Architecture for privacy-preserving brokerage of analytics using Multi Party Computation, Self Sovereign Identity and Blockchain

Master of Science in Technology Thesis
University of Turku
Department of Computing
Security of Networked Systems
2022
Donato Pellegrino

Supervisors:
Jarno Marttila (TX - Tomorrow Explored)
Seppo Virtanen (University of Turku)
Petri Sainio (University of Turku)

UNIVERSITY OF TURKU
Department of Computing

Donato Pellegrino: Architecture for privacy-preserving brokerage of analytics using Multi Party Computation, Self Sovereign Identity and Blockchain

Master of Science in Technology Thesis, 52 p.
Security of Networked Systems
June 2022

In our increasingly digitized world, the value of data is clear and proved, and many solutions and businesses have been developed to harness it. In particular, personal data (such as health-related data) is highly valuable, but it is also sensitive and could harm the owners if misused.

In this context, data marketplaces could enhance the circulation of data and enable new businesses and solutions. However, in the case of personal data, marketplaces would necessarily have to comply with existing regulations, and they would also need to make users privacy protection a priority. In particular, privacy protection has been only partially accomplished by existing datamarkets, as they themselves can gather information about the individuals connected with the datasets they handle.

In this thesis is presented an architecture proposal for KRAKEN, a new datamarket that provides privacy guarantees at every step in the data exchange and analytics pipeline. This is accomplished through the use of multi-party computation, blockchain and self-sovereign identity technologies. In addition to that, the thesis presents also a privacy analysis of the entire system.

The analysis indicated that KRAKEN is safe from possible data disclosures to the buyers. On the other hand, some potential threats regarding the disclosure of data to the datamarket itself were identified, although posing a low-priority risk, given their rare chance of occurrence. Moreover the author of this thesis elaborated remarks on the decentralisation of the architecture and possible improvements to increase the security. These improvements are accompanied by the solutions identified in the paper that proposes the adoption of a trust measure for the MPC nodes.

The work on the paper and the thesis contributed to the personal growth of the author, specifically improving his knowledge of cryptography by learning new schemes such as group signatures, zero knowledge proof of knowledge and multi-party computation. He improved his skills in writing academic papers and in working in a team of researchers leading a research area.

Keywords: Data market, Multi-party computation, Privacy analysis

Contents

1 Introduction

The volume of digital data collected worldwide offers unprecedented business opportunities at every level of the data science pipeline: from data collection, to preprocessing, until analysis and interpretation. There is in particular an increasing attention for personal data, due to the widespread public adoption of smartphones and fitness trackers [1] [2] [3].

New health-related data sources are not limited to fitness trackers, but can also be simple smartphones, whose role has been central for the tracking of COVID-19 cases [4] [5]. Furthermore, advances in machine learning have made possible to infer medically relevant informations in novel ways. For example, with the possibility of predicting solid tumors just from blood samples [6] and with the non-invasive measurement of blood sugar levels, useful for type 2 diabetes patients [7].

Not only can these devices benefit individual users (by providing a history of health-related recordings), but, when combined together, the data collected from many of them acquires an entirely new value in itself, this time in the context of clinical research [8]. However, these opportunities do not come without potential risks, especially when dealing with highly sensitive datasets, such as medical ones [9]. In these cases, there is a strong demand for solutions that fully preserve people's privacy while also not sacrificing any potential insight that could be gained from the data collected.

In this direction, solutions have started to emerge in the form of data marketplaces. These are online platforms that allow users to monetise the sharing of datasets of interest, which are typically stored in the cloud. With data marketplaces, the data collected by entities can be exploited by others instead of being a resource contributing only to the entities themselves, thus generating value in the world and creating a new source of revenue. Being privacy a strong concern for medical-related data, a diverse range of cryptographic protocols is being used by the medical data marketplaces created so far.

To name some of these datamarkets, we have Medicalchain [10], MyHealthMyData [11], Enveil [12], Wibson [13], and Agora [14]. However, all of these present some tradeoffs, either on the efficiency side (e.g. by requiring customized encryption from the marketplace for every single dataset that is processed) or regarding privacy (e.g. letting the marketplace access the analytics performed on the data).

For example, by leveraging functional encryption, Agora allows data consumers to carry out calculations on users data without ever getting access to the data itself. However, the security with regard to the privacy of users against the marketplace itself is not taken into account.

KRAKEN is a project funded by the European Union that aims to “enable the sharing, brokerage, and trading of potentially sensitive personal data, by returning the control of this data to citizens (data providers) throughout the entire data lifecycle” [15]. The project includes research and development in every aspect needed for its realisation such as regulations, user experience, system architecture design and software development.

This thesis presents a privacy preserving architecture proposal for the system architecture design research area of KRAKEN. The main contributions by the author of this thesis have been published in the paper: K. Koch, S. Krenn, D. Pellegrino, and S. Ramacher, “Privacy-preserving analytics for data markets using MPC”, in "Privacy and Identity Management", M. Friedewald, S. Schiffner, and S. Krenn, Eds., Cham: Springer International Publishing, 2021, pp. 226–246. Specifically, the contribution of the author influenced every sections of the paper [16] and his main role consisted in leading the threat modeling and privacy analysis of KRAKEN, i.e. in systematically assessing the security risks affecting the platform.

Differently from Agora and other earlier marketplaces, the architecture proposed in this thesis is able to carry out all requested analysis without invasively acquiring knowledge about users data. The development of this architecture is centered on three pivotal technologies: multi-party computation (MPC), blockchain and self-sovereign identity. Multi-party computation is the technology that allows the marketplace to handle users' data and perform analytics on them without having access to the data itself. The blockchain is the component that, together with MPC, provides decentralisation to the system which is exploited for the decision making. Self sovereign identity contributes to the privacy of the users, allowing them to demonstrate their eligibility to purchase access to sensitive datasets without relying on centralised parties and preserving their privacy by providing the minimum necessary information.

The architecture is followed by a privacy analysis, based on the LINDDUN [17] methodology, used to evaluate the privacy risks linked to the architecture. "LINDDUN is a privacy threat modeling methodology that supports analysts in systematically eliciting and mitigating privacy threats in software architectures" [18]. For this reason it constitutes a suitable candidate to conduct the privacy analysis as the final purpose of KRAKEN is to preserve privacy.

The rest of the thesis is organized as follows. In chapter 2, the background literature upon which KRAKEN relies will be discussed. Starting from the basic cryptographic components followed by the concepts of blockchain, self-sovereign identity and finally of threat modeling technologies, particularly LINDDUN. In chapter 3, the motivation and the objectives of KRAKEN will be outlined. In chapter 4, the architecture of the KRAKEN datamarket will be described in-depth, going through each component and stakeholder. In chapter 5, the LINDDUN privacy analysis of KRAKEN will be carried out. Following the construction of the threat tables, the elicitation, prioritization and finally mitigation of threats will be conducted. In chapter 6 the conclusions will be laid out, summarizing the work done and highlighting possible future directions.

2 Literature review

In this chapter will be described the literature behind the core components of the KRAKEN data marketplace architecture [16]. Starting from the basic cryptographic priors, self-sovereign identity will be discussed, followed by multi-party computation (perhaps KRAKEN's main advantage compared to other similar marketplaces) and finally by LINDDUN privacy analysis.

2.1 Cryptographic building blocks of KRAKEN

The privacy features offered in the KRAKEN architecture depend heavily on cryptography. The main innovative cryptographic mechanisms are three: Multi Party Computation (MPC), Group signatures, and Zero-knowledge proof of knowledge, which are described in the following sections.

2.1.1 Group signatures

Using group signatures [19], a member of a group can demonstrate to someone that a data exchange is happening with a member of the group itself without revealing the identity of the member. The receiver of a message can use the group's public key to verify that the member that sent the message does in fact belong to the group, but any other information about the member is protected.

Depending on their functionality, many types of group signature schemes have been implemented [20]. In cases where a large number of signatures has to be verified in a short time (e.g. for Vehicle-to-Vehicle communication), a group signature scheme has been proposed by Kim et al [21].

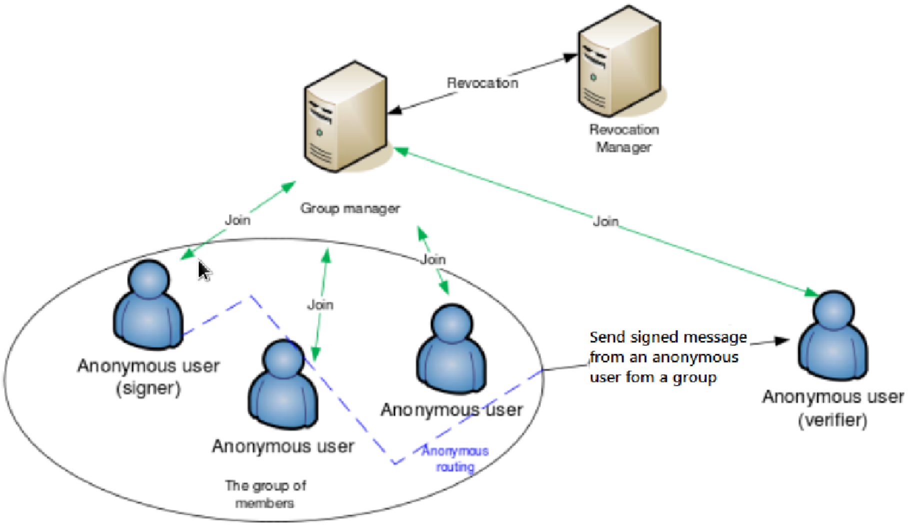


Figure 2.1: Principles of Group Signatures Scheme [22].

In a typical group signature setting (such as the one in [19]), there are three parties involved: the group members, the group manager and an external verifier. The group manager is in charge of the creation of the group (and of its parameters, e.g. public key), and subsequently for the admission and removal of users from the group itself (this is done through the manager's master key).

Group signatures find use in all those settings where verifiers find acceptable to only get confirmation about a signer's group identity (with the underlying assumption that, in case of necessity, the group manager could still get to know the signer's individual identity). For example, in *conceal organizational structures* [23], e.g. a company that allows its employees to carry out procedures on behalf of itself, knowing that, should an employee misuse this capability, a manager could immediately find out the personal identity of the culprit. Other applications [24] include for example electronic voting schemes [25] and auction protocols [26].

In other situations, however, the presence of a centralized manager can be unnecessary and moreover undesirable from a privacy perspective. For this reason, several works have been done to remove the need for a centralized manager [24].

This is also the case for KRAKEN, in which manager examinations are not needed, and thus public keys are constructed to prevent any party from knowing their complementary secret keys. This can be accomplished, for example, by setting the public key of the revocation manager to the hash value of a random string. A similar approach has also been taken in Intel's Enhanced Privacy ID (EPID) system [27].

2.1.2 Zero-knowledge proofs of knowledge

In our daily life experience, we are used to the fact that in order to prove the truthfulness of a particular claim, some information must be revealed about the claim itself. In 1982, Goldwasser, Micali and Rackoff [28] showed that this is indeed not necessary, by introducing zero-knowledge proofs of knowledge (ZK-PoK). In these schemes, a prover is able to ascertain to a verifier about possessing a certain information, wthout allowing the verifier to learn anything about the information itself. There are many potential applications for such proofs, going from authentication systems [29] [30], to voting schemes [31], e-cash, up to nuclear disarmament [32].

To achieve its goal, a ZK-PoK requires interaction between the verifier and the prover, with the former challenging the latter until the claim of possessing the information has been indirectly verified. One can explain an interactive ZK-PoK with a typical example, in which there are Alice and Bob (two friends), and Bob is color-blind, but Alice is not. Alice has two billiard balls, one red and one green, indistinguishable from each other apart from their color. Being color-blind, Bob (in this situation, the verifier) is skeptical that the two balls are in fact distinguishable, and Alice (the prover) wants to demonstrate him that they indeed are. The proof is simple: Alice puts the green ball in one of Bob's hands, and the red ball in his other hand. Bob then puts his hands behinds his back, and he can decide if switching the red ball with the green one, or leaving them both in their original place. He then brings his hands out, and Alice can say whether he switched the balls or not. If Alice would be randomly guessing, she would be correct with probability of 0.5. Therefore, Bob can simply repeat the experiment n times, and if Alice is always correct, the probability that she would be so just by guessing would be 2^{-n}. After some iterations (say, 100), with that probability getting very close to zero, Bob will convince himself of the difference in the two balls color, despite never gaining any information on how to distinguish the color himself.

