reimbursement in the event of an incident (Steps 16 – 19). Moreover, the IC issues a VC to the PH to regulate access to the Cyber Insurance Blockchain network. This Blockchain comprises Smart Contracts and security data related to PHs, managing all aspects of cyber insurance (Steps 20 – 23).

A record within the `PINSA` Smart Contract includes the following attributes:

- PH_{ID}: A unique identification number indicating the PH (e.g., *0987654321*).
- IC_{ID}: A distinctive identifier describing the PH (e.g., *0192837465*).
- *Premium*: Represents the price of the cyber insurance contract (e.g., 5096 €).
- *Deductible*: Represents the amount the PH must pay before IC reimburses damages (e.g., 2K €).
- *Obligations*: Represents the obligations of both PH and IC under the cyber insurance contract (e.g., the PH shall perform four awareness phishing campaigns every four months).
- *Reputation*: A score evaluating the PH based on compliance with the cyber insurance contract requirements, starting at 100. If the PH violates obligations, the reputation decreases.
- $Incident_{ID}$: A unique identifier characterizing a cybersecurity incident associated with specific incident evidence (e.g., SOC logs)

submitted by the PH and investigated by the IC.

- *Incident*: The type of cybersecurity incident for which the PH is requesting reimbursement (e.g., ransomware).
- *Reimbursement*: The sum given by IC to PH to compensate for costs incurred due to the corresponding cybersecurity incident.
- *StartDate*: Represents the cyber insurance contract's starting date (e.g., *2023-08-15T00:00:00Z*).
- *EndDate*: Represents the cyber insurance contract's expiration date (e.g., *2024-08-15T00:00:00Z*).
- *Coverages*: A collection of covered cybersecurity incidents along with their meanings provided by IC.

The Smart Contract being used within the `PINSA` ecosystem is comprised by the following functions:

- ***PH Creation:*** This function creates the record for the PH within the corresponding IC's Smart Contract in the Cyber Insurance Blockchain network. It takes PH_{ID}, IC_{ID}, *Premium*, *Liability Limit*, *Deductible*, *Start Date*, and *End Date* as input and produces a new record containing this information. This function is exclusively executed by ICs.
- ***PH Reading:*** This function returns the cyber insurance contract for the PH stored on the Cyber Insurance Blockchain. It takes PH's and IC's unique identifiers (PH_{ID} and IC_{ID}, respectively) as input and returns the unique identifiers of both participating *PH* and IC, *Premium, Limit of Liability, Deductible, Start Date,* and *End Date* as output. This function is executed by both participating *PH* and IC.
- ***Incident Report:*** Triggered only by a PH, this function reports the occurrence of a cybersecurity incident. It takes PH_{ID}, IC_{ID}, $Incident_{ID}$, and *Incident* as inputs and alerts the corresponding PH regarding the incident as output.
- ***Incident Response:*** Responsible for approving or rejecting a reimbursement for a corresponding cybersecurity incident, this function is executed by an IC. It takes PH_{ID} and IC_{ID} as inputs and modifies the Limit of liability value as output.
- ***PH Obligation Checks:*** Executed by the IC to ensure PH adherence to obligations, this function takes PH_{ID} and IC_{ID} as inputs and outputs the value of Obligations along with a YES/NO indicating whether the PH meets them.
- ***Asset Transfer:*** Accessible through the *Responseincident* function, this function transfers funds from an IC to the PH. It takes PH_{ID}, IC_{ID}, and *Reimbursement* as inputs and notifies the PH that the asset transfer to its account was successful.
- ***Violation:*** Activated by the *CheckPHObligation* function, this function lowers the PH's reputation when the PH fails to satisfy its obligations. It outputs an updated value for *Reputation*.
- ***Contract Analysis:*** Prompted by both a PH and an IC, this function presents the incidents for which the PH is covered and its responsibilities concerning those coverages. It takes PH_{ID} and IC_{ID} as inputs and outputs the values of *Coverages* and *Obligations*.

Algorithm 1 illustrates a segment of the Smart Contracts functions within the `PINSA` ecosystem, specifically the *Analyze Contract*, *Check Obligations*, and *Violate* functions. We have chosen to present these functions as they are invoked during the investigation of a cybersecurity incident. The first function, *Analyze Contract* (Lines 1 – 7), takes PH_{ID} and IC_{ID} as input (Line 1). The function checks the existence of the corresponding identifiers and ensures they are associated with the same record (Line 2). If the comparisons hold true, the function returns the obligations the corresponding PH has against its cyber insurance contract for covered cybersecurity attacks (Lines 3 – 6). The second function, *Check Obligations* (Lines 9 – 26), is responsible for verifying whether the PH complies with its cybersecurity contract obligations. This function also takes PH_{ID} and IC_{ID} as input (Line 9) and checks the existence and alignment of the corresponding identifiers within the same record (Line 10). If the comparisons are valid, the function retrieves the current cybersecurity control of the PH along with its obligations under the cyber insurance contract (Lines

11 – 12). Subsequently, the function compares the number of PH's installed cybersecurity controls with its obligations (Line 13). If the comparison is not equal, the *Violate()* function is invoked (Lines 14 – 16). The function also examines whether the PH's installed cybersecurity controls match its obligations; if a discrepancy is found, the *Violate()* function is triggered (Lines 17 – 26). The final function, *Violate*, is responsible for decreasing the PH's reputation (Lines 18 – 32). Taking PH_{ID} and IC_{ID} as input (Line 28), the function checks for the existence and alignment of the corresponding identifiers within the same record (Line 29). If the comparison holds true, the reputation value is decreased by 10 (Lines 29 – 30).

Towards the conclusion, the IC issues a VC to the corresponding PH. The VC issued by the IC serves the purpose of authenticating the rightful ownership of the cyber insurance agreement by the associated PH. Through this credential, the PH gains access to the Smart Contract stored within the Cyber Insurance Blockchain network. This access empowers the PH to execute various actions pertaining to the cyber insurance contract, including the submission of reports on cybersecurity incidents. After that, the DIDComm channel is terminated, allowing the IC to initiate unscheduled audits on the PH. These audits aim to identify Smart Contract violations and enhance cybersecurity-related data concerning the PH, establishing a system for continuous risk monitoring (Steps 24 – 26).

5.3 Incident Report and Reimbursement within the PINSA Ecosystem

Throughout a cyber insurance policy, a PH may find it necessary to report a cybersecurity event to its IC for reimbursement. This process is integral to the cyber insurance policy and the rules outlined in the Smart Contract. Figure 4 visually depicts this process, detailing how the PH initiates the reporting of a cybersecurity incident, the subsequent investigation conducted by the IC, and the issuance of a payment order by the IC. These steps impact traditional cyber insurance procedures, namely *Claims Submission*, *Validation and Auditing of Claims*, and *Claims Payment*. To commence the *Claims Submission process*, the PH leverages automated features like

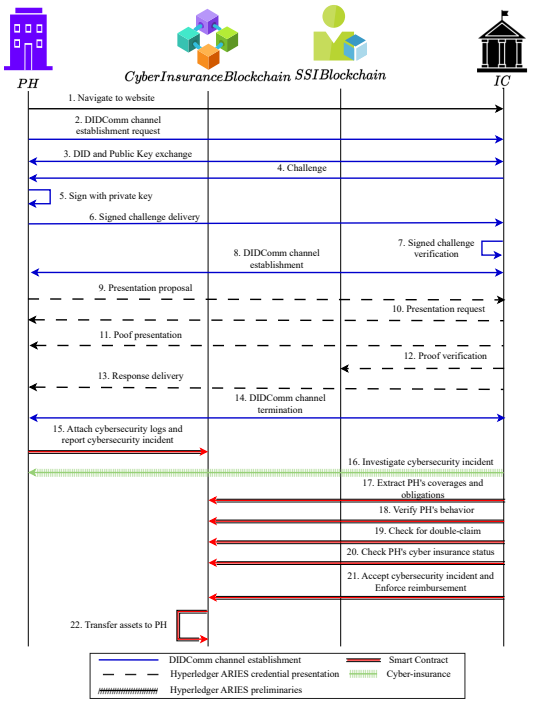

Figure 4 Cybersecurity incident handling within the PINSA ecosystem (bold lines highlight PINSA's novelty)

the *Incident Report* and *Incident Response* Smart Contracts. This automation replaces the cumbersome methods of reporting cybersecurity events via email and surveys. Similarly, the *Claims Payment* method is streamlined. Upon the IC consenting to reimburse the PH, a Smart Contract function is invoked, promptly transferring funds to the PH. PINSA enhances the *Claims Validation and Auditing* process by extracting accurate vulnerability information from the VCs. This data introduces novel insights, including employee reactions to simulated attacks and their adherence to organizational policies [34]. Additionally, the process gains flexibility through the *Check PH Obligations* Smart Contract function, which aids investigators by indicating whether the PH fulfills its obligations during the incident period.

To initiate the reporting of a cybersecurity incident (Steps 1 – 23 in Figure 4), the PH begins by navigating to the IC's website to carry out the reporting actions (Step 1). The first step involves

Algorithm 1 PINSA's Smart Contract Pseudocode

```
1: function AnalyzeContract(Policyholder_id, InsuranceCompany_id)
2: if exists PH record with PH_id = Policyholder_id and ICid = InsuranceCompany_id then
3:     coverages = coverages of PH record
4:     obligations = obligations of PH record
5: end if
6: return coverages, obligations
7: end function
8:
9: function CheckObligations(Policyholder_id, InsuranceCompany_id)
10: if exists PH record with PH_id = Policyholder_id and ICid = InsuranceCompany_id then
11:     controls = controls of PH record
12:     obligations = obligations of PH record
13:     if number of controls != number of obligations then
14:         call Violate(Policyholder_id, Insurancecompany_id)
15:         return false
16:     end if
17:     for each key of controls do
18:         if controls[key] != obligations[key] then
19:             call Violate(Policyholder_id, Insurancecompany_id)
20:             return false
21:         end if
22:     end for
23:     return true
24: end if
25: return false
26: end function
27:
28: function Violate(Policyholder_id, InsuranceCompany_id)
29: if exists PH record with PH_id = Policyholder_id and ICid = InsuranceCompany_id then
30:     reputation of PH record = reputation of PH record - 10
31: end if
32: end function
```

the initiation of the presentation procedure of the Hyperledger Aries VC between the IC and the PH, establishing a DIDComm channel. Specifically, the potential PH submits a DIDComm channel establishment request via message to the relevant IC, and both parties exchange DIDs and public keys (Steps 3 – 4). Before establishing a DIDComm channel with the designated PH, the IC employs a challenge-response protocol to authenticate the PH and sends a challenge to them. The PH signs the received challenge with its private key, securely stored within its digital identity wallet, and sends it back to the IC for verification. It is imperative to note that the PH's wallet must be safeguarded against potential attacks seeking to retrieve the private key. If the verification succeeds, the DIDComm channel is established (Steps 5 – 8).

Once the DIDComm channel is successfully established, the IC, serving as the VC verifier, initiates the Hyperledger Aries VC presentation by sending a request to the PH for a specific set of verifiable credentials. Subsequently, the PH responds by sending a presentation proposal message to the IC, specifying the credentials it intends to present. Following this, the IC issues a presentation request message outlining any specific requirements and constraints for the presentation. Upon receiving the presentation request, the PH generates a proof of possession of the requested credentials, typically employing zero-knowledge proof. The IC then verifies the proof and the contents of the presentation message, responding with either a presentation acknowledgment or rejection message. It is crucial to note that all these interactions occur within the established DIDComm channel. The DIDComm channel is terminated upon the completion of the VC presentation (Lines 9 – 13).

Following the successful completion of the VC presentation, the PH provides comprehensive reports and data, such as SOC logs, detailing the cybersecurity event (Step 15). Subsequently, the IC investigates the event (Step 16) and decides whether to approve the refund request based on their findings. Additionally, the IC utilizes Smart Contract functions to ascertain the PH's contractual responsibilities and ensure compliance. The IC delves into the Cyber Insurance Blockchain to confirm that the PH has not submitted the same claim (e.g., recovery costs from the same BEC attack) to another IC. Furthermore, the IC verifies the PH's limit of liability and deductible. Subsequently, the IC checks whether the PH has

sufficient funds in coverage. The process concludes when the amount equals zero. In this scenario, the IC approves and triggers automatic payment to the PH's wallet. With 96K€, the IC compensates the PH. Following reimbursement, the responsibility limit is immediately reduced to 404K€ for subsequent incidents. If the request is declined or a duplicate claim is identified, the process concludes in Step 7 or Step 8, depending on the circumstances. Additionally, if the PH violates the rules of the Smart Contract, the IC initiates a Smart Contract function to impact the PH's reputation (Steps 17 – 22).

Table 2 Smart Contract Functions

Function Name	Query Request	Invoke Request	Total Requests
policyholderExists	1	0	1
readPolicyholder	1	0	1
analyzeContract	2	0	2
updatePolicyholder	1	1	2
reportIncident	1	1	2
responseIncident	1	1	2
violate	1	1	2
checkObligations	3	1	4

6 Evaluation

This section presents the evaluation of PINSA, including a detailed description of the setup and implementation of the testbed, a thorough performance evaluation of the Cyber Insurance Blockchain and the SSI Blockchain networks, as well as, a security and privacy analysis of the whole system against well-know cybersecurity attacks that can target the PINSA's backbone technologies affecting its security posture.

6.1 Experimental Setup and Implementation

This section outlines the experimental setup and implementation details of PINSA. We employed a permissioned immutable transaction ledger called Hyperledger Fabric [15] to maintain a distributed network of peer nodes. By developing a ledger, named *Cyber Insurance Blockchain*, we configured 2 nodes. We then distributed these nodes into two separate organizations: *Insurance Company 1* (IC_1) and *Insurance Company 2* (IC_2), forming a channel named Cyber Insurance Channel. In this channel, each organization has one peer. Therefore, the one peer of IC_1 is called $IC_1.peer0$, while the peer of IC_2 is named $IC_2.peer0$. The transaction ordering, packaging into blocks, and distribution for validation and commitment are managed by the Hyperledger Fabric Orderer, named Insurance Orderer. Our implementation utilizes the Raft consensus mechanism within the Orderer Organization to ensure agreement on transaction order and achieve a consistent state in the Blockchain ledger. More specifically, we chose Hyperledger Fabric version 2 with the Raft consensus method over version 1's Kafka-based solution due to its notable benefits in operational ease and system dependability. Raft simplifies network setup and maintenance by removing external dependencies such as Zookeeper, which is necessary for Kafka. It also offers a more direct and reliable consensus method that improves the fault tolerance and efficiency of the Blockchain network. Node JS is employed to develop the Smart Contract, named Cyber Insurance Contract, featuring ten different functions.

