UNIVERSITY OF SOUTHAMPTON

Dealing with Privacy Risk: Solutions to Data Sharing under the GDPR for Data Controllers

by

Runshan Hu

Thesis for the degree of Doctor of Philosophy

in the

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

September 2020

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Doctor of Philosophy

by Runshan Hu

Personal data are increasingly needed to improve scientific research and decision making in several contexts. However, when collecting or processing data refers to individual respondents, privacy-preserving techniques must be implemented to sanitise or protect the data and guarantee the fundamental right to privacy of data subjects. The growing demand for consistent and comprehensive protection of personal data leads to the adoption of the new General Data Protection Regulation (GDPR).

In this thesis, we investigate privacy risk and data sharing solutions under the GDPR, providing data controllers with some data protection techniques to comply with the GDPR. We first explore the implications of a fundamental terminology - *personal data*, highlighted in the GDPR by interpreting three types of related data: pseudonymised data, Art.11 data and anonymised data, aiming to help data controllers identify what kind of data they are holding. We deploy a risk-based approach to determine how the existing data anonymisation techniques can be assessed in harmony with the new data types in the GDPR.

In light of the promotion of risk assessment methods in the GDPR and our proposed risk-based approach, we further develop a privacy risk mining framework based on machine learning, which consists of a two-phase clustering algorithm and a privacy risk tree model to detect record linkage risk of publishing a new sanitised dataset. This empowers data controllers to envisage the re-identification vulnerabilities and apply more reliable measures for data publishing.

Finally, being aware of the risk and the insufficiency of existing data protection techniques, firstly we propose a privacy management framework for data controllers to improve the utility and security of differentially private data sharing with blockchain technology. Secondly, another framework which combines the blockchain and homomorphic encryption is proposed to outsource centralised anonymisation service and help data owners share data with multiple data controllers.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere appreciation to Professor Vladimiro Sassone, for his continuous support and guidance through the four years.

I would like to extend my gratitude to the researchers in Cyber Security group and my friends at the University. It is nice to work with them and enjoy my Ph.D. life.

Finally, I would also like to thank my parents for their support and encouragement throughout my study.

Declaration of Authorship

I, Runshan Hu , declare that the thesis entitled and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: Runshan et al., 2017, Mu Yang et al., 2018

Signed:.....

Date:.....

Chapter 1

Introduction

Information related to individuals has become a very valuable commodity. The increasing digitisation of personal information in the form of medical records, administrative and financial records, social networks, location trajectories, and so on, creates new avenues for data analytics research. Sharing and mining of this large scale personal data help improve the quality of people's life. However, sharing and analysing the data, or even deriving aggregates over the data raise concerns over the confidentiality of individual participants in the dataset. Therefore, there is a real need to establish appropriate technical solutions in which personal data may be used and exploited whilst guaranteeing basic human right, that is, the right to privacy.

At the same time, the debate about personal data protection has intensified as a result of increasing demand for consistent and comprehensive protection of personal data leading to the adoption of the new law. The new General Data Protection Regulation (the GDPR) (2018) came into effect since May 2018, which is specifically applicable to all the European Union Member States and intended to benefit the EU-based data subjects. According to the Information Commissioner, the GDPR represents an "evolution" regulation (Pantlin, Wiseman, and Everett, 2018). The GDPR has brought some key concepts of data privacy into focus, such as the difference between anonymisation and pseudonymisation, the concept of additional information, and especially the definition of personal data. In addition, the GDPR honours rights of the data subject more seriously than ever before, for example, the right to consent, the right to rectification, the right to be forgotten, and the right to restriction of processing etc. All these rights provide significant protection to data subjects' privacy and meanwhile put a stricter restriction on what data controllers should do and must do. The GDPR has introduced some key changes that are giving rise to closer scrutiny of the personal data use in place for data controllers and, in turn, a shift will be seen in the approach adopted by data controllers in negotiating and implementing data processing arrangements.

