



HAL
open science

Privacy-Preserving Analytics for Data Markets Using MPC

Karl Koch, Stephan Krenn, Donato Pellegrino, Sebastian Ramacher

► **To cite this version:**

Karl Koch, Stephan Krenn, Donato Pellegrino, Sebastian Ramacher. Privacy-Preserving Analytics for Data Markets Using MPC. 15th IFIP International Summer School on Privacy and Identity Management (Privacy and Identity), Sep 2020, Maribor, Slovenia. pp.226-246, 10.1007/978-3-030-72465-8_13. hal-03703757

HAL Id: hal-03703757

<https://inria.hal.science/hal-03703757v1>

Submitted on 24 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Privacy-preserving Analytics for Data Markets using MPC*

Karl Koch¹, Stephan Krenn²[0000–0003–2835–9093],
Donato Pellegrino³, and Sebastian Ramacher²[0000–0003–1957–3725]

¹ Graz University of Technology, Graz, Austria
`karl.koch@iaik.tugraz.at`

² AIT Austrian Institute of Technology, Vienna, Austria
`{stephan.krenn,sebastian.ramacher}@ait.ac.at`

³ TX Tomorrow Explored, Helsinki, Finland
`donato@tx.company`

Abstract Data markets have the potential to foster new data-driven applications and help growing data-driven businesses. When building and deploying such markets in practice, regulations such as the European Union’s General Data Protection Regulation (GDPR) impose constraints and restrictions on these markets especially when dealing with personal or privacy-sensitive data.

In this paper, we present a candidate architecture for a privacy-preserving personal data market, relying on cryptographic primitives such as multi-party computation (MPC) capable of performing privacy-preserving computations on the data. Besides specifying the architecture of such a data market, we also present a privacy-risk analysis of the market following the LINDDUN methodology.

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

1 Introduction

For the last decades, the amount of data generated, processed, and shared has been ever-increasing [29]. Especially personal data has become more and more interesting [12]. One of the trends in this area is fitness and health data as more and more people are using fitness trackers, e.g. Garmin’s connect [21] or Apple’s Health app [4], where the collected data can then be used in clinical research [33]. Relatedly, machine learning-based approaches facilitate the development of small sensors for measuring bodily functions of chronically ill patients. As an example, diabetes patients needing to draw blood multiple times a day, benefit from novel, noninvasive monitoring methods of their blood sugar levels [37]. Very recently, location data collected by mobile network operators have become of interest to help combat the COVID-19 pandemic [9, 34].

* This project leading to this publication has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 871473 (“KRAKEN”).

However, personal data is highly sensitive and regulations such as the General Data Protection Regulation (GDPR) have to be taken into account when collecting, transmitting, storing, or processing the data. Therefore, these regulations present unique challenges in the design of data markets offering any kind of personally identifiable data. To profit from the opportunities as an individual (e.g., by sharing fitness data with insurance companies for better premiums) or for the common good (e.g., for limiting the outbreak of pandemics or clinical research), these challenges have to be overcome first, as otherwise misuse of the data could lead to discrimination, e.g., on the job market or by insurance companies [3].

More and more data markets are tackling these challenges via cryptographic means offering a wide variety of differing privacy guarantees. Mediacalchain [1] among many others facilitates the exchange of medical data end-to-end secured. Agora [26] goes a step further and offers a data market place for privacy-sensitive data built from functional encryption (FE) [7] where data consumers are able to buy evaluations of certain functions on user data. Thereby only the results of the evaluation are exchanged without the need to transfer the original data sets. Besides proposing the specific FE-based architecture, Koutsos et al. also provide a security model for confidentiality of processed data and consider payments in their analysis. However, the security model does not consider confidentiality of the data against the broker. Similarly, MyHealthMyData [31] offers FE-based analytics with the focus on medical data while providing confidentiality of the data throughout the system. Enveil [2] is a platform for outsourced computation using fully homomorphic encryption (FHE), but also offers possibilities for consumers to perform some analytic functions on the data. Wibson [20] provides a smart contract based market place focusing on different privacy aspects, namely the privacy of seller's and buyer's identity.

Brokerage and Market platform for personal data (KRAKEN). The H2020 project KRAKEN [27] develops a data market for privacy-sensitive data. To achieve that, the project “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” [27]. KRAKEN mainly builds upon three pillars: (i) a data market place, (ii) self-sovereign identity (SSI) [17], and (iii) a toolbox of cryptographic primitives for privacy-preserving computation to achieve that. The market place acts as a broker between data providers and consumers. SSI is used to manage authentication, authorization, and, e.g., key management between data consumer and producer. Privacy-preserving cryptographic protocols and primitives including secure multi-party computation (MPC) are used to enable privacy-preserving analytics.