Several zero-knowledge proofs that do not require verifier-prover interactions have also been proposed, such as non-interactive zero-knowledge proofs (NIZK), zero-knowledge scalable transparent argument of knowledge (zk-STARK) and zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARK) [33].In particular, zk-SNARKs are rapidly becoming popular due to their remarkable computational performance.

The zk-SNARKs are "succinct" due to their briefness (they are small and are tipically verifiable in a matter of milliseconds), and they are "non-interactive" because they necessitate the prover to send just one message to the verifier [34]. To achieve this, a preliminary phase is required, in which a shared reference string is generated according to pre-determined rules. This is a particularly delicate phase, and there is ongoing work on how to make it secure without resorting to at least partly centralized strategies. The zk-SNARK protocol is central in Zcash [35], a cryptocurrency focusing on a strengthened privacy compared to Bitcoin or other alternatives.

2.1.3 Secure Multi Party Computation

Let's consider the famous *Millionaire's Problem*, in which two millionaires want to know, without revealing their net-worth, who is the richest among them. One can reframe this as having two numbers, x and y, and wanting to know whether $x > y$ without disclosing the values of x and y themselves.

The problem was presented and solved (albeit with exponential cost) by Andrew Yao in 1982 [36], and it is the first example of secure multi-party computation (MPC). In MPC, several parties contribute to the computation of some predetermined function (in this case, a simple inequality) with the restriction that each node cannot acquire any knowledge aside from its own input and from the final output of the function itself. This makes MPC an attractive choice for cases where data privacy is important.

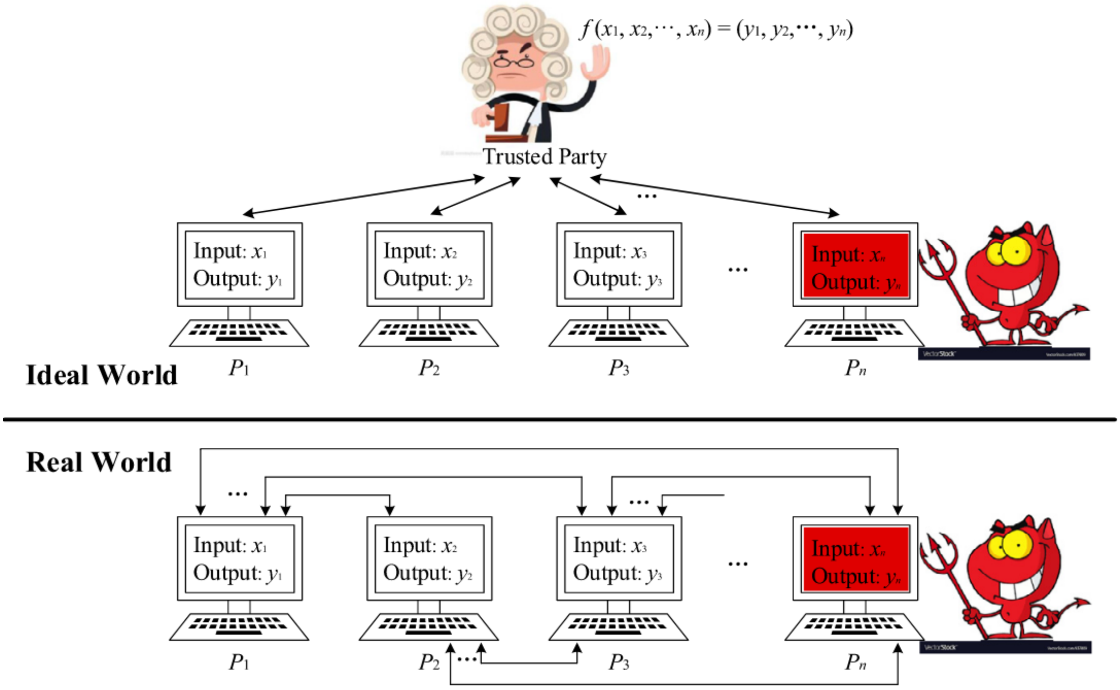

$f(x_1, x_2, \cdots, x_n) = (y_1, y_2, \cdots, y_n)$
Trusted Party
Input: x_1 Output: y_1
Input: x_2 Output: y_2
Input: x_3 Output: y_3
Input: x_n Output: y_n
Ideal World
P_1
P_2
P_3
P_n
Real World
Input: x_1 Output: y_1
Input: x_2 Output: y_2
Input: x_3 Output: y_3
Input: x_n Output: y_n
P_1
P_2
P_3
P_n

Figure 2.2: Comparison of ideal and real simulation settings [37].

But what if some of the parties are malicious? To measure the security of MPC, one can refer to the work of Canetti [38], where the ideal/real paradigm is defined: this is still today the standard way to define the security of MPC. In practical terms, one compares two scenarios: an "ideal" one, where the parties involved in the MPC send their input to an independent, incorruptible party, and get back an output, and a "real" one, where no incorruptible independent party exists, and so the parties need to jointly follow some protocol. If the most harm that an eventual malicious agent can do to the system in the real scenario is the same that could be done in the ideal one, the designed protocol is said to be secure. Several technologies have been developed to implement MPC, such as oblivious transfer, garbled circuits, and secret-sharing mechanisms (based on Shamir's algorithm, see Figure 2.3).

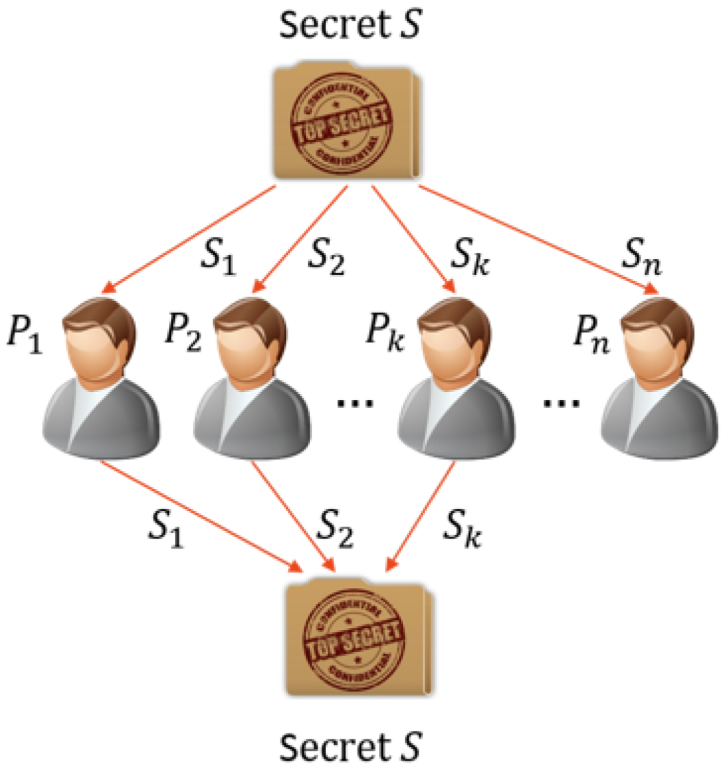
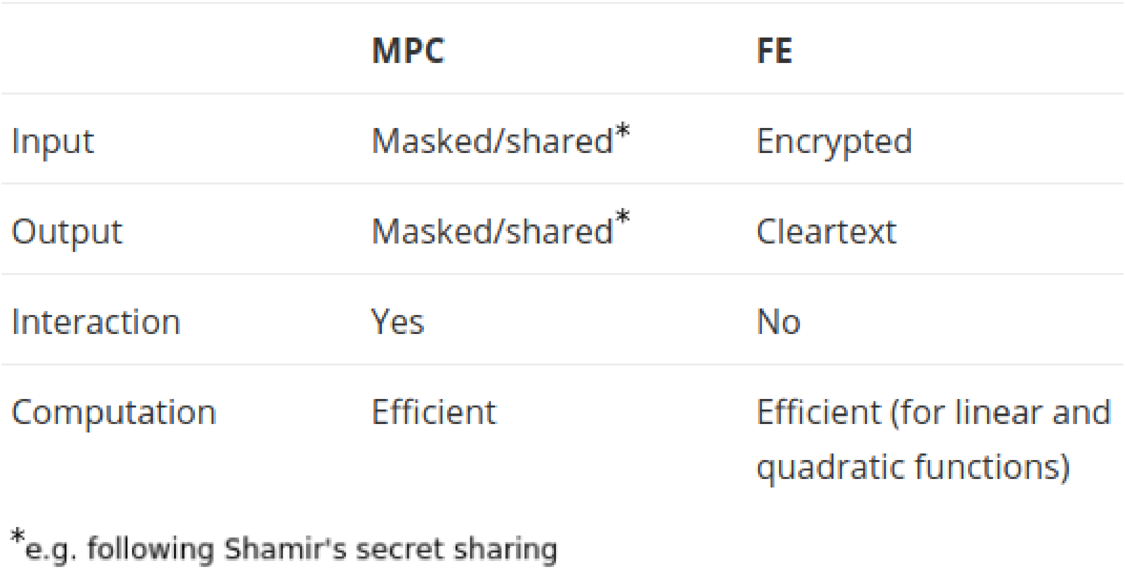


Figure 2.3: A sketch of Shamir's algorithm [39]. The secret S is splitted in n parts (one for each participant), and a minimum of k parts is required to reconstruct it (with $k < n$).

2.1.3.1 Comparison with Functional Encryption

Functional encryption (FE) is a scheme similar to a public-key encryption scheme, in which a user having a decryption key can learn *only a specific function* of the encrypted data [40]. No other information about the data is leaked in the process.

In the context of the KRAKEN marketplace, this would make FE a good fit for allowing the users of the platform to carry out privacy-preserving analysis. However, a limitation of FE schemes is that they require the broker to encrypt every data batch according to the specific function that will be computed on it. This factor implies that the marketplace can have access to the content of the datasets provided by the data providers, and it is the reason why MPC was chosen.



	MPC	**FE**
Input	Masked/shared*	Encrypted
Output	Masked/shared*	Cleartext
Interaction	Yes	No
Computation	Efficient	Efficient (for linear and quadratic functions)

*e.g. following Shamir's secret sharing

Table 2.1: Comparing MPC and FE [41].

2.2 Blockchain

The first blockchain was developed in 1991 by Haber and Stornetta [42], to address the problem of trusted digital time stamping. Their work was directly cited by

Satoshi Nakamoto [43], when the anonymous inventor first outlined the concept of bitcoin.

A blockchain is a sequence of records, called blocks, which are connected using cryptographic techniques. Each block contains some data (in the context of digital currencies, the transaction data, e.g. the two parties involved and the amount), the cryptographic hash of the block and the one of the previous block. This design naturally provides resistance to data modifications. Additional security is provided by the fact that blockchains are not centrally managed, but distributed in a peer-to-peer network. Each node communicates with others, leading to a constant control over the blocks of each node and to validating new blocks only if collectively approved.

2.2.1 Types of blockchains

Blockchains can be divided in two main categories:

- **Permissionless**, or also public blockchains are fully decentralized [44] and can be accessed by anyone. Examples of this type are practically all existant digital currencies, such as Bitcoin, Ethereum and Monero.
- **Permissioned**, i.e. private blockchains, are systems designed to exploit the blockchain technology within a selected group of entities, excluding unwanted actors from the system. In a permissioned blockchain the participants to the consensus are approved by the already present ones. This kind of setting finds the best application in situations where known parties want to rely on a trusted entity to intermediate their transactions such as consortia (e.g. KRAKEN). A permissioned blockchain is considerably more scalable than a public one, however this comes with a cost in the grade of decentralisation that it provides.

2.2.2 Consensus mechanisms

Due to the absence of a central authority, to approve some decision (such as the creation of a new block) blockchains need to employ a different paradigm, which goes under the name of *consensus protocol*. Only when all nodes of a blockchain have accepted the new decision, consensus is reached, and the blockchain gets modified in the determined way. Many algorithms have been proposed for this task, with the constraints that, to be valid, they need to be:

1. Decentralized
2. Byzantine fault tolerant (BFT), which refers to a famous 1982 paper [45]. This practically means that the blockchain, while elaborating consensus, needs to be resilient to the presence of malicious nodes, which try to manipulate the process by sending misleading signals to different nodes (e.g. telling to some node that a decision is valid and to some other nodes that it is not).

