Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Terje Haugum
Bjørnar Hoff

A Tool to Support Blockchain Threat Modeling

Master's thesis in Computer Science
Supervisor: Jingyue Li
June 2022

NTNU
Norwegian University of
Science and Technology

Terje Haugum
Bjørnar Hoff

A Tool to Support Blockchain Threat Modeling

Master's thesis in Computer Science
Supervisor: Jingyue Li
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Context: The concern about security in blockchain technology has risen naturally due to expanded interest over the last few years. Potential threats disclosed by existing threat modeling tools are related to general IT systems. There is a lack of research on how existing threat modeling tools can share and categorize threats regarding blockchain technology.

Objective: The primary purpose of this research is to analyze blockchain threats in the first layer and integrate this analysis into existing threat modeling tools. Our goal is to record and analyze existing knowledge, provide a new prototype plugin for existing threat modeling tools regarding blockchain technology and identify new future research opportunities.

Method: We conducted a literature review with predefined procedures for this research and followed a design science methodology to build the prototype. 20 relevant papers from the literature search became the primary papers. 73 different blockchain related threats were identified, and 18 blockchain interoperability vulnerabilities were extracted from our multivocal literature review. To evaluate, the paper conducts asynchronous remote usability testing combined with a digital survey where the evaluator gets the artifact and a link to an online survey. The evaluation method used is justified as the most efficient due to participants' different time zones and living places. Responses were then analyzed and evaluated.

Results: Our review identified security threats in the blockchain. We analyzed existing literature and categorized the threats into different categories. We adopted the threat model approach STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of service, Evaluation of privilege) in order to link the parameters to each identified blockchain threat. Building a plugin prototype for threat modeling to blockchain technology was accomplished. Our prototype provides features such as adding new blockchain technology to the database, discovering all identified threats found in our literature search, and adding newly discovered threats. Finally, the evaluation shows that participants were optimistic about the application's overall experience and believed that the application could provide value for further blockchain technology development and implementation.

Conclusion: As the technology grows, the security concern arises and the need for modeling threats for the blockchain are crucial for future development and implementations. Existing threat modeling tools do not cover the technology. Therefore, our research provides a trustworthy starting point. This paper summarizes threats found and presents a prototype for a threat modeling tool for blockchain, which can be potentially integrated with existing threat modeling tools in the future.

Sammendrag

Kontekst: Bekymringen for sikkerhet i blokkjedeteknologi har økt naturlig på grunn av økt interesse de siste årene. Potensielle trusler avslørt av eksisterende trusselmodelleringsverktøy er relatert til generelle IT-systemer. Det er mangel på forskning på hvordan eksisterende trusselmodelleringsverktøy kan dele og kategorisere trusler angående blokkjedeteknologi.

Mål: Hovedformålet med denne forskningen er å analysere blokkjedetrusler i det første laget og integrere denne analysen i eksisterende trusselmodelleringsverktøy. Målet vårt er å registrere og analysere eksisterende kunnskap, tilby en ny prototype-plugin for eksisterende trusselmodelleringsverktøy angående blokkjedeteknologi og identifisere nye fremtidige forskningsmuligheter.

Metode: Vi gjennomførte en litteraturgjennomgang med forhåndsdefinerte prosedyrer for denne forskningen og fulgte en designvitenskapelig metodikk for å bygge prototypen. 20 relevante artikler fra litteratursøket ble hovedpapirene. 73 forskjellige blokkkjederelaterte trusler ble identifisert, og 18 interoperabilitetssårbarheter for blokkjeder ble trukket ut fra vår multivokale litteraturgjennomgang. For å evaluere, gjennomfører papiret asynkron ekstern brukervennlighetstesting kombinert med en digital undersøkelse der evaluatoren får artefakten og en lenke til en online undersøkelse. Evalueringsmetoden som benyttes er begrunnet som den mest effektive på grunn av deltakernes ulike tidssoner og oppholdssteder. Svarene ble deretter analysert og evaluert.

Resultater: Gjennomgangen vår identifiserte sikkerhetstrusler i blokkjeden. Vi analyserte eksisterende litteratur og kategoriserte truslene i ulike kategorier. Vi tok i bruk trusselmodelltilnærmingen STRIDE for å koble parametrene til hver identifisert blokkjedetrussel. Å bygge en plugin-prototype for trusselmodellering til blokkjedeteknologi ble oppnådd. Prototypen vår gir funksjoner som å legge til ny blokkjedeteknologi til databasen, oppdage alle identifiserte trusler funnet i vårt litteratursøk, og legge til nyoppdagede trusler. Til slutt viser evalueringen at deltakerne var optimistiske med hensyn til applikasjonens samlede opplevelse og mente at applikasjonen kunne gi verdi for videre utvikling og implementering av blokkjedeteknologi.

Konklusjon: Etter hvert som teknologien vokser, oppstår sikkerhetsbekymringer, og behovet for modellering av trusler for blokkjeden er avgjørende for fremtidig utvikling og implementeringer. Eksisterende trusselmodelleringsverktøy dekker ikke nå teknologien. Derfor gir vår forskning et pålitelig utgangspunkt. Denne artikkelen oppsummerer trusler funnet og presenterer en prototype for et trusselmodelleringsverktøy for blokkjede, som kan potensielt integreres med eksisterende trusselmodelleringsverktøy i fremtiden.

Acknowledgement

We want to express our gratitude to our primary supervisor, Jingyue Li, who provided an exciting project and guided us while finishing up the work.

We wish to acknowledge the support by the technical and support staff in PaaSforChain (Platform as Service Technologies for High-performance Blockchain-based Supply Chain Management Systems) at NTNU and the Research Council of Norway (No.309494).

The overall assistance provided by Mohammed L. K. Alsadi was greatly appreciated. Alsadi offered deep blockchain knowledge and valuable feedback which we used in our project.

Based on the work with the specialization project, we got invited to the 3rd Blockchain Software Engineering Workshop, hosted by The International Conference on Evaluation and Assessment in Software Engineering (EASE). Attending the online conference and glimpsing what work is currently being researched in the field motivates further research and implementation.

Lastly, we would be remiss in not mentioning our family and other students at the department for their belief and moral support. We strongly appreciate our collaboration throughout the research process.

Contents

List of Figures

List of Tables

Acronyms

ASF Application Security Frame

DFD Data Flow Diagram

DSR Design Science Research

ER Entity Relationship

IT Information Technology

MLR Multivocal Literature Review

NTNU Norges Teknisk-naturvitenskapelige Universitet

OWASP Open Web Application Security Project

RQ1 Research Question 1

RQ2 Research Question 2

UML Unified Modeling Language

1 Introduction

The technology, named blockchain, was first revealed in Bitcoin Whitepaper by Satoshi Nakamoto back in 2008 [4]. Even though the groundbreaking paper was published without the author's true identity, blockchain as a technology has attracted a lot of attention from the industry in recent years. As the underlying technique of Bitcoin, the blockchain has been introduced and applied to many different fields, such as healthcare [5] [6], Internet of things [7] [8], and software engineering [9] [10]. As the industry grows and the technology is being adapted in different sectors, the rise of security concerns arises naturally.

Given a cutting-edge technology, there have been numerous reported attacks and vulnerabilities identified in blockchain [11] [12] [13]. Recurrence of a series of thefts (wallets and accounts) and hacking, it is urgent to establish security tools to improve the blockchain system security [14].

Designing secure computer systems is a complex problem. As the security techniques get more complicated than ever before, attackers routinely break into systems. In a book by Bruce Schneier [15], he states:"Security is a chain; its only as secure as the weakest link. Security is a process, not a product". Addressing security threats in an early stage provides fewer time constraints, lower cost, and no necessity for retrofit security into existing systems [16] [17]. In order to identify these threats, threat modeling tries to identify all possible threats and whether or not they can be exploited. Accepting or mitigating the risk are decisions made during the threat modeling process. Any system that interacts with the digital world will benefit from threat modeling regardless of which stage of the development process. Therefore, research in threat modeling for blockchain technology is necessary and valuable for further research and development.

Several papers cover blockchain security threats [14] [18] [19]. There exist several studies on threat modeling for information technology systems [20] [21] [22], but none of them targets blockchain technology. The closest research on security threat modeling for blockchain is written by Landuyt et al [23]. In their research, they integrate decentralized architectures such as distributed ledgers or blockchains into Data Flow Diagrams (DFDs). As stated in their future work; "tool support for more sophisticated threat elicitation approaches" is needed [23].

Since blockchain eliminates the presence of a central authority, all blockchain operations and transactions must be protected and securely stored on a distributed ledger. As a result, blockchain consists of different layers, and each layer has its distinct functionality. However, a set of different vulnerabilities follows the prescribed layers.

Motivated by the lack of research, limitations, and the point for future work by Lanuyt et al. [23], our contribution will cover threats that have their origin in layer 0 and layer 1. These layers consist of the blockchain's infrastructure and consensus. However, there are essential aspects in higher layers affected by vulnerabilities that have their origin in lower layers. For that reason, we also included some additional information and threats regarding higher levels. Instead of delving into higher levels with multiple variations of features, starting at a basic foundation and defining potential threats with origin in lower layers was necessary. Due to an unsearched field and the thesis's time, these restrictions were made.

This paper presents two research questions:

RQ1: What are the most important characteristics of vulnerabilities related to blockchain?

RQ2: How to develop a tool to facilitate blockchain threat modeling by incorporating the blockchain vulnerability characteristics?

RQ1 aims to identify and analyze threats of blockchain mentioned by researchers in scientific literature by categorizing each threat into respective categories. **RQ2** aspires to integrate the analysis found in the first research question into an application created for modeling threats in blockchain technology.

Based on the research accomplished, our contribution to analyzing 73 threats related to blockchain technology extracted from 20 primary papers, and integrating these threats into a threat modeling tool marks the starting point for future blockchain threat modeling.

1.1 Organization of the report

The structure of the paper is as follows. Section 2 presents the background of this work and introduces blockchain technology and blockchain interoperability, and additionally educates the reader about threat modeling for information technology systems. Section 3 analyzes and discusses work related to our proposal and existing threat modeling tools for blockchain. In section 4, we describe the method used and provide detailed information on how we conducted the review and the research methodology used to create a prototype of the artifact. Section 5 presents the result of RQ1 in addition to the implementation process and results of RQ2 related to our research questions. Section 6 presents the evaluation of the artifact. In section 7 we provide discussions on the results. Finally, we conclude and summarize our work, as well as provide implications to academics and industry in section 8. Our planned future work is listed in section 9.

2 Background

To help people understand blockchain security issues, we think it is necessary to explain and define general blockchain terms and some challenges regarding any distributed system. In this section, we provide an introduction to the technology and different strategies to achieve blockchain and their interoperability. Additionally, the section introduces threat modeling related terms. Some sections in the following chapter contain the same detailed descriptions provided in the author's MLR [3]; therefore, sections where citation [3] appear, are extracted descriptions.

2.1 Blockchain technology

As explained in [3], the blockchain technology was first introduced by Satoshi Nakamoto in 2008 [4]. As an underlying technology of Bitcoin, this was a huge technological improvement for the financial industry. The technology could manage vast amounts of transactions without downtime due to singular faults was increasingly more difficult.

To this date, traditional conventions of processing transactions, such as intermediation, are still the most used solution. Occurring financial crises such as the collapse of the investment bank Bear Sterns in 2008 establishes the significance of a reliable capable of digital transaction handling [24]. Nakamoto disclosed a new technology in the Bitcoin Whitepaper building trust within a distributed system utilizing the architecture of each block in a chain of blocks where the trust is emanated by cryptographic means [25].

The blockchain is capable of maintaining the history of all transactions by each block points to the previous block utilizing a hash value with a corresponding timestamp [24]. The quantity of transactions within a block relies on block size and the transaction itself. The transactions get validated through asymmetric cryptography. In the case of Bitcoin, this is accomplished peer to peer, causing the validation process to be decentralized where a central agency is not required [26]. The structure and content of a typical blockchain are shown in Figure 1.

2.1.1 Blockchain structure

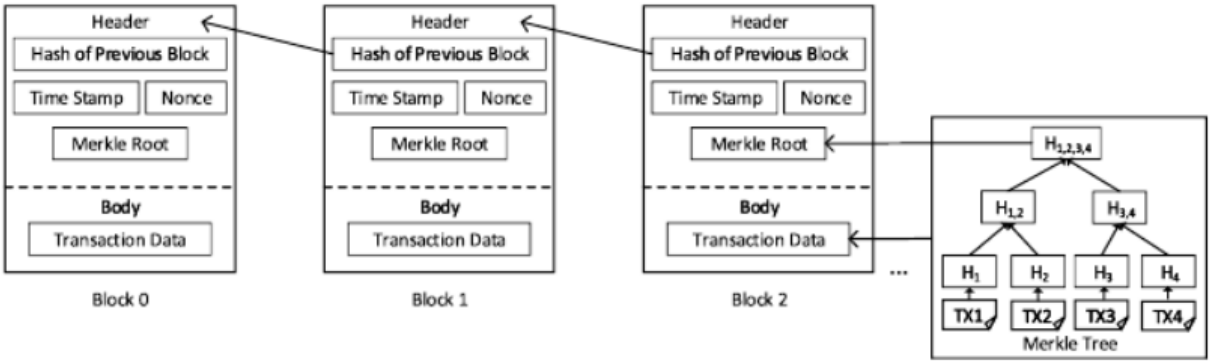


Figure 1: Typical blockchain [1]

2.1.2 Blockchain trilemma

Summarized in [3], the traditional distribution system's central awareness is noted by Brewers' CAP theorem, where a distributed web service cannot guarantee both consistencies, partition, and availability [27]. Besides, researchers have incorporated the trilemma of blockchain technology where a blockchain cannot provide secure systems which are both scalable and completely decentralized [28]. The Figure 2 illustrates the three main issues that developers encounter when building blockchains.

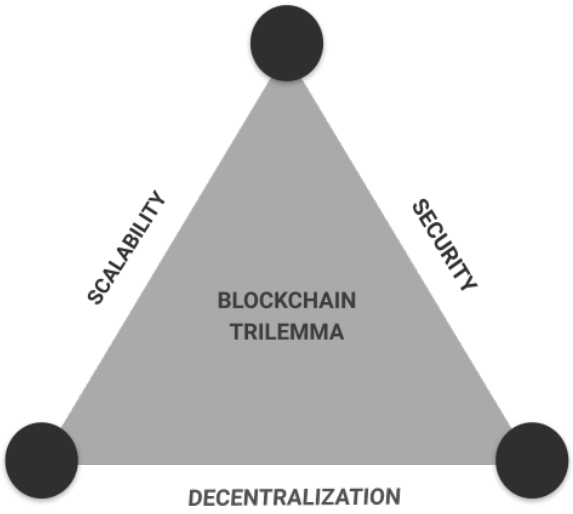


Figure 2: Blockchain Trilemma

Before we delve into further details, it is crucial to define security, scalability, and decentralization in the context of blockchain:

- **Security** refers to the ability to secure data on the blockchain and how the blockchain defends against different types of attacks.
- **Scalability** refers to the ability to handle and support the increasing volume of transactions, as well as an increasing number of participating nodes in the network.
- **Decentralization** refers to a network redundancy that transfers the decision-maker and supervision from a centralized entity to a more dispersed network.

2.1.3 Blockchain layers

Since blockchain technology grow rapidly, different architectures are being developed in the literature and the overall expansion of the technology is challenging to cover. In this section, we will provide a categorization of different blockchain layers based on currently identified layers and those layers that are relevant for our thesis. The stack and layers of a typical blockchain are defined and shown in Figure 3 [2].

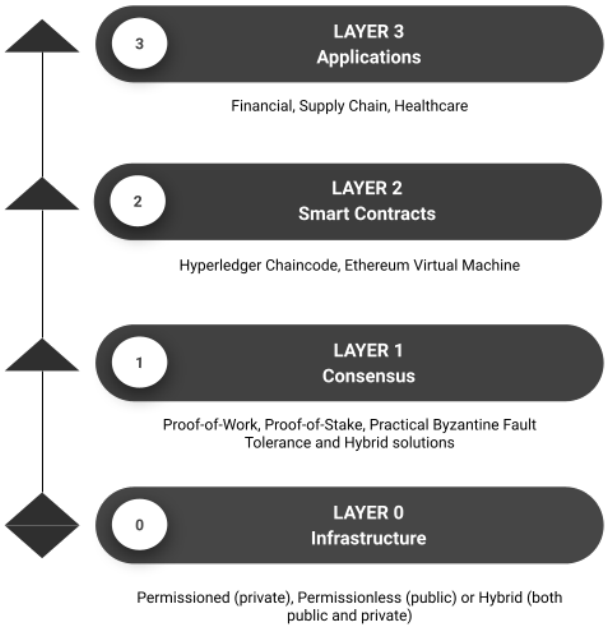


Figure 3: Layers of Blockchain Architecture, inspired by [2]

- **Layer 0** refers to the underlying infrastructure of a blockchain. This layer comprises components like hardware, network, and connections to provide communication between chains created on top of this layer.
- **Layer 1** often refers to the consensus layer or implementation layer. This layer consists of a consensus process, block time, and other parameters to maintain basic blockchain functionality and security based on its immutability. Bitcoin is well known as a layer one blockchain.

- **Layer 2** refers to solutions (ex. smart contracts) that help scale the blockchain and improve problems in lower layers. These problems can be transaction speed and throughput. Transactions and processes can occur independently of the main chain without sacrificing network security. Bitcoin Lightning Network is a Layer 2 solution striving to increase the throughput.
- **Layer 3** refers to the application layer of the blockchain. This layer enables Decentralized Applications (DApps) to be built on top of the blockchain. Some blockchains have a collection of these applications, and some have not. Ethereum enables DApps, while Bitcoin does not have layer three functionality.

2.1.4 Consensus algorithms

In blockchain, consensus is a fault-tolerant mechanism to agree on a state in the network among distributed nodes and processes. A typical mechanism is a transformation of the Byzantine Generals problem [29]. Blockchains do not have any central authority to ensure that nothing malicious happens; all nodes need to trust each other. The following section will present common approaches to reaching a consensus within the blockchain.

2.1.5 Approaches to consensus

In [3], we explain that there exist multiple consensus algorithms [30] with multiple variations. According to [31], we will, in short, represent three of the most common ones.

Proof-of-work: It is the most common consensus mechanism utilized and is used by Bitcoin. For a block to be established, the transactions are validated by every node. However, only the first node that has succeeded in computing it will be able to add it to the chain. The reward provided by the blockchain is usually the respect token of the blockchain. The computation, named mining, is not energy efficient and can be very costly [3].

The purpose of the computation is to find the nonce of the previous block in the chain and then add a block to the chain of blocks. This is hard to compute while easy to verify by others [31].

Proof-of-stake: This is another standard consensus algorithm made to decrease the energy expenses of proof-of-work, substituting it with staking. A stake is identical to the amount of value in the respected token of the blockchain the node (validator) is willing to lock. According to the amount of time and the portion of the stake, the validator with the highest value acquires to add a block to the chain [26].

Practical Byzantine Fault Tolerance (PBFT): Is a voting-based consensus model made with the idea of existing malicious nodes in the network. PBFT will still work as long as there exists less than $\frac{1}{3}$ of malicious nodes in the network [3].

PBFT operates utilizing the node's ability to communicate with each other. There exists one elected leader node and the rest are backup nodes. Whenever a client issues a transaction, it is sent to the leader, broadcasting it to the backup nodes. The nodes decide whether to execute the requests and send a reply to the client. If the client receives replies from $f+1$ (f is the number of possible malicious nodes), an agreement has been made [26]. The leader node is being replaced using the deterministic algorithm round-robin [3].

Byzantine Fault Tolerance has been well researched, which this algorithm derives. Even though PBFT is implemented to provide cost-effective validations, it comes with its disadvantages with scaling and security. As the consensus fails if more than $\frac{1}{3}$ of the nodes are malicious, the model's security thrives with more nodes. However, with more nodes, the algorithm needs additional communication between the nodes and, therefore, lacks latency. In order to fix this, one would build an alternative or a hybrid version, including more than one algorithm to find consensus [31] [3].

2.2 Blockchain interoperability

In [3], we explained that blockchain technology is growing rapidly and driving an expanded number of separated and unconnected systems. As a result, blockchain networks today operate within silos. Interoperability is one of the most challenging and crucial aspects for further adoption to make interactable blockchains.

As described in [3], to clearly understand how different blockchains can communicate, it is crucial to furnish a clear definition of interoperability. In 1996, Peter Wegner stated: "Interoperability is the ability of two or more software components to cooperate despite differences in language, interface, and execution platform." [32]. Blockchain interoperability is the ability to share, see, transact and access information across different blockchain networks without any centralized authority. Providing communication between two blockchains involves one source blockchain and a target blockchain, where the source blockchain initiates a transaction to be executed on the target blockchain [33]. As Figure 4 illustrates, in an own defined ecosystem, various blockchains allow communication, transfer of digital assets and data between one another, and enable collaboration across different blockchain networks.

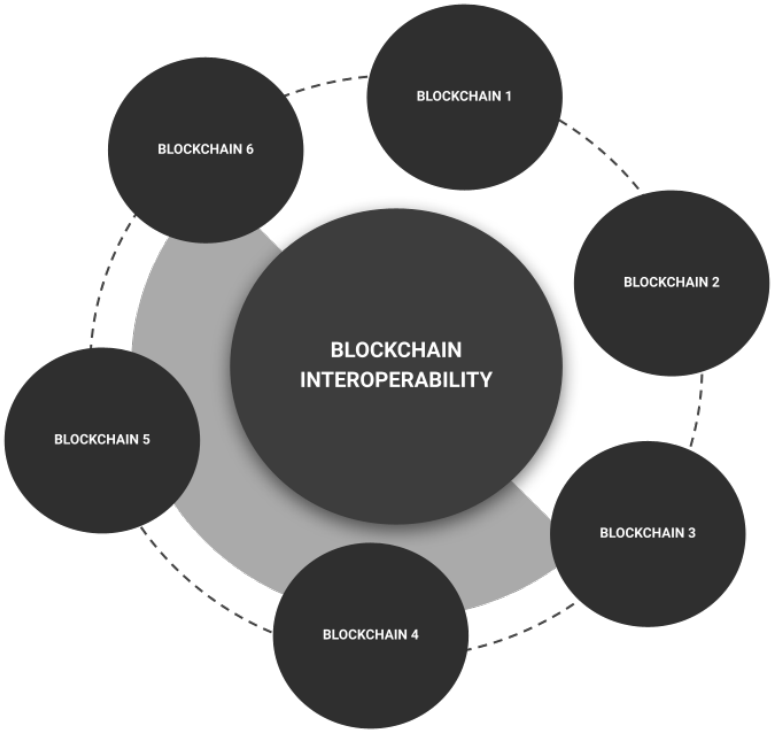


Figure 4: Blockchain Interoperability [3]

In the following section, we will cover strategies to reach interoperability. Our goal is to equip the reader with the necessary insight for understanding how these interoperable blockchains communicate. These strategies are extracted from a survey conducted by Belchior et. al [34].

2.2.1 Strategies for chain interoperation

Belchior et. al [34] shows that there exist three main strategies to reach chain interoperation. As recapped in [3], the following section will present these strategies and provide a concise description of how it is achieved.

Notary scheme: A notary scheme is the most straightforward strategy to facilitate cross-chain communication. This strategy applies a trusted entity that monitors all events that happen on multiple chains [33]. A notary mechanism is a trusted entity that can claim to one chain that a given event on another chain took place. Figure 5 shows a simple visualization of how the notary scheme is achieved.

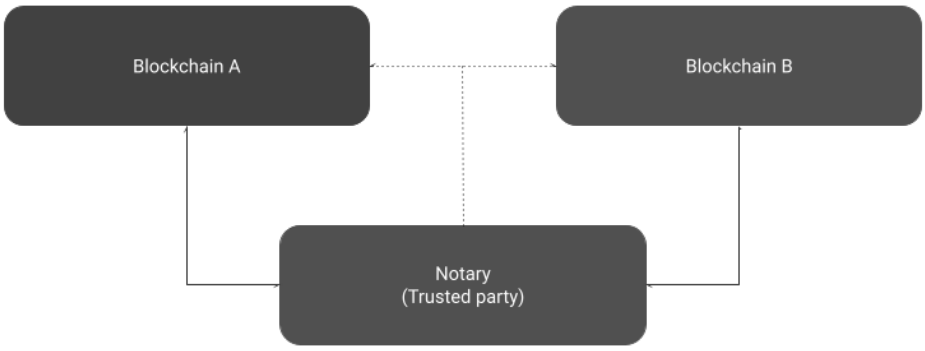


Figure 5: Notary Scheme

Sidechains/Relay: Instead of relying on a trusted entity, relays are a mechanism inside a blockchain that can read and validate states in other blockchains. The mainchain (i.e the original blockchain) keeps a ledger of assets connected to the sidechain, providing the mainchain ability to comprehend changes on the sidechain [33]. A sidechain is referred to as a side blockchain that can interact alongside the mainchain. A mainchain regards a sidechain as an elongation of itself. In addition, they can be sidechains of each other [34] [3]. Figure 6 shows how sidechains and mainchains are connected.