Another integral component is the Hyperledger Caliper [55], a benchmarking tool from the Linux Foundation, used to evaluate the efficacy of the Cyber Insurance Blockchain implementation. Hyperledger Caliper, executing on client machines, broadcasts transactions on the Fabric channel. As mentioned before, the nodes are the ICs, while the PHs are simulated through the Caliper's workload modules of workers (equal to 10). Hyperledger Indy, a permissioned identity Blockchain, is used for credential-based cyber insurance data. The Hyperledger Aries framework [16], built on top of Indy, implements secure DID channel establishment, issuance, revocation, and verification of cyber insurance data credentials. Peers serve as the issuer, holder, and verifier, and Aries Agents, developed using the Aries Cloud Agent Python [56], facilitate interaction, verifiable credential exchange, and DID document management. Aries agents consist of the Aries Framework component for interaction with ledgers, wallets, and other agents, and the Aries Controller component to control behavior based on business rules. Each Aries agent maintains a wallet using an SQLite database [57], which is the location where secret keys, connections, and verifiable credentials are stored.

Table 3 Smart Contract workflows

Function Name	Input Parameters	Internal Calls
policyholderExists	PH_{ID}, IC_{ID}	-
readPolicyholder	PH_{ID}, IC_{ID}	-
analyzeContract	PH_{ID}, IC_{ID}	policyholderExists, readPolicyholder
updatePolicyholder	PH_{ID}, IC_{ID}, risklevel, totalMoneyRisk	readPolicyholder
reportIncident	PH_{ID}, IC_{ID}, incidentID, incidentName, indemnification	readPolicyholder
responseIncident	PH_{ID}, IC_{ID}, incidentID	readPolicyholder
violate	PH_{ID}, IC_{ID}	readPolicyholder
checkObligations	PH_{ID}, IC_{ID}	policyholderExists, readPolicyholder, violate

Table 4 Smart Contract data attributes

Function Name	Attributes Read	Attributes Updated
policyholderExists	PH_{ID}, IC_{ID}	-
readPolicyholder	all PH attributes	-
analyzeContract	obligations, coverages	-
updatePolicyholder	all PH attributes	riskLevel, totalmoneyRisk
reportIncident	all PH attributes	incidents list
responseIncident	all PH attributes (incidents, deductible, limit, cost)	limit
violate	reputation	reputation
checkObligations	controls, obligations, reputation	reputation (via violate)

Notably, all ICs' peers, Hyperledger Caliper client machines, and the SSI Blockchain peer are deployed using Docker containers. This software configuration was deployed on an Ubuntu 22.04 personal computer with an Intel CPU @ 3.20 GHz and 12 GB RAM for our proof-of-concept implementation.

Within Hyperledger Fabric, transactions are classified into query requests and invoke requests. Consequently, Table 2 outlines the count of query requests and invoke requests for each function within the *Cyber Insurance Blockchain*. The *"Total Requests"* column represents the cumulative sum of *"Query Requests"* and *"Invoke Requests"*. Table 3 provides a comprehensive summary of the workflow behavior of each Smart Contract function, outlining their input parameters and internal dependencies, while Table 4 complements this overview by mapping the specific data attributes that each function reads or updates. This delineation of attribute access patterns highlights the functional responsibilities of each function and illustrates how state changes propagate through the ledger, offering a structured view of the system's data manipulation processes. Specifically:

- The Smart Contract function *policyholderExists* performs a ledger query to determine the existence of the PH with specific PH_{ID} and IC_{ID}. It returns true if the PH exists and false otherwise. Consequently, the *policyholderExists* Smart Contract function executes a single query request.
- The Smart Contract function *readPolicyholder* conducts a ledger query, returning all attributes of the PH identified by the parameters PH_{ID} and IC_{ID}. The attributes encompass PH_{ID}, IC_{ID}, premium, limit, deductible, risk level, total money risk, reputation, start date, end date, coverages, obligations, controls, and incidents. Consequently, the *readPolicyholder* Smart Contract function entails a single query request.
- The Smart Contract function *analyzeContract* receives parameters PH_{ID} and IC_{ID}, and initiates a ledger query to ascertain the existence of the PH by invoking the Smart Contract function *policyholderExists*. If the PH is confirmed to exist, the function proceeds to invoke the Smart Contract function *readPolicyholder* to retrieve the attributes *"Obligations"* and *"Coverages"* for the specified PH. Consequently, the *analyzeContract* Smart Contract function involves the execution of two query requests. The first query request occurs when calling the *policyholderExists* function, and the second query request transpires when invoking the *readPolicyholder* function.

- The Smart Contract function *updatePolicyholder*, which accepts parameters PH_{ID}, IC_{ID}, riskLevel, and totalmoneyRisk, engages in two key operations. Initially, it invokes the Smart Contract function *readPolicyholder* with parameters PH_{ID} and IC_{ID} to retrieve the attributes of the specified PH. Subsequently, it updates the ledger with the new values of the PH's attributes, specifically riskLevel and totalmoneyRisk. As a result, the *updatePolicyholder* Smart Contract function entails one query request when invoking *readPolicyholder* and one invoke request when updating the ledger with the updated attribute values.
- The Smart Contract function *reportIncident*, receiving parameters PH_{ID}, IC_{ID}, incidentID, incidentName, and indemnification, performs two main actions. Initially, it invokes the Smart Contract function *readPolicyholder* with parameters PH_{ID} and IC_{ID} to retrieve the attributes of the specified PH. Subsequently, it updates the ledger by adding a new incident to the corresponding PH, with the incidentID, incidentName, and indemnification parameters from the *reportIncident* Smart Contract function. Consequently, the *reportIncident* Smart Contract function involves one query request when calling *readPolicyholder* and one invoke request when updating the ledger with the new incident for the PH.
- The Smart Contract function *responseIncident*, taking parameters PH_{ID}, IC_{ID}, and incidentID, initiates by invoking the Smart Contract function *readPolicyholder* with parameters PH_{ID} and IC_{ID} to retrieve the attributes of the specified PH. Upon obtaining the attributes of the requested PH, the algorithm within the *responseIncident* Smart Contract function identifies the reported incident of the PH. Subsequently, it updates the ledger with the new value of the PH's attribute "limit". The calculation of the new limit value follows a two-step process. Firstly, the algorithm checks if the indemnification of the incident is greater than the deductible. If true, the actual indemnification is computed as the difference between the requested indemnification and the deductible, defined as the final indemnification. If false, the final indemnification is set to zero, and there is no need to update the "limit" attribute. Secondly, the algorithm checks if the final indemnification is simultaneously less than both the limit and the cost of the incident. In such cases, the value of the "limit" attribute is reduced by the amount of the final indemnification. The cost of each incident is predefined in the Smart Contract. As a result, the *responseIncident* Smart Contract function involves one query request when calling *readPolicyholder* and one invoke request when updating the ledger with the new value of the PH's "limit" attribute.
- The Smart Contract function *Violate* that takes parameters PH_{ID} and IC_{ID} calls the Smart Contract function *readPolicyholder* with parameters PH_{ID} and IC_{ID} to retrieve the attributes of the requested PH and then to update the ledger with the new value of attribute Reputation, that is reduced by one unit. Therefore, the Smart Contract function *Violate* executes one query request when it calls the Smart Contract function *readPolicyholder* and one invoke request when it updates the ledger with the new value of PH's reputation.
- The Smart Contract function *checkObligations*, accepting parameters PH_{ID} and IC_{ID}, engages in a series of operations. Initially, it queries the ledger to verify the existence of the PH by calling the Smart Contract function *policyholderExists*. If the PH is confirmed to exist, the function proceeds to invoke *readPolicyholder* to retrieve the attributes *"Controls"* and *"Obligations"* for the specified PH. The algorithm subsequently assesses whether the number of the PH's controls and obligations are equal. If inequality is detected, the algorithm invokes the Smart Contract function *Violate* to reduce the PH's reputation. As a result, the *checkObligations* Smart Contract function involves one query request when calling *policyholderExists*, one query request when calling *readPolicyholder*, one query request when invoking *Violate*, and one invoke request when calling *Violate* again.

Table 5 Parameters of Experiments

Network Parameters	Nodes	2, 4, 6, 8, 10
	Batch Size	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500, 2000
Benchmark Parameters	Function	policyholderExists, readPolicyholder, analyzeContract, updatePolicyholder, reportIncident, responseIncident, checkObligations, violate
	# of Transactions	100, 1000, 1500, 2000

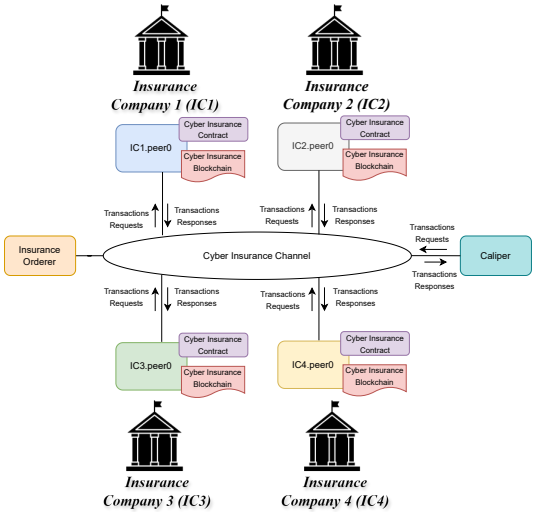

Figure 5 Experiment Environment of Cyber Insurance Blockchain

Figure 5 illustrates the dynamic interaction between the Cyber Insurance Blockchain and Caliper [55] facilitated by the Cyber Insurance Channel. This channel serves as a conduit for communication and transactions among peers within the network of the Cyber Insurance Blockchain. Specifically, the Cyber Insurance Channel comprises peers who have entered into a mutual agreement to collaborate. Their collaboration involves collectively sharing and administering identical copies of the Cyber Insurance Blockchain associated with the Cyber Insurance Channel. When Caliper initiates a transaction request through the Cyber Insurance Channel, it specifies the desired operation to be executed on the Cyber Insurance Blockchain. Upon receiving the transaction request via the Cyber Insurance Channel, the involved peers simulate the desired transaction and submit their transaction responses to the Insurance Orderer. The Insurance Orderer then organizes these transactions into blocks, ensuring consensus among all Cyber Insurance Blockchain network peers. The resulting block is disseminated to all peers in the Cyber Insurance Channel, and each peer verifies its content. The validation response received from this verification process indicates whether the transaction has been successfully executed and committed to the Cyber Insurance Blockchain. As the initiator of the transaction, Caliper collects and stores all transaction responses for record-keeping and auditing purposes. This intricate process ensures the integrity and reliability of transactions within the Cyber Insurance Blockchain network.

6.2 Performance Evaluation of Cyber Insurance Blockchain

To evaluate the performance of the *Cyber Insurance Blockchain*, we assess the following performance indicators: *Throughput*, *Average Latency*, and *CPU Utilization*. Throughput represents the number of successful transactions per second, where a successful transaction is processed, included in a network block, and committed to the Blockchain. Average Latency is the average time interval between transaction submission and ledger commitment. CPU Utilization reflects the load consumed by peers during each experiment. Measurement of these performance metrics is conducted through experiments with varied parameters. Each experiment was conducted five times to record the measurement. Table 5 outlines two categories of parameters: (*i*) Network Parameters, which pertain to the technical infrastructure of the *Cyber Insurance Blockchain*, and (*ii*) Benchmark Parameters, configuring the behavior execution and result collection for the experiments. In more detail, the Network Parameters encompass the following: Nodes (indicating the number of peers in the *Cyber Insurance Blockchain*) and Batch Size (representing the number of transactions per block in the *Cyber Insurance Blockchain*). The Benchmark Parameters include Function (referring to the name of the Smart Contract's function) and

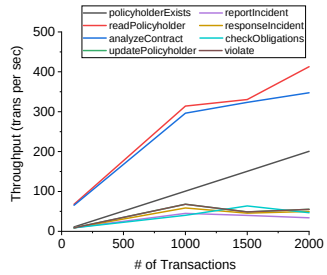

Figure 6 Throughput of Batch Size 2000 and Nodes 2

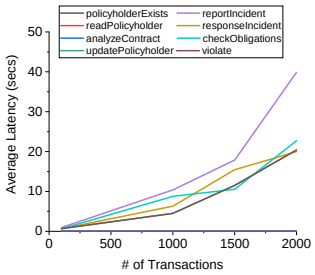

Figure 7 Average Latency of Batch Size 2000 and Nodes 2

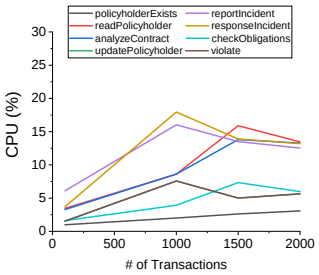

Figure 8 CPU of Batch Size 2000 and Nodes 2

# of Transactions (indicating the quantity of sent transactions). The experiments aim to illustrate how the *Cyber Insurance Blockchain* is influenced by variations in nodes, batch size, function, and the number of sent transactions. Furthermore, this section validates the [**H1**].

6.2.1 Evaluating Throughput

Figure 6 illustrates the Throughput across varying numbers of sent transactions, ranging from 100 to 2000. The results are presented in transactions per second and are derived from experiments conducted on the *Cyber Insurance Blockchain* with two nodes. Each function of the *Cyber Insurance Contract* is represented by a distinct color in the figure. Firstly, the Throughput of the *policyholderExists* function exhibits a predominantly linear increase as the number of sent transactions grows. Also, the observations reveal that the Throughput of the *readPolicyholder*, and *analyzeContract* functions fluctuates between 65 to over 300 transactions per second. Notably, the Throughput of these functions is almost double that of the other functions—*updatePolicyholder*, *reportIncident*, *responseIncident*, *checkObligations*, and *violate*—which ranges from 9 to 55 transactions per second. This discrepancy can be attributed to the fact that, according to Table 2, these three functions (*policyholderExists*, *readPolicyholder*, and *analyzeContract*) solely involve query requests to the *Cyber Insurance Blockchain*, which have a shorter duration compared to invoke requests in other functions. Notably, the Throughput of all functions experiences a faster growth from 100 to 1000 sent transactions, after which it stabilizes beyond 1000 sent transactions. Notice that additional experiments with various batch sizes (e.g., 100, 200, 300, 400, 500, 600, 800, 900, 1000, and 1500) reveal a consistent pattern, suggesting that batch size does not significantly impact the Throughput.