The most famous consensus algorithm is the **Proof of work** (PoW), used, among others, by bitcoin. In this case, the process of creation of new blocks is commonly referred to as "mining". Here, multiple parties compete to be the first to solve a *hashcash* problem, i.e. finding a hash (in a trial and error way) for the next block which contains at least a certain number of zeros at beginning. The more zeroes are required from the blockchain, the harder is to solve the PoW.

2.2.3 Smart contracts

In 1997, long before the advent of digital currencies, the concept behind smart contracts was proposed by Nick Szabo [46]. At their most basic level, smart contracts can be described as computer programs stored on a blockchain, of which they inherit the advantages. Thus, they provide a way for securely carrying out transactions between parties according to commonly decided rules, which are agreed beforehand among the parties involved and are then stated in a contract's code itself. Several digital coins support the use of smart contracts, such as Ethereum, that was created in 2014 [47]. Bitcoin supports smart contracts too, but it is severely limited in this regard compared to many other cryptocurrencies.

2.2.4 Permissioned blockchain via Hyperledger Fabric

The permissioned blockchain chosen for KRAKEN is Hyperledger Fabric [48], which stems from Hyperledger [49], a project founded by the Linux Foundation in 2015. Fabric is a private blockchain, designed to be implemented in business settings: in fact, in these situations, some of the features of public blockchains (e.g. the fact that every node in the world sees all the transactions, or the need for expensive consensus mechanism) can actually become detrimental to the users. Fabric's architecture is based on several main tenets [50]:

- **Channels**, i.e. detached parts of the blockchain, that can be used by a group of members to carry out transactions invisible to other ones.
- **Scalability**, as Fabric is built to easily allow the scaling of its nodes number, while also being optimized for requiring as little resources as possible to process large amounts of data.
- **Modularity**. This is perhaps Fabric's most important characteristic: depending on the use, most of the network's components can be removed or added whenever needed. This allows Fabric to serve the widest possible range of companies needs.

In january 2020, Fabric 2.0 was released [51], bringing several improvements especially on the smart contracts and privacy side.

2.3 Self Sovereign Identity (SSI)

Identity on the web has been handled in many different ways. The most common one has always been the set up of username and password. However this method centralises the management of the identities of the users of a certain system. Moreover the data needed for identity verification is shared by the user with every system that she is interacting with. This represents a considerable privacy concern. Another method broadly adopted after this latter one is federated identity. Which allows systems to directly retrieve from other systems (such as Google, Facebook) the user's identifiable information needed for the user registration. This latter registration mechanism facilitated the registration of users on other systems, however it did not solve the problem of privacy and represents a single point of failure for potentially multiple identity theft. A common flaw of both these systems is that the user has no control over her personal data which means that the operators of the services she relies on have the technical possibility (although legally regulated) to use her data in any way without the user awareness.

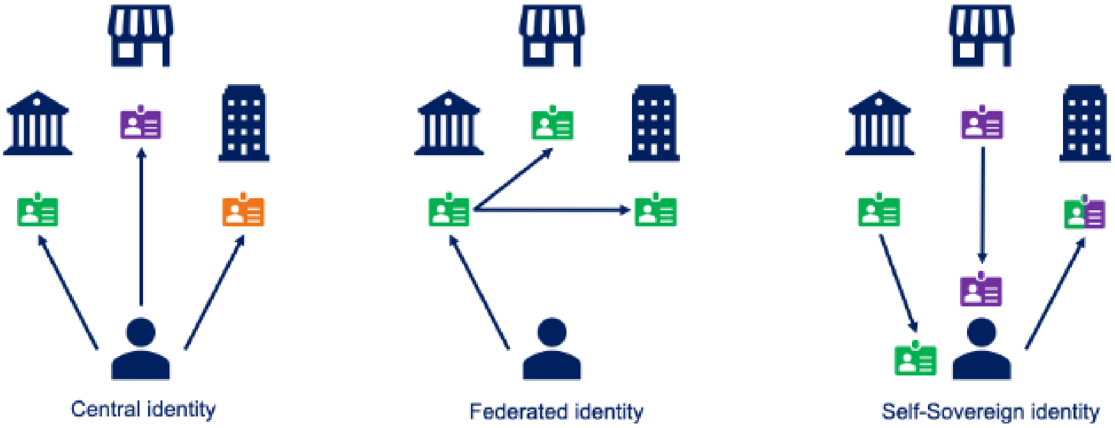


Figure 2.4: Comparison of different digital identities approaches [52].

To provide a digital identification scheme that would not introduce any intermediary between the user data and the 3rd party receiving it, self-sovereign identities

(SSI) were introduced. To clarify the main concepts underlying SSIs, one can refer to the following list [53] [54]:

1. **Existence.** It must exist indepedently from its digital representation.
2. **Control.** Users must have ultimate control over their digital identities.
3. **Access.** It should always be possible for users to access their own data.
4. **Transparency.** Systems and algorithms used to manage a network of identities must be open and transparent.
5. **Persistence.** Identities should be long-lasting (ideally, forever, or as long as the user wants to).
6. **Portability.** Users identities should be transportable. This increases identities lifetimes and further guarantees to users full control over them.
7. **Interoperability.** The same identity should be usable in as many contexts as possible.
8. **Consent.** A user permission must be given before identity information can be used.
9. **Minimalization.** When giving an identity's data to accomplish a task, only the minimum required amount of information should be shared.
10. **Protection.** Users rights must always be protected when at risk.

2.3.1 Verifiable credentials

A crucial component of any SSI system, verifiable credentials are pieces of information that could represent for example identity cards, driving licenses, credit cards etc, i.e. documents issued by some trusted authority that a user can leverage to demonstrate something about herself to a third party without relying on any centralised system.

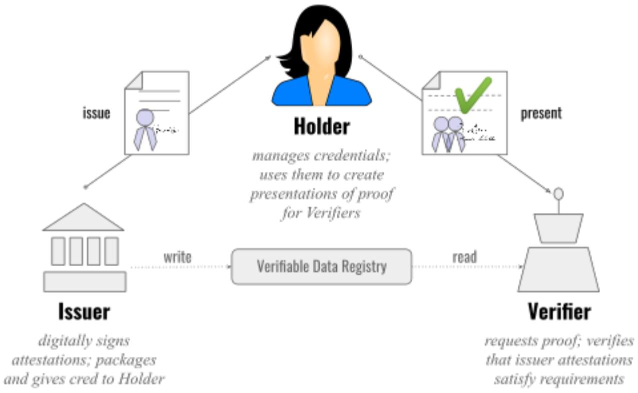

issue
Holder
manages credentials; uses them to create presentations of proof for Verifiers
present
Issuer
digitally signs attestations; packages and gives cred to Holder
write
Verifiable Data Registry
read
Verifier
requests proof; verifies that issuer attestations satisfy requirements

Figure 2.5: Basic mechanism behind the exchange of verifiable credentials [55].

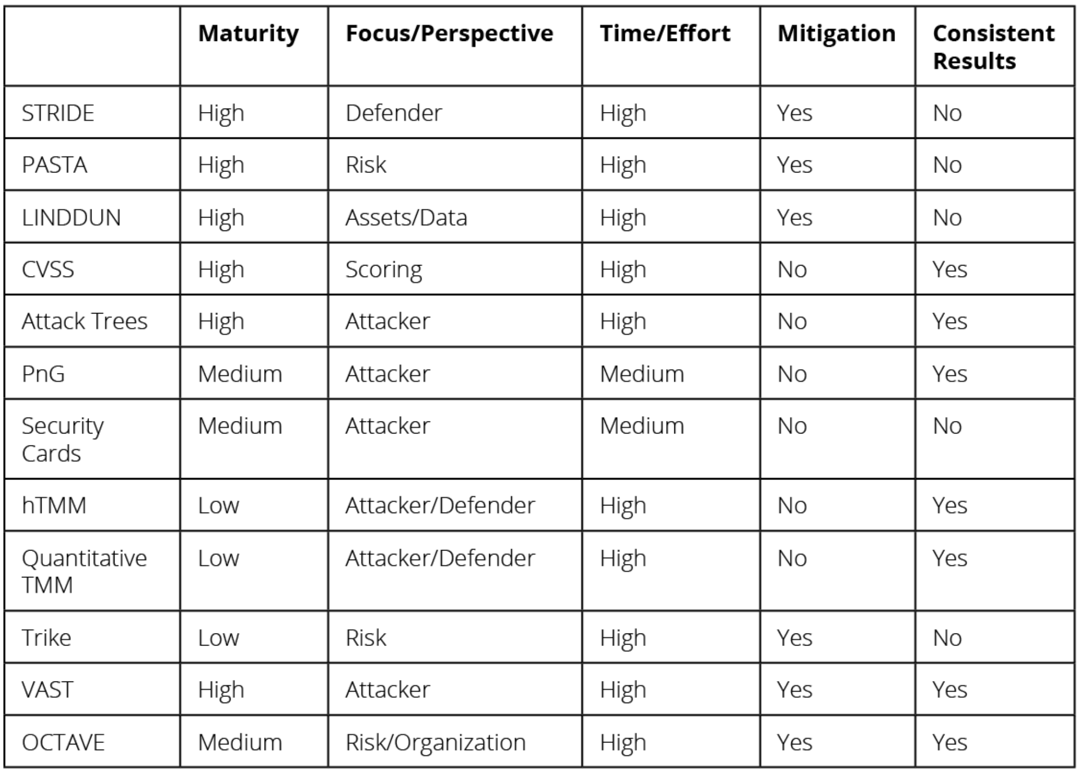
Three main subjects are involved in this process:

- **Issuers**, that release the verifiable credentials to the holders. Examples of issuers are governments, banks, or educational institutions.
- **Holders**, that request the credentials from the issuers, and can then use them to prove some claim about themselves to the verifier.
- **Verifiers** are any party trying to authenticate a holder's claim.

2.4 Threat modeling methodologies

Threat-modeling is the process through which a system's vulnerabilities can be discovered, listed, and prioritized for subsequent interventions. Several threat-modeling methods are today available, each one with pros and cons for a specific scenario.

Among popular methods, there is **STRIDE**, developed by Microsoft [57]. The name is an acronym for the six possible threats that method seeks to identify: Spoofing identity, Tampering with data, Repudiation threats, Information disclosure, Denial of service, Elevation of privilege. There is then **PASTA** (Process for



	Maturity	Focus/Perspective	Time/Effort	Mitigation	Consistent Results
STRIDE	High	Defender	High	Yes	No
PASTA	High	Risk	High	Yes	No
LINDDUN	High	Assets/Data	High	Yes	No
CVSS	High	Scoring	High	No	Yes
Attack Trees	High	Attacker	High	No	Yes
PnG	Medium	Attacker	Medium	No	Yes
Security Cards	Medium	Attacker	Medium	No	No
hTMM	Low	Attacker/Defender	High	No	Yes
Quantitative TMM	Low	Attacker/Defender	High	No	Yes
Trike	Low	Risk	High	Yes	No
VAST	High	Attacker	High	Yes	Yes
OCTAVE	Medium	Risk/Organization	High	Yes	Yes

Table 2.2: Comparison of 12 most common TMM strenghts and weaknesses [56].

Attack Simulation and Threat Analysis), which is broader in scope and can even be applied to non-coding scenarios [58].

Then, **LINDDUN**, which is a threat modeling methodology for systematically analyzing *privacy* threats in software architectures [18]. It is partially inspired from STRIDE: threats are analyzed along the categories linkability, identifiability, non-repudiation, detectability, disclosure of information, unawareness, and non-compliance. On top of the threat analysis, it offers mitigation strategies to handle the identified threats.

2.4.1 LINDDUN

To probe the KRAKEN's vulnerability to privacy threats, the chosen methodology was LINDDUN [17], which focuses on privacy aspects and thus was the best fit for the usage case. The LINDDUN framework consists in several subsequent steps [59]:

1. The first step is the creation of a model of the system for which privacy threats must be elucidated. In particular, LINDDUN requires this model to be a data-flow diagram (DFD) i.e. a scheme for information flow composed by four types of building blocks (processes, data flows, data stores, external entities) [18].

2. Second, potential threats are assigned to each of the elements of the DFD. Each of the four DFD building blocks is subject to different threat risks, with an overall total of seven threat categories – which can be easily remember using the aconym LINDDUN itself (linkability, identifiability, non-repudiation, detectability, disclosure of information, unawareness, and non-compliance)

3. The third step is composed of several phases. Initially, using threat tree patterns, a refinement of the threats is carried out. Then, assumptions (about the trustfulness of the architecture elements) are documented. Finally, threats themselves are also documented using a threat description template.

