Cybersecurity applications of Blockchain technologies

by
María del Mar Giménez Aguilar

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Technology

Universidad Carlos III de Madrid

Advisor(s):
Lorena González Manzano, Ph.D.
José María de Fuentes García-Romero de Tejada, Ph.D.

Tutor:
Lorena González Manzano, Ph.D.

April 2023

A Loki, por aguantarme todos los días. Siempre te querré.

A Juan, Eduardo y Chema, que han compartido conmigo y aconsejado en este proceso que es hacer la Tesis.

A Javier, Ana, Mario, Estephy, César, Iñaki y Fer que han sido pacientes conmigo y nunca me han reprochado no poder salir.

A mi familia, por ser un apoyo durante estos años.

Published and submitted content

Published

1. Gimenez-Aguilar, M., De Fuentes, J. M., Gonzalez-Manzano, L., & Arroyo, D. (2021). Achieving cybersecurity in blockchain-based systems: A survey. Future Generation Computer Systems, 124, 91-118. **The content of this article is wholly included in the Thesis on Chapter 3 and it is available at** `https://www.sciencedirect.com/science/article/pii/S0167739X21001576`. The material from this source included in this thesis is not singled out with typographic means and references.

2. Gimenez-Aguilar, M., De Fuentes, J. M., González-Manzano, L., & Camara, C. (2021). Zephyrus: An information hiding mechanism leveraging Ethereum data fields. IEEE Access, 9, 118553-118570. **The content of this article is wholly included in the Thesis in Chapter 4. It is available at** `https://ieeexplore.ieee.org/document/9520396`. The material from this source included in this thesis is not singled out with typographic means and references.

Submitted

1. Gimenez-Aguilar, M., De Fuentes, J. M., González-Manzano L. (2023). Malicious uses of blockchains by malware – from the analysis to Smart-Zephyrus. International Journal of Information Security. **The content of this article is wholly included in the Thesis on Chapter 5 and Chapter 3** . The material from this source included in this thesis is not singled out with typographic means and references.

Abstract

With the increase in connectivity, the popularization of cloud services, and the rise of the Internet of Things (IoT), decentralized approaches for trust management are gaining momentum. Since blockchain technologies provide a distributed ledger, they are receiving massive attention from the research community in different application fields. However, this technology does not provide cybersecurity by itself. Thus, this thesis first aims to provide a comprehensive review of techniques and elements that have been proposed to achieve cybersecurity in blockchain-based systems. The analysis is intended to target area researchers, cybersecurity specialists and blockchain developers. We present a series of lessons learned as well. One of them is the rise of Ethereum as one of the most used technologies.

Furthermore, some intrinsic characteristics of the blockchain, like permanent availability and immutability made it interesting for other ends, namely as covert channels and malicious purposes.

On the one hand, the use of blockchains by malwares has not been characterized yet. Therefore, this thesis also analyzes the current state of the art in this area. One of the lessons learned is that covert communications have received little attention.

On the other hand, although previous works have analyzed the feasibility of covert channels in a particular blockchain technology called Bitcoin, no previous work has explored the use of Ethereum to establish a covert channel considering all transaction fields and smart contracts.

To foster further defence-oriented research, two novel mechanisms are presented on this thesis. First, Zephyrus takes advantage of all Ethereum fields and smart-contract bytecode. Second, Smart-Zephyrus is built to complement Zephyrus by leveraging smart contracts written in Solidity. We also assess the mechanisms feasibility and cost. Our experiments show that Zephyrus, in the best case, can embed 40 Kbits in 0.57 s. for US$ 1.64, and retrieve them in 2.8 s. Smart-Zephyrus, however, is able to hide a 4 Kb secret in 41 s. While being expensive (around US$

1.82 per bit), the provided stealthiness might be worth the price for attackers. Furthermore, these two mechanisms can be combined to increase capacity and reduce costs.

Keywords: Blockchain, Ethereum, Steganography, Cybersecurity, Malware, Covert channels, Smart-contracts

Resumen

Debido al aumento de la conectividad, la popularización de los servicios en la nube y el auge del Internet de las cosas (IoT), los enfoques descentralizados para la gestión de la confianza están cobrando impulso. Dado que las tecnologías de cadena de bloques (blockchain) proporcionan un archivo distribuido, están recibiendo una atención masiva por parte de la comunidad investigadora en diferentes campos de aplicación. Sin embargo, esta tecnología no proporciona ciberseguridad por sí misma. Por lo tanto, esta tesis tiene como primer objetivo proporcionar una revisión exhaustiva de las técnicas y elementos que se han propuesto para lograr la ciberseguridad en los sistemas basados en blockchain. Este análisis está dirigido a investigadores del área, especialistas en ciberseguridad y desarrolladores de blockchain. A su vez, se presentan una serie de lecciones aprendidas, siendo una de ellas el auge de Ethereum como una de las tecnologías más utilizadas.

Asimismo, algunas características intrínsecas de la blockchain, como la disponibilidad permanente y la inmutabilidad, la hacen interesante para otros fines, concretamente como canal encubierto y con fines maliciosos.

Por una parte, aún no se ha caracterizado el uso de la blockchain por parte de malwares. Por ello, esta tesis también analiza el actual estado del arte en este ámbito. Una de las lecciones aprendidas al analizar los datos es que las comunicaciones encubiertas han recibido poca atención.

Por otro lado, aunque trabajos anteriores han analizado la viabilidad de los canales encubiertos en una tecnología blockchain concreta llamada Bitcoin, ningún trabajo anterior ha explorado el uso de Ethereum para establecer un canal encubierto considerando todos los campos de transacción y contratos inteligentes.

Con el objetivo de fomentar una mayor investigación orientada a la defensa, en esta tesis se presentan dos mecanismos novedosos. En primer lugar, Zephyrus aprovecha todos los campos de Ethereum y el bytecode de los contratos inteligentes. En segundo lugar, Smart-Zephyrus complementa Zephyrus aprovechando los con-

tratos inteligentes escritos en Solidity. Se evalúa, también, la viabilidad y el coste de ambos mecanismos. Los resultados muestran que Zephyrus, en el mejor de los casos, puede ocultar 40 Kbits en 0,57 s. por 1,64 US$, y recuperarlos en 2,8 s. Smart-Zephyrus, por su parte, es capaz de ocultar un secreto de 4 Kb en 41 s. Si bien es cierto que es caro (alrededor de 1,82 dólares por bit), el sigilo proporcionado podría valer la pena para los atacantes. Además, estos dos mecanismos pueden combinarse para aumentar la capacidad y reducir los costes.

Palabras clave: Cadena de bloques, Ethereum, Esteganografía, Ciberseguridad, Malware, Canales encubiertos, Contratos inteligentes

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

This Chapter introduces the context of the thesis, the statement of the problem, the main objectives of the thesis, the contributions achieved and the document organization.

1.1 Context

Nowadays, Internet connectivity is being offered in an increasing amount of places. The widespread use of cloud technologies and connected devices has enabled new forms of data and computation outsourcing, along with the irruption of the so called Internet of Things (IoT). Besides the explosion of IoT devices, network technologies are also evolving very fast. Speed and reliability of networks are continuously improving, thus enabling a permanent connectivity of devices as it happens with the recently developed 5G technology (1).

Both the increase of devices and the improvement of technologies have motivated the raise of decentralization approaches for many scenarios. Thus, instead of relying on a single device or entity, it is common to provide services and resources as a result of the collaboration among multiple communication nodes.

In what comes to cybersecurity of computer systems, one of the main concerns is where to put the trust. If a given application needs to manage sensitive information, it is usually solved by protecting a given node or device. Although this solution is cost-effective (only one resource needs to be protected), it also involves the single point of failure risk (2). Thus, if the node is compromised, the whole cybersecurity

is threatened. Thanks to decentralization, the likelihood of success of a cyber-attack can be reduced. In this vein, blockchain technologies offer a decentralized storage in which data can be securely stored without the need of any single trusted party (3). Information is managed through a distributed ledger in which data is consecutively appended to existing records. Remarkably, nodes maintaining the ledger do not need to be mutually trusted, which promotes its application in trustless scenarios (4).

Blockchain is a well-known technology that allows the execution of transactions ensuring their integrity. It can be described as "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way" (5). It has been a growing technology since Satoshi Nakamoto proposed it in 2008 (6). Blockchain technologies have already been applied in many different scenarios. Cryptocurrencies leverage blockchains, in such a way that any economic transaction is appended as a new record. Every node connected to the network is able to verify that a given amount of funds have been transferred, thus preventing the overspending problem (i.e., that the payer uses the same coin in two or more payments) (7). Bitcoin (6) and Ethereum are two of the main cryptocurrencies nowadays (8) Thus, blockchains open up a vast array of novel applications and production models (e.g., social manufacturing (9), health (10) or energy (11)).

In the last years we have witnessed a myriad of contributions focused on achieving cybersecurity when blockchain technologies are at stake. Cryptographic experts such as Bruce Schneier have already identified an unjustified hype surrounding this technology, by pointing out its limitations – *"A blockchain probably doesn't solve the security problems you think it solves. The security problems it solves are probably not the ones you have."* (12). Therefore, although blockchains provide with *some* cybersecurity guarantees, they cannot be regarded as a holistic solution. Indeed, the extensive adoption of blockchain technologies and, in special, cryptocurrencies makes them interesting for other purposes besides the provision of cybersecurity.

Blockchain provides a set of intrinsic features: availability, integrity and anonymity. On the one hand, availability and integrity make them interesting to build covert channels (that is, a way to secretly send information) over a publicly available medium (that is, the list of transactions of the cryptocurrency). Thus, any data inserted into the ledger will remain unaltered and readable by any party virtually anytime.

Since a great variety of use cases can be devised, three situations in which such a covert communication is interesting, although with different degrees of immediacy are provided. First, in the *panic button case* a threatened individual is willing to leave some secret material (e.g., account keys) to be released in case of emergency and thus, without immediacy in mind. Second, in a *sabotage case* a malicious insider aims to immediately exfiltrate sensitive data without being detected. Third, in a *censorship case* an individual is willing to share information in a controlled and censored environment.

To hide a secret in such a setting, steganography is the art of concealing messages within a non-secret piece of data called cover (13). It is a branch of cryptography used when discretion is a priority. Steganographic approaches can be generally divided into implicit and explicit ones. Implicit techniques rely upon the way in which the system is used (14). For example, if the sending time is odd or even, it can be understood by the receiver as 1 or 0, respectively. On the other hand, explicit approaches base on modifying the cover to embed the secret (15). In this thesis we focus on this latter, information-based approaches.

Indeed, several works have already applied steganography and covert channels in Bitcoin (16; 17; 18; 19). For example, it has been applied to counter censorship (20). However, few efforts have been devoted to information hiding in Ethereum. To the best of the author's knowledge, only (21) leveraged a single data field, and (22) proposed the use of a Ethereum-related protocol. Thus, no previous work has focused on leveraging Ethereum for covert communications considering all its data

fields. Most of them have never been used for steganographic purposes and this thesis addresses this issue for the first time.

On the other hand, due to their anonymity and availability, blockchains are also used for malicious purposes (23). For example, Bitcoin is widely used in the darknet to pay for forbidden products (24). Similarly, malware developers and cybercriminals have already used this technology as well (25; 26; 27; 28). As a massive phenomenon, the relationship between malwares and blockchains is worth being studied.

Furthermore, as malwares are typically hard to detect, the combination of them with covert channels could be very useful (thus dangerous). Thus, it should foster further defence-based research in this direction.

1.2 Motivation

The main purpose of this thesis is to characterize the use of blockchain in relation with cybersecurity as well to develop mechanisms that make use of this technology in regard with this field of study.

The hype surrounding blockchain has led to a huge amount of contributions. However, even though some of their intrinsic characteristics can make this solution appropriate to provide cybersecurity, this is not a solution for everything. Moreover, some of their characteristics could facilitate their use for malicious puposes.

The previous issues lead to the identification of four specific problems addressed in this thesis:

P1. Lack of a systematic review of approaches in regard to cybersecurity and blockchain

The increasing protagonism of blockchain technologies is attracting attention from both industry and academia. However, the frenetic pace of evolution can make this technology seems the Swiss knife for every new approach, thus leading to improper uses. Moreover, many related efforts can be carried out in parallel,

resulting in overlapping approaches. Both issues can threat the widespread adoption of this technology.

P2. Lack of studies about how malwares take advantage of blockchain intrinsic characteristics to enhance their malicious capabilities

Malwares have infected at least one third of the computers worldwide (29) and it is predicted to cost the world $6 trillion annually (30). Previous works have studied this relationship. For example, (31) analyzed the effect of cryptocurrencies on ransomware. Similarly, (32) provided a systematic analysis of blockchain-based botnets. However, most of the previous studies focus on a single type of malware (e.g botnets or ransomware) or a single type of blockchain technology (e.g. Ethereum or Bitcoin). As a massive phenomenon, the relationship between malwares and blockchains is worth being studied.

P3. No previous work makes use of all Ethereum fields to build a covert channel

Ethereum is gaining momentum thanks to its support to distributed apps by means of smart contracts. They are pieces of software that can be executed without human intervention. Since they are stored in the blockchain, they remain permanently (33). Moreover, the underlying data structures that are also stored in the blockchain are heavily different to those present in Bitcoin. Thus, the unique features of Ethereum motivate the need for proposing a tailored mechanism for covert communications. Since this cryptocurrency holds significant differences against Bitcoin, it is necessary to characterize its suitability for this purpose.

P4. Lack of a steganographic tool that uses a smart-contract language

As stated in P2, Ethereum is gaining popularity thanks to the support of smart contracts. This smart contracts are written in a high level language, being Solidity the most relevant, that is later compiled and stored on chain. The written contract can be later verified and stored in a blockchain explorer like Etherscan (34). Smart contracts verification provides additional transparency and increases the level of

trust in the ecosystem (35). Therefore, it makes sense to take advantage of this feature to build a covert channel, either for normal communication or to improve malware resilience.

1.3 Objectives and contributions

The general goal of this thesis is to provide a systematic review of how blockchain is used with cybersecurity purposes and also from the perspective of an attacker (via malware), as well as to provide tools that take advantage of the intrinsic characteristics of this technology for different use cases.

In our opinion there was a need to address the previous research topics, which have been reflected in the objectives of this thesis:

O1. Provide a **systematic review of how blockchains are used for cybersecurity purposes**. It should set the boundaries on suitable uses, current state of the art and open research and development directions.

O2. Study and categorize the **different ways in which blockchains have been used by malwares**. It should explain the current state of the art, how malwares take advantage of different blockchain characteristics and point out open research areas.

O3. Develop a **steganographic tool** that **leverages all Ethereum data fields**. The stealthiness of the secret should be preserved and the mechanism should be assessed in terms of cost efficiency.

O4. Develop a **steganographic tool** that **uses Solidity as a medium of hiding information**. As in O2, the stealthiness of the information should be maintained and the mechanism efficiency should be evaluated.

The achievement of these objectives has led to the next three contributions:

C1. Achieving cybersecurity in blockchain-based systems and malicious uses of blockchains by malwares (see Chapter 3). It aims to provide with a comprehensive review of techniques and elements that have been proposed

to achieve cybersecurity as well as a study and categorization of the different ways in which blockchain has been used by malwares. For the first part, an analysis of 272 papers from 2013 to 2020 and 128 industrial applications has been carried out. A taxonomy of elements involved in the proposed analysis namely their cybersecurity properties and related techniques, application areas, technologies and the justified use of blockchain is provided. For the second part, a study of 104 proposals has been carried out. They are analyzed in terms of type of malware, technologies, desired properties, elements, communication features, purpose and cost for the attacker.

C2. Zephyrus: An information hiding mechanism leveraging Ethereum data fields (see Chapter 4). Zephyrus is a steganographic tool for Ethereum. While most tools and previous examples of steganography on blockchain were based in Bitcoin, this technology explores Ethereum as a way to embed secret messages. It allows inserting secret messages in 8 different fields of Ethereum, like the receiver address or the value field. Regarding contracts, it hides information in the swarm hash or bytecode, as well as in the function arguments and constructor. For it design, a large amount of real-world Ethereum blockchain data was analyzed to ensure the stealthiness of the secret. Indeed, the mechanism is assessed in terms of capacity, stealthiness and cost. Furthermore, an open-source proof of concept is released to foster further research. It is also have been used to assess the time taken for embedding and revealing a secret in a real-world Ethereum network.

C3. A steganographic tool leveraging Solidity code: Smart-Zephyrus (see Chapter 5). As mentioned on the malware analysis carried out in C1, it has been identified that covert communications have not received much attention. To foster further defence-oriented research, a novel mechanism (dubbed Smart-Zephyrus) is built leveraging smart contracts written in Solidity. It extends the mechanism proposed in C2. As well as in C2, the mechanism has been designed based on real-world data. It is, also assessed in terms of capacity, stealthiness and cost. As in the previous case, an open-source proof of concept has been developed and used to

assess real-time consumption of the mechanism on the Ethereum network.

The relationship between the problems detected, the research objectives and the contributions achieved is shown in Table 1.1.

Problem	**Objective**	**Contribution**
P1. Lack of a systematic review of approaches in regard to cybersecurity and blockchain	O1. Provide a systematic review of how blockchain is used with cybersecurity purposes	C1. Achieving cybersecurity in blockchain-based systems and malicious uses of blockchains by malware
P2. Lack of studies about how malware takes advantage of blockchain intrinsic characteristics to enhance their malicious capabilities	O2. Study and categorize the different ways in which blockchains have been used by malwares	
P3. No previous work makes use of all Ethereum fields as a covert channel	O3. Develop a steganographic tool that leverages all Ethereum data fields.	C2. Zephyrus: An information hiding mechanism leveraging Ethereum data fields
P4. Lack of a steganographic tool that uses a smart-contract language	O4. Develop a steganographic tool that use Solidity as a medium of hiding information	C3. A steganographic tool leveraging Solidity code: Smart-Zephyrus

Table 1.1: Relationship between problems, objectives and contributions

We find that these issues are a step towards the categorization of the use the blockchain to provide cybersecurity, as well as a better understanding of different ways the blockchain (specifically Ethereum, but not limited to that one) can be leveraged with other purposes, like exfiltrating information or by an attacker to provide malware with certain characteristics.

All contributions of this thesis are summarized in Figure 1.1. We can distinguish different entities that interact with the blockchain according with their intentions. First, legit users (in blue) that utilize the blockchain for legal purposes. We can include in this group service providers and consumers as well as those users who want exfiltrate information in a "panic button"-like scenario (recall Section 1.1).

Then, we have the second group, the neutral users (in gray), in which whether their intentions are good or bad are based more on morality. An example of this case would be those that want to use the blockchain as an "anti-censorship" mechanism (Section 1.1). After that, there are the malicious users (in red) who leverage the blockchain to cause damage, often in an illegal way. This is the case for attackers and malware programs that interact with the chain. Finally we have the victims that may or may not interact with the blockchain, for example to make payments to an attacker.

On the one hand, those legit users that interacts with the blockchain in order to request, provide or consume services often use this technology because it provides some kind of cybersecurity property, either by using blockchain intrinsic characteristics, like immutability and availability, by their own or combined with other mechanisms, like homomorphic encryption (studied in C1).

On the other hand, there are the data exfiltrators and consumers. Data exfiltration occurs when data is extracted from a system without authorization. The panic button case is included here because even though the user could have the authority to share that information, they want to remain hidden. We also include the anti-censorship one in here because the purpose is to avoid an interception by an authority. Those that consume the data for both previous cases can be of any type of user. Moreover, the situation in which an attacker sends or receives information from and to a malware is also included here. All this cases benefit from a way to hide in the system. The first case only wants to make the information public after certain event. The second one wants to share information with a selected group of individuals without being detected and intercepted. The third one, could use steganography in order to difficult the detection of the malware, for example by hiding the commands sent to a botnet behind legit transactions (C1, C2 and C3).

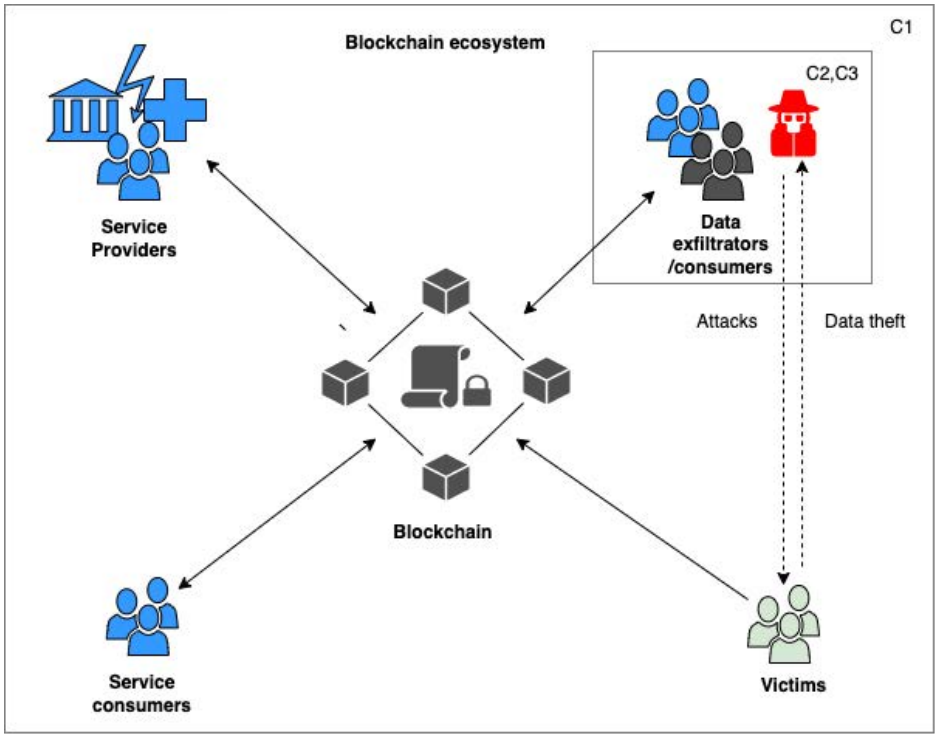

Figure 1.1: Context and scope of this thesis

1.4 Document organization

This thesis is composed by several chapters distributed along five parts:

Part I. Introduction. This part introduces the whole document, and contains the present Chapter.

Chapter 1. Introduction. This is the present Chapter, and contains the thesis context, the statement of the problem, the research objectives and the main contributions achieved.

Part II. State of the art. This part analyses the state of the art that is closely related to this thesis. The reviewed topics have been organised into one chapter.

Part III. Proposal. This part includes the proposal elaborated to fulfil the research objectives established above. Each of the four contributions is presented in a separate chapter.

Chapter 3. Achieving cybersecurity in blockchain-based systems and

malicious uses of blockchains by malware. In this Chapter we present a comprehensive review of all techniques and elements that have been proposed to achieve cybersecurity as well as a study and categorization of the different ways in which blockchain has been used by malwares. For the first Section, we first present the research methodology used to retrieve and select the pertinent papers. We then realize an analysis an extract a taxonomy of all the elements extracted. After that, we describe the different lessons learned based on the data examined. Furthermore, we provide a summary of the previous studies and compare them to the current proposal. For the second Section, we also first describe the research methodology used to retrieve and select the studied works and categorize them according with how they use the blockchain. We summarize our findings in different lessons learned and expose some research gaps. Lastly we compare the current work with previous ones.

Chapter 4. Zephyrus: An information hiding mechanism leveraging Ethereum data fields. In this Chapter we propose Zephyrus, a steganographic tool for Ethereum. First we describe the model on which the system will be based. Then, we present the result of a preliminary study of Ethereum data. Based on that results, we describe the mechanism. After that, its evaluation is described. Last but not least, a summary of all related works regarding steganography and blockhain is introduced and the proposed mechanism is compared with the existing works.

Chapter 5. A steganographic tool leveraging Solidity code: Smart-Zephyrus. In this Chapter Smart-Zephyrus, a steganographic tool that leverages Solidity code to embed information. First, we present an overview of the proposal . Then, we expose the motivation for this contribution. We carry out a preliminary contract study. After that, we justify the design decisions for the mechanism. The assessment of the proposed solution is presented afterwards. Lastly, we summarize the related work and describe how our mechanism compares with it.

Part IV. Evaluation and Conclusions. The evaluation of the thesis contributions and the conclusions are presented in this part, which is formed by one chapters.

Chapter 6. Conclusions and Future Work. In this Chapter, the conclusions of this thesis are provided. A critical discussion of the work performed in this thesis is presented. In addition, future research directions that may be derived from this thesis are outlined.

Part V. Bibliography and Appendices. This part includes the bibliography in use and a set of appendices that complement the main content.

Bibliography. The bibliography contains the list of references to other research papers, technical documents and standards used in the thesis.

Part II

Background

Chapter 2

Background

This Chapter introduces the main concepts of cybersecurity, blockchains, covert channels and malware. In particular, the foundations of blockchain technologies are presented in Section 2.1. Afterwards, in order to simplify the presentation of concepts in the analysis of the literature, a unified model of blockchain technologies is presented in Section 2.2. Then, the main goals of cybersecurity are described in Section 2.3. The criteria used to justify the need for blockchains is presented in Section 2.4. Afterwards, the main notions of Ethereum are described in Section 2.5. Then, the basics on steganographic systems and covert channel are introduced in Section 2.6. Finally, different types of malware used in blockchains are explained in Section 2.7.

2.1 Blockchain overview

Blockchain technologies enable having a distributed ledger in which data is appended in the form of consecutive blocks(6). One important matter is that there is no need for a single, centralized trusted party – trust is distributed among all nodes. Therefore, in order to add data to the ledger, a consensus is usually needed to be reached among all (or a qualified portion of) involved nodes (36).

In order to provide with a general overview, blockchains can be classified depending on their nature and their underlying technology. There are a number of elements that are present on the blockchain. Furthermore, blockchain also provides, by default, a set of properties. Each of these issues is introduced below.

2.1.1 Nature

There are two factors that determine the nature of a blockchain, namely their access control and their data validation policy. Concerning access control, they can either be public, where everyone can join freely, or private, where only selected members can take part. With respect to the validation policy, it is related to the way in which the nodes allowed to update the ledger (called *miners* in the case of proof-of-work based blockchains, and validators from a general point of view) are chosen. Thus, blockchains in which any node can be a validator are called "permissionless", whereas those where only a specified set of users can take this role, are referred to as "permissioned".

2.1.2 Technologies

There are three blockchain technologies that are widely used. Bitcoin was the first cryptocurrency and also the first technology to build a blockchain. Proposed by Satoshi Nakamoto in 2008, it allows two parties to send transactions between each other without the involvement of a third one. It has a non-Turing-complete scripting language which supports different advanced features, such as the use of timelocks to prevent the execution of a given action before a deadline. Considering previous classification parameters, Bitcoin is a public permissionless technology.

On the other hand, Ethereum was released in 2015 and allows the execution of smart contracts. These are software artifacts that are executed by Ethereum nodes through its Ethereum virtual machine. They are written in specific languages such as Solidity or Serpent and then compiled into bytecode. Ethereum's main network is public (37; 38). A more detailed explanation of this technology is provided later on in Section 2.5.

Other technologies and cryptocurrencies have been developed, for example, Monero, which uses a public ledger with privacy-enhancing technologies (ring signatures and stealth addresses) that obfuscate transactions to achieve anonymity

and fungibility, that is the ability of a good or asset to be interchanged with other individual goods or assets of the same type (39). Other technology is IOTA, a distributed ledger and cryptocurrency aimed to record and execute transactions between devices in the Internet of Things (IoT) ecosystem. One of IOTa's innovation is Tangle, a system of nodes used for confirming transactions based on a Proof-of-Work consensus (40; 41). Because of that, it could be argue whether it is actually a blockchain, as there is no chain of blocks. However, given that some authors consider it a blockchain ((42; 43; 44), the same approach is followed in this thesis. On the other hand, NKN incorporates a blockchain layer on top of existing TCP/IP in order to allow both individuals and large ISPs to better optimize data usage and reduce costs. NKN can also be used to develop decentralized web 3.0 applications (45). Dash is an alternative currency that was forked from the Bitcoin protocol (46). It includes other improvements such as PrivateSend, for increasing fungibility, and InstantSend, which allows instant transaction confirmation without a centralized authority (47). Other cryptocurrencies, just to mention a few, are BitcoinCash (Bitcoin fork), Emercoin and Litecoin.

Finally, the Hyperledger Project consists of a community of software developers building blockchain frameworks and platforms. It was announced at the end of 2015 by the Linux Foundation. There are different blockchain technologies included in this project, like Hyperledger Fabric or Hyperledger Iroha (48). Among them, probably the most used is Hyperledger Fabric (49), which is a blockchain framework intended as a foundation for developing applications or solutions with a modular architecture. In this case, it is typically oriented towards private permissioned networks and it allows different consensus algorithms. Smart contracts written in Go, node.js or Java, can be executed and are called chaincodes (50).

2.1.3 Elements

Blockhain is a distributed database shared among nodes. The data is usually recorded in the form of transactions. When a transaction is recorded in the blockchain, details of the transaction are recorded, verified and distributed across all nodes (51). Recorded data can vary among technologies, though some data is common in all of them. First, there is the sender address, who/which originates the transaction. Depending of the technology, it can be one or more receiver addresses, those who/ which are intended to receive the transaction. If the technology is a cryptocurrency a field that represents the transferred amount or value is usually included. Some technologies also allow the addition of arbitrary data in the blockchain. This is the case of the data field in Ethereum (52), OP_RETURN transaction type in Bitcoin (53) or payment id in Monero (54).

Moreover, some technologies also allow the execution of programs in the blockchain, for example scripts in Bitcoin or smart contracts in Ethereum, as introduced before.

2.1.4 Properties

As a distributed decentralized ledger, blockchain provides some properties by default:

- Immutability. It is the ability of the ledger to remain unchanged, unaltered and inedible (55). This property provides some benefits, like complete data integrity and auditability. However, this characteristic has some limitations, namely the 51% attack. In the 51% attack an attacker take control of the majority of the nodes (51%), allowing him to change the transaction data (56).

- Decentralization. Control and decision-making capacity is taken from a centralized entity to a distributed network. These networks aim to reduce the

level of trust among participants. Some benefits of decentralization include the ability to provide an untrusted environment, reduce points of weaknesses and optimize resource distribution (57).

- Data availability. Data can be accessed and used in a timeliness and reliable way(58). Read availability (consulting data) in blockchains is typically high because of its immutability and decentralization (59).

- Pseudoanonymity. Anonymity [1] can be defined as the situation in which someone's identity is not given or known. Although some blockchain technologies are anonymous (e.g. Monero), most of them are pseudoanonymous. This is due to the fact that the user has a public address that could be traced back to an exchange account or IP address via network analysis thus being possible the revealing of the user's real identity (60).

2.2 Blockchain model

There are several entities or elements at stake when it comes to maintaining and using a blockchain (Figure 2.1). On the one hand, there are a set of nodes (referred to as Blockchain Nodes, BCN) that are in charge of keeping the blockchain information itself, which could be either in clear or encrypted. Then, they cooperate to update blockchain data based on a consensus algorithm. Consensus is typically reached among a subset of BCNs. Indeed, data to be included in the blockchain is proposed by one of the BCNs. Such node is either chosen in a deterministic way or randomly validated based on some established mechanism. Therefore, the so called miners (which are not present in all types of blockchains), are a subset of BCNs.

Apart from BCNs, there is always another entity that comes into play – Blockchain Users (BCUs). BCUs are willing to insert information into the blockchain. Let us consider a hospital that manages clinical reports through a

[1] `https://dictionary.cambridge.org/es/diccionario/ingles/anonymity`

blockchain. Whereas BCNs will be nodes from the hospital taking care of the data, BCUs can be tablet devices held by doctors which send updated health results to be stored in the blockchain. In another setting, two BCUs that are cooperating may want to record the status of their transactions. For example, BCU_1 offers a service and BCU_2 wants to pay for it. Both of them will use the blockchain to store offers and payments, respectively.

Blockchain Observers (BCOs), on the other hand, are those users of the blockchain that gain something, by only retrieving the data present in the blockchain or by observing it. They do not contribute to the blockchain by adding data themselves. However, their retrieval might be recorded in the blockchain by means of a transaction or a interaction with a smart contract. For example, in (61) a transaction must be sent to the blockchain for data retrieval in order to check permissions and deliver information accordingly. Following the hospital case, patients (or insurance companies) are BCOs as long as they only want to see health information. This can be also the case of auditors of different systems, third companies buying or analyzing data, users retrieving their own data by using the blockchain as storage when IoTs are involved, etc.

Given that BCNs are, *de facto*, the only nodes having direct access to the blockchain, both BCUs and BCOs should trust them in that the information purportedly stored in or retrieved from the blockchain is the one that is being exchanged. However, no BCN is assumed to be trusted – any attempt to alter blockchain information will be mitigated by the underlying consensus protocol or access control/permission mechanisms in force. To further prevent mistrust, BCNs count on incentives such as rewards per successful transaction inserted in the ledger.

Moreover, as storing information in the blockchain is usually expensive and conveys scalability concerns (62) some additional storage (AS) in the form of cloud storage or Distributed Hash Tables (DHT), for example, could be used for this purpose (63). In this case, blockchain vouches for data integrity and auditability

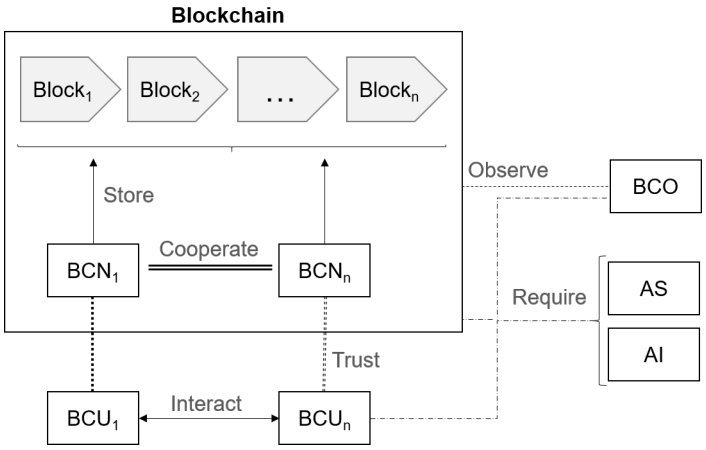

Figure 2.1: Blockchain entities model

by performing data anchoring, i.e., by storing in the ledger pointers to the data and the corresponding time stamping (64).

In addition, in some blockchain systems there are other software and hardware elements to provide all required capabilities. This is the case of extra hardware or off-chain software components. These elements are called Additional Infrastructure (AI).

2.3 Cybersecurity goals

According to the US National Institute of Standards and Technology (NIST), cybersecurity is defined as the "prevention of damage to, protection of, and restoration of computers, electronic communications systems, electronic communications services, wire communication, and electronic communication, including information contained therein, to ensure its availability, integrity, authentication, confidentiality, and non-repudiation"(65). Therefore, cybersecurity is indeed a generic name that refers to the aforementioned five security dimensions, that will be defined in the following.

Authentication refers to "the process of establishing confidence in the identity of users or information systems"(66). As opposed to integrity, confidentiality and

non-repudiation, this feature is related to system stakeholders and not to the data at stake.

With respect to confidentiality, it refers to the fact that "sensitive information is not disclosed to unauthorized entities"(66). This matter is not provided by blockchain technologies by design, as soon as they have been designed to provide auditability, that is, enabling any party to verify the data contained therein.

Concerning non-repudiation, it corresponds to the "protection against an individual falsely denying having performed a particular action" (66). Due to the large amount of potential actions that can be devised into an IT system, variants of this feature have appeared such as sender or receiver non-repudiation. In particular, sender non-repudiation is also essential in some blockchain use cases such as cryptocurrencies to avoid the double-spending problem (7). Indeed, blockchains already provide with sender non-repudiation mechanisms via digital signatures.

A special situation happens with the remaining pair of cybersecurity features. On the one hand, availability is defined as "ensuring timely and reliable access to and use of information" (65). Regarding availability, it is necessary to recall Brewer's CAP theorem – it is very complex to achieve consistency, availability and partitioning as a whole. Thus, for the sake of simplicity we assume that availability is provided to some extent when blockchains come into play. On the other hand, integrity is "a property whereby data has not been altered in an unauthorized manner since it was created, transmitted or stored" (67). In this thesis, each cybersecurity property will be addressed separately, since each blockchain-based proposal may provide some of them. However, in the case of integrity, it is an intrinsic property of the blockchain technology itself. Therefore, the mere use of this technology provides integrity.

2.4 Actual need for blockchains

Blockchain is an emerging technology and its use has been steadily increasing over the years. However, sometimes its use can be unjustified. According to Greenspan (68), there are eight criteria that must be met in order to ensure that blockchains are suitable for a given use case:

- Need for a shared database, including a set of transactions forming a ledger.
- Existence of multiple writers willing to insert data to the said database.
- Inter-writer mistrust, so each writer is not willing to allow any peer to edit its entries.
- Disintermediation, so writers are not willing to give a third party full control over the database.
- Transaction interaction, so there is a certain undeniable link between transactions.
- Transaction verifiability, so each transaction can be accepted under a set of (automatically verifiable) requirements.
- Existence of validators, that is, nodes that verify transactions.
- Storing value, so each entry represents something that has real-world value.

2.5 Ethereum

As mentioned in Section 2.1.2 Ethereum is a blockchain technology capable of running smart-contracts . They are pieces of code executed by the nodes maintaining the network. In this way, censorship or code changes by third parties are avoided, thus enabling building distributed applications (69). Apart from smart contracts, Ethereum enables sending funds among parties as any other cryptocurrency. Every

interaction with this blockchain is carried out through transactions. To avoid false transactions, a distributed consensus algorithm is run by Ethereum nodes. This process involves a computational task called mining. In Ethereum, before the mining involved a trial-and-error process, called Proof-of-Work(PoW), until the result of a cryptographic process (in particular, a hash function) meets a given condition (70). Since the 6th of September of 2022 PoW consensus has been substituted by Proof-of-Stake (PoS). In it, miners stake ether on the blockchain. The winner is selected by the network based on this amount and they need to stake (lock) the native cryptocurrency of the blockchain. The network then selects a winner based on the amount of crypto staked who will validate the block (71).

In order to run smart contracts and maintain the blockchain, Ethereum comes with a decentralized virtual machine called EVM. It uses a Turing-complete language to be able to create sophisticated smart contracts (72).

In the following, the different data items of transactions and smart contracts are introduced.

2.5.1 Ethereum transactions

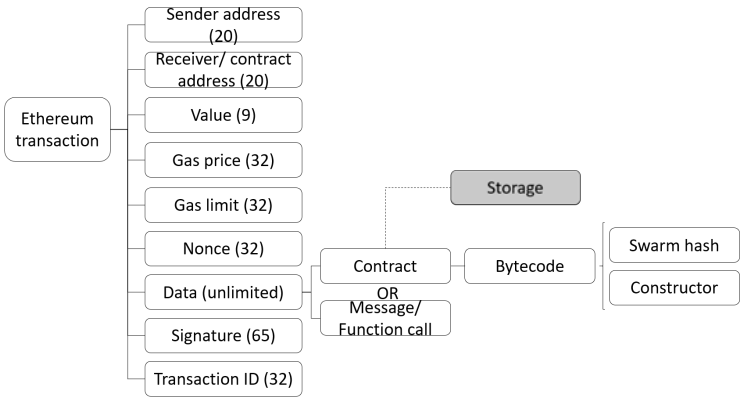

Figure 2.2: Ethereum transaction fields with their size in bytes. White boxes are stored immutably in the blockchain, while the greyed one is not.

A transaction in Ethereum contains several fields of different size (Fig. 2.2). Each transaction is sent by a user, identified by a *Sender address* that results from

the 20 last bytes of the hash of the user's public key. The receiver is also identified by an address. Note that it can be another user (using a *Receiver address*) or a smart contract. In the latter case, the *Contract address* comes from the hash of the sender address and the amount of transactions sent by that account. A particular case happens when the receiver address is null, that is when a contract is deployed.

On the other hand, the *Value* field includes the amount of funds at stake. Apart from the value itself, two fields (*Gas price* and *Gas limit*) express the costs that the sender assumes to include this transaction in the blockchain. In particular, *Gas price* defines the cost per operation and *Gas limit* sets the maximum incurred cost (73). Such a cost depends on the information included in the data field. This can be of three types, namely a *text message*, a *contract* to be deployed (as explained in Section 2.5.2) or a *function call* to an existing contract. In the latter case, an encoded representation (called ABI (74)) of the called function name and arguments is included.

Last but not least, each transaction is identified by a serial number (*Nonce* field) and an *Identifier* which is the hash of the previous fields. The legitimacy of the transaction is shown by the sender's digital signature. The *signature* is formed by three values, namely V, r and s, which are the result of applying the ECDSA algorithm (75).

2.5.2 Smart contracts

A smart contract is a piece of code formed by functions to be executed by any Ethereum node through its EVM. It can be written in different languages, for example, Solidity, Serpent, LLL or Vyper. Those are high-level languages that contain functions, arguments and control flow instructions and operators. Since Solidity is the most widespread one, it is the one considered in this proposal. Solidity is similar to Java, as most elements of a Solidity contract are similar to those in Java language; however, Solidity is specially designed for developing smart contracts

and, thus, it contains elements that are characteristic of the language (76). These elements include functions, which have input arguments in different number or type (77; 78); modifiers, that restrict the behaviour of certain functions; variables; and events, that allow publishing information about something in the chain.

When Solidity code is compiled, it is transformed into a hexadecimal string known as *bytecode*. It is formed by opcodes or low-level, human-readable instructions the EVM can execute. For example, JUMP instruction indicates the EVM to go to a particular part of the program and execute it. The order and placement of instructions is essential because they set the execution flow. The cost of each instruction depends on its type (72).

For the interest of this proposal, it is relevant to understand how the bytecode ends its execution. Before the last instruction (STOP or INVALID), usually a JUMP is placed, which makes the code continue its execution from an address pointed out in that instruction. That address must contain a JUMPDEST instruction – otherwise the EVM throws an error. We will refer to this region as the *JUMP-JUMPDEST block*.

Apart from *bytecode*, metadata describing the contract at stake is produced by the compiler when Solidity is used. This information is intended to be published in an external repository to help in verifying the contract integrity. Thus, the *Swarm hash* field (which is the hash of the contract, including its file name) is included at the end of bytecode. It serves as a pointer to find the contract in a content-addressed storage outside the blockchain (79).

Smart contracts may optionally store their current state. Such *storage* can be initialized with a special function called *constructor*. Stored values can be updated by calling to the appropriate contract functions. Since storage leverages Ethereum nodes' memory space, these operations involve additional costs.

Smart contracts work as decentralized applications. Example of smart contracts applications include trading, investing, gaming or voting. Indeed, there are smart

contracts called tokens and, according to their behavior, represent a variety of transferable and countable goods such as digital and physical assets, shares, votes, memberships, or loyalty points. Any participant of the chain can create smart contracts and then, they can develop, define and distribute their own named tokens (80). The most widely used token is, by far, Ethereum's ERC20 (81), followed by others like ERC721, or ERC165.

Such smart contracts can be found in websites like Openzeppelin, which is an open-source platform for building secure distributed applications (called dApps). The framework provides the required tools to create and automate dApps. In addition, companies of any size can refer to OpenZeppelin's audit services to find the best practices in the industry. Solidity's programming language is used to develop modular and reusable contracts within its library, including ERC-20-related OpenZeppelin contracts. In addition, the community has tested and reviewed contracts which, according to OpenZeppelin, are the most popular in the industry (82).

2.6 Covert channels and steganographic systems

It is known as covert channel: "any communication channel that can be exploited by a process to transfer information in a manner that violates the system security policy "(83). Their main purpose is to protect privacy or to increase security of critical information. However, they can also be used to establish connections that are theoretically prohibited by the security policy (83). A covert channel has different properties:

- Stealthiness: the capacity of avoiding detection, thus data sent or received through a covert channel is not able to be discovered by an undesired party.
- Efficiency: the amount of information that can be sent using the covert channel in relation to the cost or total size of the channel.
- Secret integrity/resilience: the capacity of hidden information's integrity to

remain over changes and time.

Communication channels (cover or not) are commonly limited to a certain capacity, i.e. the amount of information that can be transmitted through the channel. In most cases, the goal is to maximize the amount of shared information (efficiency), though it may impact on stealthiness and thus, a balance between both issues should be considered. On the other hand, in the case of the blockchain, as its content on chain is immutable, resilience is provided by default.

A typical technique to implement covert communications is steganography[2]. In a steganographic system different elements are involved, see Fig.2.3. A sender and one or more receivers share a key to hide or extract a secret message. This key is essential to provide confidentiality by means of encryption (84). Basically, the sender has an element (e.g. code, image, text, etc.), called cover, in which a secret is hidden. The steganographic system receives the cover, the secret and the key and outputs a steganographic object. This object is sent to the receiver through the channel. In the destination, the receiver uses the steganographic object and the shared key to get the hidden message. Between both parties, an attacker or warden might be placed. It can be passive, thus eavesdropping the channel, or active, thus being able to tamper with the transmitted data (85).

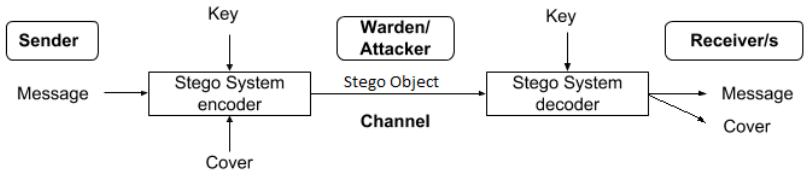

Figure 2.3: Entities in a steganographic system.

In order to transmit information, different operations have to be carried out for hiding (also called embedding) and retrieving (usually called revealing) the secret at stake. In information-based steganography, one of the most well-known examples is the Least Significant Bit (LSB) (86). In LSB, the cover's rightmost bit(s) are

[2] https://www.merriam-webster.com/dictionary/stealthy#:~:text=1%20%3A%20slow%2C%20deliberate%2C%20and,radar%20return%20a%20stealthy%20airplane

replaced by the secret. This technique should be applied while keeping the cover appearance to avoid raising suspicions.

2.7 Blockchain-related malware types

It is known as malware any software designed to cause intentional damage to a computer, server, client or network(87). They can be classified in different types, depending on the goal of the attacker and the way it acts. Despite the amount of malware types, this section introduces those which are known to have been used in blockchain.

In recent times, one of the most popular types of malware is ransomware. It prevents users from accessing their system or files and demands a payment as ransom in order to regain access (88). Modern ransomware families, also known as cryptoransomware, encrypt certain file types on the target infected system in order to force victims to pay the ransom via an online payment method in order to get the decryption key. They usually take advantage of the anonymity provided by cryptocurrencies in terms of payments and the ransom quantity may vary depending of variant of the malware and the price of the different coins (89).

With the ever growing interest in decentralization, another common types of malware that exploit this characteristic are botnets. A botnet is a network of infected computers that can be remotely controlled and forced to perform different actions or attacks without the consent of the device owners (90). Orders are given by a command and control (C& C) server, that is a computer controlled by an attacker to send commands to compromised systems, called bots, and receive stolen data from a target network. A bot is "a program that performs automated, repetitive and predefined tasks". They can carry out multiple and assorted tasks, for instance, hacking, spying or interrupting a service (91). A botmaster is a person who operates the C& C server for remote process execution (92). Note that a botnet is not considered a malware by itself but as they can be used for malware propagation

and because of their use for malicious purposes, herein we considered botnets within malware types category, in line with (93) (94).