1.1 Contribution

In this paper, we propose and analyse a candidate architecture for the KRAKEN personal data marketplace that provides privacy-preserving distributed analytics features through the usage of MPC. The KRAKEN platform ensures user privacy and security of the overall system by relying on the decentralization of

its core subsystems, SSI-based user management, and MPC-based processing of data. KRAKEN does not provide user data to the buyers. The core goal is to link buyers and sellers on the basis of metadata and policies, and enable data transfer between them in a privacy-preserving and decentralized manner. Thereby, KRAKEN closes the gap left open in Agora. Our contribution is twofold:

Architecture. We describe a candidate architecture of the KRAKEN platform in detail, thereby explaining the necessary cryptographic background, suggesting instantiations of the building blocks to be used, and justifying any necessary design choices. The platform is designed in a way that allows for computations over inputs from potentially many different data sources in a single computation to allow for, e.g., statistics over many users.

Privacy analysis. To validate the privacy requirements of the architecture, data flow diagrams and a privacy analysis based on LINDDUN [16] are presented. This analysis considers all the privacy goals of KRAKEN, defines the related threats, and proposes mitigation strategies.

2 Preliminaries

The following sections give the necessary background on cryptographic building blocks, self-sovereign identities, and the LINDDUN methodology.

2.1 Cryptographic Building Blocks

Besides standard primitives such as encryption, our cryptographic architecture relies on a set of advanced privacy-preserving cryptographic mechanisms, which we will briefly introduce in the following. We want to stress that practically efficient solutions and instantiations are available for each of these building blocks.

Group signatures. Group signatures, initially introduced by Chaum and van Heyst [14] allow a party to sign a message on behalf of a group. That is, the verifier receives cryptographic guarantees that a member of the group indeed signed the message, yet he does not learn the identity of the actual signer. To achieve this goal, a *group manager* generates a group public key gpk as well as a master secret key msk . Now, when a user U joins the group, she engages in a protocol with the group manager to receive her secret key sk_U which she uses for signing messages, while the verifier only needs access to gpk to verify the validity of signatures.

It is worth noting that in group signatures, signer privacy is not absolute: in order to avoid abuse, a dedicated *inspector* holding a inspection secret key isk is able to revoke anonymity and reveal the originator of a signature. For the remainder of this paper we assume that all inspector public keys are generated in a way that no entity knows the corresponding secret key (e.g., by setting the public key to the hash value of a public nonce), as the inspection feature is not required in our scenario. The resulting primitive is then akin to Intel’s Enhanced Privacy ID (EPID) scheme [8], which also allows for signing messages on behalf

of a group without anonymity revocation functionality. Recently, Kim et al. [25] proposed the first group signature scheme supporting batch verifications, which significantly speeds up the verification process in case of many signatures.

Zero-knowledge proofs of knowledge. A zero-knowledge proof of knowledge (ZK-PoK) is a two party protocol between a *prover* and a *verifier*, which achieves two intuitively contradictory goals: it allows the prover to convince the verifier that she knows a secret piece of information, while at the same time revealing no further information than what is already revealed by the claim itself.

Such protocols were first introduced by Goldwasser et al. [22], and are a central building block for many privacy-preserving applications such as anonymous credential systems [11, 13], voting schemes [23], e-cash [10], or group signatures. In recent years, zero-knowledge succinct non-interactive arguments of knowledge (SNARKs) [5] achieving very high efficiency have gained significant attention.

Secure multi-party computation. Since its introduction by Yao [39], secure multi-party computation (MPC) has developed to an important building block for a variety of privacy-preserving applications. It allows a group of nodes to jointly evaluate a function on their inputs without revealing the inputs to the other nodes or any trusted third entity. Two major branches of MPC exist: while techniques based on garbled circuits are more efficient for bitwise operations [39], integer arithmetic can be computed highly efficiently using secret sharing based mechanisms [6, 36] due to the algebraic properties of these schemes. Especially during the last decade, research has come up with practically efficient protocols which have also been deployed in real-world scenarios and products [15, 19, 24].

2.2 Self-Sovereign Identity

The Self-Sovereign Identity (SSI) [17, 32] model describes an identity management concept which grants the owners of digital identities complete control over their data. The goal of SSIs are ensuring the security and privacy of users' identity data, full portability of the data, no central authorities, and data integrity.