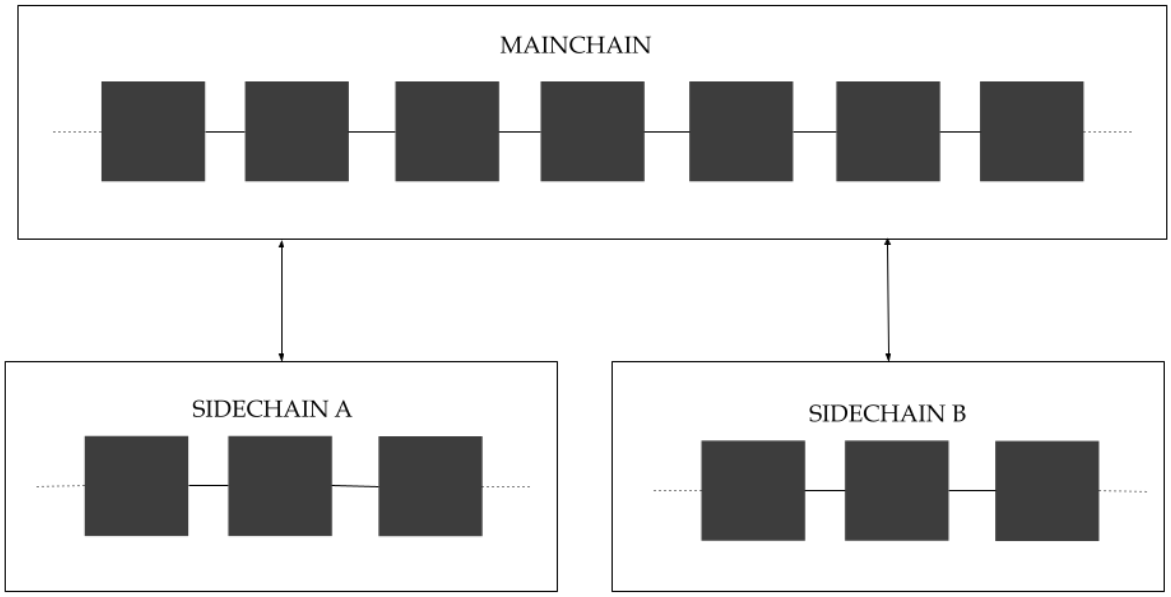


Figure 6: Sidechain/Relay

Hashed Time-Lock Contracts: Hashed Time-Lock Contract is an approach to accomplishing atomic cross-chain operations. The technique includes the use of timelocks and hash locks [34]. It furnishes atomic transactions between parties by committing the trader to provide cryptographic proof before the timeout when making the transaction. With this feature, the blockchains require to know considerably less about each other [33] [3]. A simple visualization of how hashed time-lock contracts work regarding interoperability is illustrated in Figure 7.

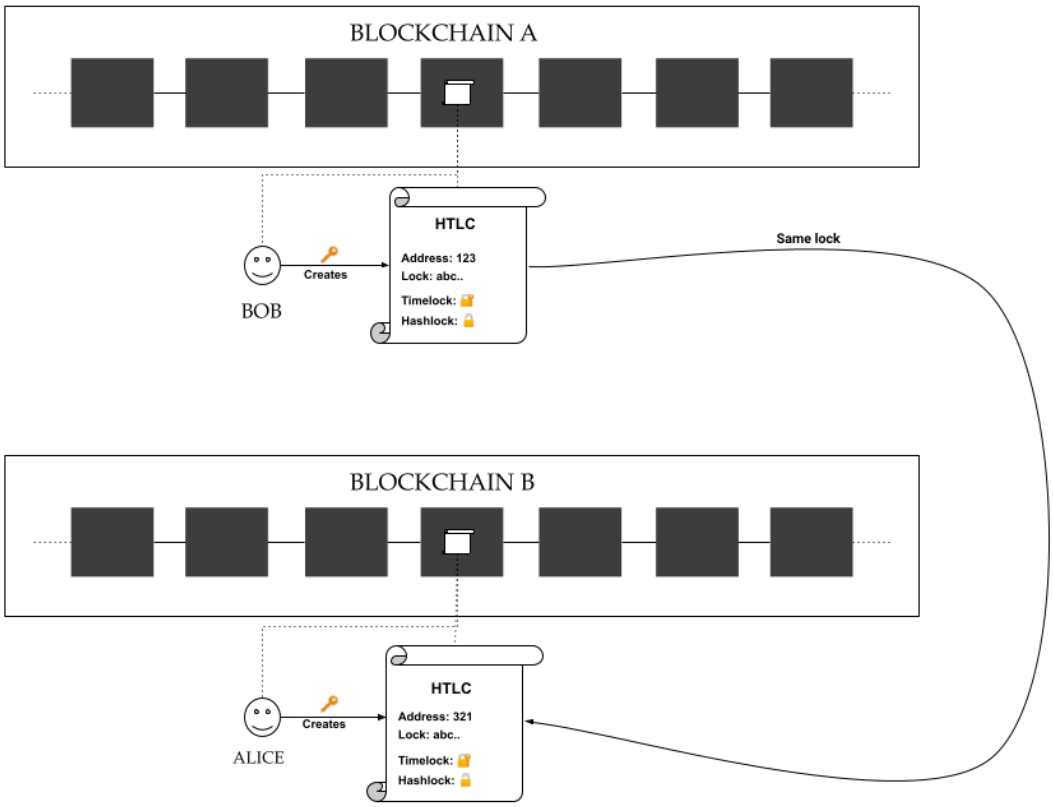


Figure 7: Simple Hashed Time-Lock Contracts illustration

As the figure illustrates, Bob firstly creates a hash-time lock contract on blockchain A. The contract contains information needed to execute the transaction, such as an address, amount, lock, hash lock, and timelock. Then, using a hash function, Bob generates a lock based on a picked secret. Bob then deposits the amount Alice and Bob agreed to exchange and sends the lock to Alice.

The contract enforces two possible outputs. The timelock ensures that the amount gets refunded and returned to Bob if nothing happens in a given time. If Alice can provide the secret, the contract automatically executes and transfer it to Alice´s address.

Similarly, Alice creates a contract on Blockchain B using the same lock Bob sent to her. Bob can now use the secret to unlock the hash lock in the contract created by Alice. Due to public operations on the blockchain, Alice can see the secret and unlock Bob´s contract. Without trust involved, both contracts get executed supported by hash locks and timelocks that work as security mechanisms.

3 Related work

While there has been a lot of attention towards blockchain technology and its security issues, relatively few contributions in the field focus on threat modeling and threats regarding interoperable blockchains. From our research in the field, there are no existing threat modeling tools capable to share and categorize threats regarding blockchain. Potential threats disclosed by existing threat modeling tools are related to applications and general IT-systems. In this section we will disclose the research already presented in both detailing different threats in regard to blockchain and existing threat modeling tools.

3.1 Studies summarizing blockchain security threats

Showcasing the potential threats within any IT system is an important step in maintaining security. However, it is not the only important step. The process of labeling threats into categories of potential attack locations/origins is necessary for further analysis [35]. This step is also important in blockchain technology systems. However, in this field of work, the current research mainly focuses on presenting the threats, often with descriptions, but there is a lack of research focusing on labeling the threats into categories and sub-categories for further threat analysis and knowledge building.

Blockchain systems is an attractive target for cybercriminals [36] [37]. Guggenberger et al. [36] argue that it is because of the high value within the traffic of the respective systems. The same research also presents a systematic literature review displaying multiple potential attacks on the different features of the blockchain stack, similar to our presentation of the blockchain layers in Figure 3. Despite the fact that [36] presents a broad literature review of 161 primary papers and lists the threats resulting from their analysis, the research fails to both describe and analyze the resulting threats.

In a research done by Sumit et al. [38] in 2019, they produced a security analysis on blockchain systems where they listed multiple threats formed by a literature review. Similar to the research by Guggenberger et al. [36], they fail to perform any analysis other than describing the threats and their potential attacking points.

In the research done by Conti et al. [39], they conducted an extensive survey on privacy and security issues in Bitcoin. They produced detailed research on the vulnerabilities found. However, they only focused on Bitcoin and therefore limiting their scope to a fraction of the field. Similarly, in a research by Samreen et al. [40] in 2021, they performed a literature review on vulnerabilities in Ethereum Smart Contracts. While giving important contributions to the field by providing state-of-the-art knowledge, it is limited to only including vulnerabilities within the Ethereum Smart Contracts.

As mentioned above in the article by Patrick Mallory [35], threat categorization is an important aspect of security analysis. In 2020, Shrivas et al. [41] did a research having threat categorization as their primary focus. According to the authors, this was an important contribution to the field of work. They listed threats from a literature review and categorized them into 6 separate categories. "Blockchain Runtime Environment Threats", "Communication Protocol Threat", "Consensus Protocol Threat", "Smart Contract Threat", "Cryptographic Threat", and "Blockchain Services Threat". These categories defined the scope of the research and are a step forward when analyzing security issues within blockchain. The shortcoming of the research by Shrivas et al. [41] is the lack of narrowing the threats into smaller sub-categories as a result of pinpointing the issues. Their amount of threats is also rather limited, and they do not provide any information about their process of conducting the literature review. A similar research was presented by Jamal Hayat Mosakheil in 2018 [42] where he classified different blockchain threats where his primary focus relies on blockchain 1.0 and blockchain 2.0. The paper is also limited by the focus of the consensus model. From the result of the research, Mosakheil [42] presents mainly issues related to PoW and Bitcoin. In contrast to the research by Shrivas et al. [41], Mosakheil [42] presents the layers of his focus and is able to label each threat to the layers defined within his scope of the research.

3.2 Threat modeling theory and approaches

Understanding the potential threats a business or system can face when implementing certain features is key for building a secure product. This is possible with threat modeling. The main job of a threat model is to identify, communicate and provide information about the potential threats [43].

The Open Web Application Security Project (OWASP) details the threat modeling process in a three-step process [44]:

- Decompose the Application
- Determine and Rank Threats
- Determine Countermeasures and Mitigation

Step 1: Decompose the Application related to information gained about the application itself. This is an important step where threats can be found in separate individual features.

Step 2: Determine and Rank Threats relates to the identification of threats. OWASP suggests using a threat categorization model such as STRIDE or Application Security Frame (ASF) in this step.

Step 3: Determine Countermeasures and Mitigation relates to the possible actions one should take based upon the threats already determined. A system's vulnerability might have mitigations built from a mapping between countermeasures and the threat itself. From a business perspective, the consequence of risk has to be evaluated. OWASP [45] describes three possible options for addressing these risks:

- **Accept:** decide that the business impact is acceptable
- **Eliminate:** remove components that make the vulnerability possible
- **Mitigate:** add checks or controls that reduce the risk impact, or the chances of its occurrence

The benefits of utilizing threat modeling throughout the development phase yield a more straightforward decision-making process for all security-related implementations as all the information is collected through the threat modeling process. It also generates an assurance through well-documented evidence. We, hereby, will introduce some typical threat modeling approaches.

3.2.1 STRIDE

STRIDE is a threat modeling approach proposed by Microsoft for categorizing threats to build networks, applications, and systems that will be secure by design. The framework presents six different types of threats: **S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service and **E**valuation of Privilege [46].

- **Spoofing:** the attacker pretends to be someone else
- **Tampering:** the attacker modifies legitimate information in transit or at rest
- **Repudiation:** the attacker disowning actions executed, cannot be traced back
- **Information disclosure:** the attacker gets unauthorized access to confidential information
- **Denial of service:** the attacker disrupting services for legitimate users
- **Evaluation of Privilige:** the attacker gets higher privilege access and can perform actions as unauthorized

STRIDE threat modeling analyzes threats on different components. It is a well-known threat modeling approach for security experts where the main goal is to break down all processes and data flows in a system to help reason and discover vulnerabilities [47].

3.2.2 DREAD

DREAD is a threat modeling approach developed by Microsoft similar to STRIDE and is described in detail on the website of OWASP [48]. DREAD is a model for ranking the different threats. This is done by giving points divided into five different categories.

- Damage potential
- Reproducibility
- Exploitability
- Affected users
- Discoverability

For each category, you can give points to the threat from 0 - 10, where 10 is the most significant. The overall result of the DREAD model is produced by finding the average from the points.

3.2.3 Attack trees

Bruce Schneier first described threat modeling utilizing attack trees in Dr. Dobb's Journal in 1999 [49]. In this journal, Schneier describes attack trees as a formal way of achieving information about the security of systems. Attack trees represent attacks where each node has an informative meaning. The root node indicates the goal of the attack and the leaf node describes the different ways of achieving that goal. Hierarchical nodes that are not leaf nodes or the root, are subgoals of the root. Each node is also represented by an "AND"/"OR"-relationship. If a node is a type of "AND", all of the children nodes need to be met in order for the parent to fulfill its purpose. If a node is of type "OR" at least one of the children needs to be met.

A simple example of an attack tree showing a potential attack path of obtaining an administrator password is shown below in Figure 8. Here the type of node is displayed by AND/OR - gates with corresponding paths.

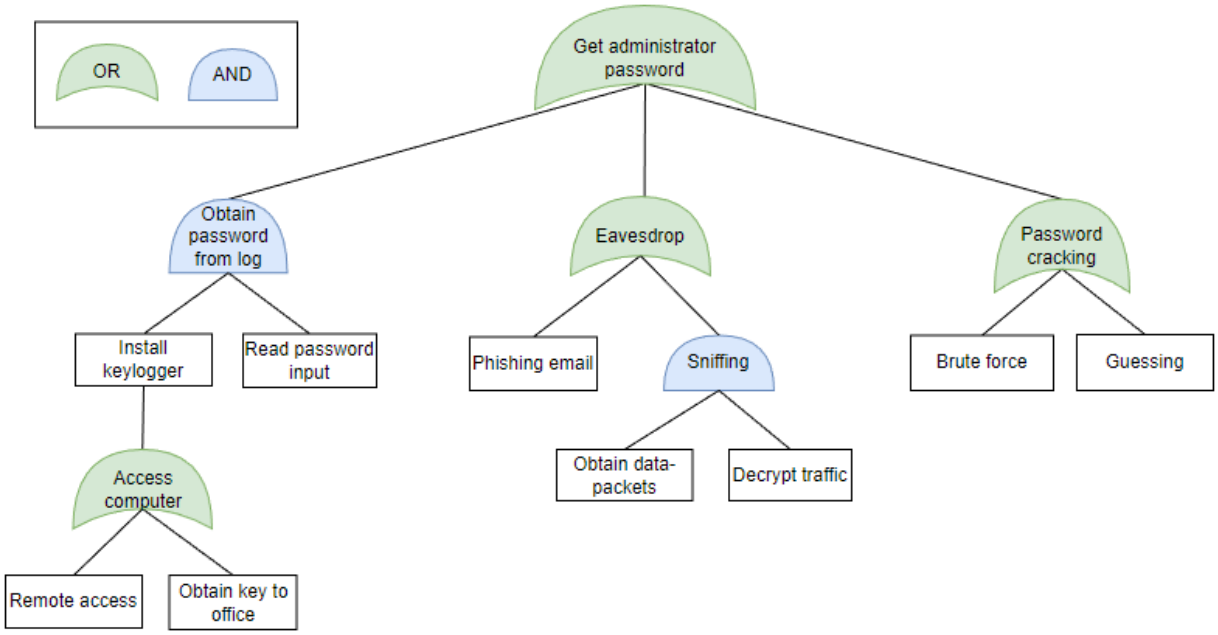


Figure 8: Example of an Attack Tree

3.2.4 Fuzzy Logic

Fuzzy logic (FL) has its basis in the fuzzy set theory, which revolves around placing objects in a set with members of different degrees [50]. This means objects or classes of objects do not have clear and defined boundaries such as 0 or 1, but a degree between true or false. An example provided by Mathworks describes fuzzy logic by using the days of the week and weekends [51]. There is no question that Monday, Friday, and Saturday are days part of the week. However, when asked if Friday is part of the weekend, it gets fuzzier. To give a variable a degree of membership in a set, a membership function is applied (often called fuzzification [52] [53]). This function produces a membership value between 0 and 1. However, the function can be customed to your benefit.

In order to use fuzzy logic in any sense, the fuzzy inference process is used to map an input to output through fuzzy logic. The process revolves around four steps [54]:

- **Fuzzification:** This is process where inputs are transformed to a membership in a fuzzy set by utilizing a membership function.
- **Rule Evaluation:** After the fuzzification you know the degree of the antecedent of each rule. If the antecedent has more than one value, a fuzzy operator (AND/OR) is applied to result in on single number representing the rule antecedent [55]. In turn the input of the rule evaluation is two or more members of the fuzzy set, and the output is a single digit.
- **Aggregation:** In the third step you will unify all outputs from all rules built from the Rule Evaluation. The output of this aggregation is a one set of fuzzy every output.
- **Defuzzification:** In the fourth and final step, your goal is to transform the aggregatet set of fuzzy outputs to a single number making it clear. There exists many methods for defuzzification, however the most common one is centroid calculation.

In order to use fuzzy logic, Matlab has a framework called fuzzy logic toolbox with built in function for this type of support [56].

3.2.5 T-Map

T-Map is a framework for threat modeling where the main feature is attack path analysis [22]. T-map builds heavily of the work of Bruce Schneiers attack trees [49] as described above in section 3.2.3. However, this model helps give the attack paths weights for further analysis. The framework consists of 3 important observations [57]:

1. The more security features that are unattended within an IT-system, the more insecure it is.
2. The importance of features for different IT-servers might vary to fulfil the business's core values.
3. With more exposed vulnerabilities that could be of motivation for a malicious actor, the risk becomes bigger for a potential attack.

In order to score the attack paths to give them weights, one attack could be more severe in one business while less severe in another. This is due to the numerical weight of the attack given by the formula:

$$\text{Risk} = \text{Probability} * \text{Size of Loss}$$

Here the size of loss could differ from business to business.

Corresponding to each attack path created while conducting the analysis using T-MAP, 4 types of class diagrams are used to create the nodes in the attack tree and weighted ratings [22]. The 4 classes of UML are:

- **Access:** details how the attack gains system information. For example by infiltrating the communication layer.
- **Vulnerability:** details the vulnerability the attack utilises.
- **Target Asset:** details what software, which computer and/or which server obtains a vulnerability.
- **Affected Value:** details the business value under attack. For instance productivity or reputation.

From these classes the weighted ratings are formed. However, the authors of "Value Driven Security Threat Modeling Based on Attack Path Analysis" [22] details that the ratings can vary from business to business depending on their business values.

3.3 Existing threat modeling tools

Existing application threat modeling tools can detect potential threats in the development phase for general IT systems and are a vital tool for any business developing security-intensive applications.

3.3.1 Microsoft threat modeling tool

A popular existing tool is the Microsoft threat modeling tool building on the Security Development Lifecycle (SDL) developed by Microsoft [20]. This threat modeling tool shows potential threats connected to the system architecture and/or data flow portrayed by the user. This tool utilizes STRIDE for categorizing the different threats and DREAD for ranking the threats. All threats are accumulated from the data flow diagrams (DFD) and architecture drawn by the user. These profiles and results can be stored for further analysis and compared to other reports done in the same tool. The inputs needed to form an analysis and produce threats are the DFDs provided by the user.

3.3.2 SecureITree

SecureITree, developed by Amenaza [21] is software providing the user the ability to build and populate attack trees based on their AND/OR logic. The software is a Java application capable of running on Mac, Linux, and Windows computers. The nodes of the trees are easily linkable and show the possible combination of the leaf nodes forming the path of the problem. The profiles of each attack tree can be stored and analyzed further in a separate analysis. Saini et al. [58] did research on threat modeling using attack trees where they looked into SecureITree. They described the tool as a helpful tool to draw forth complex conclusions about the security of a system. However, they also mention that attack trees have no guarantee of completeness as an attacker profiling is a "best-guess" scenario, meaning there is information based on assumptions. They also mention that detailed trees from larger systems can have a significant cost from both building and maintaining the attack trees. This tool simplifies the process of building attack trees. Therefore, the inputs are the node specifications of all the nodes wanted in the tree. Meaning name, cost, and node-type (AND/OR). From this, the tool can produce an analysis of the attack path.

3.3.3 Fuzzy Logic tools

The tool providing automation of the fuzzy logic threat modeling technique described in section 3.2.4 is called MATLAB Fuzzy logic toolbox [56]. This toolbox is not software, but a MATLAB framework that allows the user to model complex systems. Sodiya et al. [52] utilized the fuzzy logic toolbox to perform threat modeling, indicating the ability to identify threats based on fuzzy logic. Their technique uses fuzzification on the clear inputs based upon the categories from STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and/or Elevation of Privilege). Albeit, newer threat modeling tools have been developed prior to this study, they provided an efficient threat analysis of 5 different systems.

In 2017 Alali et al. [53] utilized the Fuzzy Inference System provided by the MATLAB toolbox to create a viable threat model to improve the cybersecurity of IT systems. However, they do not include a comparison to other types of threat modeling tools. For their model they have four clear inputs: *Very high*, *High*, *Medium*, *Low*, *Very low*. If you compare it with the model used in the research done by Sodiya et al. [52] they have different inputs, be that as it may, fuzzy logic needs a clear variable as input.

3.3.4 CORAS

CORAS is a graphical threat modeling language that has its basis in UML diagram. CORAS is scenario-specific, meaning it profiles each potential threat as a use case which gets analyzed from self-written diagrams [59]. The tool related to CORAS is an editor [60]. This tool is rather limited in contrast to other types of threat modeling tools as it only provides the ability to draw diagrams for further analysis. The input used to conduct an analysis using Coras tool is the UML produced by the user.

3.3.5 Tiramisu

Tiramisu is a threat modeling tool developed at the University of Southern California and described in detail in the paper "Value Driven Security Threat Modeling Based on Attack Path Analysis" by Chen et al. [22]. The basis of the tool is T-MAP, described in section 3.2.5, and functions as automation for attack path analysis. The inputs of tiramisu are listed below:

- Information about the vulnerability
- Infrastructure of the IT-system
- Dependency between infrastructure and business values.

Together with the inputs from the user, Tiramisu has a layer called Automated Data Collecting Engine, which collects all published attacks from "CERT/CC, NIST, SANS, SecurityFocus, Symantec, and Microsoft websites" automatically [22]. Tiramisu will output attack paths from already existing attacks corresponding to the inputs from the user

Table 1: Existing threat modeling tools and their techniques

Tool	**Threat modeling approach**	**Blockchain Adoption**	**Reference(s)**
Microsoft threat modeling tool	STRIDE, DREAD	No	[20]
SecureITree	Attack Trees	No	[21] [58]
Fuzzy Logic	Fuzzy Inference, STRIDE	No	[56] [53] [52]
Coras	UML	No	[59] [60]
Tiramisu	T-Map	No	[22]

The threat modeling tools described in this section, as shown in Table 1 do not include threats regarding blockchain technology. In the tools where the user is able to build the architecture and data flow of their own system, they could potentially produce a system with similar architecture and data flow that incorporates blockchain technology. However, the threats connected to the different features in that system would be a result of security issues in relation to general IT systems. The tools presented in this section are all built around security management, whether analyzing attack paths and/or presenting potential threats to a system. This is done by having a set of inputs. Table 2 below shows the different inputs and outputs from each model.

Table 2: Inputs and outputs from the threat modeling tools

Tool	**Input(s)**	**Output**
Microsoft threat modeling tool	Data Flow Diagram	Potential threats
SecureITree	Node spesifications	Attack Path analysis
Fuzzy Logic	Clear variables	Membership degree
Coras	UML	Case analysis
Tiramisu	Vulnerability information, IT infrastructure, Infrastructure and business value dependency	Attack path analysis

With the increased interest in using blockchain as a distributed ledger in businesses and software, the necessity of providing security systems that incorporate blockchain technology also increases. To our knowledge, no existing threat modeling tool has any adaptation for threats regarding blockchain technology.

4 Research design

4.1 Research motivation

Blockchain technology is a hot topic for businesses and researchers around the world. Security challenges and concerns arise naturally, and we expect interest to rise appreciably in the future. Previous scientific literature [18] [19] [38] covers most of the security challenges in blockchain systems. Still, there is a lack of research on specific layer one threats and how to integrate these into threat modeling tools. Additionally, existing threat modeling tools for information technology systems do not cover blockchain technology, and their applicability needs to be investigated.

The motivation for conducting this research is the rapid growth of the technology and the lack of research. In a multivocal literature review conducted by the authors, as shown in appendix B [3], the study mapped the theoretical coverage of security and privacy challenges in blockchain interoperability. Based on the MLR, we believe our research will positively contribute to the field by pinpointing specific security issues related to blockchain and blockchain interoperability in our created threat analysis plugin.

4.2 Research questions

Our main objective is to conduct a literature review on blockchain security issues in layer one. Additionally, we want to incorporate these issues and interoperability threats into existing tools to determine security threats. Thus, this thesis focuses on two research questions.

RQ1: What are the most important characteristics of vulnerabilities related to blockchain?

RQ2: How to develop a tool to facilitate blockchain threat modeling by incorporating the blockchain vulnerability characteristics?

RQ1 aims to identify and analyze threats of blockchain mentioned by researchers in scientific literature. **RQ2** aspires to integrate the analysis found in the first research question and security challenges encountered by authors MLR [3] into existing threat modeling tools created as a plugin.

Scope of the research: The research conducted here will be limited to the lower layers of blockchain shown in Figure 3. This is due to the lack of existing research regarding threat modeling integration with blockchain structures and the time frame of our research. Even though we have our primary focus on the lower layers, threats shown at a higher layer can have an origin from lower layers and should therefore be disclosed.

4.3 Research design to answer RQ1

The paper follows an instructional process to answer the research question correctly and provides a firm foundation for the research topic. The instructional process is established by the literature [61] [62]. To answer research question 1, we conducted a literature review. The following section presents the search strategy and terms used to include relevant literature and how the papers were selected.

In order to conduct our review, we created a flowchart that establishes all steps and strategies involved in our search and evaluation, shown in Figure 9. All steps are described in the sections below.