Part III

Proposal

Chapter 3

Achieving cybersecurity in blockchain-based systems and malicious uses of blockchains by malware

---

3.1 Summary of the chapter

In this Chapter we present a comprehensive review of all techniques and elements that have been proposed to achieve cybersecurity in blockchain in acadamic proposals as well as a study and categorization of the different ways in which this technology has been leveraged by malware.

This Chapter is divided therefore in two sections. In Section 3.2, the review of the state of the art of proposal regarding cybersecurity is defined. In Section 3.3, the current state of the art on how malware utilizes blockchain is explained.

3.2 Achieving cybersecurity in blockchain-based systems

In this Section a comprehensive review of all techniques and elements that have been proposed to achieve cybersecurity in blockchain is . The research methodology used to retrieve and select the pertinent papers is explained in Section 3.2.1. An analysis

of the studied sample and a taxonomy based on the findings is presented on Section 3.2.2. A description of the different lessons leaned based on the data studied is depicted in Section 3.2.3. Moreover, in Section 3.2.4 we provide a summary of the previous studies and compare them to the current study

3.2.1 Research methodology

In order to ensure the validity of our study and its repeatability, the Typical Systematic Review Stages proposed in (95) have been followed. For the sake of clarity related phases have been grouped together, resulting in the following set of steps:

1. Identify and define the question this study intends to address.
2. Determine and search the relevant studies regarding the previous question.
3. Identify those studies that meet the criteria.
4. Extract and synthetize the findings from the studies.
5. Write a report and consider potential effects.

Each of these steps are introduced below, with the exception of the latter which is indeed materialized in this manuscript.

3.2.1.1 Identify and define the question

The purpose of this study is to analyze how cybersecurity has been tackled when blockchain technologies are at stake. So, the main question is: *Which mechanisms or techniques have been proposed to achieve cybersecurity in blockchain-based systems?* In order to answer this general question, a set of more concrete matters are identified:

1. RQ1. Which techniques have been adopted to achieve cybersecurity?
2. RQ2. In which application areas have cybersecurity been achieved assisted by blockchains?

3. RQ3. Which are the blockchain technologies that have been more/less combined with cybersecurity?

4. RQ4. Is there any evidence of unjustified use of this technology for cybersecurity in academic papers?

Each of the aforementioned questions is targeted to a different audience profile. In particular, RQ1 is relevant for cybersecurity practitioners, whose interest lies on the concrete details of the techniques that turn a blockchain-based system into a cybersecurity solution. On the other hand, RQ2 is interesting for researchers working on the provision of advanced services in a given topic area (e.g., IoT). In this case, they are not willing to know the internal, low-level description of the mechanisms but, the set of provided cybersecurity that if often guaranteed/provided for each one. RQ3 is interesting for blockchain developers as they want to spot which design decisions have received more attention and which ones are subject to further research. Last but not least, RQ4 is interesting for a general audience in order to know the real advancement from the academic perspective. In order to provide with a more complete understanding of the matter, the evolution of each of these issues over time is considered as well.

3.2.1.2 Determine and search the relevant studies regarding the previous questions

The set of papers at stake is formed by both journal and conference/workshop papers. Due to the huge amount of publications in the last couple of years (2019 and 2020), the methodology for selecting academic papers published in these years is slightly different from the previous ones.

DBLP database (96) is considered to retrieve all manuscripts. Only contributions published in top venues are taken into consideration. Thus, only papers published in the first quartile of Computer Science in the Journal Citation Reports ranking (97) are at stake. Concerning conferences, those ranked in type A of the

GII-GRIN-SCIE ranking (98) are selected. On the other hand, Google Scholar has been considered to filter out those papers with 100 citations or more. This promotes that papers that have not been published in the said venues, but are relevant for the research community, become part of the sample. However, this issue is not considered in 2019 and 2020 because of the little time for them to achieve a high number of citations.

The following query has been developed to filter out relevant contributions based on their title:

> *(Blockchain OR Bitcoin OR Hyperledger OR Ethereum OR Solidity) AND (contract OR secur* OR priva* OR accountab* OR anonym* OR authentic* OR confiden* OR identity OR access* OR trust* OR distributed OR encrypt* OR hash OR cryp* OR DDoS OR malware OR anomal* OR avail*) AND NOT (survey OR (literature AND review))*

The query above ensures that the main cybersecurity terms are considered, even in different forms. After this step, a total of 506 journal and conference papers were retrieved. Note that not all the used databases allow such a query. In such cases, it has been transformed into an equivalent set of queries using the allowed operators.

3.2.1.3 Identify those studies that meet the criteria

Once the initial amount of proposals is automatically retrieved, a manual review is carried out. This ensures that those papers that are not relevant for the sample (e.g., literature surveys) or that do not contain any particular application for cybersecurity (e.g., smart contracts design related papers) are filtered out. After this analysis, the sample is definitely formed by 272 articles – 166 journal papers and 106 conference/workshop papers.

3.2.1.4 Extract and synthesize the findings from the studies

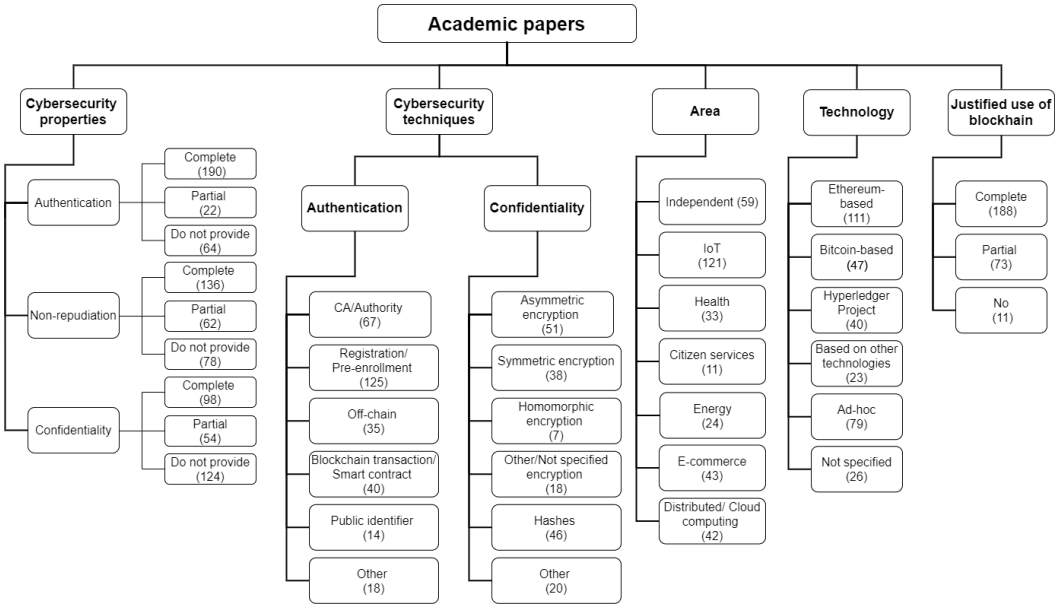

Figure 3.1: Taxonomy of elements involved in the analysis. Numbers in brackets correspond to the amount of proposals that fit in each category.

The chosen proposals are studied in detail, classifying them according to different features. In the case that a certain proposal fits into more than one category per feature, (e.g., belonging to IoT and Health areas), it is counted in each of them. Four aspects have been analyzed in each proposal, namely the offered cybersecurity properties and techniques, the application area, the underlying blockchain technology and the justification of using a blockchain. This classification, depicted in Figure 3.6, will be used as the basis for the following analysis.

3.2.2 Approaches study

In this section, all papers are analyzed to answer the proposed research questions – RQ1 to RQ4 (recall Section 3.2.1.1). In particular, section 3.2.2.1 answers RQ1 by explaining how cybersecurity properties are fulfilled and which techniques provide them. Afterwards, Section 3.2.2.2 answers RQ2 describing the areas in which blockchain-based systems have been applied to achieve cybersecurity. Section 3.2.2.3 analyzes the use of blockchain technologies and implemented cyberse-

curity properties to answer RQ3. Finally, Section 3.2.2.4 answers RQ4 by studying whether the use of blockchain technologies is justified in each proposal. For the sake of clarity, a table supporting this study is included in the Appendix.

3.2.2.1 Cybersecurity properties and related techniques.

Cybersecurity properties defined in Section 2.3 are individually analyzed. Three categories are considered for each property– whether it is fully, partially or not provided. Figure 3.2 summarizes the provision of each property over time. The different techniques applied to provide them are also introduced.

Authentication

Authentication is studied considering all blockchain entities (recall Section 2.2).

- Complete authentication: All entities are always authenticated. To do so, a Certification Authority (CA) can be used (99; 100); the user can be registered or included in a private network (in which at least the administrator of the network knows his/her identity) (101; 102); roles can be assigned accordingly (103; 104); some off-chain communication or registration system can be applied (105); or a unique public identifier can be used for authentication purposes (106).

- Partial authentication. Not all entities are authenticated. This occurs in (107), where vendors and system operators are publicly known but the system user is not. Other example is (108) where nodes within the same group know the identity of each other, but they do not know the nodes outside the group.

- No authentication. Authentication is not provided in any way or it is not mentioned, such as in (109; 110).

On the other hand, the following techniques are commonly applied for authentication purposes:

- CA/Authority. Some proposals provide authentication by means of a CA or other similar entity (e.g. governments, Key Generation Centers, Attribute Grant Unit, etc.). These authorities usually grant the requester an identifying item (e.g., certificates, keys, roles/attributes). Proposals like (111; 112; 113) are included in this category.

- Registration/Pre-enrollment. Some works need participants to be registered in the system beforehand. This is the case of proposals using a private blockchain, in which only specific entities are able to join and have to be known or approved beforehand, by, for example, the blockchain administrator (114; 115). On the other hand, some works based on pre-existing public blockchain networks (e.g. Ethereum main network) need BCUs, BCOs and/or BCNs to register in the system before using it (116; 117).

- Off-chain. Authentication is carried out outside of the blockchain. For example, in (118) the user needs to know the service identity to generate the compound identity. This identity is shared among two or more parties, where at least one becomes owner and the rest have restricted access (become guests) (118).

- Blockchain transaction/Smart contract. A transaction in the blockchain or the smart contract (or equivalent) is used to authenticate the different entities, e.g. (102; 119).

- Public identifier. A unique public attribute is used. This is the case of (120) in which the DNS name is used, or (121) where the artist profile is linked to an Ethereum address (121).

- Other. Another method is used for authentication purposes. For instance, (122) uses biometric data to provide authentication and a token is applied in (123).

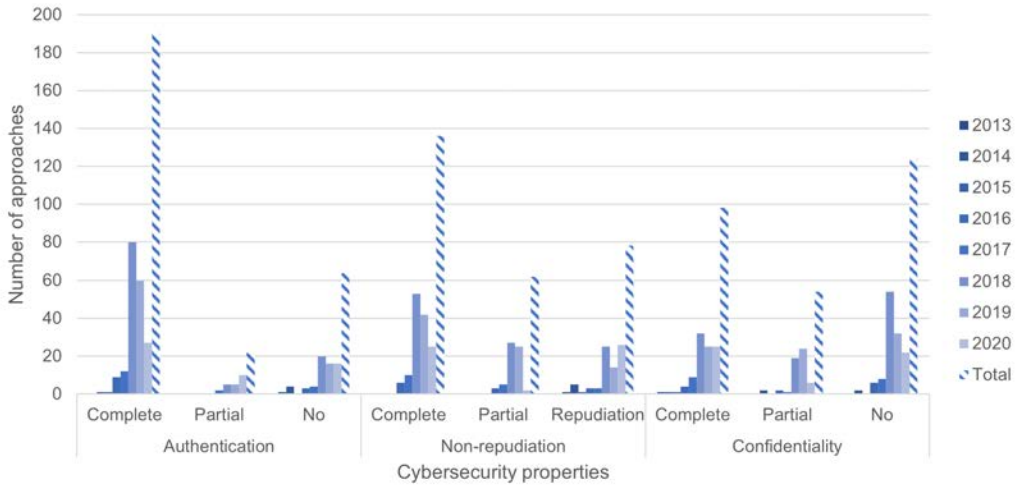

Figure 3.2: Number of approaches regarding cybersecurity properties per year.

The use and need of authentication in blockchains has increased since 2014 (recall Figure 3.2), when this feature was initially offered. In the following years, this number has increased and complete authentication is provided in around 67% of proposals in the last three years. Thus, the ratio has remained quite constant in the last years and it could be related to the appearance and raise of private and permissioned networks, versus the initial public permissionless ones (e.g., Bitcoin).

In terms of techniques (Table 3.1), in 2014 and 2015 authentication was provided off-chain. In the following years, the preferred method is registration/pre-enrollment. This technique could have a lot to do with the increase of private networks. Moreover, it is a significantly easy way to manage access control, as it has been used for many years now. Similarly, the use of an authority or trusted third party is also a common easy way to provide authentication even outside of blockchain systems. Indeed, the use of trusted third parties is a widespread solution to solve security issues, although these entities become a single point of failure. However, the use of these trusted entities has significantly decreased in 2020, trying to look for completely decentralized approaches.

Table 3.1: Number of approaches regarding cybersecurity techniques per year

	2013	2014	2015	2016	2017	2018	2019	2020	Total
Authentication									
CA/Authority	0	0	0	2	6	31	25	3	67
Registration/Pre-enrollment	0	0	0	5	9	51	47	13	125
Off-chain	0	1	1	3	3	10	5	12	35
Blockchain transaction/smartcontract	0	0	0	1	4	20	10	5	40
Public id.	0	0	0	2	4	0	7	1	14
Other	0	0	0	1	0	9	6	2	18
Confidentiality									
Asymmetric encryption	0	0	0	2	5	13	18	13	51
Symmetric encryption	0	0	1	2	0	8	12	15	38
Homomorphic encryption	0	0	0	0	1	1	5	0	7
Other/Not specified encryption	0	1	0	0	1	10	3	3	18
Hashes	0	0	1	1	1	17	15	11	46
Other	1	1	0	1	2	5	6	4	20

Non-repudiation

Non-repudiation is studied considering all blockchain entities in the system, analyzing how it is provided.

- Complete non-repudiation: Actions of all entities are recorded in the blockchain, e.g. (124; 125). Blockchain technologies already provide some kind of non-repudiation by means of digital signatures.

- Partial non-repudiation. A low number of actions of some entities, e.g. BCOs, are not logged in the system. This occurs in (126; 127) where the audit process is not recorded in the system. This could lead to entities denying having performed an action (e.g. the audit process).

- Repudiation. Entities can deny having done actions or there is not much information to infer this issue, e.g. (128; 129).

All proposals between 2013 and 2015 do not provide non-repudiation, see Figure 3.2. From 2016 onwards, proposals provide complete non-repudiation in 50% of the cases or more and partial in around 25%. Despite the growth in the amount of papers, the ratio remains almost constant, except for 2020 in which it has decreased.

The provision of some kind of non-repudiation is specially appropriate to look for better traceability and auditing processes.

Concerning applied techniques, non-repudiation is achieved in a simple way. Either all actions of the different elements in the system are recorded in the blockchain or not. Logging is the only identified technique to achieve this cybersecurity property.

Confidentiality

Confidentiality is analyzed within the blockchain network, thus assessing whether the content of a message within the network is only accessible to authorized entities.

- Complete confidentiality. Only selected entities are able to know information from other entities, though there are elements inherent to the blockchain operation that are public and cannot be hidden, such as block headers (130). This property is offered, for instance, in (131; 132), where the block content is encrypted; or in (133; 134), where hashes are the only interchanged data.
- Partial confidentiality. Some information is accessible to a particular set of entities while other data is public or can be used to infer additional information. For instance, in (135), which proposes a voting system using blockchain, votes are encrypted but the registration content is not. (136) proposes a similar approach, in which users' data is encrypted, but cloud service providers, token and resource addresses are not.
- No confidentiality. The content is public to all entities that interact with the blockchain, such as (111; 137).

Confidentiality is achieved by encrypting transactions' content mainly, sharing only hashes of information and some other special cases. As different types of cryptographic algorithms are applied, different techniques are distinguished.

- Asymmetric encryption. It is also referred as public-key cryptography. It uses a pair of keys: public keys, which may be disseminated widely, and private keys, which are known only to the owner (138). Different approaches use this kind of encryption. (139; 140) use asymmetric encryption to encrypt the blockchain content.

- Symmetric encryption. In this case, the same key is used for encryption and decryption purposes. This key, also called secret key, is usually shared beforehand. It is also possible to derive the secret key based on assorted parameters. (114; 117) use symmetric key algorithms to encrypt exchanged data.

- Homomorphic encryption. It is a special kind of encryption that allows performing calculations over encrypted data. It is adopted in some cases as a means to protect privacy and, implicitly, confidentiality. Proposals like (141; 142) use this type of encryption.

- Other/Not specified encryption. Proposals that use other encryption types and those works in which the type is unknown are included in this category. For example, (143) uses secure certificateless multireceiver encryption which allows the sender to generate the same ciphertext for a chosen group of receivers solving the certificate management problem, while in (144) the type of encryption is not specified.

- Hashes. A hash is the result of applying a cryptographic non-reversible function, called hash function. They are usually used as indexes or as proofs of integrity because hash values are identical when applied over the same data. As in (145), because hashes cannot be used to discover the original message or any of its characteristics will be considered to provide confidentiality. Hashes can be identified as pseudorandom numbers or strings with no meaning and they do not provide information by themselves. For example, in (134) and

(146) document hashes are stored in the blockchain.

- Other. Any other technique is used to provide confidentiality. For example, (147) leverages additive secret sharing. Thus, the Key Distribution Center distributes n shares, derived from the requester secret key, to each user. Then, each user adds a share to contribute on blinding a given piece of data. All subsequent operations are performed on blind data. Finally, the impact of the share on the aggregated result can be eliminated by recovering the requester secret key.

Analyzing the trend over the years (recall Figure 3.2), confidentiality is specially considered since 2017. However, less than half of the proposals provide it completely and a big fraction do not care about confidentiality. This may be reasonable as the need for this property may depend on the type of data at stake, e.g. health data should be considered confidential. By contrast, if some authentication and non-repudiation techniques are in place and a private network is applied, some level of confidentiality protection is achieved, despite not using encryption.

Considering the different techniques (Table 3.1), in 2013, the only work that falls into the 'other' category uses a mixing service and a coin distribution service to change the transmitted amount of money. In 2014, hashes, encryption and 'other' techniques are equally applied. In 2015, symmetric encryption was used, but from that year onwards, authors prefer the asymmetric one in most cases, following by only sharing hashes. The use of encryption is the most common way to provide confidentiality, regardless of the use of blockchains. Moreover, asymmetric encryption seems to be more appropriate in a distributed environment as there is no need to share decryption keys privately. Thus, in the last three years, in which confidentiality techniques have been specially applied, asymmetric, symmetric encryption and hashes are the most common alternatives.

3.2.2.2 Application areas and cybersecurity purposes

In this work seven areas are distinguished based on the content and goals of studied proposals:

- IoT: Internet of Things (IoT) is defined as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) devices based on interoperable information and communication technologies (148). All proposals that suggest the use of blockchain in relation to IoT are included in this area. Devices/elements could be smartphones (149; 150); smart homes or related equipments (114; 151); sensors (102; 152): vehicles (100; 111; 153) or some other resource-constrained and potentially portable devices.

- Distributed/Cloud computing: Distributed computing is a system whose components are located on different networked computers, which communicate and coordinate their actions by passing messages to one another (154). Cloud computing refers to the on-demand delivery of computer power, database storage, applications and other IT resources (155). This could be used to increase the storage or computing power of a given system or application. In this category all kind of parallel, distributed and cloud computing systems are included. A special case of cloud computing is secure multiparty computation, which is used to increase data security by computing cryptographic operations, while keeping some data private (110; 110; 156).

- E-commerce: It focuses on the trade of goods via online services or over the Internet. Some common cases in this area are fair trade or fair lottery (110; 157); as well as the relevance of user security when buying or trading online (105; 141; 158; 159). Moreover, within this category we also consider e-business use cases, that is applications that affect economy in some way, but that are linked to business, such as the use of blockchain for doing supply

chain inventory (160), or carrying out human resources' records management (161).

- Citizen services: It involves proposals typically related to smart cities, where part of the government and citizenship duties, as well as institutional relationships between them are automated or centralized. In this way, electronic government (e-government) (162; 163); centralized student records (164); or electronic voting (e-voting) (135) are included in this area.

- Energy: Smart grids and power distribution supply proposals using blockchains fall in this area. Their goal is the improvement of energy distribution (127; 131; 144; 165).

- Health: Patient data or any kind of healthcare-related data could be managed through a blockchain. It can be applied, e.g., for improving patients access and control of their data (101; 166) or for sharing data between health professionals or institutions (167; 168).

- Independent: These proposals can be regarded as "area-independent" uses of blockchains. Some of them are indeed unrelated to any particular scenario, while other proposals focus on specific ones (e.g. software factories) but they could be easily adopted in other settings as well. Some examples include general access management mechanisms (143; 169); data provenance (170); or information sharing applications (132; 171); as well as malware analysis or other cybersecurity-centered proposals like DDoS prevention (172).

In spite of the previous classification, some proposals fall in several areas. For example, (110) is related to IoT and distributed/cloud computing; and (99; 173) can be involved in IoT, health and distributed/cloud computing proposals.

An analysis over time is depicted in Figure 3.3. In the early years, most works were focused on e-commerce but this trend has changed. Although this area is still present in the following years, its percentage has decreased. Distributed/cloud

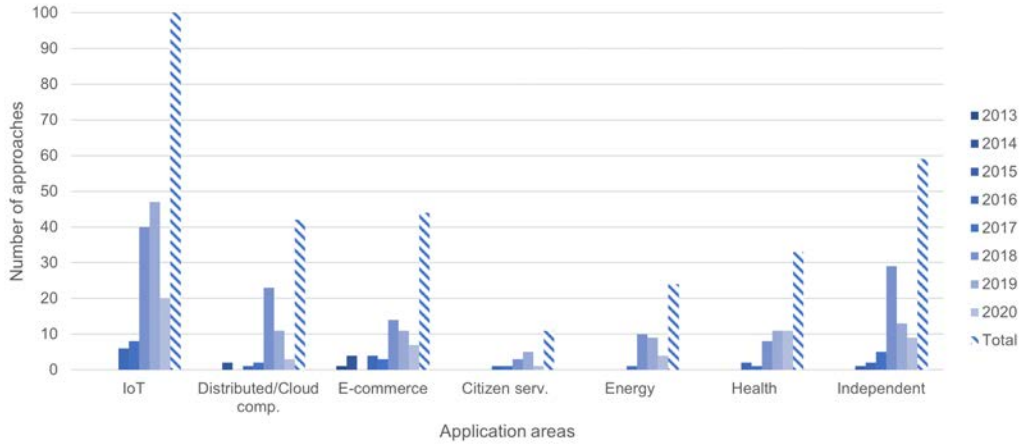

Figure 3.3: Number of approaches per application area and year.

computing related works seemed to be popular in 2014 and 2018 (40% and 21.9% of proposals respectively), but its popularity has also decreased over the years. IoT is the most popular area from 2016 to 2020, with more than 45.7% of the works. The second most popular one is area-independent, in an attempt to achieve generic solutions. 2017 and 2018 are specially remarkable because 27.8% and 27.6% of approaches fall in this category respectively.

3.2.2.2.1 Cybersecurity properties vs areas

Table 3.2: Number of approaches regarding cybersecurity properties per application area

		IoT	Distributed/ cloud computing	E-commerce	Citizen services services	Energy	Health	Independent
Authentication	Complete	88	27	22	9	18	28	41
	Partial	8	3	3	1	2	3	6
	Not provided	25	12	19	1	4	2	16
Non-repudiation	Complete	68	16	19	6	8	18	32
	Partial	26	14	11	3	8	5	12
	Not provided	27	12	14	2	8	9	18
Confidentiality	Complete	41	16	16	4	6	17	21
	Partial	26	13	10	2	2	7	7
	Not provided	54	13	15	5	16	9	34

Cybersecurity properties and areas are simultaneously studied herein (Table 3.2) to identify if there are properties specially related to particular areas. The fulfillment of each of these cybersecurity characteristics will be achieved on the

same bases as in Section 3.2.2.1– complete, partial or not provided.

Authentication

Almost all studies in citizen services (9 works), energy (18) and health (28) provide complete authentication and something similar happens in the IoT field (88). By contrast, e-commerce is the field in which this matter is less prevalent, just 22 proposals provide it completely and 3 partially. These results are probably related to the kind of provided service because, for instance, health and citizen services related proposals usually need to identify and authenticate users in the system before providing the service. Likewise, energy related works also need to authenticate entities as well as some other information (e.g. location). On the contrary, the use of blockchain in e-commerce was born to change the need of authentication, thus the lack of this property in these proposals is not surprising.

Non-repudiation

Most approaches in the health (23 works), citizen services (9) and IoT fields (94) provide complete and partial non-repudiation. Area-independent proposals also provide complete and partial non-repudiation in a large number of them, that is in 32 and 12 respectively. The remaining areas (cloud computing, e-commerce and energy) also present high provision of this property, 68.7% on average, but little lower that in other areas. Non-repudiation is usually a very important feature for the health and citizen services area, as personal information is commonly at stake. Being able to trace who accesses to which data and how it is carried out is very useful for accountability purposes. IoT devices sometimes also have access to private information related to people homes and lives, so the same reasoning applies. 68% of distributed/cloud Computing and energy proposals also provide this property. In these fields logging operations are not regarded as critical as the operations themselves.

Confidentiality

E-commerce is the field in which the biggest percentage of works offer complete confidentiality, 16 completely and 10 partially. Given that blockchain technologies where initially used for cryptocurrencies, where confidentiality could also be desirable to hide transactions' content, specially in permissioned networks. Health-related works usually count on high levels of authentication, strict access control policies and use private networks. However, probably because of the management of sensible data in health systems, this is the second area which provides confidentiality the most, 17 proposals completely and 7 partially. By contrast, energy works do not really care about this property, just 6 provide confidentiality completely and 2 partially. It is presumably due to the use of authentication techniques and the use of the blockchains to store power consumption data which is not considered sensible by itself.

3.2.2.3 Blockchain technologies and cybersecurity properties

Different technologies can be used when blockchains are involved. Bitcoin, Ethereum and the Hyperledger Project are three representative alternatives (recall Section 2.1.2). However, since there are different variants, several categories are identified. On the one hand, some authors rely upon a technology derived from Bitcoin or Ethereum, referred to as Bitcoin-based and Ethereum-based. Other authors propose an ad-hoc technology, for example by proposing new block or transaction formats that suit their needs. Another subset of proposals are based on different alternatives (classified as 'other'), that is, existing technologies different from the main ones. For example, (174) uses Monero, whereas (170) opts for Scrybe. Additionally, some proposals are technology-independent or can work with multiple ones and thus they will be included in each of the previous categories. For instance, (175) and (176) combine a public ledger with a private one. Last but not least, technology is not always specified – authors may not explicitly mention this issue

or the proposal is so general that can be implemented using several technologies but without giving details in this regard, e.g. (139). In these cases, proposals are classified as 'not specified'.

Figure 3.4 shows the amount of proposals per technology and year. The most common technology is Ethereum-based, possibly due to its flexibility and the use of smart contracts (177; 178; 179). The second largest group is ad-hoc technologies (144; 166; 180). The third most popular technology is Bitcoin-based (105; 113; 181). Hyperledger Project is in fourth place, being Fabric chosen in most cases (101; 150; 156). One exception is (150) which uses Iroha. The fourth largest group correspond to proposals based on other technologies, for example LSB (139), BigchainDB (181), Zerocoin (182), Multichain (183), Scrybe (170) or Monero (174).

As the blockchain concept has gained popularity, new technologies have been developed. As seen in Figure 3.4, Bitcoin (2013-2015) was the most well-known technology at the very beginning and received attention in 2017, but no proposal is identified in 2020. Nonetheless, after Ethereum emergence (2016-onwards), this technology gained ground, being the main one used in the whole period except for 2017. In 2016, ad-hoc technologies appeared for the first time, and have been gaining momentum over the years, being the second most popular choice since 2018. The great used of Ethereum can be linked to the fact that it allows the development of Turing-complete smart contracts and it can be used as a public network or as a private one.

Cybersecurity properties and the different technologies are simultaneously studied herein to identify if there is some link between them (Table 3.3). Note that proposals in which properties are not managed, because they are not explicitly pointed out or they cannot be inferred, are classified as "Not specified".

Authentication

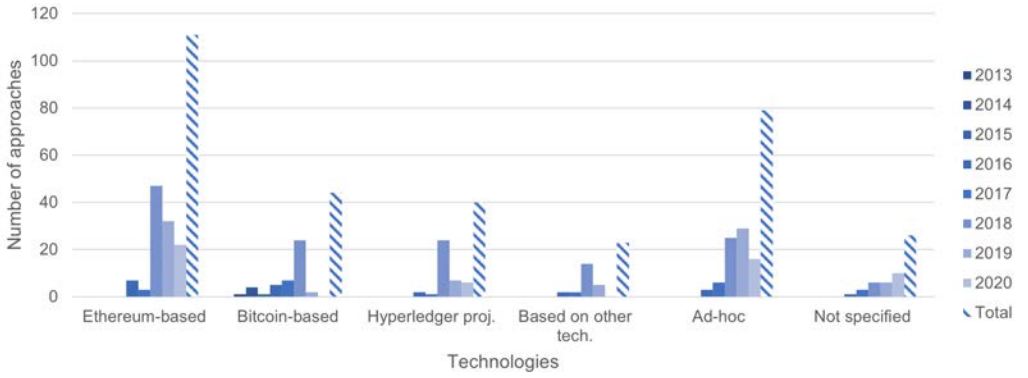

Figure 3.4: Number of approaches regarding blockchain technology implementation per year.

Table 3.3: Number of approaches regarding cybersecurity properties per technology

		Ethereum-based	**Bitcoin-based**	**Hyperledger project**	**Based on other technologies**	**Ad-hoc**	**Not specified**
Authentication	Complete	78	28	31	27	53	21
	Partial	9	2	3	5	7	2
	Not provided	24	17	6	13	20	3
Non-repudiation	Complete	56	16	26	20	43	13
	Partial	28	8	7	9	14	5
	Not provided	27	23	7	4	23	7
Confidentiality	Complete	28	19	12	19	20	12
	Partial	14	5	4	7	4	3
	Not provided	35	18	15	13	25	5

The great majority of papers based on Hyperledger project (31 works), not specified (21) and Ethereum-based (78) categories provide complete authentication. Those based on other technologies also provide authentication in most of them (27 completely and 5 partially). On the other hand, a smaller set of works use Bitcoin-based technology (28 completely and 2 partially). These results are probably due to the fact that Hyperledger Project technologies are often private, so they need some kind of user authentication. Ethereum allows the use of private networks too, which could be the reason of providing authentication in most cases. However, regardless of the technology this cybersecurity property is provided quite often.

Non-repudiation

This property is considered in all technologies to some extent. Hyperledger project is present in the highest amount of proposals (26 completely and 7 partially)

whereas Bitcoin-based is in the lowest one (16 completely and 8 partially). Thus, non-repudiation is most provided in technologies that allow private and/or permissioned networks (Hyperledger Project, Ethereum and ad-hoc) and less in public/permissionless ones (Bitcoin). However, in all cases complete non-repudiation is is preferred – in some cases it doubles the amount of proposals in contrast to partial non-repudiation.

Confidentiality

Confidentiality provision does not seem to be linked to particular technologies in any way. Proposals based on Ethereum-based and ad-hoc technologies are those in which it is less considered, 42 (18.9%) and 24 (15%) proposals respectively. By contrast, those Bitcoin-based or based on other technologies apply complete confidentiality more frequently, 26 (42.2%) and 24 (40.4%) proposals respectively. It may be due to being public networks in most cases.

3.2.2.4 Use of Blockchain. Justification

The actual need for blockchain is studied in all proposals, considering the principles stated by Greenspan (recall Section 2.4). Based on the fulfillment of these principles, three different categories have been considered:

- Complete justification. All criteria are met. This includes proposals like (177; 184). At first glance, it may seem that private and permissioned networks do not achieve *Inter-writer mistrust* and/or *Disintermediation* because some level of trust is required between the peers– they often need to trust the organization(s) controlling the network. However, according to (68), users cannot trust each other even between the same organization. As a special note, those systems that only share hashes in the blockchain will be included in this category, as long as they fulfill all the remaining conditions and assuming that, though they do not *represent something that has real-world value* per se, they do serve as a pointer or proof to something that does (e.g. (185; 186)).

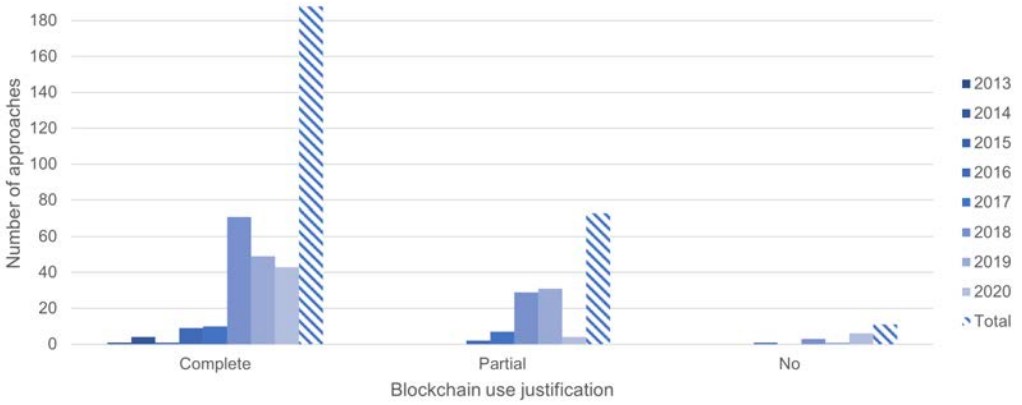

Figure 3.5: Number of approaches regarding blockchain use justification per year

- Partial justification. Systems in which a trusted third party or authority knows the nodes writing into the blockchain fall into this category. In this case, the *disintermediation* and even the *inter-writer mistrust* principles are not fully met. Thus, proposals are included in this category as long as the remaining principles are met. For example (111; 187).

- No justification. The criteria are not fulfilled (except for those exceptions mentioned above). This happens, for example in (188) where, even though there are multiple users in the system, only OriginStamp submits transactions to the blockchain. Another example is (189), where an entity may be able to modify data stored in the blockchain.

Most proposals provide a complete justification (188), though this number has significantly increased in 2018 (71), see Figure 3.5. A smaller amount of them integrate the blockchain in their systems with partial justification, being 2018 and 2019 years that stand out from the rest (29 and 31 proposals respectively). The high number of proposals with partial justification could be due to the need to trust an entity and the raise of technologies that allow private and permissioned networks in contrast to the initial preference for public ones (e.g. Bitcoin). On the other hand, the use of blockchain is not justified in 11 proposals. Though this is

not a high number, it shows that some research results are using blockchains in an improper manner.

Whether the use of blockchain is justified or not has changed over the years, but maintained high percentages of complete justification (more than 60%). Most unjustified proposals are relatively novel as they belong to 2018 (4), 2019 (1) and 2020 (6). One potential reason is that blockchains were initially developed as an e-commerce technology. From then on, they have been used for many other activities and maybe some of them do not need all features they offer.

3.2.3 Lessons learned

Once the considered sample has been analyzed, it is possible to identify a set of lessons learned to summarize the main findings of the study.

Lesson 1. Recent and growing interest. The academic interest of blockchain when cybersecurity is at stake has rocketed since 2016. Although our thesis covers since 2013, it is in 2016 when a dramatic increase on the amount of papers is perceived.

Lesson 2. Preferred cybersecurity properties. Authors usually tend to implement some cybersecurity properties over others. Authentication and non-repudiation mechanisms are often provided, and though their joint application provides some kind of secrecy, if such countermeasures are bypassed, confidentiality would not be achieved. Indeed, confidentiality is applied to a lesser extent.

Lesson 3. Simpler and most well-known techniques to provide cybersecurity properties is often used. Authors seem to prefer the easiest, most well-known cybersecurity techniques when applied. For example, Registration/Pre-enrollment or simply using a CA/Authority in order to provide authentication, or asymmetric encryption and sharing hashes for confidentiality. There is a lack of approaches relying upon novel lightweight or non-conventional cryptographic techniques.

Lesson 4. Topic alignment, under-represented areas. All approaches are similar in their choice of focus – IoT and area-independent proposals are very prominent. While area-independent approaches can be perfectly valid, there is an underlying threat of forgetting specific requirements (e.g., tailored trust assumptions) that might render a particular use case unsuitable for blockchains. On the other hand, energy applications are developed in just a small set in academia, considering only approaches in which cybersecurity is addressed using blockchain and not the general use of blockchain for energy provision. Something similar happens in academia concerning cybersecurity in citizen applications, as this area has received little attention.

Lesson 5. Preferred cybersecurity properties are strongly related to the area of the proposal. Depending on the area of the proposed system, some cybersecurity properties are preferred over others. For example, authentication and non-repudiation are often implemented in areas like health and citizen services, while not so much in e-commerce.

Lesson 6. Ethereum prevalence. Proposals are firmly choosing Ethereum-based technologies. One reason is the use of smart contracts, which are at stake in the majority of the papers. Another factor could be that most of the technical books, references and sources of information about blockchain are centered in Ethereum and Bitcoin. However, ad-hoc technologies have gained momentum over the years so this trend may change and it is considered the second preferred alternative, followed, by far, by Bitcoin-based and Hyperledger project technologies.

Lesson 7. The use of blockchain is mostly justified in academic approaches. Most academic proposals use blockchain technologies in a justified manner, though around 26% in a partial way.

3.2.4 Related works

Blockchain is a trending topic nowadays and lots of studies have been developed in this regard. Security has not been neglected either. For example, in (190) security issues of blockchain technologies are studied. (191) presents a deeper analysis, describing vulnerabilities and attacks of blockchain technologies. Also, (192) surveys security threats and real attacks against blockchain systems. Considering blockchain security but from a different perspective, (193) explores business, organizational and operational issues. In this vein, security issues at different levels, such as data, smart contracts or networking protocols are studied.

In terms of blockchain applications and cybersecurity, (194) points out blockchain advantages and classifies blockchain applications for cybersecurity. Similarly, (195) and (196) analyze blockchain-based applications, though the latter also points out blockchain security and privacy challenges. (197) surveys blockchain applications in the area of fog-enabled IoT. Given the relationship between blockchain and fog computing in IoT, it is studied the fulfillment of cybersecurity goals. In this same context, (198) analyzes the integration of blockchain, categorizing applications but without a clear focus on security, though highlighting its need and pointing it out as a challenge. Focusing on Industry 4.0, which can be considered within IoT solutions, (199) presents an extensive survey on how blockchain systems can overcome cybersecurity barriers. Some cybersecurity issues are analyzed, like failure of key nodes in centralized plarforms, but this work does not directly analyze cybersecurity properties and techniques. To the best of the authors knowledge, only (200; 201) focus on studying how cybersecurity is achieved when applying blockchains. In both cases the sample is substantially smaller, 30 and 33 papers in (200) and (201) respectively. Moreover, in (200) the analysis is quite limited, without providing a careful review of methods to provide cybersecurity properties. By contrast, (201) bases on electronic health record systems exclusively.

A summary of related works considering the proposed research questions is de-

picted in Table 5.5. Note that there are many other proposals focused on blockchain cybersecurity which look for analyzing attacks, threats and vulnerabilities, but only those that share the goal of this study are considered herein. Indeed, this thesis studies the relationship between cybersecurity objectives and blockchain capabilities, considering their application areas or technologies among other issues. Moreover, technologies at stake and the analysis on the justified use of blockchains (questions RQ3 and RQ4) have not been explored yet.

Table 3.4: Related works comparison concerning proposed research questions.

	RQ1	**RQ2**	**RQ3**	**RQ4**
(192)	x	x	x	x
(193)	x	x	x	x
(194)	x	√	x	x
(195)	x	√	x	x
(191)	x	x	x	x
(197)	√*	√*	x	x
(196)	x	√	x	x
(199)	x	√**	x	x
(200)	√	√	x	x
(201)	√***	√***	x	x
OURS	√	√	√	√

*In fog– IoT

**Industry – IoT

***In healthcare

3.3 Malicious uses of blockchains by malware

In this section, how malware utilizes blockchain is analyzed. In Section 3.3.1, we first describe the research methodology used to retrieve and select the studied works. In Section 3.3.2 categorize them according with how they use the blockchain. In Section 3.3.3we summarize our findings in different learned lesson and expose some research gaps. Lastly, in Section 3.3.5 we review past related works and compare them to our study.

3.3.1 Research methodology

The same methodology used in Section 3.2.1 is used. In this case, our main research question would be *How different malwares use the blockchain?* Therefore, the goal is to answer that question and identify research gaps to develop a mechanism afterwards. The applied methodology bases on starting searching for relevant research papers (both journal and conference/workshop papers) in Google Scholar.

The following queries have been developed to filter out relevant contributions based on their title:

> *blockchain AND (malware OR botnet OR ransomware OR trojan OR spy*)*
>
> and
>
> *(Blockchain OR Bitcoin OR Hyperledger OR Ethereum) AND (malware OR trojan OR backdoor OR botnet OR spy OR ransom*)*

The queries above ensures that the main terms are considered, even in different forms. After this step, a total of 400 proposals were retrieved, ordered by relevance. Then, a manual review was carried out, which includes a snowball search of the papers. This ensures that those papers that are not relevant for the sample (e.g., literature surveys), are repeated among queries or that do not use the blockchain to enhance the malicious capabilities of the malware (e.g. cryptominers) are filtered out. After this analysis, the sample is definitely formed by 125 proposals.

Identified proposals are studied in detail, classifying them according to different features. In the case that a certain proposal fits into more than one category per feature, (e.g., possibility of use different technologies), it is counted in each of them. The following features are analyzed in each proposal.

- Type of malware. The analysis of the malware type related to blockchains may provide information to defenders for a better understanding of how to react against such malicious programs.

- Blockchain technology. As each one exhibits different features (recall Section 2.1.2), it may help defenders to identify which ones are present (e.g. real anonymity if Monero is at stake) to counter a threat.

- Desired properties. As mentioned in Section 2.1.4, blockchains provide some properties by default. Knowing which ones are relevant for malwares may be helpful for defending against them.

- Used blockchain elements. The analysis of used blockchain elements (recall Section 2.1.3) provides a deeper understanding of how the system is developed and what it is really needed to work properly. This entails a better understanding of how a malware works and what resources it uses, which can help in an earlier detection and better defense.

- Seed address. If a malware is using a blockchain, an address is needed for such interaction. How this address is delivered to the malware could help defenders prevent this communication taking place.

- Data protection. Blockchain data can be protected to provide confidentiality, or even secrecy with covert channels. In this way, the malware communication stealthiness is studied in order to provide a better understanding and early detection of possible hidden communications.

- Goal of using blockchains. Studying the use of blockchain for malicious purposes helps defenders understand what they can expect or search.

- Cost for the attacker. Whether there is a cost for the attacker and how much an attack would cost are important matters in terms of proposals' feasibility. If the cost of using a given approach is too high, its feasibility can be negatively affected.

These features, depicted in Figure 3.6, are used as the basis of the following analysis. Numbers in brackets represent the amount of proposals in each category.

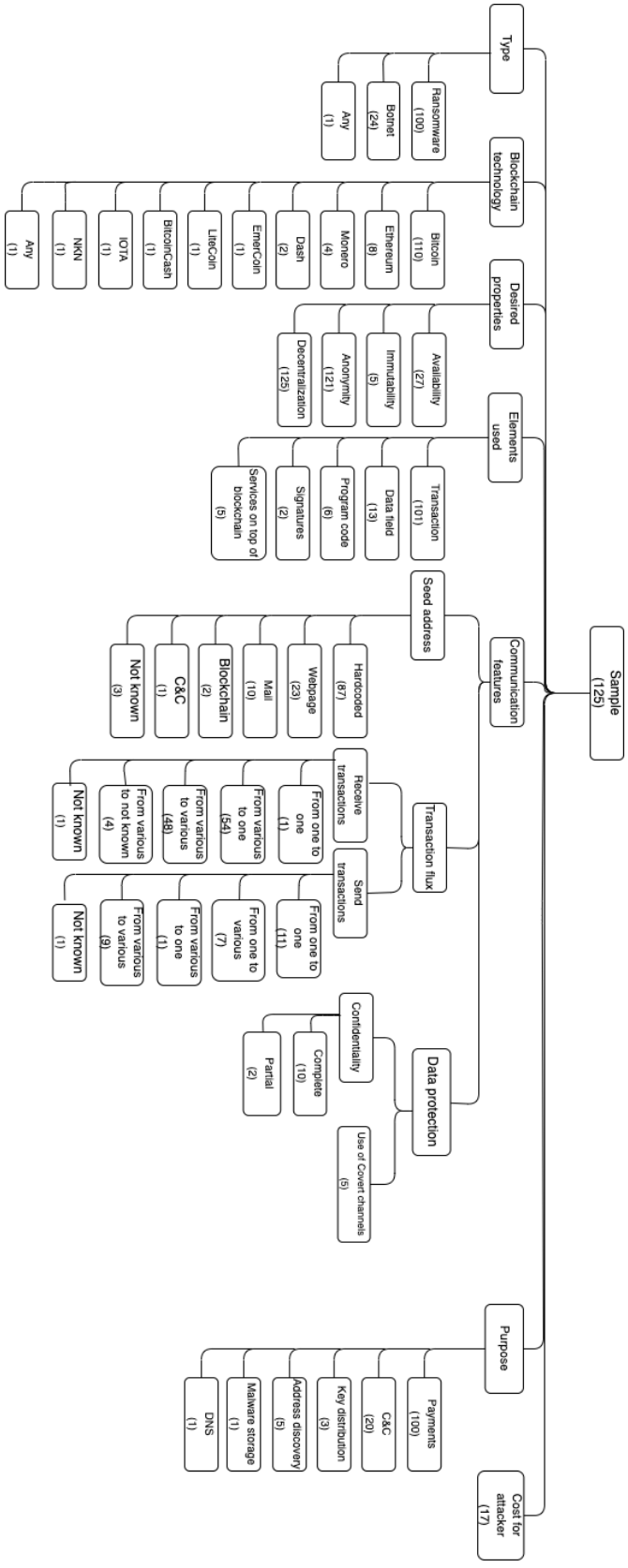

Figure 3.6: Proposals analysis

3.3.2 Malware and blockchain analysis

How malwares are being used leveraging blockchain is studied herein (recall Section 3.3.1). For the sake of clarity, the analysis is structured following the dimensions shown in Figure 3.6.

In particular, Section 3.3.2.1 discusses the blockchain properties relevant for malware. Section 3.3.2.2 presents the communication features. The set of blockchain elements at stake and the data protection issues are described in Section 3.3.2.3 and 3.3.2.4, respectively. The purpose, malware type, blockchain technology and cost for the attacker are presented in Sections 3.3.2.5, 3.3.2.6, 3.3.2.7, 3.3.2.8. The summary of the analysis is addressed in Section 3.3.3. Furthermore, a discussion of how Mitre Att&CK matrix could be applicable in our research is available in Section 3.3.4 . Related work regarding this topic is available in Section 3.3.5.

3.3.2.1 Desired properties

Properties provided by blockchain in malware proposals are studied in the following:

- Availability. Blockchain information is always available, although accessibility is limited by the nature of the blockchain (private and public). In our study, 27 works of the sample aims to exploit this characteristic. For example, (25; 202; 203) intend to provide availability.

- Immutability. Content, once mined, is very difficult to change as an attacker needs to be in control of 51% of the nodes of the network (recall Section 2.1). Only 5 proposals use this feature (e.g. (202; 203; 204)).