2.3 LINDDUN Methodology

LINDDUN [16] is a threat modeling methodology for systematically analyzing privacy threats in software architectures. 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. The analysis consists of the following, cf. also Figure 1:

- (1) First, a data-flow diagram (DFD) is created detailing all involved entities, processes, trust boundaries, data stores and data flows.
- (2) In the second step, the DFDs are mapped to the threat categories and for each element of the DFD potential threat categories are identified.

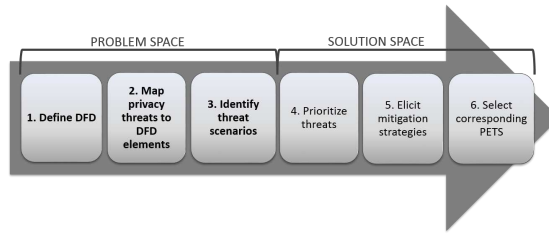


Figure 1. Overview of the LINDDUN methodology [28].

- (3) Third, the identified threats are refined and documented. Also, assumptions that are made in the architecture are documented.
- (4) Fourth, the threats are prioritized based on their risk.
- (5) Next, mitigation strategies are defined, taking into account the risks that have been associated to each threat.
- (6) Finally, effective countermeasures are selected by mapping the defined mitigation strategies to suitable privacy-enhancing technologies.

3 KRAKEN Architecture

As discussed by Koutsos et al. [26], a data market has to at least satisfy data privacy and output verifiability which are defined as follows:

Data privacy: No party can learn any information about on the data of the data owners. Only the result of computations on the data is known to the data consumers.⁴

Output verifiability: Authenticity of the data and results has to be ensured, i.e., falsified or incorrect results cannot be sold to a consumer.

Note that in contrast to Koutsos et al. [26] we omit *atomicity of payments*, which requires that data owners are correctly reimbursed, as for simplicity we do not (yet) consider per-access payments at this stage, but assume that data owners are compensated through a lump sum when uploading their data. However, we consider it important that data owners stay in control of their data, and therefore request that data owners need to be able to define for which (types of) computations their data may be used, and which computations are considered to privacy-invasive by the data owner.

We will discuss one a candidate architecture of the KRAKEN market place. The core idea of this architecture is centered around the use of MPC for privacy-preserving computation on data, group signatures to ensure data authenticity while preserving anonymity, and SNARKs for output verifiability. In our architecture we consider the following actors:

⁴ Note that in contrast to our definition, Agora allows data brokers, i.e. the market place, to learn the results as well.

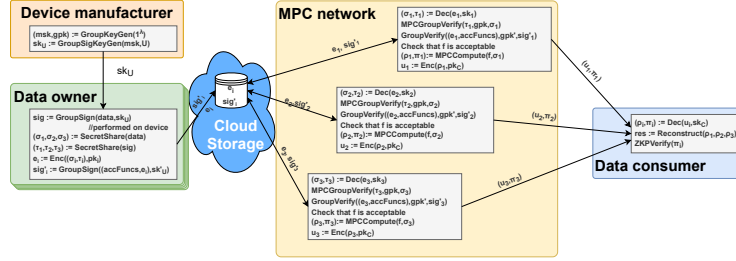


Figure 2. KRAKEN cryptographic architecture overview.

Device manufacturers produce devices that collect sensitive data. All devices contain a signing key for a group signature scheme where the group is managed by the manufacturer. Devices sign the data using their key.

Data owners use devices from the device manufacturer to collect sensitive data. The owner defines a family of functions which are allowed for performing computation on the data.

Data consumers define the functions that should be used for the analysis of the data. They are in possession of a public-key encryption key.

Computation nodes perform the analysis as defined by the data consumer on the data of the data owner using a secret sharing-based MPC protocol. They are in possession of a public-key encryption key. These computation nodes can be run by cloud computing providers that offer “computation as a service”, various participants of the market place including the data owners and consumers, or by operator of the market place.

KRAKEN market place handles the registration of data owners and consumers and manages listings of available data sets. The information is stored on an internal blockchain and database.

Cloud storage provider offers storage to data owners without the need for registration.

An overview of the data processing is depicted in Figure 2. We will now discuss the three typical data flows.

User registration. We start with the registration of a data owner and data consumers on the market place which is centered around credentials from a SSI system. In this step we assume that they obtained their credential from an identity provider before, as this is a step only performed once independent of the registration at the market place.

- (1) The user is in possession of SSI credentials and uses them to create an account on the data market place.
- (2) The user and the market place perform the group signature joining procedure. At the end of the interaction, the user obtains a group signing key sk'_U that she may use to sign policies specifying the types of computations that may be performed on her data.