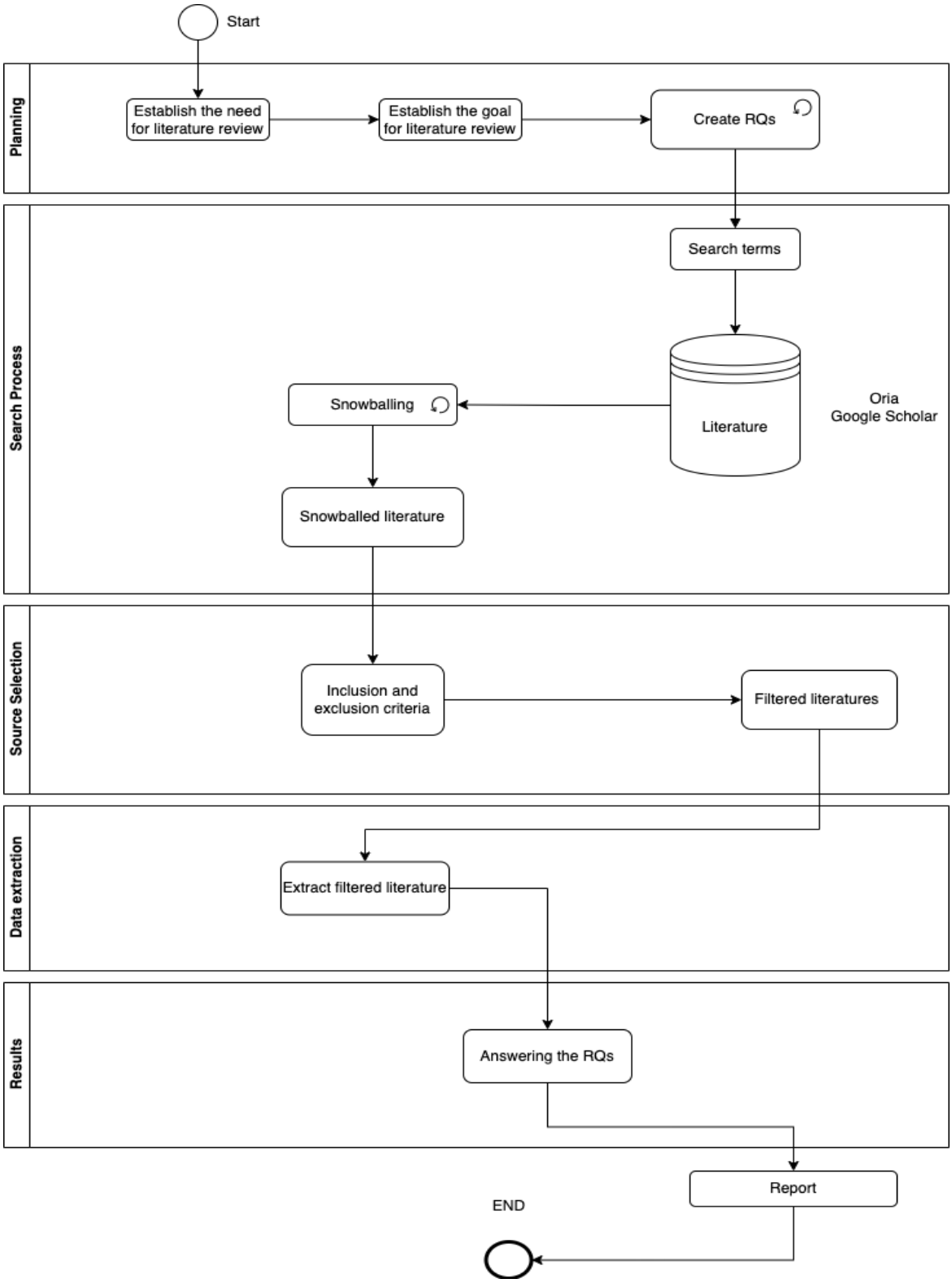


Figure 9: Search Process

4.3.1 Search strategy

For this research, we applied a strategy consisting of building search terms used in a specific domain. Furthermore, we made restrictions to the search and dynamically selected the resulting surveys based upon title, abstract, and content. After this process, we extracted the data systematically by categorizing the threats found in each survey.

4.3.2 Search terms

Relevant keywords within the field of blockchain threat domain formed the basis for our search terms. In addition, our results had to be made up of different surveys within the same field, making "survey" an important term to utilize. Finally, we defined the search string as follows shown in Table 3:

| X1 **AND** (Y1* **OR** Y2* **OR** Y3*) **AND** Z1 |

X1. Blockchain	**Y1.** Threat **Y2.** Vulnerability **Y3.** Issue	**Z1.** Survey

Table 3: Search Terms

4.3.3 Search Modules and Restrictions

The searches for the literature review were all done in Google Scholar [63] and Oria [64]. This is popular search engines for scientific research as it includes many digital databases for literature such as ResearchGate, IEEE, Science Direct, and Xplore. These databases hold most of the resources within the current field and are accessible to us through NTNU. We limited the resulting papers by utilizing Google Scholar's and Oria's page rank. This was set to 8 as the resulting literature on higher page ranks included less interesting content related to our search terms.

4.3.4 Inclusion and Exclusion

The surveys found from the literature search need to contribute to our field with content providing necessary information to answer research question 1. For that reason, a table of inclusions and exclusions was made to specify important factors in the literature, making the selection process more systematic. The Table 4 will keep track of inclusions that indicate important criteria for answering research question 1 and exclusions that detail unsatisfactory literature for our study.

Table 4: Inclusions and Exclusions criteria for our surveys

Inclusion	**Exclusion**
Includes privacy/security issues within blockchain	Not about blockchain
Introduces description about to the vulnerabilites	Only includes title of the vulnerability
Vulnerabilities are supported by data	No references
Objective	Biased
Conclusion is objective	Fulltext/Abstract is not available
Survey has a clear literature review	No author

Our literature should consist of surveys having a clear literature review with references to their findings. It is essential that all the vulnerabilities found in each survey regarding blockchain are detailed with a description and not just the name of the vulnerability. This is due to the data being applied for research question 2. If the survey does not include a description, it has to have an apparent reference to a description of that particular vulnerability. The credibility of our literature review is essential, and by applying Table 4 of inclusion and exclusion, the data applied to our application for research question 2 will be strengthened.

4.3.5 Selection process

The next step in the search strategy is a selection of relevant literature. We were looking for surveys that included threats regarding general blockchain. This was done dynamically, meaning while searching, we read the abstract of each survey, and if they seemed to include interesting findings, we kept reading the content. If the survey listed several threats, it was included and part of our literature.

4.3.6 Data extraction

After selecting the literature with meaningful content, the surveys were thoroughly examined and the content extracted. This was done in correlation with the statements for inclusions and exclusions shown in Table 4 above.

The extraction process followed the following iterative process:

1. Extract from the same sources
2. Both researchers working for this thesis extract data (threats and categories) independently
3. Store extracted data in a spreadsheet
4. Compare extracted data
5. Seek consensus about the data
6. Document the results in a spreadsheet

All the threats found in the different surveys were systematically entered in a spreadsheet with appropriate categorizations. Further details can be found in 5.1.1. For the categorization we opted to analyze the surveys collected and see how they portrayed the different threats.

4.4 Research design to answer RQ2

We established design science research (DSR) as a methodology for this study to answer this research question precisely. DSR is a problem-solving paradigm aiming to enhance science and technology knowledge by creating innovative artifacts. The DSR approach aims to develop knowledge of how something can and should be created or designed to acquire a selected set of goals. The overall goal of using DSR is to solve problems and improve the environment [65]. DSR provides a valuable guideline for evaluating the research project implemented as a plugin, and it focuses on developing new artifacts that solve identified problems [66]. In order to accomplish a proper DSR process, it is necessary to identify and understand the different cycles mentioned by A.Hevner and S.Chatterjee [66], as illustrated in Figure 10; which shows our study in the context of the DSR process.

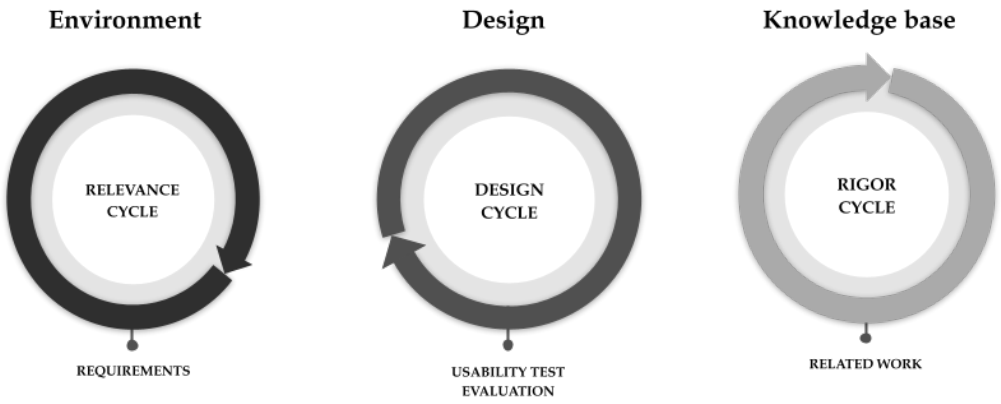


Figure 10: Design science research cycle

The design science research approach does not provide any detailed description of how to conduct DSR. However, the process follows an iterative process. Here, we will go through each cycle regarding our work and identify these three cycles illustrated in Figure 10.

The research process started with analyzing the relevant literature on existing threat modeling tools for information systems. Existing threat modeling tools existed; however, a lack of research and tools relevant to blockchain technology was specified. Based on related work in the field, characteristics of threat modeling were identified, which produced a set of requirements for implementing threat modeling for blockchain as a plugin. These requirements and the study of RQ1 can be placed in the *Relevance Cycle*. The relevance cycle identifies problems that can improve the environment. This cycle specifies requirements (the opportunity/problem to be addressed) for the research and defines acceptance criteria for evaluating the result. The overall goal is to determine the application context and define the requirements for the artifact [66]. Our used knowledge base for researching existing systems can be positioned in the *Rigor Cycle*. The rigor cycle ensures the grounding of the research while adding new knowledge to the existing knowledge base. This cycle guarantees that the produced artifact contributes to the study based on existing knowledge [66].

Based on the outcome from the relevance and rigor cycles, it established the starting point for developing and designing a digital prototype in the *Design Cycle*. The design cycle emphasizes the artifact's design, development, and evaluation, which can result from the relevance cycle and rigor cycle [66]. The study of RQ1, requirements and results of the author's MLR [3] was used to produce a final prototype of a threat modeling tool for blockchain technology. The data provided by [3] was implemented to improve the scalability of the application where it should be possible to locate threats regarding blockchain interoperability. In order to evaluate the prototype, usability testing with open-ended questions where performed. Open-ended questions were conducted as a survey to identify strengths, weaknesses, and possible features. A qualitative data analysis approach to the data generated was utilized. In order to analyze the data sufficiently, a thematic analysis was conducted [67]. Using a thematic inductive approach provides flexibility to modify for our needs. It is a bottom-up analytic strategy where identified vital themes emerge from raw data through repeated comparisons and examinations. Coding and analyzing the data using this approach produce insightful and trustworthy findings [67].

5 Research implementation and results

This section covers the implementation process and the results of the thesis conducted. Our findings derive from a strategic search process of existing literature on security issues regarding blockchain and threat modeling tools. For research question 1 we will only introduce the results as the procedure of the literature review is described in section 4.3. For research question 2, we will first introduce the implementation process and then the results.

5.1 Research question 1

Security and threats in blockchain technology are well-known terms in the literature. In order to provide a tool for developers and researchers to model threats in the technology, the threats need to be extracted from the literature and categorized.

Selected papers contribute to the field, and the following section will present the results from the search process and how we categorized each threat to build the prototype.

5.1.1 Results

This section describes details surrounding our choices when applying the data from our literature review to the application. All the threats from the surveys had to be categorized to make it easier for the user to display the vulnerabilities within their systems. Deciding on what threats to be added to our database and/or merged whether they describe a similar issue is also a decision we had to make.

5.1.1.1 Paper searching results The search process is described in detail in section 4.3. We applied the search strings to the search engines resulting in a mixture of existing surveys, scientific papers and other types of literature. However, not all of them were relevant for our research. In order to select the most relevant papers, we relied on the page rank algorithm. This procedure will mark our research scope and hopefully provide a holistic overview of the research space. To ensure that we filter out relevant papers that provide necessary information to answer research question 1, we applied the inclusion/exclusion 4.3.4 on retrieved papers from the search. In the following process, selection and data extraction were conducted in the same procedure. As a result, 20 relevant papers from our search became our primary papers to answer research question 1 accurately.

The generated list of primary papers are listed in Table 5.

Table 5: Primary papers

Title	Reference
A survey of blockchain security issues and challenges	[68]
A survey on the security of blockchain systems	[19]
A survey of blockchain from security perspective	[69]
A Survey on Security and Privacy Issues of Blockchain Technology	[70]
A Comprehensive survey on Blockchain: Working, security analysis, privacy threats and potential applications	[71]
Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions	[72]
On Blockchain Security and Relevant Attacks	[73]
Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network	[74]
Exploring the attack surface of blockchain: A systematic overview	[75]
Blockchain Security Attack: A Brief Survey	[76]
Blockchain Attacks, Analysis and a Model to Solve Double Spending Attack	[77]
A survey on blockchain cybersecurity vulnerabilities and possible countermeasures	[78]
A Survey of security threats and defense on Blockchain	[79]
Detecting blockchain security threats	[80]
Blockchain Technology: Methodology, Application and Security Issues	[81]
A survey on privacy protection in blockchain system	[82]
A survey on blockchain systems: Attacks, defenses, and privacy preservation	[83]
Exploring the Attack Surface of Blockchain: A Comprehensive Survey	[37]
Review of blockchain technology vulnerabilities and blockchain-system attacks	[84]
The Risks of the Blockchain A Review on Current Vulnerabilities and Attacks	[85]

In order to improve the overall security of blockchain technology, there is a need to collect and gather information about the challenges the technology faces. The primary papers indicate the vulnerabilities found by using the technology, but none mention how to use this information for further analysis.

When analyzing the primary papers, different threats and vulnerabilities were identified. A spreadsheet was created to keep track of all threats mentioned by researchers. This spreadsheet is supplemented as an attachment to this report [86] due to the large number of threats identified. The final result of the search process yielded 73 different threats mentioned and 18 vulnerabilities extracted from the author's previous MLR [3] regarding interoperable blockchains, found in appendix B.

In addition to identifying threats from our survey papers, we extracted their used categories. For each survey, we extracted their categories and placed them in a spreadsheet with number of occurences. From that, we can see what categories were most used.

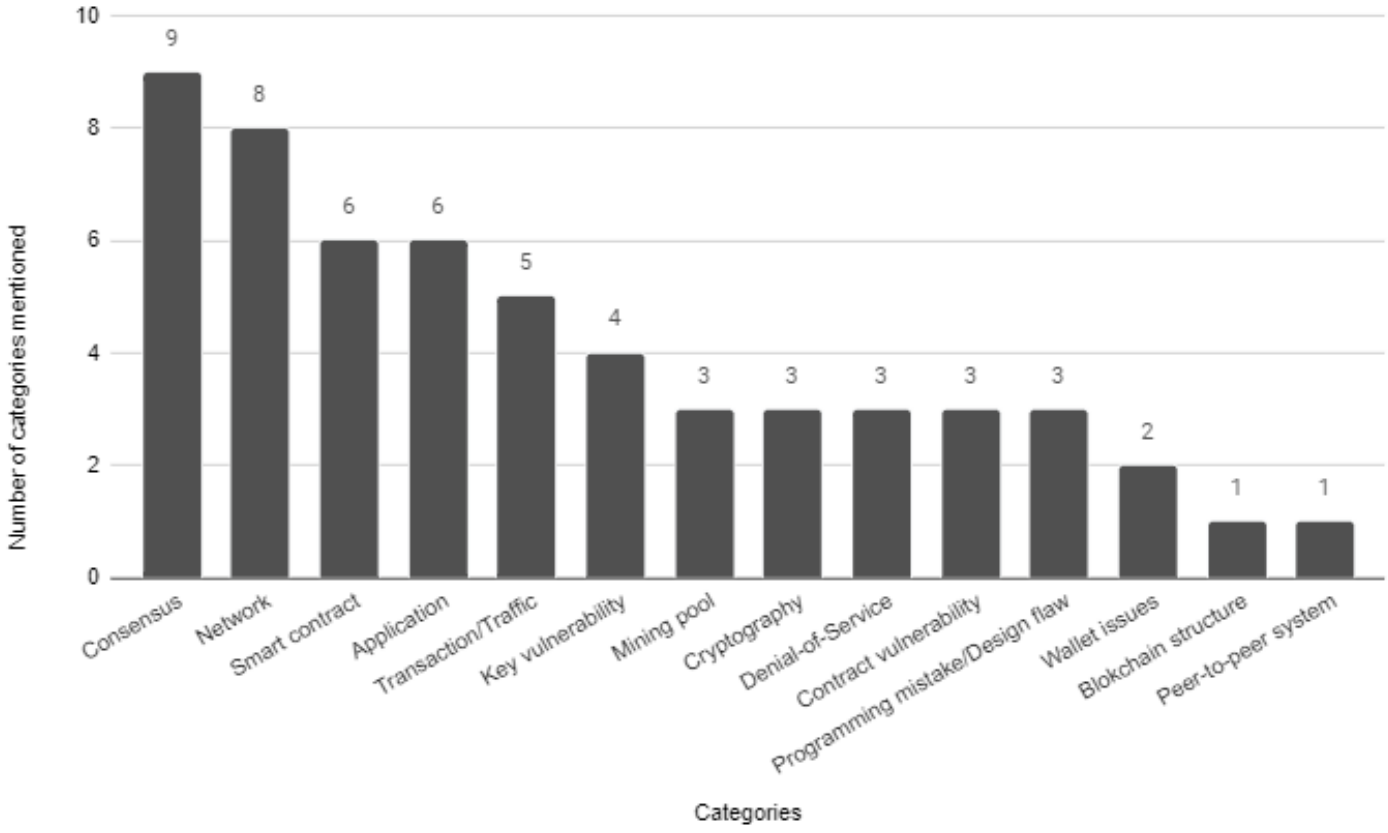


Figure 11: Extracted categories

As you can see from the Figure 11 above, "Consensus" and "Network" were most used, with "Application", "Smart contract", "Transaction/Traffic", and "Key vulnerability" following thereafter. One important aspect worth noting is that there is a need of removing redundant categories as best as possible. This is for creating a more user friendly application where threats are not to be repeated. For this we merged some of the categories.

"Application" and any categories related to smart contracts or human error were merged into "Application/Human error". Any categories related to the creation of blocks or finding of blocks were merged into "Block creation". Our categories after the process of merging which are applied to our application is as shown:

- **Consensus:** Extracted threats targeting or has an origin in the consensus mechanism of the blockchain technology.
- **Network:** Threats directly targeting one or multiple nodes of the network or has an origin in the network layer of the blockchain.
- **Cryptography:** Threats linked to the cryptographic technology of the blockchain. We limited the scope of our research to having cryptographic measures set to true or false. This was done in order to decrease the complexity of the application produced for research question 2 where threats regarding cryptography would only be shown if no measure against such attacks was found.
- **Application/Human Error:** Threats targeting or has an origin in the application layer of the blockchain. This is often related to code-errors related to contracts or applications on the blockchain technology.
- **Transaction:** Threats targeting or has an origin in the transaction between nodes of a blockchain.
- **Block creation:** Threats targeting or has an origin in the blocks of the blockchain.

Each identified threat was assigned with a description and ordered into a category supported by the literature. After categorizing the different threats, Figure 12 was produced to show the number of threats connected to each category.

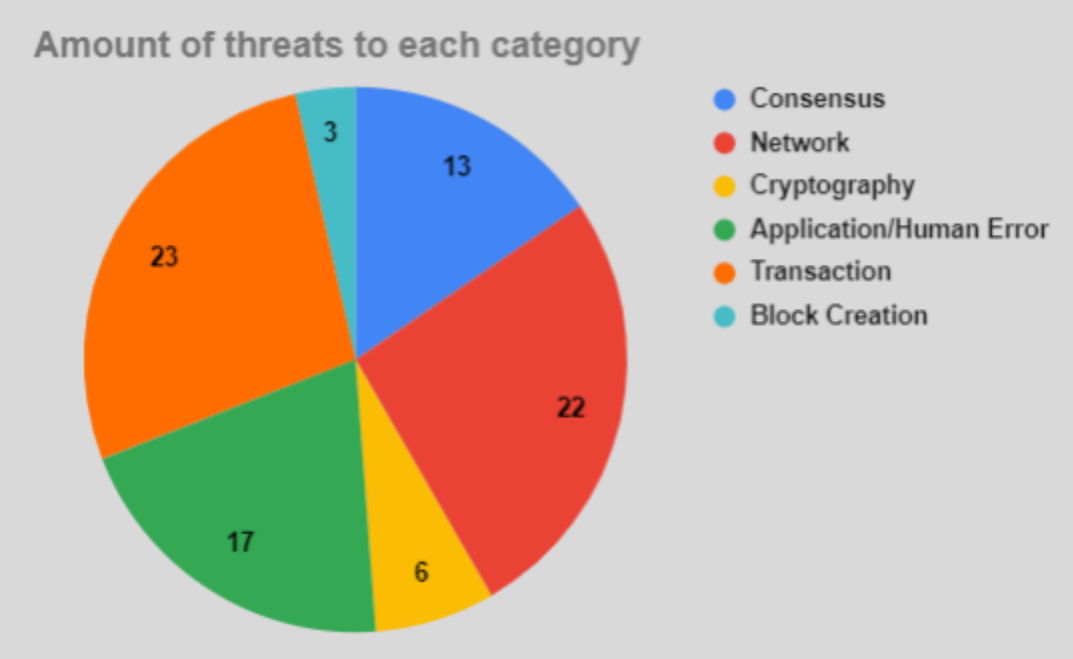


Figure 12: Search Process

The most used categories from our primary papers are “Consensus” and “Network”. This implies that there are more threats connected to those categories as well. This argument forms the number of threats for each category. However, we merged every threat related to human error, application, and smart contract to “Application/Human Error”. Due to the process of merging “Application/Human Error” has more threats connected to it than “Consensus”, even though the latter category is more referenced in our primary papers.

The number of occurrences in the literature can provide a holistic overview of all threats related to blockchain technology. Figure 13 illustrates the number of each threat mentioned in the survey papers covered by the literature review.

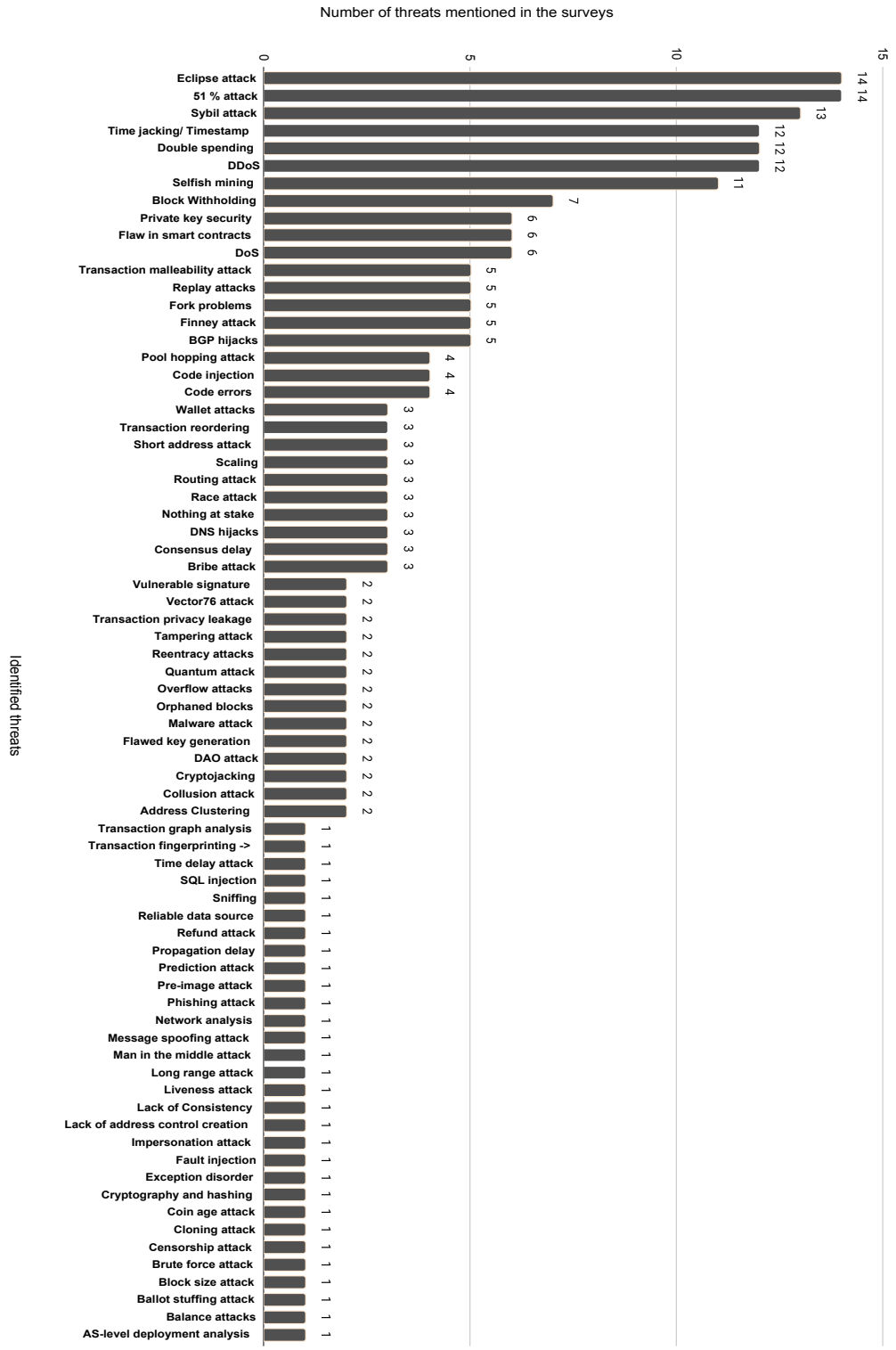


Figure 13: Count occurrence of the threats mentioned in the survey papers.

The three most mentioned attacks are "Eclipse Attack" and "Sybil Attack" targeting nodes in the Network and "51% Attack" targeting the consensus of the blockchain. This is as expected due to the occurrences of categories seen in Figure 11, proving that "Consensus" and "Network" are highly researched. Of the 73 threats found in the primary papers, there are some threats that can have an origin in multiple categories. For instance, "Denial of Service" is a threat that can be targeted on nodes in the network, but also systems regulating transactions or smart contracts. Table 6 shows the threats with their redundant categorizations.

Table 6: Threats with redundant categories.

Threats	**Categories**
Fork problems	Consensus
	Network
	Block creation
Scaling	Consensus
	Application/Human Error
Time jacking	Cryptography
	Block creation
Denial of Service	Network
	Application/Human Error
	Transaction
Distributed Denial of Service	Network
	Application/Human Error
	Transaction
Balance Attacks	Network
	Transaction
Replay Attacks	Network
	Cryptography
Cryptojacking	Cryptography
	Application/Human Error
Wallet attacks	Network
	Application/Human Error
Sniffings	Network
	Transaction

5.1.1.2 Matching threats: In order to build a prototype where the user can quickly discover threats regarding blockchain technology and interoperable blockchains, there was a need to split consensus and network categories into sub-categories. For example, the user is more interested in which threats are related to their consensus mechanism, not all threats related to the consensus category. As a result, the following section will present tables that show how we matched different category-related threats with each sub-category.