- Decentralization. As a distributed ledger, blockchain is decentralized by design and some level of decentralization is provided in all works.

- Anonymity. Authors use the blockchain with the intention of providing pseudo-anonymity in 117 works (e.g. (25; 26; 205)) and real anonymity in

4 of them ((206; 207; 208) and (209) if Monero is the chosen currency to pay with).

According to this study, blockchain are mainly used to provide some level of decentralization, thus making more difficult to locate and take a malware down, and anonymity/pseudoanonymity, to hide data origin. Availability, although useful, is only considered in less than 21.6% of the cases, for example by (202) to download different parts of the malware at any time. This property is also common in the development of botnets (e.g. (210) and (211)) to make them more resilient and always accessible. Immutability of the content is the least used property, though (202) provides it for different chunks of malware stored in the blockchain to not be changed over time if discovered. (203) also exploits this feature, a semi-autonomous ransomware is developed and its code in the smart contract cannot be changed.

3.3.2.2 Communication features

3.3.2.2.1 Seed address

The seed address can be the one receiving payments ((26), (205),(212)). It can also be used to manage the malware, for example by using smart contracts (203). There is only one case in which there is no seed address, as the system checks the validity of every OP_RETURN transaction posted in the blockchain (210).

The way the seed address is delivered is classified as follows:

- Hardcoded. The address is written in the malware itself, for example in a file or in the code. Most of the seed addresses are hardcoded (87 proposals) such as in (26), (205) or (27).

- Webpage. The address is delivered via web when the client or victim accesses a website ((212), (213),(214)). Delivery via website is the second most common method, with 23 proposals.

- Mail. The address is delivered via email, e.g. (215),(216) or (217). This is the third most common category with 10 cases.
- Blockchain. The address is extracted from the blockchain itself. This happens with all addresses except for the first one in (218) and (211) proposals.
- C&C server. The address is delivered via a C&C server. Each time an address has been used, a component in charge of monitoring the health of the C&C server, sends the new address to this server, which then sends it to bots (219). 1 work uses this method.
- Unknown. How the address is delivered to the system is unknown. (203), XLockerv5.0 in (220) and (206) are the ones that fall in this category.

3.3.2.2.2 Transaction flux

This Section focuses on the communication patterns between the malware and the blockchain. In particular, this communication will take place by means of sent or received transactions.

Transactions can be sent from one or multiple sources (e.g unique attacker address or different attackers/addresses) to one or various seed addresses which will be used by the receivers (bots) to interact with the malware or victims . In the case of receiving transactions the roles are reverse.

Most transactions are sent from one to one (11 works), as in (221) where the attacker only interacts with the smart contract address used to control the bots. Furthermore, there are 9 cases in which there are either various attackers or various addresses used by the attackers to communicate with a range of different seed addresses ((219),(207),(204)) . After that, the third most common flux is from one address to various addresses with 7 proposals of this kind (e.g. (222) and (223)). Moreover, in (224) the author of the ransomware, or an individual renting its services, communicates with the ransomware smart contract, thus being a various

to one communication. In the case of (225) it is known that the malware sends transactions but it is not specified how.

Regarding received transactions, most of them are from various to one (54 works). They are mostly ransomware in which various victims pay the ransom to a unique attacker address (e.g.(26), (226)). This category includes those works whose attacker address is static for a long time, but it is changed at some point. For example, FakeGlobe ((220)) . FakeGlobe ((220)) changes its address per malware campaign. The second largest group are those in which different clients communicate with a range of attacker addresses (48 proposals). For example, in Locky (220) a wide range of attackers addresses are found. However, one collector address which sends money back to the attacker address (from one to one) is used in (223). On the other hand, the number of seed addresses are not identified in 4 samples. For example, Dharma addresses are distributed by mail in (227). Fantom (228) is another example in which it is unknown if it uses an unique address or not. Finally, in (225) it is known that the malware receive transactions but the way it is not specified.

3.3.2.3 Used blockchain elements

This section studies what elements of the blockchain are used in the analysed proposals:

- Transactions common fields. In every blockchain system there are fields that are usually common. Transactions usually need a sender address, a receiver address (in some of them can be null) and a value to exchange (this one is specially common in cryptocurrencies, recall Section 2.1.3). These fields are applied in 101 proposals, such as in (26; 205; 212).

- Data. Field or type of transaction in which arbitrary data can be included (Section 2.1.3). This has been considered apart from transaction common

fields because some technologies (e.g., Bitcoin) only include this field in specific transaction types. This is the case of 13 proposals ((202; 204; 210)).

- Program code. A program hosted in the blockchain (e.g. smart contracts in Ethereum) is used in 6 works. They are often used to send commands to the bots of a botnet. For example (203; 221; 229).

- Nonce for digital signatures. Blockchain usually use elliptic curves to generate signatures linked to transactions. In this process a secure nonce is generated per message for later use in the computation of the curve. This value is modified in a pair of proposals (211; 230). As it happens with the program code, the nonce is usually used as a way to send commands to bots.

- Services on top of blockchain. For example, EmerDNS is used to provide a DNS service over EmerCoin and (28) uses it. Whisper is a message protocol on top of Ethereum, applied in (231). Only 5 works use these services.

3.3.2.4 Data protection

This section studies how information is protected by analyzing whether it is confidential or if covert channels are used to conceive data exchanged. First in order to evaluate the stealthiness of the mechanism an attacker model is proposed in Section 3.3.2.4.1. Then the data is analyzed in Section 3.3.2.4.2.

3.3.2.4.1 Attacker model

In this section the sample is studied in terms of data protection.

To evaluate covert channel stealthiness we adopt the following attacker model: three different types of attackers are considered. A pair of them are assumed to be passive, inspecting blockchain contents using a block explorer (e.g., Etherscan (34)). However, while one of them is an eavesdropper (Basic Eavesdropper, BE), the other one might carry out syntactic checks on each transaction (Advanced Eavesdropper, AE). The third type of attacker (Interactive Attacker, IA) is active,

being able to make transactions. Therefore, while BE and AE threaten the secret's confidentiality, IA aims to impact its integrity. Lastly, we will also consider a special case of BE, BE* (Basic Eavesdropper with a simple hiding technique) when there is no really a hidden technique in place or it is extremely simple, so the message will be easily spotted by anyone.

3.3.2.4.2 Data protection analysis

Some proposals need to exchange the data order for the malware to work properly. This data can be transmitted via normal channels or covert channels (recall Section 2.6). To evaluate covert channel stealthiness we adopt the attacker model defined in 3.3.2.4.1, composed of three different types of attacker.

Data posted in the blockchain can be either in clear or hidden in some way, e.g. encrypted. For example, (232) and (219) both use RC4 to encrypt data transmitted to the blockchain. It can also be obscured by sharing only hashes of the data ((221)).

Table 3.5 shows the maximum capacity (in bytes) of each proposal in which data is exchanged in the blockchain; which element is used to exchange the information; and the attacker model applied in terms of stealthiness in case they use a covert channel. There is some kind of information exchange in the blockchain in 24 works, while the remaining ones do not use this technology for that end.

According to the table, OP_RETURN in Bitcoin, the data field in Ethereum and Payment id in Monero are actually used to insert data in the blockchain. Because of that, very low stealthiness is considered as it is a field which usually contains messages (e.g. (204),(232),(207))

On the other hand, some works use function arguments (e.g. (229), (203), (233), (234)). They do not hide the information in any way – data is exchanged using the corresponding argument type. For example, normal text is shared by using string arguments, hashes and keys by using bytes32, etc. Because of this, their stealthiness is considered low (BE*). Furthermore, their capacity usually depends

on how many arguments each function has and which type they are (recall Section 2.1.3), categorized as Argument-related limit (ArL).

Hiding information as a receiver address is a common way to disguise it (e.g. (16),(235)) as, in general, no transformation is applied to the data, so reading information from the address is trivial. For these reasons, although they do use a covert channel, their stealthiness is considered low.

Formatting the data as the value field in the transaction (e.g amount of transferred Ethers or Bitcoins) has also been used in (236). This is less common and often some transformation is applied previously to convert information to a suitable value. Their stealthiness is considered medium as the message is not easily readable, but usually a simple transformation from integer to hexadecimal allows to retrieve the message.

Whisper messages ((231)) are encrypted and private, however, all the members know there is a message being exchange even though they cannot decrypt it. Masked Authenticated Messages are a gossip protocol used in (42) that allows to publish encrypted messages on IOTA Tangle. An adversary is capable of lurking the messages even though the botmaster can fork the channel and establish a new encryption key. In (28) data is embedded on EmerDNS as a DNS record. Although it theoretically has a big capacity of insertion it should be a valid DNS for the protocol to work and it is completely in clear text. (237) uses a noise plugin on top of the Bitcoin Lighting Network. Private messages are included in the payload inside the onion packets that are routed over the hops until reaching the recipient, thus, the situation is very similar to whisper messages. Regarding (225) it is mentioned that DApp would be used, but without giving details in this regard. Finally, embedding the message in the nonce of the signature is first proposed by (211), and data is difficult to be distinguished from normal transactions. Furthermore, information is also obscured, so even if it is retrieved, it is difficult to extract the message. Thus, its stealthiness is considered high.

Table 3.5: Data protection and cost

Source	Max capacity (bytes)	Confidentiality	Field	Attacker model	Cost for the attacker	Current value (dollars)
(202)	80	No	OP_RETURN	BE*	0.008 btc	164.63
(25)	6	No	Receiver address	BE	0.10 btc	2057.89
(203)	64	No	Function arguments	BE*	0.002582 eth	3.79
(204)	80	No	OP_RETURN	BE*	Free	0
(229)	ArL	No	Function arguments	BE*	0.069062 eth	101.25
(210)	80	No	OP_RETURN	BE*	0.00000546 btc	0.11
(221)	20000	Obscured	Function arguments	BE*	0.0113092 eth	16.58
(211)	32	Obscured	Nonce of signature	ALL	Not known	-
(232)	80	Encrypted	OP_RETURN/Data field	BE*	0.00000226 btc	0.047
(238)	80	Encrypted	OP_RETURN	BE*	0.00000546 btc	0.11
(233)	ArL	No	Function arguments	BE*	Not known	-
(239)	20	No	Value & Receiver address	BE,AE	0.00000001 btc	0.0002
(222)	80	No	OP_RETURN	BE*	0.0000675 btc	1.39
(240)	ArL	No	Function arguments	BE*	0.0004725 eth	0.69
(219)	80	Encrypted	OP_RETURN	BE*	0.00000546 btc	0.11
(27)	2	No	Value	AE	Not known	-
(28)	20512	No	Service on top of blockchain (EmerDNS)	BE*	Message and chain dependant	-
(234)	ArL	No	Function arguments	BE*	Not known	-
(231)	64000	Encrypted	Service on top of blockchain (Whisper message)	BE*	Free	-
(223)	1	No	Value	AE	0.00000154 btc	0.031
(230)	80.3	No	OP_RETURN & signature	BE,AE	0.00008 btc	0.16
(207)	32	Encrypted	Payment id	BE*	0.00006 xmr	0.009
(237)	1300	No	Service on top of blockchain(Onion payload)	BE*	Free (Testnet)	-
(241)	+80 (bigger because Testnet)	Encrypted	OP_RETURN	BE*	Free (Testnet)	-
(42)	1300	Encrypted	Service on top of blockchain(MAM message)	BE*	Free	-
(225)	Not specified	Encrypted	Service on top of blockchain(DApp)	Not enough information	-	-
(242)	80	No	OP_RETURN	BE*	0.00000546 btc (downstream) /Free (Testnet, upstream)	0.11
(243)	80	No	OP_RETURN	BE*	0.00001897 btc	0.39
(244)	50000	Encrypted	OP_RETURN	BE*	Free (Testnet)	-

* simple hiding technique

3.3.2.5 Purpose

Different proposals use the blockchain with different goals in mind. The following categories are distinguished:

- Payments. Works use the blockchain as a way to send payments, generally to the attacker. There are 100 proposals with this purpose, such as (26), (205) or (212).

- C&C communication. The blockchain is used to control bots and send instructions to them, as well as some other C&C related communication, like registering bots or answering commands. (221), (211) and (232) use the blockchain as a C&C server. This happens in 20 proposals.

- Address discovery. The blockchain is used to provide the malware with new sources of information/commands, generally to new C&C servers after a take down ((219), (210)). 5 works use the blockchain to provide some kind of address discovery.

- Key distribution. The blockchain is used to distribute keys. For example, keys to decrypt data encrypted by a ransomware, i.e. (203) and (204). 3 proposals are included within this category.

- DNS. The blockchain is used as a DNS service in (28).

- Malware storage. Malicious software is stored in the blockchain in one proposal, namely (202).

3.3.2.6 Malware type

Malware type, together with the purpose, blockchain properties and elements are studied herein. Table 3.6 summarizes this data.

Different kinds of malware use the blockchain (recall Section 2.7). Most proposals study ransomwares (100 works), e.g. (26), (205), (25), followed by botnets (24 proposals), e.g. (211), (232), (222). Any type of malware (202) are barely relevant, with 1 work.

Concerning purpose, results show that when the malware is a ransomware, the blockchain is mostly used as a way to obtain payments (100 proposals). This happens for example in (26), (205), (25). Key distribution (3 works) is the second purpose most used for this kind of malware (203), (204), (229). C&C and Address discovery is the goal in 1 work, (229) and (25) respectively. Regarding botnets, the main purpose of the blockchain is working as a C&C server (19 proposals), i.e. (221), (211) or (232), followed by discovering addresses (4 works), i.e. (210), (238), (219) or (234). Besides, DNS is the purpose for 1 proposal (28). In (202) any type of malware uses the blockchain to store itself.

In terms of blockchain properties, ransomware mostly takes advantage of the blockchain's decentralization (100 works) and pseudo-anonymity (99 works, such as (26), (205) or (25)), and the same happens with botnets but this latter to a lesser extent (decentralization in 24 works and anonymity in 22 proposals). Moreover, in botnets, blockchains are sometimes applied due to their availability (22 works such as (221), (222) or (27)), in contrast to traditional systems, where servers can be easily taken down. By contrast, the paper that is suitable for any kind of malware uses blockchains to provide decentralization, availability and immutability

in complete and equal manner. As anything posted in the blockchain cannot be altered without a huge effort (recall Section 2.1) it ensures that different parts of the malware remain available and unalterable.

Regarding the blockchain elements used per type of malware, ransomwares mostly use transaction common fields (98 works). They also use program code in 2 proposals and data fields in 1. Botnets, on the other hand, use data fields in 11 of them. Program code are used in 4, transaction common fields in 3, the nonce of the signatures in 2 and services on top of the blockchain in 5. Any malware use the data field.

In terms of transactions flux, the majority of ransomware receives transactions from various victims to one address in 50 cases, i.e. (26) or (205); 46 proposals receive transactions from various to various attacker accounts, i.e. (213); and in 4 works various victims send transaction to attackers, but whether the attacker has a single address or multiple is unknown, i.e. (206) or (245). Ransomware generally does not send too many transactions to the system, except in some proposals, namely, (204) has a different attacker address sending transactions to different victim addresses, (25) only has one known attacker address that sends transactions to various victims, (229) from various to one and (203) from one to one, smart contract in this last case.

Botnets proposals, on the other hand, send more transactions (usually botnets commands) than receive them. In this case, 4 proposals receive transactions from various victims to one unique attacker address. For example, botmaster address (231) or smart contract address (221). (244) and (42) receive transactions from various to various and (223) from one to one. When sending transactions, most of them (9) send from one address to one address, being this address, for example, a managing smart contract or OP_RETURN transaction in which the commands are posted (e.g. (221), (232)). Moreover, 8 works apply one to various communications e.g. bots in (28) and (231), and 6 from various attacker addresses to various

clients/victims (e.g. (218) or (219)). In (225) the senders and receivers are not known.

Table 3.6: Malware type summary

		Ransomware	**Botnet**	**Any**
Purpose	Payments	100	0	0
	Key distribution	3	0	0
	Address discovery	1	4	0
	C&C	1	19	0
	DNS	0	1	0
	Malware storage	0	0	1
Blockchain properties	Decentralization	100	24	1
	Pseudo-anonymity	99	22	0
	Availability	4	22	1
	Immutability	3	1	1
Blockchain elements	Transaction common fields	98	3	0
	Data	1	11	1
	Nonce of signature	0	2	0
	Program code	2	4	0
	Services on top of blockchain	0	5	0
Sends transactions	From one to one	1	9	1
	From one to various	1	6	0
	From various to one	1	0	0
	From various to various	1	8	0
	Not known	0	1	0
Receiving transactions	From various to one	50	4	0
	From various to various	46	2	1
	From various to not known	4	0	0
	From one to one	0	1	0
	Not known	0	1	0
Total		100	24	1

3.3.2.7 Blockchain technology

Most proposals use Bitcoin (110 works) such as (205) or (25). Ethereum is the second most used one, with 8 works, i.e. (229) or (221). Dash ((212), (232)) is used in 2 works and Monero ((207),(206)) in 4, while Bitcoincash, Emercoin and Litecoin, NKN and IOTA are applied in a single proposal ((42; 225; 232)). On the other hand, in (211) any type of blockchain technology can be used.

Blockchain technologies in relation to malware purposes, type and blockchain properties and elements are analysed. Table 3.7 shows a summary.

In Bitcoin, the blockchain is mostly used for payment purposes (95 proposals), such as in (26) and (205). Moreover, blockchain is also used as a C&C server in 10 proposals (e.g. (232), (218)), to provide addresses to the system in 5 works (e.g. (210) or (238)) and just in one proposal Bitcoin is used for key distribution (204)

and malware storage (202). In contrast, Ethereum is applied for payment purposes and key distribution in 2 proposals ((203), (229)), while it is significantly used as a C&C server (7 works), probably because of the possibility of using smart contracts (recall Section 2.1), e.g. (221) or (233). Besides, Dash and Monero are equally used as a mean to pay (1 and 2 proposals) (206) and to establish a C&C server (1 work each) (207). Emercoin is used to provide a DNS service by using EmerDNS (Application on top of blockchain) (28). LitecoinBitcoincash, IOTA and NKN, as well as (211) in which any technology can be used, are exclusively applied as C&C server.

Regarding desired properties, those works which use Bitcoin look for the provision of decentralization and pseudo-anonymity (110 and 108 works respectively). However, availability is demanded in 16 proposals and immutability in 2 of them. Similarly, Ethereum main purposes are to provide decentralization (8 works), as well as pseudo-anonymity (6 works), availability (7 proposals) and immutability (3 works), although this latter to a lesser extent. In the case of Monero, real anonymity is provided in 4 works, as well as decentralization and availability just in one. Dash, IOTA, EmerCoin, Litecoin and Bitcoin cash just focus on decentralization and pseudo-anonymity provision. NKN, besides decentralization and pseudo-anonymity also provides availability.

Used blockchain elements and technology are also jointly studied. In Bitcoin, common transaction fields are used in the majority of proposals (98 of them), while the data field is used in 12 and the nonce of the signature just in 1, as well as application on top of blockchain. In Ethereum, however, most of the works use program code (6 proposals) and data field and services on top of blockchain are used just in 1 each. Dash both use the data and transaction common fields equally (1 work), while Monero uses transactions in 2 and the data field in 1. Emercoin, NKN and IOTA use services on top of blockchain and Litecoin and BitcoinCash both use data fields.

By type of malware, Bitcoin is mostly used by ransomwares (95 proposals), followed by botnets (14 works). Ethereum, on the other hand, is mostly used for botnets (6 works) and ransomware (2 works) cases. Dash and Monero are the least relevant as they are applied in only one proposal for ransomware and botnet in the case of Dash and 2 for rasomware and one for botnet in the case of Monero. The rest of technologies are used for botnets.

Table 3.7: Blockchain technologies summary

		Bitcoin	**Ethereum**	**Dash**	**Monero**	**Bitcoincash**	**Emercoin**	**Litecoin**	**IOTA**	**NKN**	**Any**
Total		110	8	2	4	1	1	1	1	1	1
Purpose	Payments	95	2	1	3	0	0	0	0	0	0
	Key distribution	1	2	0	0	0	0	0	0	0	0
	Address discovery	5	0	0	0	0	0	0	0	0	0
	C&C	10	7	1	1	1	0	1	1	1	1
	DNS	0	0	0	0	0	1	0	0	0	0
	Malware storage	1	0	0	0	0	0	0	0	0	0
Blockchain properties	Decentralization	110	8	2	4	1	1	1	1	1	1
	Pseudo-anonymity / Anonymity	108	6	2	4	1	1	1	1	1	1
	Availability	16	7	0	1	0	1	0	0	1	1
	Immutability	2	3	0	0	0	0	0	0	0	0
Blockchain elements	Transaction common fields	98	0	1	2	0	0	0	0	0	0
	Data field	12	1	1	1	1	0	1	0	0	0
	Nonce of signature	1	0	0	0	0	0	0	0	0	1
	Program code	0	6	0	0	0	0	0	0	0	0
	Services on top of blockchain	0	1	0	0	0	1	0	1	1	0
Malware types	Ransomware	95	2	1	3	0	0	0	0	0	0
	Botnet	14	6	1	1	1	1	1	1	1	1
	Any	1	0	0	0	0	0	0	0	0	0

3.3.2.8 Cost for the attacker

The attacker usually has to assume some cost, specially in those cases in which transactions are sent to the blockchain because they contain some kind of information.

Table 3.5 shows the different costs incurred in each case. In order to compute the cost, the mean (average) of the value of each coin from the last three months (from December 10, 2022 to March 10, 2023) has been considered: 1 ether is $1,466.03 (246), 1 xmr (Monero) is $157.907 (247) and 1 btc is $20,578.9 (248). For those which use OP_RETURN method and do not indicate the cost, the cost of the minimum transaction value is considered (249).

Even considering previous costs, most proposals are relatively cheap, with the exception of (25) which spends 0.10 btc ($2,057.89 currently) and (229) which spends 0.069062 eth ($101.25). There are some of them, which are actually free, such as (204), where a fee to the victim is asked first and the transaction the attacker sends reuses this money, so there is no extra cost for the attacker. Others are free because they use a testnet (237; 244). However, testnets can be deprecated and moved to read only, leading to unusable malware. For example, (244) uses the Bitcoin testnet, which, among other things, is cost free as Bitcoin can be obtained from public faucets (an app or a website that distributes small amounts of cryptocurrencies as a reward for completing easy tasks) (250). Moreover, (231) applies a service on top of the Ethereum blockchain, named Whisper. This service does not send traditional transactions and exchanging messages is always free.

As a result, the benefits of using blockchain are higher than the cost for attackers. For example, Cerber (25) ransomware generated $2.3 million in annual revenue. Profit when using botnets is also high, for example Methbot was making $3-5 million a day on its peak (251). Indeed, given that traditional attacks may also involve some cost (252), e.g. for using or maintaining servers (an hour-long DDoS attack using a cloud server will cost criminals $7 (253), the use of blockchain seems to be affordable.

3.3.3 Summary of the analysis

This Section presents the main findings of the analysis (Section 3.3.3.1) and the identified open research issues (Section 3.3.3.2).

3.3.3.1 Lessons learned

According to the data study, the main findings are:

- Bitcoin is the most used technology. It is followed by Ethereum as the second most common technology.

- Malware proposals do not use the full potential of the blockchain. Some level of decentralization is always provided but some characteristics which are intrinsic to the blockchain like immutability, availability or anonymity are not used at full in most cases.

- Most seed addresses are hardcoded. If the address is flagged or the private key is lost or stolen the malware becomes useless.

- Blockchain transactions are preferred as the way for malwares and blockchains to interact. Very few use smart contracts or other services on top of the chain.

- There is not much exchange of information on the chain. The blockchain is barely used to share data and when applied, data is almost never hidden or covert. When used, it is typically for sharing command information to a botnet.

- Blockchain primary use is for payments. In relation to the previous point, blockchain is mostly used as a mean to provide payments to the attacker by a high majority of proposals.

- The most common type of malware is ransomware. This is related to the previous point, as it is used to pay ransoms.

- It is cheap to use the blockchain. The cost for the attacker is usually cheaper using the blockchain compared to traditional attacks, which also involve some costs. For example, a DDoS attack using botnets is estimated to cost from $50 to several thousand dollars in the case of a 24-hour operation (254).

3.3.3.2 Research gaps

Based on the previous analysis, the following research gaps have been identified:

- Covert channels are barely used. This entails that communications between attackers and victims are relatively easy to identify. Although their contents

are not always understandable due to encryption, the existence of the communication can be discovered.

- Smart contracts have been applied in malware, but not used for sophisticated covert channel purposes. Information is usually formatted as the corresponding arguments types in functions in the contracts, allowing trivial data retrieval. Furthermore, high level languages like Solidity have not been used to insert information.

3.3.4 Mitre ATT&CK

This thesis focuses on how malwares can leverage blockchains to increment their malicious capabilities. Therefore, there are parts and characteristics of a malware like how it infects a system or propagates that are out of scope. Most tactics of MITRE ATT&CK matrix are not really applicable to this study because they are related to how the malware works in relation with the victim and not the own malware infrastructure. However, the following couple of tactics and techniques, pointed out by their identifier, are linked to this proposal:

- **Command and Control (TA0011)**. Blockchain has been used to provide commands to bots (232). (211) provides some level of data obfuscation (T1001) by means of steganography (T1001.002) to share commands. (232), among others, provides an Encrypted channel (T1573). Blockchain is used also as a way to achieve Fallback Channels (T1008), as it is used to indicate the new name of the server to connect to bots (238) or to indicate the new address of the C&C channel (211). Furthermore, (202) uses the blockchain to store different malware parts. Although they do not seem to use a C&C to download the different parts of the malware, a similar technique could be used to provide tools and files, which could also allow "Ingress Tool Transfer" (T1105). It could be also argued whether they provide Web Service (T1102) because blockchain technology can be used as a way to hide the noise that

commands to the bot may produce. Due to this fact, in (241) some bots do not directly connect to the blockchain but to a blockchain explorer (a web service).

- **Exfiltration (TA0010)**. Some of the botnets allow bidirectional communication (241) with the intention of the bots to send information in response of the commands over the blockchain. In this case, the "Exfiltration Over C2 Channel" (T1041) technique is applied.

Besides those tactics and techniques that Mitre matrix presents, we consider that blockchain provides some interesting features not included in ATT&CK related to, for instance, a new tactic called 'the management of the infection' in which the following techniques could be included:

- **Resilience against domain/C&C take down**. Blockchain is a distributed network. Therefore, its nature makes difficult to take down C&C servers and then, the level of danger of a botnet increases.
- **Payment infrastructure**. Blockchain facilitates anonymous payments, therefore the possibility of identification of, for example, an attacker controlling a ransomware is more difficult.
- **Attacker-victim communication channel**. Blockchain allows to insert arbitrary data, which could facilitate, among other issues, the delivery of keys to a victim after the payment of a ransom.

3.3.5 Related work

There are some works related to blockchain and malware. Financial issues are significantly considered, (31) studies the effect of cryptocurrencies on ransomware and what could influence a victim to pay. Complementary, (255) explains how the blockchain is used for payment and key delivering, also describing how modern ransomware works and possible defenses. Moreover, (220) provides a comprehensive,

evidence-based picture on the global direct financial impact of ransomware attacks. They empirically analyze Bitcoin transactions related to 35 ransomware families. Also in the economic field, (256) presents a comprehensive study on all recent ransomware and reports their economic impact from the Bitcoin payment perspective. Besides, in (257) a measurement framework is developed for performing a large-scale, two year, end-to-end measurement of ransomware payments, victims, and operators.

On the other hand, (258) studies the ability of Bitcoin to store metadata and shows basic approaches to improve blockchain privacy. It identifies and classifies blockchain transactions embedding metadata of major protocols running on top of Bitcoin. It also exposes the possibility of using stealthy addresses, as well as the use of smart contracts to automate ransom payments.

In the field of botnets, (32) provides a comprehensive systematization of the state of the art of blockhain-based-botnets, along with an abstract model of such system.

Table 3.8 shows a comparison between previous works and our study. None of existing proposals analyses in depth how the blockchain is used. They do not study desired properties, the different elements of the blockchain being used or how the information is exchanged in the system. They usually focus on one technology (namely Bitcoin) and ransomware. By contrast, our study analyses, from a cybersecurity point of view, different technologies and malwares, as well as other issues like malware purposes and the cost for the attacker.

Table 3.8: Malware in blockchain study

	Study					
Proposal	**Elements used**	**Desired properties**	**Data exchange**	**Cost for the attacker**	**Blockchain technologies**	**Types of malware**
(31)	X	X	X	X	Bitcoin	Ransomware
(255)	X	X	√	X	Bitcoin	Ransomware
(220)	X	X	X	X	Bitcoin	Ransomware
(258)	√	X	√	X	Bitcoin	Ransomware
(256)	X	X	X	X	Bitcoin	Ransomware
(257)	X	X	X	X	Bitcoin	Ransomware
(32)	√	√	√	√	All	Botnet
ours	√	√	√	√	All	Botnet, Ransomware, Worm

Chapter 4

Zephyrus: An information hiding mechanism leveraging Ethereum data fields

---

4.1 Summary of the chapter

In Section 3.2.3 of the previous chapter it has been stated that there is a prevalence of Ethereum as technology. Moreover, in Section 3.3.3.2 we discover that covert channels are barely used. Because of this, in this Chapter we propose Zephyrus, a steganographic tool for Ethereum.

In Section 4.2 we describe the model on which the system will be based in order to construct the mechanism. In Section 4.3, we share the results of a preliminary study of real Ethereum data in order to better stablish the base parameters for the mechanism. Based on those results, the mechanism is proposed in Section 4.4. After that, in Section 4.5 a evaluation of Zephyrus is shown. In Section 4.5.2.2, a summary of all related work regarding steganography and blockhain is explained and the proposed mechanism compared with the related work.

It should be noted that, for the sake of completion, the use cases used for the analysis of the feasibly of this mechanism are those exposed on Section 1.1, being: *panic button case*, *sabotage case* and *censorship case*. Nonetheless, Zephyrus could be also be leveraged by malware. This use case, however, will be further explored in Chapter 5 with Smart-Zephyrus.

4.2 Model

The model consists of the description of the involved entities and attackers (Section 4.2.1), goals at stake (Section 4.2.2) and working assumptions (Section 4.2.3).

4.2.1 Entities and attacker model

In this steganographic system two entities are identified, namely sender and receiver, communicating through a channel – the Ethereum blockchain. While the sender transmits information, the receiver is merely an observer of the blockchain.

The attacker model will be the same used in Section 3.3.2.4.1.

4.2.2 Goals

The development of a steganographic mechanism should be designed to be resilient against any kind of suspicion. In this regard, the following goals are identified:

- **Stealthiness**: embedded messages should be difficult to identify for an attacker.
- **Simplicity**: any user who is able to interact with the Ethereum blockchain should be able to use the mechanism.
- **Efficiency**: the mechanism should be efficient in terms of time and amount of sent information. It should allow sending a practical amount of information in an affordable amount of time.
- **Cost**: sending hidden information should be economically affordable for the sender.
- **Secret integrity**: hidden information's integrity should remain over time.

4.2.3 Working assumptions

The following assumptions are considered in this proposal:

- The receiver knows the following data items to get access to the secret:
  - The first transaction identifier. Sending a secret may involve several transactions, but knowing just the first identifier should be enough to retrieve the whole message.
  - Cryptographic materials, that is, the encryption key and a random number called *nonce*.
  - Fields in which the secret is embedded.
- Secrets are sent sequentially, thus avoiding sending simultaneous messages.
- The sender uses the same source or destination Ethereum address for a given secret.

4.3 Preliminary Ethereum data study

Since *Zephyrus* aims to achieve stealthiness, transactions and contracts including secrets must mimic existing ones. For this purpose, we have analysed 16,942,215 transactions and 65,346 contracts. In order to reflect the evolution of transactions over time, we have considered one week every six months from 2017 to 2019. In particular, transactions are collected between March 24th and April 1st, and between September 24th and October 1st each year.

Once collected, transactions have been classified into three categories: sent to another blockchain user (8,998,787 transactions), to a function in a contract (7,943,428 transactions) and to deploy a contract (65,346 transactions). It should be noticed that among those that deploy contracts, 4,736 of them include constructor arguments when deployed and 58,644 presents the last JUMP-JUMPDEST block structure (recall Section 2.5) . The proportion is in line with expectations, as transactions among Ethereum accounts are the most prevalent ones.

Since the secret is embedded in one or more transaction fields, or as part of the contract code, the analysis is carried out for each one independently. It must

be noted that some fields are freely set by the user whereas others are the result of a cryptographic operation (e.g., hash function). Therefore, the techniques to characterize each field are different. In the former case the variability of each field is analysed by using statistical measures (Section 4.3.1). In the latter, entropy is studied because randomness is an essential cryptographic property (Section 4.3.2).

Note that the *Data* field has not been analysed for all transactions. Using this field to insert a secret would raise suspicions. However, function arguments and contract information (which are contained in this field for transactions related to contracts) have been characterized as they could potentially be used for covert communications.

4.3.1 Variability

Intuitively, a high variability of a given data field is beneficial for the sake of embedding secret information. Otherwise, if a given data field always has the same value, any alteration would easily be noticed. Thus, determining the variability of a field requires analysing the amount of different values, and their statistical distribution with respect to all potential values. Several metrics have been considered, namely the coverage of the value range, the mean and standard deviation of the amount of appearances per value, and the prevalence of the most frequent values for each field. Concerning coverage, it must be noted that the amount of collected transactions is usually smaller than the range size. Therefore, the minimum between these two factors will be considered. With respect to prevalences, the accumulated frequency for the 8 and 16 most frequent values is computed. Table 4.1 summarizes the analysis.

There are some fields that exhibit a suitable variability. For example, *Value* in transactions to another user, shows a reasonable degree of homogeneity. On the contrary, some fields (such as *Gas limit* to other users or *Value* to functions) are discarded as only two values account for the vast majority of cases (see Table A.13

Table 4.1: Characterization per data field

Field	Type	Number of different values	Min(# transactions, possible values)	Coverage (%)	Mean appearances	St. dev. appearances	Top 8 accum. freq.	Top 16 accum. freq.	Selected
Value	to user	5633180	8998787	62.60%	1.60	93.82	5.75%	8.11%	✓
	to function	165939	7943428	2.09%	47.87	18186.83	95.38%	96.01%	X
	contract	84	65346	0.13%	777.93	7069.38	99.87%	99.89%	X
Gas Price	to user	69032	8998787	0.77%	130.36	6928.91	48.68%	**65.20%**	✓
	to function	61139	7943428	0.77%	129.92	5294.73	39.96%	**57.98%**	✓
	contract	9007	65346	13.78%	7.26	117.42	40.12%	**57.13%**	✓
Gas Limit	to user	5713	8998787	0.06%	1575.14	57930.74	86.70%	92.24%	X
	to function	133700	7943428	1.68%	59.41	3966.78	11.30%	16.25%	X
	contract	6322	65346	9.67%	10.34	170.77	**52.48%**	72.39%	✓
Function arguments	uint256	3810193	11756671	32.41%	3.09	325.47	12.22%	16.21%	✓
	bytes32	1792159	2135859	83.91%	1.19	8.69	0.87%	1.09%	✓
Constructor arguments	uint256	607	2328	26.07%	3.84	12.46	31.40%	42.23%	✓
Bytecode	PUSH1	53	256	20.70%	1754.60	5167.38	**98.09%**	99.47%	✓
	PUSH20	17	16275	0.10%	957.35	3937.99	99.94%	99.99%	X

in the Appendix).

For those fields which do not have such a variability but cannot be discarded either, the accumulated frequency of the top 8 and 16 elements has been studied. If that frequency is beyond 50%, a given set of values are frequent, so they could also be used to represent a secret. This happens, for example, for the *Gas price* field in bold in Table 4.1.

The analysis of function and constructor arguments requires special handling, as it is necessary to study each type of argument independently. For simplicity, the most common types are considered herein (see Table 4.2). In the case of *Function arguments*, they are "uint256", "address" and "bytes32", which together cover 92.25% of transactions (adding Function arguments percentages from Table 4.2). In the case of *Constructor arguments*, we focus on "uint256" and "address", which account for 76.11% of cases (adding both percentages of Constructor arguments from Table 4.2). The third most common type, "string", has not been considered as it is usually human-readable.

Table 4.2: Most common types of arguments and their coverage

Field	Type	Coverage (%)
Function arguments	uint256	45.88%
	address	38.04%
	bytes32	8.33%
Constructor arguments	uint256	17.90%
	address	58.21%
	string	16.82%

The analysis of the values contained in these argument types reveals one interesting feature. Particularly, some fields show a prevalent pattern a number ending with a variable amount of consecutive zeros. This is the case of uint256 type for *Function and Constructor arguments*, as well as the *Value* field. For the sake of illustration, 76.36% of transactions to users show this pattern in *Value* field (see Table 4.4). Table 4.3 shows the most prevalent patterns. For each one, patterns containing more non-zero digits are regarded as suitable, as they show nice coverage and homogeneity. The only exception is in the *Value* field, due to economic issues explained later (see Section 4.4.2).

Table 4.3: Analysis of patterns

Field	Type	Length	Zeros	Number of different values	Min (# transactions, possible values)	Coverage (%)	Mean appearances	St. dev. appearances	Selected
Value	to user	18	10	594039	614569	96.66%	1.03	0.76	X
		18	12	109550	202425	54.12%	1.84	22.9	X
		17	10	384834	469609	81.95%	1.22	4.64	✓
		17	16	9	9	100.00%	22712.18	26543.6	X
Function arguments	uint256	22	18	8075	8100	99.69%	22.675	50.7	✓
		22	21	9	9	100.00%	19619.22	23869.9	X
		21	18	810	810	100.00%	250.9	442.44	X
		21	20	9	9	100.00%	23318.33	20027.3	X
Constructor arguments	uint256	10	2	103	148	69.59%	1.43	0.84	✓
		10	9	6	9	66.67%	22.66	44.36	X
		1	0	8	9	88.89%	24.75	31.36	X

Table 4.4: Fields with highest prevalence of patterns

Field	Type	% of values ending in zero(s)
Value	To user	76.36%
Gas price	To user	95.09%
	To function	96.47%
	Contract deployment	85.28%
Gas limit	To user	97.09%
	To function	71.64%
	Contract deployment	75.14%
Function argument	uint256	67.84%
Constructor argument	uint256	82.95%

Beyond patterns, the address argument is considered a crypto-related field. Concerning bytes32, given the low mean and standard deviation (see Table 4.1), as the lack of patterns, it is also studied as crypto-related (see Section 4.3.2).

Last but not least, the *Bytecode* field requires a tailored analysis. In particular, it

is relevant to characterize the amount of instructions, their frequency and the value of their arguments, if any. In all cases, we focus on the last JUMP-JUMPDEST block (recall Section 2.5), as it is the region that can be altered with lower risks. Concerning the amount of instructions, 9 and 13 are selected as they are the biggest amounts among the most common ones (see Table A.12 in the Appendix). POP and PUSH1 are the most common opcodes, covering 39.87% of the cases (see Table A.11 in the Appendix). However, as there is room for more variable instructions, the 20 most used opcodes are chosen. Among them, just the variability of PUSH1 and PUSH20 is studied because the remaining opcodes do not use bytes as parameters. PUSH20 is discarded due to its high cost and low variability (see Table A.13 in the Appendix). Regarding the values for PUSH1, the 8 most common values are selected, covering 98.09% of the sample (see Table 4.7). Finally, the 4 most common pairs of instructions after the JUMPDEST and before the JUMP are selected (see Tables 4.5 and 4.6). They cover 62.91% and 76.35% of the sample respectively. It is worth to mention that the code in this block ends with one or more POPs in 60.26% of the cases.

Table 4.5: Top 4 pairs of instructions after JUMPDEST

Instructions	Contracts	Coverage (%)
[POP, POP]	22525	38.41
[PUSH1, SLOAD]	8967	15.29
[PUSH1, DUP1]	3353	5.72
[PUSH1, PUSH1]	2047	3.49

Table 4.6: Top 4 pairs of instructions before the JUMP

Instructions	Contracts	Coverage (%)
[POP, POP]	32009	54.58
[AND, DUP2]	9262	15.79
[SWAP1, SSTORE]	2779	4.74
[POP, SWAP1]	724	1.23

Table 4.7: Top 8 values for PUSH!

Value	Quantity	Frequency(%)
00	26170	28.14%
01	19245	20.69%
02	14758	15.87%
a0	12092	13.00%
40	8736	9.39%
20	6848	7.36%
ff	2960	3.18%
05	408	0.44%

4.3.2 Entropy

Entropy has been computed, combined and individually, in those fields that are the result of cryptographic operations and those meant to represent binary information. The calculus of the individual entropy involves computing Shannon entropy per value in each field, normalized from 0 to 1 (259). By contrast, combined entropy is calculated concatenating all values per field and computing Shannon entropy. In this way, a high individual entropy ensures random value fields and a high combined entropy guarantees that value fields are different among transactions. Moreover, in both cases the mean and standard deviation are also computed.

Table 4.8 presents the results of the analysis. Concerning hashes, namely *Receiver address* and *Swarm hash*, we have computed entropy for all studied transactions. Furthermore, we have generated multiple hashes to compare their entropy. This allows us to reason about the possibility of generating hashes with the same or similar entropy to avoid suspicions. Their high entropy with low standard deviation support the uniqueness of the values and their possible use for embedding purposes.

4.4 Proposed mechanism

This Section presents *Zephyrus* by introducing both the embedding and revealing procedures of hidden messages for all transaction fields. Given the different nature of the fields at stake as well as their value distribution (recall Section 4.3), several

Table 4.8: Entropies per field

Field	Subfield	Combined entropy	Mean of entropies	Standard deviation of entropies
Sender address	to user	1.00	1.00	0.00
	to function	1.00	1.00	0.00
	contract	1.00	1.00	0.00
Standard deviation of entropies				
Receiver address sample		1.00	0.99	0.01
Receiver address generated		1.00	1.00	0.01
Swarm hash		1.00	1.00	0.00
Swarm hash generated		1.00	1.00	0.00
Function arguments (filled)	address	1.00	0.99	0.01
	bytes32	1.00	0.98	0.10
Constructor arguments (filled)	address	1.00	1.00	0.00
r	to user	1.00	1.00	0.00
	to function	1.00	1.00	0.00
	contract	1.00	1.00	0.00
s	to user	1.00	1.00	0.00
	to function	1.00	1.00	0.00
	contract	1.00	1.00	0.00
Transaction id	to user	1.00	1.00	0.00
	to function	1.00	1.00	0.00
	contract	1.00	1.00	0.00
Public key	to user	1.00	1.00	0.00
	to function	1.00	1.00	0.00
	contract	1.00	1.00	0.00

embedding strategies are firstly proposed in Section 4.4.1.

It must be noted that for the covert communication to take place, the mining procedure must be carried out (260). However, it is out of the scope of this Section as it is the regular process for every Ethereum transaction. For the sake of brevity, the embedding procedure does not describe the potential retransmissions needed if a transaction is not included in the blockchain. The notation used in the remainder of this proposal is shown on Table 4.9.

Table 4.9: Notation. Cost and capacity-related symbols (left). Cost magnitudes (right)

Symbol	Description
T_{Nc}	Transaction nonce
$\|S_0\|$	Amount of bytes '0' in the embedded message
$\|S_{n0}\|$	Amount of bytes not '0' in the embedded message
$\|B_{Ct}\|$	Amount of bytes in contract code
$\|B_{Ct0}\|$	Amount of bytes '0' in contract code
$\|B_{Ctn0}\|$	Amount of bytes not '0' in contract code
S	Secret to embed
LBB	Limited By Balance
$\|S\|$	Length of the secret
$LBGL$	Limited By Gas Limit
ML	Memory Limit
ArL	Argument-related limit
$\|IdF_0\|$	Amount of bytes '0' of function identifier
$\|IdF_{n0}\|$	Amount of bytes not '0' of function identifier
N_F	Number of operations in a function
StL	Storage Limit

Symbol	Description	Cost in gas (see (72))
C_{Tr}	Baseline transaction cost	21,000
C_{Ct}	Contract creation cost	32,000
C_{CtCd}	Cost per byte of contract code	200
C_{B0}	Cost per byte '0' of data or code	4
C_{Bn0}	Cost per byte not '0' of data or code	68
C_F	Cost per operation in a function	Variable
C_{St}	Cost per contract storage	20,000

Table 4.10: Embedding strategy per selected field

		S1 (full field)	S2 (top values)	S3 (pattern-based)	S4 (instruction encoding)
Addresses	Sender	x			
	Receiver	x			
	Contract	x			
Transaction info	Value			x	
	Gas limit (to contract only)		x		
	Gas price		x		
	Signature: r	x			
	Signature: s	x			
	Identifier	x			
	Sender public key	x			
	Function args: type uint256			x	
	Function args: type address	x			
	Function args: type bytes32	x			
Smart contract info	Swarm hash	x			
	Bytecode: PUSH1 values		x		
	Bytecode instructions				x
	Constructor args: type uint256			x	
	Constructor args: type address	x			

4.4.1 Embedding strategies

According to the previous analysis four different embedding strategies are identified. Table 4.10 summarizes the strategy applied for each data field. It must be noted that not all fields for all transaction types are considered. For example, not all types of *Function arguments* are selected. Similarly, the *Gas limit* field is used in transactions related to contracts.

On the one hand, crypto-related fields (such as *Receiver addresses*) and and those without patterns (bytes32 and address type) have been shown to have high entropies. Since the embedding mechanism will encrypt the secret (as explained later), the result exhibits high entropy as well. Indeed, this happens for the whole secret and for each individual fragment. Therefore, these fields are used in full (strategy S1).

On the other hand, strategy S2 is applied over those fields which count on acceptable variability, but in which a subset of *numval* values are prominently common. Such values are used for embedding purposes, though the amount of them depends on each field. This leads to a capacity given by Equation 4.1. For example, if $numval = 8$, 3 bits can be embedded.

$$Capacity_{S2}(bits) = \lfloor \log_2 numval \rfloor \tag{4.1}$$

Strategy S3 is applied in fields with acceptable variability and exhibiting some patterns in their values. In this case, the embedding operation uses these patterns to ensure that the result seems legitimate. Based on our observations, patterns are formed by a prefix and a suffix. Prefixes are formed by a set of digits ending in any number but 0. Suffixes are a sequence of z zeros. Therefore, for a value of total length l, the capacity of this strategy is given by Equation 4.2. For instance, for values of length 17 ending with 10 zeros, 22 bits can be embedded.

$$Capacity_{S3}(bits) = \left\lfloor \log_2(81 \times 10^{l-z-2}) \right\rfloor \tag{4.2}$$

Last but not least, a bytecode-specific strategy S4 is also proposed. As opposed to the previous ones, S4 does not consider the values of the data fields, but the set of instructions contained in the bytecode. Therefore, it provides with variable capacity, as it is explained in the following.

4.4.2 Embedding procedure

The embedding process starts by preparing the secret to make it suitable for Ethereum transactions. Afterwards, data is hidden in fields according to their size and type. The capacity of each field per transaction (summarized in Table 4.11) is studied, as well as the applied embedding strategy selected according to last column of Table 4.1 and highlighted in bold in case of S2, and Table 4.3.

Note that embedding operations, regardless of the field, are limited by LBB and $LBGL$. LBB refers to the fact that the sender's balance should be bigger than the cost of sending the transaction (incluiding deploying a contract or calling functions). By contrast, $LBGL$ refers to the maximum block gas limit, which depends on the network at stake – no transaction can surpass this limit (261; 262).