6.2.2 Evaluating Average Latency

Figure 7 shows the Average Latency over different numbers of sent transactions from 100 to 2000. The figure is presented in terms of seconds and is reported for experiments that were conducted in the *Cyber Insurance Blockchain* of 2 nodes. The figure presents the results for all functions of *Cyber Insurance Contract*, each one with a different color. On the one hand, we observe that the Average Latency of *policyholderExists*, *readPolicyholder* and *analyzeContract* functions are steady, fluctuating around 0.15 secs. On the other hand, the functions *updatePolicyholder*, *reportIncident*, *responseIncident*, *checkobligations* and *violate* follow a clear pattern that the values of Average Latency starts between 0.67 secs to 39.88 secs and gradually increasing as the number of sent transactions progress. More specifically, from 100 to 1000 transactions there is an upward trend, with Average Latency rising sharply. Following the point of 1000 transactions, the increasing rate of Average Latency is reduced. The huge difference of Average Latency between the functions *policyholderExists*, *readPolicyholder* and *analyzeContract* and the functions *updatePolicyholder*, *reportIncident*, *reportIncident*, *responseIncident*, *checkViolations* and *violate* is explained because according to Table 2 the first three functions send only query requests to *Cyber Insurance Blockchain* that last less than invoke requests of other functions of *Cyber Insurance Blockchain*. Similarly, as

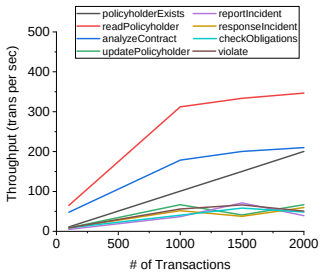

Figure 9 Throughput of Batch Size 2000 and Nodes 4

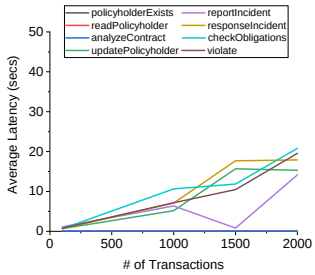

Figure 10 Average Latency of Batch Size 2000 and Nodes 4

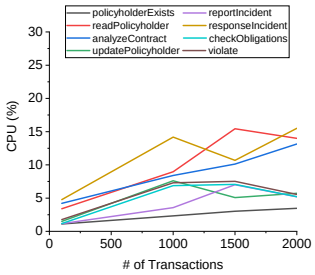

Figure 11 CPU of Batch Size 2000 and Nodes 4

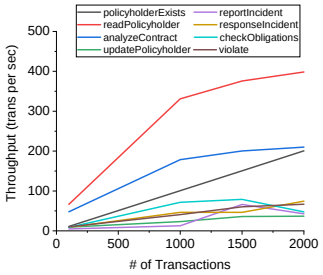

Figure 12 Throughput of Batch Size 2000 and Nodes 6

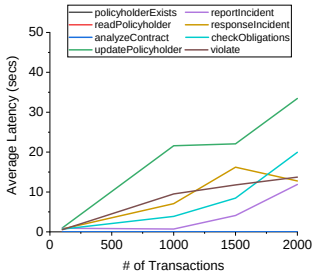

Figure 13 Average Latency of Batch Size 2000 and Nodes 6

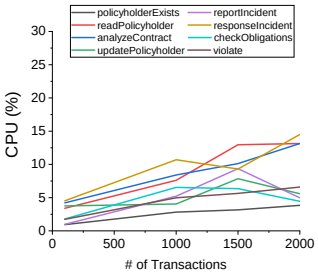

Figure 14 CPU of Batch Size 2000 and Nodes 6

mentioned in Section 6.2.1 experiments were conducted with different batch sizes (e.g. 100, 200, 300, 400, 500, 600, 800, 900, 1000, and 1500) revealed that batch size does not affect the Average Latency.

6.2.3 Evaluating CPU Utilization

Figure 8 depicts CPU utilization across different numbers of sent transactions, ranging from 100 to 2000. The percentages are reported for experiments conducted on the *Cyber Insurance Blockchain* with two nodes. Each function of the *Cyber Insurance Contract* is represented by a distinct color in the figure. The figure illustrates a pronounced increase in CPU utilization percentages up to 1000 sent transactions, beyond which the percentage of CPU utilization stabilizes. Finally, experiments with various batch sizes (e.g., 100, 200, 300, 400, 500, 600, 800, 900, 1000, and 1500), as mentioned in Section 6.2.1 and 6.2.2, exhibit a consistent pattern. This suggests that batch size does not significantly impact CPU utilization.

6.2.4 Scalability

We now outline the outcomes of experiments on scaling the number of nodes. The scalability was assessed using 10, 1000, 1500, and 2000 transactions at two different node configurations: two nodes and four nodes. The figures below portray the metrics of Throughput (Section 6.2.1), Average Latency (Section 6.2.2), and CPU Utilization (Section 6.2.3) with a Batch Size of 2000. The choice to compare varying numbers of nodes (two and four nodes), while maintaining a consistent batch size of 2000, is based on our earlier findings in the preceding sections, where we determined that batch size does not influence the metrics above.

Figure 6, 9, 12, 15, and 18 present the Throughput results for node configurations ranging from two to ten nodes. The Throughput is expressed in transactions per second and corresponds to the same set of smart contract functions used in prior sections. Upon examination of the results across all node topologies, a notable similarity emerges: the smart contract functions again form two distinct groups. The first group

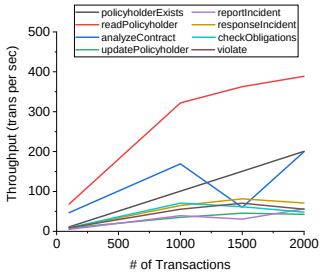

Figure 15 Throughput of Batch Size 2000 and Nodes 8

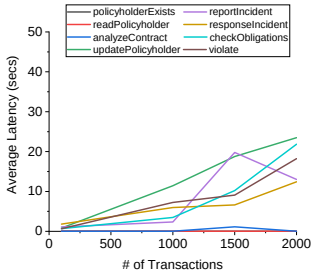

Figure 16 Average Latency of Batch Size 2000 and Nodes 8

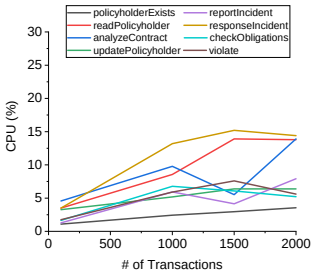

Figure 17 CPU of Batch Size 2000 and Nodes 8

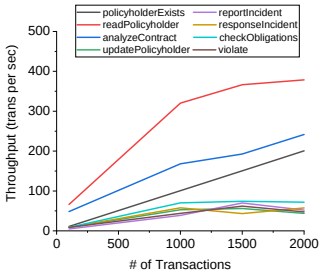

Figure 18 Throughput of Batch Size 2000 and Nodes 10

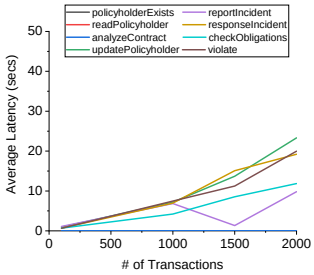

Figure 19 Average Latency of Batch Size 2000 and Nodes 10

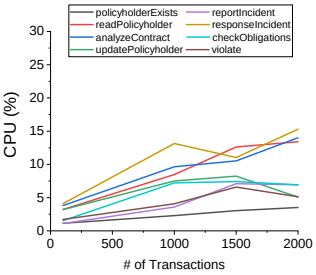

Figure 20 CPU of Batch Size 2000 and Nodes 10

includes the *readPolicyholder*, and *analyzeContract* functions, which issue simple query requests to the Cyber Insurance Blockchain. The second group comprises the *updatePolicyholder*, *reportIncident*, *responseIncident*, *checkOblications*, and *Violate* functions, which generate invoke requests and consequently require additional endorsement and commit operations. As also reflected in earlier findings, the Throughput of the first group is consistently lower than that of the second group, a difference justified by the nature of query versus invoke requests. Across all node configurations (i.e., 2, 4, 6, 8, and 10 nodes) the Throughput of each group follows the same general pattern: it rises steadily between 100 and 2000 transactions and subsequently stabilizes, exhibiting only minor fluctuations. Also, across all node configurations the *policyholderExists* function follows a linear way. Importantly, increasing the number of nodes does not negatively impact the overall Throughput; the Throughput curves across all network sizes remain closely aligned, indicating that the system scales horizontally without introducing performance degradation.

Figure 7, 10, 13, 16, and 19 present the Average Latency for the same node configurations and transaction ranges. The Average Latency values are expressed in milliseconds, and, similarly to the Throughput results, the functions separate into two distinct groups. The first group (the query functions, now including the *policyholderExists* function) consistently demonstrates low Average Latency with only negligible variations across all network sizes. The second group exhibits higher Average Latency due to the greater computational cost of invoke transactions. A prominent similarity across all node configurations is that checkObligation maintains the highest Average Latency, a behavior explained by the high number of internal chaincode requests associated with this function. Across 2, 4, 6, 8, and 10-node networks, the Average Latency increases gradually as the number of transmitted transactions rises, which aligns with expected behavior under higher transaction loads. Importantly, however, the increase in node count does not introduce additional overhead: the Average Latency curves for all node configurations remain almost identical, differing

only by marginal, statistically insignificant values. This demonstrates that the endorsement and commit phases of transaction processing are not adversely affected by a larger peer network.

Figure 8, 11, 14, 17, and 20 depict CPU utilization for all node configurations. In contrast to the clear grouping observed in Throughput and Average Latency, the CPU usage patterns do not form distinct categories based on smart contract function type. Instead, CPU utilization consistently increases up to 2000 transactions and then continues rising more steadily for all workloads. Notably, the CPU usage trends for 2, 4, 6, 8, and 10 nodes exhibit nearly identical behavior. No configuration shows abnormal spikes, saturation, or instability. As with earlier experiments, the results confirm that batch size does not influence CPU utilization. More importantly, the stability in CPU usage across different node counts indicates that adding nodes does not generate additional processing overhead significant enough to impact system performance.

Furthermore, the results of the CPU consumption of the PINSA's Cyber Insurance Blockchain present that even under the maximum workload of 2000 transactions, the smart-contract functions in Hyperledger Fabric consume only about 3–15% CPU per function, with consistent behavior as the network scales from 2 to 10 nodes. Since all nodes were co-located on a single machine, these results represent a worst-case scenario for consolidated resources; in a real distributed deployment, per-node CPU usage would likely be lower. The stable and moderate CPU footprint indicates that Fabric's consensus, validation, and execution processes run efficiently on commodity hardware, requiring only modest resources and resulting in low operational and maintenance overhead. Thus, the PINSA's blockchain layer demonstrates predictable performance and modest infrastructure demands, supporting scalable deployment without the need for specialized hardware.

Finally, we also measure the smart contract execution time. For that experiment, we perform 9 testing rounds executing 200 transactions targeting a specific function of the PINSA smart contract. During **Round 1 – CreatePolicyholder**, every worker invokes the CreatePolicyholder(), generating new ledger entries by assembling PH attributes and storing them. Next, in **Round 2 – ReadPolicyholder** retrieves a specific PH's record from the ledger using its composite key. Also, in **Round 3 – UpdatePolicyholder**, each transaction updates the Risklevel and Totalmoneyrisk fields of an existing PH. **Round 4 – AnalyzeContract** extracts and returns only the Coverage and Obligation components of a PH, testing read-only logic. **Round 5 – CheckObligations**, compares obligations with actual security posture to reduce PH's reputation committing the update to the ledger in case of violation. **Round 6 – Violate**, calls the Violate function to decrement the reputation field and store the modified record. Moreover, **Round 7 – ReportIncident** adds a new incident to the PH's incidents array, testing array operations and manipulation. **Round 8 – ResponseIncident** calculates indemnification and deductibles, adjusts the remaining limit, updates incident status, and writes the new state back to the ledger. Finally, **Round 9 – DeletePolicyholder** iterates through keys and removes each one measuring the cost of ledger cleanup operations. In network configuration with 2, 4, 6, 8, 10 nodes, 10 workers were equally distributed to each node to continuously sign, submit, and endorse transactions under different nodes, which affects endorsement pathways, peer selection, and ledger commit performance. By distributing transactions across multiple nodes' certificates, the benchmark realistically reflects how complex Fabric consortium behaves under load.

Table 6 PINSA's smart contract execution time

Number of Organizations (IC)	Execution Time (minutes)
2	3.8
4	4.7
6	5.8
8	7
10	9

Table 6 presents that execution time increases as the number of nodes (i.e., IC) in the Fabric network grows. This is primarily due to the additional peers required to endorse transactions under multi-organization endorsement policies, the extra network communication overhead from gRPC connections and TLS handshakes, and the increased workload on orderers and peers during

commit phases, as transactions must be validated and committed across multiple organizations. Scaling from 2 to 4 organizations increases execution time by approximately 22%; from 4 to 6 organizations, by 23%; from 6 to 8 organizations, by 21%; and from 8 to 10 organizations, by 29%. These increments are roughly linear but can vary depending on network resources, endorsement policy complexity, and the type of chaincode operation. Function sensitivity analysis indicates that write-heavy rounds (i.e., CreatePolicyholder, UpdatePolicyholder, ReportIncident, ResponseIncident, Violate, DeletePolicyholder) experience the most significant latency increase as organizations scale, because more peers must endorse and commit each transaction. In contrast, read-heavy rounds (i.e., ReadPolicyholder, AnalyzeContract, and CheckObligations) are less affected, as reads can often be served locally by a single peer without requiring full multi-organization endorsement.

6.3 Performance Evaluation of SSI Blockchain

To evaluate the performance of the SSI Blockchain, our attention is directed toward two key metrics: the Average Time and the Frequency of issuing a verifiable credential and verifying a presentation. Measurement of these key metrics is conducted through experiments with varied parameters. Each experiment was conducted five times to record the measurement. In our implementation, three Aries agents are employed: an Aries agent for NCSA, an Aries agent for IC, and an Aries agent for PH. The Hyperledger Aries credential is configured with the following fundamental attributes:

1. ***PH Name:*** Legal name of PH (e.g., Cars Dealer Inc.)
2. ***Date:*** Establishment date of the entity (e.g., *2020-11-27*)
3. ***Number of Breaches:*** Total incidents of data breaches for PH (e.g., *3*)
4. ***Timestamp:*** Creation timestamp of the verifiable credential (e.g., *2023-10-21 11:46:57.792711*)
5. ***Risk Factor:*** Risk level associated with PH (e.g., *100*)
6. ***Revenue:*** Total annual earnings of PH (e.g., *732,900 €*)
7. ***Industry:*** Industry to which PH belongs (e.g., Commerce)
8. ***Certified:*** Indication of whether PH holds any cybersecurity certification (e.g., ISO/IEC 27001)
9. ***Legal Responsible:*** Individual at PH responsible for legal matters (e.g., John Doe)
10. ***Address:*** Headquarters address of PH (e.g., 123 Maple Street, Anytown, PA 17101)

The experiments were conducted by simultaneously submitting bulk requests (5, 10, 50, 100, 200, and 300 requests) with a specific number of attributes (5, 10, and 20 attributes per credential/presentation). In the experiments with five attributes, the following attributes were considered: PH's Name, Date, Number of Breaches, Timestamp, and Risk Factor. For experiments involving ten attributes, the previously mentioned five attributes were included along with Revenue, Industry, Certified, Legal Responsible, and Address. The experiments with 20 attributes extended the ten attributes to include numerical attributes with values randomly generated between 0 and 100. Each experiment's measurement was recorded as the average value derived from three executions of the experiment with the same parameter values. Furthermore, this section validates the [**H1**].