Data pre-processing and registration. After a data owner registered on the market place, she is able to register her data for subsequent analysis in the market place. To do so, the data owner performs the following steps:

- (1) The data owner collects data produced by her device, which is signed also by the device's group signature signing key sk_U .
- (2) The data owner produces shares of the data and associated signature, using a secret sharing scheme compatible with the deployed MPC protocol, resulting in shares σ_i , $i = 1, 2, 3$ – one per MPC node.
- (3) Using the public key pk_i of MPC node i , the data owner encrypts σ_i .
- (4) The data owner signs the encrypted shares and an acceptable family of functions using their group signature key sk'_U .
- (5) The data owner sends all information, i.e. encrypted shares, acceptable family of functions, and the signature under sk'_U , to a cloud storage provider of their choice.
- (6) The data owner registers the offering on the market place and informs the market place on the location of the encrypted secret shares.

For simplicity, in the proposed architecture, we assume that data owners are reimbursed by the market place via a lump sum when registering their data, and no further payments will take place on a per-usage basis.

Analysis. A consumer uses the market place to find data sets that are of interest and negotiates their use via the market place. During the negotiation, the consumer declares the function to be evaluated on the data. Once an agreement is reached, the evaluation is performed as follows:

- (1) The market place informs the computation nodes of the function to perform and the location of the data items.
- (2) The computation nodes fetch the encrypted secret shares and signatures from the cloud storage.
- (3) After receiving all encrypted shares and signatures and the function f , the computation nodes first verify, for each data item, the signature on the data evaluation policy, and that f is an eligible function with respect to this policy. They then decrypt the shares, jointly verify the shared signatures, and start the MPC protocol to compute f on the data.
- (4) The shares obtained as result of the computation, are then encrypted with respect to the consumer's public key, pk_C .
- (5) The nodes provide a ZK-PoK/SNARK that they computed the function f on the received data and that the obtained signature verified on the inputs.
- (6) The nodes send the encrypted results and the proof to the consumer.
- (7) The consumer decrypts the shares of the result and combines them to obtain the result of the evaluation. The consumer also verifies its correctness by checking the proofs sent by the nodes.

Similarly to before, in the current architecture we assume that consumers pay the market place, but no (direct or indirect) payment from the consumer to the data

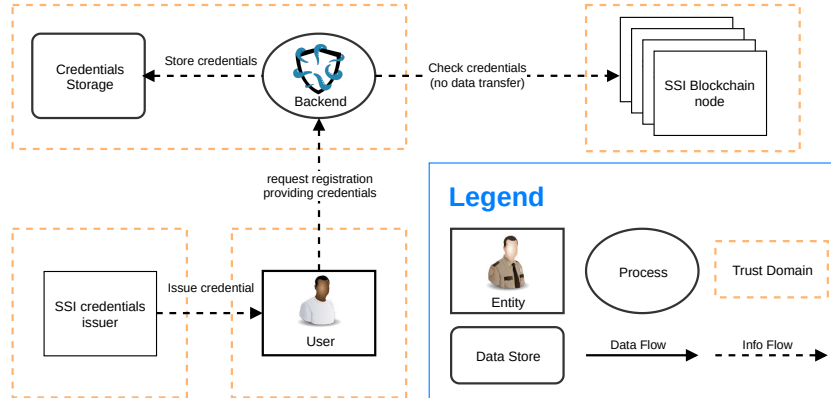


Figure 3. DFD for the user action of registering.

owner takes place. Further suggestions for a more fine-granular reimbursement concept can be found in Section 5.

4 LINDDUN Analysis of KRAKEN

We now present the LINDDUN analysis for KRAKEN’s architecture. In the analysis, we focus on three user actions: (1) Register user, (2) Register availability of data, and (3) Perform data analysis.

We start with the DFDs. For reasons of clarity and comprehensibility, we split the data flow into two categories. First, a *real data flow*, which contains (parts of) personal data from data owners; such as encrypted shares or an analytics result. Second, an *info flow*, for other types of data flow; such as registering availability of data or invoicing data analysis. Figure 3 visualizes the DFD for the user action of registering a user. Figure 4 visualizes the DFD for the user actions of registering availability of data and performing data analysis.