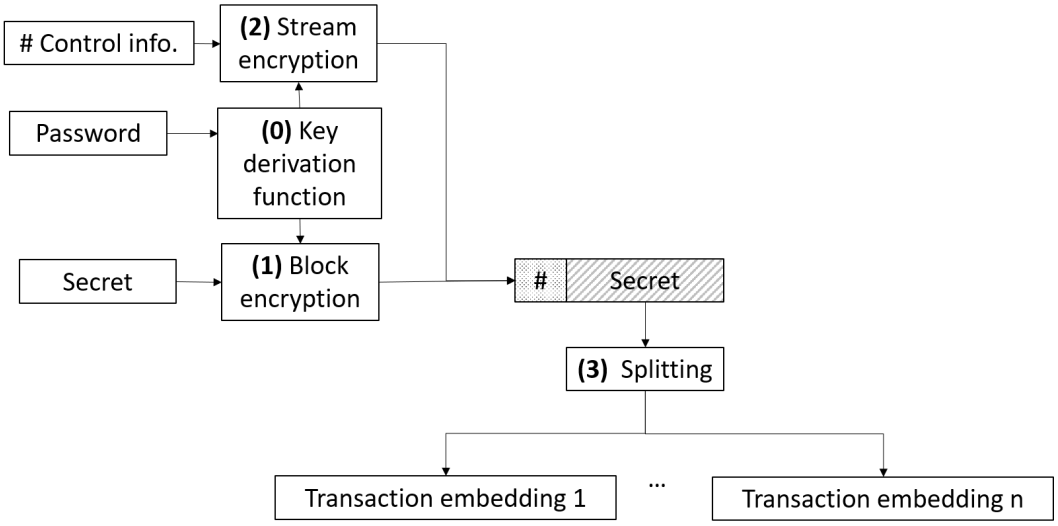

Figure 4.1: Secret preparation process

4.4.2.1 Secret preparation

The preparation process is depicted in Fig. 4.1. A key generation function is used to generate keys for the encryption processes (step 0). Firstly, the secret is symmetrically encrypted (step 1). In this process, the secret is adjusted by including encrypted control information. This is essential in the revealing process. In particular, the message length is necessary to distinguish between the secret itself and padding information. Moreover, additional data should be included for *Executable bytecode* (as explained in Section 4.4.2.2). Secondly, control data is also encrypted but with a stream cipher to keep the resulting size at a minimum (step 2). To randomize the output, the nonce from the last existing sender's transaction is also taken as input for this cipher. Finally, the secret is split if it exceeds the capacity of the transaction fields at stake (step 3).

4.4.2.2 Data hiding

For the sake of clarity, the description of the hiding process is divided into three main blocks, namely addresses, transaction information and smart contract data.

4.4.2.2.1 In addresses

The three types of addresses (namely *Sender, Receiver* and *Contract* ones) can be modified in all cases, thus S1 strategy is applied. However, the required computational effort is dramatically different.

Recalling that the *Sender address* is the hash of a public key, the embedding process is limited by the computation of a valid inverse (i.e., private key) according to the cryptographic algorithm at stake (in particular, secp256k1 (72; 263)). Consequently, embedding data in this field involves a trial-and-error procedure.

A similar situation happens with *Contract addresses*. Since they are computed considering the number of transactions sent by the contract creator (recall Section 2.5.2), a trial-and-error process is carried out to find a suitable number.

On the contrary, the *Receiver address* is not under any restriction. Therefore, it can be modified at will.

4.4.2.2.2 In transaction information

Capacity and effort to do the embedding varies greatly among fields.

The *Value* field can be used considering its underlying patterns (strategy S3). However, it is limited by LBB as the secret is represented as the payment amount. In this case, only values of length 17 and 10 ending zeros will be considered as a trade-off between capacity and cost. Note that value to functions and contracts is discarded because the top 2 values (though for simplicity not presented in Table 4.1) represent more than 90% of the sample. Thus, it would allow a very small capacity.

On the other hand, the most prominent *Gas limit* and *Gas price* values (strategy S2) are considered for representing the secret. In this case, their use is bounded by LBB, and also by $LBGL$ in *Gas limit*. These limitations depend on the sending account and the Ethereum blockchain, respectively. In the same line as *Value field* to functions and contracts, *Gas limit* to users is discarded because the top 2 values cover more than 62% of the sample.

As opposed to the previous field, signature values r and s and the *Sender public key* can be used in full (strategy S1). Furthermore, there is no technical limitation for the secret. However, a trial-and-error process must be followed to find the right cryptographic materials and produce a value that represents the fragment of the secret at stake.

4.4.2.2.3 In smart contracts

Depending on the field, a different embedding strategy is used, specially when bytecode is at stake.

Swarm hash field can be used in full (strategy S1) and with no limitations, since block scanners such as Etherscan do not currently check its value.

Function arguments appear within function calls or in a *Contract constructor*. Each function receives a different number of arguments and of varied types. In practice, the capacity is limited by LBB, $LBGL$ and the technical limit for each argument type (called ArL). For instance, uint256 corresponds to 32 bytes and uint8 to 1 byte (264). As it was stated in Section 4.3.1, only uint256, address and bytes32 types are used to embed information in *Function arguments* and uint256 and address types for *Constructors arguments*. In address and bytes32 types the whole capacity (S1) is used, while uint256 type follows a pattern (strategy S3).

With respect to the bytecode, there are two limitations in this regard – the code should look like a valid set of instructions and it has to be well-formed. In particular, two alternatives can be chosen – including instructions to represent the secret in an unreachable part of the code (called *Non-executable bytecode*) or in a reachable one (called *Executable bytecode*). Thus, *Non-executable bytecode* is placed between the JUMP-JUMPDEST block and the STOP/INVALID instruction (recall Section 2.5.2). The code added in that region is never executed by any function of the contract. However, this should look like a legitimate JUMP-JUMPDEST block, so starting and ending instructions should follow the regular distribution.

The second way, *Executable bytecode*, involves including instructions in the JUMP-JUMPDEST block. It requires managing instructions carefully to keep the state of the stack and cause a failure. Therefore, the stack should be correctly restored.

In order to encode the secret, two strategies are followed. On the one hand, the choice of instructions (strategy S4) – the 20 most used opcodes (Table A.11 in the Appendix) are divided in a couple of sets, one to represent 0 and another to represent 1. Thus, one opcode is chosen on a random weighted way. On the other hand, the argument of PUSH1 follows strategy S2.

In both cases, the capacity of the bytecode is limited in practice by EVM's total memory (ML), as well as LBB and $LBGL$. Moreover, the amount of instructions

to be inserted is limited by the usual size of the JUMP-JUMPDEST block (9 and 13 instructions, recall Section 4.3.1). It must be noted that in the *Executable bytecode* case some instructions are needed to restore the stack. Therefore, they do not convey the secret themselves. As a result, in the *Executable bytecode* the capacity is limited by the number of secret-related opcodes applied. They are all instructions except for PUSH1, used to control the stack (if any), and the final POPs at stake. However, arguments of PUSH1 instructions are still used to embed information. Since the amount of secret-related instructions is not known in advance by the receiver, such information should be included as control data. Fig. 4.3 illustrates the process – colored instructions represent the secret, and the stack is properly managed to keep the execution of the bytecode. The secret message corresponds to "00000111", such that "000" is encoded with PUSH1 00 by codifying the values in Table 4.7, "011" with PUSH1 a0 with the same mechanism and the last "1" with ADD operation. Then, after the initial state of the stack, PUSH1 00 is pushed to the stack (State 1), then PUSH1 a0 is pushed (State 2) and thirdly ADD (State 3). Finally, the stack should be restored by POPing all elements (Final state).

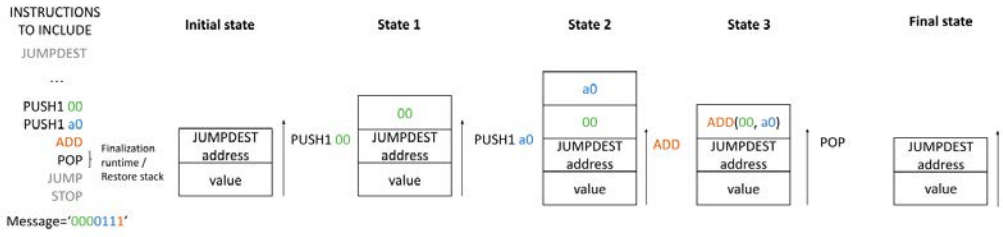

Figure 4.2: Embedding process in executable bytecode

By contrast, in the *Non-executable bytecode*, all instructions are secret-related, which also includes PUSH1 arguments. No extra information is required as the message is inserted in a new JUMP-JUMPDEST block that is never executed. Thus, the stack remains in the same state. The process is, therefore fairly similar to the *Executable bytecode*, but the first 2 instructions (PUSH1 00,PUSH1 a0) are selected by codifying the opcodes pairs on Table 4.5; and the last two by codifying

the pairs in Table 4.6.

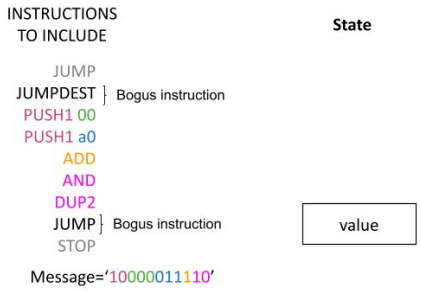

Figure 4.3: Embedding process in non-executable bytecode

4.4.3 Revealing mechanism

This process is analogous to the embedding one but in reverse order. Firstly, hidden data is extracted considering the field at stake. Secondly, control information is decrypted to delimit the message appropriately. Finally, the decryption is enforced.

However, one significant difference regarding the embedding procedure is that extraction does not require trial-and-error procedures. However, there is a performance overhead if the secret is to be revealed immediately (i.e., the sabotage case, recall Section 1.1). In this situation, the receiver has to wait until all transactions containing the secret are included (after mining) in the blockchain.

4.5 Evaluation

The evaluation of the proposed mechanism is performed from a theoretical and a practical point of view. Firstly, the compliance of established goals is analysed (Section 4.5.1). Secondly, an experimental analysis has been carried out to determine the actual cost and time required to hide a secret per Ethereum transaction field (Section 4.5.2).

4.5.1 Goals compliance

Table 4.11 summarizes the analysis on the imposed goals per Ethereum field, whose compliance is discussed in the following sections.

4.5.1.1 Stealthiness

The type of attacker in terms of stealthiness to which each field is resistant is depicted in Table 4.11(recall the attacker model defined in Section 3.3.2.4.1). Since the secret has been tailored to be disguised as normal values for each field (Section 4.3), almost all fields pass unnoticed to both BE and AE attackers, as there are no hints they might leverage on. For example, the *Swarm hash* has been proved to be random enough to be used in full and AE would need to have the original contract with the same file name to verify it, though there are situations with certain limitations (* is applied). In case of *Gas Limit* field, the study shows that it does not always match with the spent gas in the transaction and then, an attacker could have suspicions. Moreover, in *Executable bytecode*, the attacker should debug and understand that some of the instructions are really "dummy" code tailored as legitimate one but it is considered tedious and not really worthy. Nonetheless, there are a couple of exceptions in which just a single type of attacker applied. AE would notice some deviations from normality in *Non-executable bytecode*, as this code is never executed and could be more easily debugged.

4.5.1.2 Simplicity

The proposed mechanism achieves simplicity as long as there is no special requirement to embed secret information in any of the fields. However, the computational effort varies among fields. Most of them, marked as $O(1)$, only require one operation to hide information. Thus, the original contents of the field are replaced (partially or in full) by the secret. However, *Sender address*, *Contract address*, *Hash*, signature fields and *Public key* involve several repetitive operations until the

Table 4.11: Goals assessment per Ethereum transaction field. (*) means conditional achievement

	Field / Element	Subfield	Cost (gas)	Max capacity per transaction (bits)	Stealthiness (BE, AE, ALL)	Simplicity	Embedding computational cost	Secret integrity (IA)
Addresses	**Sender**		TransCost(1,0,0)	160	ALL	√	O(PoW)	√
	Receiver		TransCost(1,0,0)	160	ALL	√	O(1)	√
	Contract		TransCost(T_{Nc}-1,0,0) + ContCost(1, $\|B_{Ct}\|$, $\|B_{Ct0}\|$, $\|B_{Ctn0}\|$)	160	ALL	√	O(PoW)	√
Transaction info	**Value**	to user	C_{Tr} + S	22	ALL	√	O(1)	√
	Gas limit	to contract	TransCost(1,*,*) or ContCost(1, $\|B_{Ct}\|$, $\|B_{Ct0}\|$, $\|B_{Ctn0}\|$)	3	ALL*	√	O(1)	√
	Gas price	ALL		4	ALL	√	O(1)	√
	Nonce	-	-	0	-	-	-	-
	Signature: V	-	-	0	-	-	-	-
	Signature: r	ALL	TransCost(1,*,*) or ContCost(1, $\|B_{Ct}\|$, $\|B_{Ct0}\|$, $\|B_{Ctn0}\|$)	256	ALL	√	O(PoW)	√
	Signature: s	ALL		256	ALL	√	O(PoW)	√
	Identifier	ALL		256	ALL	√	O(PoW)	√
	Sender Public Key	ALL	TransCost(1,0,0)	512	ALL	√	O(PoW)	√
	Function args	address	TransCost(1,$\|IdF_0\|+\|S_0\|$, $\|IdF_{n0}\|+\|S_{n0}\|$) + C_F* N_F	160	ALL	√	O(1)	√
		bytes32		256	ALL	√	O(1)	√
		uint256		12	ALL	√	O(1)	√
Smart contract info (contained in **data** field	**Swarm hash**		ContCost(1, $\|B_{Ct}\|$, $\|B_{Ct0}\|$, $\|B_{Ctn0}\|$)	256	ALL	√	O(1)	√
	Bytecode	Non-executable	ContCost(1, $\|B_{Ct}\|$, $\|B_{Ct0}\|$ + $\|S_0\|$, $\|B_{Ctn0}\|$ + $\|S_{n0}\|$)	46	AE	√	O(1)	√
		Executable		33	ALL*	√	O(1)	√
	Constructor args	address	ContCost(1, $\|B_{Ct}\|$, $\|B_{Ct0}\|$, $\|B_{Ctn0}\|$) + C_{St} * ($\|S\|$/32)	160	ALL	√	O(1)	√
		uint256		26	ALL	√	O(1)	√

right value is found. Since the required effort is analogous to solving proof-of-work computational puzzles (265), they are marked as $O(PoW)$.

4.5.1.3 Efficiency

Though time efficiency will be studied in Section 4.5.2.3, efficiency in terms of the amount of sent information is studied herein. For this purpose, the size of the secret has to be higher than the data to be privately shared with the receiver beforehand – otherwise, the mechanism would not be needed. The data shared with the receiver is formed by 388 bits, namely transaction identifier (256 bits), encryption key (64 bits), nonce (64 bits) and fields to hide the secret (4 bits). Note that the use of functions and the constructor in smart contracts may require to know the ABI code but this is not necessary if such contracts are verified.

Efficiency of the amount of sent information, called Information Efficiency (IE),

depends on the secret size $\|S\|$ and it is calculated following Equation 4.3.

$$IE = \frac{\|S\|}{388} \tag{4.3}$$

The system is efficient as long as IE > 1. It must be noted that the individual capacity of each field per transaction would not meet this condition. However, *Zephyrus* enables using a series of transactions to hide a secret. In this way, as explained in Section 4.5.2.2, in our experiments secrets range from 400 to 40,000 bits, thus leading to $1.90 < IE < 315.05$. Moreover, an analysis per field is shown in Table 4.12.

Table 4.12: Maximum secret size, cost and IE per field in our experiments

Field	**Max secret size (bits)**	**Additional cost (ether)**	**Cost \ Fee (ether)**	**Cost \ Fee (USD)**	**IE**
Receiver address	40,760	-	0.005355	$ 1.64	105.05
Swarm hash	65,240	-	0.2631	$ 80.44	168.14
Gas Price	1,000	0.09	-	$ 27.51	2.58
Value	5,560	8.3137	-	$ 2,542	14.33
Gas limit	736	-	0.2575	$ 78.73	1.90
Function arguments	43,824	-	0.01073	$ 3.28	112.95
Constructor arguments	122,240	-	0.4490	$ 137.28	315.05
Non-executable bytecode	4,496	-	0.2215	$ 67.72	11.59

4.5.1.4 Cost

Embedding information in each of the fields has an associated cost. It is related to the fees required for sending information to Ethereum's blockchain. Particularly, sending transactions or deploying contracts have an associated cost, which can be measured according to Ethereum's documentation (72). These costs are described

by Equations 4.4 and 4.5 for transactions and contracts, respectively. In both cases, they have a fixed cost per operation and a variable part depending on the amount of data at stake.

$$TransCost(a, b, c) = a \times C_{Tr} + b \times C_{B0} + c \times C_{Bn0} \tag{4.4}$$

$$\begin{aligned} ContCost(a, b, c, d) = a \times (C_{Tr} + C_{Ct}) + \\ C_{CtCd} \times b + c \times C_{B0} + d \times C_{Bn0} \end{aligned} \tag{4.5}$$

Table 4.11 shows the cost per field leveraging these equations. *Sender* and *Receiver addresses* only need to send a transaction, and no additional payload is required. Optionally, some Ether could be included in the value to look like a natural transaction. Regarding *Contract addresses*, apart from deploying the contract, it is necessary to send transactions so as to make the nonce value lead to the required address value. Other fields involve a transaction in which the variable part increases with the size of the secret. In some of them, such part is increased with some inherent costs, such as the name of the function at stake in the case of *Function arguments*. It should be noted that in some cases (e.g., *Gas limit* or *Signature* and *Hash* fields) the sender might decide using a transaction to another user or a to a function in a contract or deploy a contract for embedding information. Last but not least, most contract-related fields involve deploying a contract, with some additions like the cost of storing information. To illustrate this discussion, Section 4.5.2.3 describes the real costs incurred by each of these fields in real transactions.

4.5.1.5 Secret integrity

The immutability property of Ethereum ensures that the secret embedded in most fields can always be recovered. In particular, even if the IA attacker creates any transaction, the secret message is not affected. Nevertheless, the only exception is the use of the contract storage.

4.5.2 Experimental study

A proof of concept has been implemented to measure the time taken for the proposed mechanism, as well as its associated costs. The implementation is described in Section 4.5.2.1. The description of the experimental settings is presented in Section 4.5.2.2. Afterwards, the obtained results are presented in Section 4.5.2.3.

4.5.2.1 Proof of Concept.

Zephyrus has been implemented in an open-source software tool available in Gitlab[1]. Through a command-line interaction, the user will be asked to provide the input required depending on the field at stake.

From a technical viewpoint, the tool has been developed in Python 3.5. For encryption purposes, AES in Counter (CTR) mode is applied for the secret and ChaCha20 for the control data. Encryption keys are derived by means of the Password-Based Key Derivation Function 2 (PBKDF2) algorithm (266). Sender and receiver/s can agree on an AES password and a nonce in a initial stage and increase this one per message transmission. ChaCha20 password is derived from the AES one and it changes per transaction to avoid patterns in encrypted information. Besides, the sender may send a message in different transactions and smart contracts to different receivers.

Regarding network connection options, *Zephyrus* is able to connect to a local node by interacting with the Go Ethereum Client (geth (267)), or to a Infura (268) node, so neither the sender or receiver/s need to have the blockchain synchronized, saving space and computational power.

In this current version of the implementation, all $O(1)$ (recall Table 4.11) methods have been implemented, except for *Gas price* to functions and contracts, as it is significantly cheaper, and *Executable bytecode*, as it allows embedding fewer information. Besides, only one field can be used for each secret.

[1] `https://gitlab.com/MarGA2503/zephyrus`

4.5.2.2 Experimental settings

Experiments have been run in a AMD FX-8370 8-Core processor equipped with Debian 9 OS with 16 Gb. of RAM. Note that the mining process is not part of our system and *Zephyrus* would work in any computer with similar characteristics and once installed Python 3.5 and used libraries (described in the prototype implementation[2]). Concerning the blockchain, Ropsten (269) has been used. Addresses have been provided with enough funds to carry out all transactions and Infura nodes have been used to connect to the blockchain.

To ensure the validity of our results, each embedding and revealing operation has been carried out 5 times. Afterwards, the arithmetic mean has been computed.

Concerning applied elements, the secret is a random set of 400, 2,000, 4,000, 8,000, 24,000 and 40,000 bits. On the other hand, the cover is different depending on the field at stake. In case of regular transaction fields, a tailored transaction has been created. In contract-related fields, different smart contracts are at stake. For the *Swarm hash* field and the *Non-executable bytecode* one, the same contract has been used (270). Regarding *Constructor arguments*, another contract with a constructor function has been applied (271). Most common smart contracts in Ethereum use ERC-20 tokens, the most popular ERC-20 token by market capitalization (272) has been used to test *Function arguments*. For the *Gas limit* field the contract used is (273). The gas limit for the rest of the fields has been set up according to a method available in Ethereum which estimates the necessary gas to complete the transaction (274).

Strategies and values analyzed in Section 4.3 has been used and function "approve", selected from (272), is applied to test *Function arguments*. For the sake of a balance between computational cost and time, the experiment allows a maximum of 255 transactions per field.

The use existing contracts provides realism to our results – *Zephyrus* could be

[2] `https://gitlab.com/MarGA2503/zephyrus`

applied immediately leveraging the current Ethereum contents.

4.5.2.3 Results

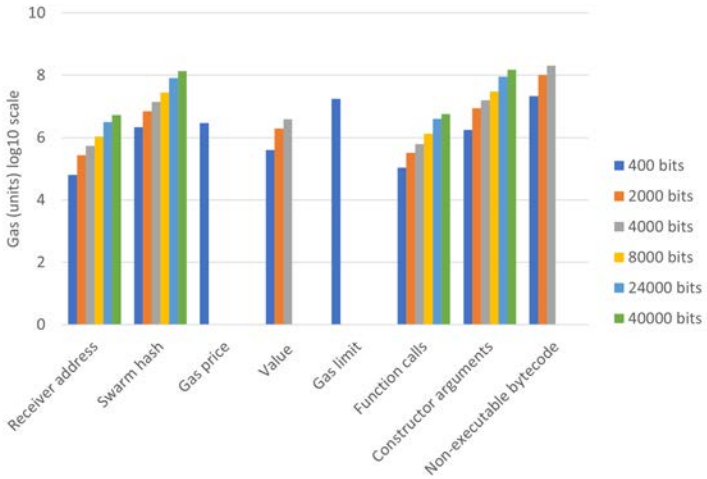

Figure 4.4: Gas cost per field (log scale)

Concerning the actual costs incurred by *Zephyrus*, Fig. 4.4 shows the gas cost per field depending on the secret size in bits. Note that when a contract deployment is at stake the amount of gas is affected by the size of the contract. Similarly, the capacity of the constructor and function arguments field depends on the number and types of arguments. As expected, the cost increases with the secret size, but depending on the field more or less data can be embedded. In the case of *Gas price* and *Gas limit* 400 bits can be embedded, as they are fields with embedding restrictions. It costs 6.5 and 7.2 gas units respectively. Indeed, the best alternative from the cost point of view is the use of the *Receiver address* and *Function calls* – their cost is 6.7 gas units in both cases when embedding 40 Kbits and 4.8 and 5.0 for 400 bits.

Table 4.11 depicts the maximum capacity per individual transaction, identifying fields in which up to 256 and 512 bits can be embedded. Moreover, Table 4.12 shows the maximum capacity of each field and the actual cost in USD, along with their IE ratio for all carried out transactions (255 in this experiment). For this purpose,

the average price (275) of 1 Ether in 2020 (1 Ether=$ 305.76) has been considered, taking the cheapest gas price (1 Gwei) (276). Note that embedding into *Value* and *Gas price* fields involves an additional cost.

The most efficient field, regarding stealthiness and cost is to embed a message in *Function arguments* allowing up to 43,824 bits for $ 3.28. However, inserting data in the *Receiver address* also provides great results. In relation to the quantity of embedded data, *Constructor arguments* method is the best with the tested contract. The most expensive one, *Value*, allows 5,560 bits for around $ 2,542, as real Ether is transferred. Besides, in terms of IE, results show that the system is efficient even using a single field in all cases. Nevertheless, significant differences exist between them like *Gas price* or *Non-executable bytecode*.

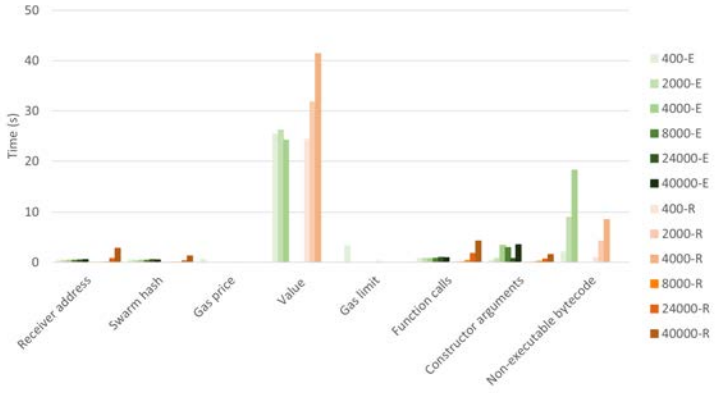

Figure 4.5: Embedding and revealing time per field

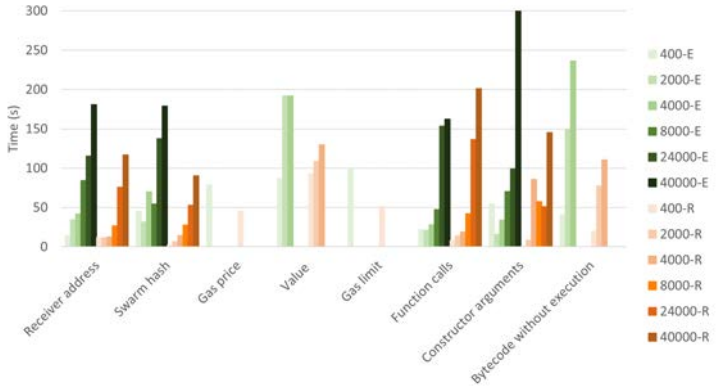

Figure 4.6: Network management time per field

With respect to the time taken by *Zephyrus* and linked to the efficiency goal, it can be divided into three main parts, namely embedding time, network management time and revealing time. Note that network management time corresponds to sending the transaction to the blockchain, its retrieval and the mining time. Although such management is out of the scope of *Zephyrus*, it will be unavoidably required for its usage in the real world.

The embedding (E) and revealing (R) time for every field depending on the secret's size is presented in Fig. 4.5. Embedding involves the encryption of the secret and the preparation of required transactions. Conversely, revealing requires extracting the secret and decrypting it afterwards. The time of encryption and decryption is quite similar for all fields, $4.5\text{x}10^{-3}$ s on average for both operations. As expected, the size of the message directly affects the time of embedding and revealing but not to a great extent. However, the secret's size slightly affects the time spent in the revealing, as more transactions are managed and more checking operations are required. Though the time differs between embedding and revealing, it is minimum considering the applied scale (max. 65 s). This is in line with expectations, as both operations are similar but applied in reverse order. Also noticeable is the fact that including data in the *Value* is the toughest operation because of the amount of required numbers according to the selected pattern (recall Section 4.3.1), it takes 24.2 s for embedding and 41.4 s for revealing 4 Kbits, which is the worst case. As a trade-off between efficiency and stealthiness, the best choice is the use of *Function calls*, as they allow embedding 40 Kbits in 0.93 s and retrieving them in 4.3 s. On the contrary, just looking for time restrictions, *Receiver address* provides meaningful results as embedding and revealing time is quite reduced even for 40 Kbits – 0.57 s and 2.8 s respectively.

For illustration purposes, though it depends on the network status, a complete overview of the steganographic process is analysed including network management time. Fig. 4.6 presents the time of embedding (E) and revealing (R) data. E

includes the mining time and that of sending data to the network, and R includes the retrieving time from the network. When contracts are not involved, for instance, mining time for *Receiver address* and 40 Kbits takes around 50 s on average, whereas the time to connect and send the information to Infura is around 130 s, 180.1 s for the whole embedding process. However, in those cases involving a contract deployment, for example the *Swarm hash*, mining times are usually higher, 84 s, and 94 s to send the transaction to the network, leading to a total embedding time of 178.1 s. Regarding revealing time, it depends on the amount of transactions at stake and how they are mined (i.e., same or different block and distance between them). For instance, transactions with smaller sizes mined in the same blocks take less time to be retrieved than larger transactions in different blocks. In the worst case, the revealing time of *Function calls* is 201.1 s for 40 Kbits. In the best case, it is 90.4 s for *Swarm hash* and the same amount of transmitted data.

This analysis shows the feasibility of using *Zephyrus* for building covert channels. In sum, on average, the embedding and revealing procedures take 8.07 s, while network management takes 154.23 s and thus, 162.3 s (2.71 min) in total.

4.6 Related work

Several works combine the concepts of steganography and blockchain. For instance, in (277) secrets are embedded in images which are later shared through a blockchain and a file system. On the other hand, in (278) each transaction is divided in two parts which are hidden in videos. However, here we focus on the use of the blockchain itself for steganographic purposes – the secret is directly hidden in blockchain data. In this regard, recent results show that it has not been detected in Bitcoin (279), although most proposals focus on this cryptocurrency. This is the case of Ken Shirriff's blog (16), which presents some basic steganographic techniques. They use encoding (e.g. hexadecimal, base64, etc.) to hide different messages, texts or files but any of them is really sophisticated. In this

way, decoding is the only process required to access secret messages. One example is the use of the receiver address to store data, namely the inclusion of an image of Nelson Mandela and a tribute text. Many transactions were generated to store all information, in hexadecimal, into receiver addresses. Each transaction can contain 20 bytes of data. The use of an arbitrary field of 100 bytes or more, in the coinbase block (e.g. the initial block of the chain) has been also applied to hide data, namely a political sentence or some prayer names. Another example pointed out is the concealment of data in the hash of the public key script (P2PKH), used to verify performed transactions. Dan Kaminsky used this method to embed a tribute to the cryptographer Len Sassaman. It is also common to replace keys in a multi-signature transaction, in these case the 1-3 type. A final example is the use of Nulldata transactions, in which the OP_RETURN (Null data transaction) field is applied for invalid transactions. This technique has been used to store lyrics of Rick Astley. However, though OP_RETURN can be used once per transaction, its use should be limited to not raise suspicions.

Also with the focus on Bitcoin, (236) presents different fields to hide messages without the use of encryption. Data is included in the timestamp (nLockTime) and in the sequence number, but a combination of multi-signature (1-12) inputs and outputs, transaction amount and Nulldata transactions were finally used. In the case of signatures, the secret message is embedded when a valid signature is computed; and in case of the transaction amount, the budget is split in multiple transactions based on a combinatorial composition.

In relation to this cryptocurrency, A. Sward et al. (17) review the different existing data insertion methods like including information in the public key in a Pay to Public Key (P2PK) transaction, or in the hash of the public key in a Pay to Script Hash (P2SH), both methods using the ScriptPubKey of a transaction. Regarding the ScriptSig, P2SH script could also be used, either inserting data on the Reedem part of the script or in the Data Input part.

R. Matzutt et al. (18) analyse the impact of inserting content on Bitcoin, explaining different methods and naming some of the existing tools that are able to perform this action.

On the other hand, R. Recabarren at al. (20), propose Thitonious, an anti-censorship Bitcoin tool, using the scriptSig of a P2SH multisignature transaction by inserting the message on the 28 most significant bytes. Thitonious allows users to access free-altruistic content published in clear text, or pay for on-demand content. In this case, the information is encrypted.

M.D. Sleiman et al. (280) propose inserting text in the transaction amount of Bitcoin by using an arithmetic encoding, which provides a space of eight characters (seven plus the termination symbol in an ideal case) that should be lower-case English characters, spaces or periods. Multiple transactions can be used to insert larger messages.

Tian et al. (281) use the OP_RETURN and Private key in Bitcoin transactions. The private key (32 bytes) is used to embed the message while the OP_RETURN is used in order to change the labels between messages which are generated in a dynamic way and are statistically indistinguishable from normal transactions. Data is encrypted before the embedding process.

Fionov (282) reviews briefly the existing covert channel in Bitcoin. Furthermore, he proposes a method based on permutations of transaction outputs, inputs and values (payments), whose number affects capacity. The secret message is encrypted. However, just one transaction is used to justify the number of inputs and outputs in this study and further analysis is highlighted as a necessity.

Torki *et al.* (283) propose a pair of algorithms to embed data in blockchains. One of them has high embedding capacity as secret data is embedded in transactions' data and the other one has medium capacity embedding data in sender addresses. Though both algorithms seem to be general, they are directly related to Bitcoin transactions.

D. Frkat et al. (284) propose ChainChannels, a scheme to send hidden information to bots within ECDSA signatures. The sender introduces the message in the random number used to generate the signature and the receiver needs to know the signature private key to retrieve the message. Besides, Bitcoin network is used for evaluation purposes.

By contrast, Ethereum is used by Basuki et al. (285), working with image steganography. Instructions for recovering the secret within the image are included in the timestamp of a smart-contract, allowing 29 bits of capacity. Then, the image is stored in a web server and clear text data is stored in the blockchain for the secret recovery.

Gao et al. (286) use kleptographic algorithms in order to identify which transactions have secret information. Even though different fields of Bitcoin and Ethereum blockchains are mentioned, most of them are not studied. According to a proof of concept, just Ethereum OP_RETURN and data field are used for steganographic purposes with 80 bytes of capacity, embedding encrypted data.

Ethereum is also used in Liu et al. (21). They only use the Value field. They propose three different ways of including information with a maximum of 1, 30 and 15 bits per transaction.

Some other proposals are applicable to a different range of blockchains and cryptocurrencies. J. Partala (235) suggests a method for securely embedding covert messages into a general blockchain. The sender generates payments and the secret message is embedded, bit by bit, in the LSB of each receiver address. Then, the sender and the receiver have to order and collect bits accordingly.

N. Alsalami et al. (287) propose the use of CryptoNote framework, applied in cryptocurrencies like Monero, by embedding a message in the ring signature's random numbers.

Finally, Xu et al. (288) embed secret information in the blockchain using the sender address of preselected transactions according to a certain key. Selected

transactions are arranged in a certain way in order to carry the secret message. The amount of data that this method is able to transmit depends on the quantity of transactions that can fit in a block and the number of different senders. However, the sender of the secret message should mine the block and the receiver needs the key to retrieve it.

4.6.1 Summary of related work

In the case of (21), authors focus on the *Value field* by characterizing its entropy and length. However, their approach does not consider the frequent patterns appearing in this field, as we have discovered in our study. Moreover, some of their proposed schemes require the message starts by 1, which may be of interest for an attacker.

Note that there are elements marked as equivalent to smart contracts, but they correspond to an extremely simplified version of them. Capacity is expressed in terms of a single input and output in a Bitcoin transaction, though Bitcoin transactions may have multiple one. In Bitcoin, the maximum size of inputs is of 1,650 bytes, each element in the stack can have a maximum size of 520 bytes, whereas that of the whole transaction should not exceed 100,000 bytes. Ethereum, on the other hand, works per transaction. It means that each action in the chain requires a different transaction. However, the maximum capacity per single input/output in Bitcoin is comparable with *Zephyrus* in many cases.

On the other hand, (235; 284; 287; 288) could be used in Ethereum too. In the case of (280), Bitcoin could be replaced by Ether, but it should be noticed that this method only allows English text messages, whereas *Zephyrus* can transmit any binary information.

On the other hand, stealthiness is the most remarkable issue, except for (235; 281; 282; 283; 284; 286; 287; 288) and (20) when paying for content, all techniques offer a limited protection against BE. Approaches in (236; 285) embed data in clear text and (16) applies encoding. *Zephyrus* allows users the exchange of encrypted (or

Table 4.13: Related work summary

Reference	Cryptocurrency	Method	Equivalent in Ethereum	Max capacity (bits)	Stealthiness (BE, AE, ALL)	Simplicity	Embedding computational cost	Secret integrity (IA)	Comparision of embedded data with normal content in blockchain
(16)	Bitcoin	Receiver address(in hex)	Receiver address	160	BE*	√	O(1)	√	X
	Bitcoin	Block coinbase	Block Extra Data	Up to 800	BE*	-	O(1)	√	X
	Bitcoin	ScriptPubKey:paytopubkeyhash	Smart contract**	160	BE*	√	O(1)	√	X
	Bitcoin	ScriptPubKey:paytoscripthash-multisig	Smart contract**	1560	BE*	√	O(1)	√	X
	Bitcoin	Null data transaction	Transaction data	640	BE*	√	O(1)	√	X
(236)	Bitcoin	ScriptSig	Smart contract**	8	ALL*	√	O(PoW)	√	X
	Bitcoin	ScriptPubKey:paytoscripthash-multisig	Smart contract**	2766	ALL*	√	O(PoW)	√	X
	Bitcoin	Transaction amount	Transaction value	4	ALL*	√	O(1)	√	X
	Bitcoin	nLockTime	-	32	ALL*	√	O(1)	√	X
	Bitcoin	Sequence number	-	32	ALL*	√	O(1)	√	X
(17)	Bitcoin	ScriptPubKey:paytopublickey	Smart contract**	520/264	BE,AE	√	O(1)	√	X
	Bitcoin	ScriptPubKey:paytoscripthash	Smart contract**	160	BE,AE	√	O(1)	√	X
	Bitcoin	ScriptSign(Redeem script):paytoscripthash	Smart contract**	4136	BE,AE	√	O(1)	√	X
	Bitcoin	ScriptSign(Data input):DataDropwithoutSignature	Smart contract**	13040	BE,AE	√	O(1)	√	X
	Bitcoin	ScriptSign(Data input):DataDropwithSignature	Smart contract**	12232	BE,AE	√	O(1)	√	X
	Bitcoin	ScriptSign(Data input):DataHashwithoutSignature	Smart contract**	12480	BE,AE	√	O(1)	√	X
	Bitcoin	ScriptSign(Data input):DataHashwithSignature	Transaction dataSmart contract**	11688	BE,AE	√	O(1)	√	X
(20)	Bitcoin	ScriptSign:multisignature	Smart contract**	448	BE,AE/ALL	√	O(PoW)	√	X
(280)	Bitcoin	Transaction amount	Transaction value	64	BE,AE	√	O(PoW)	√	X
(281)	Bitcoin	Sender private key	Sender private key	256	ALL	√	O(1)	√	√
(282)	Bitcoin	Transaction inputs + outputs + value	Transaction sender address + receiver + value	N/A	BE/ALL (depends of numbers per field)	√	O(1)	√	X
(283)	Bitcoin	Receiver address + ScriptPubKey:paytopublickeyhash + ScriptSign(Redeemscript):paytoscripthash	Transaction receiver address + Smart contract**	81.9	ALL	√	$O(2^m)$	√	X
(285)	Ethereum	Timestamp in smart-contract (Function call with string)	Smart-contract	29	BE	√	O(1)	√	X
(21)	Ethereum	Value field	Value field	30	AE	√	O(1)	√	√
(235)	Any	Receiver address	Receiver address	1	ALL	√	O(1)	√	X
(284)	ECDSA-based	Nonce of the signature	Sender address/Sender public key	256	ALL	√	O(1)	√	X
(287)	CryptoNote-based-based	Signature	Sender address/Sender public key	504	ALL	√	O(1)	√	X
(288)	Any	Sender address	Sender address	Blockchain depending	ALL	-	O(1)	√	X
(286)	Bitcoin	Null data transaction	Transaction data	640	BE*	√	O(1)	√	X
	Ethereum	Transaction data	Transaction data	640	BE*	√	O(1)	√	X
Zephyrus	Ethereum	Sender address Receiver address Contract address Transaction data Signature R Signature S Transaction identifier Sender Public Key Gas price Transaction value Gas Limit Function calls Swarm hash Constructor arguments Executable bytecode Non-executable bytecode	-	Several values (cf. Table 4.12))	AE/ALL	√	O(1)/ O(PoW)	√	√

** simplified version of smart contracts * simple hidden technique

clear) messages without any retrieving cost, unlike Thitonious which is a commercial service for demand and encryted content. (281) uses dynamic labels in order to hide the communication, *Zephyrus* could also change the sender address for each transaction using the control and pre-shared information.

In terms of simplicity, almost all techniques can be used by regular users and just 'Block coinbase' (16) and (288) require to be a miner. Similarly, the computation cost of embedding is also analogous in most approaches (O(1)), except for (236) in which 'ScriptSig' and 'ScriptPubKey:scripthash-multisig' need to compute a valid key to make coins reedemable; (20) requires generating valid-looking quadratic residues; and (287) uses a non-lineal message retrieval process. Finally, secret integrity is achieved in all cases, thus being resistant against IA.

Chapter 5

A steganographic tool leveraging Solidity code: Smart-Zephyrus

5.1 Summary of the chapter

In order to further explore the prevalence of Ethereum and the lack of use of covert channel, as well as to complete the mechanism proposed in Chapter 4, in this Chapter we propose Smart-Zephyrus, a steganographic tool that leverages Solidity code to embed information. With this chapter we want to address two things. First, another research gap exposed in Section 3.3.3.2 – the lack of use of smart contracts for sophisticated covert channel purposes and the fact that high-level languages are not used. Furthermore, although Zephyrus is capable to insert secret information in contracts, it is not capable of inserting it on the high-level language used to program them.

In Section 5.2 we present an overview of the proposal, including a comparison with the mechanism proposed in Chapter 4. In Section 5.3 we explain the reasons behind the proposal of the mechanism. In Section 5.4, we analyze real Ethereum contract data a in order to identify the best way to insert information. Based on those results, the mechanism is proposed in Section 5.4.1. In Section 5.5 a evaluation of Smart-Zephyrus is shown in ordder to assess the mechanism feasibility. In Section 5.5.1, a summary of all related work regarding steganography and malware

is explained. Smart-Zephyrus is also compared with the related work.

In this Chapter, the experimental tests will be done considering attacker-malware communication, although it can be used for any other kind of covert communication.

5.2 Overview

Smart-Zephyrus works over Zephyrus (recall Section 4) but instead of only using EVM/bytecode elements of the smart contracts, a high level programming language to embed information is applied. Figure 5.1 shows a comparison between Zephyrus and our current proposal, Smart-Zephyrus. While Zephyrus embeds information on blockchain raw data, Smart-Zephyrus allows embedding information in a high level program language (Solidity). This language is then compiled and transformed to contract bytecode to be sent to the blockchain. In order to retrieve the Solidity code, the contract code needs to be verified in a blockchain explorer, which checks that the Solidity code and the bytecode deployed on chain matches.

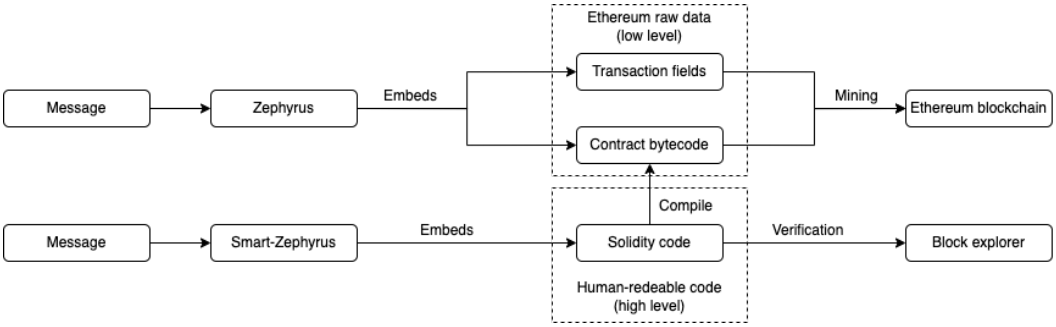

Figure 5.1: Zephyrus vs Smart-Zephyrus

To know how to embed information in smart contracts without raising suspicion, a study is initially carried out (Section 5.4). We have retrieved a sample of different common and used contracts to study their characteristics in order to mimic them. Once these characteristics are analysed the selection and design of contracts to embed information is performed. As explained later, contracts will be built by

altering different issues of a pre-existing one, such as the amount of contracts to inherit (called parent contracts) or the types of libraries they use, to name a few.. Two different versions of the mechanism are developed, one more statistically similar to the studied sample, and, thus, stealthier; and another one that allows more capacity, but less similar to the studied data.

After that, a practical experiment has been carried out in order to validate the similarity of our mechanism to the studied sample, as well as to measure the time and cost of sending and retrieving different sizes of hidden messages, while justifying feasibility of the proposal.

5.3 Motivation

We have observed through the previous study (Section 3.3.2) that blockchain has some intrinsic characteristics (i.e availability and immutability) that attracts attackers, for instance, to increase the difficulty in taking down a source of information (i.e C&C servers).

On the other hand, different kind of malware could use the blockchain for information sharing purposes:

- Ransomware. The blockchain could be used to share new payment addresses or decryption/encryption keys. Common keys/message sizes include 256 bits (Elliptic Curves (289)), 1024 bits (minimum size for RSA (290)) and 4096 bits (RSA maximum size (291)).

- Botnet. The blockchain can be used to exchange commands and new addresses for the system. A typical message is usually small (less than 512 bytes) (292).

Identified as research gaps (recall Section 3.3.3.2), covert channels are barely used. Thus, all communications between malware and attacker or victim can be easily accessed by anyone. This makes easy for analysts and defenders to identify and flag botnets (particularly their C & C servers), or attacker addresses.

Therefore, the motivation of building a covert channel leveraging blockchains is of direct interest for attackers. If communications are hidden, it reduces the chance of being somehow intercepted.

5.4 Preliminary smart contract study

A study of Ethereum smart contracts is firstly carried out to generate stealthy smart contracts afterwards. These contracts were downloaded from the Ethereum blockchain. A total of 103,106 contract files are analyzed (available on Gitlab[1]). Within those files, it is possible to find numerous contracts, interfaces and libraries which have relations and dependencies among them.

Contracts in Solidity are similar to classes in object-oriented languages. Interfaces are one kind of contracts that do not contain any logic, just definitions. Libraries, on the other hand, contain logic that can be later referenced in contracts.

After processing the smart contracts, the following findings are highlighted. As shown in Table 5.1, most contracts belong to Openzeppelin. Ownable is intended to allow the transfer or withdrawal of a contract ownership, so it has little real-world application by its own. On the contrary ERC20 is a well-known token – a blockchain-based asset that can be traded. Therefore, Smart-Zephyrus will leverage ERC20 token contracts.

To ensure the representativeness of the considered ERC20 contracts, 7,143 of them were retrieved from (293). To the best of authors' knowledge, it is the biggest dataset in this regard. Those whose Solidity code was available (6,632 contracts) were analyzed according to the ERC20 standard peculiarities. It must be noted that their code is used for embedding purposes. The results of the analysis are shown in the left column of Table 5.3.The following is discovered:

- In the case of those ERC20, the mean of contracts per archive is 6.05 and the standard deviation is 16.87. In the case of interfaces the mean is 1.12

[1] `https://gitlab.com/MarGA2503/retrieved-contracts`

Table 5.1: Top 10 smart contracts

Name	Appearences
Ownable	28,292
ERC20	21,227
Context	19,426
StandardToken	11,835
ERC20Basic	9,748
BasicToken	7,350
Token	5,341
Pausable	5,109
Owned	5,070
ERC20Interface	4,972

per contract with a standard deviation of 2.96. There are 1.39 libraries per contract with a standard deviation of 3.54.

- A large variety of names have been found. The most common word in the name was "Token" that appeared in 6.83% of the retrieved token names. Regarding a possible relation among words in the names, the most common pair appears 69 times. There is 13,197 different pairs. We can conclude that words in the token names are not necessarily related in any way. Furthermore, there % of unique words in the name is 55.65.