4. Next, a risk assessment is carried out to prioritize the identified threats.

5. In the fifth step, strategies to resolve (or at least mitigate) the threats are outlined.

6. Finally, the devised mitigation strategies are mapped into concrete privacy requirements that will have to be included in future development iterations.

3 Brokerage and Market platform for personal data (KRAKEN)

A data marketplace is a system where data can be exchanged between users. There are many kinds of data marketplaces, KRAKEN in particular “aims to enable the sharing, brokerage, and trading of potentially sensitive personal data, by returning the control of this data to citizens (data providers) throughout the entire data lifecycle” [15]. Data marketplaces can be implemented in different ways. The most simple is a totally centralised system, the system would collect data from data providers and provide the same data to the data consumers either by buying and reselling or by acting as a broker. This model works under a technical standpoint, but presents a set of problems derived by its centralisation. The marketplace has access to all the data coming from the users and the eventuality of misuse of this data depends solely on the marketplace. This includes disclosure of information to third parties, incorrect computation performed on special kinds of data products such as analytics computations and others.

The KRAKEN marketplace aims to be decentralised and privacy preserving by design. This is accomplished thanks to the combination of three main technologies that in the KRAKEN project are identified as the three pillars: a decentralised marketplace, Self Sovereign Identity (SSI), and a toolbox of cryptographic primitives for privacy-preserving computation. The marketplace itself will let individuals and institutions to trade access to personal data in a privacy preserving way.

By exploiting these pillars, the marketplace can offer three types of data products: Batch data products (consisting in the transfer of datasets), real time data products (consisting of real time data streams) and Analytics data products (a data product offering analytics computations on one or more datasets performed in a privacy preserving way). This thesis describes the Analytics data product use case.

The Blockchain is the decentralised element that brings power to the users. The users will have the same experience as with traditional marketplaces, but with the difference of a decentralised decision making mechanism powered by blockchain technology. Thanks to its decentralised nature, the marketplace can be considered a trustless intermediary between data owners and data consumers.

Self Sovereign Identity is the pillar that in KRAKEN is exploited to manage identities of data owners and data consumers through Verifiable Credentials. This pillar is essential for the marketplace as the decision making needs to work on the authentic input provided by verifiable credentials to check the eligibility of the users.

The toolbox of cryptographic primitives is the pillar used to manage exactly the personal data of the users. The marketplace is always unaware of the content of the data being exchanged between users. This is guaranteed for most of the data products types available on the marketplace, however in the case of the “Analytics data product”, the marketplace needs a trustless privacy preserving system to perform statistics on user’s datasets. This kind of system is provided, together with other cryptographic primitives, by the Multi Party Computation (MPC) technology. The architecture designed in the paper [16] and described in this paper tackles specifically the way KRAKEN will handle the “Analytics Data Product” by exploiting MPC.

4 Architecture for privacy preserving analytics

As outlined in [14], any valid datamarket must comply with at least two requirements:

- **Data privacy:** it must be guaranteed that the procedure will not leak any information about the data, to anyone (apart from the data analysis results provided to the buyers)
- **Output verifiability:** the datamarket must also guarantee that the data analysis results are truthful (i.e. not altered) and accurate.

For the time being, we do not take into account the *atomicity of payments* [14] as our architecture would need privacy preserving payment methods (that would be integrated with it). On the other hand, data privacy must be guaranteed right from the start. In particular, data owners must always be capable of specifying (if needed) which kind of analysis can be safely run on the data, and which kind of data consumers can be granted permission to purchase and access their data. The architecture delineated in the rest of this chapter was designed starting from these requirements.

A bird's eye view of the architecture is depicted in Figure 4.1. The pipeline starts from **device manufacturers**, who build the instruments that will acquire

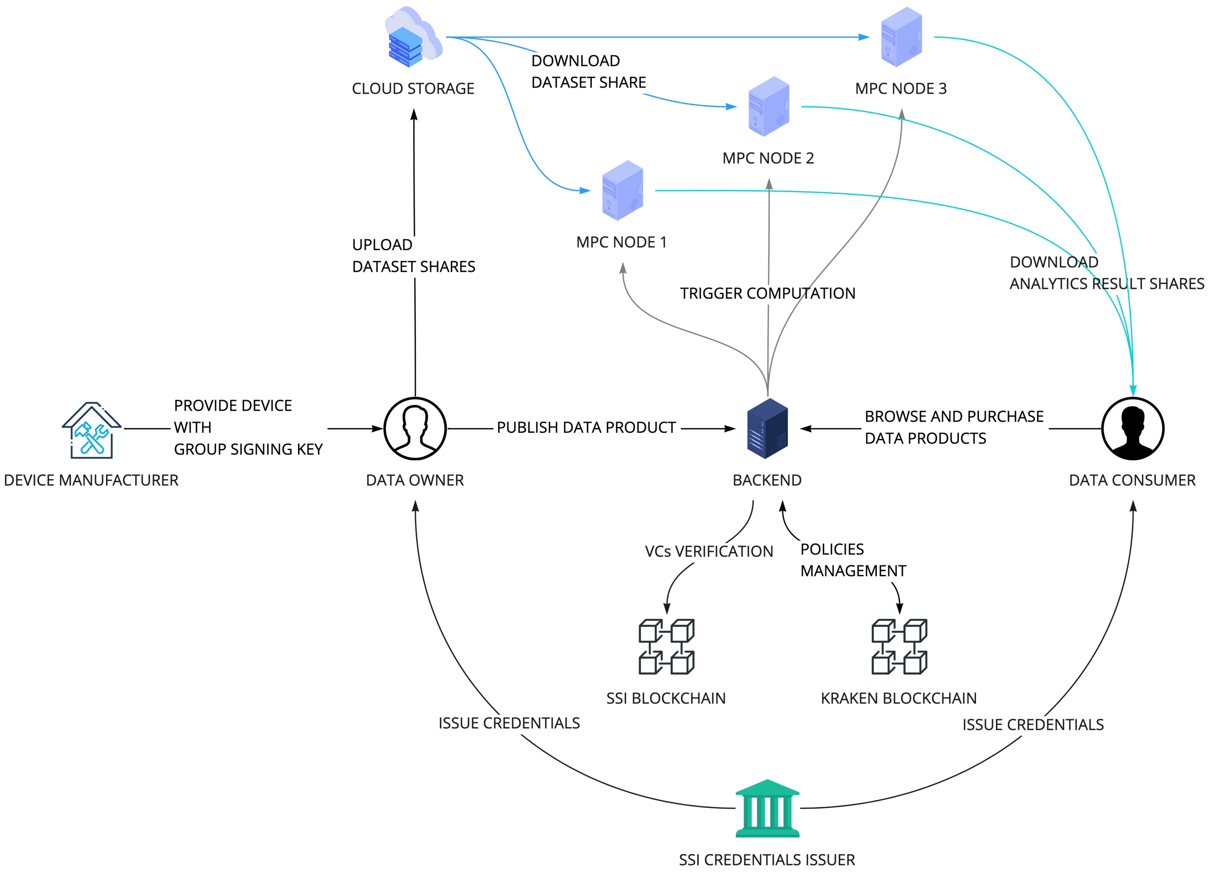


Figure 4.1: Overview of KRAKEN's architecture.

the data. The manufacturers provide these instruments with a group signing key which will serve to link the collected data to the device that it was collected from.

Data collection is carried out by **data owners**, who should also decide which analytics can or cannot be performed on their datasets. Together with this information, the data is uploaded in the cloud (a **cloud storage provider** is needed for this), and can be requested by the **data consumers**, who request for specific computations.

If the request aligns with the data owners indications, computation is performed on **computation nodes**, through a MPC procedure. The role of the **KRAKEN market place** is to handle the registration of data owners and consumers and

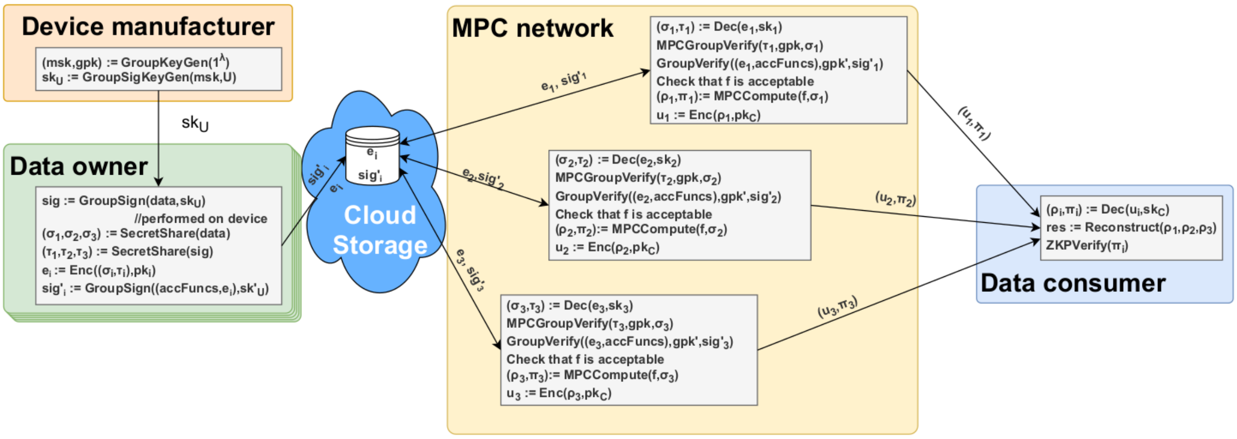

Device manufacturer
$(msk,gpk) := GroupKeyGen(1^\lambda)$
$sk_U := GroupSigKeyGen(msk,U)$
sk_U
Data owner
$sig := GroupSign(data,sk_U)$
//performed on device
$(\sigma_1,\sigma_2,\sigma_3) := SecretShare(data)$
$(\tau_1,\tau_2,\tau_3) := SecretShare(sig)$
$e_i := Enc((\sigma_i,\tau_i),pk_i)$
$sig'_i := GroupSign((accFuncs,e_i),sk'_U)$
sig'_i
e_i
e_i
sig'_i
Cloud Storage
MPC network
e_1, sig'_1
$(\sigma_1,\tau_1) := Dec(e_1,sk_1)$
$MPCGroupVerify(\tau_1,gpk,\sigma_1)$
$GroupVerify((e_1,accFuncs),gpk',sig'_1)$
Check that f is acceptable
$(\rho_1,\pi_1):= MPCCompute(f,\sigma_1)$
$u_1 := Enc(\rho_1,pk_C)$
e_2,sig'_2
$(\sigma_2,\tau_2) := Dec(e_2,sk_2)$
$MPCGroupVerify(\tau_2,gpk,\sigma_2)$
$GroupVerify((e_2,accFuncs),gpk',sig'_2)$
Check that f is acceptable
$(\rho_2,\pi_2):= MPCCompute(f,\sigma_2)$
$u_2 := Enc(\rho_2,pk_C)$
e_3, sig'_3
$(\sigma_3,\tau_3) := Dec(e_3,sk_3)$
$MPCGroupVerify(\tau_3,gpk,\sigma_3)$
$GroupVerify((e_3,accFuncs),gpk',sig'_3)$
Check that f is acceptable
$(\rho_3,\pi_3):= MPCCompute(f,\sigma_3)$
$u_3 := Enc(\rho_3,pk_C)$
(u_1,π_1)
(u_2,π_2)
(u_3,π_3)
$(\rho_i,\pi_i) := Dec(u_i,sk_C)$
$res := Reconstruct(\rho_1,\rho_2,\rho_3)$
$ZKPVerify(\pi_i)$
Data consumer

Figure 4.2: KRAKEN cryptographic architecture overview [16].

manages listings of available data sets. The information is stored on an internal blockchain and database. The Architecture proposed in the paper and summarised in the image 4.1 includes all of these elements, that are described in more detail in the following sections.

4.1 Users

Users are divided in two categories: data owners and data consumers. In the following paragraphs will be described these two different categories, their interests in joining the platform and the derived requirements for the architecture.

4.1.1 Data owners

The data owners are the sellers of the platform. They own the raw datasets and their interest in joining the platform consists in selling analytics about their data in a privacy preserving way. This means that the architecture must comply with the following requirements:

- No one other than the data owners is able to know the original data;

- No analytics functions other than the ones allowed by the data owners can be performed on users data on the platform;
- The results must be sold to eligible buyers exclusively;
- The eligibility of data consumers must be checked with institutional level certificates in a privacy preserving way;
- All the above must be performed in a trust-less environment.