4.1 Threat Tables

We use the LINDDUN mapping template [16] to map the DFD elements to the seven threat categories: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance. In the LINDDUN mapping template, entities are affected only by linkability, identifiability and unawareness threats, while the other DFD elements are affected by every threat except unawareness. We consider processes, flows and entities internal to trust domains not affected by any privacy threat from outside due to the fourth assumption. Data Flows and Info Flows are not affected by Non-compliance threat as the communications are secured with TLS. The Non-compliance threat does not affect Data Stores as well as they are implemented with the data-minimization principle. Also processes are not affected by the Non-compliance threat as no

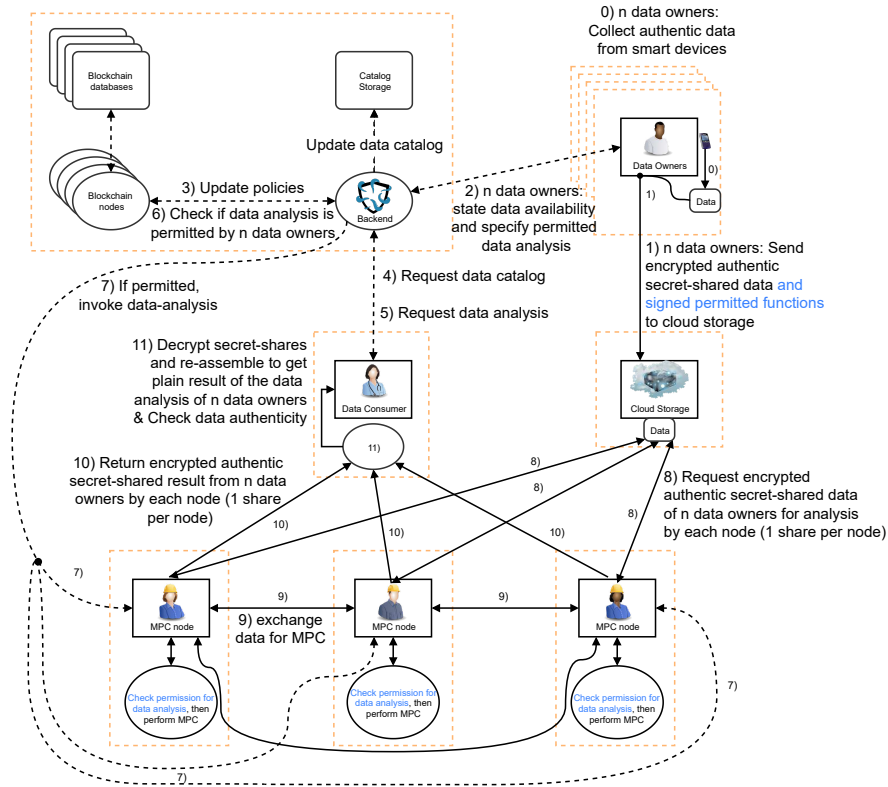


Figure 4. DFD for the user actions of registering data and performing data analysis. The legend is as in Fig. 3.

sensitive information (user personal data) is handled by them; in the case of MPC nodes the second assumption rules out the privacy threat.

Table 1 shows the threat table of our 1st user action, performing user registration. Table 2 shows the threat table of our 2nd user action, performing registration of data availability. Table 3 shows the threat table of our 3rd user action, performing data analysis.

4.2 Threat Elicitation

After having identified the elements of the DFD susceptible to privacy threats, the next step is to describe them in detail. In our scenario we make four assumptions, which we present and briefly discuss in the following.

Assumption 1. The SSI’s credential issuer is a trusted entity. If the KRAKEN backend colludes with the credential issuer, the KRAKEN backend gets to know not only the SSI identity, but also the users’ real identity. The

Table 1. LINDDUN’s threat table of the 1st user action, performing user registration. An **X** 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	Credentials Storage	X	X					
Info Flow	Issue credentials	X	X	X	X	A3		
	Request Registration	X	X	X	X	A3		
	Check credentials	X	X	X	X	A3		
	Store credentials							
Process	Backend							
Entity	User							
	SSI credentials issuer	A1	A1	A1	A1	A1	A1	A1
	SSI Blockchain							

assumption of an honest SSI’s credential issuer, gives us the guarantee that the KRAKEN backend cannot link SSI identities to real identities.

While this assumption might look overly strong at first glance, it could be achieved by designing the issuer as a distributed party deploying threshold cryptography, or, on a policy-level, through regular audits. Furthermore, by informing users about this requirements, they can also check for, e.g., legal relationships between these two entities, before revealing any personal data.

Assumption 2. At least one MPC node is honest. For MPC, we can use protocols which give security guarantees although all parties bar one are malicious. We refer to these kind of protocols as *fully-malicious protocols*. Thus, with this assumption and the respective protocols, no MPC node either gets to know the underlying data, nor the analysis result. Furthermore, due to this assumption and the provided policies by the data owners, we do not need to worry about the case, that MPC nodes compute without authorization, since the honest MPC node aborts the computation, which leads to a global termination of the computation.