Table 7: Matching security threat with consensus

Consensus threat	**Proof of Work**	**Proof of Stake**	**Practical Byzantine Fault Tolerance**
51% attack	x	x	x
Fork problems	x	x	
Scaling	x		x
Selfish mining	x		
Nothing at stake		x	
Consensus delay	x	x	x
Orphaned blocks	x	x	x
Brute force attack	x		
Pool hopping attack	x		
Block size attack	x		
Transaction reordering	x	x	
Lack of consistency	x	x	

With support from the information about the threats found in the survey papers researched in our literature review, we further categorized each threat related to the approaches to consensus. By doing so we looked at the three most common consensus mechanisms; Proof of Work, Proof of Stake and Practical Byzantine Fault Tolerance [31] [75]. For each threat, we analyzed the information given by the survey papers and snowballing references to see if one of the three consensus mechanisms are prone to the specific threat. If no data was provided from the survey papers found in our literature review to prove the threat-consensus-correlation, we did not include any marks in Table 7.

Table 8: Matching network related threats with blockchain type

Network threat	**Public Blockchain**	**Private Blockchain**	**Comment**
Fork problems	x		No reports on private blockchains
Eclipse attack	x	x	Albeit much harder for private
Short address attack	x	x	
Sybil attack	x	x	
Time jacking	x	x	
DoS	x		Significantly harder on private
DDoS	x		Significantly harder on private
Time delay attack	x		With known actors this will turn impossible for private blockchains
Routing attack	x		No reports for private blockchain
Balance attack	x		No reports for private blockchain
Replay attack	x	x	
Overflow attack	x		Attack on smart contracts
Pre-image attack	x	x	
BGP hijack	x	x	Protocol implemented in both private and public blockchain
DNS hijack	x	x	Possible to perform on private, but easy to detect
Impersonation attack	x	x	Possible to perform on private, but easy to detect
Wallet attack	x	x	
Address clustering	x		No reports on private blockchain
Network analysis	x	x	
Cloning attack	x		Highly unlikely for private blockchains due to centralization
Sniffing	x	x	

Table 8 shows the correlation between the threats categorized into the "Network" category, and further categorized into "Public blockchain" and "Private Blockchain". Public blockchains describe a network where anyone can join and take part, while in private blockchains the contributors are restricted [85]. These categorizations were made with support from the information about the different threats extracted from the survey papers from our literature review with multiple network related threats [85] [84] [82] [79] [78] [69] [37] and their own references. If no information about the correlation between the sub-categories and threats were found, we did not provide the correlation. The last column of Table 8 shows important comments about the specific threats describing if there were lack of information about a threat or if a threat has a bigger impact on one or the other.

Table 9: Matching network related threats with different network types

Network threat	**Synchronous**	**Partially synchronous**	**Asynchronous**
Fork problems	x	x	x
Eclipse attack	x	x	x
Short address attack		x	x
Sybil attack	x	x	x
Time jacking	x	x	x
DoS		x	x
DDoS		x	x
Time delay attack	x	x	x
Routing attack	x	x	x
Balance attack	x	x	x
Replay attack		x	x
Overflow attack		x	
Pre-image attack	x	x	x
BGP hijack	x	x	x
DNS hijack	x	x	x
Impersonation attack	x	x	x
Wallet attack	x	x	x
Address clustering		x	x
Network analysis	x	x	x
Cloning attack		x	x
Sniffing	x	x	x

Similar to the sub-categories shown in Table 8, the network category can be categorized into "Synchronous Network", "Partially Synchronous Network" and "Asynchronous Network". These types are describing the different timings(latency) of communication. Saad et al. [37] mentions there is a need for more research regarding different types of blockchain networks. By categorizing the different network related threats, our work can provide a start to that research. Each threat related to network found from our literature review was categorized forming the result of Table 9. In addition to building the knowledge base regarding network threats, this was done to further classify the threats, making it easier for the user to see what specific threat is related to them in our application for RQ2. The correlation between the threats and the sub-categories were found by analyzing the survey papers formed by the literature review similar to how we produced Table 8.

Table 10: Matching security issues with strategies for blockchain interoperability

Inteoperability threat	**Notary scheme**	**Sidechains/Relays**	**Hashed Time-Lock Contracts**
Wormhole attack			x
Collusion attack		x	x
Denial of service in atomic swaps			x
Asset locking			x
Loss of funds in atomic swaps			x
Double spending attack		x	x
No transaction finality	x	x	
Timing-attack		x	x
Different signature algorithms	x	x	x
Single point failure	x	x	
Private key attack	x	x	x
Sybil attack	x	x	x
Eclipse attack	x	x	x
Denial of service	x	x	x
No liveness	x	x	x
Fraud-proof attack		x(plasma)	

The results are derived from our MLR [3], appendix B

Table 10 is the results conducted by [3] where it is detailed. This can also be found in appendix B.

Threats and security issues listed in both the spreadsheet [86] and tables 7, 8, 9, 10 are extracted from the literature and are the final result of answering research question 1. To precisely answer research question 2, transferring the result to the following RQ was accomplished.

5.2 Research question 2

The section presents the process and how the authors prioritized functionality during the implementation of the threat modeling plugin. Furthermore, it provides information about the high-level functionality, ER diagram, required libraries, and supporting technologies used in the prototype for the threat modeling plugin.

5.2.1 Implementation

This section describes in detail how we implemented the features in the application. The section presents the requirements created, all supporting technologies, and the implementation done during the development process. We also customized the STRIDE model to be more fitting to blockchain-related issues. We chose to use STRIDE rather then other threat modeling approach to develop the tool because STRIDE is used in most of the current threat modeling tools on the market, as shown in Table 1. Firstly, using the STRIDE approach instead of other models, such as DREAD (explained in section 3.2.2), suits our intent for the application because STRIDE offers a threat classification model. Secondly, STRIDE is more business-oriented and can be more advantageous for non-technical persons to understand the risks. The results are presented at the end of the section, described in detail in section 5.2.2.

5.2.1.1 Requirements: A set of requirements was established and placed in the relevance cycle during the research process of existing threat modeling tools. These requirements are both functional and platform-related, and it was extracted from the literature based on intended value for further research and implementations. Examples of the requirements are listed in Table 11.

Table 11: Requirements

No.	Requirement
R1	The user should be able to create new blockchain based on input
R2	The user should be able to easily discover all threats related to input
R3	Threats should be generated with low latency
R4	The threats should be mapped to a set of correlating STRIDE threats
R5	The user should be able to add new threats
R6	Description about the threats should be easily accessible
R7	The user should be able to store their session for future analysis

5.2.1.2 Prioritized functionality: Prioritizing functionality is essential for building a prototype. The goal was to make the application functional and make it possible to implement extra functionalities easily. Building a prioritized functionality Table 12 gives an advantage for developing and controlling the implementation process's direction.

One of the most important features was making the tool dynamic. The user can interact with the data stored in the database, and the application provides necessary information about threats. As a result, highly prioritized functionalities are the fundamentals of the application.

Medium-prioritized functionalities provide more usefulness and novelty to the application, although these features are not strictly necessary.

Some features can be implemented later and are not highly prioritized because other features are more important for the purpose. However, providing features like opening a web browser for more details about the threat and exporting data files to existing online vulnerabilities databases are features that could potentially give the application more value in future work.

Other functionalities might be prioritized and implemented throughout the implementation process based on the evaluation. Chapter 6 presents the evaluations.

Table 12: Functionality prioritized

No.	Functionality	Priority
F1	Add blockchain threats	High
F2	Show all related threats	High
F3	Add new blockchain technology	High
F4	Categorize threats	High
F5	STRIDE parameters	Medium
F6	Threat description	Medium
F7	Interoperability threats	Medium
F8	Open web browser	Low
F9	Export datafiles	Low

Based on what the existing literature does not cover on threat modeling tools for blockchain technology, these functionalities were inspired by existing threat modeling tools for information technology systems and through collaboration with our supervisor. The overall goal is to allow the user to quickly access, add and discover threats to the technology so that users can be aware of newly discovered threats to both general blockchain and interoperable blockchains. The tool can be utilized to analyze different threats when modeling new features. The tool covers threats found in the literature and hopefully be valuable for future research and implementations. Integrating the tool with existing threat modeling tools as a plugin can be a practical implementation for researchers and developers working with blockchain technology.

5.2.1.3 Customized STRIDE: For our application, we need to connect every single threat the model produces to one or more STRIDE category. To fully categorize the threats within the field of blockchain, we customized the STRIDE model similar to Howard Poston [87]. This was done due to the limiting scope of the threat "Elevation of Privilege" where it does not detail the attacker's advantage point. "Elevation of Privilege" is therefore expanded into three subgroups; "Account", "Blockchain" and "Smart contract".

Table 13 below shows a more detailed overview of our use of STRIDE in our application:

Table 13: Customized STRIDE parameters

Acronym	Threat	Desired Property
S	Spoofing	Authenticity
T	Tampering	Integrity
R	Repudiation	Non-repudiability
I	Information Disclosure	Confidentiality
D	Denial of Service	Availability
E(A)	Elevation of Privilege Account	Authorization
E(B)	Elevation of Privilege Blockchain	Authorization
E(S)	Elevation of Privilege Smart Contract	Authorization

5.2.1.4 ER model: An entity-relationship (ER) model is essential for modeling data relations stored in the database. Figure 14 presents the overview of the database and how relations and entities are stored in the database. Since the application aims to analyze threats related to different blockchain technologies and blockchain components, each entity has a relationship with the threat entity. The relationships between entities differ; therefore, the relationships illustrated in the Figure 14 are clarified as follows:

- **0...1:** Refers to zero to one relationship.
- **0...*:** Refers to zero to many relationship.
- **1...1:** Refers to one to one relationship.
- **1...*:** Refers to one to many relationship.

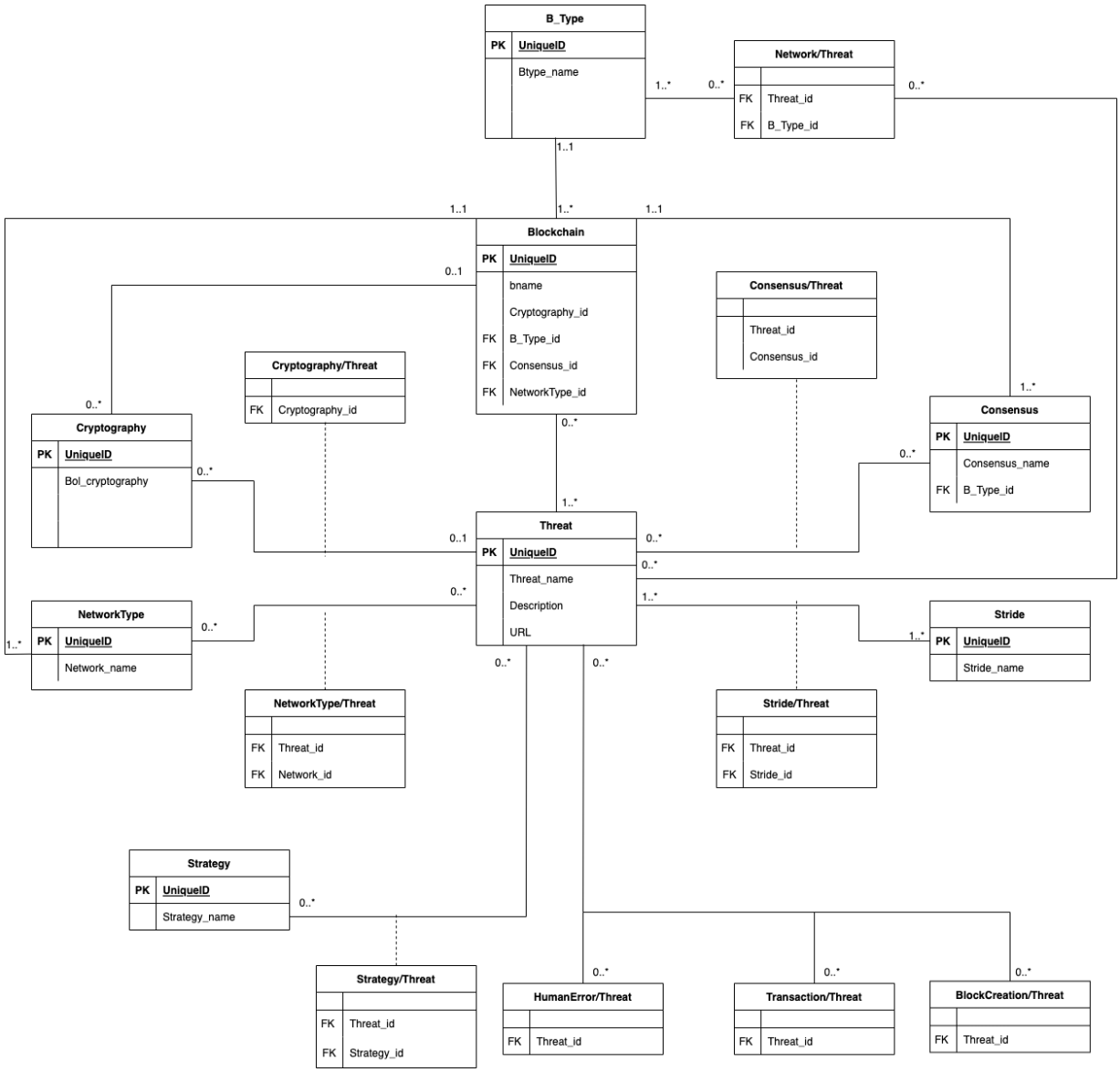


Figure 14: ER diagram

From the ER diagram shown above the main idea is to show the relationship between the entities and especially between threat and the different categories created in RQ1. The diagram shows the amount of many to many relationship between them. This means that a specific threat can target multiple instances of an entity and a specific entity can have multiple threats. This is an important observation to show the complexity of a threat modeling tool

5.2.1.5 Supporting technologies

Python: Python is an open-source, general-purpose language and a popular programming language used by academics and the software industry. Python provides high-level data structures, dynamic typing, and dynamic building, which allow the developer to build applications and scripting rapidly. Python supports many modules and packages which provide high versatility [88].

Tkinter: As Python provides many different libraries and packages, Tkinter [89] is one of the most well-documented libraries and has become the chosen GUI library for implementation. Tkinter relies on 'widgets' to create a graphical interface. Widgets refer to any object, such as buttons, frames, labels, etc. In order to place object after preference, Tkinter uses grid manager. The grid manager places widgets into a 2-dimensional table and can be adjusted their position by their row and column value as shown in Table 14 [90].

Table 14: Grid management Tkinter

<table>
<tr><td rowspan="3">Grid(0,0)
rowspan = 3</td><td>Grid(0,1)</td></tr>
<tr><td>Grid(1,1)</td></tr>
<tr><td>Grid(2,1)</td></tr>
<tr><td>Grid(3,0)</td><td>Grid(3,1)</td></tr>
<tr><td colspan="2">Grid(4,0) columnspan = 2</td></tr>
</table>

SQLite: SQLite is a well-documented database engine which implements standards from SQL [91]. SQLite allows users to interact with relational databases and provides an in-memory open-source library that does not require any pre-installation [92]. A python library named sqlite3 is used to implement the application. This library provides minimal error and keeps database files up to date. Sqlite3 operates with a cursor, which executes SQL commands on the database and stores values returned by the command. To access the result returned, call fetchone() or fetchall() functions on the cursor, and the result is iterable [91].

GitHub: GitHub is a cloud-based service that helps developers worldwide manage, store code, and keep track of changes in their code. GitHub consists of two connected parts; *Git* and *version control* [93]. *Git* is a distributed version control system that allows the developer to keep track of history and the entire codebase. Additionally, branching and merging are two standard features. *Version control* keeps track of changes made in the codebase and stores new versions of the software project code. All changes can be reverted if needed. Using GitHub in the implementation provides seamless collaboration without compromising the integrity, and version control for the project codebase is one of the most significant advantages.

The source code for our application can be found on a Github repository created by the authors [94].

5.2.2 The implemented functions of our threat modeling plugin

The following sections present the prototype developed and appropriate descriptions to comprehend the application flow.

5.2.2.1 Add blockchain menu: The "add blockchain" tab's overall goal is to allow users to feed the database with blockchains to their needs. Each input area is characteristic of a blockchain. Once the user presses "add blockchain technology to database" a new blockchain object is created in the database. Figure 15 illustrates how the user interacts with the input for each blockchain component. Clicking the "show blockchains from database" button returns a Tkinter 'treeview' of all stored blockchains, shown in Figure 16 where the user locates all blockchains stored in the database.

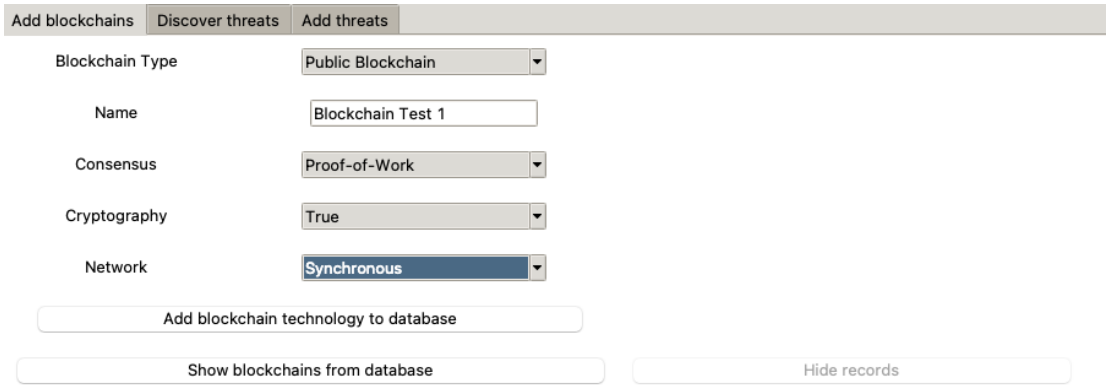
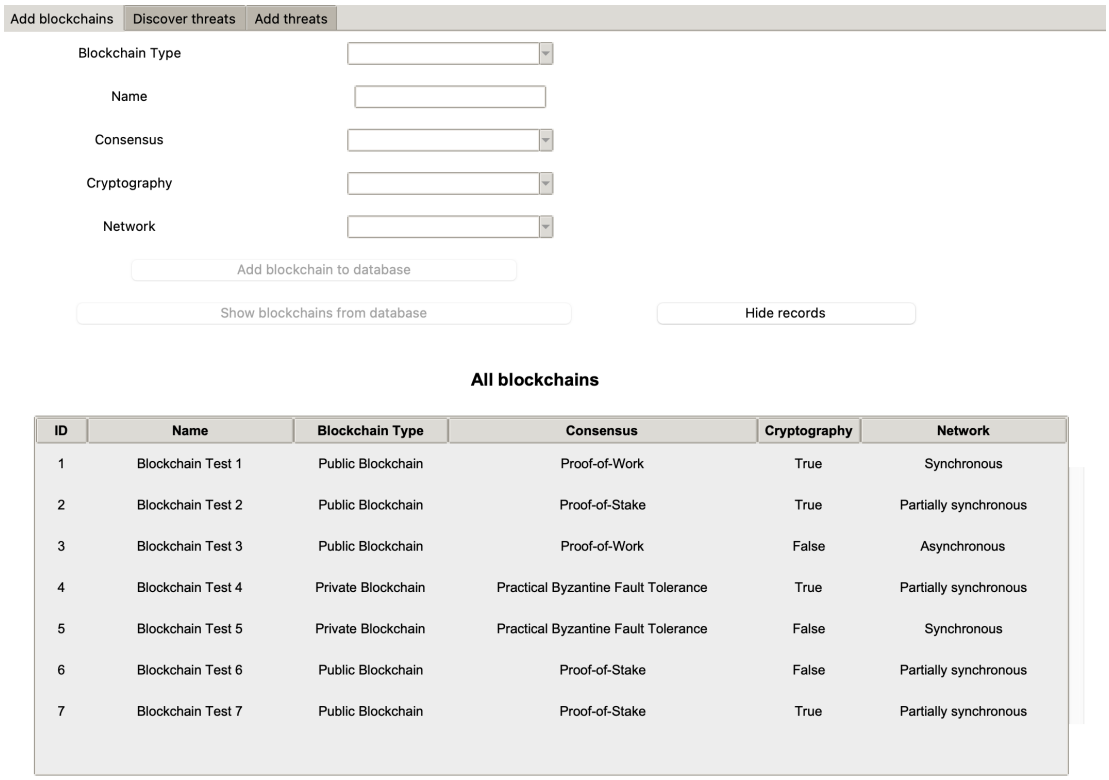


Figure 15: Add new blockchain to database



ID	Name	Blockchain Type	Consensus	Cryptography	Network
1	Blockchain Test 1	Public Blockchain	Proof-of-Work	True	Synchronous
2	Blockchain Test 2	Public Blockchain	Proof-of-Stake	True	Partially synchronous
3	Blockchain Test 3	Public Blockchain	Proof-of-Work	False	Asynchronous
4	Blockchain Test 4	Private Blockchain	Practical Byzantine Fault Tolerance	True	Partially synchronous
5	Blockchain Test 5	Private Blockchain	Practical Byzantine Fault Tolerance	False	Synchronous
6	Blockchain Test 6	Public Blockchain	Proof-of-Stake	False	Partially synchronous
7	Blockchain Test 7	Public Blockchain	Proof-of-Stake	True	Partially synchronous

Figure 16: Show blockchains from database

5.2.2.2 Discover threats menu: The discover threats menu lets the user discover threats from the literature presented in RQ1 and authors' MLR [3]. As Figure 17 illustrates, the user can reuse the blockchain created in the "add blockchain" menu and connect another blockchain with one of three chain interoperability strategies, which return a hierarchical tree view of all threats related to the input. The inputs do not allow the user only to select one blockchain. Therefore, the hierarchical overview presents threats to each component of the blockchain technology input, regardless of the second input and which blockchain strategy is specified. By utilizing this technique, the user can separately analyze threats to each of the inputs and the threats related to the interoperability strategy if desired.

The threats are classified into consensus, network, transaction, block creation, human error, cryptography(if not used), and interoperability, as shown in Figure 18. Additionally, each threat has its description, STRIDE parameters, and a URL. If the user double-clicks the threat, the application opens the connected URL in the web browser so that the user can get more information about the threat. Figure 19 shows how the user can use the tool to analyze blockchain threats related to components from the input.