6.3.1 Scrutinizing the Average Time

The experiments on the Average Time required for issuing a VC are illustrated in Figure 21 and pertain to the time interval commencing from the *Send offer-credential* step of NCSA, concluding at the *Receive of Credential-Ack* step of NCSA. To elaborate further, the Average Time required for issuing a credential with 5 attributes is 0.4125 seconds, 0.365 seconds, 0.3325 seconds, 0.3425 seconds, 0.3475 seconds, and 0.3433 seconds for the issuance of 5, 10, 50, 100, 200, and 300 credentials, respectively. Additionally, the average time required for issuing a credential with 10 attributes is 0.4033 seconds, 0.3533 seconds, 0.32 seconds, 0.34 seconds, 0.3466 seconds, and 0.35 seconds for the issuance of 5, 10, 50, 100, 200, and 300 credentials, respectively. Furthermore, the average time required for issuing a credential with 20 attributes is 0.4733 seconds, 0.4 seconds, 0.3533 seconds,

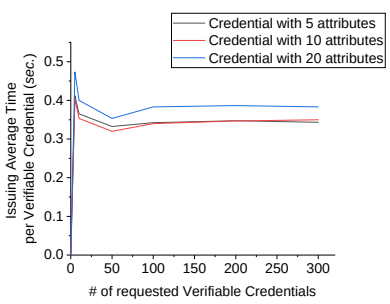

Figure 21 Average Time of Credential Issuance

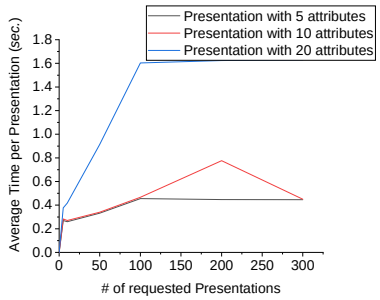

Figure 22 Average Time of Presentation Verification (satisfying predicate)

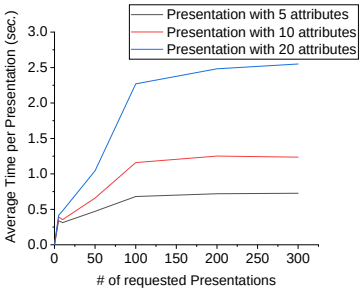

Figure 23 Average Time of Presentation Verification (no satisfying predicate)

0.3833 seconds, 0.3866 seconds, and 0.3833 seconds for the issuance of 5, 10, 50, 100, 200, and 300 credentials, respectively. Notably, the average time necessary for issuing a VC tends to increase with the growing number of requested attributes. Additionally, fluctuations in the average time required for issuing a credential are observed, irrespective of the number of requested attributes. This time decreases as the number of requested credentials increases from 5 to 50, rises from 50 to 100, and remains relatively constant when the number exceeds 100.

The experiments regarding the Average Time required to verify a presentation are illustrated in Figure 22 and 23, denoting the timeframe from the *Send request presentation* step of IC to the *Receive ack-presentation* step of PH. In Figure 22, the experiments involve PH responding to IC by sending a verifiable presentation that satisfies the requested predicate specified by IC. The predicate is a Boolean condition, requiring the "Risk Factor" attribute to be equal to or less than 50. To elaborate, the average time needed to verify a presentation with 5 attributes is 0.27 seconds, 0.26 seconds, 0.3325 seconds, 0.455 seconds, 0.4475 seconds, and 0.4466 seconds for the verification of 5, 10, 50, 100, 200, and 300 presentations, respectively. Similarly, the average time required to verify a presentation with 10 attributes is 0.2833 seconds, 0.27 seconds, 0.34 seconds, 0.4667 seconds, 0.7766 seconds, and 0.45 seconds for the verification of 5, 10, 50, 100, 200, and 300 presentations, respectively. Furthermore, the average time needed to verify a presentation with 20 attributes is 0.3766 seconds, 0.4166 seconds, 0.9133 seconds, 1.6033 seconds, 1.6233 seconds, and 1.63 seconds for the verification of 5, 10, 50, 100, 200, and 300 presentations, respectively. In Figure 23, the experiments involve PH responding to IC with a message indicating that PH cannot find a presentation satisfying the predicate requested by IC. The requested predicate remains a Boolean condition, specifying that the "Risk Factor" attribute is equal to or less than 50. Specifically, the average time needed to verify a presentation with 5 attributes is 0.34 seconds, 0.312 seconds, 0.472 seconds, 0.68 seconds, 0.72 seconds, and 0.726 seconds to verify 5, 10, 50, 100, 200, and 300 presentations, respectively. Additionally, the average time required to verify a presentation with 10 attributes is 0.392 seconds, 0.354 seconds, 0.658 seconds, 1.16 seconds, 1.252 seconds, and 1.236 seconds to verify 5, 10, 50, 100, 200, and 300 presentations, respectively. Moreover, the average time needed to verify a presentation with 20 attributes is 0.414 seconds, 0.478 seconds, 1.044 seconds, 2.27 seconds, 2.482 seconds, and 2.55 seconds to verify 5, 10, 50, 100, 200, and 300 presentations, respectively.

When comparing Figure 22 and 23, it is evident that the average time required to verify a presentation increases with the growing number of requested attributes. Generally, the trend is an increase in average time as the number of requested presentations rises. Specifically, the average time shows a slight increase up to 100 requested presentations, which remains relatively stable for requests exceeding 100 presentations. Additionally, the average time aligns with the number of attributes. The blue line, representing the average time needed to verify presentations

with 20 attributes, surpasses the red line (10 attributes), and the red line, in turn, is superior to the black line (5 attributes). A notable distinction between Figure 22 and 23 is that the average time needed to verify a presentation when predicates are not satisfied is greater than the time needed for presentations where predicates are satisfied. The values on the y-axis for each line in Figure 23 are higher than the corresponding values in Figure 22. This discrepancy is expected, as the experiments in Figure 23 involve PH taking additional time to search for its wallet and ensure that no presentation satisfies the requested attributes and predicates. Overall, the average time need to verify a presentation in SSI Blockchain (Figure 22 and 23) is bigger than the average time needed to issue a VC in SSI Blockchain (Figure 21).

We now measure credential issuance latency. Latency, in comparison to Average Time credential issuance, which measures only the time required by the issuer to issue the VCs, records the exact time each VC takes from the moment the issuer triggers the send operation until the holder completes the protocol for that specific VC. This per-credential tracking exposes variations caused by factors such as network delays, ACA-Py's internal message scheduling, asynchronous protocol handling, and concurrency-induced queuing inside the agent processes. We measured latency using the NCSA agent (issuer) and the PH agent (receiver). NCSA generates VC with varying numbers of attributes (i.e., 5, 10, 20) and issues them to the PH. Each VC is individually tracked with timestamps recorded at issuance and upon acknowledgment by the receiver, allowing precise measurement of end-to-end latency. Multiple tests were conducted with varying numbers of credentials (i.e., 5, 10, 50, 100, 200, 300) to examine `PINSA`'s performance under increasing load. This evaluation demonstrates how per-credential latency measurement captures the effects of attribute complexity, network conditions, and agent processing dynamics, providing insights into real-world credential issuance behavior that overall throughput metrics cannot reveal.

Figure 24 presents the latency trends in relation to the number of credentials and attribute size, indicating the following patterns: For 5 attributes per VC, the Latency starts at 1.72 seconds for 5 VCs, increases to 3.2 seconds for 10 VCs,

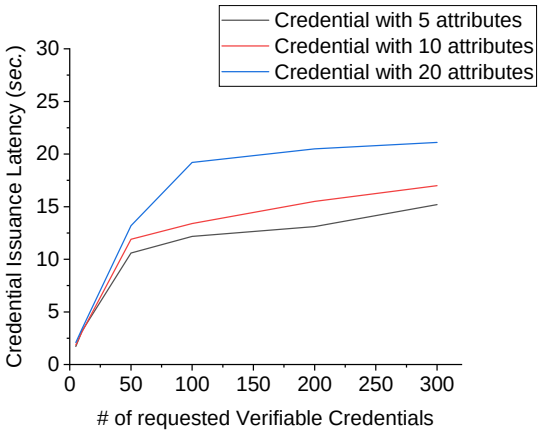

Figure 24 SSI Credential Issuance Latency

and grows steadily to 10.6 seconds for 50, 12.17 seconds for 100, 13.1 seconds for 200, and 15.2 seconds for 300 VCs accordingly. This shows that while small batches scale almost linearly, the per-credential latency growth slows slightly at higher volumes, suggesting some efficiency in batching or asynchronous processing. Next, for 10 attributes per VC, the Latency is slightly higher, starting at 1.85 seconds for 5 and 3.1 seconds for 10 VCs accordingly. For larger loads, latency increases significantly, with values of 11.9 seconds for 50, 13.4 seconds for 100, 15.5 seconds for 200, and 17 seconds for 300 VCs accordingly. The additional attributes per credential add measurable overhead, indicating that serialization and processing time scale with attribute count. Finally, for 20 attributes per VC the Latency becomes more pronounced, starting at 2.08 seconds for 5 credentials, 3.4 seconds for 10, 13.2 seconds for 50, 19.2 seconds for 100, 20.5 seconds for 200, and 21.1 seconds for 300 VCs accordingly. The jump is especially notable between 50 and 100 VCs, showing that larger VCs exacerbate queuing and processing delays under higher loads.

Overall, we observe the following two insights: (*i*) The number of VCs issued is the dominant factor affecting latency, with larger batches leading to higher end-to-end times, and (*ii*) VC's complexity (attribute count) adds additional latency, particularly under heavier loads.

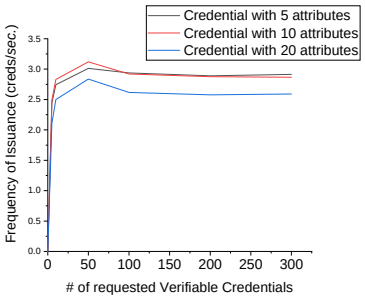

Figure 25 Frequency of Credential Issuance

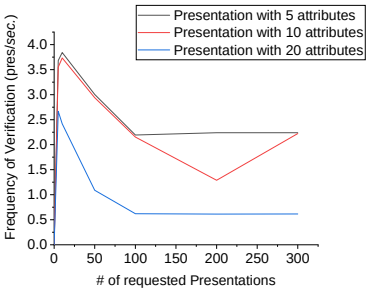

Figure 26 Frequency of Presentation Verification (satisfying predicate)

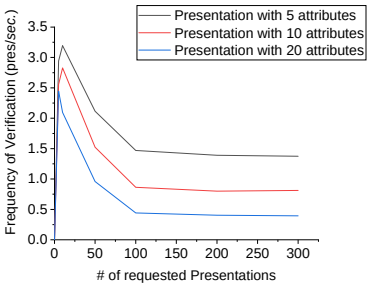

Figure 27 Frequency of Presentation Verification (no satisfying predicate)

6.3.2 Evaluating Issuance and Verification Frequency

The experiments focusing on the issuance rate of VCs, succinctly described as the frequency of issuance, are illustrated in Figure 25. The time frame for these experiments spans from the *Send offer-credential* step of NCSA to the *Receive Credential-Ack* step of NCSA. Specifically, the issuance rate of VCs (with 5 attributes) per second is as follows: 2.4425 creds/sec, 2.7425 creds/sec, 3.0125 creds/sec, 2.9375 creds/sec, 2.8900 creds/sec, and 2.9133 creds/sec for requests of 5, 10, 50, 100, 200, and 300 credentials, respectively. Additionally, the issuance rate of VCs (with 10 attributes) per second is 2.4966 creds/sec, 2.8300 creds/sec, 3.1200 creds/sec, 2.9200 creds/sec, 2.8766 creds/sec, and 2.8666 creds/sec for requests of 5, 10, 50, 100, 200, and 300 credentials, respectively. Moreover, the issuance rate of VCs (with 20 attributes) per second is 2.1066 creds/sec, 2.5000 creds/sec, 2.8366 creds/sec, 2.6166 creds/sec, 2.5766 creds/sec, and 2.5900 creds/sec for requests of 5, 10, 50, 100, 200, and 300 credentials, respectively. Observing the results, it becomes apparent that the frequency of issuance increases proportionally with the number of requested attributes for the credential. Furthermore, there is an observable fluctuation in the issuance rate, independent of the number of requested attributes. The rate experiences a sharp increase when the number of requested credentials goes from 5 to 50, followed by a slight decrease from 50 to 100, and stabilizes when the number of requested credentials surpasses 100.

The experiments focusing on the frequency of verification, measured as the number of presentations verified per second, are illustrated in Figure 26 and 27. These experiments span from the *Send request presentation* step of IC to the *Receive ack-presentation* step of PH. Figure 26 presents experiments where PH responds to IC by sending a verifiable presentation that satisfies the requested predicate. The predicate is a Boolean condition stipulating that the "Risk Factor" attribute is equal to or less than 50. Specifically, the number of presentations (with 5 attributes) verified per second is 3.6825 pres/sec, 3.8400 pres/sec, 3.0050 pres/sec, 2.1925 pres/sec, 2.2400 pres/sec, and 2.2400 pres/sec for 5, 10, 50, 100, 200, and 300 presentations, respectively. Additionally, the number of presentations (with 10 attributes) verified per second is 3.5500 pres/sec, 3.7333 pres/sec, 2.9400 pres/sec, 2.1533 pres/sec, 1.2900 pres/sec, and 2.2233 pres/sec for 5, 10, 50, 100, 200, and 300 presentations, respectively. Moreover, the number of presentations (with 20 attributes) verified per second is 2.6666 pres/sec, 2.4166 pres/sec, 1.0900 pres/sec, 0.6200 pres/sec, 0.6133 pres/sec, and 0.6166 pres/sec for 5, 10, 50, 100, 200, and 300 presentations, respectively.

Figure 27, on the other hand, depicts experiments where PH responds to IC with a message indicating that PH cannot find a presentation satisfying the requested predicate. The predicate remains a Boolean condition requiring the "Risk Factor" attribute to be equal to or less than 50. The number of presentations (with 5 attributes) verified per second is 2.944 pres/sec, 3.194 pres/sec, 2.116 pres/sec, 1.47 pres/sec, 1.39 pres/sec, and 1.374 pres/sec for 5, 10, 50, 100, 200, and 300 presentations, respectively. Additionally, the number of presentations (with 10 attributes) verified per second is 2.558 pres/sec, 2.828 pres/sec, 1.524 pres/sec, 0.864 pres/sec, 0.8

pres/sec, and 0.812 pres/sec for 5, 10, 50, 100, 200, and 300 presentations, respectively. Moreover, the number of presentations (with 20 attributes) verified per second is 2.446 pres/sec, 2.096 pres/sec, 0.958 pres/sec, 0.442 pres/sec, 0.404 pres/sec, and 0.394 pres/sec for 5, 10, 50, 100, 200, and 300 presentations, respectively.