To minimize the impact of this assumption, the hosts of the MPC nodes should be carefully selected in order to minimize intended collusions among them or their simultaneous corruptions. A possible approach might be to include more than the required three computation nodes in the ecosystem, and letting the user choose which nodes to support when encrypting her data. Note that a selected node can still get corrupted, but it might be less likely. Though, after all, (1) the probability that all nodes get corrupted might decrease too, and (2) all nodes need to get corrupted to leak sensitive data when using a fully-malicious protocol.

Especially for highly sensitive data, one of these nodes might even be deployed at the data owners. Thereby, the honest node is in control of the data owner and hence leaking sensitive data is low.

Table 2. LINDDUN’s threat table of the 2nd user action, performing registration of data availability. An **X** 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	X	X						
	Cloud Storage	X	X						
Info Flow	collect authentic data from smart devices								
	send encrypted authentic secret-shared data to cloud storage and signed permitted functions	X	X	X	X	A3			
	state data availability and specify permitted data analysis	X	X	X	X	A3			
	Update data catalog								
	Update policies								
Process	Backend								
	Blockchain nodes								
Entity	Data Owner							X	
	Data Consumer								
	Cloud Storage								

Assumption 3. Each communication between actors of different trust domains is secured using transport-layer security (TLS). When two actors of different trust domains communicate, this assumption guarantees that no content is leaked during transit. Only the communication’s metadata can still leak, like the IP addresses of the source and target. Secure communication protocols are heavily deployed in many scenarios, and can be considered state-of-the-art.

Assumption 4. Anybody who gains access to any of trust boundaries is considered to have the same possibilities as the corrupted entity itself, and trust boundaries are implemented in a secure way. If someone hacks into a any of trust domains, the intruder (normally) gets access to the corresponding data stores, processes, and data flow. Hence the intruder has full control over the trust domain’s system.

To a major extent, this is rather a simplification than an assumption, as it significantly simplifies the analysis. We do not need to distinguish between insider attacks (e.g., a system administrator), and an external attack (e.g., an attacker partial gaining control over an entity), but we assume that once a trust boundary is violated, the entire entity is fully controlled by the adversary.

Mapping LINDDUN’s Privacy Threats to the DFDs. In this section the threats identified using the threat tables are described in detail.

Threat 1 (Linkability in one or more storages). An insider of KRAKEN links data coming from the catalog, credentials, policies or purchases storages.

Table 3. LINDDUN’s threat table of the 3rd user action, performing data analysis. An **X** 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	X	X						
	Cloud Storage			X	X				
	Data Consumer								
Info Flow	4) Request data catalog	X	X	X	X	A3			
	5) Request data analysis	X	X	X	X	A3			
	7) Invoke data analysis	X	X	X	X	A3			
Data Flow	8) Request enc. auth. se-sha. data	X	X	X	X	A3			
	9) Exchange se-sha. data	X	X	X	X	A3			
	10) Return enc. analysis result	X	X	X	X	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								

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 requests.

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 consumer 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 doesn't 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).

Table 4. Threat prioritization depending on likelihood and impact.

Likelihood	Impact	Priority	Likelihood	Impact	Priority	Likelihood	Impact	Priority
low	low		low	high		medium	high	
low	medium	low	medium	medium	medium	high	medium	high
medium	low		high	low		high	high	

Table 5. Overview of threat prioritization. Threats that are not effective due to our assumptions are not included in the table.

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

4.3 Prioritizing Threats

For the prioritization of the threats, first a likelihood and impact value is assigned to every threat identified in the threat table. Both values are taken from low, medium, and high indicating a low to high likelihood and impact, respectively. The likelihood value depends on the joint evaluation of difficulty and outcome of performing the specific action, while the impact value depends on threatened assets where identifiability and disclosure of information are high impact, linkability is medium and Non-repudiation, Detectability, Unawareness, Non-compliance are low. Table 4 shows how threats are prioritized depending on likelihood and impact values.

Table 5 gives an overview of the prioritization of the identified threats. In the following, we give a brief justification for each 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.

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.

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.

4.4 Mitigating Threats

For every threat in non low priority, we propose a set of mitigations expressed in the following list:

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 leads to identifiability. To avoid the misactor to understand that the communication is happening with KRAKEN, avoiding linkability and resulting identifiability, we propose onion routing like Tor [18].

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 [38] 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.