4.1.2 Data consumers

The data consumers are the buyers of the platform, and their interest in joining the platform consists in buying results of analytics computation performed on the data owners dataset.

Thus, the architecture is also subject to the following requirements:

- The analytics must be performed on datasets whose provenance is guaranteed;
- The analytics must be performed correctly.

4.2 Device manufacturers

The device manufacturers have the role of producing the devices that will be used to collect the records of the measurements of users. These measurements will constitute the datasets to be analysed. The importance of these actors in the architecture is motivated by the requirement of data provenance. To ensure the quality of the raw data, users will need to use devices provided with an hardware feature to sign collected data with a key belonging to a publicly known group signature schema.

4.3 Blockchain

The Blockchain is one of the two decentralised element of the architecture. The role of this component is to allow transactions happening between data owners and data consumers. The transactions allowance depends on:

- The policies set by the data owner at the moment of data registration;
- The credentials owned by the data consumer (that have not been revoked) at the moment of the transaction;
- And on the regulations that the specific transaction needs to comply with in the nation of the data owner and the nation of the data consumer.

The transactions are mediated by the backend that receives them from the users.

4.4 Backend

This component has multiple roles:

- Storage for user credentials and data products catalog;
- Frontend provider for users;
- API to receive requests from data owners and data consumers;
- Webhook to alert the MPC network of new data products transactions and the blockchain to forward request of allowance of transactions.

4.5 Frontend

The Frontend is the software running on the user system (laptop or smartphone). This component has multiple roles:

- Provide features to the users in the form of UI tools to allow them to provide the platform with their SSI credentials, perform the registration on the platform, browse the data catalog, publish and buy data products and other features (such as account page, etc...);
- Send requests to the backend;
- Perform encryption and decryption of the results;
- Other actions related to data product publication described in the section 4.12.3.

4.6 MPC network

The MPC network is the element of the architecture that performs the privacy preserving analytics computation. This network is composed by the MPC nodes.

This component has multiple roles:

- Receive messages from the backend to trigger analytics;
- Receive messages from the data owners to get the location of their encrypted shares of the datasets to be analysed;
- Retrieve datasets from the cloud storage of the data owners.
- Send analytics results to data consumers.

The role of this component is empowered by the fact that the nodes, to perform their operations, receive information that must not be shared with the other nodes. For this reason, to every MPC node is associated a keypair. The public key of every MPC node is considered a public information, well known by the software running on the user's system.

The keypairs of n MPC nodes are defined in this way:
$Np_1, Np_2, ...Np_{n-1}, Np_n$ are the public keys of the nodes.
$Ns_1, Ns_2, ...Ns_{n-1}, Ns_n$ are the secret keys of the nodes.

The MPC nodes will belong to well known organisations that are supposed to behave correctly. However the network is secure against malicious behaviour of a subset of its components. This security depends on the security settings that can be tuned to the point of needing just one honest MPC node to ensure that no unexpected operations are performed. However, the more secure is the network, the less scalable it becomes. On the basis of this, in the paper we assumed that at least one MPC node is always honest.

4.7 SSI agent

The SSI agent is the software component responsible for the management of identities in KRAKEN. It will connect and communicate with the agents of the users' SSI wallets. The roles of this component are:

- Receive requests from the backend;
- Query the SSI blockchain;
- Establish DID connections with the SSI wallets of the users;
- Issue credentials and receive proofs of credentials from users.

4.8 SSI credential issuers

The SSI credentials issuers are the entities that provide the credentials to the users. A credentials issuer is an entity that could belong to institutions, companies or other kind of entities that can state a certain characteristic of a person. A typical example in the KRAKEN case is the need to demonstrate the membership in a research center or an hospital to buy the access to a data product. The research center/hospital SSI credentials issuer would have to release a certificate to the member. The member will then use this certificate on the platform to perform the purchase. This entity is considered a trusted party as an assumption.

4.9 SSI wallet

The SSI wallet is an application running on the user's smartphone. This application includes an SSI agent that will be used to establish a DID connection with the KRAKEN SSI agent 4.7.

The SSI wallet will store the SSI credentials released to the user and present them to KRAKEN. Depending on the operation performed on the platform, the wallet will also present SSI credentials to KRAKEN.

4.10 Cloud storage

The Cloud will be used by the KRAKEN users to store the encrypted datasets. The storage is not specific; the users can freely decide to use any cloud storage system until the access to the dataset is public and can be performed through a link. This component will receive the datasets to store from the users and provide the datasets to the MPC nodes. A peculiar characteristic of this component is that the datasets are never retrieved by any component except the MPC network.

4.11 Requirements and assumptions

In the previous sections have been identified requirements and assumptions of the KRAKEN platform:

REQUIREMENTS:

1. No one other than the data owners is able to know the original data;
2. No analytics functions other than the ones allowed by the data owners can be performed on users data on the platform;
3. The results must be sold to eligible buyers exclusively;
4. The analytics must be performed on datasets whose provenance is guaranteed;
5. The analytics must be performed correctly;
6. The eligibility of data consumers must be checked with institutional level certificates in a privacy preserving way;
7. All the above must be performed in a trust-less environment.

ASSUMPTIONS:

1. At least one MPC node is honest;
2. The credentials issuers are considered trusted parties.

4.12 Actions

These are the three typical data flows.

4.12.1 User registration

User registration is performed exploiting SSI technology. The actors involved are the user, the backend and the SSI credentials issuer. The steps to perform this action are the following:

- The user receives one or more SSI credentials from the organisations he belongs to. The details of this operation depend on the specific organisation issuing the credentials, that is considered a trusted party as an assumption.
- The second step is the request for registration on KRAKEN. To do this, the user will establish a DID connection with the backend agent and present his credentials.
- Once the Agent confirms the validity of the credential consulting the SSI blokchain, the credential is saved in the credentials storage;
- The last step consists in providing to the user a group signing key to be able to sign messages on behalf of the group of users of the marketplace without risking to reveal his identity. The specific use of this key is exposed later in this chapter.

4.12.2 Data collection and pre-processing

The collection of the data records is a process that happens on a user's device. This device can be of any kind and can register any kind of data, the only requirement is the hardware feature of signing every record with the device's group signing key Ugs.

4.12.3 Data registration

To publish the dataset on the marketplace, the user needs to preprocess the data because of the requirements previously described (4.11). The steps to perform this action are the following:

- Data collection (4.12.2);
- Login

- Filling of the metadata such as title, description, image, etc...;
- Set up of the policies that will govern the criteria used by the blockchain to select eligible buyers;
- Preparation of the dataset for SMPC computation. This step consists in splitting the dataset in a number of shares equal to the number of nodes that have been deployed to run the MPC network.
- Encryption of the shares using MPC nodes public keys.
- Signing of every share using the group signing key Ugs;
- Signing of the allowed functions using the group signing key;
- Storage on a cloud storage chosen by the data owner of a bundle of all the pre-processed dataset with signatures.
- Publication request to the Backend.

4.12.4 Data analysis request

To make a purchase on the platform, the user needs to perform the following steps:

- Login
- Browse catalog to find datasets of interest;
- Declare the function to be evaluated on the data;
- Send purchase request to backend;
- User's policies check on the blockchain and approval (if not approved, stop);
- The backend communicates to the MPC network the new computation request;

- The MPC network retrieves the data from the user's cloud storage. Specifically, every node fetches its own share and all the nodes fetch the allowed functions and signatures;

- The MPC nodes analyze these information, verifying the validity of the data signatures and that the requested function is allowed for the computation (according to the data owner's policy).

- The shares are decrypted using the nodes' private keys, then the network verifies that the shared signatures are valid and calculate the output (encrypted with the consumer's public key) to provide to the data consumer.

- The data consumer receives the output of the computation. In addition to this, a guarantee that the analysis was performed correctly and that the inputs verification was successful.

5 LINDDUN Privacy analysis

This section is based on the author's contributions in a published research paper [16]. As already explained in section 2.4.1, LINDDUN allows to take into account architectural privacy aspects, and in particular it does so early on during the development lifecycle [18]. In this chapter, the steps leading to LINDDUN privacy analysis of KRAKEN's architecture are reported. Following the standard procedure [59], the paper starts by outlining the data-flow diagrams (DFDs) corresponding to KRAKEN's most important user actions, which are: **(1) User registration**, shown in Fig 5.1, **(2) Data availability registration**, in Fig 5.2, and **(3) Perform data analysis**, also in Fig 5.2.

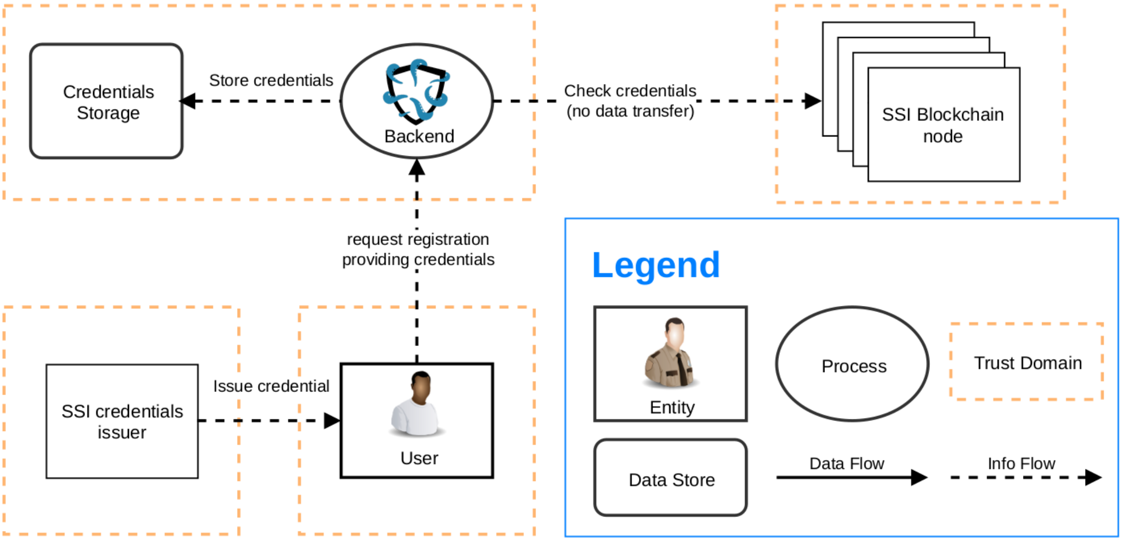


Figure 5.1: DFD for the user action of registering. [16]