When scrutinizing Figure 26 and 27, it is evident that the frequency of verification decreases with the increasing number of requested attributes for the presentation. Additionally, fluctuations in the verification frequency are observed irrespective of the number of requested presentations. The frequency sharply increases when the number of requested presentations goes from 0 to 5, decreases slightly from 5 to 50, and remains steady when the number of requested presentations exceeds 100. Moreover, the frequency is inversely proportional to the number of attributes. The black line, representing the frequency needed to verify presentations with 5 attributes, surpasses the red line (10 attributes), and the red line, in turn, is superior to the blue line (20 attributes). An important observation from Figure 26 and 27 is that the values on the y-axis for each line in Figure 26 are greater than the values on the y-axis of the corresponding line in Figure 27. This difference is expected since PH requires more time to search for its wallet and confirm the absence of any presentation satisfying the requested attributes and predicates in the experiments presented in Figure 27.

6.4 Security, Privacy and Legal Analysis

This section thoroughly analyzes the security, privacy, and legal aspects inherent in the PINSA framework. This evaluation leverages the distinctive properties associated with DIDs and VCs. The analysis aims to provide valuable insights into the effectiveness of the proposed solution in countering security attacks and ensuring privacy and legal compliance throughout data sharing. Furthermore, it assesses the subsequent access to shared data by Smart Contracts with an oracle, shedding light on the framework's overall security posture and its commitment to safeguarding user privacy. Furthermore, this section validates the **[H2]** and **[H3]**.

To achieve ***GDPR Compliance*** [58], PINSA implements privacy-by-design and privacy-by-default via DIDs and VCs, allowing PHs to retain full control over their personal and cybersecurity-relevant data. Only pseudonymous identifiers and cryptographic proofs are stored on-chain, reducing re-identification risk in line with the GDPR definition of pseudonymization (GDPR Article 4(5)), while supporting immutable audit trails. Selective disclosure and data minimization (GDPR Article 5(1)(c)) prevent unnecessary exposure of sensitive information, balancing privacy with operational needs. Furthermore, this design ensures that Smart Contracts and oracles access only the data strictly required for policy verification and claims processing,reinforcing confidentiality and meeting security-of-processing requirements (GDPR Article 32).

The ***eIDAS Regulation*** [59] and its upcoming ***eIDAS 2.0 framework*** establish a legal and technical standard for electronic identification, authentication, and trust services across the EU. These regulations are critical for ensuring that digital identities are legally recognized, interoperable, and secure in cross-border transactions. Within PINSA, VCs can function as legally recognized attestations of identity and attributes, supporting authentication, non-repudiation, and cross-border interoperability, as defined in eIDAS Articles 6-12 (eIDAS schemes) and Articles 25-40 (electronic signatures and seals). The cryptographic verification of credentials ensures message integrity and authenticity, strengthening defenses against identity-based attacks such as Sybil and Man-in-the-Middle attacks. Integration with the emerging European Digital Identity (EUDI) Wallet further enables secure cross-border interactions while maintaining compliance with regulated trust services (eIDAS Articles 41-50).

Smart Contract–based claims handling, combined with selective-disclosure KYC processes, ensures transparency, auditability, and adherence to Solvency II [60], IDD [61], and EIOPA guidance [62], proving an existing ***Insurance Regulation Compliance***. AML obligations are satisfied without excessive collection of personal identifiers, reducing the risk of data leakage. Immutable ledger entries, verifiable credentials, and cryptographically secured communications provide an

additional layer of security, supporting supervisory oversight and regulatory audits.

Overall, these compliance mechanisms are tightly coupled with the framework's technical security features, enhancing the resilience of the cyber insurance system against both operational and adversarial risks.

Data Minimization in cyber insurance plays a crucial role in preserving the privacy of PHs and enhancing data security. ICs should selectively collect and retain only the essential data necessary for underwriting policies, assessing risk, and processing claims. This approach minimizes the risk of exposing sensitive information, thereby mitigating the potential consequences of data breaches and unauthorized access. By adhering to data minimization principles, ICs can strike a balance between acquiring essential data for evaluating cyber risks and upholding the confidentiality of PHs, fostering trust and confidence in their cybersecurity protocols. To achieve this objective, `PINSA` employs the decentralized identity framework Hyperledger Aries, which utilizes various mechanisms for effective data minimization. Firstly, it facilitates selective disclosure, allowing PHs to disclose only the necessary data for a specific transaction, avoiding unnecessary exposure of personal information. Additionally, Hyperledger Aries promotes the use of VCs, enabling PHs to present evidence of specific attributes without revealing the underlying data. This approach reduces reliance on storing and transmitting extensive personal data. In the context of `PINSA`, Hyperledger Aries empowers PHs with enhanced control over their personal information by implementing selective disclosure and VCss. This not only facilitates the necessary data for identity verification and trust establishment but also effectively reduces the amount of disclosed information, aligning with data minimization principles.

Replay Attacks aim to deceive by repeatedly transmitting authenticated messages, such as requests for policy information or claims submissions. The goal is to fraudulently manipulate cyber insurance contracts or exploit vulnerabilities within the system. To counter the risk of replay attacks, `PINSA` leverages the inherent features of Hyperledger Fabric, including immutability and transparency, coupled with Smart Contract automation. Blockchain technology ensures transactional integrity, presenting significant challenges for potential attackers attempting to manipulate or duplicate messages. Smart Contracts, enforcing predefined logic and insurance regulations, mitigate the risk of unauthorized alterations or the submission of duplicate claims. Additionally, `PINSA` incorporates Hyperledger Aries, a framework with multiple defensive measures against replay attacks. It employs distinct message identifiers and timestamps within communication protocols. These identifiers and timestamps enable Hyperledger Aries to detect and reject replicated messages by recognizing duplicate characteristics or outdated timestamps. Furthermore, cryptographic techniques like digital signatures and message authentication codes guarantee the integrity and authenticity of exchanged messages. These cryptographic mechanisms authenticate the message source and ensure integrity, thwarting adversaries' attempts to replay or alter messages undetected. By implementing these strategies, Hyperledger Aries significantly enhances the security of communication channels, effectively mitigating the risks associated with replay attacks.

Man-in-the-Middle Attacks enable attackers to position themselves between parties, intercept messages, and potentially alter or inject malicious content to deceive and exploit the system. However, `PINSA` integrates Blockchain, Smart Contracts, and VCs technologies to provide robust defenses against Man-in-the-Middle Attacks. On the one hand, the inherent immutability of Hyperledger Fabric ensures the integrity of recorded transactions, establishing a reliable and trustworthy communication repository. The Raft consensus mechanism for communication between nodes utilizes encryption and digital signatures to secure communication between nodes, ensuring that messages are authenticated and tamper-proof. Smart Contracts are designed to uphold predetermined regulations and principles of insurance agreements, creating a robust barrier against any attempts by malicious actors to manipulate or modify contractual terms. On the other hand, VCs through Hyperledger Aries provide cryptographic evidence of the genuineness and unaltered state of messages, enabling parties to authenticate the legitimacy of communications and detect

any instances of tampering. Collectively, these technologies enhance the security and integrity of cyber insurance systems in the online domain, effectively mitigating potential hazards associated with Man-in-the-Middle attacks.

Meet-in-the-Middle Attacks are cryptographic attacks that seek to exploit vulnerabilities by dividing the encryption process into two parts and searching for a matching pair in the middle. However, `PINSA` integrates Blockchain and SSI technologies to defend against such attacks. On the one hand, the immutability and transparency of Hyperledger Fabric ensure that recorded transactions remain impervious, making it challenging for attackers to corrupt or alter data in transit. Implementing Smart Contracts adds an extra layer of security by enforcing predetermined rules and logic, ensuring transactions conform to agreed-upon conditions, and thwarting unauthorized modifications or entries. On the other hand, VCs by Hyperledger Aries, relying on cryptographic techniques such as zero-knowledge proofs, provide evidence of authenticity and integrity. This creates a significant obstacle for potential attackers attempting to manipulate or counterfeit credentials without detection. The utilization of VCs effectively mitigates the potential for a successful Meet-in-the-Middle attack in online cyber insurance systems.

In ***Session Hijacking Attacks***, adversaries intercept and gain control of users' (e.g., PHs) sessions within an online system (e.g., cyber insurance claim). `PINSA` aims to prevent this attack by integrating Hyperledger Fabric and Hyperledger Aries technologies. First, the immutability and transparency of Hyperledger Fabric provide a secure foundation for recording and verifying transactions. By leveraging Blockchain, online cyber insurance systems can establish tamper-resistant session management mechanisms. Smart Contracts, operating on the Blockchain, enforce strict access controls and authentication protocols, ensuring that only authorized participants can interact with the system and protecting against unauthorized session hijacking attempts. Second, Hyperledger Fabric, offering its VCs, plays a vital role in session security by providing cryptographic proof of authentication and integrity. Using VCs, `PINSA` can ensure the validity of PHs' identities and session attributes, making it challenging for attackers to forge or manipulate session credentials.

Sybil Attacks are attacks where an attacker creates multiple fake identities or nodes to gain control or manipulate a system. `PINSA` strategically leverages the advantages of Hyperledger Fabric and Hyperledger Aries to defend against this attack. Firstly, Hyperledger Fabric's decentralized and transparent nature provides an effective defense against Sybil attacks. `PINSA` utilizes Raft within the Cyber Insurance Blockchain, relying on a majority vote for leader election and log replication. To mitigate Sybil attacks, Raft requires the election of a legitimate leader by a majority of nodes, making it challenging for a minority of malicious nodes to control the system. Smart Contracts operating on the Blockchain enforce predefined rules and logic, including identity verification and registration processes. These contracts implement checks to ensure that each identity or node is associated with a unique, verifiable credential, reducing the likelihood of Sybil attacks. Furthermore, Hyperledger Aries VCs offer a means to authenticate and verify the legitimacy of ICs within `PINSA`. By leveraging these credentials, `PINSA` establishes trust that requires proof of identity before granting access or participation.

Denial of Service Attacks pose a significant threat to online cyber insurance systems. `PINSA`, leveraging Hyperledger Fabric, utilizes the technology's decentralized nature, making it challenging for attackers to orchestrate a Denial of Service attack. The distributed structure of the Blockchain network ensures the absence of a central point of failure. In our implementation, the consensus mechanism employed by the Cyber Insurance Blockchain, Raft, prevents malicious nodes from overwhelming the network with excessive traffic. Nodes can employ timeouts and leader election mechanisms to recover from unresponsive nodes. Furthermore, Smart Contracts operating on the Blockchain enforce rules and logic related to access control and transaction processing. Robust access management mechanisms implemented by Smart Contracts help mitigate the impact of Denial of Service attacks by restricting the flow of requests, ensuring that only legitimate users are granted access to the system. In addition, Hyperledger Aries VCs add an extra layer of security in mitigating Denial of Service attacks. By requiring

users to present valid and verifiable credentials, PINSA can effectively filter out malicious traffic and requests, reducing the impact of potential Denial of Service attacks.

In ***Eclipse Attacks***, adversaries strategically isolate a node by manipulating its network communication, thereby obstructing its involvement in leader election or communication with other nodes. PINSA, leveraging Raft as its consensus mechanism, enables nodes to employ randomized election timeouts. This intentional variation diminishes the probability of multiple nodes competing for leadership simultaneously, consequently mitigating the success rate of eclipse attacks.

Adversaries may endeavor to ***Modify the Data in Log Entries***, introducing inconsistencies into the replicated logs. However, the PINSA Cyber Insurance Blockchain is fortified with Raft, employing a blend of term numbers and voting mechanisms to uphold the uniformity of log entries. If a log entry fails to secure a majority of votes, it is deemed uncommitted, thereby thwarting unauthorized data modifications.

Identity theft involves an adversary impersonating a legitimate user by stealing or spoofing their digital identity credentials. PINSA via Hyperledger Aries uses DID and VC, which eliminate centralized identity stores that adversaries can target. Each identity is backed by cryptographic key pairs that are controlled solely by the identity holder (i.e., PH), and the associated public keys are anchored on a tamper-evident ledger. Since private keys never leave the holder's wallet, stealing an identity would require compromising the user's local cryptographic wallet, which is significantly harder than breaching a centralized database.

Forgery refers to the creation or alteration of identity documents, credentials, or claims to deceive a verifying party. PINSA's VC issued in a Hyperledger Aries ecosystem are cryptographically signed by trusted issuers (i.e., NCSA), and their signatures can be verified independently against public keys stored on the underlying blockchain or ledger. This prevents adversaries from forging credentials, altering attributes, or creating counterfeit identities because any tampering invalidates the cryptographic signature. Additionally, Zero-Knowledge Proof capabilities in Aries allow holders to prove attributes without exposing raw credential data, preventing attackers from gleaning sensitive information to replicate or forge identity attributes. Hyperledger Fabric complements this by ensuring that any credential-related transactions recorded on the ledger are immutable and traceable.

Contract manipulation occurs when an adversary attempts to alter the logic, parameters, or execution outcomes of a digital or smart contract for personal gain. PINSA's Hyperledger Fabric provides an immutable, permissioned blockchain with fine-grained access control, endorsement policies, and deterministic chaincode execution. This ensures that any contract (smart contract or business logic) cannot be unilaterally modified by a malicious party since changes must satisfy predefined endorsement rules involving multiple trusted peers. Since all transactions are cryptographically linked and validated through Hyperledger Fabric's consensus, attempts to alter a contract's terms, execution history, or state transitions will be immediately detected. Hyperledger Aries can additionally bind identities and VC to Fabric transactions, ensuring that only authenticated and authorized actors can invoke or update contracts, further preventing manipulation by unauthorized users.

Furthermore, we perform empirical experiments to quantitatively prove the security claims coming from the SSI features. Particularly, we prove Unlinkability and selective disclosure, both use Zero-Knowledge Proof technique, are active in our implementation.