4.5 Privacy Analysis Outcome

We adopted an iterative approach in the design of the architecture that used the LINDDUN privacy analysis to identify threats and plan the changes for the iterations. It’s worth mentioning the most relevant changes that this approach generated. The DFDs (Figures 3 and 4) highlight the differences in information and data flow. This division in typology of data flows has been key leading us to construct an architecture where personal data is exchanged solely between data owner and consumer, without passing through centralized parties unencrypted.

Another key element derived from the analysis is the use of group signatures. Public keys of the users represent a risk for identification of the user or could be linked to other actions. For this reason we decided to adopt group signatures

to allow the user to sign data and permitted functions on behalf of a group. In this way the user can demonstrate to be part of the users of KRAKEN and can sign data and functions anonymously while retaining authenticity.

Finally, we identified a set of threats that do not imply architectural changes, but instead have to be considered in a development context. These threats and their mitigations affect single elements of the architecture: the Credentials storage, Catalog storage, Policies storage, Purchases storage, Cloud storage, and the Data owner. A set of changes need to be implemented in these elements to address the mitigations. In particular, we applied a principle of data minimization in the context of the backend and the blockchain storages, while in the user software development, we considered a set of tools to be provided to the Data owner for documentation, analytics, and threat detection purposes.

5 Conclusion and Future Work

In this paper, we presented a privacy-preserving data market platform and analysed its privacy-guarantees following the LINDDUN methodology. The proposed solution allows users to sell data without any disclosure of information in regards to the data itself. The LINDDUN analysis revealed threats related to linkability and disclosure of information that could have a relevant impact, however these threats have a low likelihood and the system’s methodologies in collecting information related to those threats is implemented in a way that highly minimizes the collected information. The LINDDUN analysis does not reveal any threat related to disclosure of information of the owners data sets.

The proposed architecture is based on cryptographic mechanisms, and in particular secure multi-party computation (MPC). With that approach, a data consumer is able to obtain privacy-preserving data analysis results from data owners, while the consumer receives only the analysis result. Furthermore, our marketplace does neither learn the owners’ data content nor the analysis result, but only metadata,. As opposed to Agora [26], where the broker, their marketplace, gets to know the analysis result. Our security guarantees in terms of privacy analysis, however, depend on the assumption that at least one MPC node behaves honestly. A possible future work would be on realising a trust measurement to drive the choice of MPC nodes. Another possible field of research would go towards moving the MPC computations on data owners. The obstacle to overcome in this case would be the problem of user availability during the computation, as users typically do not have constantly running servers available.

The architecture (cf. Section 3) only considers lump sums to reimburse the data owner. However, it might be practically more preferable to get paid *per usage*, i.e., whenever one’s data is actually used in a computation, resulting in additional privacy challenges. A straightforward solution might be to add an exchange service. This service would then be able to link usages of a user’s data, thereby being able to profile a user, especially when collaborating with a data consumer. Thus, such a service would need to be highly trusted akin to traditional banks in a physical world. An alternative approach could be to

leverage privacy-friendly crypto currencies like Monero [35] or z.cash [30]. To further increase trust in the system, the MPC nodes could publish cryptographic yet privacy-preserving proofs which data was used for which computation, such that a user could audit that she was indeed paid for every computation involving her data. The precise format of such auxiliary outputs of the MPC nodes is currently being investigated.

References

1. Medicalchain: Whitepaper 2.1 (2018), <https://medicalchain.com/Medicalchain-Whitepaper-EN.pdf>
2. Enveil: Encrypted Veil (2020), <https://www.enveil.com/>
3. Allen, M.: Health Insurers Are Vacuuming Up Details About You - And It Could Raise Your Rates (2020), <https://www.propublica.org/article/health-insurers-are-vacuuming-up-details-about-you-and-it-could-raise-your-rates>
4. Apple-Inc.: A more personal Health app. For a more informed you (2020), <https://www.apple.com/ios/health/>
5. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: ITCS. pp. 326–349. ACM (2012)
6. Bogdanov, D., Niitsoo, M., Toft, T., Willemson, J.: High-performance secure multi-party computation for data mining applications. *Int. J. Inf. Sec.* **11**(6), 403–418 (2012)
7. Boneh, D., Sahai, A., Waters, B.: Functional encryption: a new vision for public-key cryptography. *Commun. ACM* **55**(11), 56–64 (2012)
8. Brickell, E., Li, J.: Enhanced privacy ID from bilinear pairing for hardware authentication and attestation. In: SocialCom/PASSAT. pp. 768–775. IEEE (2010)
9. Bruni, A., Helminger, L., Kales, D., Rechberger, C., Walch, R.: Privately Connecting Mobility to Infectious Diseases via Applied Cryptography. *IACR Cryptol. ePrint Arch.* **2020**, 522 (2020)
10. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Balancing accountability and privacy using e-cash (extended abstract). In: SCN. LNCS, vol. 4116, pp. 141–155. Springer (2006)
11. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: EUROCRYPT. LNCS, vol. 2045, pp. 93–118. Springer (2001)
12. Chandler, S.: We're giving away more personal data than ever, despite growing risks (2020), <https://venturebeat.com/2019/02/24/were-giving-away-more-personal-data-than-ever-despite-growing-risks/>
13. Chaum, D.: Blind signatures for untraceable payments. In: CRYPTO. pp. 199–203. Plenum Press, New York (1982)
14. Chaum, D., van Heyst, E.: Group signatures. In: EUROCRYPT. LNCS, vol. 547, pp. 257–265. Springer (1991)
15. Cybernetica: Sharemind MPC. <https://sharemind.cyber.ee/sharemind-mpc/> (2020)
16. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requir. Eng.* **16**(1), 3–32 (2011)