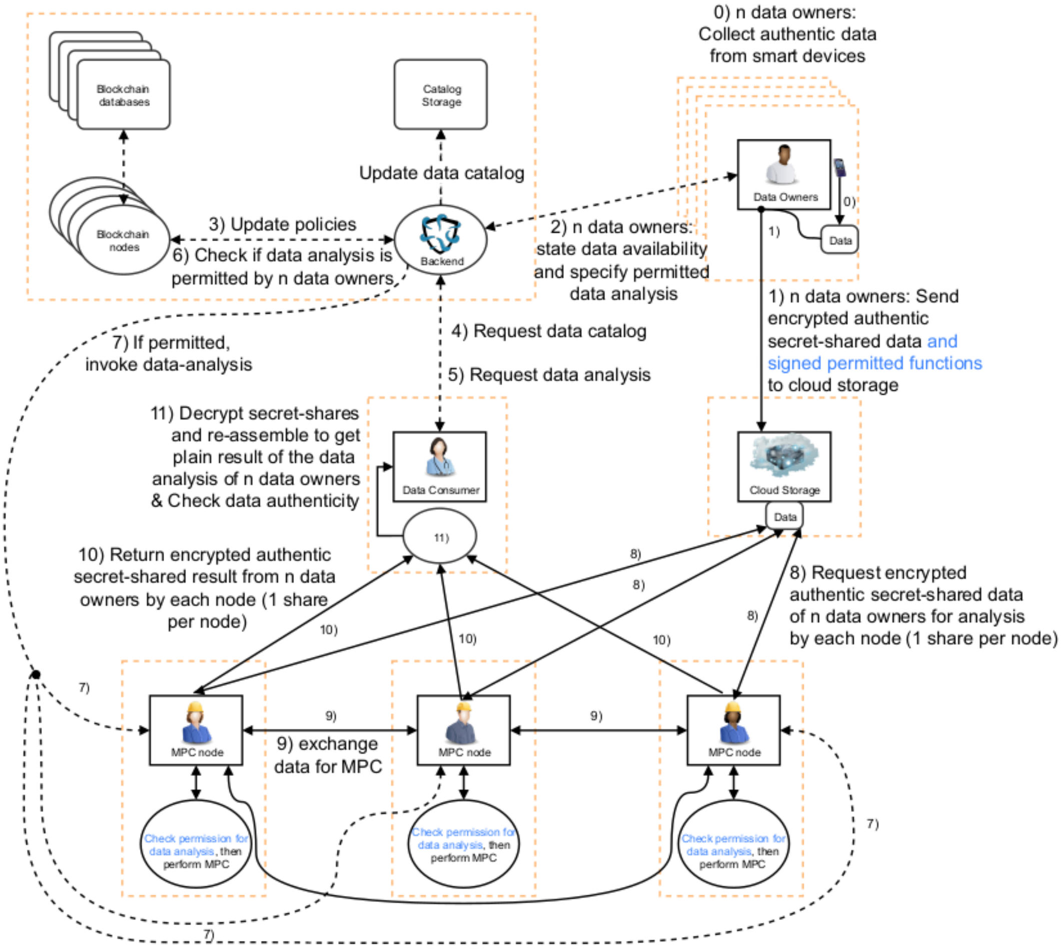
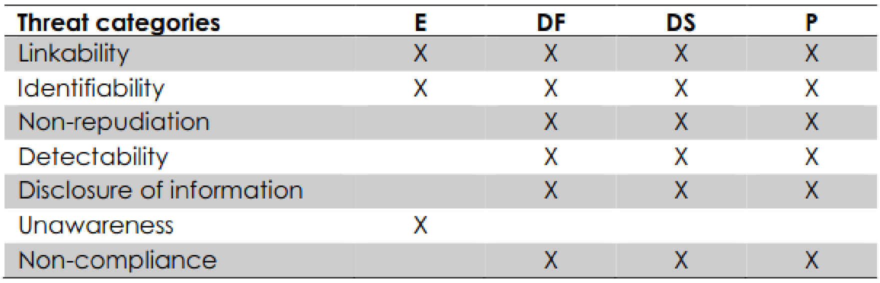


Figure 5.2: DFD for the user actions of registering data and performing data analysis. The legend is as in Fig. 5.1 [16]

More specifically, and for easier interpretability, every DFD is split in two parts, corresponding, respectively, to the actual personal data flow (from its uploading up to its analysis results) and to the flow of information, i.e. all the complementary data required for the user actions to be completed. Finally, the components of each DFD are mapped to LINDDUN threat categories, the threats are prioritized for intervention, and mitigation strategies are proposed.

5.1 Tables of threats

Following LINDDUN mapping template [17], The elements of each DFD is mapped to the seven threat categories: **L**inkability, **I**dentifiability, **N**on-repudiation, **D**etectability, **D**isclosure of information, **U**nawareness, **N**on-compliance. DFDs contain different element types, which are subject to different kinds of threats, as shown in Table 5.1.



Threat categories	E	DF	DS	P
Linkability	X	X	X	X
Identifiability	X	X	X	X
Non-repudiation		X	X	X
Detectability		X	X	X
Disclosure of information		X	X	X
Unawareness	X			
Non-compliance		X	X	X

Table 5.1: Mappig LINDDUN components (privacy threats) to DFD element types: E=Entity, DF=Data Flow, DS=Data Store, P=Process (from [59]).

Thus, one has to first identify all the elements composing each DFD, and then outline its possible threats in a threat table. The outcome is reported in table 5.2 (user registration), in table 5.3 (registration of data availability) and in table 5.4 (performing data analysis).

Note that in these tables some threats that according to the LINDDUN baseline should be highlighted, are not. This is because we can make some considerations and assumptions (that will be laid out in the next section 5.2) that are specific to KRAKEN's architecture. The threat Non-compliace can also be excluded for several DFD elements: for the data and information flows (due to the adoption of TLS), for the Data Stores (due to the data-minimization principle) and, finally, also for processes (as KRAKEN does not deal with any personal data). The MPC network is a decentralised and consequently trusted entity. For this reason no threat was assigned to it.

DFD Elements	Threat Target	Privacy Threats						
		L	**I**	**N**r	**D**e	**D**i	**U**	**N**c
Data Store	Credentials Storage	✗	✗					
Info Flow	Issue credentials	✗	✗	✗	✗	A3		
	Request Registration	✗	✗	✗	✗	A3		
	Check credentials	✗	✗	✗	✗	A3		
	Store credentials							
Process	Backend							
Entity	User							
	SSI credentials issuer	A1	A1	A1	A1	A1	A1	A1
	SSI Blockchain							

Table 5.2: LINDDUN's threat table of the 1 st user action [16], performing user registration. An ✗ in a cell indicates a privacy threat for the corresponding threat target. Cells labeled by "Ax" are no threats because of the indicated assumptions.

DFD Elements	Threat Target	Privacy Threats						
		L	**I**	**N**r	**D**e	**D**i	**U**	**N**c
Data Store	Catalog Storage	✗	✗					
	Cloud Storage	✗	✗					
Info Flow	collect authentic data from smart devices							
	send encrypted authentic secret-shared data to cloud storage and signed permitted functions	✗	✗	✗	✗	A3		
	state data availability and specify permitted data analysis	✗	✗	✗	✗	A3		
	Update data catalog							
	Update policies							
Process	Backend							
	Blockchain nodes							
Entity	Data Owner						✗	
	Data Consumer							
	Cloud Storage							

Table 5.3: LINDDUN's threat table of the 2$^{\text{nd}}$ user action [16], performing registration of data availability. An ✗ in a cell indicates a privacy threat for the corresponding threat target. Cells labeled by "Ax" are no threats because of the indicated assumptions.

DFD Elements	Threat Target	Privacy Threats						
		L	**I**	**Nr**	**De**	**Di**	**U**	**Nc**
Data Store	Catalog Storage	✗	✗					
	Cloud Storage			✗	✗			
	Data Consumer							
Info Flow	4) Request data catalog	✗	✗	✗	✗	A3		
	5) Request data analysis	✗	✗	✗	✗	A3		
	7) Invoke data analysis	✗	✗	✗	✗	A3		
Data Flow	8) Request enc. auth. se-sha. data	✗	✗	✗	✗	A3		
	9) Exchange se-sha. data	✗	✗	✗	✗	A3		
	10) Return enc. analysis result	✗	✗	✗	✗	A3		
Process	6) Check permission of data analysis							
	9) Check permission & Perform MPC					A2		A2
	11) Decrypt analysis result & Check authenticity							
Entity	Cloud Storage							
	MPC Nodes							
	Data Consumer							

Table 5.4: LINDDUN's threat table of the 3rd user action [16], performing data analysis. An ✗ in a cell indicates a privacy threat for the corresponding threat target. Cells labeled by "Ax" are no threats because of the indicated assumptions.

5.2 Elicitation of threats

The choice of which threats (among all the theoretically possible ones) should be targeted and which ones should not can be aided by making assumptions. More precisely, and as stated in the LINDDUN tutorial [59], "assumptions are explicit or implicit choices to trust an element of the system (e.g., human, piece of software) to behave as expected". The following list shows the four assumptions that have been made in the paper [16]:

Assumption 1. Trusted SSI credential issuer. An eventual collaboration between the KRAKEN backend and the credential issuer could reveal to the backend the real identity of the users. This assumption is necessary to prevent the kraken backend to link real identities with users. In a real world scenario, this assumption could be enforced by multiple security measures such as designing the issuer as a distriburted party that adopts threshold cryptography or through regular audits.

Another safety measure would be to inform users of the risk of KRAKEN collaborating with the credential issuer to reveal real identities so that they could check the legal relationship between the two entities before using the service.

Assumption 2. The MPC network requires a minimum of one honest node. The protocol version adopted in the MPC network in KRAKEN is the strongest in terms of security, which means that only one honest node is needed to avoid security breaches. With this setting, having at least one honest MPC node prevents any malicious actor to access the computation results or the original datasets. To operate, the MPC network needs the participation of every node, so to prevent the actions just described, the only honest node would just need to refuse participating to preserve the security of the system. A mitigation of the risks related to this assumption is to carefully select the participants in the MPC network. The nodes could be even more than the three required for the protocol to function and the choice on which ones to adopt can be delegated to the user. For special cases, one of the MPC nodes needed for a computation could be even deployed in the data owner's facility to have the guarantee that at least one node is honest.

Assumption 3. Every communication happening between two entities not belonging to the same trust domain happens through transport-layer security (TLS) This assumption regards any kind of communication happening between entities in different trust domains. We assume that the only possible leak of information that could happen during these communications is the metadata as the rest of the information transferred is protected by TLS.

Assumption 4. Trust boundaries are implemented in a secure way and any entity that is violated by a malicious actor is considered in total control of the actor. This assumption can be considered also a simplification for the analysis. It considers corrupted systems to be totally corrupted and excludes partial corruptions of systems. With this assumption in place, any trust domain violated by a corrupted actor makes the trust domain in her total control.

5.2.1 Mapping LINDDUN's Privacy Threats to the DFDs.

In the following, will be enumerated (following the "LINDDUN" acronym itself) and outlined the threats discovered with the threat tables in the paper [16].

Threat1 (Linkability in one or more storages). An insider of KRAKEN links data coming from the catalog, credentials, policies or purchases storages.

Assets, stakeholder, threats: Linking different users or different information of the same user could lead to gain more information about users than expected.

Primary misactor: An internal user that has access to the data storages of the backend and/or of the internal blockchain.

Basic flow: (1) The insider gains specific information by querying the data store. (2) The obtained set of information can be linked.

Preconditions: The user has updated the system with some informations or is at least registered.

DFD elements: Credentials storage, Catalog storage, Policies storage, Purchases storage, Cloud storage.

Remarks: This threat could lead to identification. When applied to the credentials storage, the probability is much lower as credentials have a high level of minimization of information.

Threat 2 (Identifiability in one or more storages). An insider of KRAKEN identifies one or more users in a set of data coming from one or more storages. Assets, stakeholder, threats: The identity of the user must be unknown in the KRAKEN.

Primary misactor: An internal user that has access to the data storages of the backend and/or of the internal blockchain.

Basic flow: (1) The insider gains specific information by querying one or more data stores. (2) The obtained set of information can be linked and can lead to identification of one or more users.

Preconditions: The user has updated the system with some information or is at least registered.

DFD elements: Credentials storage, Catalog storage, Policies storage, Purchases storage, Cloud storage.

Threat 3 (Detectability of data existence). The user uploads the data on the cloud without publishing on KRAKEN, revealing the existence of data.

Assets, stakeholder, threats: The detection of the existence of the data must take place at the will of the user.

Primary misactor: The cloud or an external actor.

Basic flow: The misactor checks periodically the cloud storage until the data is uploaded.

DFD elements: Cloud storage

Threat 4 (Detectability in communication between different trust domains). An internal/external actor can detect user actions by listening to re- quests.

Assets, stakeholder, threats: The detectability of user actions is not expected outside of the scope of the interested actors.

Primary misactor: A skilled internal/external actor that has access to the network of the user and can inspect user's packets.

Basic flow: (1) The misactor intercepts packets between a user and KRAKEN. (2) Whenever a packet is sent, an action has been detected.

DFD elements: All the data flows between two different trust domains.

Remarks: This threat disclosure of information is not expected as the communication happens through TLS.

Threat 5 (Linkability of IP addresses in communication between different trust domains). An internal/external actor can link different events to the same user by listening to user's requests.

Assets, stakeholder, threats: Any information that can be gained by linking user actions are not expected to be known by anyone except the user.

Primary misactor: A skilled internal/external actor that has access to the network of the user and can inspect user's packets.

Basic flow: (1) The misactor intercepts packets between a user and KRAKEN. (2) Whenever a packet is sent, IP addresses are collected. (3) The misactor links packets with the same IP.

DFD elements: All the data flows between two different trust domains.

Remarks: This threat disclosure of information is not expected as the communication happens through TLS.

Threat 6 (Linkability of IP addresses in communication between different trust domains leads to identifiability). An internal/external actor can identify users by linking different events to the same IP by listening to user's requests.