Nearly every organisation processes personal data, whether by digital or by manual means. Almost without exception, organisations in the public and private sectors, charities, unincorporated associations, sole traders and individual persons engaged in processing personal data will be affected by the GDPR, in some cases, quite seriously (Bainbridge and Pearce, 1998). These bodies and persons are *data controllers* under the GDPR. As indicated by the name, the data controller manages the overall purpose and means of using the data. There may also be situations where a data controller has to use the service of an external entity to process the data. Using external data processing service does not mean that the data controller outsources the control of personal data to that entity (2018). Such entities are called data processors (Art. 28, General Data Protection Regulation, 2018), who process the data only according to the purpose and instructions given by the data controller. Therefore, in this thesis, we will mainly focus on the responsibilities of data controllers and propose solutions for them.

Data controllers are required to show "sufficient guarantees" to comply with the GDPR by implementing appropriate technical and organisational measures. As such, many organisations will have to modify their processing activities significantly to comply with the new law and, furthermore, more obligations are placed upon them as compared with the former data protection Acts. However, legislators deliberately avoided the idea of recommending specific technical frameworks or privacy-preserving methods for implementing the legal requirements introduced by the GDPR (Politou et al., 2018). Instead, they followed a technology-agnostic approach by specifying the functional requirements in a highly abstracted level, as far as their underlying implementation is concerned as such they did not bind the provisions of the law with the current trends and state-of-the-art technologies in computer science.

There is a knowledge gap between the legal requirements and the privacy-enhancing technologies that must be bridged in order to understand the capability of current data protection approaches and propose better solutions. Therefore, the work in this thesis will contain two main parts in a progressive way. We will help data controllers understand the category and privacy risks of the data they are holding according to the GDPR. Then, we will provide one solution for the controllers to better utilise the data they collected, and to process data in a transparent and tamper-proof way. We will propose another solution for both the data controllers and data owners to alleviate the overhead and privacy risk of performing the anonymisation by utilising encryption technology to protect the privacy of the data and blockchain to manage the outsourcing process.

1.1 Motivation

According to the definition of the term *data sharing* in the *data sharing code of practice*, data sharing indicates the disclosure of data from one organisation to a third party organisation or organisations (ICO, 2018). This definition shows that data can be shared from one party to another, or from one party to multiple parties. These two different data disseminations are also reflected in different scenarios of data sharing. In general, there are three different kinds of scenarios involving the data owner and the data controller.

- data publishing phase, when data controllers believe the data has been sanitised to a re-identification risk-free level and want to share the data with the public,
- data sharing phase, when data are leaving the hands of data owners to data controllers,
- data processing phase, when data are further being processed and analysed by data controllers.

Releasing data to the public poses a threat to the privacy of individuals in the datasets. As the data are completely exposed to the public, an attacker can use any background knowledge within his/her capability to re-identify the data subjects in the dataset. Therefore, a privacy risk assessment is necessary for sanitised data publishing. Besides, the importance of privacy risk assessment is emphasised in the GDPR. The assessment is especially critical for privacy-preserving data publishing due to the fact that the published data will no longer be protected under any confidential measures offered by the data controllers.

However, the privacy risk assessment of privacy-preserving data publishing has proved to be challenging for multiple datasets publishing (Byun, T. Li, et al., 2009). Although each published dataset poses a small privacy risk to individuals, recent studies showed that the risk increases when different organisations have some common records pertaining to the same individuals and these organisations publish their datasets independently. If these records are linked together, it is possible to re-identify specific individuals and hence comprise their privacy. This type of privacy breach is called linkage attack (Dwork, Roth, et al., 2014). A few studies have been done to mitigate this attack (N. Li, T. Li, and Venkatasubramanian, 2007; Machanavajjhala et al., 2007; Sweeney, 2002). However, none of them has discussed the associated risks of linkage attack against multiple heterogeneous datasets. Therefore, it is vital to develop a privacy risk analysis framework for data controllers to understand the potential risk during the privacy-preserving data publishing phase.

Besides releasing sanitised data records in a raw manner, differential privacy offers strict protection when sharing the statistical information of the data. Differential privacy (Dwork, McSherry, et al., 2006) was first proposed over a decade ago and has now

become the de-facto standard for privacy protection due to its provable guarantee and nice composition properties. Informally, a randomised algorithm ensures differential privacy if its output distributions are approximately the same when executed on two input databases that only differ in a single individual record. This requirement prevents an attacker with access to the output of differential private algorithms from learning anything substantial about the presence or absence of any single individual.