- The mean length of the token name is of 1.76 words, with a standard deviation of 1.01.

- The most common symbol is SMT, with a percentage of 0.09% in relation to the total found token symbols. This result entails that a great variety of token symbols have been defined in the studied contracts. On the other hand, it has been identified that the symbol contains the letters of the name in 81.80 % of the cases.

- Regarding the length of the symbol, the mean is 4.02 letters with a standard deviation of 2.04. The most used length is 3 letters, followed by 4 letters.

- The token contracts inherits from a mean of 1.49 contracts with a standard deviation of 1.04.

- In the token contract, 10.44% of the inherit contracts are in different order from the order in which they are defined in the contract file.

5.4.1 Proposal

In this section we present the mechanism. Section 5.4.2 explains some design decisions taken regarding the construction of the tool. Lastly, Section 5.4.3 descirbe how data is codified and hidden in the contracts.

5.4.2 Design decisions

Based on the findings explained in the previous Section, some design decisions for the construction of steganographic (recall Section 2.6) smart contracts have been taken.

On the one hand, the base contract will be ERC20 due to its popularity (recall Section 5.4). On top of this choice, it will inherit or use a number of contracts, will utilize a set of libraries and will provide with some interfaces. All these decisions will be inspired by our preliminary study. Thus, the number of chosen contracts (besides ERC20) the token contract will either use or inherit from will be 2, and the amount of libraries and interfaces will be adapted to the actual choice of contracts.

In what comes to the selection of contracts, libraries and interfaces must be compatible among them. They must also be valid for the current Solidity compiler (Pragma 0.8) and should not produce inheritance loops. Thus, the set of contracts to choose from are Ownable, Pausable, ERC20Burnable, ERC20Capped, ERC165, ReentrancyGuard, ERC20Permit, TokenTimelock and ERC20Snapshot. All of them belong to Openzeppelin (82), in line with our analysis, to promote stealthiness.

5.4.3 Data insertion mechanisms

In this proposal two modes of operation will be defined considering different level of capacity and stealthiness:

- Stealthy. Mechanism with lower capacity but more accurate and similar to the studied values extracted from original contracts.
- Capacity. Allows more capacity, but it sacrifices stealthiness making the steganographic contract more different from legitimate ones.

Figure 5.2 shows the main steps of Smart-Zephyrus. The message is symmetrically encrypted in first place. Thus, confidentiality is provided making difficult any statistical analysis and the possibility of an entity to spot the malicious communication. Control information, required in the revealing process, is also included and encrypted, but with a stream cipher to reduce length at a minimum. Finally, data is embedded according to proposed operation modes, producing one or more smart contracts with portions of the secret.

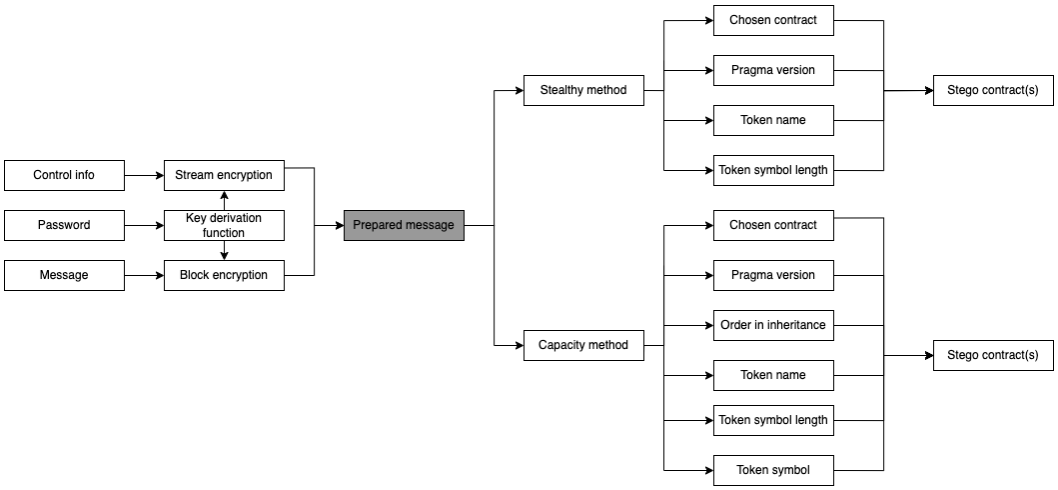

Figure 5.2: Smart-Zephyrus main steps

In any of the modes, Smart-Zephyrus tries to generate smart contracts undetectable to potential attackers, thus hidden messages embedded in smart contract

are indistinguishable from existing ERC20 contracts. There are different ways in which information is embedded:

- Chosen contract encoding. Each contract (except from ERC20 which will always be present) will be assigned a value (3 bits). For example: Contract0=0, Contract7= 7. Once Contract0 is chosen, it is removed from the list, and Contract8 is added and values are assigned again. Contracts are ordered by appearance in the studied (i.e. real-world) sample, meaning that the most common contracts have a bigger chance of being chosen.

- Order of the contract in inheritance. A contract can inherit from different contracts. The order in which these contracts appear in the token child contract inheritance definition can be used to insert information. For example if a contract inherits from Ownable and Pausable, they can be ordered in two ways. Each combination is then assigned a value to insert information. As a majority of contracts preserve that same order (recall Section 5.4), this will only be used in the Capacity mode.

- Pragma version. It goes from 0.8.0 to 0.8.17 (by now) so 4 bits can be encoded (log2(18)).

- Token name. In order to embed as much information as possible, a dictionary has been used to give a number for each word. The Word Game Dictionary[2] was adopted. In particular, $log_2 266,336 = 18$ bits were used per word in the Capacity mode. For the Stealthy mode, all words shorter than 4 were removed (as they were infrequent in token names, only 13.64% of them). This leads to $log_2 11,202 = 13$, bits per word. In any case 2 words appear in token names in line with our observations (recall Section 5.4).

- Token symbol length. 3 and 4 characters are chosen, allowing 1 bit for embedding information.

[2] `https://www.wordgamedictionary.com/sowpods/download/sowpods.txt`

- Token symbol characters. This is only used in the Capacity version. Considering the target length, a substring of that size is produced based on all character combinations from the selected token name. Embedding is thus done by assigning a value to each combination.

Let *nph* be the number of permutations of parent contracts and *ncl* the combinations without repetition of the token name characters. Table 5.2 shows a summary of the different methods used to embed information and their capacity.

Table 5.2: Steganographic capacity per method

Method	**Possible values**	**Quantity of bits**	Type of mechanism
Chosen contract encoding	6	3*2	Both
Order of the contracts in the token inheritance	nph	$\log_2 nph$	Capacity
Pragma version	18	4	Both
Token name	11202(Stealthy)/ 266336 (Capacity)	2*13 /2*18 (Capacity)	Both
Token symbol length	2	1	Both
Token symbol letters	ncl	$\log_2 ncl$	Capacity

5.5 Assessment

The proposed approach is assessed by creating a list of smart contracts and confronting them against real-world ones. Furthermore, the cost and time for the attacker is also measured.

The experimental settings are described in Section 5.5.1. Afterwards, the statistical comparison against existing contracts is shown in Section 5.5.2. Lastly, the measurements on cost and time are presented in Section 5.5.3.

5.5.1 Experimental settings

Experiments have been run in an Intel Core i7-1165G7 processor equipped with Debian WSL for Windows 10 with 16 Gb. of RAM. A proof of concept implementation of Smart-Zephyrus is publicly released to foster further research[3].

Note that the mining process is not part of our system and Smart-Zephyrus works in any computer with similar characteristics and once installed Python 3.8

[3] `https://gitlab.com/MarGA2503/smart-zephyrus`

(or higher) and used libraries. Concerning the blockchain, Sepolia has been used to carry out the experiments because it is the one recommended by Ethereum.org as the default testnet for application development (294) and the rest of the testnets are currently deprecated. Addresses have been provided with enough funds to carry out all transactions and Infura nodes have been used to connect to the blockchain. To ensure the validity of our results, each embedding and revealing operation related to network usage in Section 5.5.3 has been carried out 5 times, while for program computation times (embedding/retrieving of the message and ciphering) it has been repeated 250 times, thus ensuring the soundness of results. Note that network time has been limited to 5 repetitions for being a time consuming task. Afterwards, the arithmetic mean and the standard deviation have been computed.

Table 5.3: Original contracts vs Smart-Zephyrus generated contracts

Feature	**Measurement**	**Original ERC20 contracts**	**Generated contracts (stealthy mode)**	**Generated contracts (capacity mode)**
Number of contracts	Mean(std) in file	6.05(16.87)	5.23 (0.42)	5.23(0.42)
Number of interfaces in file	Mean(std)	1.12(2.96)	2.67(0.59)	2.67(0.59)
Number of libraries in file	Mean(std)	1.39(3.54)	1.56(1.58)	1.53(1.59)
Contracts and functions in the same order as standard (all)	% Mean(std)	70.31(32.07)	100(0)	100(0)
Contracts and functions in the same order as standard (pragma 8)	% Mean(std)	95.94(5.79)	100(0)	100(0)
Token names	% of unique words	55.65	48.75	94.68
Token name lengths	Mean	1.76(1.01)	2.05(0.22)	2.06(0.23)
Token names	% of appearance of the word “Token”	6	2.67	2.93
Token symbols	% of unique symbols	85.62	86.65	77.97
Token symbols lengths	Mean(std)	4.02(2.04)	3.51(0.49)	3.50(0.49)
Token symbols lengths	% of all letters in token name	81.80	81.51	100
Token herency contracts number	Mean(std)	1.49(1.04)	2.07(0.54)	2.06(0.54)
Token herency contracts	% of different order	10.44	5.48	49.24

5.5.2 Comparison of studied vs generated smart contracts

First, in order to ensure the mechanism compliance with the studied sample, a comparison among studied contracts and the ones generated is depicted in Table 5.3. The embedded secret message consists of *lorem ipsum* text encrypted with

AES. The same quantity of smart contracts as in the sample is selected to make a fair comparison.

The number of smart contracts, libraries and interfaces in the generated contract files are really similar to those studied. The highest differences are in the number of contracts, as the mean of the generated samples is 6.05 in both versions versus 5.23 of the existing ones; and in the number of interfaces, with 1.12 of average in the generated contracts as compared to 2.67 in the existing ones.

In what comes to the order and names of functions, the mean of files that include Openzeppelin contracts as they are (same order and function names) is 95.94% for the same compiler version as our generated contracts. This has been replicated in the output of Smart-Zehpyrus – Openzeppelin contracts will always be present as they can be retrieved from the creator.

Concerning token names, for the stealthy insertion method, the number of unique words is a little less with 48.75% and "Token" appears 2.67% of the times. This is slightly lower than the original sample. Note that the appearance of the word "Token" follows a random distribution which tries to imitate the original one and then, the difference between the generated samples and the original ones may change depending on the status of the applied random generator. On the contrary, the difference is noticeable in the capacity method, in which the number of unique words increases to 94.68%.

Regarding the number of words used in the name, the results between the studied and generated sample are really similar, being the mean of the former 1.76 with a standard deviation of 1.01 and 2.05 and 2.06 with a standard deviation of 0.22 and 0.23 for the stealthy and capacity method respectively.

The percentage of unique symbols in the studied contracts is 85.62% while in the generated contracts this percentage is a bit smaller, 77.97% for the capacity version but more similar (86.65%) for the stealthy version.

Something similar happens with the mean and standard deviation of the number

of letters of symbols. In the studied sample the mean is 4.02 letters per symbol, with a standard deviation of 2.04. In the generated contracts, the mean is 3.51 characters per symbol for the stealthy method and 3.50 in the capacity one with standard deviations of 0.49.

With respect to the order in inheritance, for the studied sample, only 10.44% of contracts differs in order. For the generated contracts, in the stealthy method this percentage is 5.49%, while for the capacity is 49.24%. This is because the stealthy mode is randomly generated according to the reference value while in the capacity mode the goal is to embed a significant amount of data.

According to these results the mechanism generates contracts that are fairly similar to the legitimate contracts already deployed on the chain.

5.5.3 Experimental results

Experiments are carried out to measure the gas cost, according to the capacity of generated steganographic smart contracts, and the time of the embedding and revealing process. The length of embedded messages is: 256, 1,024 and 4,096 bits. They are based on common key lengths as a possible way to exchange keys for any kind of malicious purposes.

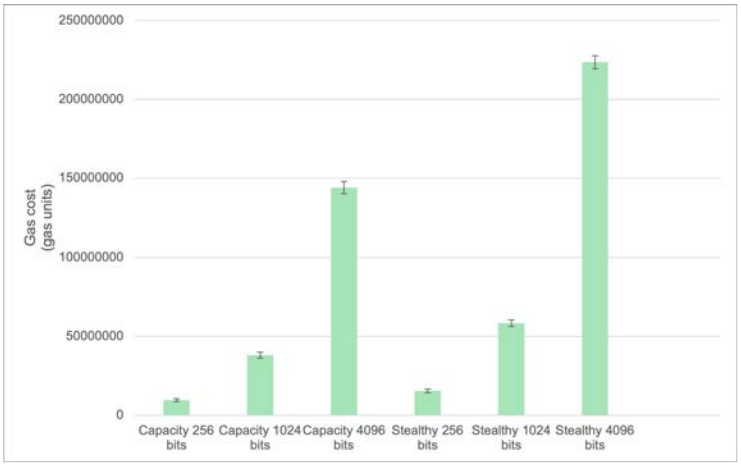

Figure 5.3: Gas cost per method

5.5.3.1 Cost assessment

Concerning the actual costs incurred by Smart-Zephyrus, Fig. 5.3 shows the gas cost per method depending on the message size. As expected, as it allows less capacity per contract and thus it needs more contracts for the same size, the stealthy method is more expensive than the capacity one. The cost also increases with the secret size. For the maximum secret size (4,096 bits) the mean gas cost for the capacity method is 144,001,954.7 gas versus 223,530,291.9 for the stealthy one. Table 5.4 shows the cost on Ether and USD of each mechanism and secret. In order to calculate the cost on Ether, the average gas price for the last three months (December 10, 2022 to March 10, 2023) (295) has been considered, being 27.22 gwei. On the other hand, 1 ether is $1,466.03 (recall Section 3.3.2.8).

Table 5.4: Steganographic capacity per method

Method	Gas cost (mean)	Cost in ether	Cost in USD
Capacity 256 bits	9,574,380	0.2606	382.07
Capacity 1024 bits	38,184,016	1.0393	1,523.74
Capacity 4096 bits	144,001,955	3.9197	5,746.45
Stealthy 256 bits	15,447,935	0.4204	616.46
Stealthy 1024 bits	58,412,600	1.5899	2,330.97
Stealthy 4096 bits	223,530,292	6.0844	8,920.05

These costs seem expensive but there are a couple of issues to take into account. First, traditional attack methods also incur a cost for the attacker (252). For example, an hour-long DDoS attack using a cloud server costs criminals $7 (253) and such server could be still taken down if discovered. In this regard, in botnets like Zeus or Mirai, the malware package costs from $700 up to <$10K and <$30, respectively (296). Besides, while the cost of maintenance in Mirai is unknown, in Zeus is of $62k (296). Furthermore, revenue from ransomware is usually higher than these costs. For example, Wannacry has three known Bitcoin addresses with payments of 54.43 BTC. In today's value that is around USD 1,120,109.53 so a cost of USD 8,920.05 (our highest) is not much in comparison.

On the other hand, our mechanism provides something previous proposals do not – a high level of stealthiness. According to studied works (recall Section 3.3.2.4) most of the time the communication is in clear and not hidden, thus traceable and blockable. Furthermore, among those proposals which actually use covert channels to hide information (25; 27; 211; 230; 239), the only proposal that effectively hides information surpassing all models of attackers is Chainchannels (211) but its cost is unknown. Furthermore, it does not use contracts to hide information. Our proposal, on the other hand, imitates fully usable smart-contracts and hides information in a way that makes them indistinguishable from the normal ones. Moreover, information can be encrypted, thus achieving equal stealthiness (ALL - attacker model) to (211) and also providing an estimated cost.

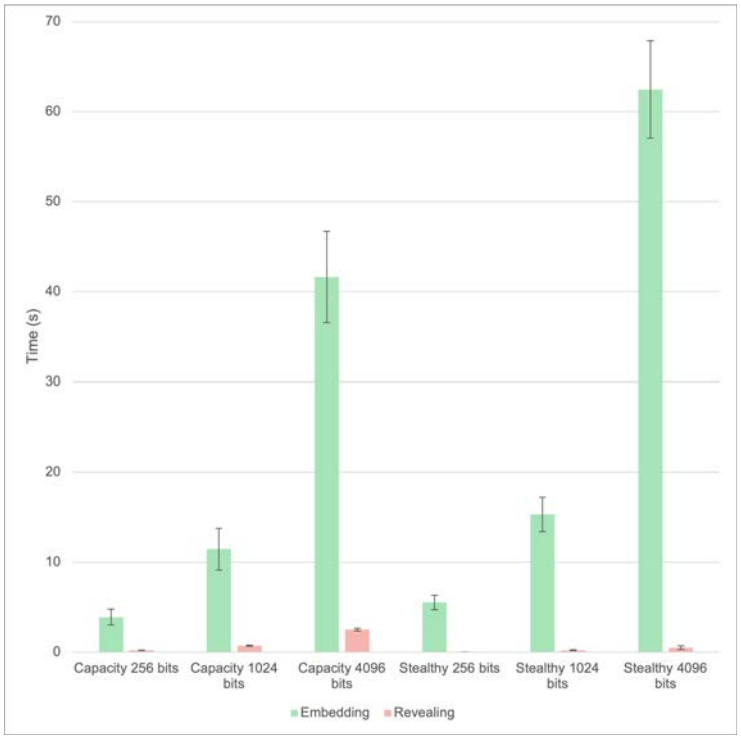

Figure 5.4: Embedding and revealing time per method

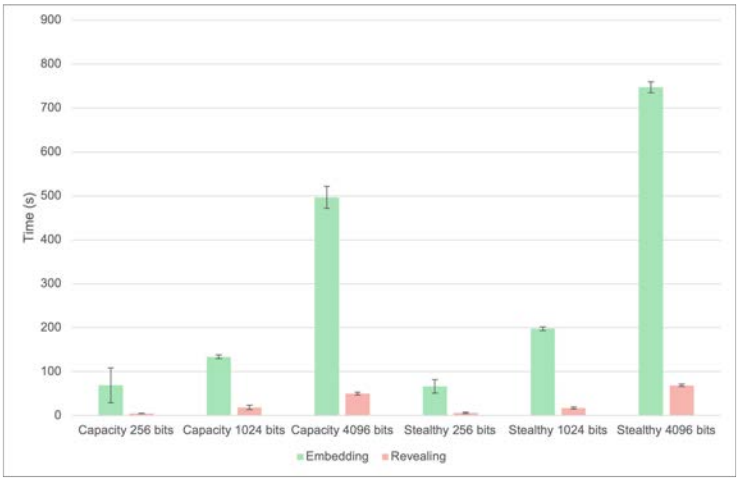

Figure 5.5: Network management time per method

5.5.3.2 Time assessment

The time taken by Smart-Zephyrus is divided in computation and network time. The former refers to the time required for secret preparation, retrieval and encryption, while the latter refers to sending the transaction to the blockchain, its retrieval, mining time and contract verification time on Etherscan. Although network time is out of the scope of Smart-Zephyrus because it depends on external factors, the real-world suitability analysis of the mechanism requires such time.

The embedding and revealing time for each method depending on the secret's size is presented in Fig. 5.4. Embedding times includes encryption and preparation of the message and its concealment in contracts. On the other hand, revealing time includes extracting the secret and decrypting it afterwards. The time of encryption and decryption is quite small regarding the total time, and very similar among the same message sizes. It ranges from 0.001 seconds to 0.0035 seconds. As expected, the size of the message directly affects the time of embedding and revealing. As the capacity method performs more operations and transformations to embed the message, it takes longer on the revealing whereas the stealthy needs more contracts to embed the same quantity of information and then, it takes longer. It is specially noticeable on the 4,096 bits case, where it takes an average of 41 seconds with a

standard deviation of 5.07 for the capacity method in the embedding versus 62 seconds and 5.40 for the stealthy one. Regarding the revealing times, it takes 2.56 seconds to reveal a 4,096 bit message for the capacity method with a standard deviation of 5.07 seconds versus 0.47 seconds with a standard deviation of 0.20 seconds for the stealthy method. These times can be considered very low as the applied scale has a maximum of 70 *s*.

Fig. 5.5 presents the time of embedding and revealing messages for each method. It can be observed that the network time, although it usually increases with the message size is largely dependent on the state of the network. Most of the time is consumed by the mining and verification process. For example, for the capacity method with 4096 bits of information the average mining time is 88 seconds, while the verification time is 383 seconds. Nonetheless, it should be noticed that verification times are similar to each other, with a standard deviation of 4.28 seconds, while the mining time has a standard deviation of 27 seconds.

5.6 Related work

Some works already embed information in smart contracts. Ken Shirriff (16) identified how to embed information in the blockchain or Bitcoin scripts, which can be considered analogous to smart contracts. In particular, the hash of the public key script (P2PKH) is used to insert information.

Also with the focus on Bitcoin, (236) presents different fields to hide messages without the use of encryption. In the case of Pay-to-Pubkey and Pay-to-PubKeyHash scripts the public key and the signature are used to embed information. In multi-signature scripts, keys located in the public key script are used to transmit data. Besides, A. Sward et al. (17) proposes different insertion methods in Bitcoin. Information is included in the public key in a Pay to Public Key (P2PK) transaction, or in the hash of the public key in a Pay to Script Hash (P2SH), being ScriptPubKey of a transaction used for this purpose. Information can be addition-

ally embedded in ScriptSig, which is the complementary part of a complete and valid transaction. R. Recabarren at al. (20) also proposes the use of ScriptSig for inserting information. They present Thitonious, an anti-censorship Bitcoin tool in which scriptSig of a P2SH multisignature transaction is used to embed a message on the 28 most significant bytes.

(297) proposes the use of Ethereum smart contracts and Bitcoin for information exchanges. Two types of smart contracts are defined: a voting contract which uses the OP_RETURN to transmit a hash, and applies the order and option of the voting addresses or the addresses themselves for embedding purposes; and a bidding contract which also uses the OP_RETURN to transmit a hash, and embeds data in the bids.

By contrast, just Basuki et al. (285) applies Ethereum. Their purpose is on hiding instructions for recovering secrets within images in the smart contract's timestamp, having 29 bits of capacity. Additionally, (298) proposes the use of Ethereum smart contracts using the bytecode, constructor arguments and the swarm hash to hide information. The maximum capacity in bits per transaction is 46, 160 and 256 respectively.

In sum, considering Table 5.5, Smart-Zephyrus is the only one who actually uses a high-level language to insert information in the blockchain providing also a high level of stealthiness (ALL) against the attacker models defined by (298), which is more powerful than most proposals (except for (20; 236; 297; 298)). In Smart-Zephyrus the capacity is approximately 42 bits for the capacity method and 32 for the stealthy one according to experimental results. Although the maximum capacity is lower than other works, it is still higher than in (236) for the ScriptSig, in Basuki et al. (285) and in Zephyrus for the Bytecode method (298). Furthermore, contracts are completely verifiable on Etherscan making them seem legitimate and increasing the level of trust in the ecosystem (35). On the other hand, the embedding procedure is compatible with other methods, like Zephyrus' Swarm hash

method or constructor arguments one.

Table 5.5: Smart contacts' embedding proposals

Reference	Technology	Method	Max capacity (bits)	Stealthiness	General mechanism	Use of high level language	Secret integrity	Comparison of embedded data with normal content in blockchain
(16)	Bitcoin	ScriptPubKey:paytopubkeyhash	160	BE	√	X	√	X
(236)	Bitcoin	ScriptSig	8	ALL	√	X	√	X
	Bitcoin	ScriptPubKey:paytoscripthash-multisig	2766	ALL	√	X	√	X
(17)	Bitcoin	ScriptPubKey:paytopublickey	520/264	BE,AE	√	X	√	X
	Bitcoin	ScriptPubKey:paytoscripthash	160	BE,AE	√	X	√	X
	Bitcoin	ScriptSign(Redeem script):paytoscripthash	4136	BE,AE	√	X	√	X
	Bitcoin	ScriptSign(Data input):DataDropwithoutSignature	13040	BE,AE	√	X	√	X
	Bitcoin	ScriptSign(Data input):DataDropwithSignature	12232	BE,AE	√	X	√	X
	Bitcoin	ScriptSign(Data input):DataHashwithoutSignature	12480	BE,AE	√	X	√	X
	Bitcoin	ScriptSign(Data input):DataHashwithSignature	11688	BE,AE	√	X	√	X
(20)	Bitcoin	ScriptSign:multisignature	448	BE,AE/ALL	√	X	√	X
(297)	Bitcoin, Ethereum	OP_RETURN + sender address/order	640 (hash of key words), 32 (message)	BE, AE/ALL (encryption)	X	X	√	X
(285)	Ethereum	Timestamp in smart contract (Function call with string)	29	BE	√	X	√	X
(298)	Ethereum	Constructor arguments	160	ALL	√	X	√	√
	Ethereum	Bytecode	46	AE	√	X	√	√
	Ethereum	Swarm hash	256	ALL	√	X	√	√
Smart-Zephyrus	Ethereum	Solidity contract code	42/32	ALL	√	√	√	

Part IV

Conclusions

Chapter 6

Conclusions

This Chapter contains the thesis conclusions and final remarks, and summarizes the contributions achieved. A critical discussion on the developed work is also presented. Additionally, future research directions that derive from the thesis results are proposed.

6.1 Conclusions and summary of contributions

The work developed in this thesis has been focused on providing a systematic review on how blockchain can be used to provide cybersecurity as well as how some of its intrinsic characteristics can be exploited for other purposes. Furthermore, two tools, Zephyrus and Smart-Zephyrus have been developed as a proof of concept of how these characteristics can be leveraged.

Blockchain-based approaches to provide with cybersecurity guarantees have rocketed in the last years. It has been shown that blockchain is an enabling technology that is paving the way for smarter, enriched services. Our analysis shows the prevalence of Ethereum. A worrisome fact is that there is a fraction of academic papers that are using blockchain disregarding (or at least not providing evidences of satisfaction of) all principles that justify its use.

Blockchain features also make it interesting for other purposes. On the one hand, blockchain permanent availability makes it an appealing feature to build covert communication on top of it. On the other hand, blockchain also provides certain characteristics that malware can use to improve their attacks, i.e. permanent

availability and immutability.

In this vein, we have studied how this technology has been used by different types of malware and presented a comprehensive analysis from different perspectives. In this research, we have found that Ethereum is the second most used technology, only behind Bitcoin and mostly due to the fact that the blockchain is mostly used as a payment method for ransomware. Among other open research issues, it has been found that the use of covert communication channels has not been explored in this area.

To explore the suitability of Ethereum for covert communications, we designed two mechanisms. Although other technologies have used blockchain before for this purpose, no previous work has considered Ethereum and all its fields to embed information. Ethereum is not only the second most used cryptocurrency by market cap (8) but also it is the technology that prevails when blockchain is used to provide cybersecurity as our study has demonstrated.

First, Zephyrus, a mechanism to hide information in Ethereum fields (including smart-contracts) has been proposed. An open-source implementation has been released to foster further research in this area. Our results show that some information can be concealed in most transaction fields while remaining stealthy if some limits are observed. Moreover, cost and time incurred have been characterized, supporting the real-world suitability of this proposal.

But Zephyrus has a limitation – it does not leverage the smart-contract high-level language. In order to tackle this issue, another technique (called Smart-Zephyrus) has been proposed. It uses a high-level smart contract language (Solidity) to insert information achieving a high level of stealthiness and thus reducing the chance for an attacker to be detected. The time and cost for the attacker have also been characterized.

6.2 Critical analysis on the developed work

Blockchain is a rather new technology, and therefore, it can change very quickly in a short period of time. Therefore, it can be questioned the significance of this work, specially of the study that has been carried out. However, this thesis not only presents current data, but also analyzes trends and proposes open issues and future lines of research that can be used as a base for future research works.

Both proof of concepts developed in this thesis use Ethereum as a means of providing a covert channel. A question arises on whether the use of exclusively this technology would limit the relevance of this work. However, the same approach taken to develop these tools can be adapted, to some extent, to use any other blockchain-related technology. Furthermore, both Zephyrus and Smart-Zephyrus are completely compatible with other EVM compatible chains like Avalanche, Binance Smart Chain or Polygon (299).

Regarding introducing information in the Solidity code, it can be argued that the information does not live on chain per se, and therefore can be deleted. In this work, Etherscan has been used as a means to store the code as it is the original and most important block explorer and the default place to retrieve contract code(300). However, this code also can be uploaded on a distributed file system, like IPFS, which would prevent the information to be lost in case Etherscan or any other block explorer is down.

Another acceptable critic is the cost of inserting information on the blockchain. Indeed, our most expensive mechanism (Smart-Zephyrus stealthy mode) cost $6,088 for 4096 bits. This cost can be seen as expensive. However, from the point of view of an attacker and as explained in Section 5.5.3.1, traditional attack methods also incur a cost for the attacker (252). Maintaining bots can cost up to $62k (296) while a ransomware like Wannacry was estimated to have a revenue of 54.43228033 BTC ($1,041,833.84 in current value). Considering this data, the cost of the mechanism seems to be affordable. Furthermore, even with that in mind, the cost can

be reduced by using one of the previously mentioned compatible chains. This is specially important from the data exfiltration point of view in which the sender may gain nothing or would not have to pay anything if they were using other method. Indeed, the cost of deploying an ERC20 Token contract in Avalanche is 0.479938 AVAX, which are $6.15 in current value, while in Ethereum can be more than $170 (301).

Capacity is usually another issue related to steganographic systems. Cover objects usually have a limited capacity (302). Furthermore, depending on the mechanism the final capacity can be different even when using the same cover object. In this thesis, we were able to insert up to 40k bits of information using Zephyrus. It is still a low capacity for the cost. However, the immutability and resiliency of the mechanism should be also taken into account.

6.3 Challenges and future research lines

The work developed in this thesis opens up the door to several innovative research lines (with their associated challenges), which are mainly focused on complementing the approach or even extending it to other related areas.

The analysis on the risks posed by external technologies, like authentication authorities or service providers, when interacting with the blockchain needs to be further explored. It must be noted that the direct or indirect interaction among technologies may be helpful to develop novel attacks. This may be extremely relevant in critical infrastructures such as industrial facilities.

The need of the development of a taxonomy to choose the right type of blockchain according to the area and desired cybersecurity properties could be another future research line. For instance, a public blockchain can be specially useful in e-commerce, while a private one could be more appropriate in health applications. Combining this matter with cybersecurity technologies, a semaphore-like scheme could be created to easily represent the actual guarantees provided by a

proposal. This is in line with current practices, such as the privacy 'nutrition label' required by Apple to app developers (303). This scheme might be developed leveraging current taxonomies on decentralized technologies, such as the one proposed by Samer Hasan *et al.* (304).

Moreover, an analysis of computationally efficient techniques to provide cybersecurity could also be an interesting line of future work, as current techniques are usually computationally costly. In this thesis we have seen techniques like homomorphic encryption algorithms in order to achieve confidentiality, but this type of algorithms is computationally costly (305) and other alternative could be preferable.

Furthermore, blockchain technology is not isolated. Therefore, an analysis of the provision of cybersecurity properties concerning laws and regulations in different countries could be necessary.As several traditional services, such as identity management or public notaries may leverage blockchains, achieving cybersecurity properties may not only be advisable but even forced by upcoming legislations. This is particularly relevant for pseudoanonymization and data confidentiality in relation with regulations such as the European General Data Protection Regulation (306), among others, should be carefully studied and considered accordingly.

The need of a unified criteria to use blockchain technologies can be inferred. There are different authors that analyze when a blockchain is necessary. In this thesis Greenspan criteria are used for being well-known, but there are others like the framework in (307), the steps proposed in (308), or the set of questions created by Nitish Singh (309) that allow choosing the type of blockchain. Given the current widespread use of blockchain technology, the definition of common criteria about when and how to use this technology would help researchers and companies in the development of products and systems which really need a blockchain.

Regarding the developed tools, control structures can be adapted to efficiently support multi-field usage and combination between Zephyrus and Smart-Zephyrus.

This would lead to a bigger capacity with the same costs. Open and interactive channel communications can also be implemented, by letting Zephyrus or Smart-Zephyrus scan the network and automatically retrieve content that fulfil certain characteristics, for example transactions sent from a certain address. Furthermore, adaptive steganographic technique can be considered to improve capacity considering the existing Ethereum contents, for example adapting the quantity of bits according to the normal values of a single contract or for a specific period of time. Indeed, transactions from Zephyrus and Smart-Zephyrus are currently sent to the blockchain in very low intervals of time which could raise suspicions. This issue can be solved by analysing how transactions are sent to the network in terms of time and frequency. In this way a model could be created and a timer introduced in the implementation that sends transactions based on the studied distribution. Also, the use of other languages for smart contracts (e.g., Vyper) would also contribute to characterize the generalization of this technique.

Last but not least, the development of detection techniques against this type of covert communication is also needed. Content in the blockchain is immutable and cannot be deleted. Therefore, intercepting or detecting the communication could be crucial when an attack takes place. In Zephyrus and Smart-Zephyrus content is formatted according to blockchain normal values, so an statistic attack would generally not be successful. However, there can be circumstances in which the current values of the blockchain could differ from the standard values, for example, if there is a drastic drop or an increase on the value of Ether or for a specific contract that works differently from the observed ones. An active eavesdropper could also be able to detect some of the steganographic techniques, for example by debugging and executing the bytecode modified by Zephyrus and realizing that instructions are either not executed or do nothing. Automating this process could also be a possible challenge.

Part V

Bibliography and appendices

Bibliography

[1] Andrews, Jeffrey G and Buzzi, Stefano and Choi, Wan and Hanly, Stephen V and Lozano, Angel and Soong, Anthony CK and Zhang, Jianzhong Charlie. What will 5G be? IEEE Journal on sel areas in comms. 2014;32(6):1065–1082.

[2] Kshetri, Nir. Can blockchain strengthen the internet of things? IT professional. 2017;19(4):68–72.

[3] Swan, Melanie. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc."; 2015.

[4] Beck, Roman and Stenum Czepluch, Jacob and Lollike, Nikolaj and Malone, Simon. Blockchain–the gateway to trust-free cryptographic transactions. 2016;.

[5] Iansiti, Marco and Lakhani, Karim R. The truth about blockchain. Harvard Business Review. 2017;95(1):118–127.

[6] Nakamoto, Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008;.

[7] Karame, Ghassan O and Androulaki, Elli and Capkun, Srdjan. Double-spending fast payments in bitcoin. In: Proc. 2012 ACM CCS. ACMp.906–917.

[8] Bagshaw, Rick. Top 10 cryptocurrencies by market capitalisation;. Last access April 2021. `https://coinrivet.com/top-10-cryptocurrencies-by-market-capitalisation/`.

[9] Leng, Jiewu and Jiang, Pingyu and Xu, Kailin and Liu, Qiang and Zhao, J Leon and Bian, Yiyang and Shi, Rui. Makerchain: A blockchain with chemical signature for self-organizing process in social manufacturing. Journal of Cleaner Production. 2019;234:767–778.

[10] Dhillon, Vikram and Metcalf, David and Hooper, Max. Blockchain in Health Care. In: Blockchain Enabled Applications. Springerp.125–138.

[11] Jindal, Anish and Aujla, Gagangeet Singh and Kumar, Neeraj. SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment. Computer Networks. 2019;153:36–48.

[12] There's No Good Reason to Trust Blockchain Technology; 2019). (Last access Feb. 2021. Available from: `https://www.wired.com/story/theres-no-good-reason-to-trust-blockchain-technology/`.

[13] Katzenbeisser, Stefan and Petitcolas, Fabien. Information hiding techniques for steganography and digital watermarking. Artech house; 2000.

[14] Cuff, Paul and Zhao, Lei. Coordination using implicit communication. arXiv preprint arXiv:11083652. 2011;.

[15] Abuadbba, Alsharif and et al. . Robust privacy preservation and authenticity of the collected data in cognitive radio network—Walsh–Hadamard based steganographic approach. Pervasive and Mobile Computing. 2015;22.

[16] Shirriff, Ken. Hidden surprises in the Bitcoin blockchain and how they are stored.;. Last access April 2021. `http://www.righto.com/2014/02/ascii-bernanke-\wikileaks-photographs.html#ref6`.

[17] Sward, Andrew and Vecna, Ivy and Stonedahl, Forrest. Data insertion in Bitcoin's Blockchain. Ledger. 2018;3.

[18] Matzutt, Roman and Hiller, Jens and Henze, Martin and Ziegeldorf, Jan Henrik and Müllmann, Dirk and Hohlfeld, Oliver and Wehrle, Klaus. A quantitative analysis of the impact of arbitrary blockchain content on bitcoin;. .

[19] Ding, Feng. Broadcasting Steganography in the Blockchain. vol.. 12022. Springer Naturep.256.

[20] Ruben Recabarren and Bogdan Carbunar. Tithonus: A Bitcoin Based Censorship Resilient System. Proceedings on Privacy Enhancing Technologies. 2019;2019(1).

[21] Liu, Shaoyuan and Fang, Zhi and Gao, Feng and Koussainov, Bakh and Zhang, Zijian and Liu, Jiamou and Zhu, Liehuang. Whispers on Ethereum: Blockchain-based Covert Data Embedding Schemes. In: Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructurep.171–179.

[22] Zhang, Lejun and Zhang, Zhijie and Jin, Zilong and Su, Yansen and Wang, Zhuzhu. An approach of covert communication based on the Ethereum whisper protocol in blockchain. International Journal of Intelligent Systems. 2021;36(2):962–996.

[23] Kane, Ethan. Is Blockchain a General Purpose Technology? SSRN. 2017;.

[24] Fenton, Andrew. Almost half of bitcoin payments are now made on the darknet; 2019. Available from: `https://micky.com.au/almost-half-of-bitcoin-payments-are-now-made-on-the-darknet/`.

[25] Pletinckx, Stijn and Trap, Cyril and Doerr, Christian. Malware Coordination using the Blockchain: An Analysis of the Cerber Ransomware. In: 2018 IEEE Conference on Communications and Network Security (CNS)p.1–9.

[26] Aidan, Jagmeet Singh and Verma, Harsh Kumar and Awasthi, Lalit Kumar. Comprehensive survey on petya ransomware attack. In: 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS). IEEEp.122–125.

[27] Eisenkraft, Kobi and Olshtein, Arie. Pony's C&C servers hidden inside the Bitcoin blockchain. Technical Report. Check Point. https://research. checkpoint. com/2019/ponys . . . ; 2019.

[28] Ilascu, Ionut. New botnet hides in Blockchain DNS mist and removes Cryptominer. BleepingComputer; 2018. Available from: `https://www.bleepingcomputer.com/news/security/new-botnet-hides-in-blockchain-dns-mist-and-removes-cryptominer/`.

[29] Report: Malware poisons One-third of world's computers; 2014. Available from: https://www.technewsworld.com/story/report-malware-poisons-one-third-of-worlds-computers-80707.html.

[30] Ventures, Cybersecurity. Cybercrime damages are predicted to cost the world $6 trillion annually by 2021; 2018. Available from: https://www.prnewswire.com/news-releases/cybercrime-damages-are-predicted-to-cost-the-world-6-trillion-annually-by-2021-300540158.html.

[31] Kshetri, Nir and Voas, Jeffrey. Do Crypto-Currencies Fuel Ransomware? IT Professional. 2017;19(5):11–15.

[32] Böck, Leon and Alexopoulos, Nikolaos and Saracoglu, Emine and Mühlhäuser, Max and Vasilomanolakis, Emmanouil. Assessing the threat of blockchain-based botnets. In: 2019 APWG Symposium on Electronic Crime Research (eCrime). IEEEp.1–11.

[33] draglet. Smart Contracts Explained;. Last access April 2021. https://www.draglet.com\/blockchain-services/smart-contracts/.

[34] Etherscan;. Last access April 2021. https://etherscan.io/.

[35] Lukic, Milica. 5 important reasons to verify smart contracts - how to do it. Blog — Tenderly; 2022. Available from: http://blog.tenderly.co/guide-to-smart-contract-verification-methods/.

[36] Axon, LM and Goldsmith, Michael. PB-PKI: a privacy-aware blockchain-based PKI. 2016;.

[37] Wood,Gavin. Ethereum Yellow Paper. 2019 mar;Available from: https://ethereum.github.io/yellowpaper/paper.pdf.

[38] Wood,Gavin. PoA Private Chains. 2015 nov;Available from: https://github.com/ethereum/guide/blob/master/poa.md.

[39] Bovaird, Charles. What to know before trading Monero. CoinDesk; 2017. Available from: https://www.coindesk.com/markets/2017/05/28/what-to-know-before-trading-monero/.

[40] Frankenfield, Jake. What is iota (miota)? definition, how it works, and concerns. Investopedia; 2022. Available from: https://www.investopedia.com/terms/i/iota.asp.

[41] Frankenfield, Jake. Tangle (iota): What it means, how it works. Investopedia; 2023. Available from: https://www.investopedia.com/terms/t/tangle-cryptocurrency.asp.

[42] Gao, Haoyu and Li, Leixiao and Chang, Xiangyang and Wan, Jianxiong and Li, Jie and Du, Jinze and Zhang, Xiaoxu. BlockchainBot: A Novel Botnet Infrastructure Enhanced by Blockchain Technology and IoT. Electronics. 2022;11(7):1065.

[43] Guo, Fengyang and Xiao, Xun and Hecker, Artur and Dustdar, Schahram. A Theoretical Model Characterizing Tangle Evolution in IOTA Blockchain Network. IEEE Internet of Things Journal. 2022;.

[44] Bhandary, Mohan and Parmar, Manish and Ambawade, Dayanand. A blockchain solution based on directed acyclic graph for IoT data security using IoTA tangle. In: 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEEp.827–832.

[45] Zbruceli, Lukas. All-in-One NKN FAQ; 2019. Available from: https://forum.nkn.org/t/all-in-one-nkn-faq/164.

[46] Daly, Lyle. What is dash cryptocurrency?. The Motley Fool; 2021. Available from: https://www.fool.com/investing/stock-market/market-sectors/financials/cryptocurrency-stocks/dash-cryptocurrency/.

[47] Dashpay. Whitepaper · dashpay/dash wiki;. Available from: https://github.com/dashpay/dash/wiki/Whitepaper.

[48] Hyperledger; Last access Feb. 2021. Available from: https://www.hyperledger.org/use.

[49] Permissioned Blockchain: 5 Hyperledger Projects in Depth; 2018). (Last access Feb. 2021. Available from: https://4irelabs.com/5_hyperledger_projects_in_depth.

[50] Hyperledger chaincode tutorials; Last access Feb. 2021. Available from: https://hyperledger-fabric.readthedocs.io/en/release-1.3/chaincode.html.

[51] Ray, Shaan. Blockchains: The Technology of Transactions. Towards Data Science; 2021. Available from: https://towardsdatascience.com/blockchains-the-technology-of-transactions-9d40e8e41216.

[52] What is the Ethereum Transaction Data Structure?; 1964. Available from: https://ethereum.stackexchange.com/questions/1990/what-is-the-ethereum-transaction-data-structure.

[53] OP_RETURN;. Available from: https://en.bitcoin.it/wiki/OP_RETURN.

[54] Moneropedia: Payment ID;. Available from: https://www.getmonero.org/resources/moneropedia/paymentid.html.

[55] Srivastav, Kaushiki. A guide to blockchain immutability and challenges - dzone security. DZone; 2021. Available from: https://dzone.com/articles/a-guide-to-blockchain-immutability-and-chief-chall.

[56] EC-Council. What is blockchain immutability and how does it help?; 2021. Available from: https://blog.eccouncil.org/what-is-blockchain-immutability-and-how-does-it-help/.

[57] What is Decentralization in Blockchain?. Oberbaumpresse;. Available from: https://aws.amazon.com/es/blockchain/decentralization-in-blockchain/.

[58] Managing data availability;. Available from: https://www1.udel.edu/security/data/availability.html#:~:text=Data%20availability%20is%20about%20the,considered%20supplementary%20rather%20than%20necessary.

[59] Weber, Ingo and Gramoli, Vincent and Ponomarev, Alex and Staples, Mark and Holz, Ralph and Tran, An Binh and Rimba, Paul. On Availability for Blockchain-Based Systems. In: 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)p.64–73.

[60] Adrian, Michal. Is cryptocurrency anonymous? the myth of anonymity debunked;. Available from: https://www.ulam.io/blog/is-cryptocurrency-anonymous/.

[61] Ramani, Vidhya and Kumar, Tanesh and Bracken, An and Liyanage, Madhusanka and Ylianttila, Mika. Secure and Efficient Data Accessibility in Blockchain based Healthcare Systems. In: 2018 IEEE Global Communications Conference (GLOBECOM). IEEEp.206–212.

[62] Exploding Costs of storing information on a Blockchain. 2017) (Last access Feb 2021;Available from: https://content-blockchain.org/newsarchive/2017/07/20/exploding-costs-of-storing-data-on-a-blockchain/.

[63] Distributed Hash Table. In: . Boston, MA: Springer USp.903–904.

[64] Stavrou, Angelos and Voas, Jeffrey. Verified time. Computer. 2017;50(3):78–82.

[65] Paulsen, Celia and Paulsen, Celia and Toth, Patricia. Small business information security: The fundamentals. NIST; 2016.

[66] Grassi, Paul A and Garcia, M and Fenton, J. DRAFT NIST Special Publication 800-63-3 Digital Identity Guidelines. NIST. 2017;.

[67] Barker, Elaine and Barker, William and Burr, William and Polk, William and Smid, Miles. Nist special publication 800-57. NIST Special publication. 2007;800(57):1–142.

[68] Greenspan, Gideon. Avoiding the pointless blockchain project. 2015;Available from: https://www.multichain.com/blog/2015/11/avoiding-pointless-blockchain-project/.

[69] Ethereum Foundation. Ethereum. Blockchain application platform.;. Last access April 2021. https://www.ethereum.org/.

[70] Zheng, Zibin and Xie, Shaoan and Dai, Hongning and Chen, Xiangping and Wang, Huaimin. An overview of blockchain technology: Architecture, consensus, and future trends. In: Big Data, 2017 IEEE Intl. Congress on. IEEEp.557–564.

[71] Binance Academy. Proof of work (POW) vs. Proof of Stake (POS). Binance Academy; 2022. Available from: https://academy.binance.com/en/articles/proof-of-work-vs-proof-of-stake.

[72] Wood, Gavin. Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper. 2014;151:1–32.

[73] Buterin, Vitalik and others. A next-generation smart contract and decentralized application platform. white paper (2014). 2014;.

[74] Abi spec;. Last access April 2021. https://solidity.readthedocs.io/en/develop/abi-spec.html.

[75] Johnson, Don and Menezes, Alfred and Vanstone, Scott. The Elliptic Curve Digital Signature Algorithm. Intl jnl of information security. 2001;1(1):36–63.

[76] Cano-Benito, Juan and Cimmino, Andrea and García-Castro, Raúl. Toward the Ontological Modeling of Smart Contracts: A Solidity Use Case. IEEE Access. 2021;9:140156–140172.

[77] Cvllr, Jean. Solidity tutorial: All about functions. Medium; 2021. Available from: https://jeancvllr.medium.com/solidity-tutorial-all-about-functions-dba2ccb1e931.

[78] Soldiity Types;. Available from: https://docs.soliditylang.org/en/v0.8.10/types.html.