Assets, stakeholder, threats: User's identity and any information that can be gained by linking user actions are not expected to be known by anyone except the user.

Primary misactor: A skilled internal/external user that has access to the network of the user and can inspect user's packets and knows or can link to an IP address the user's identity.

Basic flow: (1) The misactor intercepts packets exchanged between a user and KRAKEN. (2) Whenever a packet is sent, IP addresses are collected. (3) The misactor links packets with the same IP. (4) The gained information, together with any information that can link the IP to a user (e.g., insecure traffic with other systems) leads to the identification of the user.

DFD elements: All the data flows between two different trust domains.

Threat 7 (Non-repudiation of encrypted data). The cloud storage cannot repudiate that encrypted data is available.

Primary misactor: Data stores which do not handle data access properly.

DFD elements: Cloud storage (data store; user action (UA) 2/3).

Threat 8 (Non-repudiation of communication between different trust domains). An entity cannot repudiate that he sent a message to another entity within a different trust domain.

Primary misactor: An external user that has access to the network of the user and can inspect user's packets.

DFD elements: All data flows between two different trust domains.

Threat 9 (Unawareness of the data owner). First, a data owner provides data for which he is not allowed, such as by national law. Second, a data owner does not take care of the defined analysis policies/permissions, such that a con- sumer could learn something about the owner based on the analysis result. For example, if an owner allows an analysis without any other owners in addition (aggregated analysis), then, e.g., an average would reveal the actual data.

Primary misactor: A data owner making data available.

DFD elements: Data owner (entity; UA 2).

Threat 10 (Non-deletion of data in cloud storage). The data owner is not aware that the cloud storage is in possession of his data.

Primary misactor: A cloud storage not deleting user's data.

Basic flow: (1) The data owner requests the cloud storage to delete his data. (2) The cloud storage does not delete the data. (3) The data owner is not aware that the data is stored on the cloud storage.

DFD elements: Data owner (entity; UA 2).

5.3 Prioritization of threats

Before tackling the identified threats, it is practically useful to rank them by the urgency with which they need to be addressed, also known as priority. This quantity

Likelihood	Impact	Priority
low	low	low
low	medium	
medium	low	

Likelihood	Impact	Priority
low	high	medium
medium	medium	
high	low	

Likelihood	Impact	Priority
medium	high	high
high	medium	
high	high	

Table 5.5: Threat prioritization depending on likelihood and impact [16].

results from the likelihood and impact of a given threat. The degree of intensity of these two parameters is ranked with the values: "low", "medium" and "high".

The likelihood is a result of the easiness of accomplishing a given threatening action and the gain of the threatening actor. The impact is given by the severity of a successful attack for the user. Depending on the threat, its value is outlined in table 5.7.

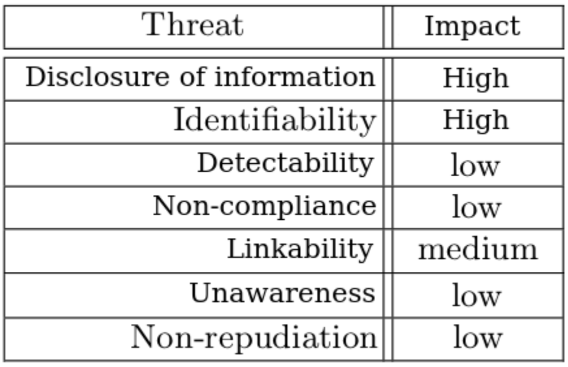
The table 5.5 describes how the priority is derived from the combination of the intensities of likelihood and impact. The table 5.6 lists the intensities of likelihood and impact applied to every threat and the implied priority value. The following list contains the reasons presented in the paper [16] for the assigned values of likelihood and impact of every threat.

- **Linkability in one or more storages.** In this threat the likelihood value is medium as even if the misactor needs to be an insider, exploiting more than one storages leads to better outcomes in trying to link user's data. The impact is medium as the threatened asset is the linkability of user's data, that if combined with identifiability reveals which users performed certain actions.
- **Identifiability in one or more storages.** The likelihood value is low as the misactor would need more information other than the ones contained in the KRAKEN system to identify one or more users. The impact is high as the threatened asset is the identity of users that is considered high priority asset.

Threat	Likelihood	Impact	Priority
Linkability in one or more storages	medium	medium	medium
Identifiability in one or more storages	low	high	medium
Detectability of data existence	medium	low	low
Detectability in communication between different trust domains	low	low	low
Linkability of IP addresses in communication between different trust domains	low	medium	low
Linkability of IP addresses in communication between different trust domains leads to identifiability	low	high	medium
Non-repudiation of encrypted data	low	low	low
Non-repudiation of communication between different trust domains	low	low	low
Unawareness of the data owner	low	high	medium
Non deletion of data in cloud storage	low	low	low

Table 5.6: Overview of threat prioritization [16]. Threats that are not effective due to our assumptions are not included in the table.

- **Detectability of data existence.** The likelihood is medium as the threatened information is public by default. The misactor could be an external user without any specific capability that needs to know by other means that the specific data is destined to KRAKEN. In the case where the misactor is the cloud storage that may know the identity of the user, the cloud storage would still need to know by other means that the specific data is destined to KRAKEN. In a hospital scenario, If a patient decides to adopt the hospital's cloud system, the hospital could make assumptions on the content of the dataset by linking the detection of the dataset existence with information related to the patient. However, this situation is highly unlikely as the user can choose any cloud system without relying on the hospital's one. The impact is low as the data is always encrypted, existence of data may be detected, but the data itself does not leak.



Threat	Impact
Disclosure of information	High
Identifiability	High
Detectability	low
Non-compliance	low
Linkability	medium
Unawareness	low
Non-repudiation	low

Table 5.7: Overview of the impact value of the different threats.

- **Detectability in communication between different trust domains.** The likelihood value is low as the misactor is an external skilled individual that has access to the network of the user or to the KRAKEN network. The impact is low as the threatened asset is the detectability of user actions, which is considered a low-priority asset.

- **Linkability of IP addresses in communication between different trust domains.** This threat depends on the same actions and actor needed to perform the previous one, so the likelihood is the same. The impact is medium as the threatened asset is the linkability of user's data, that if combined with identifiability reveals which users performed certain actions.

- **Linkability of IP addresses in communication between different trust domains leads to identifiability.** This threat depends on the same actions and actor needed to perform the previous one, so the likelihood is the same. The impact is high as the threatened asset is the identity of users that is considered high priority asset.

- **Non-repudiation of encrypted data.** As cloud-storage providers usually use unguessable file links, the likelihood for this threat is low. The impact is low as one cannot identify the receiver of the ciphertext recover its content.

- **Non-repudiation of communication between different trust domains.** Similar as for detectability of communication, likelihood and impact are low.

- **Unawareness of the data owner.** The likelihood value is low as the personal data provided belongs to the user and therefore it is her own interest to provide data that does not affect her in terms of non compliance with regulations. Moreover (for the second case) the outcome of publishing the analysis of a dataset without a pool of other user's datasets would not be appealing for a possible buyer. The impact is high as the threatened asset is the personal information of users that is considered high priority asset.

- **Non deletion of data in cloud storage.** The likelihood value is low as the outcome of performing this action would lead the cloud storage to have an encrypted dataset that is not possible to consume in any way. Because of Assumption 2, the cloud storage cannot collaborate with the MPC nodes to unveil the data as at least one of them is honest. The impact is low as the threatened asset is the unawareness of users that is considered low priority.

5.4 Mitigation of threats

The set of mitigations identified and listed in the paper [16] are hereby exactly reported:

- **Linkability in one or more storages.** To mitigate the threat on the SSI storage side, on registration phase the system can request to the user the minimum set of credentials required to allow the user to get registered and do not lead to linkability/identification. To mitigate the threat on the other storages, the system can display a suggestion to user saying to non include any identifiable information before the publication of any product.

- **Identifiability in one or more storages.** This threat depends on the previously described threat “Linkability in one or more storages”, the mitigation applied in that threat mitigate consequently also this one.

- **Linkability of IP addresses in communication between different trust domains.** To avoid the misactor to understand that the communication is happening with KRAKEN, avoiding linkability and resulting identifiability, onion routing (like Tor [60]) can be used.

- **Unawareness of the data owner.** The mitigation can be implemented on the user’s frontend side in two complementing ways. First, the system provides thorough documentation that explains potential risks when offering certain data sets for data analytics. Second, based on the type of data and the acceptable function families, privacy metrics [61] are displayed to make the user aware of any risks. Thereby, the system is able to warn the user, e.g., before allowing the computation of an average but where the user’s input is the only considered data set.

5.5 Privacy Analysis Outcome

The privacy analysis was part of a cyclic methodology of continuous improvements. At every cycle, threats were discovered through the analysis and became the inputs for the planning of the next cycle.

There are some main threats and solutions derived from this approach. The distinction of the data flows in information flow and personal data flow was a key improvement to focus the architecture around the protection of different kinds of data. Specifically, it is noticeable in the current DFDs that the personal data of the users is never handled by any entity in a not encrypted manner and that only the data consumers eligible to access data providers’ data actually access the data.

The direct interaction of the user with the platform could have led to identifiability threats and this is what primarily moved the decision of adopting group signatures. Through this kind of signature the user accomplishes the same purposes with more privacy. The remaining of the collected threats are relative to unique elements of the architecture and require modifications that influence their internal functioning. The mitigations identified suggest the adoption of practices like data minimisation and documentation and privacy tool to provide to the data owner.

6 Conclusions and Future Work

In this thesis was outlined an architecture proposal for the KRAKEN marketplace, explained the technologies that enable it and reported the privacy analysis that validates it. The architecture allows a data marketplace to compute analytics on datasets without ever knowing inputs and outputs of the computations. In this way users can profit from the usage of their data in a totally privacy preserving way and with the security of a decentralised system. This is thanks to the cryptographic tools adopted to build its core components and the decentralised technologies adopted for the decision making.

Some observations need to be done on the security, privacy and decentralisation aspects of the system. The MPC network is the privacy preserving decentralised intermediary between data providers and data consumers. The privacy of the data is secured by the decentralisation of the MPC network whose consensus algorithm can tolerate a minimum of one honest node to work properly. However the decision making regarding the data consumers that can access a data product depends on the permissioned blockchain. A permissioned blockchain (such as the one adopted in KRAKEN: Hyperledger Fabric) can adopt a variety of consensus algorithms that have a malicious nodes tolerance threshold considerably lower than the MPC network. With these considerations it is possible to conclude that the security guarantees provided by the MPC network are valid in the specific context of transactions happening between the users allowed by the blockchain. However considering blockchain and MPC network together, the security guarantees are of the entire system are the ones of the weakiest point that in this case is the blockchain.

Another weak point is the lump sum payment model. The architecture does not provide the possibility for data providers to receive payments directly for every time their data products are used by data consumers. This is due to the currently available payment methods that do not provide full privacy when performing transactions. A solution could be to rely on a centralised exchange with bank-level privacy on transactions or adopt one of the currently available privacy preserving cryptocurrencies such as Monero [62] and Zcash [35].

The LINDDUN analysis describes a set of threats related to the privacy of users. All of the threats do not regard the disclosure of the data providers datasets with the exception of one: "Unawareness of the data owner" 5.3. However, this threat is generated by risks related to the purpose for which a user decides to use the marketplace for. The purpose is the analytics computation on their data. The highest priority threats have very low probability and for each of them mitigations have been found. Moreover, the fact that the marketplace stores just the minimal amount of essential metadata, makes the threats even more mitigated.

The possible future work exposed by the paper [16] indicates the possibility of creating privacy preserving proofs for the executed computations. As an addition to that, the author of the thesis identified other points that could be improved in the current architecture. Even if the architecture exploits decentralisation on certain elements to ensure security and privacy, some features that are used also by the decentralised systems are centralised. One of them is the SSI Agent in the backend of the marketplace. SSI is a technology that exploits decentralisation to allow certified communication between two centralised parties. However in this case, users interact with a decentralised system. The consequence is that the MPC network and the blockchain need to trust the SSI agent for users certification.

Another centralisation point of the architecture regards the communication between blockchain and MPC network. The two are currently communicating through the backend that intermediates every message exchanged between them. This implies that the MPC network needs to trust the backend on any information claimed to be coming from the blockchain and the other way around.