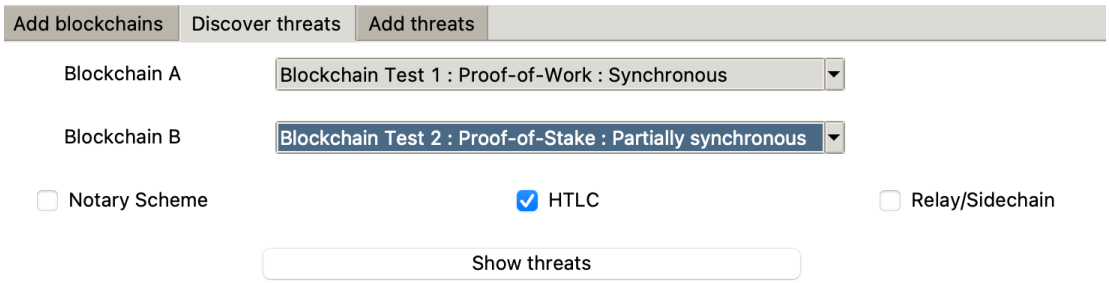


Figure 17: Discover threats input overview

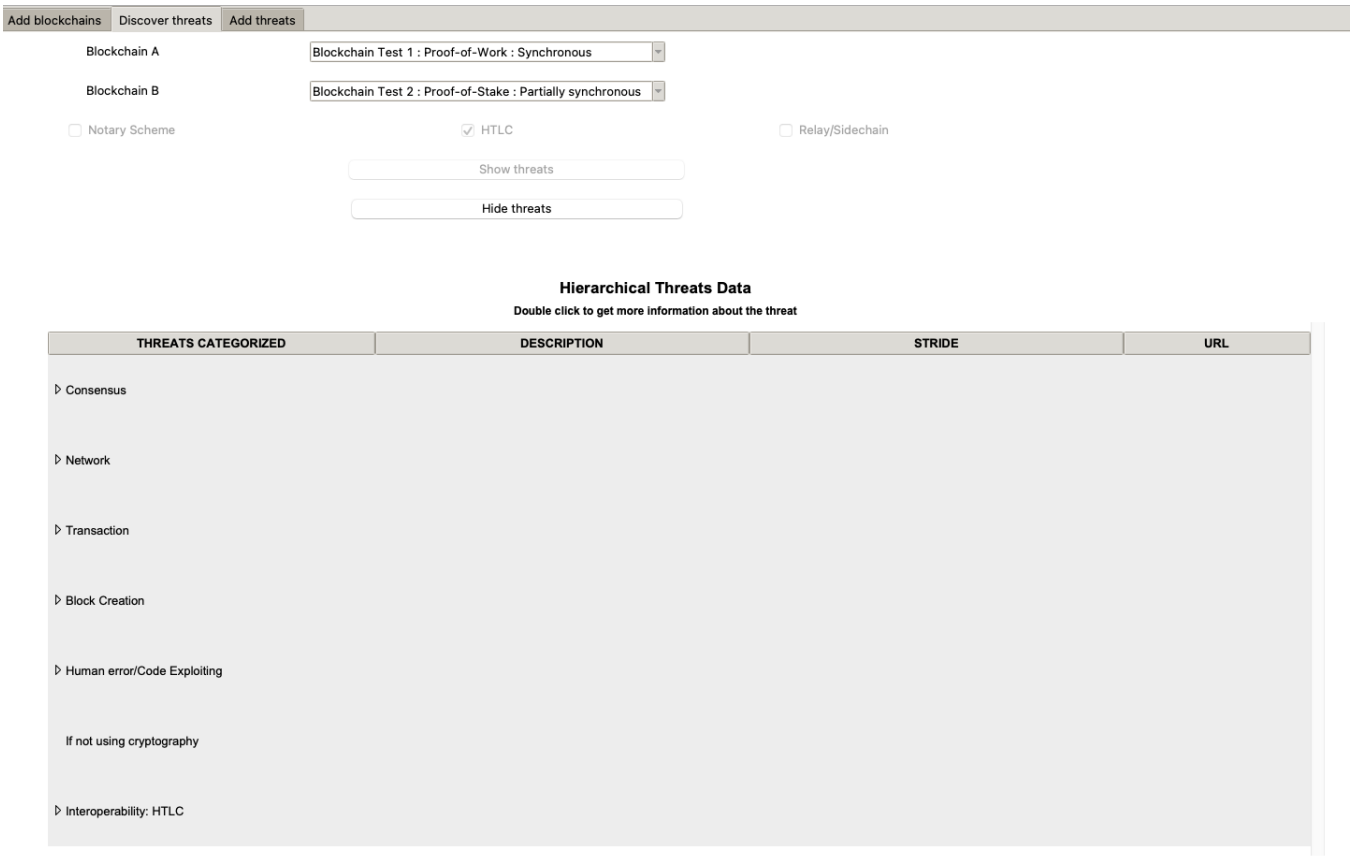


Figure 18: Hierarchical overview

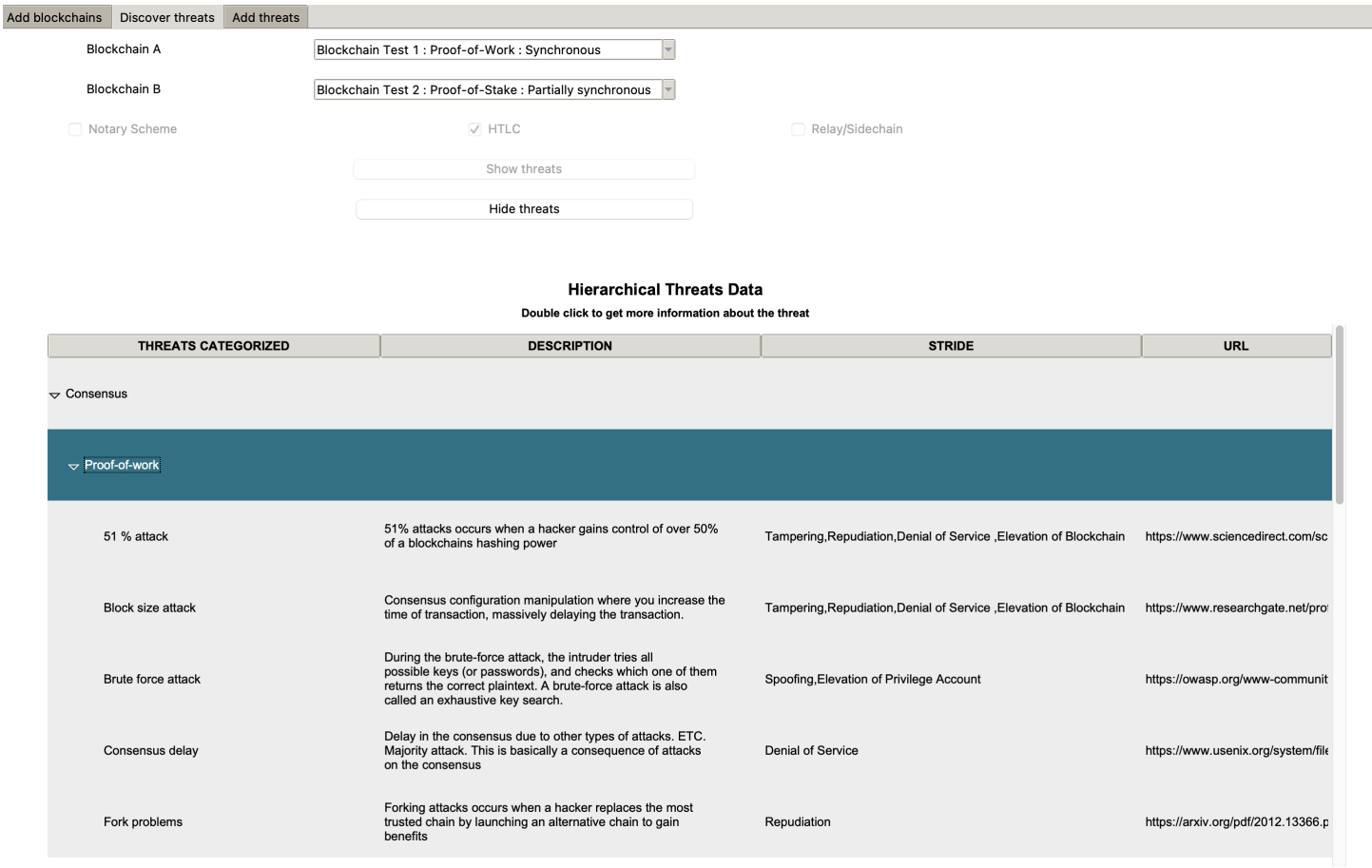


Figure 19: Hierarchical threat overview with subgroups

5.2.2.3 Add threats menu: The add threats menu tab builds on the same technical principles as the add blockchain menu, allowing users to update the database with discovered threats. When a user discovers a new threat, it is possible to document the threat with a name, a description of the threat, a URL for more information, which category the threat is related to, and affiliated STRIDE parameters. When the user submits the newly created threat, it will automatically update the database. This threat will appear in the hierarchical tree view in the discover threat menu if the input matches the newly created threat. Figure 20 shows an example of how to document a newly discovered threat.

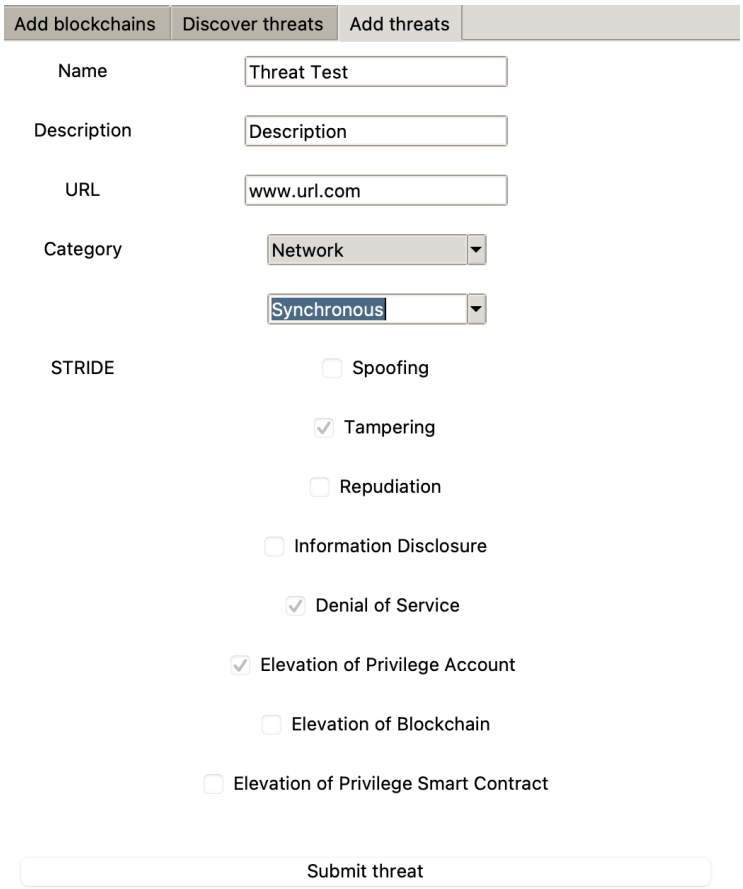


Figure 20: Add new threat

5.2.2.4 SQLite database files: To connect and build relations within the application, SQLite represents the underlying database. SQLite provides relational databases without accessing "real" databases. Described in more detail in the section 5.2.1.5. When the user interacts with the "show records" or "add records" buttons, a SQL query gets executed on the database. Figure 21 shows all data files located in one folder as CSV files, which can be updated and queried when the user wants data. Figure 22 shows how threats are stored in CSV. Relations were created with primary and foreign keys, inspired by the ER diagram shown in section 5.2.1.4. Appendix C presents the database, all connected tables, and relations within the database.

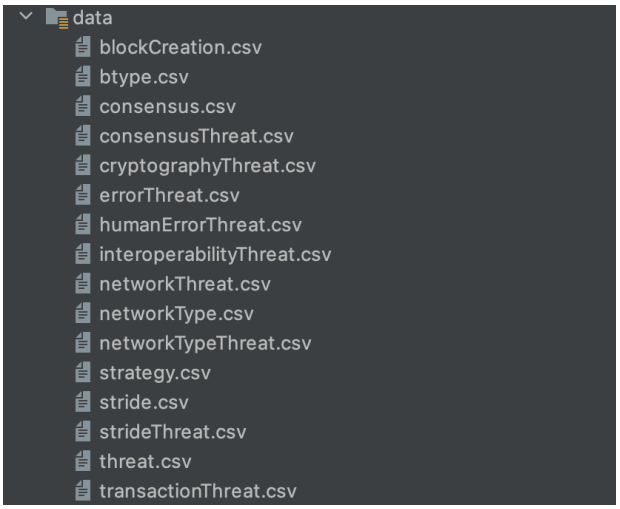


Figure 21: All CSV data files

ThreatID	Threat_Name	Description	URL
1	51 % attack	51% attacks occu	https://www.sciencedirect.com/science/article/pii/S0167739X17318332
2	Fork problems	Forking attacks oc	https://arxiv.org/pdf/2012.13366.pdf
3	Scaling	The ability to han	https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9133427
4	Selfish mining	Is when a miner c	https://link.springer.com/chapter/10.1007/978-3-662-45472-5_28

Figure 22: An example of how threats are stored in CSV

5.2.2.5 Prototype: When looking at how to integrate threats found from RQ1 into existing threat modeling tools, the literature provided no descriptive work. In order to answer the research question, it was necessary to focus on existing threat modeling tools, for example, Microsoft Threat Modeling tool [20] (more detail in section 3), to do an in-depth analysis of how these tools analyze threats in general information technology systems.

Based on the analysis of existing tools, some requirements were established. Threats related to system architecture and integration with STRIDE parameters were two starting-point requirements for the prototype. Throughout the development of the prototype, the envision has been that it should be able to be user-friendly and perform required tasks to manage and discover threats related to blockchain technology. As an extra feature, the user can interact with threats related to interoperable blockchains, based on the results from the specialization project [3], shown in appendix B. Table 15 illustrates which requirements and features were satisfied and not throughout the development phase.

Table 15: Matching requirements and features

Requirements	Satisfied	Not satisfied	Features
R1	x		Add new blockchain technology
R2	x		Discover threats
R3	x		Low latency with sqllite
R4	x		Customized STRIDE
R5	x		Add new threat
R6	x		Threat description
R7		x	Store session state

The prototype can perform the primary tasks and provide the features it was designed for. The current system allows users to interact and perform different tasks outlined in section 5.2.2. The application allows users to analyze threats related to their specific blockchain quickly. The user can add blockchains to their needs and discover threats mentioned in the literature. If the user researches new threats, the user can easily add these threats to the database. Supplying these features allows the application to be updated constantly and provides value for future research and development.

Despite the prototype fulfilling its overall goal, the prototype allows the user to interact with the database to add newly discovered threats and connect these threats to a category. Additionally, the user can open a web browser by double-clicking the threat. The importance of having these features is not only to provide an overview of threats in the literature but educate the user with necessary information about the threat and hopefully provide more in-depth knowledge. The prototype can constantly be updated by guaranteeing that the user can add new threats to the database and provide value for future blockchain implementations and research.

In general, many features and implementations have been discussed throughout the process, but they could not be prioritized due to the time constraint. Some of the features discussed can be found in section 9 where future work is being outlined. However, it is worth mentioning that building a plugin for existing threat modeling tools should contain basic functionality due to user-friendliness. Building more robust systems and tools increases the complexity and potential cost of the system, which deviates from the primary focus and goal.

6 Evaluation

A survey and hands-on usability testing were conducted to gather necessary information and identify the novelty in the evaluation process. This feedback is helpful when designing a new plugin for existing threat modeling tools and creates a shared understanding about whether it is valuable for businesses and future implementations. The following section presents the evaluation method, participants, procedure, data analysis, and finally, the evaluation results.

6.1 Method

The evaluation method used in this paper has primarily been inspired by remote usability testing, more precisely, asynchronous remote usability testing [95] [96] [97]. The users and evaluators are both separated in time and space. The overall goal of this approach is to collect rapid, cost-efficient, and brief evaluations of the artifact. The paper conducts asynchronous remote usability testing combined with a digital survey where the evaluator gets the artifact and a link to an online survey. The evaluation is accomplished in participants' everyday environments. Responses are then sent back to developers for further analysis [95]. The method brings difficulties to follow-up questions for clarification. However, the online survey is built with quality open-ended questions which allow the user to provide detailed explanations and clarify any misconceptions. The evaluation tests the application's usefulness and not the GUI directly, which satisfies our evaluation criteria.

6.2 Expected participants

Since there was a lack of people who had general blockchain knowledge in the author's environment, the participants in the evaluation process were chosen by suggestions from other participants based on their blockchain knowledge and experience with the technology. However, the participants are located in different places in Norway and worldwide, providing valuable evaluations supported by their backgrounds.

The participants of the study must have one of the following traits:

- Basic knowledge of blockchain technology
- Experience with existing threat modeling tools
- Interest in IT-security

Participation was voluntary, and they received the necessary information and description about the thesis and evaluation before answering the survey.

6.3 Procedure

The evaluation procedure was accomplished by a survey and hands-on application testing. Participants' locations and spare time made this procedure the most optimal solution due to different time zones and the time it took to conduct the process.

The survey held 18 questions ordered into demographic questions(3), background(2), product feedback(7), user experience(3), and general feedback questions(3). All questions were mandatory, and the answers were anonymous. The majority of the questions were designed as open-ended, allowing the participants to provide complementary explanations. Additionally, the intention is to prevent the participants from answering yes/no and presenting answers based on the usability of GUI instead of the application's usefulness. The survey questions can be found in appendix A, Table 19.

6.4 Data analysis

As described in section 4, an inductive thematic analysis approach was utilized in order to summarize essential features, examine perspectives, emphasize dissimilarities and similarities, and furnish unforeseen insights [67]. As a result, the following section presents the results and themes derived from the analysis.

Once the data was extracted from the survey responses, each response was allocated codes to represent the content of the data. The coding process was accomplished manually by the authors. Once the data was coded, themes were made to sort and analyze the responses. The themes derived made the addressing and summarizing of interesting and essential data more efficient. Table 16 shows the themes derived from the analysis by a total of 8 participants and 83 responses. Table 17 illustrates an example where we analyzed the response to one of the survey questions.

Table 16: Themes derived from the analysis

Themes	**Total**	**Total (%)**
Threat model	4	4.82
Threat overview	18	21.69
Graphical user interface	13	15.66
Program update	7	8.43
Threat risk rank	6	7.23
Threat mitigation	1	1.20
Program integration	2	2.41
Risks insight	6	7.23
Program support	2	2.41
User satisfaction	17	20.48
User background	2	2.41
Confused user	2	2.41
Other	3	3.61
SUM	83	100 %

Table 17: An example of codes and themes derived from the analysis

What problem/goal do you think the application is trying to solve/achieve?

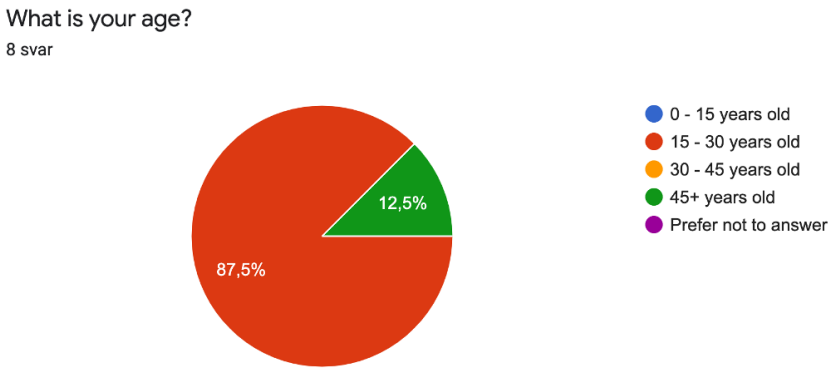
Responses	**Code**	**Theme**
threat modeling central database	Application for threat detection	Threat model
Finding pros and cons for the different blockchain technology's security	Application for threat detection	Threat model
To give an overview of existing vulnerabilities for a blockchain, including interoperability between two blockchains	Giving overview of threats/security issues	Threat overview
I believe the application is attempting to solve the problem of blockchain security. Specifically getting an overview of the current existing security challenges and attack vectors, including interoperability	Giving overview of threats/security issues	Threat overview
I believe this application is trying to solve threat detection and be some sort of middle point for threat analasys	Application for threat detection	Threat model
Showcase threats related to blockchain based on your own development	Giving overview of threats/security issues	Threat overview
I think the application is trying to give information about potential threats within blockchain technology	Application for threat detailing	Threat overview
Threat modeling for blockchain technology	Application for threat detailing	Threat model

6.5 Results

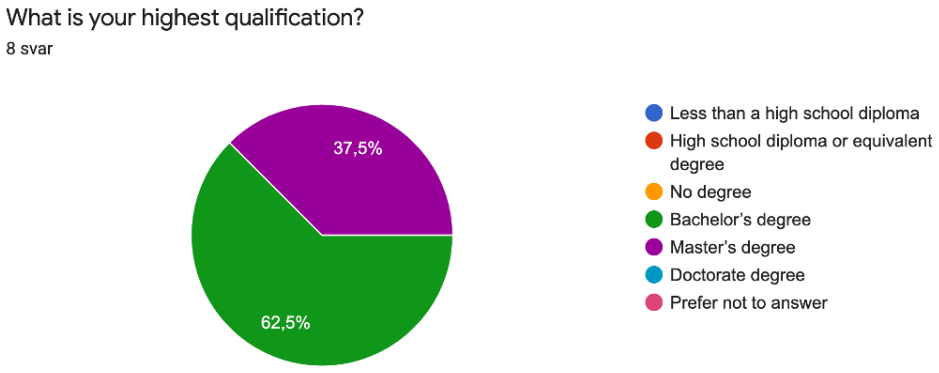
This section provides a holistic overview of all the responses provided by the participants that evaluated the application. The subsequent sections present illustrations and diagrams of the responses to survey questions shown in appendix A.

6.5.1 Participants

Eight participants in total evaluated the application. 8 of 8 participants were registered as male, and the majority of participants' ages seemed to be in the same interval. There is almost an equal distribution of bachelor's and master's degrees. Figure 23 shows the responses.



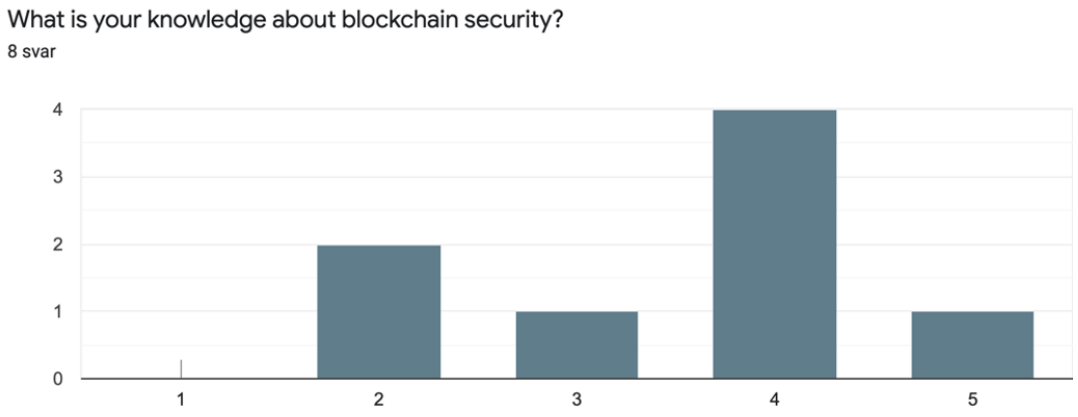
(a) Age distribution



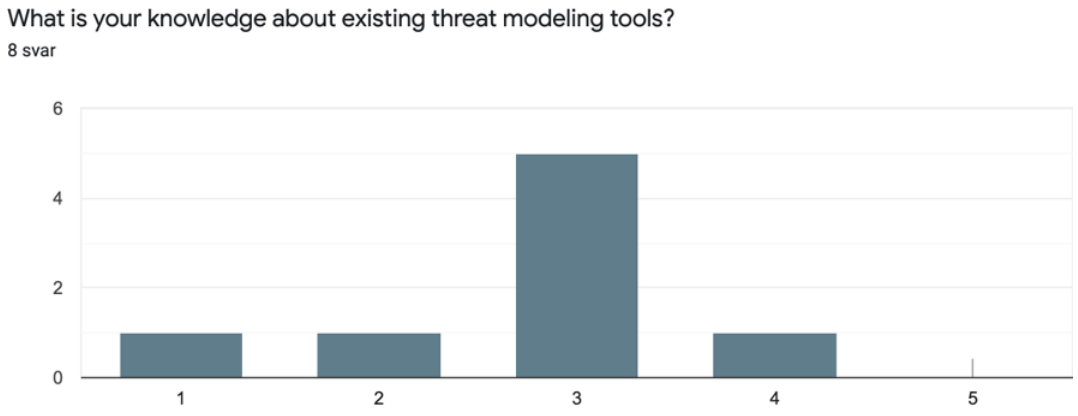
(b) Qualification distribution

Figure 23: Responses to demographic questions

In order to set the standards for the responses, a couple of background questions about existing knowledge in blockchain and threat modeling tools were asked. The participants needed to know about the technology or existing threat modeling tools to provide complementary and valuable explanations. Figure 24 illustrates the background knowledge of all the participating evaluators. As shown in the figure below, the participant's background knowledge is relatively high, which is crucial to providing credible responses. Despite some lack of knowledge in existing threat modeling tools, it is still the knowledge base in the middle of the linear scale. As a result, participating participants will provide feedback that adds value to the prototype and present qualified explanations.



(a) Blockchain knowledge distribution



(b) Threat modeling knowledge distribution

Figure 24: Responses to background questions

6.5.2 Open-ended questions

When examining the survey responses from the participants, the prototype's overall goal seems to be in touch with what the participants expected. Based on the product feedback from the participants, the prototype provides intuitive and user-friendly features, independent of the user knowledge base.

Some responses pointed out GUI-related features, which are valuable for future implementations. However, in our case, the aim and focus for the prototype did not include GUI-related features.

Despite the GUI responses, more filter and input options are clearly stated, and a list of all possible threats regarding blockchain technology was mentioned. The responders mentioned mitigations, more grouping of threats, and a similar OWASP top 10 list, would be valuable for future implementations.

Overall, all the participants were optimistic about the overall experience using the application. They did not have anything else to add to the quality or use of the application that we had not covered, despite modernizing the GUI and interface of the prototype. The most important aspect of providing the incentives to use the prototype in the future is continuous and automatic updates threats and following risks regarding the technology.

Positive feedback	**Features/improvement needed**
• Worked as expected	• More input options
• User friendly	• Graphical user interface
• Great idea for further development	• List of all existing exploits
• Easy to use	• List only one blockchain
• Valuable	• Filtering options
• Clear and informative	• Export data
• Good overview of exploits	• Threat mitigation
• Easy to register data	• Threat risk rank
	• Explanations of abbreviations

At the end of the survey, we asked the participants if there was anything else to add to the quality or use of the application that we had not covered. This was a voluntary question, but some responses mentioned GUI updates and more explanations and descriptions for less informed individuals. Based on the overall evaluation provided by the participants, a new set of requirements and functionalities were mentioned and extracted. Due to time constraints for conducting this review and development of the prototype, Table 18 show a list of requirements and new functionalities that can be valuable for forthcoming development and implementations.

Table 18: Requirements and functionalities extracted from the evaluations

Requirements

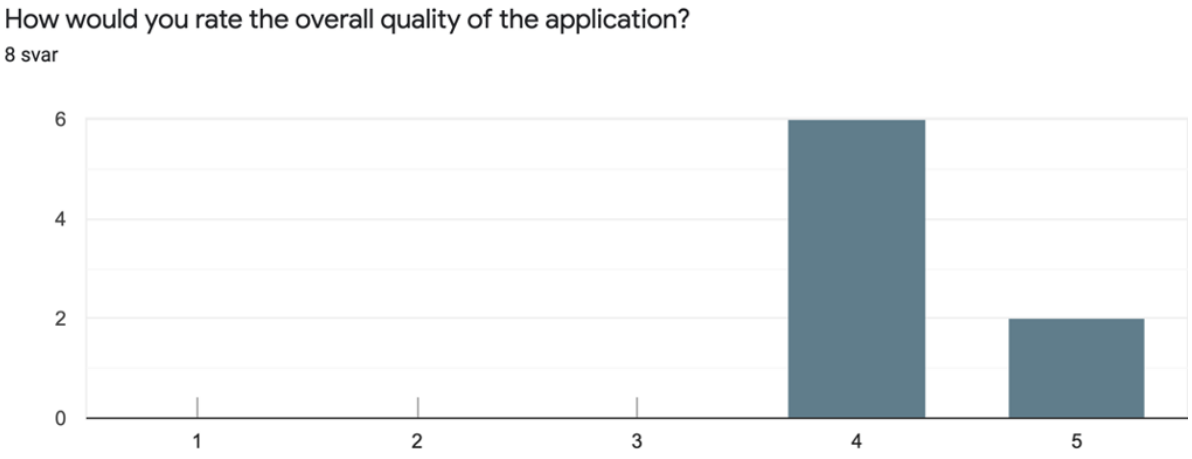
No.	**Requirement**
R1	The user should be able to get a complete list of all existing threat without a input
R2	Threats should be automatically be updated due to inaccuracy over time
R3	The user should be able to discover threats for only one blockchain
R4	The user should be able to export data

Functionalities

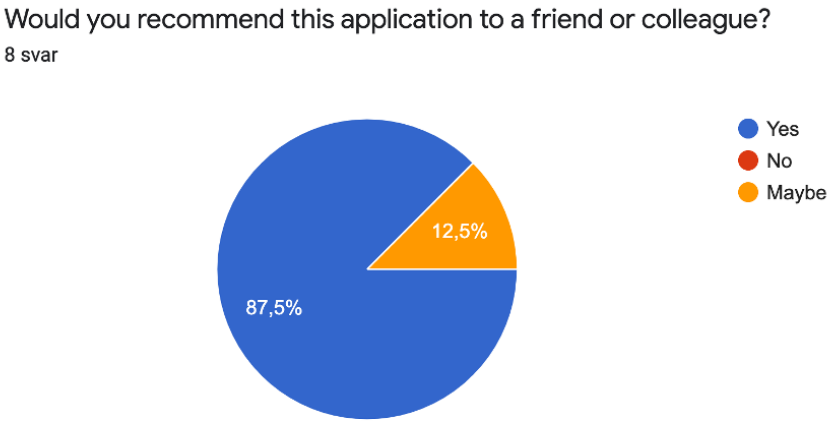
No.	**Functionality**	**Priority**
F1	Threat risk rank	High
F2	Provide mitigations to each threat	High
F3	Filter threats based on most common to least common	Medium
F4	Expand the blockchain input fields	Medium
F5	More details and descriptions about the threats	Low
F6	More explanations and descriptions for less informed individuals	Low

6.5.3 General feedback

A set of general feedback questions was provided in the survey. This type of question allows the participants to rate the overall quality of the application and provide general feedback that specifies errors and omissions. Additionally, the user is free to clarify features that assemble more value for the user and the application in the future. As shown in Figure 25, the overall quality of the application is high, and the majority would recommend the application to a friend or colleague.