[79] Solidity metadata specification;. Last access April 2021. https://solidity.readthedocs.io/en/v0.5.10/metadata.html.

[80] Victor, Friedhelm and Lüders, Bianca Katharina. Measuring ethereum-based erc20 token networks. In: International Conference on Financial Cryptography and Data Security. Springerp.113–129.

[81] Somin, Shahar and Gordon, Goren and Altshuler, Yaniv. Network analysis of erc20 tokens trading on ethereum blockchain. In: International Conference on Complex Systems. Springerp.439–450.

[82] What is OpenZeppelin? The Ultimate Guide " moralis " The ultimate web3 development platform; 2021. Available from: https://moralis.io/what-is-openzeppelin-the-ultimate-guide/.

[83] Understanding covert channels of communication;. Available from: https://www.isaca.org/resources/news-and-trends/isaca-now-blog/2017/understanding-covert-channels-of-communication.

[84] Westfeld, Andreas and Pfitzmann, Andreas. Attacks on steganographic systems. In: International workshop on information hiding. Springerp.61–76.

[85] Anderson, Ross J and Petitcolas, Fabien AP. On the limits of steganography. IEEE Journal on selected areas in communications. 1998;16(4):474–481.

[86] Chang, Chin-Chen and Hsiao, Ju-Yuan and Chan, Chi-Shiang. Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy. Pattern Recognition. 2003;36(7):1583–1595.

[87] Archiveddocs. Defining malware: FAQ;. Available from: https://docs.microsoft.com/en-us/previous-versions/tn-archive/dd632948(v=technet.10)?redirectedfrom=MSDN.

[88] Ransomware: What is ransomware: Ransomware attack;. Available from: https://www.malwarebytes.com/ransomware.

[89] Ransomware;. Available from: https://www.trendmicro.com/vinfo/us/security/definition/ransomware.

[90] Belcic, Ivan. Avast; 2021. Available from: https://www.avast.com/c-botnet?redirect=1.

[91] Kaspersky. What are bots? – definition and explanation; 2021. Available from: https://www.kaspersky.com/resource-center/definitions/what-are-bots.

[92] Radware. Botmaster;. Available from: https://www.radware.com/security/ddos-knowledge-center/ddospedia/botmaster/.

[93] Jadhav, Suyash and Dutia, Shobhit and Calangutkar, Kedarnath and Oh, Tae and Kim, Young Ho and Kim, Joeng Nyeo. Cloud-based android botnet malware detection system. In: 2015 17th International Conference on Advanced Communication Technology (ICACT). IEEEp.347–352.

[94] Vengatesan, K and Kumar, Abhishek and Parthibhan, M and Singhal, Achintya and Rajesh, R. Analysis of Mirai botnet malware issues and its prediction methods in internet of things. In: International conference on Computer Networks, Big data and IoT. Springerp.120–126.

[95] Briner, Rob and Denyer, David. In: Systematic Review and Evidence Synthesis as a Practice and Scholarship Toolp.112–129.

[96] dblp computer science bibliography; Last access Feb. 2021. Available from: https://dblp.uni-trier.de/.

[97] Clarivate. Journal Citation Reports; 2021. .

[98] GII-GRIN-SCIE. The GII-GRIN-SCIE (GGS) Conference Rating; 2021. .

[99] Rahman, Md Abdur and Hassanain, Elham and Rashid, Md Mamunur and Barnes, Stuart J and Hossain, M Shamim. Spatial blockchain-based secure mass screening framework for children with dyslexia. IEEE Access. 2018;6:61876–61885.

[100] Malik, Nisha and Nanda, Priyadarsi and Arora, Arushi and He, Xiangjian and Puthal, Deepak. Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks. In: IEEE Intl. Conf. On Trust, Security And Privacy In Computing And Communications. IEEEp.674–679.

[101] Mikula, Tomas and Jacobsen, Rune Hylsberg. Identity and access management with blockchain in electronic healthcare records. Proceedings - 21st Euromicro Conference on Digital System Design, DSD 2018. p.699–706.

[102] Li, Dongxing and Peng, Wei and Deng, Wenping and Gai, Fangyu. A blockchain-based authentication and security mechanism for IoT. Proceedings - International Conference on Computer Communications and Networks, ICCCN. 2018;2018-July:1–6.

[103] Tsolakis, Apostolos C and Moschos, Ioannis and Votis, Konstantinos and Ioannidis, Dimosthenis and Dimitrios, Tzovaras and Pandey, Pankai and Katsikas, Sokratis and Kotsakis, Evangelos and García-Castro, Raúl. A Secured and Trusted Demand Response system based on Blockchain technologies. In: 2018 Innovations in Intelligent Systems and Applications (INISTA). IEEEp.1–6.

[104] Mell, Peter. Managed Blockchain Based Cryptocurrencies with Consensus Enforced Rules and Transparency. In: IEEE Intl. Conf. On Trust, Security And Privacy In Computing And Communications. IEEEp.1287–1296.

[105] Heilman, Ethan and Baldimtsi, Foteini and Goldberg, Sharon. Blindly Signed Contracts: Anonymous On-Blockchain and Off-Blockchain Bitcoin Transactions. p.43–60.

[106] Yang, Zhe and Yang, Kan and Lei, Lei and Zheng, Kan and Leung, Victor CM. Blockchain-based decentralized trust management in vehicular networks. IEEE Internet of Things Journal. 2018;6(2):1495–1505.

[107] Liang, Xueping and Shetty, Sachin and Tosh, Deepak and Ji, Yafei and Li, Danyi. Towards a Reliable and Accountable Cyber Supply Chain in Energy Delivery System Using Blockchain. In: Intl. Conf. on Security and Privacy in Communication Systems. Springerp.43–62.

[108] Wang, Jingzhong and Li, Mengru and He, Yunhua and Li, Hong and Xiao, Ke and Wang, Chao. A blockchain based privacy-preserving incentive mechanism in crowdsensing applications. IEEE Access. 2018;6:17545–17556.

[109] Ibrahim, Maged Hamada. SecureCoin: A Robust Secure and Efficient Protocol for Anonymous Bitcoin Ecosystem. IJ Network Security. 2017;19(2):295–312.

[110] Andrychowicz, Marcin and Dziembowski, Stefan and Malinowski, Daniel and Mazurek, Lukasz. Secure multiparty computations on bitcoin. In: 2014 IEEE Symposium on Security and Privacy. IEEEp.443–458.

[111] Lu, Zhaojun and Liu, Wenchao and Wang, Qian and Qu, Gang and Liu, Zhenglin. A privacy-preserving trust model based on blockchain for VANETs. IEEE Access. 2018;6:45655–45664.

[112] Liu, Yuke and Zhang, Junwei and Gao, Qi. A Blockchain-Based Secure Cloud Files Sharing Scheme with Fine-Grained Access Control. In: 2018 International Conference on Networking and Network Applications (NaNA). IEEEp.277–283.

[113] Zhu, Yan and Qin, Yao and Gan, Guohua and Shuai, Yang and Chu, William Cheng-Chung. TBAC: transaction-based access control on blockchain for resource

sharing with cryptographically decentralized authorization. vol.. 1. IEEEp.535–544.

[114] Xue, Jingting and Xu, Chunxiang and Zhang, Yuan. Private blockchain-based secure access control for smart home systems. KSII Transactions on Internet and Information Systems. 2018;12(12):6057–6078.

[115] L. Xie and Y. Ding and H. Yang and X. Wang. Blockchain-Based Secure and Trustworthy Internet of Things in SDN-Enabled 5G-VANETs. IEEE Access. 2019;7:56656–56666.

[116] Weber, Ingo and Xu, Xiwei and Riveret, Régis and Governatori, Guido and Ponomarev, Alexander and Mendling, Jan. Untrusted business process monitoring and execution using blockchain. In: Intl. Conf. on Business Process Management. Springerp.329–347.

[117] Kfoury, Elie F and Khoury, David J. Secure end-to-end volte based on ethereum blockchain. In: 2018 41st International Conference on Telecommunications and Signal Processing (TSP). IEEEp.1–5.

[118] Zyskind, Guy and Nathan, Oz and Pentland, Alex Sandy. Decentralizing privacy: Using blockchain to protect personal data. Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015. p.180–184.

[119] S. Wang and R. Pei and Y. Zhang. EIDM: A Ethereum-Based Cloud User Identity Management Protocol. IEEE Access. 2019;7:115281–115291.

[120] Ali, Saqib and Wang, Guojun and Bhuiyan, Md Zakirul Alam and Jiang, Hai. Secure Data Provenance in Cloud-Centric Internet of Things via Blockchain Smart Contracts. In: IEEE SmartWorld. IEEEp.991–998.

[121] H. R. Hasan and K. Salah. Combating Deepfake Videos Using Blockchain and Smart Contracts. IEEE Access. 2019;7:41596–41606.

[122] Shahzad, Basit and Crowcroft, Jon. Trustworthy electronic voting using adjusted blockchain technology. IEEE Access. 2019;7:24477–24488.

[123] Gai, Keke and Wu, Yulu and Zhu, Liehuang and Qiu, Meikang and Shen, Meng. Privacy-preserving energy trading using consortium blockchain in smart grid. IEEE Transactions on Industrial Informatics. 2019;15(6):3548–3558.

[124] W. She and Q. Liu and Z. Tian and J. Chen and B. Wang and W. Liu. Blockchain Trust Model for Malicious Node Detection in Wireless Sensor Networks. IEEE Access. 2019;7:38947–38956.

[125] H. Liu and Y. Zhang and S. Zheng and Y. Li. Electric Vehicle Power Trading Mechanism Based on Blockchain and Smart Contract in V2G Network. IEEE Access. 2019;7:160546–160558.

[126] Li, Shuai and Liu, Meilin and Wei, Songjie. A distributed authentication protocol using identity-based encryption and blockchain for LEO network. In: Intl. Conf. on Security, Privacy and Anonymity in Computation, Communication and Storage. Springerp.446–460.

[127] Decusatis, Casimer and Lotay, Kulvinder. Secure, Decentralized Energy Resource Management Using the Ethereum Blockchain. IEEE Intl Conf on Trust, Security and Privacy in Computing and Communications. p.1907–1913.

[128] Heilman, Ethan and Alshenibr, Leen and Baldimtsi, Foteini and Scafuro, Alessandra and Goldberg, Sharon. Tumblebit: An untrusted bitcoin-compatible anonymous payment hub. In: Network and Distributed System Security Symposium; 2017. .

[129] Ozyilmaz, Kazim Rifat and Yurdakul, Arda. Designing a Blockchain-based IoT with Ethereum, swarm, and LoRa: the software solution to create high availability with minimal security risks. IEEE Consumer Electronics Magazine. 2019;8(2):28–34.

[130] Frankenfield, Jake. Block Header (Cryptocurrency). Investopedia; 2019). (Last access Feb. 2021. Available from: `https://www.investopedia.com/terms/b/block-header-cryptocurrency.asp`.

[131] Gao, Jianbin and Asamoah, Kwame Omono and Sifah, Emmanuel Boateng and Smahi, Abla and Xia, Qi and Xia, Hu and Zhang, Xiaosong and Dong, Guishan. Gridmonitoring: Secured sovereign blockchain based monitoring on smart grid. IEEE Access. 2018;6:9917–9925.

[132] Wang, Shangping and Zhang, Yinglong and Zhang, Yaling. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. IEEE Access. 2018;6:38437–38450.

[133] Augot, Daniel and Chabanne, Hervé and Clémot, Olivier and George, William. Transforming face-to-face identity proofing into anonymous digital identity using the bitcoin blockchain. In: 15th Annual Conf. on Privacy, Security and Trust (PST). IEEE; 2017. .

[134] Sui, Zhimei and Lai, Shangqi and Zuo, Cong and Yuan, Xingliang and Liu, Joseph K and Qian, Haifeng. An Encrypted Database with Enforced Access Control and Blockchain Validation. In: International Conference on Information Security and Cryptology. Springerp.260–273.

[135] Dagher, Gaby G and Marella, Praneeth Babu and Milojkovic, Matea and Mohler, Jordan. BroncoVote: Secure Voting System using Ethereum's Blockchain. 2018;.

[136] Bendiab, Keltoum and Kolokotronis, Nicholas and Shiaeles, Stavros and Boucherkha, Samia. WiP: A novel blockchain-based trust model for cloud identity management. In: Intl Conf on Dependable, Autonomic and Secure Computing. IEEEp.724–729.

[137] Lin, Di and Tang, Yu. Blockchain Consensus Based User Access Strategies

in D2D Networks for Data-Intensive Applications. IEEE Access. 2018;6:72683–72690.

[138] Stallings, W. . Cryptography and Network Security: Principles and Practice. Prentice Hall; 1999.

[139] A. Dorri and M. Steger and S. S. Kanhere and R. Jurdak. BlockChain: A Distributed Solution to Automotive Security and Privacy. IEEE Communications Magazine. 2017 Dec;55(12):119–125.

[140] Kanza, Yaron and Safra, Eliyahu. Cryptotransport: blockchain-powered ride hailing while preserving privacy, pseudonymity and trust. In: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACMp.540–543.

[141] Ruffing, Tim and Moreno-Sanchez, Pedro. ValueShuffle: Mixing confidential transactions for comprehensive transaction privacy in bitcoin. In: International Conference on Financial Cryptography and Data Security. Springerp.133–154.

[142] Shen, Meng and Tang, Xiangyun and Zhu, Liehuang and Du, Xiaojiang and Guizani, Mohsen. Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities. IEEE Internet of Things Journal. 2019;6(5):7702–7712.

[143] Lin, Chao and He, Debiao and Huang, Xinyi and Choo, Kim-Kwang Raymond and Vasilakos, Athanasios V. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. Journal of Network and Computer Applications. 2018;116:42–52.

[144] Li, Zhetao and Kang, Jiawen and Yu, Rong and Ye, Dongdong and Deng, Qingyong and Zhang, Yan. Consortium blockchain for secure energy trading in industrial internet of things. IEEE transactions on industrial informatics. 2017;14(8):3690–3700.

[145] Jason Andress. The Basics of Information Security (Second Edition). second edition ed. Boston: Syngressp.69–88. Available from: https://www.sciencedirect.com/science/article/pii/B9780128007440000051.

[146] Rathore, Shailendra and Kwon, Byung Wook and Park, Jong Hyuk. BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network. Journal of Network and Computer Applications. 2019;143:167–177.

[147] Zhao, Ke and Tang, Shaohua and Zhao, Bowen and Wu, Yiming. Dynamic and privacy-preserving reputation management for blockchain-based mobile crowdsensing. IEEE Access. 2019;7:74694–74710.

[148] SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS. Recommendation ITU-T Y2060. 2012 Jun;.

[149] Dai, Weiqi and Deng, Jun and Wang, Qinyuan and Cui, Changze and Zou, Deqing and Jin, Hai. SBLWT: A secure blockchain lightweight wallet based on trustzone. IEEE Access. 2018;6:40638–40648.

[150] Takemiya, Makoto and Vanieiev, Bohdan. Sora Identity: Secure, Digital Identity on the Blockchain. Intl Computer Software and Applications Conference. 2018;2:582–587.

[151] Dorri, Ali and Kanhere, Salil S and Jurdak, Raja and Gauravaram, Praveen. Blockchain for IoT security and privacy: The case study of a smart home. In: Intl. Conf. on pervasive computing and communications workshops. IEEEp.618–623.

[152] Novo, Oscar. Blockchain meets IoT: An architecture for scalable access management in IoT. IEEE Internet of Things Journal. 2018;5(2):1184–1195.

[153] Xu, Cheng and Liu, Hongzhe and Li, Peifeng and Wang, Pengfei. A remote attestation security model based on privacy-preserving blockchain for V2X. IEEE Access. 2018;6:67809–67818.

[154] Tanenbaum, Andrew S. and Steen, Maarten Van. 1st ed. USA: Prentice Hall PTR; 2001.

[155] Amazon. What is Cloud Computing? - Amazon Web Services. Amazon; Last access Feb. 2021. Available from: `https://aws.amazon.com/what-is-cloud-computing/`.

[156] Benhamouda, Fabrice and Halevi, Shai and Halevi, Tzipora. Supporting private data on hyperledger fabric with secure multiparty computation. Proc IEEE Intl Conf on Cloud Engineering. p.357–363.

[157] Lu, Yuan and Tang, Qiang and Wang, Guiling. ZebraLancer: Private and anonymous crowdsourcing system atop open blockchain. Proceedings - International Conference on Distributed Computing Systems. 2018;2018-July(i):853–865.

[158] Hasan, Haya R and Salah, Khaled. Proof of delivery of digital assets using blockchain and smart contracts. IEEE Access. 2018;6:65439–65448.

[159] Diaz, Jesus and Choi, Seung Geol and Arroyo, David and Keromytis, Angelos D and Rodriguez, Francisco B and Yung, Moti. Privacy in e-shopping transactions: Exploring and addressing the trade-offs. In: International Symposium on Cyber Security Cryptography and Machine Learning. Springer, Champ.206–226.

[160] Omar, Ilhaam A and Jayaraman, Raja and Salah, Khaled and Debe, Mazin and Omar, Mohammed. Enhancing vendor managed inventory supply chain operations using blockchain smart contracts. IEEE Access. 2020;8:182704–182719.

[161] Kim, Tai-Hoon and Kumar, Gulshan and Saha, Rahul and Rai, Mritunjay Kumar and Buchanan, William J and Thomas, Reji and Alazab, Mamoun. A privacy preserving distributed ledger framework for global human resource record management: The blockchain aspect. IEEE Access. 2020;8:96455–96467.

[162] Mora, Olga B and Rivera, Rogelio and Larios, Victor M and Beltrán-Ramírez, J Raul and Maciel, Rocio and Ochoa, Alberto. A Use Case in Cybersecurity

based in Blockchain to deal with the security and privacy of citizens and Smart Cities Cyberinfrastructures. In: 2018 IEEE International Smart Cities Conference (ISC2). IEEEp.1–4.

[163] Biswas, Kamanashis and Muthukkumarasamy, Vallipuram. Securing smart cities using blockchain technology. In: IEEE Intl. Conf. on high performance computing and communications. IEEEp.1392–1393.

[164] Sharples, Mike and Domingue, John. The blockchain and kudos: A distributed system for educational record, reputation and reward. In: European Conference on Technology Enhanced Learning. Springerp.490–496.

[165] Aitzhan, Nurzhan Zhumabekuly and Svetinovic, Davor. Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams. IEEE Trans on Dependable and Secure Computing. 2018;15(5):840–852.

[166] Guo, Rui and Shi, Huixian and Zhao, Qinglan and Zheng, Dong. Secure Attribute-Based Signature Scheme with Multiple Authorities for Blockchain in Electronic Health Records Systems. IEEE Access. 2018;6:11676–11686.

[167] Hofman, Darra and Shannon, Casey and McManus, Bruce and Lemieux, Victoria and Lam, Karen and Assadian, Sara and Ng, Raymond. Building Trust & Protecting Privacy: Analyzing Evidentiary Quality in a Blockchain Proof-of-Concept for Health Research Data Consent Management. In: IEEE Intl. Conf. on Internet of Things. IEEEp.1650–1656.

[168] Azaria, Asaph and Ekblaw, Ariel and Vieira, Thiago and Lippman, Andrew. Medrec: Using blockchain for medical data access and permission management. In: 2016 2nd International Conference on Open and Big Data (OBD). IEEEp.25–30.

[169] Wang, Yuxiao and Gao, Juntao. A regulation scheme based on the ciphertext-policy hierarchical attribute-based encryption in bitcoin system. IEEE Access. 2018;6:16267–16278.

[170] Medury, Sai and Skjellum, Anthony and Brooks, Richard R and Yu, Lu. SCRaaPS: X. 509 Certificate Revocation Using the Blockchain-based Scrybe Secure Provenance System. In: Intl. Conf. on Malicious and Unwanted Software. IEEEp.145–152.

[171] Xing, Qianqian and Wang, Baosheng and Wang, Xiaofeng. Poster: Bgpcoin: A Trustworthy Blockchain-Based Resource Management Solution for BGP Security. In: ACM SIGSAC Conf. on Comp. and Comms. Security. ACMp.2591–2593.

[172] Saad, Muhammad and Thai, My T and Mohaisen, Aziz. POSTER: deterring ddos attacks on blockchain-based cryptocurrencies through mempool optimization. In: Asia Conference on Computer and Communications Security. ACMp.809–811.

[173] Rahman, MD Abdur and Hossain, M Shamim and Loukas, George and Hassanain, Elham and Rahman, Syed Sadiqur and Alhamid, Mohammed F and Guizani, Mohsen. Blockchain-based mobile edge computing framework for secure therapy applications. IEEE Access. 2018;6:72469–72478.

[174] Le, Tam and Mutka, Matt W. . Capchain: A privacy preserving access control framework based on blockchain for pervasive environments. Proceedings - 2018 IEEE International Conference on Smart Computing, SMARTCOMP 2018. p.57–64.

[175] Kosba, Ahmed and Miller, Andrew and Shi, Elaine and Wen, Zikai and Papamanthou, Charalampos. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016. p.839–858.

[176] Gu, Jingjing and Sun, Binglin and Du, Xiaojiang and Member, Senior. Consortium Blockchain-Based Malware Detection in Mobile Devices. 2018;6.

[177] Hammi, Mohamed Tahar and Hammi, Badis and Bellot, Patrick and Serhrouchni, Ahmed. Bubbles of Trust: A decentralized blockchain-based authentication system for IoT. Computers and Security. 2018;78(2018):126–142.

[178] D. Chatzopoulos and S. Gujar and B. Faltings and P. Hui. Privacy Preserving and Cost Optimal Mobile Crowdsensing Using Smart Contracts on Blockchain. In: IEEE Intl. Conf. on Mobile Ad Hoc and Sensor Systemsp.442–450.

[179] Azaria, Asaph and Ekblaw, Ariel and Vieira, Thiago and Lippman, Andrew. MedRec: Using blockchain for medical data access and permission management. Intl Conf on Open and Big Data. p.25–30.

[180] Xia, QI and Sifah, Emmanuel Boateng and Asamoah, Kwame Omono and Gao, Jianbin and Du, Xiaojiang and Guizani, Mohsen. MeDShare: Trust-less medical data sharing among cloud service providers via blockchain. IEEE Access. 2017;5:14757–14767.

[181] Gipp, Bela and Kosti, Jagrut and Breitinger, Corinna. Securing Video Integrity Using Decentralized Trusted Timestamping on the Bitcoin Blockchain. Proc of the 10th Mediterranean Conf on Information Systems (MCIS). p.51.

[182] Sarier, Neyire Deniz. Cyberspace Safety and Security. In: p.254–269.

[183] Bramm, Georg and Gall, Mark and Schütte, Julian. BDABE-Blockchain-based Distributed Attribute based Encryption. In: ICETE (2)p.265–276.

[184] Leiding, Benjamin and Norta, Alex. Mapping requirements specifications into a formalized blockchain-enabled authentication protocol for secured personal identity assurance. In: Int. Conf. on Future Data and Security Engineering. Springerp.181–196.

[185] Zhang, Yinghui and Deng, Robert H. and Shu, Jiangang and Yang, Kan and Zheng, Dong. TKSE: Trustworthy keyword search over encrypted data with two-side verifiability via blockchain. IEEE Access. 2018;6:31077–31087.

[186] He, Yunhua and Li, Hong and Cheng, Xiuzhen and Liu, Yan and Yang, Chao and Sun, Limin. A blockchain based truthful incentive mechanism for distributed P2P applications. IEEE Access. 2018;6:27324–27335.

[187] Tahir, Shahzaib and Rajarajan, Muttukrishnan. Privacy-Preserving Searchable Encryption Framework for Permissioned Blockchain Networks. In: IEEE Intl. conf. on Internet of Things. IEEEp.1628–1633.

[188] Hepp, Thomas and Wortner, Patrick and Schönhals, Alexander and Gipp, Bela. Securing Physical Assets on the Blockchain: Linking a Novel Object Identification Concept with Distributed Ledgers. In: Proc. 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems. CryBlock'18. ACMp.60–65.

[189] Dai, Mingjun and Zhang, Shengli and Wang, Hui and Jin, Shi. A low storage room requirement framework for distributed ledger in blockchain. IEEE Access. 2018;6:22970–22975.

[190] Lin, Iuon-Chang and Liao, Tzu-Chun. A survey of blockchain security issues and challenges. IJ Network Security. 2017;19(5):653–659.

[191] Hasanova, Huru and Baek, Ui-jun and Shin, Mu-gon and Cho, Kyunghee and Kim, Myung-Sup. A survey on blockchain cybersecurity vulnerabilities and possible countermeasures. International Journal of Network Management. 2019;29(2):2060.

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

[193] Leng, Jiewu and Zhou, Man and Zhao, Leon J and Huang, Yongfeng and Bian, Yiyang. Blockchain security: A survey of techniques and research directions. IEEE Transactions on Services Computing. 2020;.

[194] Dai, Fangfang and Shi, Yue and Meng, Nan and Wei, Liang and Ye, Zhiguo. From Bitcoin to cybersecurity: A comparative study of blockchain application and security issues. In: 2017 4th International Conference on Systems and Informatics (ICSAI). IEEEp.975–979.

[195] Fran Casino and Thomas K. Dasaklis and Constantinos Patsakis. A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and Informatics. 2019;36:55 – 81.

[196] Mohanta, Bhabendu Kumar and Jena, Debasish and Panda, Soumyashree S and Sobhanayak, Srichandan. Blockchain technology: A survey on applications and security privacy challenges. Internet of Things. 2019;8:100107.

[197] Tariq, Noshina and Asim, Muhammad and Al-Obeidat, Feras and Zubair Farooqi, Muhammad and Baker, Thar and Hammoudeh, Mohammad and Ghafir, Ibrahim. The security of big data in fog-enabled IoT applications including blockchain: A survey. Sensors. 2019;19(8):1788.

[198] Baniata, Hamza and Kertesz, Attila. A survey on blockchain-fog integration approaches. IEEE Access. 2020;8:102657–102668.

[199] Leng, Jiewu and Ye, Shide and Zhou, Man and Zhao, J Leon and Liu, Qiang and Guo, Wei and Cao, Wei and Fu, Leijie. Blockchain-Secured Smart Manufacturing in Industry 4.0: A Survey. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2020;.

[200] Paul J. Taylor and Tooska Dargahi and Ali Dehghantanha and Reza M. Parizi and Kim-Kwang Raymond Choo. A systematic literature review of blockchain cyber security. Digital Communications and Networks. 2020;6(2):147

– 156. Available from: http://www.sciencedirect.com/science/article/pii/S2352864818301536.

[201] Shi, Shuyun and He, Debiao and Li, Li and Kumar, Neeraj and Khan, Muhammad Khurram and Choo, Kim-Kwang Raymond. Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey. Computers & Security. p.101966.

[202] Moubarak, Joanna and Chamoun, Maroun and Filiol, Eric. Developing a Kary malware using blockchain. In: 2018 IEEE/IFIP Network Operations and Management Symposiump.1–4.

[203] Oscar Delgado-Mohatar and José María Sierra-Cámara and Eloy Anguiano. Blockchain-based semi-autonomous ransomware. Future Generation Computer Systems. 2020;112:589–603. Available from: https://www.sciencedirect.com/science/article/pii/S0167739X19317406.

[204] Sinegubko, Denis. Website ransomware – CTB-Locker goes blockchain. Sucuri Blog; 2018. Available from: https://blog.sucuri.net/2016/04/website-ransomware-ctb-locker-goes-blockchain.html.

[205] Fayi, Sharifah Yaqoub A. . Information Technology - New Generations. In: . Cham: Springer International Publishingp.93–100.

[206] Hurtuk, Ján and Chovanec, Martin and Kičina, Michal and Billík, Roman. Case Study of Ransomware Malware Hiding Using Obfuscation Methods. In: 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA)p.215–220.

[207] Mengidis, Anagnostis. Blockchain-based command and control for next generation botnets. 2019;.

[208] Meskauskas, Tomas. Avoslocker ransomware. PCrisk; 2022. Available from: https://www.pcrisk.com/removal-guides/21388-avoslocker-ransomware.

[209] Meskauskas, Tomas. Darkside ransomware. PCrisk; 2021. Available from: https://www.pcrisk.com/removal-guides/18504-darkside-ransomware.

[210] Dimitri Kamenski and Arash Shaghaghi and Matthew J. Warren and Salil S. Kanhere. Attacking with bitcoin: Using Bitcoin to Build Resilient Botnet Armies. CoRR. 2020;abs/2004.01855. Available from: https://arxiv.org/abs/2004.01855.

[211] Frkat, Davor and Annessi, Robert and Zseby, Tanja. ChainChannels: Private Botnet Communication Over Public Blockchains. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)p.1244–1252.

[212] Lemmou, Yassine and Souidi, El Mamoun. Cryptology and Network Security. In: . Cham: Springer International Publishingp.154–174.

[213] Lemmou, Yassine and Souidi, El Mamoun. Infection, Self-reproduction and Overinfection in Ransomware: The Case of TeslaCrypt. In: 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)p.1–8.

[214] Meskauskas, Tomas. Locky ransomware [updated]. PCrisk; 2021. Available from: https://www.pcrisk.com/removal-guides/9807-locky-ransomware.

[215] Meskauskas, Tomas. BTCWare ransomware. PCrisk; 2021. Available from: https://www.pcrisk.com/removal-guides/11101-btcware-ransomware.

[216] Abrams, Lawrence. The Globe Ransomware wants to purge your files. BleepingComputer; 2016. Available from: https://www.bleepingcomputer.com/news/security/the-globe-ransomware-wants-to-purge-your-files/.

[217] Meskauskas, Tomas. Random6 ransomware. PCrisk; 2020. Available from: https://www.pcrisk.com/removal-guides/11409-random6-ransomware.

[218] Security, Panda. Computer worms - panda security;. Available from: https://www.pandasecurity.com/en/security-info/worm/.

[219] Curran, Tom and Geist, Dana. Using the bitcoin blockchain as a botnet resilience mechanism. 2016;.

[220] Paquet-Clouston, Masarah and Haslhofer, Bernhard and Dupont, Benoît. Ransomware payments in the Bitcoin ecosystem. Journal of Cybersecurity. 2019 05;5(1). Tyz003. Available from: https://doi.org/10.1093/cybsec/tyz003.

[221] Platdrag. Platdrag/UnblockableChains: Unblockable chains - a POC on using blockchain as infrastructure for malware operations;. Available from: https://github.com/platdrag/UnblockableChains.

[222] Falco, Gregory and Li, Caleb and Fedorov, Pavel and Caldera, Carlos and Arora, Rahul and Jackson, Kelly. NeuroMesh: IoT Security Enabled by a Blockchain Powered Botnet Vaccine. In: Proceedings of the International Conference on Omni-Layer Intelligent Systems. COINS '19. New York, NY, USA: Association for Computing Machineryp.1–6. Available from: https://doi.org/10.1145/3312614.3312615.

[223] Kurt, Ahmet and Erdin, Enes and Cebe, Mumin and Akkaya, Kemal and Uluagac, A Selcuk. LNBot: a covert hybrid botnet on bitcoin lightning network for fun and profit. In: European Symposium on Research in Computer Security. Springerp.734–755.

[224] Karapapas, Christos and Pittaras, Iakovos and Fotiou, Nikos and Polyzos, George C. . Ransomware as a Service using Smart Contracts and IPFS. In: 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)p.1–5.

[225] Kang, Yanze and Yu, Xiaobo and Meng, Weizhi and Liu, Yining. BlockRAT: An Enhanced Remote Access Trojan Framework via Blockchain. In: Science

of Cyber Security: 4th International Conference, SciSec 2022, Matsue, Japan, August 10–12, 2022, Revised Selected Papers. Springerp.21–35.

[226] Grinler. New cryptotorlocker2015 Ransomware discovered and easily decrypted - archived news; 2015. Available from: https://www.bleepingcomputer.com/forums/t/565020/new-cryptotorlocker2015-ransomware-discovered-and-easily-decrypted/.

[227] Meskauskas, Tomas. Dharma ransomware. PCrisk; 2022. Available from: https://www.pcrisk.com/removal-guides/10672-dharma-ransomware.

[228] Meskauskas, Tomas. Fantom ransomware. PCrisk; 2021. Available from: https://www.pcrisk.com/removal-guides/10418-fantom-ransomware.

[229] Karapapas, Christos and Pittaras, Iakovos and Fotiou, Nikos and Polyzos, George C. . Ransomware as a Service using Smart Contracts and IPFS. In: 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)p.1–5.

[230] Ali, Syed Taha and McCorry, Patrick and Lee, Peter Hyun-Jeen and Hao, Feng. Zombiecoin: Powering next-generation botnets with bitcoin. In: International Conference on Financial Cryptography and Data Security. Springerp.34–48.

[231] Baden, Mathis and Ferreira Torres, Christof and Fiz Pontiveros, Beltran Borja and State, Radu. Whispering Botnet Command and Control Instructions. In: 2019 Crypto Valley Conference on Blockchain Technology (CVCBT)p.77–81.

[232] Yin, Jie and Cui, Xiang and Liu, Chaoge and Liu, Qixu and Cui, Tao and Wang, Zhi. CoinBot: A Covert Botnet in the Cryptocurrency Network. In: International Conference on Information and Communications Security. Springerp.107–125.

[233] Sweeny, J. Botnet resiliency via private blockchains. SANS Institute Information Security Reading Group. 2017;.

[234] Oliveira,Alexandre and Gonçalves,Vinícius and Filho,Geraldo R. . Using Ethereum Smart Contracts for Botnet Command and Control; 2020. Copyright - Copyright Academic Conferences International Limited Jun 2020; Última actualización - 2021-07-13. Available from: https://www.proquest.com/conference-papers-proceedings/using-ethereum-smart-contracts-botnet-command/docview/2453793786/se-2?accountid=14501.

[235] Partala, Juha. Provably Secure Covert Communication on Blockchain. Cryptography. 2018;2(3):18.

[236] Okupski, Krzysztof S. . (Ab) using Bitcoin for anti-censorship tool. Technische Universiteit Eindhoven Master Thesis (2014). 2014;.

[237] Kurt, Ahmet and Erdin, Enes and Akkaya, Kemal and Uluagac, A Selcuk and Cebe, Mumin. D-LNBot: A Scalable, Cost-Free and Covert Hybrid Botnet on Bitcoin's Lightning Network. arXiv preprint arXiv:211207623. 2021;.

[238] Horejsi, Jaromir and Chen, Joseph C. Glupteba hits routers and updates C&C Servers; 2019. Available from: https://www.trendmicro.com/en_us/research/19/i/glupteba-campaign-hits-network-routers-and-updates-cc-servers-with-data-from-bitcoin-transactions.html.

[239] Roffel, Douglas and Garret, Christopher. "A-novel-approach-for-computer-worm-control-using-decentralized-data-structures"; 2014. Available from: https://archive.org/stream/pdfy-E2ZwuLAVfC44kEQk/250009335-A-Novel-Approach-for-Computer-Worm-Control-Using-Decentralized-Data-Structures_djvu.txt.

[240] Malaika, NAI Majid A and Al Ibrahim, Omar. Botract: Abusing Smart Contracts and Blockchains for Botnet Command and Control;.

[241] Abellán Álvarez, Iván. Misusing bitcoin for botnet command and control communication. 2019;.

[242] Zhong, Yi and Zhou, Anmin and Zhang, Lei and Jing, Fan and Zuo, Zheng. DUSTBot: A duplex and stealthy P2P-based botnet in the Bitcoin network. PloS one. 2019;14(12):e0226594.

[243] Pirozzi, A. BOTCHAIN aka The Dark side of Blockchain; 2018.

[244] Franzoni, Federico and Abellan, Ivan and Daza, Vanesa. Leveraging bitcoin testnet for bidirectional botnet command and control systems. In: International Conference on Financial Cryptography and Data Security. Springerp.3–19.

[245] Meskauskas, Tomas. Random6 ransomware. PCrisk; 2021. Available from: https://www.pcrisk.com/removal-guides/11409-random6-ransomware.

[246] Ethereum Historical Data;. Available from: https://www.investing.com/crypto/ethereum/historical-data.

[247] Monero Historical Data;. Available from: https://www.investing.com/crypto/monero/historical-data.

[248] Bitcoin historical data;. Available from: https://www.investing.com/crypto/bitcoin/historical-data.

[249] Minimum for sending BTC from BTC wallet;. Available from: https://bitcoin.stackexchange.com/questions/105214/minimum-for-sending-btc-from-btc-wallet.

[250] CoinMarketCap. What is a crypto faucet?: CoinMarketCap. CoinMarketCap; 2021. Available from: https://coinmarketcap.com/alexandria/article/what-is-a-crypto-faucet.

[251] Carr, Sam. How do botnets make money from your ads?; 2021. Available from: https://ppcprotect.com/blog/ad-fraud/how-botnets-make-money/.

[252] Namestnikov, Yuri. The economics of botnets. Analysis on Viruslist com, Kapersky Lab. 2009;.

[253] Makrushin, Denis. The cost of launching a ddos attack; 2021. Available from: https://securelist.com/the-cost-of-launching-a-ddos-attack/77784/.

[254] Namestnikov, Yuri. The economics of botnets;. Available from: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2009/07/01121538/ynam_botnets_0907_en.pdf.

[255] Orman, Hilarie. Evil Offspring - Ransomware and Crypto Technology. IEEE Internet Computing. 2016;20(5):89–94.

[256] Conti, Mauro and Gangwal, Ankit and Ruj, Sushmita. On the economic significance of ransomware campaigns: A Bitcoin transactions perspective. Computers & Security. 2018;79:162–189.

[257] Huang, Danny Yuxing and Aliapoulios, Maxwell Matthaios and Li, Vector Guo and Invernizzi, Luca and Bursztein, Elie and McRoberts, Kylie and Levin, Jonathan and Levchenko, Kirill and Snoeren, Alex C and McCoy, Damon. Tracking ransomware end-to-end. In: 2018 IEEE Symposium on Security and Privacy (SP). IEEEp.618–631.

[258] Faisal, Tooba and Courtois, Nicolas and Serguieva, Antoaneta. The Evolution of Embedding Metadata in Blockchain Transactions. In: 2018 International Joint Conference on Neural Networks (IJCNN)p.1–9.

[259] Shannon, Claude E. A mathematical theory of communication. Bell system technical journal. 1948;27(3):379–423.

[260] S. , Jimi. Blockchain: how mining works and transactions are processed in seven steps. Good Audience; 2018). (Last access Feb. 2021. Available from: https://blog.goodaudience.com/how-a-miner-adds-transactions-to-the-blockchain-in-seven-steps-856053271476.

[261] Ethereum. Ethereum wiki; 2019. Last access April 2021. https://github.com/ethereum/wiki/wiki/Design-Rationale#gas-and-fees.

[262] Moriya, Hideyoshi. How to get Ethereum Block Gas Limit; 2018. Last access April 2021. https://medium.com/@piyopiyo/how-to-get-ethereum-block-gas-limit-eba2c8f32ce.

[263] Kobel, Vincent. Generating a usable Ethereum wallet and its corresponding keys; 2017. Last access April 2021. https://kobl.one/blog/create-full-ethereum-keypair-and-address/.

[264] Solidity types;. Last access April 2021. https://solidity.readthedocs.io/en/v0.5.10/types.html.

[265] Gervais, Arthur and et al. . On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACMp.3–16.

[266] Kaliski, Burt. PKCS# 5: Password-based cryptography specification version 2.0; 2000.

[267] Geth;. Last access April 2021. https://geth.ethereum.org/.

[268] Infura;. Last access April 2021. https://infura.io/.

[269] Ropsten;. Last access April 2021. https://github.com/ethereum/ropsten.

[270] Salt contract;. Last access April 2021. https://etherscan.io/address/0x4156d3342d5c385a87d264f90653733592000581.

[271] UniswapMakerBroker contract;. Last access May 2020. https://etherscan.io/address/0xa35f3acb4d6c43e6f9a1c2d8c136ad4be725152f#code.

[272] BNB token contract;. Last access April 2021. https://etherscan.io/address/0xB8c77482e45F1F44dE1745F52C74426C631bDD52.

[273] UniversalDeployer contract;. Last access April 2021. https://etherscan.io/address/0x252f1c9aee12a65ac113e4b6c4660a4c2f572b06#code.

[274] Estimate gas;. Last access April 2021. https://github.com/ethereum/wiki/wiki/JSON-RPC#eth_estimategas.

[275] Rudden, Jennifer. Ethereum price monthly 2016-2020; 2020. https://www.statista.com/statistics/806453/price-of-ethereum/.

[276] ETH Gas Station;. Last access April 2021. https://ethgasstation.info/.

[277] She, Wei and Huo, Lijuan and Tian, Zhao and Zhuang, Yan and Niu, Chaoyi and Liu, Wei. A double steganography model combining blockchain and interplanetary file system. Peer-to-Peer Networking and Applications. p.1–14.

[278] Liu, Si and Liu, Yunxia and Feng, Cong and Zhao, Hongguo and Huang, Yu. Blockchain Privacy Data Protection Method Based on HEVC Video Steganography. In: 2020 3rd International Conference on Smart BlockChain (SmartBlock). IEEEp.1–6.

[279] Giron, Alexandre Augusto and Martina, Jean Everson and Custódio, Ricardo. Bitcoin Blockchain Steganographic Analysis. In: International Conference on Applied Cryptography and Network Security. Springerp.41–57.

[280] M. D. Sleiman and A. P. Lauf and R. Yampolskiy. Bitcoin Message: Data Insertion on a Proof-of-Work Cryptocurrency System. In: 2015 International Conference on Cyberworlds (CW)p.332–336.

[281] Tian, Jing and Gou, Gaopeng and Liu, Chang and Chen, Yige and Xiong, Gang and Li, Zhen. Information and Communications Security. In: . Cham: Springer International Publishingp.814–830.

[282] A. Fionov. Exploring Covert Channels in Bitcoin Transactions. In: 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)p.0059–0064.

[283] Torki, Omid and Ashouri-Talouki, Maede and Mahdavi, Mojtaba. Blockchain for steganography: advantages, new algorithms and open challenges. arXiv preprint arXiv:210103103. 2021;.

[284] Frkat, Davor and Annessi, Robert and Zseby, Tanja. ChainChannels: Private Botnet Communication Over Public Blockchains;.

[285] A. I. Basuki and D. Rosiyadi. Joint Transaction-Image Steganography for High Capacity Covert Communication. In: 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)p.41–46.

[286] Gao, Feng and Zhu, Liehuang and Gai, Keke and Zhang, Can and Liu, Sheng. Achieving a Covert Channel over an Open Blockchain Network. IEEE Network. 2020;34(2):6–13.

[287] Alsalami, Nasser and Zhang, Bingsheng. Uncontrolled Randomness in Blockchains: Covert Bulletin Board for Illicit Activities;.

[288] Xu, Mengtian and Wu, Hanzhou and Feng, Guorui and Zhang, Xinpeng and Ding, Feng. Broadcasting Steganography in the Blockchain. In: International Workshop on Digital Watermarking. Springerp.256–267.

[289] Elliptic curve cryptography (ECC);. Available from: `https://cryptobook.nakov.com/asymmetric-key-ciphers/elliptic-curve-cryptography-ecc`.

[290] Size considerations for public and private keys;. Available from: `https://www.ibm.com/docs/en/zos/2.2.0?topic=certificates-size-considerations-public-private-keys`.

[291] Rafael. What is key length in cryptography and why is important?; 2022. Available from: `https://justcryptography.com/key-length/`.

[292] Wang, Xinyuan and Jiang, Xuxian. A First Step Toward Live Botmaster Traceback. 2008;.

[293] Ethereum lists;. Available from: https://github.com/MyEtherWallet/ethereum-lists.

[294] Networks;. Available from: https://ethereum.org/nb/developers/docs/networks/#sepolia.

[295] Ethereum Average Gas Price;. Available from: https://ycharts.com/indicators/ethereum_average_gas_price#:~:text=Ethereum%20Average%20Gas%20Price%20is,84.76%25%20from%20one%20year%20ago.

[296] Putman, CGJ and Nieuwenhuis, Lambert JM and others. Business model of a botnet. In: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP). IEEEp.441–445.

[297] Zhang, Lejun and Zhang, Zhijie and Wang, Weizheng and Jin, Zilong and Su, Yansen and Chen, Huiling. Research on a covert communication model realized by using smart contracts in blockchain environment. IEEE Systems Journal. 2021;.

[298] Gimenez-Aguilar, Mar and De Fuentes, Jose M. and González-Manzano, Lorena and Camara, Carmen. Zephyrus: An Information Hiding Mechanism Leveraging Ethereum Data Fields. IEEE Access. 2021;9:118553–118570.

[299] What is EVM? ethereum virtual machine;. Available from: https://www.horizen.io/blockchain-academy/technology/advanced/evm-ethereum-virtual-machine/.

[300] Geroni, Diego. Ethereum (ETH) block explorer - an overview; 2022. Available from: https://101blockchains.com/ethereum-blockchain-explorer/.

[301] López, Patricio. Ethereum vs avalanche gas cost. Medium; 2020. Available from: https://patricio-lopez-75857.medium.com/how-much-cost-use-the-avalanche-c-chian-c358292c6436.

[302] Hmood, Ali K. and Jalab, Hamid A. and Kasirun, Z. M. and Zaidan, B. B. and Zaidan, A. A. . On the Capacity and Security of Steganography Approaches: An Overview. Journal of Applied Sciences. 2010 Dec;10(16):1825–1833. Available from: https://ui.adsabs.harvard.edu/abs/2010JApSc..10.1825H.

[303] Apple Rolls Out Privacy-Focused 'Nutrition Labels' for Apps; 2020). (Last access Feb. 2021. Available from: https://www.pcmag.com/news/apple-rolls-out-privacy-focused-nutrition-labels-for-apps.

[304] Hassam, Samer and De Filippi, Primavera. Decentralised autonomous organisation; Last access Feb. 2021. Available from: https://policyreview.info/glossary.

[305] Reis, Dayane and Takeshita, Jonathan and Jung, Taeho and Niemier, Michael and Hu, Xiaobo Sharon. Computing-in-Memory for Performance and Energy Efficient Homomorphic Encryption. arXiv preprint arXiv:200503002. 2020;.

[306] Article 32 GDPR. Security of processing; 2021. Available from: https://gdpr-text.com/read/article-32/.

[307] B. A. Scriber. A Framework for Determining Blockchain Applicability. IEEE Software. 2018;35(4):70–77.

[308] Pedersen, Asger and Risius, Marten and Beck, Roman. A Ten-Step Decision Path to Determine When to Use Blockchain Technologies. MIS Quarterly Executive. 2019 06;18:99–115.

[309] Singh, Nitish. When To Use Blockchain Technology?; 2020). (Last access Feb. 2021. Available from: https://101blockchains.com/when-to-use-blockchain/.

[310] Gao, Yu-Long and Chen, Xiu-Bo and Chen, Yu-Ling and Sun, Ying and Niu, Xin-Xin and Yang, Yi-Xian. A secure cryptocurrency scheme based on post-quantum blockchain. IEEE Access. 2018;6:27205–27213.

[311] Lin, Chao and He, Debiao and Huang, Xinyi and Khan, Muhammad Khurram and Choo, Kim-Kwang Raymond. A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems. IEEE Access. 2018;6:28203–28212.

[312] Kim, Hyun-Woo and Jeong, Young-Sik. Secure authentication-management human-centric scheme for trusting personal resource information on mobile cloud computing with blockchain. Human-centric Comp and Inf Sci. 2018;8(1):11.

[313] Chen, Jollen. Hybrid blockchain and pseudonymous authentication for secure and trusted IoT networks. ACM SIGBED Review. 2018;15(5):22–28.