17. Der, U., Jähnichen, S., Sürmeli, J.: Self-sovereign identity - opportunities and challenges for the digital revolution. CoRR **abs/1712.01767** (2017)
18. Dingledine, R., Mathewson, N., Syverson, P.F.: Tor: The second-generation onion router. In: USENIX. pp. 303–320. USENIX (2004)
19. Duality Technologies Inc: Duality. <https://dualitytech.com/> (2020)
20. Fernandez, D., Futoransky, A., Ajzenman, G., Travizano, M., Sarraute, C.: Wibson protocol for secure data exchange and batch payments. CoRR **abs/2001.08832** (2020)
21. Garmin-Ltd.: connect: Fitness at your fingertips (2020), <https://connect.garmin.com/>
22. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: STOC. pp. 291–304. ACM (1985)
23. Groth, J.: Non-interactive zero-knowledge arguments for voting. In: ACNS. LNCS, vol. 3531, pp. 467–482 (2005)
24. Ion, M., Kreuter, B., Nergiz, E., Patel, S., Saxena, S., Seth, K., Shanahan, D., Yung, M.: Private intersection-sum protocol with applications to attributing aggregate ad conversions. IACR Cryptol. ePrint Arch. **2017**, 738 (2017)
25. Kim, H., Lee, Y., Abdalla, M., Park, J.H.: Practical dynamic group signature with efficient concurrent joins and batch verifications. IACR Cryptol. ePrint Arch. **2020**, 921 (2020)
26. Koutsos, V., Papadopoulos, D., Chatzopoulos, D., Tarkoma, S., Hui, P.: Agora: A privacy-aware data marketplace. IACR Cryptol. ePrint Arch. **2020**, 865 (2020)
27. KRAKEN Consortium: The Project | KRAKEN (2020), https://www.krakenh2020.eu/the_project/overview
28. linddun.org: LINDDUN privacy engineering (2020), <https://www.linddun.org/>
29. Marr, B.: How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read (2020), <https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/>
30. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from bitcoin. In: IEEE S&P. pp. 397–411. IEEE (2013)
31. Morley-Fletcher, E.: MHMD: my health, my data. In: EDBT/ICDT Workshops. CEUR Workshop Proceedings, vol. 1810. CEUR-WS.org (2017)
32. Mühle, A., Grüner, A., Gayvoronskaya, T., Meinel, C.: A survey on essential components of a self-sovereign identity. Comput. Sci. Rev. **30**, 80–86 (2018)
33. Muoio, D.: Fitbit launches large-scale health study to detect a-fib via heart rate sensors, algorithm (2020), <https://www.mobihealthnews.com/news/fitbit-launches-large-scale-consumer-health-study-detect-fib-heart-rate-sensors-algorithm>
34. Muoio, D.: Google mobilizes location tracking data to help public health experts monitor COVID-19 spread (2020), <https://www.mobihealthnews.com/news/google-mobilizes-location-tracking-data-help-public-health-experts-monitor-covid-19-spread>
35. Noether, S., Mackenzie, A.: Ring confidential transactions. Ledger **1**, 1–18 (2016)
36. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
37. Todd, C., Salvetti, P., Naylor, K., Albatat, M.: Towards non-invasive extraction and determination of blood glucose levels. Bioengineering **4**(4), 82 (2017)
38. Wagner, I., Eckhoff, D.: Technical privacy metrics: A systematic survey. ACM Comput. Surv. **51**(3), 57:1–57:38 (2018)
39. Yao, A.C.: Protocols for secure computations (extended abstract). In: FOCS. pp. 160–164. IEEE (1982)