(a) Overall quality rating



(b) Recommendations

Figure 25: Responses to general feedback questions

7 Discussion

This chapter answers the research questions and discusses the development of the current prototype with considering its potential implications. In the following section, we will discuss our findings and provide comparisons with related work and threats to validity.

7.1 Research question 1

Our first research question heavily focuses on the knowledge base formed by existing literature and provides further analysis on them. To answer the question we performed a literature review of existing surveys collecting different threats regarding blockchain systems. These threats were extracted from the surveys and categorized. The categories used were inspired by existing literature and detailed in section 5.1.1.

7.1.1 Comparison with related work

When looking at our primary papers, all of them partially answer research question 1, albeit on a much smaller scale. Our primary papers displayed in Table 5 consist of surveys collecting different threats from research papers, which we have argued to be one of the most important features for performing systematic threat modeling on blockchain systems. However, none of the surveys are detailing the amount of threats presented by our research. A difference in our literature review to the primary paper is our strategy in looking through surveys instead of research papers. With that strategy, we cover a broader area in the field of work. This is important for our proposed application answering research question 2 where we need to disclose as many vulnerabilities as possible for the application to have any practical use case.

Similar to our primary papers, we categorize the threats based upon our references. However, we also justify them by comparing them to multiple surveys and keeping the most relevant categories for further analysis. This is important for our research where categorization is also a necessary feature for performing threat modeling.

7.1.2 Implication to academia and industry

The results conducted by our literature review is a new step towards blockchain threat modeling. It contributes with a comprehensive threat overview with categorization, providing a better knowledge base of existing threats. As proven in this paper, improving the knowledge base regarding blockchain threat modeling is a great contribution to the field of work as the security aspect within business related development is paramount.

7.1.3 Threats to validity

Our report might be affected by the inaccuracy of data extracted to provide an overview of security threats in the first layer of blockchain. An incompleteness of search terms, search engines, or data extraction may be the reason for suffering from inaccuracy. In this section, we will present these validity threats.

7.1.3.1 Term selection: Creating search terms is a complex problem to solve. The literature does not mention an optimal way to create and use the combinations of search terms. As a result, there is no guarantee that all relevant literature is extracted based on our term selection. We still believe the terms conducted for this research helps find the majority of papers related to the field.

7.1.3.2 Search engines: Only utilizing Google Scholar and Oria as search engines to conduct the literature review, might have limited the range of documents found. These search engines were chosen as it provides papers from multiple databases supported by NTNU.

7.1.3.3 Data extraction: Filtering and extracting articles through multiple steps were performed by both authors simultaneously to mitigate any form of bias. However, the author's bias may be reflected in the work and create room for misjudgment. Involving more people in the process, if possible, would cause less bias and errors.

7.1.3.4 Search modules and restrictions: Including restrictions to our search process may cause inaccuracy in the papers we extracted. Including only eight pages of the search result, we truly rely on the page rank algorithm provided by the search engine. Due to time constraints, we trusted the page rank algorithm, but exciting and relevant papers might exist beyond the research scope.

7.2 Research question 2

To answer the second research question, we aimed to develop a tool to facilitate blockchain threat modeling. There was a lack of research on how to perform threat modeling on the blockchain and how to develop tools to handle threats related to blockchains. We applied a design science approach to fully comprehend existing threat modeling tools and establish requirements for developing the prototype. Throughout the research process, requirements were specified that provided the first features.

7.2.1 Comparison with related work

Although STRIDE is not commonly used in blockchain technology, we suppose that expanding the STRIDE parameter "Evaluation of Privilege" into three subgroups will suit our intent for the application and present new possibilities for further expansion of the model. STRIDE is commonly used in businesses; therefore, it is advantageous to link it to each threat so that non-technical persons can recognize and understand the threats the technology faces.

The prototype developed fulfills its overall goal, and it can dynamically be updated and keep track of threats related to different blockchain components. The prototype enables the user to analyze threats and reinforce knowledge about threats, similar to existing threat modeling tools such as Microsoft Threat Modeling Tool [20]. As shown in Table 1, none of the models furnishes blockchain adoption. As a result, the tool fulfills the literature gap. Additionally, integrating this prototype into existing threat modeling tools is valuable for further implementation, development, and research of this expanding technology.

The evaluation of the prototype, an asynchronous remote usability testing combined with a digital survey where the evaluator gets the prototype and a link to an online survey were conducted. It was manageable to get 8 participants due to a lack of people who had existing blockchain knowledge. This method potentially restricts the opportunities to do follow-up questions if any misinterpretations occur, but based on the background knowledge question about blockchain, the responses are credible and reliable. Each response was taken seriously and summarized into a new set of requirements and functionalities for future implementations.

7.2.2 Implication to academia and industry

Building systems and applications without evaluating or mitigating the security risks is almost unimaginable. Threat modeling ensures organizations identify vulnerabilities within their system and purposefully minimize their attack surface. With our prototype, the user

or organization can quickly analyze and explore different threats related to their blockchain systems. The planning phase for system developers using the tool encourages defense-in-depth and security controls so that implementations in the future are done securely. Compared to the literature, there is no similar project on how to model threats on a blockchain. Therefore, our prototype contributes to the research field and hopefully depicts an acceptable starting point for future research and implementations. However, the time limitation of the thesis affects the application complexity, but we believe our prototype will supply a value for researchers. Although the time constraints existed throughout the process, recommendations for future work are proposed in section 9.

7.2.3 Threats to validity

Regarding the application being a prototype, the application can suffer from inaccuracy in both the evaluation and how user-friendly the artifacts are. In the following section, these inaccuracies are presented.

7.2.3.1 Number of participants: The number of participants in the survey may affect the analysis of data. Fewer data points and responses can cause bias regarding the application's practical application. We felt it was important to have participant with prior knowledge of our field of work to be able to extract useful data from the evaluation. That statement could have impacted the amount of participants.

7.2.3.2 Acquaintances of researchers: Due to a lack of participants with knowledge of blockchain technology, some acquaintances of researchers were asked to evaluate the prototype. Acquaintance can lead the participant not to state their actual opinion. However, we were able to find participants with better knowledge of our field of work.

7.2.3.3 Follow-up questions: Sometimes it is desirable to take further action on the responses and provide follow-up questions but in our case makes this difficult to implement.

7.2.3.4 Anonymous survey: The survey was made anonymous, making the participants state their opinions without being identified. Anonymous surveys bring advantages and disadvantages; the drawback is the ability to clarify the respondents' complaints if the response is less specific. Our goal was to get participants to not retain any information and hopefully state their honest opinion about the usefulness, which is easier to do in an anonymous survey.

8 Conclusion

This paper aims to answer two blockchain security research questions. To answer the first research question, we followed an instructional process on how to conduct a literature review. By creating search terms, restrictions, and search modules, we could examine different security threats and differentiate them into categories. We justified categories based on how the literature mentioned them to foster a balanced and comprehensive picture of the threats. By exploring each security threat, several different threats are related to different blockchain components.

For each security threat found in RQ1, we assigned STRIDE parameters to each of them to ensure, in the following RQ, our prototype of a threat modeling plugin fulfills the CIA triad (confidentiality, integrity, and availability). Although STRIDE is not commonly used in blockchain and limited the scope of the threat "Elevation of Privilege", we expanded the "Elevation of Privilege" parameter into three subgroups.

To answer the second research question, we followed the Design Science methodology that resulted in multiple iterations researching existing threat modeling tools. As a result, the process provided a set of requirements to build a valuable prototype. The literature provided no descriptive work on integrating threats found in the first research question into existing threat modeling tools. Therefore, we assumed that building a user-friendly and manageable prototype where the user can quickly discover threats related to blockchain components was a necessity. Additionally, based on the results from the specialization project, the user can discover threats related to the three most common interoperability strategies in the blockchain.

The prototype has been evaluated by 8 participants. A survey with open-ended questions and hands-on usability testing was conducted to provide detailed explanations. The survey questions aimed to test the application's usefulness and not the GUI directly. Finally, we discussed our findings compared to related work and presented future work for the industry. This research contributes to increased knowledge of blockchain security threats, existing threat modeling tools, and how to integrate the analysis into existing threat modeling tools. A fully functional prototype was developed, and threats linked to blockchain components were analyzed.

The most common motivation for conducting this report is the considerable rise of blockchain interest the recent years. We expect a further rise in interest, and security concern arises naturally as the technology grows. Threat modeling for information technology systems is well established, but there is a lack of research on how to threat model blockchain technology. Our work summarizes all threats found, integrated with STRIDE, categorized, and finally presents a prototype of an analyzing tool that can hopefully be integrated with existing threat modeling solutions.

9 Future work

This section will present some possible features, research directions, or development that could potentially reach more novelty to our work. The list contains what we consider the most important features or implementations for future research. As the prototype is developed digitally, there are always possibilities of adding new functionalities and features that enhance the prototype.

- **Export data:** Our application provides a local database and local file management. Therefore, exporting threats data to other digital databases for vulnerabilities regarding blockchain technology could give more value in future implementation and constantly update newly found vulnerabilities.
- **Threat risk rank:** One future feature of the prototype is to score each threat with a risk rank. Ranking can be helpful for further analysis and where actions can be directed where they are most needed.
- **Threat mitigations:** Providing mitigations for each threat is one feature that can enhance value and make the application fully-fledged.
- **Integrate with existing solutions:** Our prototype is not connected to any existing threat modeling tools at the time of writing. Integration with existing threat modeling tools can allow blockchain researchers and developers to identify and resolve possible security threats to blockchain technology and interoperable blockchains when building blockchain systems.
- **Desktop application to web application:** Our application was made as a desktop application to handle the development process, accessibility, and other technical issues that potentially could lead our work to be more time-consuming than expected. Converting the application to a web application provides no location constraints, cross-platform availability, fewer resources, and many other advantages. Additionally, it provides more freedom in the development process and prototype features.
- **More research in the field:** There is a lack of research in threat modeling and threat modeling tools for blockchain technology. An elaborated guide or research on how to model threats on blockchain technology could be helpful for researchers.

References

[1] Ying-Chang Liang. *Blockchain for Dynamic Spectrum Management*, pages 121–146. Springer Singapore, Singapore, 2020.

[2] Sahil Gupta, Shubham Sinha, and Bharat Bhushan. Emergence of blockchain technology: Fundamentals, working and its various implementations. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020.

[3] Terje Haugum and Bjørnar Hoff. Multivocal literature review on security and privacy challenges in blockchain interoperability. Report, NTNU, 2021.

[4] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[5] Ariel Ekblaw, Asaph Azaria, John D Halamka, and Andrew Lippman. A case study for blockchain in healthcare:"medrec" prototype for electronic health records and medical research data. In *Proceedings of IEEE open & big data conference*, volume 13, page 13, 2016.

[6] Marko Hölbl, Marko Kompara, Aida Kamišalić, and Lili Nemec Zlatolas. A systematic review of the use of blockchain in healthcare. *Symmetry*, 10(10):470, 2018.

[7] Yu Zhang and Jiangtao Wen. The iot electric business model: Using blockchain technology for the internet of things. *Peer-to-Peer Networking and Applications*, 10(4):983–994, 2017.

[8] Hong-Ning Dai, Zibin Zheng, and Yan Zhang. Blockchain for internet of things: A survey. *IEEE Internet of Things Journal*, 6(5):8076–8094, 2019.

[9] Jacob Stenum Czepluch, Nikolaj Zangenberg Lollike, and Simon Oliver Malone. The use of block chain technology in different application domains. *The IT University of Copenhagen, Copenhagen*, 2015.

[10] Simone Porru, Andrea Pinna, Michele Marchesi, and Roberto Tonelli. Blockchain-oriented software engineering: challenges and new directions. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 169–171. IEEE, 2017.

[11] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 254–269, 2016.

[12] Vitalik Buterin. Critical update re: Dao vulnerability. *Ethereum Blog, June*, 2016.

[13] Michael del Castillo. The dao attacked: Code issue leads to $60 million ether theft. *Saatavissa (viitattu 13.2. 2017)*, 3, 2016.

[14] Hai Wang, Yong Wang, Zigang Cao, Zhen Li, and Gang Xiong. An overview of blockchain security analysis. In *China Cyber Security Annual Conference*, pages 55–72. Springer, Singapore, 2018.

[15] Bruce Schneier. *Secrets and lies: digital security in a networked world*. John Wiley & Sons, 2015.

[16] Suvda Myagmar, Adam J Lee, and William Yurcik. Threat modeling as a basis for security requirements. In *Symposium on requirements engineering for information security (SREIS)*, volume 2005, pages 1–8. Citeseer, 2005.

[17] Adam Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[18] Rui Zhang, Rui Xue, and Ling Liu. Security and privacy on blockchain. *ACM Computing Surveys (CSUR)*, 52(3):1–34, 2019.

[19] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107:841–853, 2020.

[20] Microsoft threat modeling tool. https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool. Accessed: 2022-04-03.

[21] Attack tree modeling. https://www.amenaza.com//documents.php. Accessed: 2022-04-03.

[22] Yue Chen, Barry Boehm, and Luke Sheppard. Value driven security threat modeling based on attack path analysis. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 280a–280a, 2007.

[23] Dimitri Van Landuyt, Laurens Sion, Emiel Vandeloo, and Wouter Joosen. On the applicability of security and privacy threat modeling for blockchain applications. In *Computer Security*, pages 195–203. Springer, 2019.

[24] Michael Nofer, Peter Gomber, Oliver Hinz, and Dirk Schiereck. Blockchain. *Business & Information Systems Engineering*, 59, 03 2017.

[25] Massimo Di Pierro. What is the blockchain? *Computing in Science Engineering*, 19(5):92–95, 2017.

[26] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14:352, 10 2018.

[27] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, June 2002.

[28] Qiheng Zhou, Huawei Huang, Zibin Zheng, and Jing Bian. Solutions to scalability of blockchain: A survey. *IEEE Access*, 8:16440–16455, 2020.

[29] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.

[30] Tim Kozak. Consensus protocols that meet different business demands. part i, 2018.

[31] Shubhani Aggarwal and Neeraj Kumar. Chapter eleven - cryptographic consensus mechanisms introduction to blockchain. In Shubhani Aggarwal, Neeraj Kumar, and Pethuru Raj, editors, *The Blockchain Technology for Secure and Smart Applications across Industry Verticals*, volume 121 of *Advances in Computers*, pages 211–226. Elsevier, 2021.

[32] Peter Wegner. Interoperability. *ACM Computing Surveys (CSUR)*, 28(1):285–287, 1996.

[33] Vitalik Buterin. Chain interoperability. *R3 Research Paper*, 2016.

[34] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. A survey on blockchain interoperability: Past, present, and future trends. *ACM Computing Surveys (CSUR)*, 54(8):1–41, 2021.

[35] Patrick Mallory. Cyber threat analysis, 2021.

[36] Tobias Guggenberger, Vincent Schlatt, Jonathan Schmid, and Nils Urbach. A structured overview of attacks on blockchain systems. *Twenty-fifth Pacific Asia Conference on Information Systems, Dubai*, 2021.

[37] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, DaeHun Nyang, and David Mohaisen. Exploring the attack surface of blockchain: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):1977–2008, 2020.

[38] Sumit Soni and Bharat Bhushan. A comprehensive survey on blockchain: Working, security analysis, privacy threats and potential applications. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, volume 1, pages 922–926, 2019.

[39] Mauro Conti, E. Sandeep Kumar, Chhagan Lal, and Sushmita Ruj. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4):3416–3452, 2018.

[40] Noama Fatima Samreen and Manar H. Alalfi. A survey of security vulnerabilities in ethereum smart contracts. *CoRR*, abs/2105.06974, 2021.

[41] Mahendra Kumar Shrivas, Thomas Yeboah Dean, and S. Selva Brunda. The disruptive blockchain security threats and threat categorization. In *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)*, pages 327–338, 2020.

[42] Jamal Hayat Mosakheil. Security threats classification in blockchains. 2018.

[43] Threat modeling. https://owasp.org/www-community/Threat_Modeling. Accessed: 2022-04-02.

[44] Threat modeling process. https://owasp.org/www-community/Threat_Modeling_Process. Accessed: 2022-04-02.

[45] Threat modeling process. https://owasp.org/www-community/Threat_Modeling_Process#step-3-determine-countermeasures-and-mitigation. Accessed: 2022-04-02.

[46] Peter Torr. Demystifying the threat modeling process. *IEEE Security & Privacy*, 3(5):66–70, 2005.

[47] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. Stride-based threat modeling for cyber-physical systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6. IEEE, 2017.

[48] Threat modeling process. https://owasp.org/www-community/Threat_Modeling_Process#subjective-model-dread. Accessed: 2022-04-02.

[49] Bruce Schneier. Attack trees. *Dr. Dobb's Journal*, 1999.

[50] Hans-Jürgen Zimmermann. *Fuzzy set theory—and its applications*. Springer Science & Business Media, 2011.

[51] Foundations of fuzzy logic. https://www.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html. Accessed: 2022-04-03.

[52] Adesina Simon Sodiya and B. Oladunjoye. Threat modeling using fuzzy logic paradigm. *Issues in Informing Science and Information Technology*, 4, 01 2007.

[53] Mansour Alali, Ahmad Almogren, Mohammad Mehedi Hassan, Iehab A.L. Rassan, and Md Zakirul Alam Bhuiyan. Improving risk assessment model of cyber security using fuzzy logic inference system. *Computers & Security*, 74:323–339, 2018.

[54] Inference process. https://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html#FP347. Accessed: 2022-04-03.

[55] Logical operations. https://www.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html#bp78l70-5. Accessed: 2022-04-03.

[56] Fuzzy logic toolbox. https://www.mathworks.com/products/fuzzy-logic.html. Accessed: 2022-04-03.

[57] Yue Chen, Barry Boehm, and Luke Sheppard. Measuring security investment benefit for cots based systems - a stakeholder value driven approach. 05 2012.

[58] Vineet Saini, Qiang Duan, and Vamsi Paruchuri. Threat modeling using attack trees. *Journal of Computing Sciences in Colleges*, 23, 04 2008.

[59] The coras method. http://coras.sourceforge.net/index.html. Accessed: 2022-04-03.

[60] The coras tool. http://coras.sourceforge.net/coras_tool.html. Accessed: 2022-04-03.

[61] Jan vom Brocke, Alexander Simons, Bjoern Niehaves, Bjorn Niehaves, Kai Reimer, Ralf Plattfaut, and Anne Cleven. Reconstructing the giant: On the importance of rigour in documenting the literature search process. 2009.

[62] Yair Levy and Timothy J Ellis. A systems approach to conduct an effective literature review in support of information systems research. *Informing Science*, 9, 2006.

[63] Google scholar. https://scholar.google.com/.

[64] Oria. https://oria.no/.

[65] Jan vom Brocke, Alan Hevner, and Alexander Maedche. Introduction to design science research. In *Design Science Research. Cases*, pages 1–13. Springer, 2020.

[66] Alan Hevner and Samir Chatterjee. Design science research in information systems. In *Design research in information systems*, pages 9–22. Springer, 2010.

[67] Virginia Braun and Victoria Clarke. Thematic analysis. 2012.

[68] Iuon-Chang Lin and Tzu-Chun Liao. A survey of blockchain security issues and challenges. *Int. J. Netw. Secur.*, 19(5):653–659, 2017.

[69] Dipankar Dasgupta, John M Shrein, and Kishor Datta Gupta. A survey of blockchain from security perspective. *Journal of Banking and Financial Technology*, 3(1):1–17, 2019.

[70] Tam T Huynh, Thuc D Nguyen, and Hanh Tan. A survey on security and privacy issues of blockchain technology. In *2019 international conference on system science and engineering (ICSSE)*, pages 362–367. IEEE, 2019.

[71] Sumit Soni and Bharat Bhushan. A comprehensive survey on blockchain: Working, security analysis, privacy threats and potential applications. In *2019 2nd international conference on intelligent computing, instrumentation and control technologies (ICICICT)*, volume 1, pages 922–926. IEEE, 2019.

[72] Bharat Bhushan, Preeti Sinha, K Martin Sagayam, and J Andrew. Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions. *Computers & Electrical Engineering*, 90:106897, 2021.

[73] Joanna Moubarak, Eric Filiol, and Maroun Chamoun. On blockchain security and relevant attacks. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, pages 1–6. IEEE, 2018.

[74] Saurabh Singh, ASM Sanwar Hosen, and Byungun Yoon. Blockchain security attacks, challenges, and solutions for the future distributed iot network. *IEEE Access*, 9:13938–13959, 2021.

[75] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, DaeHun Nyang, and Aziz Mohaisen. Exploring the attack surface of blockchain: A systematic overview. *arXiv preprint arXiv:1904.03487*, 2019.

[76] N Anita and M Vijayalakshmi. Blockchain security attack: a brief survey. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE, 2019.

[77] A Begum, AH Tareq, M Sultana, MK Sohel, T Rahman, and AH Sarwar. Blockchain attacks analysis and a model to solve double spending attack. *International Journal of Machine Learning and Computing*, 10(2):352–357, 2020.

[78] Huru Hasanova, Ui-jun Baek, Mu-gon Shin, Kyunghee Cho, and Myung-Sup Kim. A survey on blockchain cybersecurity vulnerabilities and possible countermeasures. *International Journal of Network Management*, 29(2):e2060, 2019.

[79] Jieren Cheng, Luyi Xie, Xiangyan Tang, Naixue Xiong, and Boyi Liu. A survey of security threats and defense on blockchain. *Multimedia Tools and Applications*, 80(20):30623–30652, 2021.

[80] Benedikt Putz and Günther Pernul. Detecting blockchain security threats. In *2020 IEEE International Conference on Blockchain (Blockchain)*, pages 313–320. IEEE, 2020.

[81] AKM Haque and Mahbubur Rahman. Blockchain technology: Methodology, application and security issues. *arXiv preprint arXiv:2012.13366*, 2020.

[82] Qi Feng, Debiao He, Sherali Zeadally, Muhammad Khurram Khan, and Neeraj Kumar. A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications*, 126:45–58, 2019.

[83] Yourong Chen, Hao Chen, Yang Zhang, Meng Han, Madhuri Siddula, and Zhipeng Cai. A survey on blockchain systems: Attacks, defenses, and privacy preservation. *High-Confidence Computing*, 2(2):100048, 2022.

[84] A Averin and O Averina. Review of blockchain technology vulnerabilities and blockchain-system attacks. In *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pages 1–6. IEEE, 2019.

[85] Lukas König, Stefan Unger, Peter Kieseberg, Simon Tjoa, and Josef Ressel Center BLOCKCHAINS. The risks of the blockchain a review on current vulnerabilities and attacks. *J. Internet Serv. Inf. Secur.*, 10(3):110–127, 2020.

[86] Related work google drive spreadsheet. https://docs.google.com/spreadsheets/d/1ppPLs9J_VYxv3bSmi29irzVF5EeJMTLdg2wd3Zij6GA/edit?usp=sharing.

[87] Howard Poston. Threat modeling for the blockchain. https://www.howardposton.com/blog/threat-modeling-for-the-blockchain, May 2022.

[88] Python Software Foundation. What is python? executive summary. https://www.python.org/doc/essays/blurb/, May 2022.

[89] Python Software Foundation. tkinter — python interface to tcl/tk. Technical report, May 2022.

[90] John E Grayson. *Python and Tkinter programming*. Manning Publications Co. Greenwich, 2000.

[91] SQL. Alphabetical list of documents. https://www.sqlite.org/doclist.html, May 2022.

[92] Mike Owens. *The definitive guide to SQLite*. Apress, 2006.

[93] Inc. GitHub. Github documentation. https://docs.github.com/en/get-started, May 2022.

[94] Threat modeling repositories. https://github.com/bjornarhoff/Blockchain-ThreatModeling.git.

[95] AHMED S Alghamdi, ALIH Al-Badi, ROOBAEA Alroobaea, and PAMJ Mayhew. A comparative study of synchronous and asynchronous remote usability testing methods. *International Review of Basic and Applied Sciences*, 1(3):61–97, 2013.

[96] Joesph S Dumas and Jean E Fox. Usability testing: Current practice and future directions. In *Human-Computer Interaction*, pages 247–268. CRC Press, 2009.

[97] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.