[314] Feng, Xiaoqin and Ma, Jianfeng and Feng, Tao and Miao, Yinbin and Liu, Ximeng. Consortium Blockchain-Based SIFT: Outsourcing Encrypted Feature Extraction in the D2D Network. IEEE Access. 2018;6:52248–52260.

[315] Huang, Xiaohong and Xu, Cheng and Wang, Pengfei and Liu, Hongzhe. LNSC: A security model for electric vehicle and charging pile management based on blockchain ecosystem. IEEE Access. 2018;6:13565–13574.

[316] Sharma, Pradip Kumar and Chen, Mu-Yen and Park, Jong Hyuk. A software defined fog node based distributed blockchain cloud architecture for IoT. IEEE Access. 2017;6:115–124.

[317] Niu, Yukun and Wei, Lingbo and Zhang, Chi and Liu, Jianqing and Fang, Yuguang. An anonymous and accountable authentication scheme for Wi-Fi hotspot access with the Bitcoin blockchain. In: IEEE/CIC Intl. Conf. on Communications in China. IEEE; 2017. .

[318] Watanabe, Hiroki and Fujimura, Shigeru and Nakadaira, Atsushi and Miyazaki, Yasuhiko and Akutsu, Akihito and Kishigami, Jay. Blockchain contract: Securing a blockchain applied to smart contracts. In: 2016 IEEE international conference on consumer electronics (ICCE). IEEEp.467–468.

[319] Miers, Ian and Garman, Christina and Green, Matthew and Rubin, Aviel D. Zerocoin: Anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy. IEEEp.397–411.

[320] Kfoury, Elie and Khoury, David. Securing NATted IoT Devices Using Ethereum Blockchain and Distributed TURN Servers. In: 2018 10th International Conference on Advanced Infocomm Technology (ICAIT). IEEEp.115–121.

[321] Wang, Siye and Zhu, Shaoyi and Zhang, Yanfang. Blockchain-based mutual authentication security protocol for distributed RFID systems. In: 2018 IEEE Symposium on Computers and Communications (ISCC). IEEEp.00074–00077.

[322] Zhou, Beini and Li, Hui and Xu, Li. An Authentication Scheme Using Identity-based Encryption & Blockchain. In: 2018 IEEE Symposium on Computers and Communications (ISCC). IEEEp.00556–00561.

[323] Nuss, Martin and Puchta, Alexander and Kunz, Michael. Towards blockchain-based identity and access management for internet of things in enterprises. In: International Conference on Trust and Privacy in Digital Business. Springerp.167–181.

[324] Lu, Zhaojun and Wang, Qian and Qu, Gang and Liu, Zhenglin. Bars: a blockchain-based anonymous reputation system for trust management in vanets. In: IEEE Intl. Conf. On Trust, Security And Privacy In Computing And Communications. IEEEp.98–103.

[325] Neisse, Ricardo and Steri, Gary and Fovino, Igor Nai. Blockchain-based Identity Management and Data Usage Control. In: IFIP International Summer School on Privacy and Identity Management. Springerp.237–239.

[326] Alexopoulos, Nikolaos and Daubert, Jorg and Muhlhauser, Max and Habib, Sheikh Mahbub. Beyond the hype: On using blockchains in trust management

for authentication. Intl Conf on Trust, Security and Privacy in Computing and Communications. p.546–553.

[327] Bonneau, Joseph and Narayanan, Arvind and Miller, Andrew and Clark, Jeremy and Kroll, Joshua A and Felten, Edward W. Mixcoin: Anonymity for Bitcoin with accountable mixes. In: Intl. Conf. on Financial Cryptography and Data Security. Springerp.486–504.

[328] Sasson, Eli Ben and Chiesa, Alessandro and Garman, Christina and Green, Matthew and Miers, Ian and Tromer, Eran and Virza, Madars. Zerocash: Decentralized anonymous payments from bitcoin. In: IEEE Symp. on Security and Privacy. IEEEp.459–474.

[329] Zhu, Yan and Qin, Yao and Zhou, Zhiyuan and Song, Xiaoxu and Liu, Guowei and Chu, William Cheng-Chung. Digital Asset Management with Distributed Permission over Blockchain and Attribute-Based Access Control. In: 2018 IEEE International Conference on Services Computing (SCC). IEEEp.193–200.

[330] Almakhour, Mouhamad and Sliman, Layth and Samhat, Abed Ellatif and Gaaloul, Walid. Trustless Blockchain-based Access Control in Dynamic Collaboration. In: BDCSIntellp.27–33.

[331] X. Chen and J. Ji and C. Luo and W. Liao and P. Li. When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design. In: IEEE Intl. Conf. on Big Datap.1178–1187.

[332] Badertscher, Christian and Gaži, Peter and Kiayias, Aggelos and Russell, Alexander and Zikas, Vassilis. Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACMp.913–930.

[333] Urien, Pascal. Blockchain IoT (BIoT): A new direction for solving Internet

of Things security and trust issues. In: Cloudification of the Internet of Things. IEEE; 2018. .

[334] Uriarte, Rafael Brundo and De Nicola, Rocco and Kritikos, Kyriakos. Towards distributed sla management with smart contracts and blockchain. In: Intl. Conf. on Cloud Computing Technology and Science. IEEEp.266–271.

[335] Zhou, Huan and de Laat, Cees and Zhao, Zhiming. Trustworthy cloud service level agreement enforcement with blockchain based smart contract. In: Intl. Conf. on Cloud Computing Technology and Science. IEEEp.255–260.

[336] Azbeg, Kebira and Ouchetto, Ouail and Andaloussi, Said Jai and Fetjah, Leila and Sekkaki, A. . Blockchain and IoT for Security and Privacy: A Platform for Diabetes Self-management. In: 4th Intl. Conf. on Cloud Computing Tech. and Apps. IEEE; 2018. .

[337] Shuaib, Khaled and Abdella, Juhar Ahmed and Sallabi, Farag and Abdel-Hafez, Mohammed. Using Blockchains to Secure Distributed Energy Exchange. In: Intl. Conf on Control, Decision and Information Technologies. IEEEp.622–627.

[338] Wang, Yuntao and Su, Zhou and Xu, Qichao and Zhang, Ning. Contract Based Energy Blockchain for Secure Electric Vehicles Charging in Smart Community. In: Intl Conf on Dependable, Autonomic and Secure Computing. IEEEp.323–327.

[339] Mendes, David and Rodrigues, Irene and Fonseca, César and Lopes, Manuel and García-Alonso, José Manuel and Berrocal, Javier. Anonymized Distributed PHR Using Blockchain for Openness and Non-repudiation Guarantee. In: International Conference on Theory and Practice of Digital Libraries. Springerp.381–385.

[340] Zhang, Ning and Li, Jin and Lou, Wenjing and Hou, Y Thomas. Privacy-Guard: Enforcing private data usage with blockchain and attested execution. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology. Springerp.345–353.

[341] Sharma, Rohit and Chakraborty, Suchetana. BlockAPP: Using Blockchain for Authentication and Privacy Preservation in IoV. In: IEEE Globecom Workshops. IEEEp.1–6.

[342] Li, Chao and Palanisamy, Balaji. Decentralized privacy-preserving timed execution in blockchain-based smart contract platforms. In: 2018 IEEE 25th International Conference on High Performance Computing (HiPC). IEEEp.265–274.

[343] Gürcan, Önder and Agenis-Nevers, Marc and Batany, Yves-Marie and Elmtiri, Mohamed and Le Fevre, François and Tucci-Piergiovanni, Sara. An Industrial Prototype of Trusted Energy Performance Contracts using Blockchain Technologies. In: IEEE Intl. Conf. on High Performance Computing and Communications. IEEEp.1336–1343.

[344] Kirkman, Stephen. A data movement policy framework for improving trust in the cloud using smart contracts and blockchains. In: 2018 IEEE International Conference on Cloud Engineering (IC2E). IEEEp.270–273.

[345] Kirkman, Stephen and Newman, Richard. A Cloud Data Movement Policy Architecture Based on Smart Contracts and the Ethereum Blockchain. In: 2018 IEEE International Conference on Cloud Engineering (IC2E). IEEEp.371–377.

[346] Li, Jingyi and Wu, Jigang and Chen, Long and Li, Jiaxing. Blockchain-based secure and reliable distributed deduplication scheme. In: International Conference on Algorithms and Architectures for Parallel Processing. Springerp.393–405.

[347] Xue, Jingting and Xu, Chunxiang and Zhang, Yuan and Bai, Lanhua.

Dstore: a distributed cloud storage system based on smart contracts and blockchain. In: Intl. Conf. on Algorithms and Architectures for Parallel Processing. Springerp.385–401.

[348] Lu, Peggy Joy and Yeh, Lo-Yao and Huang, Jiun-Long. An Privacy-Preserving Cross-Organizational Authentication/Authorization/Accounting System Using Blockchain Technology. In: IEEE Intl. Conf. on Communications. IEEEp.1–6.

[349] Rawat, Danda B and Alshaikhi, Amani. Leveraging distributed blockchain-based scheme for wireless network virtualization with security and QoS constraints. In: Intl. Conf. on Computing, Networking and Communications. IEEEp.332–336.

[350] Ourad, Abdallah Zoubir and Belgacem, Boutheyna and Salah, Khaled. Using blockchain for IOT access control and authentication management. In: International Conference on Internet of Things. Springerp.150–164.

[351] Kim, Kyoungmin and You, Youngin and Park, Mookyu and Lee, Kyungho. DDoS Mitigation: Decentralized CDN Using Private Blockchain. In: 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN). IEEEp.693–696.

[352] Park, Woo-Suk and Hwang, Dong-Yeop and Kim, Ki-Hyung. A TOTP-Based Two Factor Authentication Scheme for Hyperledger Fabric Blockchain. In: Intl. Conf. on Ubiquitous and Future Networks. IEEEp.817–819.

[353] Alnemari, Asma and Arodi, Suchith and Sosa, Valentina Rodriguez and Pandey, Soni and Romanowski, Carol and Raj, Rajendra and Mishra, Sumita. Protecting Infrastructure Data via Enhanced Access Control, Blockchain and Differential Privacy. In: International Conference on Critical Infrastructure Protection. Springerp.113–125.

[354] Kouzinopoulos, Charalampos S and Giannoutakis, Konstantinos M and Votis, Konstantinos and Tzovaras, Dimitrios and Collen, Anastasija and Katsikas, Sokratis. Implementing a forms of consent smart contract on an IoT-based blockchain to promote user trust. In: 2018 Innovations in Intelligent Systems and Applications (INISTA). IEEEp.1–6.

[355] Mendiboure, Léo and Chalouf, Mohamed Aymen and Krief, Francine. Towards a Blockchain-Based SD-IoV for Applications Authentication and Trust Management. In: International Conference on Internet of Vehicles. Springerp.265–277.

[356] Zhu, Yan and Song, Xiaoxu and Yang, Shuai and Qin, Yao and Zhou, Qiong. Secure Smart Contract System Built on SMPC Over Blockchain. In: Intl. Conf. on Internet of Things. IEEEp.1539–1544.

[357] Brousmiche, Kei Leo and Durand, Antoine and Heno, Thomas and Poulain, Christian and Dalmieres, Antoine and Hamida, Elyes Ben. Hybrid cryptographic protocol for secure vehicle data sharing over a consortium blockchain. In: IEEE Intl. Conf. on Internet of Things. IEEEp.1281–1286.

[358] Omar, Ahmad Sghaier and Basir, Otman. Identity Management in IoT Networks Using Blockchain and Smart Contracts. In: IEEE Intl. Conf. on Internet of Things (iThings). IEEEp.994–1000.

[359] Singh, Kalpana and Heulot, Nicolas and Hamida, Elyes Ben. Towards Anonymous, Unlinkable, and Confidential Transactions in Blockchain. In: IEEE Intl. Conf. on Internet of Things. IEEEp.1642–1649.

[360] Uchibeke, Uchi Ugobame and Schneider, Kevin A and Kassani, Sara Hosseinzadeh and Deters, Ralph. Blockchain access control Ecosystem for Big Data security. In: IEEE Intl. Conf. on Internet of Things (iThings). IEEEp.1373–1378.

[361] Holl, Patrick and Scepankova, Elena and Matthes, Florian. Smart Contract

based API usage tracking on the Ethereum Blockchain. Software Engineering und Software Management 2018. 2018;.

[362] Ali, Saqib and Wang, Guojun and White, Bebo and Cottrell, Roger Leslie. A blockchain-based decentralized data storage and access framework for pinger. In: Proc. 17th IEEE Intl. Conf. on Trust, Sec. and Priv. in Computing and Comms. IEEEp.1303–1308.

[363] Li, Bin and Wang, Yijie. RZKPB: a privacy-preserving blockchain-based fair transaction method for sharing economy. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEEp.1164–1169.

[364] Li, Bin and Wang, Yijie and Shi, Peichang and Chen, Huan and Cheng, Li. FPPB: a fast and privacy-preserving method based on the permissioned blockchain for fair transactions in sharing economy. In: 17th IEEE Intl. Conf. On Trust, Security And Privacy In Computing And Communications. IEEEp.1368–1373.

[365] Yuan, Yong and Wang, Fei-Yue. Towards blockchain-based intelligent transportation systems. In: Intl. Conf. on Intelligent Transportation Systems (ITSC). IEEEp.2663–2668.

[366] Yue, Xiao and Wang, Huiju and Jin, Dawei and Li, Mingqiang and Jiang, Wei. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. Journal of medical systems. 2016;40(10):218.

[367] Abeyratne, Saveen A and Monfared, Radmehr P. Blockchain ready manufacturing supply chain using distributed ledger. 2016;.

[368] Andrychowicz, Marcin and Dziembowski, Stefan and Malinowski, Daniel and Mazurek, Łukasz. Fair two-party computations via bitcoin deposits. In: Interna-

tional Conference on Financial Cryptography and Data Security. Springerp.105–121.

[369] Liang, Xueping and Shetty, Sachin and Tosh, Deepak and Kamhoua, Charles and Kwiat, Kevin and Njilla, Laurent. ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability. Proceedings - 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2017. p.468–477.

[370] Sharma, Pradip Kumar and Moon, Seo Yeon and Park, Jong Hyuk. Block-VN: A distributed blockchain based vehicular network architecture in smart City. JIPS. 2017;13(1):184–195.

[371] Lei, Ao and Cruickshank, Haitham and Cao, Yue and Asuquo, Philip and Ogah, Chibueze P. Anyigor and Sun, Zhili. Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems. IEEE Internet of Things Journal. 2017;4(6):1832–1843.

[372] Ouaddah, Aafaf and Abou Elkalam, Anas and Ait Ouahman, Abdellah. FairAccess: a new Blockchain-based access control framework for the Internet of Things. Security and Communication Networks. 2016;9(18):5943–5964.

[373] Ouaddah, Aafaf and Elkalam, Anas Abou and Ouahman, Abdellah Ait. Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In: Europe and MENA Cooperation Advances in Information and Communication Technologies. Springerp.523–533.

[374] Lee, Boohyung and Lee, Jong Hyouk. Blockchain-based secure firmware update for embedded devices in an Internet of Things environment. Journal of Supercomputing. 2017;73(3):1152–1167.

[375] Pop, Claudia and Cioara, Tudor and Antal, Marcel and Anghel, Ionut and Salomie, Ioan and Bertoncini, Massimo. Blockchain based decentralized

management of demand response programs in smart energy grids. Sensors. 2018;18(1):162.

[376] Ma, Mingxin and Shi, Guozhen and Li, Fenghua. Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario. IEEE Access. 2019;7:34045–34059.

[377] Zheng, Dong and Jing, Chunming and Guo, Rui and Gao, Shiyao and Wang, Liang. A traceable blockchain-based access authentication system with privacy preservation in VANETs. IEEE Access. 2019;7:117716–117726.

[378] Chen, Xiaofeng and Zhang, Xiaohong. Secure electricity trading and incentive contract model for electric vehicle based on energy blockchain. IEEE Access. 2019;7:178763–178778.

[379] Hinarejos, M Francisca and Ferrer-Gomila, Josep-Lluis and Huguet-Rotger, Llorenc. A solution for secure certified electronic mail using blockchain as a secure message board. IEEE Access. 2019;7:31330–31341.

[380] Abou El Houda, Zakaria and Hafid, Abdelhakim Senhaji and Khoukhi, Lyes. Cochain-SC: An intra-and inter-domain DDoS mitigation scheme based on blockchain using SDN and smart contract. IEEE Access. 2019;7:98893–98907.

[381] Wang, Yong and Zhang, Aiqing and Zhang, Peiyun and Wang, Huaqun. Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain. IEEE Access. 2019;7:136704–136719.

[382] Xiong, Ling and Li, Fagen and Zeng, Shengke and Peng, Tu and Liu, Zhicai. A blockchain-based privacy-awareness authentication scheme with efficient revocation for multi-server architectures. IEEE Access. 2019;7:125840–125853.

[383] Chai, Haoye and Leng, Supeng and Zhang, Ke and Mao, Sun. Proof-of-reputation based-consortium blockchain for trust resource sharing in internet of vehicles. IEEE Access. 2019;7:175744–175757.

[384] Daraghmi, Eman-Yasser and Daraghmi, Yousef-Awwad and Yuan, Shyan-Ming. MedChain: a design of blockchain-based system for medical records access and permissions management. IEEE Access. 2019;7:164595–164613.

[385] Ding, Sheng and Cao, Jin and Li, Chen and Fan, Kai and Li, Hui. A novel attribute-based access control scheme using blockchain for IoT. IEEE Access. 2019;7:38431–38441.

[386] Li, Hongzhi and Han, Dezhi. EduRSS: A blockchain-based educational records secure storage and sharing scheme. IEEE Access. 2019;7:179273–179289.

[387] Lu, Xin and Shi, Lingyun and Chen, Zhenyu and Fan, Xunfeng and Guan, Zhitao and Du, Xiaojiang and Guizani, Mohsen. Blockchain-based distributed energy trading in energy Internet: An SDN approach. IEEE Access. 2019;7:173817–173826.

[388] Nguyen, Dinh C and Pathirana, Pubudu N and Ding, Ming and Seneviratne, Aruna. Blockchain for secure ehrs sharing of mobile cloud based e-health systems. IEEE access. 2019;7:66792–66806.

[389] Rajput, Ahmed Raza and Li, Qianmu and Ahvanooey, Milad Taleby and Masood, Isma. EACMS: Emergency access control management system for personal health record based on blockchain. IEEE Access. 2019;7:84304–84317.

[390] She, WEI and Gu, Zhi-Hao and Lyu, Xu-Kang and Liu, QI and Tian, Zhao and Liu, Wei. Homomorphic consortium blockchain for smart home system sensitive data privacy preserving. IEEE Access. 2019;7:62058–62070.

[391] Shi, Leyi and Li, Yang and Liu, Tianxu and Liu, Jia and Shan, Baoying and Chen, Honglong. Dynamic distributed honeypot based on blockchain. IEEE Access. 2019;7:72234–72246.

[392] Sidorov, Michail and Ong, Ming Tze and Sridharan, Ravivarma Vikneswaren and Nakamura, Junya and Ohmura, Ren and Khor, Jing Huey. Ultralightweight

mutual authentication RFID protocol for blockchain enabled supply chains. IEEE Access. 2019;7:7273–7285.

[393] Tang, Fei and Ma, Shuai and Xiang, Yong and Lin, Changlu. An efficient authentication scheme for blockchain-based electronic health records. IEEE access. 2019;7:41678–41689.

[394] Wang, Shuai and Huang, Chenchen and Li, Juanjuan and Yuan, Yong and Wang, Fei-Yue. Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts. IEEE Access. 2019;7:136951–136961.

[395] Wu, Yiming and Tang, Shaohua and Zhao, Bowen and Peng, Zhiniang. BPTM: Blockchain-based privacy-preserving task matching in crowdsourcing. IEEE access. 2019;7:45605–45617.

[396] Xiong, Wei and Xiong, Li. Smart contract based data trading mode using blockchain and machine learning. IEEE Access. 2019;7:102331–102344.

[397] H. Yang and H. Cha and Y. Song. Secure Identifier Management Based on Blockchain Technology in NDN Environment. IEEE Access. 2019;7:6262–6268.

[398] Yang, Yao-Tsung and Chou, Li-Der and Tseng, Chia-Wei and Tseng, Fan-Hsun and Liu, Chien-Chang. Blockchain-based traffic event validation and trust verification for VANETs. IEEE Access. 2019;7:30868–30877.

[399] Yang, Yang and Lin, Hongrui and Liu, Ximeng and Guo, Wenzhong and Zheng, Xianghan and Liu, Zhiquan. Blockchain-based verifiable multi-keyword ranked search on encrypted cloud with fair payment. IEEE Access. 2019;7:140818–140832.

[400] S. Yao and J. Chen and K. He and R. Du and T. Zhu and X. Chen. PBCert: Privacy-Preserving Blockchain-Based Certificate Status Validation Toward Mass Storage Management. IEEE Access. 2019;7:6117–6128.

[401] Zhang, Xiaohong and Chen, Xiaofeng. Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network. IEEE Access. 2019;7:58241–58254.

[402] Zhang, Shaomin and Pu, Miao and Wang, Baoyi and Dong, Bin. A privacy protection scheme of microgrid direct electricity transaction based on consortium blockchain and continuous double auction. IEEE Access. 2019;7:151746–151753.

[403] Hao, Kun and Xin, Junchang and Wang, Zhiqiong and Cao, Keyan and Wang, Guoren. Blockchain-based outsourced storage schema in untrusted environment. IEEE Access. 2019;7:122707–122721.

[404] Yao, Yingying and Chang, Xiaolin and Mišić, Jelena and Mišić, Vojislav B and Li, Lin. BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services. IEEE Internet of Things Journal. 2019;6(2):3775–3784.

[405] S. Biswas and K. Sharif and F. Li and B. Nour and Y. Wang. A Scalable Blockchain Framework for Secure Transactions in IoT. IEEE Internet of Things Journal. 2019;6(3):4650–4659.

[406] K. Fan and S. Wang and Y. Ren and K. Yang and Z. Yan and H. Li and Y. Yang. Blockchain-Based Secure Time Protection Scheme in IoT. IEEE Internet of Things Journal. 2019;6(3):4671–4679.

[407] Gai, Keke and Wu, Yulu and Zhu, Liehuang and Xu, Lei and Zhang, Yan. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. IEEE Internet of Things Journal. 2019;6(5):7992–8004.

[408] J. Kang and R. Yu and X. Huang and M. Wu and S. Maharjan and S. Xie and Y. Zhang. Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks. IEEE Internet of Things Journal. 2019;6(3):4660–4670.

[409] M. Li and L. Zhu and X. Lin. Efficient and Privacy-Preserving Carpooling Using Blockchain-Assisted Vehicular Fog Computing. IEEE Internet of Things Journal. 2019;6(3):4573–4584.

[410] O. Novo. Scalable Access Management in IoT Using Blockchain: A Performance Evaluation. IEEE Internet of Things Journal. 2019;6(3):4694–4701.

[411] J. Pan and J. Wang and A. Hester and I. Alqerm and Y. Liu and Y. Zhao. EdgeChain: An Edge-IoT Framework and Prototype Based on Blockchain and Smart Contracts. IEEE Internet of Things Journal. 2019;6(3):4719–4732.

[412] Z. Su and Y. Wang and Q. Xu and M. Fei and Y. Tian and N. Zhang. A Secure Charging Scheme for Electric Vehicles With Smart Communities in Energy Blockchain. IEEE Internet of Things Journal. 2019;6(3):4601–4613.

[413] Wang, Huaqun and Wang, Qihua and He, Debiao and Li, Qi and Liu, Zhe. BBARS: Blockchain-based anonymous rewarding scheme for V2G networks. IEEE Internet of Things Journal. 2019;6(2):3676–3687.

[414] Xu, Jie and Xue, Kaiping and Li, Shaohua and Tian, Hangyu and Hong, Jianan and Hong, Peilin and Yu, Nenghai. Healthchain: A blockchain-based privacy preserving scheme for large-scale health data. IEEE Internet of Things Journal. 2019;6(5):8770–8781.

[415] Z. Yang and K. Yang and L. Lei and K. Zheng and V. C. M. Leung. Blockchain-Based Decentralized Trust Management in Vehicular Networks. IEEE Internet of Things Journal. 2019;6(2):1495–1505.

[416] K. Zhu and Z. Chen and W. Yan and L. Zhang. Security Attacks in Named Data Networking of Things and a Blockchain Solution. IEEE Internet of Things Journal. 2019;6(3):4733–4741.

[417] Kochovski, Petar and Gec, Sandi and Stankovski, Vlado and Bajec, Marko and Drobintsev, Pavel D. Trust management in a blockchain based fog computing

platform with trustless smart oracles. Future Generation Computer Systems. 2019;101:747–759.

[418] Feng, Wei and Yan, Zheng. MCS-Chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain. Future Generation Computer Systems. 2019;95:649–666.

[419] Lanxiang Chen and Wai-Kong Lee and Chin-Chen Chang and Kim-Kwang Raymond Choo and Nan Zhang. Blockchain based searchable encryption for electronic health record sharing. Future Generation Computer Systems. 2019;95:420 – 429. Available from: `http://www.sciencedirect.com/science/article/pii/S0167739X18314134`.

[420] Al Omar, Abdullah and Bhuiyan, Md Zakirul Alam and Basu, Anirban and Kiyomoto, Shinsaku and Rahman, Mohammad Shahriar. Privacy-friendly platform for healthcare data in cloud based on blockchain environment. Future generation computer systems. 2019;95:511–521.

[421] Yang, Mengmeng and Zhu, Tianqing and Liang, Kaitai and Zhou, Wanlei and Deng, Robert H. A blockchain-based location privacy-preserving crowdsensing system. Future Generation Computer Systems. 2019;94:408–418.

[422] Zhu, Liehuang and Wu, Yulu and Gai, Keke and Choo, Kim-Kwang Raymond. Controllable and trustworthy blockchain-based cloud data management. Future Generation Computer Systems. 2019;91:527–535.

[423] Wan, Jiafu and Li, Jiapeng and Imran, Muhammad and Li, Di and others. A blockchain-based solution for enhancing security and privacy in smart factory. IEEE Transactions on Industrial Informatics. 2019;15(6):3652–3660.

[424] Liu, Dongxiao and Alahmadi, Amal and Ni, Jianbing and Lin, Xiaodong and Shen, Xuemin. Anonymous reputation system for IIoT-enabled retail mar-

keting atop PoS blockchain. IEEE Transactions on Industrial Informatics. 2019;15(6):3527–3537.

[425] P. K. Sharma and N. Kumar and J. H. Park. Blockchain-Based Distributed Framework for Automotive Industry in a Smart City. IEEE Transactions on Industrial Informatics. 2019;15(7):4197–4205.

[426] Wang, Yuntao and Su, Zhou and Zhang, Ning. BSIS: Blockchain-based secure incentive scheme for energy delivery in vehicular energy network. IEEE Transactions on Industrial Informatics. 2019;15(6):3620–3631.

[427] Xu, Jinliang and Wang, Shangguang and Bhargava, Bharat K and Yang, Fangchun. A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing. IEEE Transactions on Industrial Informatics. 2019;15(6):3538–3547.

[428] Jiang, Li and Xie, Shengli and Maharjan, Sabita and Zhang, Yan. Blockchain empowered wireless power transfer for green and secure Internet of Things. IEEE Network. 2019;33(6):164–171.

[429] Shen, Meng and Deng, Yawen and Zhu, Liehuang and Du, Xiaojiang and Guizani, Nadra. Privacy-preserving image retrieval for medical IoT systems: A blockchain-based approach. IEEE Network. 2019;33(5):27–33.

[430] Ali, Gauhar and Ahmad, Naveed and Cao, Yue and Ali, Qazi Ejaz and Azim, Fazal and Cruickshank, Haitham. BCON: Blockchain based access CONtrol across multiple conflict of interest domains. Journal of Network and Computer Applications. 2019;147:102440.

[431] Roy, Deepsubhra Guha and Das, Puja and De, Debashis and Buyya, Rajkumar. QoS-aware secure transaction framework for internet of things using blockchain mechanism. Journal of Network and Computer Applications. 2019;144:59–78.

[432] Hu, Jiaxi and He, Debiao and Zhao, Qinglan and Choo, Kim-Kwang Raymond. Parking management: A blockchain-based privacy-preserving system. IEEE Consumer Electronics Magazine. 2019;8(4):45–49.

[433] Paliokas, Ioannis and Tsoniotis, Nikolaos and Votis, Konstantinos and Tzovaras, Dimitrios. A blockchain platform in connected medical-device environments: Trustworthy technology to guard against cyberthreats. IEEE Consumer Electronics Magazine. 2019;8(4):50–55.

[434] Wang, Qiping and Lau, Raymond Yiu Keung and Mao, Xudong. Blockchain-enabled smart contracts for enhancing distributor-to-consumer transactions. IEEE Consumer Electronics Magazine. 2019;8(6):22–28.

[435] Chen, Yun and Xie, Hui and Lv, Kun and Wei, Shengjun and Hu, Changzhen. DEPLEST: A blockchain-based privacy-preserving distributed database toward user behaviors in social networks. Information Sciences. 2019;501:100–117.

[436] Cao, Sheng and Zhang, Gexiang and Liu, Pengfei and Zhang, Xiaosong and Neri, Ferrante. Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain. Information Sciences. 2019;485:427–440.

[437] Conti, Mauro and Hassan, Muhammad and Lal, Chhagan. BlockAuth: BlockChain based distributed producer authentication in ICN. Computer Networks. 2019;164:106888.

[438] An, Jian and Yang, He and Gui, Xiaolin and Zhang, Wendong and Gui, Ruowei and Kang, Jingjing. TCNS: node selection with privacy protection in crowdsensing based on twice consensuses of blockchain. IEEE Transactions on Network and Service Management. 2019;16(3):1255–1267.

[439] Zhu, Ruiyu and Ding, Changchang and Huang, Yan. Efficient publicly verifiable 2pc over a blockchain with applications to financially-secure computations.

In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Securityp.633–650.

[440] Sani, Abubakar Sadiq and Yuan, Dong and Bao, Wei and Yeoh, Phee Lep and Dong, Zhao Yang and Vucetic, Branka and Bertino, Elisa. Xyreum: A high-performance and scalable blockchain for iiot security and privacy. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEEp.1920–1930.

[441] Faber, Benedict and Michelet, Georg Cappelen and Weidmann, Niklas and Mukkamala, Raghava Rao and Vatrapu, Ravi. BPDIMS: A blockchain-based personal data and identity management system. In: Proceedings of the 52nd Hawaii International Conference on System Sciences; 2019. .

[442] Zhang, Ce and Xu, Cheng and Xu, Jianliang and Tang, Yuzhe and Choi, Byron. Gemˆ 2-tree: A gas-efficient structure for authenticated range queries in blockchain. In: 2019 IEEE 35th international conference on data engineering (ICDE). IEEEp.842–853.

[443] Zhou, Huan and Ouyang, Xue and Ren, Zhijie and Su, Jinshu and de Laat, Cees and Zhao, Zhiming. A blockchain based witness model for trustworthy cloud service level agreement enforcement. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEEp.1567–1575.

[444] Duan, Huayi and Zheng, Yifeng and Du, Yuefeng and Zhou, Anxin and Wang, Cong and Au, Man Ho. Aggregating crowd wisdom via blockchain: A private, correct, and robust realization. In: 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom. IEEEp.1–10.

[445] Li, Chun-Ta and Shih, Dong-Her and Wang, Chun-Cheng and Chen, Chin-Ling and Lee, Cheng-Chi. A Blockchain Based Data Aggregation and Group Au-

thentication Scheme for Electronic Medical System. IEEE Access. 2020;8:173904–173917.

[446] Cai, Ting and Yang, Zetao and Chen, Wuhui and Zheng, Zibin and Yu, Yang. A blockchain-assisted trust access authentication system for solid. IEEE Access. 2020;8:71605–71616.

[447] Hojjati, Maede and Shafieinejad, Alireza and Yanikomeroglu, Halim. A Blockchain-Based Authentication and Key Agreement (AKA) Protocol for 5G Networks. IEEE Access. 2020;8:216461–216476.

[448] Liu, Ziming and Wang, Dazhi and Wang, Jiaxing and Wang, Xinghua and Li, Hao. A blockchain-enabled secure power trading mechanism for smart grid employing wireless networks. IEEE Access. 2020;8:177745–177756.

[449] Xiang, Xinyin and Wang, Mingyu and Fan, Weiguo. A Permissioned Blockchain-Based Identity Management and User Authentication Scheme for E-Health Systems. IEEE Access. 2020;8:171771–171783.

[450] Zeng, Pengjie and Wang, Xiaoliang and Li, Hao and Jiang, Frank and Doss, Robin. A Scheme of Intelligent Traffic Light System Based on Distributed Security Architecture of Blockchain Technology. IEEE Access. 2020;8:33644–33657.

[451] Xiao, Lijun and Han, Dezhi and Meng, Xiangwei and Liang, Wei and Li, Kuan-Ching. A Secure Framework for Data Sharing in Private Blockchain-Based WBANs. IEEE Access. 2020;8:153956–153968.

[452] Hinarejos, M Francisca and Ferrer-Gomila, Josep-Lluis. A Solution for Secure Multi-Party Certified Electronic Mail Using Blockchain. IEEE Access. 2020;8:102997–103006.

[453] Yang, Caixia and Tan, Liang and Shi, Na and Xu, Bolei and Cao, Yang and Yu, Keping. AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud. IEEE Access. 2020;8:70604–70615.

[454] Garg, Neha and Wazid, Mohammad and Das, Ashok Kumar and Singh, Devesh Pratap and Rodrigues, Joel JPC and Park, Youngho. BAKMP-IoMT: Design of blockchain enabled authenticated key management protocol for Internet of medical things deployment. IEEE Access. 2020;8:95956–95977.

[455] Xu, Hong and He, Qian and Li, Xuecong and Jiang, Bingcheng and Qin, Kuangyu. BDSS-FA: A blockchain-based data security sharing platform with fine-grained access control. IEEE Access. 2020;8:87552–87561.

[456] Zerka, Fadila and Urovi, Visara and Vaidyanathan, Akshayaa and Barakat, Samir and Leijenaar, Ralph TH and Walsh, Sean and Gabrani-Juma, Hanif and Miraglio, Benjamin and Woodruff, Henry C and Dumontier, Michel and others. Blockchain for Privacy Preserving and Trustworthy Distributed Machine Learning in Multicentric Medical Imaging (C-DistriM). IEEE Access. 2020;8:183939–183951.

[457] Sun, Jin and Yao, Xiaomin and Wang, Shangping and Wu, Ying. Blockchain-based secure storage and access scheme for electronic medical records in IPFS. IEEE Access. 2020;8:59389–59401.

[458] Hosen, ASM Sanwar and Singh, Saurabh and Sharma, Pradip Kumar and Ghosh, Uttam and Wang, Jin and Ra, In-Ho and Cho, Gi Hwan. Blockchain-Based Transaction Validation Protocol for a Secure Distributed IoT Network. IEEE Access. 2020;8:117266–117277.

[459] Liao, Chia-Hung and Lin, Hui-En and Yuan, Shyan-Ming. Blockchain-Enabled Integrated Market Platform for Contract Production. IEEE Access. 2020;8:211007–211027.

[460] Kakei, Shohei and Shiraishi, Yoshiaki and Mohri, Masami and Nakamura, Toru and Hashimoto, Masayuki and Saito, Shoichi. Cross-Certification Towards

Distributed Authentication Infrastructure: A Case of Hyperledger Fabric. IEEE Access. 2020;8:135742–135757.

[461] Chen, Yuling and Yin, Hongyan and Xiang, Yuexin and Ren, Wei and Ren, Yi and Xiong, Neal Naixue. CVT: A Crowdsourcing Video Transcoding Scheme Based on Blockchain Smart Contracts. IEEE Access. 2020;8:220672–220681.

[462] Miao, Ying and Huang, Qiong and Xiao, Meiyan and Li, Hongbo. Decentralized and Privacy-Preserving Public Auditing for Cloud Storage Based on Blockchain. IEEE Access. 2020;8:139813–139826.

[463] Long, Yangyang and Chen, Yuling and Ren, Wei and Dou, Hui and Xiong, Neal Naixue. DePET: A Decentralized Privacy-Preserving Energy Trading Scheme for Vehicular Energy Network via Blockchain and K-Anonymity. IEEE Access. 2020;8:192587–192596.

[464] Son, Seunghwan and Lee, Joonyoung and Kim, Myeonghyun and Yu, Sungjin and Das, Ashok Kumar and Park, Youngho. Design of Secure Authentication Protocol for Cloud-Assisted Telecare Medical Information System Using Blockchain. IEEE Access. 2020;8:192177–192191.

[465] Gu, Ai and Yin, Zhenyu and Cui, Chuanyu and Li, Yue. Integrated Functional Safety and Security Diagnosis Mechanism of CPS Based on Blockchain. IEEE Access. 2020;8:15241–15255.

[466] Debe, Mazin and Salah, Khaled and Rehman, Muhammad Habib Ur and Svetinovic, Davor. Monetization of services provided by public fog nodes using blockchain and smart contracts. IEEE Access. 2020;8:20118–20128.

[467] Pinheiro, Alexandre and Canedo, Edna Dias and De Sousa, Rafael Timóteo and Albuquerque, Robson De Oliveira. Monitoring File Integrity Using Blockchain and Smart Contracts. IEEE Access. 2020;8:198548–198579.

[468] Tomaz, Antonio Emerson Barros and Do Nascimento, José Cláudio and Hafid, Abdelhakim Senhaji and De Souza, Jose Neuman. Preserving Privacy in Mobile Health Systems Using Non-Interactive Zero-Knowledge Proof and Blockchain. IEEE Access. 2020;8:204441–204458.

[469] Ernest, Bonnah and Shiguang, Ju. Privacy Enhancement Scheme (PES) in a Blockchain-Edge Computing Environment. IEEE Access. 2020;8:25863–25876.

[470] Li, Wanxin and Guo, Hao and Nejad, Mark and Shen, Chien-Chung. Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach. IEEE Access. 2020;8:181733–181743.

[471] Rahman, Mohamed Abdur and Hossain, M Shamim and Islam, Mohammad Saiful and Alrajeh, Nabil A and Muhammad, Ghulam. Secure and provenance enhanced internet of health things framework: A blockchain managed federated learning approach. Ieee Access. 2020;8:205071–205087.

[472] Tan, Haowen and Chung, Ilyong. Secure authentication and key management with blockchain in vanets. IEEE Access. 2019;8:2482–2498.

[473] Wang, Di and Zhang, Xiaohong. Secure data sharing and customized services for intelligent transportation based on a consortium blockchain. IEEE Access. 2020;8:56045–56059.

[474] Liu, Bin and Xiao, Lijun and Long, Jing and Tang, Mingdong and Hosam, Osama. Secure digital certificate-based data access control scheme in blockchain. IEEE Access. 2020;8:91751–91760.

[475] Alghamdi, Turki Ali and Ali, Ishtiaq and Javaid, Nadeem and Shafiq, Muhammad. Secure service provisioning scheme for lightweight IoT devices with a fair payment system and an incentive mechanism based on blockchain. IEEE Access. 2019;8:1048–1061.

[476] Sheikh, A and Kamuni, V and Urooj, Asfia and Wagh, S and Singh, N and Patel, Dhiren. Secured energy trading using byzantine-based blockchain consensus. IEEE Access. 2019;8:8554–8571.

[477] Zghaibeh, Manaf and Farooq, Umer and Hasan, Najam Ul and Baig, Imran. SHealth: A blockchain-based health system with smart contracts capabilities. IEEE Access. 2020;8:70030–70043.

[478] Liu, Tonglai and Wu, Jigang and Chen, Long and Wu, Yalan and Li, Yinan. Smart Contract-Based Long-Term Auction for Mobile Blockchain Computation Offloading. IEEE Access. 2020;8:36029–36042.

[479] Badr, Mahmoud M and Al Amiri, Wesam and Fouda, Mostafa M and Mahmoud, Mohamed MEA and Aljohani, Abdulah Jeza and Alasmary, Waleed. Smart Parking System With Privacy Preservation and Reputation Management Using Blockchain. IEEE Access. 2020;8:150823–150843.

[480] Wang, Qianlong and Ji, Tianxi and Guo, Yifan and Yu, Lixing and Chen, Xuhui and Li, Pan. TrafficChain: A Blockchain-Based Secure and Privacy-Preserving Traffic Map. IEEE Access. 2020;8:60598–60612.

[481] Oprea, Simona-Vasilica and Bâra, Adela and Andreescu, Anca Ioana. Two Novel Blockchain-Based Market Settlement Mechanisms Embedded Into Smart Contracts for Securely Trading Renewable Energy. IEEE Access. 2020;8:212548–212556.

[482] Gauhar, Ali and Ahmad, Naveed and Cao, Yue and Khan, Shahzad and Cruickshank, Haitham and Qazi, Ejaz Ali and Ali, Azaz. xDBAuth: Blockchain based cross domain authentication and authorization framework for Internet of Things. IEEE Access. 2020;8:58800–58816.

[483] Liu, Xingchen and Huang, Haiping and Xiao, Fu and Ma, Ziyang. A blockchain-based trust management with conditional privacy-preserving

announcement scheme for VANETs. IEEE Internet of Things Journal. 2019;7(5):4101–4112.

[484] Zhaofeng, Ma and Lingyun, Wang and Xiaochang, Wang and Zhen, Wang and Weizhe, Zhao. Blockchain-enabled decentralized trust management and secure usage control of IoT big data. IEEE Internet of Things Journal. 2019;7(5):4000–4015.

[485] Medhane, Darshan Vishwasrao and Sangaiah, Arun Kumar and Hossain, M Shamim and Muhammad, Ghulam and Wang, Jin. Blockchain-Enabled Distributed Security Framework for Next-Generation IoT: An Edge Cloud and Software-Defined Network-Integrated Approach. IEEE Internet of Things Journal. 2020;7(7):6143–6149.

[486] Liu, Dongxiao and Ni, Jianbing and Huang, Cheng and Lin, Xiaodong and Shen, Xuemin Sherman. Secure and efficient distributed network provenance for IoT: A blockchain-based approach. IEEE Internet of Things Journal. 2020;7(8):7564–7574.

[487] Yazdinejad, Abbas and Srivastava, Gautam and Parizi, Reza M and Dehghantanha, Ali and Choo, Kim-Kwang Raymond and Aledhari, Mohammed. Decentralized authentication of distributed patients in hospital networks using blockchain. IEEE journal of biomedical and health informatics. 2020;24(8):2146–2156.

[488] Zhou, Sicong and Huang, Huawei and Chen, Wuhui and Zhou, Pan and Zheng, Zibin and Guo, Song. Pirate: A blockchain-based secure framework of distributed machine learning in 5g networks. IEEE Network. 2020;34(6):84–91.

[489] Tian, Youliang and Wang, Zuan and Xiong, Jinbo and Ma, Jianfeng. A blockchain-based secure key management scheme with trustworthiness in DWSNs. IEEE Transactions on Industrial Informatics. 2020;16(9):6193–6202.

[490] Guo, Shaoyong and Hu, Xing and Guo, Song and Qiu, Xuesong and Qi, Feng. Blockchain meets edge computing: A distributed and trusted authentication system. IEEE Transactions on Industrial Informatics. 2019;16(3):1972–1983.

[491] Cui, Hui and Wan, Zhiguo and Wei, Xinlei and Nepal, Surya and Yi, Xun. Pay as you decrypt: Decryption outsourcing for functional encryption using blockchain. IEEE Transactions on Information Forensics and Security. 2020;15:3227–3238.

[492] Cui, Zhihua and Fei, XUE and Zhang, Shiqiang and Cai, Xingjuan and Cao, Yang and Zhang, Wensheng and Chen, Jinjun. A hybrid BlockChain-based identity authentication scheme for multi-WSN. IEEE Transactions on Services Computing. 2020;13(2):241–251.

[493] Pournaghi, Seyed Morteza and Bayat, Majid and Farjami, Yaghoub. MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption. Journal of Ambient Intelligence and Humanized Computing. p.1–29.

[494] Lyu, Qiuyun and Qi, Yizhen and Zhang, Xiaochen and Liu, Huaping and Wang, Qiuhua and Zheng, Ning. SBAC: A secure blockchain-based access control framework for information-centric networking. Journal of Network and Computer Applications. 2020;149:102444.

[495] Stach, Christoph and Gritti, Clémentine and Przytarski, Dennis and Mitschang, Bernhard. Trustworthy, Secure, and Privacy-aware Food Monitoring Enabled by Blockchains and the IoT. In: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEEp.1–4.

[496] Meskauskas, Tomas. Ransomware information. PCRisk;. Available from:

https://www.pcrisk.com/search?searchword=ransomware&ordering=&searchphrase=all.

[497] Labs, Malwarebytes and Malwarebytes Labs. Explained: Spora ransomware: Malwarebytes labs;. Available from: https://www.malwarebytes.com/blog/news/2017/03/spora-ransomware.

[498] Dan Goodin. Meet Jigsaw, the ransomware that taunts victims and offers live support; 2016. Available from: https://arstechnica.com/information-technology/2016/06/meet-jigsaw-the-ransomware-that-taunts-victims-and-offers-live-support/.

[499] Settle, Andy and Leonard, Carl. Piecing together the jigsaw puzzle; 2019. Available from: https://www.forcepoint.com/es/blog/x-labs/piecing-together-jigsaw-puzzle.

[500] Oosthoek, Kris and Cable, Jack and Smaragdakis, Georgios. A Tale of Two Markets: Investigating the Ransomware Payments Economy. arXiv preprint arXiv:220505028. 2022;.

[501] TorrentLocker: Crypto-ransomware still active, using same tactics; 2016. Available from: https://www.welivesecurity.com/2016/09/01/torrentlocker-crypto-ransomware-still-active-using-tactics/.

[502] Gomez, Gibran and Moreno-Sanchez, Pedro and Caballero, Juan. Detecting Cybercriminal Bitcoin Relationships through Backwards Exploration. arXiv preprint arXiv:220600375. 2022;.

[503] Paganini, Pierluigi. EDA2, derived from the educational ransomware, is easy to break; 2016. Available from: https://securityaffairs.co/wordpress/45336/malware/eda2-easy-decryption.html.

[504] CagedTech. Flyper ransomware. EnigmaSoft; 2020. Available from: https://www.enigmasoftware.com/flyperransomware-removal/.

[505] Demonslay335. TowerWeb ransomware help; 2016. Available from: https://www.bleepingcomputer.com/forums/t/618055/towerweb-ransomware-help-support-topic-payment-instructionsjpg/.

[506] Bucbi ransomware spreading via RDP brute force attacks;. Available from: https://www.securityweek.com/bucbi-ransomware-spreading-rdp-brute-force-attacks.

[507] Abrams, Lawrence. CryptoHost decrypted: Locks files in a password protected RAR file. BleepingComputer; 2016. Available from: https://www.bleepingcomputer.com/news/security/cryptohost-decrypted-locks-files-in-a-password-protected-rar-file/.

[508] Malanga, Matt. Everything you wanted to know about Doxware; 2017. Available from: https://monstercloud.com/blog/2017/02/17/what-is-doxware/.

[509] GoldSparrow. Korean Adamlocker ransomware. EnigmaSoft; 2020. Available from: https://www.enigmasoftware.com/koreanadamlockerransomware-removal/.

[510] Alphabet ransomware virus (removal steps and protection updates); 2017. Available from: https://bestsecuritysearch.com/alphabet-ransomware-virus-removal-steps-protection-updates/.