To evaluate PINSA's **selective disclosure** capabilities, we tested whether the holder (i.e., PH) can successfully prove credential validity while revealing only a minimal subset of attributes. The issuer (i.e., NCSA) defined a credential schema with 10 attributes, including several sensitive fields. VCs were issued to the holder (i.e., PH), and the verifier (i.e., IC) requested proofs disclosing only 2–5 attributes depending on the scenario. During proof generation, the holder produced Zero-Knowledge Proofs that cryptographically attest that all credential attributes originate from an unmodified, issuer-signed credential, while exposing only the specifically requested subset. The verifier must therefore (i) confirm the credential is valid and (ii) learn nothing about undisclosed attributes. Across different proof request sizes (2/10,

Table 7 PINSA's selective disclosure capability

Issued Credentials	Verified Credentials	Failed Presentations	Success Rate	No. Attributes	No. Revealed Attributes
5	5	0	100%	10	2/10
10	10	0	100%	10	3/10
50	50	0	100%	10	5/10
100	100	0	100%	10	3/10
200	200	0	100%	10	5/10
300	300	0	100%	10	4/10

Table 8 PINSA's Unlinkability capability

Issued Credentials	Verified Credentials	Failed Presentations	No. Unlikable VCs
5	5	0	5
10	10	0	10
50	50	0	50
100	100	0	100
200	200	0	200
300	300	0	300

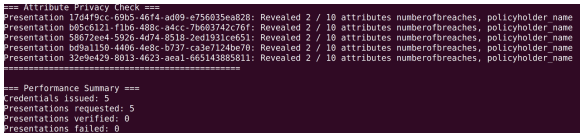

Figure 28 Selective disclosure for 5 VCs revealing attributes 2 out of 10 attributes

3/10, 4/10, and 5/10 attributes), all proofs were successfully verified. No undisclosed attributes appeared in any Aries *present-proof* messages, and proof sizes remained constant for hidden attributes, confirming that the verifier could not infer their values. Table 7 demonstrates that PINSA enforces selective disclosure correctly since the verifier consistently validates all proofs, while only the intended subset of attributes becomes revealed. Because Zero-Knowledge Proofs hide all other values and maintain a constant proof structure, the verifier cannot infer undisclosed fields, thereby confirming strong privacy preservation. This is illustrated in Figure 28, which presents the results for 5 attributes.

To evaluate PINSA's **unlinkability**, we tested whether repeated presentations of the same VC could be correlated by the verifier. The issuer (i.e., NCSA) issued VCs containing 10 attributes (i.e., policyholder name, policy date, number of breaches, risk factor, timestamp) stored in the PH's wallet. During verification, the verifier (i.e., IC) requested a proof with no revealed attributes, meaning the holder must prove possession of a valid credential without disclosing any information about its contents. For each presentation, the holder generated a Zero-Knowledge Proof using fresh randomness, producing new cryptographic commitments and nonces. This ensures that even when proofs are derived from the same underlying credential, their transcript structures are randomized and cannot be linked across sessions. To validate unlinkability, we recorded all proof presentations and analyzed the resulting transcripts for any repeated identifiers, stable commitments, signature metadata, or deterministic fields that could allow an adversarial verifier to correlate presentations. No such recurring values appeared; all cryptographic components were fresh per interaction. Each proof was verified successfully by the IC, while no attributes were revealed and no persistent identifier appeared in any of the presentations. Table 8 summarizes the number of presentations generated and the number that were confirmed to be unlinkable based on transcript inspection; while Figure 29 presents the results for 5 attributes.

Here, we offer a concise analysis of these works, accompanied by a comparative evaluation against the features and capabilities of PINSA.

Table 9 Security and Privacy requirements of related works

Security (S) and Privacy (P) Requirements	Works				
	SaCI	BlockCIS	SECONDO	INCHAIN	PINSA
S1 - Data confidentiality	✗	✓	✓	✓	✓
S2 - Data integrity and authenticity	✓	✓	✓	✓	✓
S3 - Non-repudiation	✓	✓	✓	✓	✓
S4 - Authorization and access control	✗	✓	✗	✓	✓
S5 - Accountability	✓	✓	✓	✓	✓
S6 - Physical protection	✗	✗	✗	✗	✓
P1 - Identity privacy	✦	✓	✓	✓	✓
P2 - Minimum data disclosure	✦	✗	✗	✓	✓
P3 - Privacy-preserving data accumulation	✦	✗	✗	✓	✓

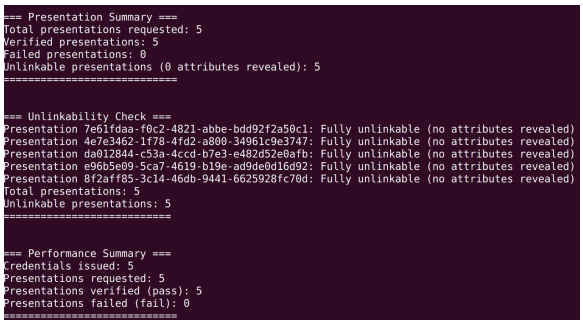

Figure 29 Unlinkability proof for 5 VCs

6.5 Comparative Analysis of Research Works in Addressing Security and Privacy Requirements

Here, we offer a concise analysis of works solely focusing on cyber insurance (see Section 2), accompanied by a comparative evaluation against the features and capabilities of `PINSA`. Table 9 compares related works with `PINSA` against the security and privacy requirements (see Section 3.5); the comparison is based on the following signs: ✓, ✗, ✦. The ✓ sign shows that the respective work fulfills the requirement. The ✗ sign shows that the requirement is considered a deficiency of the work, in the sense that the requirement is not fulfilled. When the ✦ sign is displayed, it means that the respective work does not include all the details needed, and assumptions were needed to come to a conclusion.

SaCI's framework [17] approach to anonymizing sensitive data prior to deploying contracts on the Blockchain marks a significant step towards ensuring that cyber insurance data and smart contracts are accessible only to the intended PHs and ICs. However, this approach does not solve the prevention of collusion or unauthorized access; thus, we assume that SaCI does not meet the data confidentiality requirement (S1). Furthermore, SaCI supports data integrity and authenticity (S2) by utilizing the immutable nature of Blockchain, which ensures data, once entered, cannot be altered or duplicated. Non-repudiation (S3) is assured within the SaCI framework through Blockchain's immutable ledger function, which irrefutably records the actions taken by PHs and ICs. While the authors of SaCI describe the use of Smart Contracts, which typically include mechanisms for enforcing access controls, SaCI lacks explicit details on how it ensures that access to Smart Contracts is granted only to legitimate PHs and ICs; thus we assume that SaCI does not meet the authorization and access control requirement (S4). Moreover, the Blockchain's transparent and irreversible record-keeping in the SaCI supports accountability (S5), ensuring that all actions of PHs and ICs are traceable and permanent. Finally, the primary focus of the SaCI framework is on digital security measures, with less focus on physical protection against adversarial actions (S6); thus, we assume that SaCI does not meet the physical protection requirement (S6). Regarding privacy requirements of Section 3.5.2, in SaCI, the process of anonymizing data before it is deployed on a public Blockchain is a fundamental approach to protecting PHs' identity privacy and ensuring minimal data disclosure. This approach of SaCI restricts ICs' ability to obtain crucial data. Nevertheless, there is a need for additional clarification regarding the precise methods employed to ensure the preservation of identity privacy (P1) and minimum data disclosure (P2) during transactions. Finally, the lack of SaCI to provide detailed mechanisms for unauthorized data modifications or overpricing indicates the need for

improvement in fully satisfying the requirement of privacy-preserving data accumulation (P3).

Next, BlockCIS [18] integrates private channels and selective disclosure to maintain data confidentiality (S1), ensuring that only authorized ICs and PHs can access sensitive information. The architecture of BlockCIS prioritizes using unchangeable logs for transactions and assets to enhance data integrity and authenticity (S2), ensuring secure logging of each transaction in BlockCIS. Furthermore, BlockCIS's Blockchain and accountability frameworks ensure non-repudiation (S3), where neither PHs nor ICs can disavow their actions related to the cyber insurance contracts. Also, BlockCIS defines participant roles and access permissions to ensure controlled access to smart contracts and sensitive data, in accordance with the Authorization and access control requirement (S4). The infrastructure's design ensures accountability (S5) by recording every transaction, making participants responsible for their actions as each transaction is traceable and auditable. The authors do not address the physical protection (S6). Regarding privacy requirements of Section 3.5.2, the mechanisms of private channels and selective disclosure within BlockCIS protect PH identities and ensure that ICs access only the minimal data necessary, addressing with this way the identity privacy (P1) and minimum data disclosure (P2). Finally, BlockCIS ensures secure and private data exchanges among ICs by enabling encrypted data storage and selective access, protecting PH identities from unauthorized disclosures, and preventing overpricing and competitive disadvantages, aligning with this way with privacy-preserving data accumulation (P3). Regarding privacy requirements of Section 3.5.2, the mechanisms of private channels within BlockCIS protect PH identities addressing with this way the identity privacy (P1). In addition, the authors of BlockCIS do not explicitly state if their approach of selective disclosure, which they refer to, is actually applied in BlockCIS. Therefore, we can infer that BlockCIS does not meet the requirement for minimum data disclosure (P2). Finally, the same ambiguity is observed when BlockCIS authors refer to the need to encrypt enterprise data using homomorphic encryption. Consequently, we assume that BlockCIS does not address the requirement for privacy-preserving data accumulation (P3).

Furthermore, Continuous Risk Monitoring and Blockchain module of SECONDO [19] updates the private Blockchain with information regarding the security and privacy risk, adhering to the data confidentiality (S1) by limiting access to ICs and PHs. The use of Blockchain in the architecture of SECONDO supports data integrity and authenticity (S2) as immutability feature of Blockchain guarantees that data, once recorded, cannot be changed. SECONDO utilizes smart contracts on the Blockchain to provide a mechanism for non-repudiation (S3). PHs and ICs' actions regarding their cyber insurance contracts are recorded in an unchangeable ledger, eliminating the possibility of any party denying their actions. Furthermore, SECONDO does not meet the authorization and access control requirement (S4). SECONDO focuses on quantification of cyber-physical risks and transparently publishing actions on the Blockchain, ensuring with this way accountability (S5) among participants. SECONDO does not address the physical protection (S6). Regarding privacy requirements of Section 3.5.2, SECONDO's design focuses on managing data on a private Blockchain to ensure that the identities of participating PHs are only revealed to their authorized ICs. This complies with the requirement for identity privacy (P1), as data stored in the Blockchain can only be accessed by authorized participants. Finally, SECONDO does not address the minimum data disclosure (P2) and privacy-preserving data accumulation (P3) requirements.

Finally, INCHAIN [14] ensures data confidentiality (S1) through the use of self-sovereign identity (SSI) and decentralized identifiers (DIDs), which restrict data access to authorized insurance companies ICs and policyholders PHs, preventing unauthorized access by external entities. The framework of INCHAIN ensures data integrity and authenticity (S2) by utilizing tamper-evident verifiable credentials (VCs) and cryptographic proofs to prevent data tampering and duplication, and to enable the verification of data origin. The system ensures non-repudiation (S3) through the use of Blockchain technology and cryptographic signatures, preventing PHs or ICs from denying their activities related to their responsibilities in the cyber insurance agreement. Authorization and access control (S4) are enforced using Decentralized Identifiers (DIDs) and Verifiable Credentials

(VCs) to ensure that only legitimate PHs registered with ICs can access Smart Contracts. The accountability of PHs and ICs is ensured using Blockchain and SSI, which record transactions on the Blockchain, creating an unchangeable audit trail. INCHAIN does not address the physical protection (S6) requirement. Regarding privacy requirements of Section 3.5.2, INCHAIN protects the identity privacy (P1) of PHs by utilizing Self-Sovereign Identity (SSI), allowing PHs to manage their identities and only legitimate ICs have access to their identities. Moreover, INCHAIN utilizes selective disclosure mechanisms in VCs to minimize data disclosure (P2), enabling PHs to share only necessary data with ICs for cyber insurance contracts, reducing data exposure. Finally, the architecture of INCHAIN through SSI supports privacy-preserving data accumulation (P3) avoiding modification of PHs's identities and protecting against overpricing.

6.6 Quantitative Comparison with Existing Frameworks

Here we quantitatively compare the PINSA's performance against other works from the related work. The criteria for a work to participate in the quantitative comparison against PINSA are the following.

[CR1] Solely focus on cyber insurance.
[CR2] Ecosystem development in widely accepted framework.
[CR3] Quantitative performance evaluation.

Following the aforementioned criteria, we defined that the only work that meets all three discrete criteria is [17] that solely focuses on cyber insurance (CR1), is developed on top of Ethereum (CR2), and has performed quantitative performance analysis (CR3). Table 10 compares related works against the identified criteria (see above); the comparison is based on the following signs: ✓ and ✗. The ✓ sign shows that the respective work fulfills the requirement. The ✗ sign indicates that the respective work does not include all the necessary details, and assumptions were required to reach a conclusion.

As discussed earlier, the work presented in [17] was developed on top of the Ethereum platform. To enable a direct comparison and to quantify the performance of PINSA, we evaluated our Hyperledger Fabric chaincode implementation against an equivalent Ethereum-based smart contract by analysing the operational costs associated with the execution of core functions. For this purpose, we developed an Ethereum smart contract that replicates the functionality of each chaincode function. The smart contract was implemented in Solidity, and for compilation and deployment, we used the Remix IDE. For the execution, we adopted the same gas price used in the SaCI implementation, which is equal to 20 gwei [17]. Execution costs were computed by multiplying the gas consumed by each function invocation by this gas price. Subsequently, we converted the resulting gas cost to Ether and then to euros, using both the exchange rate reported in the SaCI evaluation (600 € per Ether). Table 11 presents the execution costs for all PINSA functions, including the function that initializes the ledger.

Table 10 Related work comparison against identified criteria

Scientific Work	CR1	CR2	CR3
SECONDO [19]	✓	✗	✗
INCHAIN [14]	✓	✗	✗
SaCI [17]	✓	✓	✓
BlockCIS [18]	✓	✗	✗
Vakilinia et al. [21]	✗	✗	✗
Xu et al. [22]	✗	✓	✓
Nemade et al. [23]	✗	✗	✗

Functions of [17] map to the mechanisms implemented in PINSA. The constructor() aligns directly with the initLedger() from PINSA, as both initialize the system. The ReportDamage() from [17] corresponds to PINSA's reportIncident(), since each records a new incident along with the relevant identifiers and indemnification parameters. Similarly, acceptDamage maps to resolveIncident(), as both operations finalize the claim and update the associated limits or incident status. Also, functions such as paySecurity and declineDamage are conceptually analogous in our implementation. Particularly, registering a PH through createPolicyholder() resembles the act of providing a commitment or security deposit, even though no monetary transfer is executed. Likewise, resolving an incident without reducing coverage limits mirrors the conceptual logic of declining a claim. The acceptCounterOffer()

Table 11 Cost estimations of PINSA's function in Ethereum ecosystem

PINSA Function	Estimation in Ether (20 Gwei/Gas)	Converted in € (600 €/Ether)	Converted in € (2600 €/Ether)
initLedger	0.09342614	56.05	242.90
reportIncident	0.0026	1.56	6.76
resolveIncident	0.00222	1.33	5.77
updatePolicyholder	0.00108944	0.65	2.83
createPolicyholder	0.00431148	2.58	11.21
resolveIncident	0.00071378	0.43	1.86
checkObligations	0.0012476	0.75	3.24
analyzeContract	0.00202	1.21	5.25
deletePolicyHolder	0.00147276	0.88	3.82
readPolicyHolder	0.00060668	0.36	1.57
policyHolderExists	0.001	0.60	2.60

Table 12 Cost estimations of PINSA's function in Ethereum ecosystem

Correlation	[17] Costs in € (600 €/Ether)	PINSA Costs in € (600 €/Ether)
Initialization	constructor (65.36)	initLedger (56.05)
Reporting	ReportDamage (2.61)	reportIncident (1.56)
Decisioning	acceptDamage (0.65)	resolveIncident (1.33)
Updating	proposeToUpdateContract (1.58)	updatePolicyholder (0.65)
Decisioning	paySecurity (0.48)	resolveIncident (1.33)
Decisioning	declineDamage (1.04)	resolveIncident (1.33)
Accepting/Checking	acceptCounterOffer (0.49)	checkObligations (0.75)
Updating	agreeToUpdateContract (0.59)	updatePolicyholder (0.65)

loosely corresponds to checkObligations(), as the verification of controls and obligations can be interpreted as assessing negotiation conditions. Other actions related to payments or contract updates (e.g., payPremium, proposeToUpdateContract, and agreeToUpdateContract) also exhibit only partial correspondence. In PINSA, storing premium values or modifying a policyholder's risk attributes via createPolicyholder() or updatePolicyholder() captures these processes. However, the complete economic logic and multi-party approval workflows integral to [17] are not fully replicated in the blockchain-based implementation.