Appendices

A Survey Questions

Table 19: Survey Questions

Category	Question	Answer option
Demographic	What gender do you identify as?	Multiple Choice
	What is your age?	Multiple Choice
	What is your highest qualification?	Multiple Choice
Background	What is your knowledge about blockchain security?	Linear Scale 1-5
	What is your knowledge about existing threat modeling tools?	Linear Scale 1-5
Product feedback	What problem/goal do you think the application is trying to solve/achieve?	Long answer
	If you can change one thing about the application, what would it be and why?	Long answer
	Why will you continue to use this application? What will stop you from using it in the future?	Long answer
	What do you expect to see in our application in the future?	Long answer
	Why do you think businesses will be interested/not interested in this type of application when developing systems based on blockchain technology?	Long answer
	Are there any features you expected to find but didn't find?	Long answer
	Which feature did not work as expected?	Long answer
User experience	What is your first thought when using the application?	Long answer
	What confused you about the application?	Long answer
	Overall, what's your experience with the application?	Long answer
General feedback	How would you rate the overall quality of the application?	Linear Scale 1-5
	Would you recommend this application to a friend or colleague?	Multiple Choice
	Do you have anything else to add to the quality or use of the application that we have not covered?	Long answer

B Multivocal Literature Review

Security and Privacy Challenges in Blockchain Interoperability - A Multivocal Literature Review

Terje Haugum
terhaug@stud.ntnu.no
Norwegian University of Science and Technology
Trondheim, Norway

Bjørnar Hoff
Norwegian University of Science and Technology
Trondheim, Norway
bjohof@stud.ntnu.no

Mohammed Alsadi
Norwegian University of Science and Technology
Trondheim, Norway
mohammed.alsadi@ntnu.no

Jingyue Li
Norwegian University of Science and Technology
Trondheim, Norway
jingyue.li@ntnu.no

ABSTRACT

Transferring data and value across different blockchains is one of the biggest obstacles to further expansion. Blockchain interoperability allows different networks to communicate and transfer data between them and are increasingly crucial for blockchain applications. However, the concern about security and privacy in blockchain interoperability arises naturally. This work aims to provide the state-of-the-art related to security and privacy challenges in blockchain interoperability. We conducted a multivocal literature review (MLR) and analyzed 16 scientific and 30 grey literature, respectively. We examined different security and privacy challenges related to both blockchain in general and blockchain interoperability approaches such as Notary Schemes, Sidechains and Hashed Time-Lock Contracts. Possible mitigations are analysed, and open challenges that arose from the mitigations are highlighted.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**.

KEYWORDS

Blockchain, security, privacy, interoperability

ACM Reference Format:
Terje Haugum, Bjørnar Hoff, Mohammed Alsadi, and Jingyue Li. 2022. Security and Privacy Challenges in Blockchain Interoperability - A Multivocal Literature Review. In *The International Conference on Evaluation and Assessment in Software Engineering 2022 (EASE 2022), June 13–15, 2022, Gothenburg, Sweden.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3530019.3531345

1 INTRODUCTION

Since the introduction of Bitcoin in 2008 [37], Blockchain, which is a technology for achieving distributed database of records, has gained tremendous popularity. Blockchain is considered as the next disruptive technology under the umbrella of Industry 4.0, and it has assimilated impacts to the internet [11]. The distributed feature of Blockchain coupled with other distinguishable features, such as immutability, transparency, and security has helped the technology to be exploited beyond finance, such as Supply Chain Management, electronic voting, IoT and many others. Further, the introduction of smart contracts has brought programability, which allows users/organizations to build their own applications on top of Blockchain. This allows users to complete transactions or data exchange without the need of any centralized and trusted third-party authority [7].

Blockchain has become a crucial player in the Financial Technology (FinTech) industry. According to Research and Market report [1], The Blockchain market size is projected to grow from USD 4.9 billion in 2021 to USD 67.4 billion by 2026, at a Compound Annual Growth Rate (CAGR) of 68.4% during the forecast period. This huge market is attractive for attackers, thus concerns about security and privacy risks arises naturally. For example, in [32], researchers used Oyente [2] tool to analyse 19.366 existing Ethereum smart contracts which have a total balance of about 3,068,654 million Ethers, approximately equivalent to 30 million USD. The results showed that 8.833 contracts are vulnerable to security attacks. Further, in June 2016, a malicious user stole around 60 million US dollar from Ethereum platform caused by a bug in DAO smart contract. This allowed the attacker to recursively drain the DAO of ether collected from the sale of its token [9]. Such cases are barriers which prevent Blockchain technology from reaching it's full potential. To address these challenges, new security approaches and proposals, and new Blockchain platforms with new features are introduced.

New blockchains have emerged and grown independently with their own features claiming that its more secure and capable to offer better features than the existing platforms. This translates to a critical point in blockchain technology where the different types of blockchains are restricted to their own set of rules. The increased number of unconnected and independent blockchain systems, causes a big fragmentation in the field of research as the majority of blockchain systems operates within silos [25]. From business perspectives, these Blockchain platforms should be able to interact with each other in order enhance buisness processes and

EASE 2022, June 13–15, 2022, Gothenburg, Sweden

ACM ISBN 978-1-4503-9613-4/22/06…$15.00
https://doi.org/10.1145/3530019.3531345

[1] https://www.researchandmarkets.com/reports/5025113/blockchain-market-with-covid-19-impact-analysis
[2] https://github.com/ethereum/oyente

introduce new add-on values for both users and organisations. However, integration with other systems/Blockchains is a non-trivial challenge primarily due to differences with respect to platforms, consensus mechanism, and governance. Todays software enterprise truly relies on collaboration and interaction. Blockchains that focuses on interoperability has, therefore, increased in popularity by researchers and industrial partners in recent years.

There exists several studies on the security and privacy challenges of blockchain technology, but a few of them target security and privacy challenges in interoperable blockchain networks. A systematic literature review about blockchain interoperability [51] stated that the limitation of inter-blockchain communication are security, privacy, lack of control, scalability, and not supporting hybrid systems.

Motivated by the limitation given by [51], this paper focus on security, privacy and vulnerabilities found in interoperable blockchains. We performed a MLR covering both scientific and grey literature. The former is used to analyze current state-of-art regarding security and privacy issues in interoperable blockchains while the latter is used to help us better understand these issue due to lack of research in this field. We developed three research questions and build our search term to seek answers to these questions using various resources. A total of 489 scientific papers and 333 grey literature was found. This set was reduced to 16 and 30 in scientific and grey literature respectively after paper filtering.

The main contributions of this paper are: 1) identify the main security and privacy vulnerabilities targeting blockchain interoperability, 2) identify mitigations to address these vulnerabilities, and 3) highlight the challenges associated with these mitigations.

The structure of the paper is as follows. Details about Blockchain interoperability and currently used approaches are presented in Section 2. Section 3 presents related work. In section 4, we describe our research method. Section 5 reports research results and Section 6 discusses the results. We draw conclusions in section 7.

2 BLOCKCHAIN INTEROPERABILITY

Inspired definition of interoperability by Wegner [53], we defined Blockchain interoperability as the ability to easily share, see, transact, and access information across different blockchain networks without any centralized authority. Providing communication between two blockchains involves one source blockchain and a target blockchain. The source blockchain is responsible for initialising transactions which will be executed on the target blockchain [8]. These transactions can be either to read from the target blockchain's ledger or write to it.

The ability of different blockchain networks to interact meaningfully with one another facilitates seamless integration across diverse systems makes evolution of new use cases possible. This would help in building a customizable Web3 services. For example, interoperable smart contracts could supercharge industries like electronic voting, supply chain, or healthcare, by allowing important business information to be sent back and forth between private networks and public networks in a customizable and controllable manner. Further, blockchain interoperability has an impact on reaching a more decentralized ecosystem which presents an advanced manifestation of blockchain technology's promise to decentralize systems and economies. Instead of having one blockchain like Ethereum processing all the transactions for thousands of decentralized applications, there could be thousands of application-specific blockchains that communicate with one another. Moreover, when blockchains become able to interact with one another, independent markets and business applications that were previously considered entirely separate will be able to more easily transfer data and value. This means organizations and communities that wouldn't typically interact with one another would be able to exchange information, leverage each other's strengths, and cultivate innovation more effortlessly and effectively.

Currently, there are three main strategies to achieve chain interoperation. Details about these strategies along with their approach to reach interoperation is presented hereunder.

Notary scheme: Notary scheme is the simplest strategy to facilitate cross-chain communication. This strategy involves a trusted entity that monitors all events that happen on multiple chains [8]. Notary mechanism is a trusted entity that can claim to one chain that a given event on another chain took place. In practice, exchanges, centralized and decentralized, use a notary scheme.

Sidechains/Relay: Instead of relying on a trusted entity, relays are a mechanism inside a blockchain that can read and validate states in other blockchains. The mainchain (i.e the original blockchain) maintains a ledger of assets connected to the sidechain, giving the mainchain ability to understand changes on the sidechain [8]. A sidechain is referred to as a side blockchain that can interact alongside the mainchain. Simplified, a mainchain considers a sidechain as an extension of itself. In addition, they can be sidechains of each other [4].

Hashed Time-Lock Contracts: Hashed Time-Lock Contract is a well-known approach to achieve atomic cross-chain operations. The technique includes the use of timelocks and hash locks [4]. It provides atomic transactions between parties by committing the trader to provide cryptographic proof , before the timeout, when making the transaction. With this feature, the blockchains need to know much less about each other [8].

Polkadot and Cosmos are the most adapted interoperability solutions currently. Both platforms allow communication between multiple blockchains, but they are built with different architecture and components . Polkadot is a multichain application environment making cross-chain registries and cross-chain computation possible [39]. It can transfer arbitrary data across both permissionless and permissioned blockchains secured by Nominated Proof of Stake (NPos) and the network actors: validators, nominators, collators and fishermen. The transfers are completed through Polkadot's components: Relay chain, Parachains, Parathreads and Bridges. Cosmos [28] is a network of many independent blockchains, named as **Zones**. The first Zone created, called **Cosmos Hub**, is a multi-asset proof-of-stake cryptocurrency. All the Zones are powered by the **Tendermint Byzantine Fault Tollerant (BFT)** consensus. A comparison between Polkadot and Cosmos is provided in Table 1.

Table 1: Specifications of Polkadot and Cosmos

Item	**Polkadot**	**Cosmos**
Type	Heterogeneous	Heterogeneous
Consensus	BABE, uses verifiable random variable to assign slots to validators. GRANDPA for votation on chains	Byzantine Fault Tolerance, Tendermint
Staking Mechanism:	Nominated Proof of Stake	Bonded Proof of Stake
Messaging	Cross-Consensus Message Passing Format (XCM)	Inter-Blockchain Communication (IBC)
Goal:	Allow arbitrary data to be transferred across blockchains in a completely decentralized web where the users are in control, while maintaining security and scalability	Internet of Blockchains. Enable communication between different blockchains in a decentralized way.
Launch date:	May 26th, 2020	March 9th, 2020

3 RELATED WORK

There are tremendous scientific literature about the blockchain technology and its interoperability in general, with a few focusing on security and privacy aspects. Belcior et.al [4] conducted a survey about blockchain interoperability mapping extant literature and classifying studies in different categories. This survey provides a holistic overview of interoperable blockchains with different challenges, future work and standards. It's highlighted that the open issues towards achieving interoperable blockchains are privacy and security. Similarly, authors in [51] conducted a review gathering scientific research on interoperability among heterogeneous blockchains and observed that studies on security risks related to interoperable blockchains are insufficient. The need for blockchain interoperability and how to manage a paradigm shift where blockchains communicate is discussed in [48]. Even authors provide the current state of the art in cross-chain communication, their work does not focus on identifying security and privacy challenges regarding these protocols. On the other hand, [29] strictly follows the formal definition of interoperability and proves that it is impossible for two blockchains to interact with each other. The paper highlights that relaxing the definition gives the possibility to create a 2-in-1 blockchain with two ledgers. Additionally, a survey of all the available cross-blockchain communication solutions is presented in [42]. This survey categorizes the solutions into four categories, sidechains solutions, blockchain routers, smart contracts, and industrial solutions. Furthermore, it compares these categories and discusses their limitations and weaknesses. Authors in [12] propose a proof of concept framework using smart contracts to provide secure communication between heterogeneous blockchains. The proposed system focuses on how Ethereum blockchain can be used to securely share and transfer healthcare data. The system only supports heterogeneous (public and private) blockchains on the Ethereum platform, and not hybrid systems, such as Bitcoin.

Many surveys or literature reviews, such as [3, 5, 13, 14, 17, 21, 21–23, 31, 36, 55, 57], focus on blockchain's security and privacy. However, none of these targets the security and privacy challenge associated with interoperable blockchains. Although [3] targets blockchain security and privacy and examines the vulnerabilities in various blockchain ecosystems components, the study excludes some vulnerable components related to interoperable blockchains.

4 RESEARCH DESIGN

We aim at summarizing the state-of-the-art of vulnerabilities in existing interoperable blockchain networks and research gaps in the field of related challenges. To attain our research goal, we decided to conduct a MLR [19]. MLR is a form of Systematic Literature Review (SLR) which includes grey literature (GL), while a typical SLR use academic peer-reviewed papers only. Generally, GL is any information (not published in books or scientific papers) produced by the private industry or practitioners that is not controlled by any peer-review or publisher [19]. Given that security and privacy in blockchain interoperability is a relatively unsearched field, including GL in our research is necessary for better understanding of the field and allowing us to combine scientific literature with state of the art, produced by practitioners. In this MLR, we focused on answering the following research questions and followed the process proposed in [19].

- **RQ1**: What are the existing security and privacy challenges related to blockchain interoperability?
- **RQ2**: What are the mitigations?
- **RQ3**: What are the open challanges around the mitigations?

RQ1 aims to identify challenges regarding security and privacy, mentioned by researchers and practitioners, in both grey and scientific literature. **RQ2** seeks to provide a overview of the mitigations in the field, while **RQ3** focuses on which challenges follows these mitigations.

4.1 Search Strategy

Due to lack of research in security and privacy in interoperable blockchains, we had to fine-tune the main search string. So far, the most adapted blockchain interoperability solutions are Polkadot and Cosmos. Hence, we opted to use these blockchains as individual terms for our search string. Further, these blockchain names are used in the previously listed research works. This yielded search results suitable to our needs. In order to create the search string, we derived relevant keywords from the research scope and research

questions. Then, we defined the search string using the searching terms shown in Table 2 and their combination as follows.

(X1 **OR** X2 **OR** X3) **AND** (Y1 **OR** Y2) **AND** (Z1 **OR** Z2 **OR** Z3 **OR** Z4)

The selection process used for both scientific and grey literature consists of 4 different stages. In scientific literature the four stages are: **1) Literature search and snowballing**: In this stage, a literature search in both google scholar and Oria [1] (a search engine aggregating research papers from scientific databases, including IEEE Xplore, Springer, ACM Digital library, and Scopus), is performed using the search string. When searching using Google, We limited the results from the search by utilizing its page rank and included the first 8 pages without taking the year of publication into account. Limiting the range of the year of publication will not provide the holistic overview we are in search of. To retrieve grey literature for our research, we applied our search string to Google. Applying the first string (X1∧ Y1 ∧ Z3) resulted in 556 000 results (December 2021). Obviously, we need to rely on Google´s page rank algorithm [30], so limiting our search is necessary. This results in a list of scientific literature. Then, this list is extended using the backward snowballing. **2) Remove duplication:** where duplicates from the generated list are removed. In case of different versions, only the most recent paper will be kept. **3) abstract analysis** where the papers' abstracts are analyzed to excluded irrelevant articles. **4) deep analysis:** where the literature is further further analyzed to determine whether it will be included or excluded. This is done by reading through the papers.

For grey literature, the stages were almost the same except the third stage as grey literature resources have no abstract. For this reason, the introduction together with the title are taken into consideration and analyzed to decide whether they give sufficient information about the content of the specific literature in order to categorize them. In order to maintain the validity of each primary study, a standardized rating approach is used. The goal is to reduce the number of studies regarding their relevance. Since quality assessment for grey literature is more complicated than scientific literature, we followed the approach in [19] to determine whether a source is valid and free of bias. [19] points out that "there is no one-size-fits-all quality model", so we decided to follow their quality assessment checklist for grey literature, considering the authority of the producer, methodology, objectivity, date, novelty, impact, and outlet type. Table 3 presents a quality assessment checklist with a 3-point Likert scale (yes=1, partly=0.5, no=0) combined with the bare binary decision (true=1, false=0), to assign scores to assessment criteria questions. Based on these scorings, we defined a threshold for inclusion and calculated the average of each score, and finally rejected grey literature sources that were lower than 0.5 with a range from 0 to 1.

The MLR was conducted during the autumn of 2021. Using the aforementioned search strings, we found **489** scientific literature and **333** grey literature for further analysis. After filtering, we identified 16 primary scientific papers and and 30 grey literature. We analyzed the data in these primary studies using thematic analysis to answer our research questions.

5 RESULTS

In this section, we present results of our research questions.

5.1 RQ1: security and privacy challenges

After data analysis, we have identified the following vulnerabilities and corresponding attacks for security and privacy:

- Wormhole Attack: The wormhole attack happens when two malicious actors infiltrate the network creating a communication tunnel between them and announcing their short path of transaction handling to the other nodes. This will exclude the other nodes from taking part in the existing transaction, potentially stealing fees intended for the honest actors [33]. Hash Time Lock Contract (HTLC) is vulnerable to this attack as it encompasses no more than two rounds of communication which wormhole takes advantage of [33].
- Collusion Attack [18]: Collusion attacks relate to collaboration between multiple nodes generated by a secret agreement in order to behave maliciously. Side Chains and HTLC are vulnerable to this attack, but notary schemes are not due to their reliance on centralized third parties.
- DoS (Denial of Service) in Atomic Swap Vulnerabilities: Atomic swaps, also known as atomic cross-chain trading, offer a way to swap cryptocurrencies peer-to-peer from different blockchains directly without the requirement for a third party, such as an exchange. Atomic Swap utilizes the strategy of HTLC. Atomic Swap is vulnerable to DoS attack [15]. A malicious party could inevitably lock the assets as the initiator of the swap is in control of the abortion method.
- Loss of fund in Atomic Swap: is a security issue where funds of the parties involved could be lost if they go offline for a longer period than the timeout before the withdrawing execution and after giving their secret [44]. HTLC based interoperability solutions are vulnerable to this attack.
- Double Spending Attack: in interoperable blockchains, this attack occurs when one user has multiple accounts in a network (or are in collusion with multiple accounts). The user can first have a transaction with a user on the other network and receive service from an honest client. Afterwards, he can send the same money again from another account and again receive the service from the honest client as he is in another network unknowing of the double-spending attack [46]. Reliance on third parties in Notary schemes prevents collusion with multiple accounts. Thus Notary schemes are not subject to this attack, but Side Chains and HTLC are vulnerable.
- No transaction finality [27, 47]: Finality is used to guarantee that transactions cannot be altered, reversed or cancelled when the transaction becomes final. The longer time period for finality gives more time for additional checks to be performed and reported to the network. In HTCL, contracts are locked for a specific time, which guarantees to reach a form of finality by the end of this time. Strategies utilizing notary schemes and side chains should address this attack.
- Timing attack: Side Chains are independent blockchains that employ their own consensus models and block parameters to enhance transaction processing in terms of time. They

Table 2: Searching terms

X1. Blockchain Interoperability **X2.** Polkadot **X3.** Cosmos	**Y1.** Privacy **Y2.** Security	**Z1.** Issue **Z2.** Attack **Z3.** Challenge **Z4.** Solution

Table 3: Quality assessment for grey literature [19]

Criteria	Questions	Possible answers
Authority of the producer	Is the publishing organization reputable?	**1:** The organization is reputable; **0.5:** The organization is not well known; **0:** The organization is unknown
	Is an individual author associated with a reputable organization?	**1:** True; **0:** False
	Has the author published other work in the field?	**1:** True; **0:** False
	Does the author have expertise in the area?	**1:** Author has expertise in the area; **0:** Author has not expertise in the area
Methodology	Does the source have a clearly stated aim?	**1:** True; **0:** False
	Does the work cover a specific question?	**1:** Yes; **0.5:** Not clear; **0:** No
Objectivity	Is the statement in the sources as objective as possible? Or, is the statement a subjective opinion?	**1:** Objective; **0.5:** Partially objective; **0:** Subjective
	Does the work seem to be balanced in presentation?	**1:** Balanced; **0.5:** Partially balanced; **0:** Not balanced
	Are the conclusions supported by the data?	**1:** Supported; **0.5:** Partially supported; **0:** Not supported
Date	Does the item have a clearly stated date?	**1:** Clearly stated date; **0:** No date
Position w.r.t. related sources	Have key related GL or formal sources been linked to / discussed?	**1:** True; **0:** False
Novelty	Does it enrich or add something unique to the research?	**1:** Enriches our reasearch; **0.5:** Partially enriches our research; **0:** Does not enrich our research
Outlet type	Outlet measures	- 1st tier GL (measure=1): High outlet control/ High credibility: Books, magazines, theses, government reports, white papers - 2nd tier GL (measure=0.5): Moderate outlet control/ Moderate credibility: Annual reports, news articles, presentations, videos, Q/A sites (such as StackOverflow), Wiki articles - 3rd tier GL (measure=0): Low outlet control/ Low credibility: Blogs, emails, tweets

can also be used for interoperability. For instance, Loom is a sidechain for developing dApps on Ethereum. In [50], Amritraj Singh et al. introduce the potential issue on Loom where the transaction history is periodically updated on the mainchain, and an old version of the transaction history is therefore located on the sidechain periodically as well. This can make the sidechains vulnerable to timing attacks between the updates where the Side Chain is not updated.

- Incompatible cryptography: An interoperable blockchain network may use different signature algorithms and different hashes. This may cause complexity and transaction challenges as a consequence of more functionality and managing different signature algorithms [54].
- Single Point of Failure: A single point failure could be a critical issue for the use of third-party software and the two-way-peg [50]. The two-way-peg method is a solution for transferring coins from a mainchain to a sidechain. There are different methods of implementing two-way-peg, e.g., centralized and federated. [35] describes different types of solutions for gaining interoperability and emphasizes an issue that most notary scheme solutions build upon the trust of the notary. A notary failure could therefore induce a single point failure.
- Private Key Attacks: Storing private keys is similar to password management in many ways. If they end up in a malicious actor's hands, the system/account with the vulnerable

private key is ultimately in ownership of the malicious actor. Having systems for encryption is therefore extremely important. Nonetheless, even with these types of systems in place, the ultimate job of securely storing the password lies with the users [18].

- Sybil Attack: is an attack where a majority of actors of malicious behaviour could potentially lead to a critical security weakness of any blockchain. This attack is mentioned in many research papers on blockchain in general and is still an essential aspect of blockchain interoperability [6][38].
- Eclipse Attack: Although an attack that requires an artificial environment to manipulate a specific node is rare, it is still possible. Eclipse attack is a type of attack which takes advantage of the distributed feature of Blockchain. Attackers aim to isolate a specific user(s) rather than attacking the whole network [24]. Similar to the sybil attack, the eclipse attack is closely related to general blockchain vulnerability. However, it will be inherited by any interoperable system.
- Denial of Service (DoS): Actively preventing transactions and key transfers, locking assets, or other types of locking in the system is seen as DoS attacks. Such attacks should be mitigated in blockchain interoperability systems as best as possible [6, 15, 26, 44].
- No liveness: In the Adversary Capabilities in Practical Byzantine Fault Tolerance [52], the author reviews Polkadot's GRANDPA BFT protocol and shows mathematically that it can't achieve liveness if the adversary is allowed to reschedule the message delivery order in the underlying networks. [45] highlighted that liveness cannot be achieved with asynchronous communication protocols due to the unset response time of the communicating parties.
- Fraud-Proof Attack: Recently, several proof-based attempts at solving Ethereum's scalability problem have been introduced. In [44], an issue with an anchored blockchain on Ethereums sidechain called PLASMA which uses proof of exit and frauds for consensus has been identified. PLASMA does not deal with security where an exit proof has been given, but a malicious actor challenges with a fraud-proof, and the honest actor is offline for the entire challenge period (approximately 7 days). If this happens, the malicious actor could steal the tokens.
- Identifier Leaks in HTLC: is a privacy vulnerability described in [33], in which the payment path holds identifiers from the HTLC that is leaked and could be observed, thus making transaction and involved parties publicly visible [16, 18].

Table 4 provides a comprehensive comparison between the various security and privacy vulnerabilities and attacks, their classification based on whether they target blockchain in general or interoperable blockchains. Further, it matches these vulnerabilities to the different strategies for blockchain interoperability by highlighting the interoperability solutions which are subject to these attacks .

5.2 RQ2: mitigations

As blockchain technology grows, the need for cross-chain communications is essential for further adaption. When connecting both private and public blockchain networks together, it is necessary to do mitigation in order to secure the network and provide the right privacy policies.

Since Cosmos and Polkadot use Proof-of-Stake (PoS), they developed shared security models to share their security across the network and to prevent wealthy attackers from attacking smaller interconnected blockchains with a lower bounded stake. For example, Interchain Security Hub is introduced in Cosmos to share its set of validators with participating (child) chains. Similarly, a shared security model is introduced in Polkadot to define how all parachains connected to the Relay chain can economically benefit from the security provided by their validators [49], hence, providing stronger guarantees for security.

On the other hand, similar approaches to address the privacy issues related to interoperable blockchains are presented by Cosmos and Polkadot. In Cosmos, Secret Network is introduced as a base-layer blockchain network built using the Cosmos SDK. It is an independent blockchain network that supports smart contracts and data privacy by default, and it is capable of interoperability within the Cosmos network using Interblockchain Communication protocol (IBC). Manta Network is a project on Polkadot aiming for developing a privacy-preserving protocol for the DeFi stack on Polkadot. It offers two smart contract layers, the Decentralized Anonymous Payment (DAP) protocol and the Decentralized Anonymous Exchange (DAX) protocol. DAX is based on zk-SNARK and an automated market maker (AMM), and allows the user to anonymously trade private tokens on the platform [10, 34]. Polkdaot has another on-going project called Phale which aims for building trust in the computation cloud.

Details about mitigation approaches to security and privacy issues is presented in Table 5. It can be noticed that there are vulnerabilities where the primary papers do not provide a proper solution. This could be a result of considering several challenges while providing mitigation to only a few. For instance Malavolta et al. [33] mitigate the issue with the Wormhole Attack in HTLC without having a solution for Denial of Service and asset locking in atomic swaps. From the scientific papers, most of the mitigation to potential vulnerabilities via their own solutions are mostly presented without much evaluations. Without further investigation, we cannot reach a conclusion that the proposed mitigation implied further security and privacy challenges.In the grey literature, [54] presents the potential issue of multiple signature algorithms within different networks. Albeit a solution is not proposed, the necessity of handling different algorithms with some form of protocol in order to maintain secure interoperability between blockchains is an important information.

5.3 RQ3: mitigations' challenges

Here, we present the challenges that arose from the mitigations presented from the primary papers.

Due to blockchain trilemma [20], **scalability** is heavily influenced by the security. From most of the mitigations mentioned for RQ2, a decrease in scalability where the overall transaction speed is reduced is a negative outcome from the mitigation itself. Our surveyed studies mentioned that the following mitigation makes the system less scalable.