[511] Morelli, Olivia. Remove cryptconsole ransomware / virus (removal instructions) - jun 2018 update. 2-spyware.com; 2018. Available from: https://www.2-spyware.com/remove-cryptconsole-ransomware-virus.html.

[512] Krastev, Ventsislav. Exotic 3.0 ransomware delete and fix the affected data; 2017. Available from: https://sensorstechforum.com/exotic-3-0-ransomware-delete-fix-affected-data/.

[513] Ramos, Nicholas. Fakeglobe and Cerber Ransomware Sneaking under the radar while WeCry; 2017. Available from: https://www.trustwave.com/en-us/

resources/blogs/spiderlabs-blog/fakeglobe-and-cerber-ransomware-sneaking-under-the-radar-while-wecry/.

[514] Bilbao, Berta. New fantom virus - remove and restore .locked files; 2017. Available from: https://sensorstechforum.com/new-fantom-virus-remove-restore-locked-files/.

[515] Globe2 Ransomware;. Available from: https://anti-spyware-101.com/remove-globe2-ransomware.

[516] Ramsomeer ransomware;. Available from: https://ransomware.fandom.com/wiki/Ramsomeer_Ransomware.

[517] l1me_expl0it;. Available from: https://www.virustotal.com/gui/search/l1me_expl0it/comments.

[518] Krastev, Ventsislav. Remove Nemucod ransomware and restore .crypted encrypted files; 2017. Available from: https://sensorstechforum.com/remove-nemucod-ransomware-and-restore-crypted-encrypted-files/.

[519] 25, April and Staff, Proofpoint. Philadelphia ransomware brings customization to commodity malware: Proofpoint US; 2019. Available from: https://www.proofpoint.com/us/threat-insight/post/philadelphia-ransomware-customization-commodity-malware.

[520] GoldSparrow. Popcorn time ransomware. EnigmaSoft; 2020. Available from: https://www.enigmasoftware.com/popcorntimeransomware-removal/.

[521] Grinler. Rush (.crashed) ransomware; 2016. Available from: https://www.bleepingcomputer.com/forums/t/620829/rush-crashed-ransomware-help-support-topic-decrypt-your-fileshtml/.

[522] Woods, Alice. Remove stupid ransomware / virus (virus removal guide) - jul 2020 update. 2-spyware.com; 2020. Available from: https://www.2-spyware.com/remove-stupid-ransomware-virus.html.

[523] Remove the XTP Locker 5.0 ransomware from your PC; 2017. Available from: https://bestsecuritysearch.com/remove-xtp-locker-5-0-ransomware-pc/.

[524] Sjouwerman, Stu. Microsoft alert: Zcryptor Ransomware with Worm Feature;. Available from: https://blog.knowbe4.com/microsoft-alert-zcryptor-ransomware-with-worm-feature.

[525] Cimpanu, Catalin. KillDisk ransomware now targets linux, prevents boot-up, has faulty encryption. BleepingComputer; 2017. Available from: https://www.bleepingcomputer.com/news/security/killdisk-ransomware-now-targets-linux-prevents-boot-up-has-faulty-encryption/.

[526] New crypto-ransomware hits macos; 2022. Available from: https://www.welivesecurity.com/2017/02/22/new-crypto-ransomware-hits-macos/.

[527] Abrams, Lawrence. New LLTP ransomware appears to be a rewritten Venus Locker. BleepingComputer; 2017. Available from: https://www.bleepingcomputer.com/news/security/new-lltp-ransomware-appears-to-be-a-rewritten-venus-locker/.

[528] Cimpanu, Catalin. Android DoubleLocker ransomware activates every time you hit home button. BleepingComputer; 2017. Available from: https://www.bleepingcomputer.com/news/security/android-doublelocker-ransomware-activates-every-time-you-hit-home-button/.

[529] Intel, Elliptic. Revil revealed - tracking a ransomware negotiation and payment;. Available from: https://www.elliptic.co/blog/revil-revealed-tracking-ransomware-negotiation-and-payment.

[530] Majauskas, Giedrius. Hc6 ransomware; 2017. Available from: https://www.2-viruses.com/remove-hc6-ransomware.

[531] Satheesh Kumar, M. and Ben-Othman, Jalel and Srinivasagan, K. G. . An Investigation on Wannacry Ransomware and its Detection. In: 2018 IEEE Symposium on Computers and Communications (ISCC)p.1–6.

Appendix A

Annex

A.1 Tables for Contribution 1

The analysis of all the academic papers analyzed in the Section 3.2 of Chapter 3 is depicted in Tables A.1, A.2, A.3, A.4, A.5 and A.6.

Table A.1: Analysis of academic papers (I), where - means not specified

Name	Area	Cybersecurity properties			Blockchain technology	Type of network		Justification
		Conf.	Authen.	Non- repudiation		Nature	Permissions	
(109)	Independent	Complete	-	-	Bitcoin-based	Public	Permissionless	Complete
(149)	IoT	-	-	-	Bitcoin-based	Public	Permissionless	Complete
(310)	Independent	-	-	-	Ad-hoc	Public	Permissionless	Complete
(311)	Independent	-	Complete	Complete	Ethereum-based	Private	Permissionless	Complete
(111)	IoT	-	Complete	Complete	Ad-hoc	Private	Permissionless	Partial
(153)	IoT	-	Complete	Complete	Ad-hoc	Private	Permissionless	Partial
(185)	Distributed/Cloud computing	Partial	-	-	Bitcoin-based	Public	Permissionless	Complete
(177)	IoT	-	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(312)	IoT, Distributed/Cloud computing	-	Complete	Complete	Ad-hoc	Private	Permissionless	Complete
(114)	IoT	Complete	Complete	Complete	Ad-hoc	Private	Permissioned	Complete
(143)	Independent	Complete	Complete	Complete	Bitcoin-based	Private	Permissioned	Complete
(313)	IoT	-	Complete	-	Ad-hoc	Private, Public	Permissioned, Permissionless	Complete
(165)	Energy	-	Complete	-	Bitcoin-based	Private	Permissionless	Complete
(139)	IoT	Complete	Partial	Partial	Based on other technologies	Public	Permissioned	Complete
(189)	Independent	-	-	-	Ad-hoc	Any	Any	No
(314)	Independent	Complete	-	Complete	Hyperledger Project	-	Permissioned	Complete
(131)	Energy	Complete	Complete	Complete	Ad-hoc	Private	-	Complete
(176)	Independent	-	Complete	Partial	Any	Private, Public	Permissioned, Permissionless	Complete
(166)	Health	Complete	Complete	Complete	Ad-hoc	Private	Permissioned	Partial
(158)	E-commerce	Partial	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(186)	Independent	Complete	-	-	Bitcoin-based	Public	Permissionless	Complete
(315)	Energy, IoT	-	Complete	Complete	Bitcoin-based	Private	Permissionless	Complete
(137)	IoT	-	Complete	Complete	-	-	-	Complete
(173)	Health, IoT, Distributed/Cloud computing	Complete	Complete	Complete	Ethereum-based, Hyperledger Project	Private	Permissioned	Complete
(99)	Health, IoT, Distributed/Cloud computing	-	Complete	Partial	Ethereum-based,Hyperledger Project	Private	Permissioned	Partial
(316)	IoT, Distributed/Cloud computing	-	-	-	Ad-hoc	Public	Permissionless	Complete
(169)	Independent	Complete	Complete	Partial	Bitcoin-based	Private	Permissionless	Complete
(108)	IoT, E-commerce,Distributed/cloud computing	Complete	Partial	Partial	Bitcoin-based	-	Permissionless	Partial
(132)	Independent	Complete	Complete	Complete	Ethereum-based	Private	Permissionless	Complete
(317)	Independent	-	Complete	Complete	Bitcoin-based	Private	Permissionless	Complete
(105)	E-commerce	-	Complete	Complete	Bitcoin-based	Public	Permissionless	Complete
(175)	Independent	Complete	-	-	Any	Public, Private	Permissionless, -	Complete
(100)	IoT	Complete	Complete	Complete	Ethereum-based	Private	Permissioned	Partial
(133)	Independent	Complete	Complete	Complete	Bitcoin-based	Private	Permissionless	Partial
(126)	IoT	Complete	Complete	Partial	-	Private	-	Partial
(318)	Independent	-	-	-	Ad-hoc	Public	Permissionless	Complete
(319)	E-commerce	Complete	-	-	Bitcoin-based	Public	Permissionless	Complete
(150)	IoT	Complete	Complete	Complete	Hyperledger Project	-	Permissioned	Partial
(140)	E-commerce, IoT	Complete	-	-	Any	-	Permissioned	Partial
(320)	IoT	Complete	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(321)	Independent	Complete	Complete	Complete	-	Private	-	Complete
(322)	Independent	Complete	Complete	Complete	-	-	Permissioned	Complete
(323)	IoT	-	Complete	Complete	Any	Private	Permissioned	Complete
(127)	Energy	-	Complete	Partial	Ethereum-based	Public	Permissionless	Complete
(324)	IoT	Partial	Complete	Partial	Ad-hoc	Private	Permissionless	Partial
(171)	Independent	-	Complete	Complete	Ethereum-based	Public	Permissionless	Complete

Table A.2: Analysis of academic papers (II), where - means not specified

Name	Area	Cybersecurity properties			Blockchain technology	Type of network		Justification
		Conf.	Authen.	Non- repudiation		Nature	Permissions	
(141)	E-commerce	Complete	-	-	Bitcoin-based	Public	Permissionless	Complete
(184)	Independent	-	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(325)	Independent	-	Partial	Complete	Any	-	-	Complete
(326)	Independent	-	Complete	Partial	-	-	-	Partial
(181)	IoT	Complete	-	-	Bitcoin-based	Public	Permissionless	No
(118)	Independent	Complete	Complete	-	Bitcoin-based	Public	Permissionless	Complete
(327)	E-commerce	Complete	-	-	Bitcoin-based	Public	Permissionless	Complete
(110)	E-commerce, Distributed/Cloud computing	-	-	-	Bitcoin-based	Public	Permissionless	Complete
(328)	E-commerce	Partial, Complete	-	-	Bitcoin-based	Public	Permissionless	Complete
(329)	Independent	-	Complete	Complete	Bitcoin-based	Private	-	Partial
(330)	Independent	-	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(331)	Distributed/Cloud computing	Complete	Complete	-	Ethereum-based	Private	Permissionless	Complete
(332)	Independent	-	-	-	Ad-hoc	Public	Permissionless	Complete
(172)	Independent	-	-	-	Bitcoin-based	Public	Permissionless	Complete
(333)	IoT	-	Complete	Complete	Bitcoin-based	Public	Permissionless	Partial
(134)	Independent	Complete	Complete	Complete	Based on other technologies	Public	-	Complete
(334)	Distributed/Cloud computing	-	Complete	Partial	Ethereum-based	Public	Permissionless	Complete
(335)	Distributed/Cloud computing	-	-	Partial	Ethereum-based	Public	Permissionless	Complete
(336)	Health, IoT	-	Complete	Complete	Ethereum-based	Private	Permissioned	Complete
(337)	Energy	-	-	-	-	-	Permissionless	Complete
(113)	Independent	-	Complete	Complete	Bitcoin-based	Public	Permissionless	Partial
(182)	Independent	Complete	Complete	-	Based on other technologies	Public	Permissionless	Complete
(136)	Distributed/Cloud computing	Partial	Complete	Complete	Ad-hoc	Private	Permissionless	Complete
(338)	Energy, E-commerce	Complete	Complete	Complete	Ad-hoc	Private	Permissionless	Partial
(101)	Health	-	Complete	Complete	Hyperledger Project	Private	Permissioned	No
(339)	Health,IoT	-	Complete	-	Hyperledger Project	Private	Permissioned	Complete
(340)	IoT	-	Complete	Complete	Any	Public	Any	Complete
(61)	Health,IoT	Partial	Complete	Complete	Ethereum-based	Private	-	Complete
(341)	IoT	Partial	Complete	Partial	Ethereum-based	Private	Permissioned	Complete
(342)	E-commerce	Partial	-	-	Ethereum-based	Public	Permissionless	Complete
(343)	Energy, E-commerce	-	Complete	Partial	Ethereum-based	Private	-	Complete
(156)	Distributed/Cloud computing	Partial	Complete	Complete	Hyperledger Project	Private	Permissioned	Complete
(344)	Distributed/Cloud computing	-	Partial	-	Ethereum-based	Public	Permissionless	Complete
(345)	Distributed/Cloud computing	-	Partial	-	Ethereum-based	Public	Permissionless	Complete
(346)	Distributed/Cloud computing	Partial	Complete	Complete	-	Public	-	Complete
(347)	Distributed/Cloud computing	Partial	-	Complete	Ethereum-based	Public	Permissionless	Complete
(348)	Distributed/Cloud computing	Partial	Complete	Partial	Ethereum-based	Private	Permissionless	Complete
(102)	IoT	Partial	Complete	Complete	Hyperledger Project	Private	Permissioned	Complete
(349)	IoT, Distributed/Cloud computing	-	-	-	Ad-hoc	Public	Permissionless	Complete
(157)	E-commerce, Distributed/Cloud computing	Complete	Complete	Complete	Ethereum-based	Public	Permissionless	Partial
(183)	Independent	-	Complete	Partial	Based on other technologies	Private	Permissioned	Partial
(350)	IoT	Partial	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(135)	Citizen services	Partial	Complete	Partial	Ethereum-based	Private	Permissionless	Partial
(351)	Independent	-	Complete	Complete	Any	Private	Permissioned	Complete
(352)	Independent	-	Complete	Complete	Hyperledger Project	Private	Permissioned	Partial
(353)	Independent	-	Partial	Complete	Ethereum-based	Public	Permissionless	Complete
(354)	IoT	-	Complete	Complete	Ethereum-based	Private	-	Complete

Table A.3: Analysis of academic papers (III), where - means not specified

Name	Area	Cybersecurity properties			Blockchain technology	Type of network		Justification
		Conf.	Authen.	Non- repudiation		Nature	Permissions	
(103)	Energy	-	Complete	Partial	Ethereum-based	-	Permissioned	Complete
(355)	IoT	-	Complete	Complete	Hyperledger Project	Private	Permissioned	Complete
(162)	Citizen services, IoT	Complete	Complete	Complete	Any	Public	-	Complete
(356)	Distributed/Cloud computing	Complete	-	-	-	-	-	Complete
(357)	IoT	Complete	Complete	Partial	Ethereum-based	Private	Permissioned	Partial
(167)	Health	Complete	Complete	Complete	Any	Private	Any	Complete
(358)	IoT	Partial	Complete	Partial	Ethereum-based	-	-	Partial
(359)	E-commerce	Complete	Complete	Partial	Ethereum-based	Any	Any	Partial
(187)	Independent	Complete	Complete	Complete	Hyperledger Project	Private	Permissioned	Partial
(360)	Independent	-	Complete	Complete	Hyperledger Project	Private	Permissioned	Partial
(170)	Independent	-	Complete	Partial	Based on other technologies	-	Permissioned	Partial
(178)	Distributed/Cloud computing, IoT, E-commerce	Partial	Complete	Partial	Ethereum-based	Public	Permissionless	Partial
(188)	E-commerce	Complete	Complete	Partial	Bitcoin-based	Public	Permissionless	No
(112)	Distributed/Cloud computing	Partial	Complete	Partial	Any	-	-	Partial
(361)	Independent	-	Complete	Partial	Ethereum-based	Public	Permissionless	Complete
(107)	Energy	-	Partial	Partial	Hyperledger Project	Private	Permissioned	Complete
(174)	IoT	Complete	Complete	Complete	Based on other technologies	Public	Permissionless	Complete
(362)	Independent	-	Complete	Partial	Hyperledger Project	Private	Permissioned	Complete
(363)	E-commerce	Complete	Complete	Partial	Ethereum-based	Public	Permissionless	Partial
(364)	Distributed/Cloud computing, E-commerce	Complete	Complete	Partial	Ethereum-based	Public	Permissioned	Partial
(104)	E-commerce	-	Complete	Partial	Bitcoin-based	Private	Any	Partial
(117)	IoT	Complete	Partial	Complete	Ethereum-based	Public	Permissionless	Complete
(120)	IoT, Distributed/Cloud computing	Partial	Complete	Complete	Hyperledger Project	Private	Permissioned	Partial
(179)	Health	-	Complete	Complete	Ethereum-based	Private	Permissionless	Complete
(164)	Citizen services	-	Complete	Partial	Ethereum-based	Public	Permissionless	Partial
(163)	Citizen services	-	Complete	-	Any	Private	Permissioned	Complete
(365)	IoT	Complete	Complete	Complete	Any	-	-	Complete
(151)	IoT	-	Complete	Complete	Ad-hoc	Private	Permissioned	Complete
(116)	E-commerce	Partial	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(366)	Health, IoT, Distributed/Cloud computing	Partial	Complete	Partial	-	Private	-	Complete
(4)	E-commerce, IoT	-	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(367)	E-commerce, IoT	-	Complete	Complete	Ethereum-based	Private	-	Partial
(368)	Distributed/Cloud computing	Partial	Complete	-	Bitcoin-based	Public	Permissionless	Complete
(369)	Distributed/Cloud computing	-	Complete	Partial	Bitcoin-based	Public	Permissionless	Complete
(180)	Health, Distributed/Cloud computing	Complete	Complete	Complete	Ad-hoc	Private	Permissioned	Partial
(144)	Energy, IoT	Complete	Complete	Partial	Ad-hoc	Private	Permissioned	Partial
(128)	E-commerce	Partial	-	-	Bitcoin-based	Public	Permissionless	Complete
(152)	IoT	-	Complete	Complete	Ethereum-based	Private	Permissioned	Complete
(370)	IoT, Citizen services, E-commerce	Complete	Complete	Complete	-	Private	Permissioned	Partial
(371)	IoT	Complete	Complete	Complete	Ad-hoc	Private	Permissioned	Partial
(372)	IoT	Complete	Complete	Partial	Bitcoin-based	Public	Permissionless	Complete
(373)	IoT	-	Complete	Complete	Bitcoin-based	Public	Permissionless	Complete
(374)	IoT	-	-	Complete	Ad-hoc	Public	Permissionless	Complete
(375)	Energy, IoT, E-commerce	-	-	-	Ethereum-based	-	Permissionless	Complete
(106)	IoT	Partial	Complete	Complete	Ad-hoc	-	Permissionless	Complete
(376)	IoT	Partial	Complete	Complete	Ad-hoc	Private	Permissioned	Complete
(377)	IoT	-	Complete	Complete	-	Private	Permissioned	Complete
(378)	Energy	Complete	Complete	Complete	Hyperledger Project	Private	Permissioned	Complete

Table A.4: Analysis of academic papers (IV), where - means not specified

Name	Area	Cybersecurity properties			Blockchain technology	Type of network		Justification
		Conf.	Authen.	Non- repudiation		Nature	Permissions	
(379)	Independent	Partial	-	Complete	Bitcoin-based	Public	Permissionless	No
(380)	Independent	-	Complete	-	Ethereum-based	Public	Permissionless	Complete
(381)	Health	Partial	Complete	Partial	Ad-hoc	Private	Permissioned	Complete
(115)	IoT	Partial	Complete	Partial	Ad-hoc	Private	Permissioned	Complete
(382)	Independent	Partial	Complete	Complete	Ad-hoc	Private	Permissioned	Complete
(383)	IoT	-	Complete	Complete	Ad-hoc	Private	Permissioned	Complete
(384)	Health	Partial	Complete	Complete	Ethereum-based	Private	Permissioned	Complete
(385)	IoT	-	Complete	Partial	Hyperledger Project	Public	Permissioned	Partial
(121)	Independent	-	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(386)	Citizen services	Partial	Complete	Complete	Ethereum-based	Private	Permissionless	Partial
(125)	IoT, Energy	-	Complete	Complete	Ethereum-based	Private	Permissioned	Complete
(387)	Energy	Partial	-	-	-	-	-	Complete
(388)	Health, IoT, Distributed/Cloud computing	-	Complete	Complete	Ethereum-based	Private	Permissionless	Complete
(389)	Health	Partial	Complete	Complete	Hyperledger Project	Private	Permissioned	Complete
(122)	Citizen services	-	Complete	Complete	Ad-hoc	Public	Permissioned	Complete
(390)	IoT	Partial	Complete	Complete	Hyperledger Project	Private	Permissioned	Pasrtial
(124)	IoT	-	Complete	Complete	Ethereum-based	Private	Permissioned	Partial
(391)	Independent	Complete	Partial	Complete	Ethereum-based	Private	-	Complete
(392)	E-commerce	Partial	Complete	Partial	Ad-hoc	Private	Permissioned	Complete
(393)	Health	-	Complete	Complete	Ad-hoc	Private	Permissioned	Partial
(394)	Independent	Complete	Complete	Complete	-	Private	Permissioned	Complete
(119)	Independent	Complete	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(395)	Distributed/Cloud computing	Complete	Complete	Partial	Ethereum-based	Public	Permissionless	Partial
(396)	E-commerce	-	Complete	Complete	Ethereum-based	Private	-	Partial
(397)	IoT, Distributed/Cloud computing	Partial	Complete	-	Ad-hoc	Private	Permissioned	Partial
(398)	IoT	-	Complete	Complete	Ad-hoc	Public	Permissioned	Partial
(399)	Distributed/Cloud computing, E-commerce	Complete	Complete	Complete	Ethereum-based	Public	Permissionless	Complete
(400)	Independent	Partial	Partial	Partial	Based on other technologies	Public	-	Complete
(401)	IoT	Complete	Complete	Partial	Ad-hoc	Private	Permissioned	Complete
(402)	E-commerce, Energy	Complete	Complete	Partial	Ad-hoc	Private	-	Partial
(147)	IoT, Distributed/Cloud computing	Complete	Complete	Partial	Hyperledger Project	Private	-	Partial
(403)	Independent	-	Partial	Partial	Ad-hoc	Private	Permissioned	Complete
(404)	IoT	Partial	Complete	Partial	Ad-hoc	Private	Permissioned	Complete
(142)	IoT, Citizen services	Complete	-	-	Ad-hoc	-	-	Complete
(405)	IoT	-	Complete	Partial	Hyperledger Project	Private	Permissioned	Partial
(406)	IoT	-	Complete	Partial	-	Public	Permissioned	Partial
(407)	Energy	-	Complete	Partial	Ethereum-based	Private	Permissioned	Complete
(408)	IoT	Partial	Complete	Complete	Ad-hoc	Public	Permissioned	Partial
(409)	IoT	Partial	Complete	Complete	Ad-hoc	Private	Permissioned	Partial
(410)	IoT	-	-	Partial	Ethereum-based	-	Permissionless	Complete
(411)	IoT	-	Complete	Partial	Ethereum-based	Private	Permissioned	Complete
(412)	IoT	Complete	Complete	-	Ad-hoc	Private	Permissioned	Partial
(413)	Energy, IoT	-	Complete	-	Based on other technologies	Public	Permissionless	Complete
(414)	Health, IoT	Complete	Partial	Partial	Ad-hoc	Public	Permissionless	Complete
(415)	IoT	-	Complete	Partial	Ad-hoc	Public	Permissionless	Complete
(416)	IoT	Complete	-	Complete	Ethereum-based	-	Permissionless	Complete
(417)	IoT	-	Complete	Complete	Ethereum-based	Public	Permissionless	Partial
(418)	IoT, Distributed/Cloud computing, E-commerce	Partial	-	-	Ad-hoc	-	-	Complete
(419)	Health	Partial	Complete	-	Ethereum-based	Public	Permissionless	Complete
(420)	Health	Complete	Complete	-	Ethereum-based	-	-	Complete

Table A.5: Analysis of academic papers (V), where - means not specified

Name	Area	Cybersecurity properties			Blockchain technology	Type of network		Justification
		Conf.	Authen.	Non- repudiation		Nature	Permissions	
(421)	IoT, Distributed/Cloud computing, E-commerce	-	Complete	Complete	-	Public, Private	-	Partial
(422)	Independent	Partial	Complete	Partial	Ethereum-based	Private	-	Partial
(423)	IoT	Partial	Complete	-	Bitcoin-based	Private	Permissionless	Partial
(123)	Energy	-	Complete	-	Hyperledger Project	Private	-	Complete
(424)	IoT	Complete	Complete	Complete	Ethereum-based	Private	Permissioned	Partial
(425)	Citizen services, IoT	-	Partial	Partial	Ethereum-based	Private	Permissionless	Complete
(426)	Energy, IoT	Partial	Complete	Complete	Ad-hoc	Private	Permissioned	Partial
(427)	Distributed/Cloud computing, IoT, E-commerce	-	-	Complete	Based on other technologies	Any	Any	Complete
(428)	Energy, IoT	Complete	Complete	Partial	Ad-hoc	Private	Permissionless	Partial
(429)	Health, IoT	Partial	Complete	Complete	Ethereum-based	-	-	Partial
(430)	Independent	-	Complete	Complete	Based on other technologies	Private	Permissioned	Partial
(146)	IoT, Distributed/Cloud computing	Complete	Complete	Partial	Ethereum-based	Private	-	Complete
(431)	IoT	Complete	-	-	Ad-hoc	-	-	Complete
(129)	IoT	-	-	-	Ethereum-based	Public	Permissionless	Complete
(432)	Citizen services, IoT	Complete	Complete	Complete	Based on other technologies	Private	Permissioned	Partial
(433)	Health, IoT	-	-	Complete	Ethereum-based	Private	Permissioned	Complete
(434)	IoT	-	-	Complete	-	Private, Public	Permissioned	Partial
(435)	Independent	Partial	-	Complete	Ad-hoc	Public	Permissionless	Complete
(436)	Health	Complete	Complete	Partial	Ethereum-based	Public	Permissionless	Complete
(437)	IoT	Complete	Complete	Complete	Ad-hoc	Private	Permissioned	Partial
(11)	IoT	-	Complete	-	Ad-hoc	Public	Permissionless	Partial
(438)	IoT, Distributed/Cloud computing	Complete	-	Complete	Ethereum-based	Public	Permissionless	Complete
(439)	E-commerce	-	-	Complete	Ethereum-based	Public	Permissionless	Complete
(440)	IoT	Complete	Complete	Complete	Ad-hoc	Private	Permissionless	Complete
(441)	E-commerce	Complete	Complete	Complete	Ad-hoc	Private	Permissioned	Partial
(442)	Independent	Partial	Complete	Partial	Ethereum-based	Private	-	Partial
(443)	E-commerce	-	-	Complete	Ethereum-based	Public	Permissionless	Complete
(444)	Distributed/Cloud computing, IoT, E-commerce	Complete	-	Partial	Ethereum-based	Public	Permissionless	Complete
(445)	Health	Complete	Partial	-	Ad-hoc	Private	-	Complete
(446)	Independent	-	Partial	Partial	Ad-hoc	-	Permissioned	Complete
(447)	Independent	Complete	Complete	-	Ethereum-based	Public	Permissionless	No
(448)	Energy	-	Complete	-	Ethereum-based	-	Permissioned	Complete
(449)	Health	-	Partial	Partial	-	-	Permissioned	No
(161)	E-commerce	Partial	Complete	Complete	Ethereum-based	Public & Private	-	Complete
(450)	IoT	Complete	Complete	Complete	Ethereum-based	Public	Permissioned	Partial
(451)	IoT	Complete	Partial	Complete	Ad-hoc	Private	Permissioned	Complete
(452)	E-commerce	Complete	-	Complete	Ethereum-based	Public	-	Complete
(453)	Cloud computing	Complete	Complete	-	-	Public	-	Complete
(454)	Health	Complete	Complete	Complete	Ad-hoc	Private	Permissioned	Complete
(455)	Independent	Complete	Complete	-	Hyperledger Project	Private	Permissioned	Partial
(456)	Health	Complete	Complete	-	Ethereum-based	Private	Permissionless	Complete
(457)	Health	Complete	Complete	-	Ad-hoc	Private	-	Complete
(458)	IoT	-	-	Complete	Ad-hoc	Private	Permissioned	Complete
(459)	E-commerce	-	-	Complete	Ethereum-based	Public	Permissionless	Complete
(460)	Independent	-	Complete	Complete	Hyperledger Project	Private	Permissioned	Complete

Table A.6: Analysis of academic papers (VI), where - means not specified

Name	Area	Cybersecurity properties			Blockchain technology	Type of network		Justification
		Conf.	Authen.	Non- repudiation		Nature	Permissions	
(461)	E-commerce	-	Partial	-	Ethereum-based	Public	Permissioned	Complete
(462)	Cloud computing	Complete	-	Complete	Ethereum-based		Permissioned	Complete
(463)	Energy, Citizen services	-	Complete	Complete	Ad-hoc	Private	Permissioned	Complete
(464)	Health	Complete	Complete	-	-	Private	Permissioned	Partial
(160)	E-commerce	-	-	Complete	Ethereum-based	-	-	Complete
(465)	IoT	Partial	Partial	Complete	Ad-hoc	-	-	Complete
(466)	E-commerce	-	Partial	-	Ethereum-based	-	-	Complete
(467)	Independent	Complete	-	-	Ethereum-based	Private	Permissioned	Partial
(468)	Health	Complete	Complete	-	Ethereum-based	Private	-	Complete
(469)	Cloud computing, IoT	Complete	Complete	Complete	-	-	-	Complete
(470)	IoT	-	-	-	Hyperledger Project		Permissioned	Complete
(471)	Health	Complete	Complete	-	Ethereum-based, Hyperledger project	Private	-	Complete
(472)	IoT	-	Partial	-	-	Private	-	No
(473)	IoT	Complete	Complete	-	Ad-hoc	Private	Permissioned	Complete
(474)	Independent	Complete	Complete	Complete	-	-	-	No
(475)	IoT	Partial	-	-	Ethereum-based	Private	Permissioned	Complete
(476)	Energy	-	-	Complete	Ethereum-based	Public	Permissioned	Complete
(477)	Health	Complete	-	Complete	Hyperledger Project	Private	Permissioned	Complete
(478)	IoT	Partial	-	-	Ethereum-based	-	-	Complete
(479)	IoT	Partial	Complete	Complete	Ethereum-based	Private	Permissioned	Complete
(480)	IoT	-	-	Complete	Ethereum-based	Public	Permissionless	Complete
(481)	Energy	-	Partial	-	Ad-hoc	-	-	Complete
(482)	IoT	-	Complete	Complete	-	Public and Private	-	No
(483)	IoT	-	-	Complete	Ad-hoc	Private	Permissioned	Complete
(484)	IoT	Complete	Complete	-	Ethereum-based	Public	Permissioned	Complete
(485)	IoT	-	Complete	-	-	Private	-	No
(486)	IoT	Complete	-	Complete	Ethereum-based	Public	Permissionless	Complete
(487)	Health	Complete	Complete	Complete	-	Public	Permissioned	Complete
(488)	Independent	-	-	Complete	Ad-hoc	-	Permissioned	Complete
(489)	IoT	Complete	Complete	-	Ad-hoc	Private	Permissioned	Complete
(490)	Independent	-	Complete	Complete	Hyperledger project	Private	-	Complete
(491)	E-commerce	-	-	-	Ad-hoc	-	Permissioned	Complete
(492)	IoT	-	Partial	-	Ad-hoc	Public and Private	-	Complete
(493)	Health	Complete	Complete	Complete	Ad-hoc	Private	Permissioned , Permissionless	Complete
(494)	Independent	Partial	Complete	-	Ethereum-based	Private	Permissioned	Complete
(495)	IoT	Complete	Complete	-	-	Public	Permissioned	Complete

The analysis of all the academic papers analyzed in the Section 3.3 of Chapter 3 is depicted in Tables A.7, A.8,A.9 and A.10.

Table A.7: Studied sample (I)

					Data exchange		Transaction flux			Data protection		
Source	Name	Type	Cybersecurity properties	Elements used	Blockchain technology	Seed address	Receive transactions	Send transactions	Purpose	Confidentiality	Covert channel	Cost to attacker
(202)	Custom	Any	Immutability, Availability, Decentralization	OP_RETURN	Bitcoin	Hardcoded	-	From 1 to 1	Malware storage	No	No	Yes
(26)	Petya	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	-	No
(205)	NotPetya	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	-	No
(25)	Cerber	Ransomware	Availability, Anonymity, Decentralization	Transaction (Receiver address)	Bitcoin	Hardcoded	Various to 1	From one to various	Payments and domain generator	No	No	Yes
(203)	Custom	Ransomware	Immutability, Availability, Decentralization	Program code (Function arguments)	Ethereum	Not known	Various to 1	From 1 to 1	Key distribution, payments	No	Yes	Yes
(212)	GrandCrab	Ransomware	Anonymity, Decentralization	Transaction	Dash	Webpage	Various to 1	-	Payments	No	No	No
(213)	TeslaCrypt	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(257; 496)	Dharma	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Not known	Various to not known	-	Payments	No	No	No
(496)	Cryptowall	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(496)	CryptoLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Locky	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(257; 496; 497)	Spora	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to 1	-	Payments	No	No	No
(257; 257; 496)	CryptoDefense	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(257; 496; 497)	CoinVault	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 498; 499)	Jigsaw	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(496; 500?)	Ryuk	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(204)	CTBLocker (webpage version)	Immutability, Availability, Ransomware	Anonymity, Decentralization	OP_RETURN	Bitcoin	Webpage	Various to various	From various to various	Payments, key distribution	No	No	No
(501)	TorrentLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(496)	CryptXXX	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to one	-	Payments	No	No	No
(42)	BlockchainBot	Botnet	Anonymity, Decentralization	Services on top of blockchain (MAM message)	IOTA	Harcoded	Various to various	Various to various	C&C	Yes	No	Yes
(220)	DMALocker3	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(226)	CryptoTorLocker2015	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to one	-	Payments	No	No	No
(216; 226)	Globe	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Mail	Various to various	-	Payments	No	No	No
(496; 502)	GlobeImposter	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to one	-	Payments	No	No	No
(502?)	SamSam	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 496)	NeoLCrypt	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 503)	EDA2	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(220; 504)	Flyper	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 220)	XLocker5.0	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Not known	Various to various	-	Payments	No	No	No
(220; 496)	Globev3	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 496)	CryptoHitman	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to one	-	Payments	No	No	No
(220; 505)	TowerWeb	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to one	-	Payments	No	No	No
(220; 496)	sVr7n	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 506)	Buzbi	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Buddy	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Chimera	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 507)	CryptoHost	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No

Table A.8: Studied sample (II)

					Data exchange					Data protection		
						Transaction flux						
Source	Name	Type	Cybersecurity properties	Elements used	Blockchain technology	Seed address	Receive transactions	Send transactions	Purpose	Confidentiality	Covert channel	Cost to attacker
(220; 496)	ThunderCrypt	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 496)	Trump Locker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 508)	Doxware	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Badblock	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 509)	AdamLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 510)	Alphabet	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	DMALocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	AngleWare	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	APT	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 496)	BadEncript	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Black Feather	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	BTCWare	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Mail	Various to 1	-	Payments	No	No	No
(220; 496)	Comrade Circle	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 511)	CryptConsole	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 496)	Crypto-SweetTooth	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Cyber Splitter	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Domino	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 512)	Exotic	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 513)	FakeGlobe	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Fantom	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Mail	Various to not known	-	Payments	No	No	No
(220; 514)	Fantom (variant)	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	FireCrypt	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(515)	Globe2	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Mail	Various to various	-	Payments	No	No	No
(220; 516)	Ransomeer	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 517)	Ime_explØit	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 518)	Nemucod	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 496)	Nullbyte	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	PayDay	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 519)	Philadelphia	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 496)	Phoenix	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No

Table A.9: Studied sample (III)

						Data exchange				Data protection		
						Transaction flux						
Source	Name	Type	Cybersecurity properties	Elements used	Blockchain technology	Seed address	Receive transactions	Send transactions	Purpose	Confidentiality	Covert channel	Cost to attacker
(220; 520)	PopCornTime	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Raudon6	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Mail	Various to not known	-	Payments	No	No	No
(220; 496)	RansomPlus	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Razy	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	REKTLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Xorist	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Mail	Various to various	-	Payments	No	No	No
(220; 521)	Rush/Sanction	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 522)	Stupid	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	VenusLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(220; 496)	XLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 523)	XTPLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(220; 496)	Zyka	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(229)	Custom	Ransomware	Immutability, Availability, Anonymity, Decentralization	Program code (Function arguments)	Ethereum	Webpage	-	From various to 1	Payments, C & C, Key distributions	No	No	Yes
(256; 496)	KeRanger	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(256; 496)	Mischa	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to 1	-	Payments	No	No	No
(256; 524)	ZCryptor	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to 1	-	Payments	No	No	No
(256; 525)	KillDisk(linux)	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(256; 526)	FindZip	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(256; 527)	The LLTP Locker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to 1	-	Payments	No	No	No
(256; 496)	GoldenEye	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(256; 528)	DoubleLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(496; 500)	DarkSide	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin/Monero	Webpage	Various to various	-	Payments	No	No	No
(496; 500)	NetWalker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(496; 500)	MedusaLocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Mail	Various to various	-	Payments	No	No	No
(496; 500)	Conti	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Mail	Various to various	-	Payments	No	No	No
(500; 528)	Qlocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(? ?)	LockBit 2.0	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(496; 500)	Egregor	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(496; 500)	Black Kingdom	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(500; 529)	REvil	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(500; 528)	Qlocker	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Webpage	Various to various	-	Payments	No	No	No
(496; 500)	AvosLocker	Ransomware	Anonymity, Decentralization	Transaction	Monero	Webpage	Various to various	-	Payments	No	No	No
(500; 530)	HC6/HC7	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No

Table A.10: Studied sample (IV)

Source	Name	Type	Cybersecurity properties	Data exchange: Elements used	Blockchain technology	Transaction flux: Seed address	Transaction flux: Receive transactions	Transaction flux: Send transactions	Purpose	Data protection: Confidentiality	Data protection: Covert channel	Cost to attacker
(531)	Wannacry	Ransomware	Anonymity, Decentralization	Transaction	Bitcoin	Hardcoded	Various to various	-	Payments	No	No	No
(206)	Custom	Ransomware	Anonymity, Decentralization	Transaction	Monero	Not known	Various to not known	-	Payments	No	No	No
(210)	Custom	Botnet	Availability, Decentralization	OP_RETURN	Bitcoin	None	-	From 1 to 1	Address	No	No	Yes
(241)	Custom	Botnet	Availability, Decentralization	OP_RETURN	Bitcoin	Hardcoded	Various to 1	From various to various	C	C	Yes	No
Yes												
(221)	Unblockable Chains	Botnet	Availability, Anonymity Decentralization	Program code (Function arguments)	Ethereum	Hardcoded	Various to 1	From 1 to 1	C&C	Obscured	No	Yes
(211)	ChainChannels	Botnet	Availability, Anonymity Decentralization	Nonce of signature	Any	Hardcoded	-	Various to various	C&C	Obscured	Yes	Yes
(232)	Coinbot	Botnet	Availability, Anonymity Decentralization	Data/OP_RETURN	Bitcoin, Ethereum,Dash,Litecoin,BitcoinCash	Website	-	From 1 to 1	C&C Encrypted	No	Yes	
(238)	Gluputeba	Botnet	Availability, Anonymity, Decentralization	OP_RETURN	Bitcoin	Hardcoded	-	From various to various	Address	Encrypted	No	Yes
(233)	Custom	Botnet	Availability, Decentralization	Program code (Function arguments)	Ethereum	Hardcoded	-	From 1 to 1	C&C	No	No	Yes
(239)	Custom	Botnet, Worm	Availability, Anonymity, Decentralization	Transaction (Receiver address)	Bitcoin	Hardcoded, Blockchain	-	From various to various	C&C	No	Yes	Yes
(222)	Neuronesh	Botnet	Availability, Anonymity, Decentralization	OP_RETURN	Bitcoin	Hardcoded	-	From various to various	C&C	No	No	Yes
(240)	Botract	Botnet	Availability, Anonymity, Decentralization	Program code (Function arguments)	Ethereum	Hardcoded	-	From 1 to 1	C&C	No	No	Yes
(219)	Custom	Botnet	Availability, Anonymity, Decentralization	OP_RETURN	Bitcoin	C&C	-	From various to various	Address	Encrypted	No	Yes
(28)	Fbot	Botnet	Availability, Anonymity, Decentralization	Services on top of blockchain(ErmerDNS)	Enercoin	Hardcoded	-	From 1 to various	DNS	No	No	No
(27)	Pony	Botnet	Availability, Anonymity, Decentralization	Transaction(value)	Bitcoin	Hardcoded	-	From 1 to 1	Address	No	No	Yes
(234)	Custom	Botnet	Immutability, Availability, Anonymity, Decentralization	Program code (Function arguments)	Ethereum	Hardcoded	-	From 1 to 1	C&C	No	No	Yes
(231)	Custom	Botnet	Availability, Anonymity, Decentralization	Services on top of blockchain(Whisper)	Ethereum	Hardcoded	From various to 1	From 1 to various	C&C	Encrypted	No	No
(223)	LNBot	Botnet	Availability, Anonymity, Decentralization	Transaction(value)	Bitcoin	Hardcoded	From 1 to 1	From 1 to various	C&C	No	Yes	Yes
(230)	Zombiecoin	Botnet	Availability, Anonymity, Decentralization	OP_RETURN & signature	Bitcoin	Hardcoded	-	From 1 to various	C&C	No	Yes	Yes
(207)	Custom	Botnet	Availability, Anonymity, Decentralization	Payment id	Monero	Hardcoded	From various to 1	From various to various	C&C	Encrypted	No	Yes
(237)	D-LNBot	Botnet	Availability, Anonymity, Decentralization	Services on top of blockchain(onion payload)	Bitcoin	Hardcoded	From 1 to 1	From 1 to various	C&C	No	No	No
(225)	BlockRAT	Botnet	Availability, Anonymity, Decentralization	Services on top of blockchain(DApp)	NKN	Hardcoded	Not specified	Not specified	C&C	Yes	No	No
(242)	Dustbot	Botnet	Availability, Anonymity, Decentralization	OP_RETURN	Bitcoin	Hardcoded	From various to 1	From 1 to 1	C&C	No	No	Yes
(243)	Botchain	Botnet	Availability, Anonymity, Decentralization	OP_RETURN	Bitcoin	Webpage	-	From 1 to various	C&C	No	No	Yes
(244)	Custom	Botnet	Availability, Anonymity, Decentralization	OP_RETURN	Bitcoin	Hardcoded	From various to various	From 1 to various	C&C	Encrypted	No	No

A.2 Tables for Contribution 2

The 20 most common opcodes and their frequency is depicted in Table A.11.

Table A.11: 20 most common opcodes in smart contracts.

Opcode	Frequency(%)
POP	23.38%
PUSH1	16.49%
SWAP1	7.10%
AND	6.13%
DUP2	4.48%
SWAP2	4.32%
SWAP3	3.86%
DUP1	3.69%
SLOAD	3.66%
EXP	2.90%
PUSH20	2.89%
SUB	2.71%
MSTORE	1.86%
ADD	1.49%
MLOAD	1.49%
DUP4	1.18%
SSTORE	1.15%
SWAP4	0.99%
NOT	0.93%
OR	0.90%

The top 5 number and quantity of instruction in the JUMP-JUMPDEST block is presented in Table A.12.

Table A.12: Top 5 number of instructions in the JUMP-JUMPDEST block

# Instructions	Contracts	Frequency(%)
2	12234	20.86
9	7830	13.35
0	6863	11.70
1	5249	8.95
13	3974	6.77

The 2 most used values per field and their percentage in order to estimate their variability is presented on Table A.13.

Table A.13: Most used values and percentages

Field	Subfield	Most used value	Most used value (quantity)	Second most used value	Second most used value (quantity)	Total sample	Percentage most used value	Percentage second most used value
Receiver address		3f5ce5fbfe3e9af3971dd833426ba9b5c936f0be	209201	60d0cc2ae15859f69bf74dadb8ae3bd584434976b	120230	8998787	2.32%	1.34%
Swarm hash		fe8c241acfb8e01e0521fa3f76b871952cdb7056636e58db8766ec68c79742e93	5653	05c5851706b1ba497a4e0bc053a662a46f16f0200c85c37e4f8319d8ca9e93ab	4973	65346	8.65%	7.61%
Bytecode	opcodes	POP	131842	PUSH1	92994	563870	23.38%	16.49%
	push1 value	00	26170	01	19245	92994	28.14%	20.69%
	push20 value	ffffffffffffffffffffffffffffffffffffffff	16239	39aa39c021dfbae8fac545936693ac917d5e7563	11	16275	99.78%	0.07%
Value	toperson	10000000000000000	90384	10000000000000000	85017	8998787	1.00%	0.94%
	to function	0	7408089	8000000000000000	61157	7943428	93.26%	0.77%
	contract	0	65183	10000000000000000	20	65346	99.75%	0.03%
Gas Price	to usrr	1000000000	1052278	2000000000	871559	8998787	11.69%	9.69%
	to function	20000000000	508729	4000000000	493642	7943428	6.40%	6.21%
	contract	21000000000	6505	20000000000	4634	65346	9.95%	7.09%
Gas	to usrr	21000	3766853	50000	1823388	8998787	41.86%	20.26%
	to function	60000	823289	90000	728033	7943428	10.36%	9.17%
	contract	280000	5653	500000	5551	65346	8.65%	8.49%
Function arguments	uint256	0	427289	10000	2518559	11756671	3.63%	2.14%
	address	0000000000000000000000000000000000000000	632408	3f5ce5fbfe3e9af3971dd833426ba9b5c936f0be	101286	9746801	6.49%	1.04%
	bytes32	0000000000000000000000000000000000000000000000000000000000000000	10848	22ddafaf5214123b39e339383710020e31620857ea0b52706010313b775e0ec6bad	2044	2135859	0.51%	0.10%
Constructor arguments	uint256	0	205	1000000000	113	2338	8.77%	4.83%
	address	aa1ae5e57dc059814d83ec7fca0b3c7ee25651b7d6	2140	aa829ef1cf818b4d4d0efabad33115fe85859fe3c49	1196	7604	28.14%	15.73%
Sender address	to usrr	ea674fdde7144fd979de3edf0f56aa97116b898ec8	845986	52bc44d5378309ee2abf1539bf71de1b7d7be3b5	474980	8998787	9.40%	5.28%
	to function	fbb1b73c4f0bda4f67dca266ce6ef42f520fbb98	202552	0xa7a7899d9444fe658c4b0a1803bab2f490bd3849e	176294	7943428	2.55%	2.22%
	contract	0x9862d074e33003726fa05c74f0142995833a3250	5653	0x0031e147a79e45f24319dc02ca860cb6142fcba1	4973	65346	8.65%	7.61%
Contract address	contract	All appear only once	1	All appear only once	1	65346	0.00%	0.00%
r	to usrr	All appear only once	1	All appear only once	1	8998787	0.00%	0.00%
	to function	All appear only once	1	All appear only once	1	7943428	0.00%	0.00%
	contract	79b667ef9dcbbac55a06295ce870b07029bfcdb2dec28d959f92815b16f81798	3	The rest appear only once	1	65346	0.00%	0.00%
s	to usrr	All appear only once	1	All appear only once	1	8998787	0.00%	0.00%
	to function	All appear only once	1	All appear only once	1	7943428	0.00%	0.00%
	contract	0aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	3	The rest appear only once	1	65346	0.00%	0.00%
v	to usrr		38		37	3919194	43.57%	43.55%
	to function		37		38	2860293	36.04%	36.01%
	contract		38		37	25913	39.77%	39.66%
Transaction id	to usrr	All appear only once	1	All appear only once	1	8998787	0.00%	0.00%
	to function	All appear only once	1	All appear only once	1	7943428	0.00%	0.00%
	contract	All appear only once	1	All appear only once	1	65346	0.00%	0.00%