Overall, Table 12 demonstrates that while PINSA is not uniformly cheaper (PINSA has many more functions than [17]), it generally provides lower execution costs for most core operations related to initializing the system, reporting incidents, and updating records. Higher costs in specific areas reflect the broader logic embedded in PINSA 's functions rather than inefficiency. This supports the notion that PINSA can serve as a more economical alternative to Ethereum-based implementations for comparable cyber-insurance processes.

6.7 Summary

The performance evaluation of the Cyber Insurance Blockchain demonstrates that PINSA scales efficiently across increasing network sizes and workloads, exhibiting stable and predictable behavior across all measured indicators. The Throughput results reveal two distinct functional groups (i.e., query-based and invoke-based operations) whose performance patterns remain consistent regardless of node topology. Significantly, Throughput increases with transaction volume and stabilizes without degradation as nodes scale from 2 to 10, highlighting effective horizontal scalability. A similar bifurcation is observed in Average Latency, where lightweight queries yield consistently low delays and invoke operations incur higher but stable latency; the checkObligations function emerges as the most computationally intensive due to its internal chaincode calls. CPU utilization trends further reinforce system resilience, as all configurations exhibit nearly identical usage patterns without saturation or instability, indicating that the network effectively handles load distribution. Smart contract execution times increase moderately and proportionally with the

number of organizations, particularly for write-heavy functions requiring multi-organization endorsement, yet remain within predictable bounds. Overall, the results confirm that the Cyber Insurance Blockchain maintains performance stability and demonstrates strong scalability across heterogeneous workloads, validating its suitability for real-world cyber insurance processes requiring consistency and reliability under load.

The performance assessment of the SSI Blockchain reveals a clear relationship between credential complexity, attribute volume, and verification behavior, all of which significantly influence operational efficiency. The issuance of VCs exhibits an increasing average time as attribute counts grow, with fluctuations suggesting periodic computational overheads that are independent of attribute quantity. Verification tasks exhibit an even stronger dependency on requested attributes, with the average verification time exceeding the issuance time due to the additional steps involved, particularly in cases where predicate conditions are not met, necessitating extended wallet searches and additional local computations. VCs issuance latency further demonstrates that both batch size and credential complexity accumulate to increase end-to-end delays, especially under heavier loads. The frequency analysis reveals a consistent inverse relationship between attribute count and verification frequency, suggesting that more complex presentations require higher computational effort and consequently lower the achievable verification rate. Furthermore, verification frequency is consistently higher when predicates are satisfied, confirming that unsuccessful searches impose additional overhead. Collectively, these findings underscore that SSI performance is susceptible to the structure, size, and satisfaction of credential and presentation requirements, emphasizing the importance of optimizing attribute schemas and predicate policies for scalable deployments.

Across both blockchains, the results highlight complementary performance characteristics that collectively validate the feasibility of integrating Cyber Insurance processes with SSI-based identity and credential management in a unified privacy-preserving framework such as `PINSA`. The Cyber Insurance Blockchain exhibits strong horizontal scalability, predictable latency, and stable resource utilization, ensuring that high-volume insurance operations, such as incident reporting, policy updates, and obligation checks, can be executed efficiently even as organizational participation grows. Conversely, the SSI Blockchain demonstrates sensitivity to credential and presentation complexity, revealing that identity-driven processes introduce computational overhead that must be carefully managed through optimized credential structures and attribute selection. Despite these differences, both systems maintain consistent performance trends under increasing workloads, and neither exhibits detrimental degradation or instability. The complementary nature of their performance (fast, scalable transaction processing combined with flexible, attribute-based verification) confirms that `PINSA` can support end-to-end cyber insurance workflows with robust privacy guarantees while meeting operational demands.

Finally, `PINSA` outperforms existing cyber-insurance frameworks in security/privacy coverage. The comparative analysis highlights that, unlike the examined frameworks, `PINSA` is the only solution that simultaneously satisfies all the security and privacy requirements, particularly excelling in areas often underserved by prior work, such as physical protection, minimum data disclosure, and privacy-preserving data accumulation. Furthermore, `PINSA` provides strong security guarantees, having been tested for its capabilities in ensuring unlinkability and supporting selective disclosure. By leveraging the Zero-Knowledge Proof approach, `PINSA` successfully delivers both mandatory SSI features and robust security. At the same time, `PINSA` not only strengthens compliance with SSI principles but also reinforces compliance with security guidelines, directives, and standards.

7 Discussion

In this section, we address the research questions defined at the beginning of this article, delve into the `PINSA` limitations, and provide recommendations for potential areas of future study and development that might be explored to improve its capabilities and broaden its influence.

7.1 Addressing the Research Questions

By exploring research questions, we emphasize the transformative potential of Blockchain technologies within the cyber insurance ecosystem. Additionally, we outline a roadmap for implementing a secure, efficient, and privacy-preserving framework.

RQ1: How effectively can the Blockchain technology be integrated into the cyber insurance ecosystem? `PINSA` proposes a transformative shift towards fully automated operations, fostering a more agile and secure environment conducive to the rapid execution of cyber insurance contracts via Smart Contracts. The implemented functions of Smart Contracts (i.e., PH Creation, PH Reading, Incident Report, Incident Response, PH Obligation Checks, Asset Transfer, Violation, Contract Analysis) reduce the need for manual intervention and streamline the cyber insurance processes. The performance indicators of `PINSA`'s Cyber Insurance Blockchain—Throughput, Average Latency, and CPU Utilization—across different scales (such as number of transactions or nodes) demonstrate the efficiency of operations in completing tasks with minimal resources and time. It is noteworthy that CPU utilization of `PINSA`'s Cyber Insurance Blockchain fluctuates at a minimal rate of approximately of 5%. Furthermore, through SSI Blockchain and VCs, `PINSA` introduces a digital identity model where PHs have sole ownership over their cyber insurance data and control over how, when and to whom these data is shared. Finally, the performance key metrics of `PINSA`'s SSI Blockchain indicates that the issuance and the verification of this new digital identity in cyber insurance ecosystem is negligible. The experiments indicated that Average Time of VC's issuance is less than 0.5 secs and the Average Time of VC's presentation verification fluctuates at 1 sec approximately.

RQ2: What are the fundamental security and privacy requirements for a cyber insurance platform built on Blockchain? In `PINSA` we identified the minimum security and privacy requirements for a cyber insurance platform built on Blockchain technologies. The requirements are: (S1) Data confidentiality, (S2) Data integrity and authenticity, (S3) Non-repudiation, (S4) Authorization and access control, (S5) Accountability, (S6) Physical protection, (P1) Identity privacy, (P2) Minimum data disclosure, and (P3) Privacy-preserving data accumulation. These requirements are absolutely necessary, and we found that the absence of even one could cause risks in the cyber insurance ecosystem. For example, if the requirement of (P1) Identity privacy is not met, then the direct consequences are the heightened risk of privacy breaches and identity theft as sensitive data of PHs and ICs could be exposed to unauthorized parties, leading to potential misuses, financial frauds, and impersonations. Furthermore, the absence of (S5) Accountability may lead to PHs and ICs losing trust in the blockchain-based cyber insurance platform' ability to protect their sensitive data and its operational integrity, potentially leading to legal challenges, financial losses, and a damaged industry reputation. The last two examples indicate that the subtraction of even one of the aforementioned security and privacy requirements leads to a non-resilient cyber insurance platform.

RQ3: How can identity theft be prevented within the cyber insurance ecosystem? The SSI addresses the prevention of identity theft in the cyber insurance ecosystem. More specifically, `PINSA` the utilization of VCs, ensuring that the identity of PHs is indisputably authenticated before any transaction is executed. The VCs of the deployed SSI Blockchain of `PINSA` protects the privacy, increasing the difficulty of unauthorized access and impersonation. As a result, the risk of identity theft is eliminated, and trust among participants is strengthened, thereby creating a strong shield to protect against identity theft in the cyber insurance ecosystem.

RQ4: Which is the threat model of the cyber insurance ecosystem resulting from the incorporation of Blockchain implementations? In `PINSA`, we identified the threat model that evolves as a result of the actions of PH and ICs as well as external attackers. Internally, semi-honest PHs and ICs leverage systemic trust to engage in fraudulent activities. Examples of these fraudulent activities can be the submission of fraudulent claims or negligent cybersecurity incident investigations. Externally, adversaries focus

on attacking the technical foundations of the cyber insurance ecosystem in order to compromise Smart Contacts and disrupt Blockchain network communications.

7.2 Limitations

PINSA is susceptible to a single point of failure attacks due to using Hyperledger Aries as the SSI implementation. This vulnerability arises from concerns about the centralization of authority and data, potentially undermining the fundamental principles of SSI. The primary objective of the SSI initiative is to establish a decentralized identity ecosystem that fosters resilience and mitigates reliance on a singular governing entity. In the event of a compromise or downtime affecting the NCSA or an IC (see Figure 1), there exists the potential for significant disruption to the overall identity system. In this scenario, potential PHs could not issue VCs to renew or purchase a cyber insurance contract. Simultaneously, PHs experiencing a cybersecurity attack would be unable to report the incident. This situation can occur if the orderer nodes are centralized or not sufficiently redundant. Additionally, a single point of failure may arise if the plugged consensus service is not adequately distributed and redundant, potentially causing disruptions in the consensus process and halting transaction validation. Moreover, such an attack may result from channel configuration, mainly if it relies on a single entity to host critical components such as orderers or anchor peers. Furthermore, if the execution of Smart Contracts is not distributed across multiple peers, it could contribute to a single point of failure.

Moreover, the proposed PINSA is susceptible to attacks that exploit vulnerabilities within the network, consuming excessive resources and causing disruption or denial of service. These attacks are typically executed to overwhelm the system, slow transaction processing, and temporarily render the network unavailable. Resource attacks can manifest in various forms. First and foremost, attacks may occur through Smart Contract abuse, where a Smart Contract contains poorly optimized or malicious code that is designed to lead to excessive resource consumption during execution, thereby affecting overall network performance. In CPU exhaustion attacks, involved attackers deploy resource-intensive transactions or Smart Contracts on the network, consuming significant computational power. Consequently, legitimate transactions may experience delays or be unable to execute due to the excessive demand on the CPU. Similarly, in attacks related to memory exhaustion, attackers create transactions or invoke Smart Contracts that require large amounts of memory. Finally, attackers may flood the network with many low-value transactions, overwhelming its capacity to process them. In summary, these attacks can congest the network, slow down transaction processing for legitimate users, and deplete available resources on the nodes, leading to network unresponsiveness or crashes.

Furthermore, the performance evaluation of the Cyber Insurance Blockchain was carried out using configurations with 2, 4, 6, 8, and 10. Despite its limited scale, this setup provided a controlled environment for assessing key performance metrics, such as transaction throughput, latency, and resource consumption, while minimizing extraneous complexity. Although larger-scale deployments would provide broader insights, they entail considerable infrastructural demands. Nevertheless, given that blockchain performance characteristics often exhibit predictable scaling behaviors, the results obtained from this smaller-scale evaluation remain indicative and analytically valuable.

Additionally, this article lacks a quantitative comparison with other works in the cyber insurance sector. This gap exists for several reasons. Primarily, the literature referenced has predominantly been built on the Ethereum platform [17, 21], focusing solely on disclosing the gas requirements for each Smart Contract function. Notably, gas fees, a metric crucial for Ethereum, are not applicable within the Hyperledger Fabric ecosystem. This omission extends to other essential metrics, such as the time required to execute functions exclusively handling invoke requests. Furthermore, initiatives developed on the Hyperledger Fabric lack a comprehensive analysis and evaluation of their implementation [18]. Moreover, the work by Xu et al. [22] introduces a crowd-sourcing system that falls outside the scope of our article. Lastly, certain works offer only a theoretical approach devoid of practical implementation [14,19]. Therefore, it is evident that a quantitative comparison between PINSA and the existing literature is presently unfeasible.

7.3 Future Work

The findings of this article pave the way for numerous future research avenues. While this study presented a technical analysis of a cyber insurance ecosystem integrating Blockchain and SSI technologies—a pioneering exploration in this domain—there remains ample room for further exploration. Future investigations could delve into business cases and scenarios, emphasizing the innovative aspects of this ecosystem and showcasing its benefits to enhance the traditionally rigid cyber insurance landscape. Additionally, enhancing the practical usability of the framework is crucial for widespread adoption within the cyber insurance ecosystem. Integrating a user-friendly web application to streamline cyber insurance processes, enabling a simplified "one-click" cyber insurance experience for potential PHs, is a promising avenue for future development.

Another prospective direction involves the integration of a data-sharing platform on the Blockchain, fostering seamless collaboration among ICs. This collaborative approach aims to improve the understanding of evolving cyber risks, enhance risk modeling, and promote a more effective industry-wide response to cyber threats. Leveraging approaches such as the InterPlanetary File System (also known as IPFS) and Smart Contracts on Blockchain could be explored in this context. Furthermore, expanding cyber insurance coverages in response to evolving threats is an area of interest. Implementing a novel cyber insurance premium calculation formula within the framework could incorporate parameters such as employee behavior and awareness, along with vulnerabilities associated with PHs' IoT devices. Lastly, fortifying the framework with Artificial Intelligence (AI) and Machine Learning (ML) techniques holds promise. This enhancement could empower ICs to analyze vast datasets, identify emerging threats, and assess the cyber risk posture of potential PHs more accurately. This forward-looking integration of advanced technologies contributes to the adaptability and effectiveness of the cyber insurance ecosystem.