Table 4: Existing security and privacy challenges related to blockchain interoperability. Yes: relevant. No: Irrelevant.

Vulnerability	Vulnerability Scope		Interoperability Approach		
	General Blockchain	Interoperable Blockchain	Notary	Side Chains/Relay	HTLC
Wormhole attack	Yes	Yes	No	No	Yes
Collusion attack	Yes	Yes	No	Yes	Yes
DoS in Atomic Swap	No	Yes	No	No	Yes
Loss of fund in Atomic Swap	No	Yes	No	No	Yes
Double Spending Attack	Yes	Yes	No	Yes	Yes
No transaction finality	Yes	Yes	Yes	Yes	No
Timing Attack	Yes	Yes	No	Yes	Yes
Incompatible cryptography	No	Yes	Yes	Yes	Yes
Single point failure	Yes	Yes	Yes	Yes	No
Private key attacks	Yes	Yes	Yes	Yes	Yes
Sybil Attacks	Yes	Yes	Yes	Yes	Yes
Eclipse Attack	Yes	Yes	Yes	Yes	Yes
Denial of Service (DoS)	Yes	Yes	Yes	Yes	Yes
No Liveness	Yes	Yes	Yes	Yes	Yes
Fraud-proof attack	No	Yes	No	Yes	No
Identifier Leaks in HTLC	No	Yes	No	No	Yes

- Anonymous multi-hop locks with more rounds for solving the issues with HTCL [33].
- Private swaps for solving the identifier leaks inherited by the issue with HTCL [16].
- Mitigation to generate finality by having a wait-time for each transaction described by the developers of Bifrost and Koens et al. [27, 47].
- Implementing a two-way-peg based on simplyfied payment verification [50]. This slows the transactions due to the verification that is needed from both parties.
- The solution of having three observers on each network in order to prevent double spending attacks. This solution was presented by Kuheli Sai and David Tipper [46]. With an increase of nodes, issues, work for the observers, networks etc. will lead to possible scalability-issues.
- In order to improve the security of private keys, the devlopers of Bifrost [47] presented a solution with multiple signatures. This would lead to a slower transaction speed.

A note is that the solution with anonymous multi-hop locks by Malavolta et al.[33] tested the transaction speed and found it performing well in comparison to the original HTCL based system.

For an extra added implementation, there is often an extra added **complexity** to the blockchain system that could generate issues for implementation, updates, and further expanding the blockchain. In the mitigations for improving the security of private keys by implementing elliptic curve diffie-hellman presented by [26] and [33] generate a complexity to the system. The same issue is derived by preventing collusion in HTCL-schemes by implementing another layer on top of the interledger protocol [26] as well as the solution to the double spending attack presented by [46].

6 DISCUSSION

Based on the selection and analysis result, it's clear that there is a lack of research being done in security and privacy regarding blockchain interoperability. Related studies, such as [4, 45, 56], focused on interoperability solutions in general. To the best of our knowledge, no MLRs exist on security and privacy issues within blockchain interoperability and the corresponding mitigation. Therefore it was not possible to take inspiration from previous works other than from MLRs focusing on other types of research. Our MLR provides a list of specific vulnerabilities within different types of systems. Some of these systems are used by blockchains in order to induce interoperability between blockchains. However, there are other vulnerabilities which is not covered in our work such as Code Exploitation in smart contracts.

Mitigations: after analysing proposed mitigations for potential security and privacy challenges within blockchain interoperability, we found that the grey literature provided no real descriptive work. This is mainly due to the fact that forum posts, blogs and websites necessarily did not provide a solution to their presented issues. Within the scientific literature, we found that the issue with asset locking using HTCL was not specifically described with a solution. Moreover, most of the solutions to other challenges were not reaching a complete mitigation, meaning they did not provide 100% secure solutions. For instance, the solution to collusion attack on the Interledger Protocol (ILP) where Khosla et al. [26] implemented a layer on top of the ILP. This made it significant harder to collaborate, however, not impossible.

Mitigations' challenges: Our goal was to present the challenges arising from the mitigations. The pool of grey literature did not provide any information about these challenges, but it highlights that security and privacy are two important topics for further research and development, in order to create well-regulated blockchain networks. Existing blockchain interoperability solutions,

Table 5: Link between vulnerabilities from RQ1 and mitigation from RQ2.

Attacks and vulnerabilities	Mitigation(s)	Reference(s)
Wormhole attack	Anonymous multi-hop locks (AMHL)	[33]
Collusion attack	Additional communication layer. Exmaple with CEPA-layer on top of Interledger Protocol	[26]
Denial of Service (DoS) in Atomic Swap		
Loss of funds in Atomic Swap		
Double spending attack	Disincentivizing mechanism with three observers	[46]
	Fee for transactions for all invlolved parties (Cosmos)	[44]
No transaction finality	Implement a wait-time of x	[47][27]
Timing-attack	Anonymous multi-hop locks (AMHL)	[33]
	Additional communication layer, e.g., with CEPA-layer on top of Interledger Protocol	[26]
	Private swaps with the use of secret release	[16]
Incompatible cryptography		
Single point failure	Do not utilize third party software	[38]
	To mitigate single point failure in two-way-peg you could implement simplified payment verification	[50]
	Shared security (Polkadot and Cosmos)	[41][43]
Private key attack	Using a well researched encryption and key exchange algorithm	[47][33][26]
Sybil attack	Make an alteration to the regular Proof of Stake consensus algorithm in order to make it harder for malicious actors wanting to gain a majority-control, e.g., Multi-tokens proof of stake (MPoS)	[38]
Eclipse attack	Generate a large routing table able to hold at least some honest nodes	[6]
Denial of Service (DoS)	Presenting a protocol yielding proof of finality and liveness	[52]
No liveness		
Fraud-proof attack		
Identifier leaks in HTLC	Anonymous multi-hop locks (AMHL)	[33]
	Additional communication layer. Exmaple with CEPA-layer on top of Interledger Protocol	[26]
	Private swaps with the use of secret release	[16]

such as Polkadot and Cosmos, constantly improve their security and privacy solutions by team members and collaboration [2, 40, 41]. At the time we conduct this review, some features on security and privacy are introduced by the existing interoperability solutions. So, the lack of knowledge and research in the field is raised naturally.

In the scientific literature, scalability and complexity are the open challenges around the presented results of our second research question. However, these are challenges outside of the scope of this paper. None of the primary papers imply their further implications to security and privacy issues. One could say that increasing complexity to a system might lead to security risks. Humans tends to make mistakes, especially doing complicated tasks such as implementing complex systems. This might lead to mistakes in code which in turn could make the system vulnerable to attacks.

7 CONCLUSION AND FUTURE WORK

We performed a MLR on security and privacy challenges in blockchain interoperability. We systematically analyzed 16 scientific literature and 30 informative grey literature. We examined different security and privacy challenges, mitigation, open challenges that arose from the mitigations, and potential future research. By including grey literature in our review, we achieve a broader knowledge base and provide data not found within published literature. In addition, grey literature fosters a balanced and more comprehensive picture. In this MLR, we scrutinized blockchain interoperability in general and existing solutions, e.g., Cosmos and Polkadot. By exploring each security and privacy challenges in interoperable blockchains, we have identified several vulnerabilities, such as Hash Time Lock

Contract, private key attacks, network analysis, and so on. In addition, we have summarized the state-of-the-art mitigation against the identified vulnerabilities and the limitations of the mitigation.

In the future, we plan to evaluate and improve some of the proposed mitigation approaches to address the security and privacy issues of blockchain interoperability. In addition, we want to investigate the level of independency between security and privacy vulnerabilities related to blockchain interoperability and characteristics of the underlying consensus algorithms.

ACKNOWLEDGMENTS

This work is jointly supported by the National Key Research and Development Program of China (No. 2019YFE0105500) and the Research Council of Norway (No. 309494).

REFERENCES

[1] 2022. Oria Search Engine. http://oria.no/
[2] Billy Rennekamp Aditya, Gavin. 2021. Interchain Security. https://github.com/cosmos/gaia/blob/main/docs/interchain-security.md.
[3] Nils Amiet. 2021. Blockchain Vulnerabilities in Practice. *Digital threats (Print)* 2, 2 (2021), 1–7.
[4] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2021. A survey on blockchain interoperability: Past, present, and future trends. *ACM Computing Surveys (CSUR)* 54, 8 (2021), 1–41.
[5] Jorge Bernal Bernabe, Jose Luis Canovas, Jose L Hernandez-Ramos, Rafael Torres Moreno, and Antonio Skarmeta. 2019. Privacy-preserving solutions for blockchain: Review and challenges. *IEEE Access* 7 (2019), 164908–164940.
[6] Jeff Burdges, Alfonso Cevallos, Peter Czaban, Rob Habermeier, Syed Hosseini, Fabio Lama, Handan Kilinç Alper, Ximin Luo, Fatemeh Shirazi, Alistair Stewart, and Gavin Wood. 2020. Overview of Polkadot and its Design Considerations. *CoRR* abs/2005.13456 (2020). https://arxiv.org/abs/2005.13456
[7] Vitalik Buterin. 2013. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/wiki/White-Paper.
[8] Vitalik Buterin. 2016. Chain interoperability. *R3 Research Paper* (2016).
[9] Vitalik Buterin. 2016. Critical update re: DAO vulnerability. *Ethereum Blog, June* (2016).
[10] Shumo Chu, Yu Xia, and Zhenfei Zhang. 2021. Manta: a Plug and Play Private DeFi Stack. (2021).
[11] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, et al. 2016. Blockchain technology: Beyond bitcoin. *Applied Innovation* 2, 6-10 (2016), 71.
[12] Gaby G Dagher, Chandra L Adhikari, and Tyler Enderson. 2017. Towards secure interoperability between heterogeneous blockchains using smart contracts. In *Future Technologies Conference (FTC)*. 73–81.
[13] Dipankar Dasgupta, John M Shrein, and Kishor Datta Gupta. 2019. A survey of blockchain from security perspective. *Journal of Banking and Financial Technology* 3, 1 (2019), 1–17.
[14] Francisco José de Haro-Olmo, Ángel Jesús Varela-Vaca, and José Antonio Álvarez-Bermejo. 2020. Blockchain from the perspective of privacy and anonymisation: a systematic literature review. *Sensors* 20, 24 (2020), 7171.
[15] Martijn de Vos, Can Umut Ileri, and Johan Pouwelse. 2021. XChange: A Universal Mechanism for Asset Exchange between Permissioned Blockchains. *World Wide Web* 24 (2021). https://doi.org/10.1007/s11280-021-00870-x https://doi.org/10.1007/s11280-021-00870-x.
[16] Apoorvaa Deshpande and Maurice Herlihy. 2020. Privacy-Preserving Cross-Chain Atomic Swaps. In *Financial Cryptography and Data Security*. Springer International Publishing, Cham, 540–549.
[17] Qi Feng, Debiao He, Sherali Zeadally, Muhammad Khurram Khan, and Neeraj Kumar. 2019. A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications* 126 (2019), 45–58.
[18] John Flood and Adrian McCullagh. 2020. Blockchain's future: Can the decentralized blockchain community succeed in creating standards? *The Knowledge Engineering Review* 35 (2020). https://doi.org/10.1017/S0269888920000016
[19] Vahid Garousi, Michael Felderer, and Mika V Mäntylä. 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology* 106 (2019), 101–121.
[20] Seth Gilbert and Nancy Lynch. 2002. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. *SIGACT News* 33, 2 (June 2002), 51–59. https://doi.org/10.1145/564585.564601
[21] Harry Halpin and Marta Piekarska. 2017. Introduction to Security and Privacy on the Blockchain. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 1–3.
[22] Huru Hasanova, Ui-jun Baek, Mu-gon Shin, Kyunghee Cho, and Myung-Sup Kim. 2019. A survey on blockchain cybersecurity vulnerabilities and possible countermeasures. *International Journal of Network Management* 29, 2 (2019), e2060.
[23] Ryan Henry, Amir Herzberg, and Aniket Kate. 2018. Blockchain access privacy: Challenges and directions. *IEEE Security & Privacy* 16, 4 (2018), 38–45.
[24] Md Rafiqul Islam, Muhammad Mahbubur Rahman, Md Mahmud, Mohammed Ataur Rahman, Muslim Har Sani Mohamad, et al. 2021. A Review on Blockchain Security Issues and Challenges. In *2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC)*. IEEE, 227–232.
[25] Hai Jin, Xiaohai Dai, and Jiang Xiao. 2018. Towards a Novel Architecture for Enabling Interoperability amongst Multiple Blockchains. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 1203–1211. https://doi.org/10.1109/ICDCS.2018.00120
[26] Akash Khosla, Vedant Saran, and Nick Zoghb. 2018. Techniques for Privacy Over the Interledger. *University of California* (2018).
[27] T. Koens and E. Poll. 2019. Assessing interoperability solutions for distributed ledgers. *Pervasive and Mobile Computing* 59 (2019), 101079. https://doi.org/10.1016/j.pmcj.2019.101079
[28] Jae Kwon and Ethan Buchman. 2019. Cosmos whitepaper.
[29] Pascal Lafourcade and Marius Lombard-Platet. 2020. About blockchain interoperability. *Inform. Process. Lett.* 161 (2020), 105976.
[30] Amy N Langville and Carl D Meyer. 2011. *Google's PageRank and beyond*. Princeton university press.
[31] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. 2020. A survey on the security of blockchain systems. *Future Generation Computer Systems* 107 (2020), 841–853.
[32] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. 2016. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 254–269.
[33] Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. 2018. Anonymous Multi-Hop Locks for Blockchain Scalability and Interoperability. Cryptology ePrint Archive, Report 2018/472. https://ia.cr/2018/472.
[34] Camron Miraftab. 2021. Privacy-Preserving DeFi Stack on Polkadot. *Rarestone* (2 2021), 1. https://rarestone.capital/privacy-preserving-defi-stack-on-polkadot/.
[35] Monika and Rajesh Bhatia. 2020. Interoperability Solutions for Blockchain. In *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*. 381–385. https://doi.org/10.1109/ICSTCEE49637.2020.9277054
[36] Joanna Moubarak, Eric Filiol, and Maroun Chamoun. 2018. On blockchain security and relevant attacks. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*. IEEE, 1–6.
[37] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* (2008), 21260.
[38] Yan Pang. 2020. A New Consensus Protocol for Blockchain Interoperability Architecture. *IEEE Access* 8 (2020), 153719–153730. https://doi.org/10.1109/ACCESS.2020.3017549
[39] Polkadot. 2021. Polkadot Technology. https://polkadot.network/technology/
[40] Polkadot. 2021. Privacy Policy. *Polkadot* (2021), 1. https://polkadot.network/privacy/.
[41] Polkadot. 2021. Security of the network. *Polkadot* (2021), 1. https://wiki.polkadot.network/docs/learn-security.
[42] Ilham A Qasse, Manar Abu Talib, and Qassim Nasir. 2019. Inter blockchain communication: A survey. In *Proceedings of the ArabWIC 6th Annual International Conference Research Track*. 1–6.
[43] Billy Rennekamp. 2021. Interchain Security is Coming to the Cosmos Hub. *Cosmos* (6 2021), 1. https://blog.cosmos.network/interchain-security-is-coming-to-the-cosmos-hub-f144c45fb035.
[44] Peter Robinson. 2020. Consensus for Crosschain Communications. *PegaSys* abs/2004.09494 (04 2020).
[45] Peter Robinson. 2021. Survey of crosschain communications protocols. *Computer Networks* 200 (2021), 108488. https://doi.org/10.1016/j.comnet.2021.108488
[46] Kuheli Sai and David Tipper. 2019. Disincentivizing Double Spend Attacks Across Interoperable Blockchains. In *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. 36–45. https://doi.org/10.1109/TPS-ISA48467.2019.00014
[47] Eder J. Scheid, Timo Hegnauer, Bruno Rodrigues, and Burkhard Stiller. 2019. Bifröst: a Modular Blockchain Interoperability API. In *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. 332–339. https://doi.org/10.1109/LCN44214.2019.8990860
[48] Stefan Schulte, Marten Sigwart, Philipp Frauenthaler, and Michael Borkowski. 2019. Towards blockchain interoperability. In *International conference on business process management*. Springer, 3–10.

[49] SEQ. 2020. Polkadot — An Early In-Depth Analysis — Part Three— Limitations and Issues. *Seq* (7 2020), 1–3. https://cryptoseq.medium.com/polkadot-an-early-in-depth-analysis-part-three-limitations-and-issues-d8b0a795a3e.

[50] Amritraj Singh, Kelly Click, Reza M. Parizi, Qi Zhang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2020. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications* 149 (2020), 102471. https://doi.org/10.1016/j.jnca.2019.102471

[51] Manar Abu Talib, Sohail Abbas, Qassim Nasir, Fatima Dakalbab, Takua Mokhamed, Khawla Hassan, and Khaldoun Senjab. 2021. Interoperability Among Heterogeneous Blockchains: A Systematic Literature Review. *Trust Models for Next-Generation Blockchain Ecosystems* (2021), 135–166.

[52] Yongge Wang. 2021. The Adversary Capabilities in Practical Byzantine Fault Tolerance. In *Security and Trust Management*, Rodrigo Roman and Jianying Zhou (Eds.). Springer International Publishing, Cham, 20–39.

[53] Peter Wegner. 1996. Interoperability. *ACM Computing Surveys (CSUR)* 28, 1 (1996), 285–287.

[54] World Bank Group. 2020. BlockchainInteroperability. https://documents1.worldbank.org/curated/en/373781615365676101/pdf/Blockchain-Interoperability.pdf. Online; accessed 13 December 2021.

[55] Efpraxia Zamani, Ying He, and Matthew Phillips. 2020. On the security risks of the blockchain. *Journal of Computer Information Systems* 60, 6 (2020), 495–506.

[56] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J. Knottenbelt. 2021. SoK: Communication Across Distributed Ledgers. In *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, Berlin, Heidelberg, 3–36.

[57] Rui Zhang, Rui Xue, and Ling Liu. 2019. Security and privacy on blockchain. *ACM Computing Surveys (CSUR)* 52, 3 (2019), 1–34.

C Database

Tables (17)

Name	Type	Schema
blockCreationThreat		CREATE TABLE "blockCreationThreat" ("ThreatID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER
blockchains		CREATE TABLE blockchains (BlockchainID INTEGER PRIMARY KEY, B_name text NOT NULL, ConsensusID text NOT NULL, BtypeID text NOT NULL, CryptographyID BOOLEAN NOT NULL, NetworkTypeID text NOT NULL, FOREIGN KEY (BtypeID) REFERENCES btype (BtypeID), FOREIGN KEY (CryptographyID) REFERENCES cryptography (CryptographyID) FOREIGN KEY (ConsensusID) REFERENCES consensus (ConsensusID) FOREIGN KEY (NetworkTypeID) REFERENCES networkType(NetworkTypeID))
BlockchainID	INTEGER	"BlockchainID" INTEGER
B_name	text	"B_name" text NOT NULL
ConsensusID	text	"ConsensusID" text NOT NULL
BtypeID	text	"BtypeID" text NOT NULL
CryptographyID	BOOLEAN	"CryptographyID" BOOLEAN NOT NULL
NetworkTypeID	text	"NetworkTypeID" text NOT NULL
btype		CREATE TABLE "btype" ("BtypeID" INTEGER, "Btype_name" TEXT)
BtypeID	INTEGER	"BtypeID" INTEGER
Btype_name	TEXT	"Btype_name" TEXT
consensus		CREATE TABLE "consensus" ("ConsensusID" INTEGER, "Consensus_name" TEXT,

1

Name	Type	Schema
		"BtypeID" INTEGER)
ConsensusID	INTEGER	"ConsensusID" INTEGER
Consensus_name	TEXT	"Consensus_name" TEXT
BtypeID	INTEGER	"BtypeID" INTEGER
consensusThreat		CREATE TABLE "consensusThreat" ("ThreatID" INTEGER, "ConsensusID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER
ConsensusID	INTEGER	"ConsensusID" INTEGER
cryptography		CREATE TABLE cryptography(CryptographyID INTEGER PRIMARY KEY, Bol_cryptography BOOLEAN NOT NULL)
CryptographyID	INTEGER	"CryptographyID" INTEGER
Bol_cryptography	BOOLEAN	"Bol_cryptography" BOOLEAN NOT NULL
cryptographyThreat		CREATE TABLE "cryptographyThreat" ("ThreatID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER
errorThreat		CREATE TABLE "errorThreat" ("ThreatID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER
interoperabilityThreat		CREATE TABLE "interoperabilityThreat" ("ThreatID" INTEGER, "StrategyID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER
StrategyID	INTEGER	"StrategyID" INTEGER
networkThreat		CREATE TABLE "networkThreat" ("ThreatID" INTEGER, "BtypeID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER
BtypeID	INTEGER	"BtypeID" INTEGER

2

Name	Type	Schema
networkType		CREATE TABLE "networkType" ("NetworkTypeID" INTEGER, "Network_name" TEXT)
NetworkTypeID	INTEGER	"NetworkTypeID" INTEGER
Network_name	TEXT	"Network_name" TEXT
networkTypeThreat		CREATE TABLE "networkTypeThreat" ("ThreatID" INTEGER, "NetworkTypeID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER
NetworkTypeID	INTEGER	"NetworkTypeID" INTEGER
strategy		CREATE TABLE "strategy" ("StrategyID" INTEGER, "Strategy_name" TEXT)
StrategyID	INTEGER	"StrategyID" INTEGER
Strategy_name	TEXT	"Strategy_name" TEXT
stride		CREATE TABLE "stride" ("StrideID" INTEGER, "Stride_Name" TEXT)
StrideID	INTEGER	"StrideID" INTEGER
Stride_Name	TEXT	"Stride_Name" TEXT
strideThreat		CREATE TABLE "strideThreat" ("ThreatID" INTEGER, "StrideID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER
StrideID	INTEGER	"StrideID" INTEGER
threat		CREATE TABLE "threat" ("ThreatID" INTEGER, "Threat_Name" TEXT, "Description" TEXT, "URL" TEXT)
ThreatID	INTEGER	"ThreatID" INTEGER
Threat_Name	TEXT	"Threat_Name" TEXT
Description	TEXT	"Description" TEXT
URL	TEXT	"URL" TEXT

3

Name	Type	Schema
transactionThreat		CREATE TABLE "transactionThreat" ("ThreatID" INTEGER)
ThreatID	INTEGER	"ThreatID" INTEGER

Indices (0)

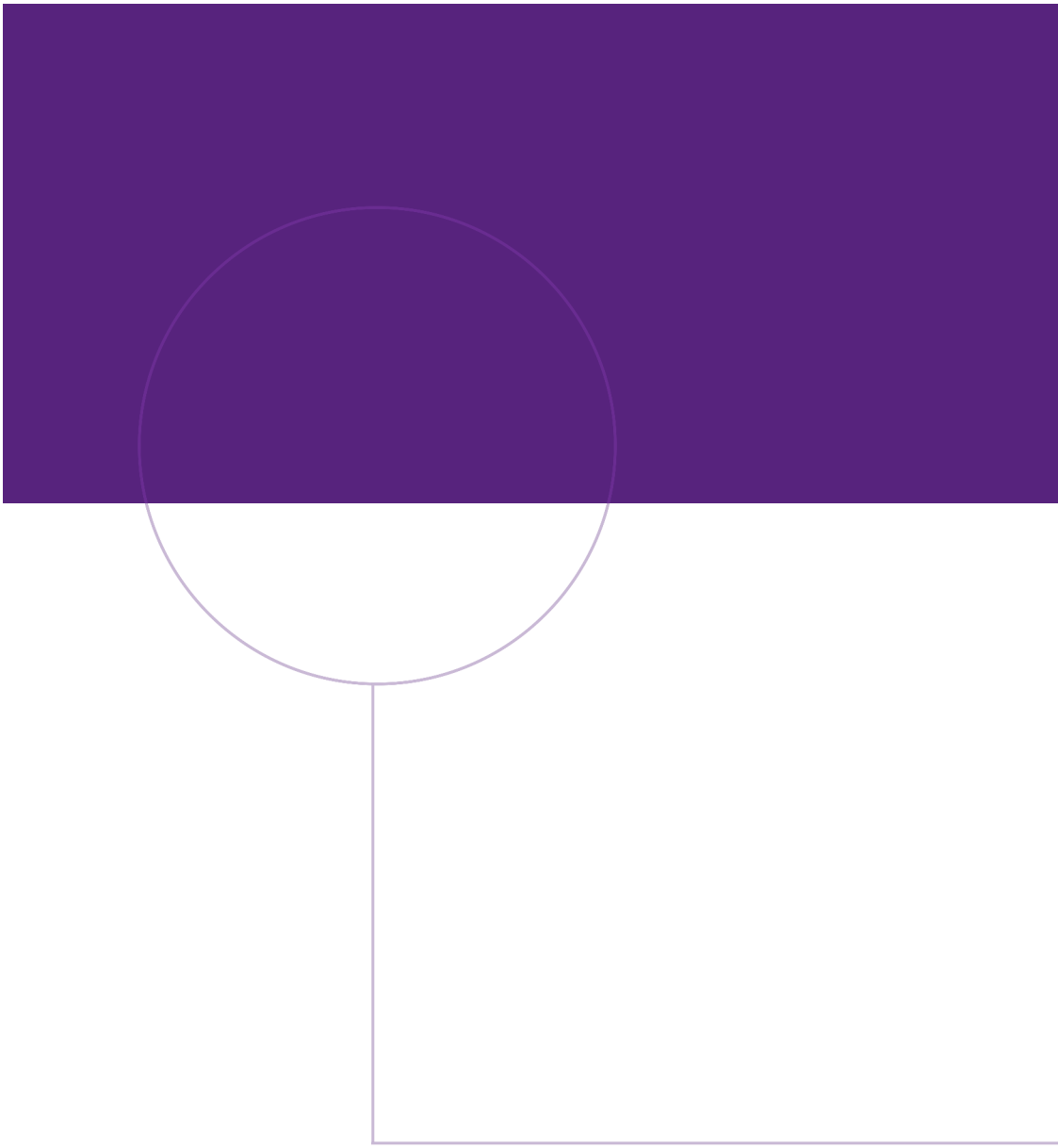
Name	Type	Schema

Views (0)

Name	Type	Schema

Triggers (0)

Name	Type	Schema



NTNU

Norwegian University of
Science and Technology