To realise a sufficiently decentralised system, the architecture could be modified by joining the nodes of the MPC network and the nodes of the blockchain. In this way, every MPC node would be able to consult directly the ledger for any information needed from the blockchain. Moreover, with an agent in every node, every node of the network would be independent in verifying verifiable credentials not needing to trust anymore a single SSI agent.

References

[1] Garmin Ltd. “connect: Fitness at your fingertips”. (accessed: 23.11.2021), [Online]. Available: `https://connect.garmin.com/`.

[2] Apple-Inc. “A more personal Health app. For a more informed you”. (accessed: 23.11.2021), [Online]. Available: `https://www.apple.com/ios/health/`.

[3] Ōura Health Ltd. “Oura Ring”. (accessed: 16.08.2021), [Online]. Available: `https://ouraring.com/`.

[4] A. Bruni, L. Helminger, D. Kales, C. Rechberger, and R. Walch, “Privately Connecting Mobility to Infectious Diseases via Applied Cryptography”, *IACR Cryptology ePrint Archive*, vol. 2020, p. 522, 2020.

[5] D. Muoio. “Google mobilizes location tracking data to - help public health experts monitor COVID-19 spread (2020)”. (accessed: 25.04.2022), [Online]. Available: `https://www.mobihealthnews.com/news/google-mobilizes-location-tracking-data-help-public-health-experts-monitor-covid-19-spread`.

[6] GRAIL. “Grail”. (accessed: 23.11.2021), [Online]. Available: `https://grail.com/`.

[7] C. Todd, P. Salvetti, K. Naylor, and M. Albatat, “Towards non-invasive extraction and determination of blood glucose levels”, *Bioengineering*, vol. 4, no. 4, p. 82, 2017.

[8] D. Muio. "Fitbit launches large-scale health study to detect a-fib via heart rate sensors, algorithm". (accessed: 23.11.2021), [Online]. Available: https://www.mobihealthnews.com/news/fitbit-launches-large-scale-consumer-health-study-detect-fib-heart-rate-sensors-algorithm.

[9] M. Allen. "Health Insurers Are Vacuuming Up Details About You - And It Could Raise Your Rates". (accessed: 26.03.2022), [Online]. Available: https://www.propublica.org/article/health-insurers-are-vacuuming-up-details-about-you-and-it-could-raise-your-rates.

[10] Medicalchain. "Medicalchain: Whitepaper 2.1". (accessed: 25.03.2022), [Online]. Available: https://medicalchain.com/Medicalchain-Whitepaper-EN.pdf.

[11] E. Morley-Fletcher, "MHMD: my health, my data", in *EDBT/ICDT Workshops*, ser. CEUR Workshop Proceedings, vol. 1810, CEUR-WS.org, 2017.

[12] Enveil. "Enveil: Encrypted Veil". (accessed: 26.03.2022), [Online]. Available: https://www.enveil.com/.

[13] D. Fernandez, A. Futoransky, G. Ajzenman, M. Travizano, and C. Sarraute, *Wibson protocol for secure data exchange and batch payments*, 2020. arXiv: 2001.08832.

[14] V. Koutsos, D. Papadopoulos, D. Chatzopoulos, S. Tarkoma, and P. Hui, "Agora: A privacy-aware data marketplace", *IACR Cryptology ePrint Archive*, vol. 2020, p. 865, 2020.

[15] KRAKEN Consortium. "The Project | KRAKEN". (accessed: 16.10.2021), [Online]. Available: https://www.krakenh2020.eu/the_project/overview.

[16] K. Koch, S. Krenn, D. Pellegrino, and S. Ramacher, "Privacy-preserving analytics for data markets using MPC", in *Privacy and Identity Management*,

M. Friedewald, S. Schiffner, and S. Krenn, Eds., Cham: Springer International Publishing, 2021, pp. 226–246.

[17] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, “A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements”, *Requirements Engineering*, vol. 16, no. 1, pp. 3–32, 2011.

[18] imec-DistriNet Research Group. “LINDDUN privacy engineering”. (accessed: 26.03.2022), [Online]. Available: `https://www.linddun.org/`.

[19] D. Chaum and E. van Heyst, “Group signatures”, in *EUROCRYPT*, ser. LNCS, vol. 547, Springer, 1991, pp. 257–265.

[20] M. Bellare, D. Micciancio, and B. Warinschi, “Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions”, in *Advances in Cryptology — EUROCRYPT 2003*, E. Biham, Ed., Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2003, pp. 614–629.

[21] H. Kim, Y. Lee, M. Abdalla, and J. H. Park, “Practical dynamic group signature with efficient concurrent joins and batch verifications”, *IACR Cryptology ePrint Archive*, vol. 2020, p. 921, 2020.

[22] A. Eldhose and T. Sukumar, “Dynamic privacy protecting short group signature scheme”, *International Journal on Cybernetics & Informatics*, vol. 5, no. 2, pp. 147–154, 2016. DOI: `10.5121/ijci.2016.5216`.

[23] M. Manulis. “Group signatures: Authentication with privacy”. (accessed: 07.01.2022), [Online]. Available: `https://nilsfleischhacker.de/publication/group-signatures-authentication-with-privacy/`.

[24] Orbs. “Fully Distributed Group Signatures”. (accessed: 18.01.2022), [Online]. Available: `https://www.orbs.com/wp-content/uploads/2019/04/Crypto_Group_signatures-2.pdf`.

[25] G. Ateniese and G. Tsudik, "Some open issues and new directions in group signatures", *Financial Cryptography*, pp. 196–211, 1999. DOI: 10.1007/3-540-48390-x_15.

[26] K. Sakurai and S. Miyazaki, "An anonymous electronic bidding protocol based on a new convertible group signature scheme", *Information Security and Privacy*, pp. 385–399, 2000. DOI: 10.1007/10718964_32.

[27] E. Brickell and J. Li, "Enhanced privacy ID from bilinear pairing for hardware authentication and attestation", in *SocialCom/PASSAT*, IEEE, 2010, pp. 768–775.

[28] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems (extended abstract)", in *STOC*, ACM, 1985, pp. 291–304.

[29] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation", in *EUROCRYPT*, ser. LNCS, vol. 2045, Springer, 2001, pp. 93–118.

[30] D. Chaum, "Blind signatures for untraceable payments", in *CRYPTO*, Plenum Press, New York, 1982, pp. 199–203.

[31] Google. "Google Fit: Coaching you to a healthier and more active life". (accessed: 23.11.2021), [Online]. Available: https://www.google.com/fit/.

[32] Princeton University. "PPPL and Princeton demonstrate novel technique that may have applicability to future nuclear disarmament talks ". (accessed: 03.09.2021), [Online]. Available: https://research.princeton.edu/news/pppl-and-princeton-demonstrate-novel-technique-may-have-applicability-future-nuclear.

[33] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer, “From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again”, in *ITCS*, ACM, 2012, pp. 326–349.

[34] E. C. Co. “Zcash technology”. (accessed: 03.09.2021), [Online]. Available: `https://z.cash/technology/zksnarks/`.

[35] E. B. Sasson, A. Chiesa, C. Garman, *et al.*, “Zerocash: Decentralized anonymous payments from bitcoin”, in *2014 IEEE Symposium on Security and Privacy (SP)*, Los Alamitos, CA, USA: IEEE Computer Society, (accessed: 23.11.2021), pp. 459–474. DOI: `10.1109/SP.2014.36`. [Online]. Available: `https://doi.ieeecomputersociety.org/10.1109/SP.2014.36`.

[36] A. C. Yao, “Protocols for secure computations (extended abstract)”, in *FOCS*, IEEE, 1982, pp. 160–164.

[37] C. Zhao, S. Zhao, M. Zhao, *et al.*, “Secure multi-party computation: Theory, practice and applications”, *Information Sciences*, vol. 476, pp. 357–372, 2019, ISSN: 0020-0255. DOI: `https://doi.org/10.1016/j.ins.2018.10.024`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0020025518308338`.

[38] R. Canetti, *Security and composition of multiparty cryptographic protocols*, Apr. 2000. [Online]. Available: `https://link.springer.com/article/10.1007/s001459910006`.

[39] R. Tso, Z.-Y. Liu, and J.-H. Hsiao, “Distributed e-voting and e-bidding systems based on smart contract”, *Electronics*, vol. 8, no. 4, 2019, ISSN: 2079-9292. DOI: `10.3390/electronics8040422`. [Online]. Available: `https://www.mdpi.com/2079-9292/8/4/422`.

[40] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges", in *Theory of Cryptography*, Y. Ishai, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 253–273.

[41] I. Chillotti. "The Three Musketeers of Secure Computation: MPC, FHE and FE". (accessed: 28.09.2021), [Online]. Available: `https://www.esat.kuleuven.be/cosic/blog/the-three-musketeers-of-secure-computation-mpc-fhe-and-fe/`.

[42] S. Haber and W. S. Stornetta, "How to time-stamp a digital document", *Journal of Cryptology*, vol. 3, pp. 99–111, 1991.

[43] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, `https://bitcoin.org/bitcoin.pdf`.

[44] M. Pilkington, *Blockchain technology: Principles and applications*, Sep. 2015. [Online]. Available: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2662660`.

[45] L. L. C. S. Laboratory, L. Lamport, C. S. Laboratory, *et al.*, *The byzantine generals problem*, Jul. 1982. [Online]. Available: `https://dl.acm.org/doi/10.1145/357172.357176`.

[46] N. Szabo, "Formalizing and securing relationships on public networks", *First Monday*, vol. 2, no. 9, Sep. 1997. DOI: `10.5210/fm.v2i9.548`. [Online]. Available: `https://firstmonday.org/ojs/index.php/fm/article/view/548`.

[47] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger", 2014. [Online]. Available: `https://ethereum.github.io/yellowpaper/paper.pdf`, (accessed: 28.09.2021).

[48] E. Androulaki, A. Barger, V. Bortnikov, *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains", in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18, Porto, Portugal: Association for Computing Machinery, 2018. DOI: 10.1145/3190508.3190538. [Online]. Available: https://doi.org/10.1145/3190508.3190538.

[49] Hyperledger Foundation. "An Introduction to Hyperledger". (accessed: 03.09.2021), [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf.

[50] Oaktree Technologies. "Hyperledger overview". (accessed: 29.03.2022), [Online]. Available: https://oak-tree.tech/blog/hyperledger-overview.

[51] Hyperledger Foundation. "What's new in Hyperledger Fabric v2.x". (accessed: 29.03.2022), [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatsnew.html.

[52] I. Duits. "The way towards self-sovereign identity". (accessed: 16.10.2021), [Online]. Available: https://innovalor.nl/en/Blogs/self-sovereign-identity.

[53] C. Allen. "The Path to Self-Sovereign Identity". (accessed: 16.10.2021), [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

[54] V. Gerard. "Designing the future identity: Authentication and authorization through self-sovereign identity". (Aug. accessed: 20.01.2022), [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid:200f1df0-adda-47a1-894c-baf54133035a?collection=education.

[55] D. H. Hardman. "Vc triangle of trust". (accessed: 03.09.2021), [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/5/51/VC_triangle_of_Trust.svg.

[56] N. Shevchenko. “Evaluating Threat-Modeling Methods for Cyber-Physical Systems ”. (accessed: 19.12.2021), [Online]. Available: https://insights.sei.cmu.edu/sei_blog/2019/02/evaluating-threat-modeling-methods-for-cyber-physical-systems.html.

[57] A. Shostack. “STRIDE chart”. (accessed: 29.03.2022), [Online]. Available: https://www.microsoft.com/security/blog/2007/09/11/stride-chart/.

[58] J. L. Jonathan Hunt. “Threat Modeling Within GitLab”. (accessed: 04.09.2021), [Online]. Available: https://about.gitlab.com/handbook/security/threat_modeling/.

[59] iMinds-DistriNet. “LINDDUN tutorial”. (accessed: 09.11.2021), [Online]. Available: https://7e71aeba-b883-4889-aee9-a3064f8be401.filesusr.com/ugd/cc602e_f98d9a92e4804e6a9631104c02261e1f.pdf.

[60] R. Dingledine, N. Mathewson, and P. F. Syverson, “Tor: The second-generation onion router”, in *USENIX*, USENIX, 2004, pp. 303–320.

[61] I. Wagner and D. Eckhoff, “Technical privacy metrics: A systematic survey”, *ACM Computing Surveys*, vol. 51, no. 3, 57:1–57:38, 2018.

[62] S. Noether and A. Mackenzie, “Ring confidential transactions”, *Ledger*, vol. 1, pp. 1–18, 2016.