8 Conclusions

This article introduces an innovative cyber insurance framework, `PINSA`, which leverages well-established technologies, namely Hyperledger Fabric and Hyperledger Aries, to address the challenges within the existing cyber insurance ecosystem. At the core of `PINSA` is Hyperledger Aries, providing features such as data minimization, robust identification, data interoperability, portability, controllability, decentralization, and transaction transparency. This backbone empowers stakeholders to enhance their trustworthiness. Complementing this, Hyperledger Fabric forms the crux of `PINSA`, offering Smart Contracts and secure data storage, ensuring security, fairness, trust, and interoperability among participating entities. Through Smart Contracts integration, `PINSA` automates critical tasks related to claim handling and payment in the event of a cybersecurity incident.

Overall, this article presents `PINSA` as an innovative cyber insurance framework that brings distinct benefits when compared to current methodologies. The integrated technologies, Hyperledger Fabric and Hyperledger Aries, have demonstrated their effectiveness in assisting `PINSA` in meeting cyber insurance requirements and addressing challenges. Additionally, the article conducts a comprehensive performance evaluation of both utilized technologies, confirming `PINSA`'s effectiveness and efficiency. Furthermore, the security and privacy appraisal complements the performance evaluation, affirming `PINSA`'s capability to effectively defend against various well-known cybersecurity attacks and meet cyber insurance security and privacy requirements. Furthermore, we have to mention that `PINSA` exhibits interoperability at its blockchain layers. The SSI system, built on top of open Hyperledger Aries standards, operates independently of the underlying distributed ledger. This architectural decoupling enabled the smart-contract logic, initially deployed on Hyperledger Fabric, to be also migrated to an Ethereum-based environment without requiring any changes. Thus, `PINSA` demonstrates strong cross-platform interoperability and confirms that it avoids platform lock-in while remaining compatible with the broader decentralized-identity ecosystem.

As the number and frequency of cybersecurity incidents are expected to increase over time,

creating new victims, cyber insurance, as the only legal method to transfer cybersecurity risk, requires technological assistance to become proactive, robust, effective, and efficient. The research outcomes of this work aim to serve as a foundation for designing robust cybersecurity frameworks and architectures for the future of the cyber insurance ecosystem.

Acknowledgments

This research has received funding from the European Commission's Horizon Europe research and innovation programs under grant agreement No. 101120962 (RESCALE) and from the European Commission's Digital Europe program under grant agreements No. 101145872 (NITRO) and No. 101128024 (CYBERUNITY).

Declarations

Competing Interest Statement: The authors declare no conflict of interest.

Research Data/Code Availability: Data/code will be made available on request.

Compliance with Ethical Standards: This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Swiss Re Group. What you need to know about the Cyber insurance market. https://rb.gy/xg2pl7, 2023. Online; Last Accessed: 03/2025.

[2] World Economic Forum. Global risks 2023: Today's crisis. https://rb.gy/6gevz9. Online; Last Accessed: 11/2025.

[3] World Economic Forum. Global risks 2024: Insight report. https://shorturl.at/L9Uwo. Online; Last Accessed: 11/2025.

[4] World Economic Forum. Global risks 2025: Insight report. https://shorturl.at/H7gnV. Online; Last Accessed: 11/2025.

[5] TechTarget. Solarwinds hack explained: Everything you need to know. https://rb.gy/4cvzcs. Online; Last Accessed: 11/2025.

[6] SOCRadar. Microsoft exchange server cyberattack timeline. https://rb.gy/zhsvf3. Online; Last Accessed: 11/2025.

[7] TechTarget. Colonial pipeline hack explained: Everything you need to know. https://rb.gy/npu4cr. Online; Last Accessed: 11/2025.

[8] Markos Charalambous, Aristeidis Farao, George Kalantzantonakis, Panagiotis Kanakakis, Nikos Salamanos, Evangelos Kotsifakos, and Evangellos Froudakis. Analyzing coverages of cyber insurance policies using ontology. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–7, 2022.

[9] Cybersecurity Pulse. Throwback attack: How notpetya accidentally took down global shipping giant maersk. https://rb.gy/924sig. Online; Last Accessed: 11/2025.

[10] Insurance Journal. Insurance to fully cover sony's cyber attack, says ceo. https://rb.gy/rje9i0. Online; Last Accessed: 11/2025.

[11] Inside Privacy. Cyber insurer seeks to void data breach coverage because of purported misstatements in policy application. https://rb.gy/8vcxlt. Online; Last Accessed: 11/2025.

[12] Microsoft. Hackers hit norsk hydro with ransomware. the company responded with transparency. https://rb.gy/gbt85i. Online; Last Accessed: 11/2025.

[13] threat post. Insurance pays out a sliver of norsk hydro's cyberattack damages. https://rb.gy/95lg1m. Online; Last Accessed: 11/2025.

[14] Aristeidis Farao, Georgios Paparis, Sakshyam Panda, Emmanouil Panaousis, Apostolis Zarras, and Christos Xenakis. Inchain: a cyber insurance architecture with smart contracts and self-sovereign identity on top of blockchain. *International Journal of Information Security*, pages 1–25, 2023.

[15] Hyperledger Foundation. Hyperledger Fabric. https://www.lfdecentralizedtrust.org/projects/fabric, 2023. Online; Last Accessed: 11/2025.

[16] Hyperledger Foundation. Hyperledger Aries. https://www.lfdecentralizedtrust.org/projects/aries, 2023. Online; Last Accessed: 11/2025.

[17] Muriel Franco, Noah Berni, Eder Scheid, Christian Killer, Bruno Rodrigues, and Burkhard Stiller. Saci: A blockchain-based cyber insurance approach for the deployment and management of a contract coverage. In *Economics of Grids, Clouds, Systems, and Services: 18th International Conference, GECON 2021, Virtual Event, September 21–23, 2021, Proceedings 18*, pages 79–92. Springer, 2021.

[18] Tancrede Lepoint, Gabriela Ciocarlie, and Karim Eldefrawy. Blockcis—a blockchain-based cyber insurance system. In *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pages 378–384. IEEE, 2018.

[19] Aristeidis Farao, Sakshyam Panda, Sofia Anna Menesidou, Entso Veliou, Nikolaos Episkopos, George Kalatzantonakis, Farnaz Mohammadi, Nikolaos Georgopoulos, Michael Sirivianos, Nikos Salamanos, et al. Secondo: A platform for cybersecurity investments and cyber insurance decisions. In *International Conference on Trust and Privacy in Digital Business*, pages 65–74. Springer, 2020.

[20] Ioannis Kalderemidis, Aristeidis Farao, Panagiotis Bountakas, Sakshyam Panda, and Christos Xenakis. Gtm: Game theoretic methodology for optimal cybersecurity defending strategies and investments. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–9, 2022.

[21] Iman Vakilinia, Shahriar Badsha, and Shamik Sengupta. Crowdfunding the insurance of a cyber-product using blockchain. In *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 964–970. IEEE, 2018.

[22] Jia Xu, Yongqi Wu, Xiapu Luo, and Dejun Yang. Improving the efficiency of blockchain applications with smart contract based cyber-insurance. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2020.

[23] Srushti Nemade, Abhijit Kamble, Sharvil Sopal, Pragati Bhale, and Vinod Pachghare. Blockchain-based crowdfunding for cyber-product insurance. In *2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, pages 1–5. IEEE, 2022.

[24] Sakshyam Panda, Daniel W Woods, Aron Laszka, Andrew Fielder, and Emmanouil Panaousis. Post-incident audits on cyber insurance discounts. *Computers & Security*, 87:101593, 2019.

[25] Aggeliki Tsohou, Vasiliki Diamantopoulou, Stefanos Gritzalis, and Costas Lambrinoudakis. Cyber insurance: state of the art, trends and future directions. *International Journal of Information Security*, pages 1–12, 2023.

[26] Insurance Fraud Bureau New Zealand Sophos News. Claiming with multiple insurers. https://rb.gy/dmczym. Online; Last Accessed: 11/2025.

[27] ENISA. Identity theft: Enisa threat landscape.

[28] George Suciu, Aristeidis Farao, Giorgio Bernardinetti, Ivan Palamà, Mari-Anais Sachian, Alexandru Vulpe, Marius-Constantin Vochin, Pavel Muresan, Michail Bampatsikos, Antonio Muñoz, et al. Samgrid: Security authorization and monitoring module based on sealedgrid platform. *Sensors*, 22(17):6527, 2022.

[29] Amwins. How Cyber and Crime Insurance Policies Respond to Social Engineering. https://rb.gy/of063c, 2024. Online; Last Accessed: 11/2025.

[30] Coalition. Funds Transfer Fraud Insurance. https://rb.gy/bxvy5c, . Online; Last Accessed: 11/2025.

[31] Corvus Insurance Holdings, LLC. Cyber Coverage Explained: Social Engineering Attacks and Cyber Crime. https://shorturl.at/Q3Jxt, 2021. Online; Last Accessed: 11/2025.

[32] Pantelitsa Leonidou, Nikos Salamanos, Aristeidis Farao, Maria Aspri, and Michael Sirivianos. A qualitative analysis of illicit arms trafficking on darknet marketplaces. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pages 1–9, 2023.

[33] Vasileios Pantelakis, Panagiotis Bountakas, Aristeidis Farao, and Christos Xenakis. Adversarial machine learning attacks on multiclass classification of iot network traffic. In

Proceedings of the 18th International Conference on Availability, Reliability and Security, pages 1–8, 2023.

[34] George Petihakis, Dimitrios Kiritsis, Aristeidis Farao, Panagiotis Bountakas, Aggeliki Panou, and Christos Xenakis. A bring your own device security awareness survey among professionals. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pages 1–10, 2023.

[35] PWC. Blockchain, a catalyst for new approaches in insurance.

[36] ZYEN. Interexchainz research project. https://rb.gy/fgb6pd. Online; Last Accessed: 11/2025.

[37] Keyun Ruan. *Digital asset valuation and cyber risk measurement: Principles of cybernomics*. Academic Press, 2019.

[38] Rainer Böhme, Galina Schwartz, et al. Modeling cyber-insurance: towards a unifying framework. In *WEIS*, pages 1–36, 2010.

[39] Angelica Marotta, Fabio Martinelli, Stefano Nanni, Albina Orlando, and Artsiom Yautsiukhin. Cyber-insurance survey. *Computer Science Review*, 24:35–61, 2017.

[40] Ruperto P Majuca, William Yurcik, and Jay P Kesan. The evolution of cyberinsurance. *arXiv preprint cs/0601020*, 2006.

[41] Sasha Romanosky, Lillian Ablon, Andreas Kuehn, and Therese Jones. Content analysis of cyber insurance policies: How do carriers price cyber risk? *Journal of Cybersecurity*, 5(1):tyz002, 2019.

[42] Savino Dambra, Leyla Bilge, and Davide Balzarotti. Sok: Cyber insurance–technical challenges and a system security roadmap. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1367–1383. IEEE, 2020.

[43] Rainer Böhme and Gaurav Kataria. Models and measures for correlation in cyber-insurance. In *WEIS*, volume 2, page 3, 2006.

[44] Rainer Böhme. Cyber-insurance revisited. In *Weis*, pages 1–17, 2005.

[45] Baharuddin Aziz et al. A systematic literature review of cyber insurance challenges. In *2020 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 357–363. IEEE, 2020.

[46] Prof. Dr. Stefan Weber, Prof. Dr. Matthias Scherer, Corina Pascu, and Marco Barros Lourenço. Cyber insurance – models and methods and the use of ai. Technical report, European Union Agency for Cybersecurity (ENISA), February 2024.

[47] Jiayi Guo, Zhiyu Quan, and Linfeng Zhang. Entity-specific cyber risk assessment using insurtech empowered risk factors. *arXiv preprint arXiv:2507.08193*, 2025. Submitted 10 Jul 2025; revised 14 Jul 2025.

[48] David Byrd and Antigoni Polychroniadou. Differentially private secure multi-party computation for federated learning in financial applications. *arXiv preprint arXiv:2010.05867*, 2020.

[49] Olivier Lopez, Michel Denuit, Mario Ghossoub, Julien Trufin, Justin Kher, Arthur Maillart, Elisabeth Raes, Hugo Rapior, Mohammed-Amine Skoubani, and Brieuc Spoorenberg. Cyber risk: Research paper on cyber insurance. Technical report, Casualty Actuarial Society (CAS), June 2025.

[50] Mingyu Hao, Keyang Qian, and Sid Chi-Kin Chau. Privacy-preserving blockchain-enabled parametric insurance via remote sensing and iot. *arXiv preprint arXiv:2305.08384*, 2023.

[51] Zixin Nie, Benjamin Tyndall, Daniel Brannock, Emily Gentles, Elizabeth Parish, and Alison Banger. A framework for the design of privacy-preserving record linkage systems. *Journal of Cybersecurity and Privacy*, 5(3):44, 2025.

[52] Darren Pain. Cyber risk accumulation: Fully tackling the insurability challenge. Technical report, The Geneva Association, November 2023.

[53] Moody's Analytics Cyber Risk Team. Insurance cyber risk: Moody's insights. https://www.moodys.com/web/en/us/capabilities/cyber-risk/insurance-cyber-risk.html, 2024. Accessed: 25 November 2025.

[54] Sakshyam Panda, Aristeidis Farao, Emmanouil Panaousis, and Christos Xenakis. Cyber-insurance: Past, present and future. In *Encyclopedia of Cryptography, Security and Privacy*, pages 1–4. Springer, 2021.

[55] Hyperledger Foundation. Hyperledger caliper. https://www.hyperledger.org/use/caliper, 2023. Online; Last Accessed: 03/2025.

[56] Hyperledger Foundation. Hyperledger Aries

Cloud Agent Python. https://github.com/hyperledger/aries-cloudagent-python, 2023. Online; Last Accessed: 11/2025.

[57] SQLite. SQLite. https://www.sqlite.org, 2023. Online; Last Accessed: 11/2025.

[58] European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), April 2016. Official Journal of the European Union, L119, 1–88.

[59] European Union. Regulation (eu) no 910/2014 of the european parliament and of the council of 23 july 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing directive 1999/93/ec (eidas regulation), July 2014. Official Journal of the European Union, L257, 73–114.

[60] European Union. Directive 2009/138/ec of the european parliament and of the council of 25 november 2009 on the taking-up and pursuit of the business of insurance and reinsurance (solvency ii), November 2009. Official Journal of the European Union, L335, 1–155.

[61] European Union. Directive (eu) 2016/97 of the european parliament and of the council of 20 january 2016 on insurance distribution (insurance distribution directive, idd), January 2016. Official Journal of the European Union, L26, 19–59.

[62] European Union. European insurance and occupational pensions authority (eiopa), 2025. EIOPA is the EU authority responsible for supporting the stability of the insurance and occupational pensions sectors, protecting policyholders, and promoting transparency.