The basic idea of differential privacy is to introduce noise to the output. The privacy level under differential privacy is represented by a parameter ε, which is usually called the *privacy budget*, with smaller values corresponding to stronger privacy guarantees. To protect sanitised data from degradation of differential privacy protection, recent approaches (McSherry, 2009) proposed to verify privacy budget which determines the amount of noise produced in the obfuscation process, and stop data sharing as soon as the budget is used up. However, due to the fact that anonymisation services themselves cannot offer adequate guarantee for controlling and tracing privacy budget, it is not trustworthy that data controller will use the privacy budget properly and honestly stop sharing data after the privacy budget being exhausted. The consequence of privacy budget misuse could end up making the de-anonymisation attack of differential privacy possible.

Therefore, achieving trustworthy decentralised management of the privacy budgets for the data controllers is then of paramount importance to ensure privacy protection of sensitive datasets, and, most of all, to help data controller enhance assurance on the anonymisation services and avoid single-point failure of privacy protection due to any untrusted or incapable data controller.

Blockchain is a novel technology that recently came to prominence when used as a public ledger for the Bitcoin cryptocurrency (Nakamoto, 2008). The blockchain network consists of distributed nodes to form a peer-to-peer network. These nodes or so-called peers are responsible for replicating and storing blocks into a consecutive and chained order. Each block is created in a decentralised manner under the agreement of the peers. The agreement is achieved using a consensus algorithm. For example, Bitcoin uses an expensive proof-of-work algorithm to reach the consensus, while Hyperledger Fabric instead employs Byzantine consensus algorithm to avoid heavy computation. Thanks to the consensus algorithms and tamper-resistant transaction, blockchain enjoys some data integrity related properties, for example, transparent and decentralised control of the data, non-repudiation and persistency of the public ledger (Ferdous et al., 2017).

In light of the transparency and decentralisation of the blockchain technology, we will develop a blockchain-based data sharing approach to allow data owners to control anonymisation processes, and to guarantee chosen privacy levels when using data controllers' anonymisation services especially to protect against attacks to differential privacy.

Another following research question in the data sharing lifecycle is where and how the sanitisation process is implemented. When the sanitisation process is performed on the data controller side, data privacy cannot be strictly guaranteed. After gaining the permission to access the data, the data controller is required to sanitise the data according to the agreement with the data owner. Meanwhile, data owners lose control over their data. The data may not be anonymised according to the privacy requirement agreed between data controllers and data owners. Specifically, what kind of sanitisation technique will be deployed by the controller or what privacy parameters will be used in the sanitisation techniques, for example, parameter k in the k-anonymity model or privacy budget in differential privacy, actually depends on the will of the data controller who now has control over the data. Even worse, malicious data controller may share the data to another entity without sanitising the data first or even use the data for unintended purposes not agreed by data owners. Although regulatory measures may impose some legal constraints on the behaviour of the data controller. Still, technical method that can guarantee those constraints and allow data owners to administer and monitor the intended sanitisation process is under research (Zhang et al., 2018). The aims of such technical methods are to enable data owners with the capability to manage which data controller can manage their data and ensure that the sanitisation process is implemented exactly according to their preference.

On the other hand, when anonymisation is carried out on the data owner side, there are other problems. For example, not every data owner has the ability to provide sufficient anonymisation protection for data. One of the more complicated issues is that when the data owner needs to share data with multiple data controllers, and each data controller has different requirements for data utility, the data owner needs to generate multiple anonymous versions of the original data to meet the data utility requirements for each data controller. This places a lot of pressure on the data owner's resource on computing and storage.

In order to resolve this dilemma between the data controller and data owner, we will propose an anonymisation outsourcing framework to enable data controller or data owner to outsource the anonymisation process to a decentralised community formed by data owners via blockchain technology.

1.2 Research Aims and Objectives

The purpose of this thesis is to fill the research gap between legal regulations and technical approaches, aiming to help data controllers process and use personal data in a GDPR-compliant way and ensure that the rights of the data subjects are guaranteed during the data processing lifecycle. To achieve the presented aim and satisfy the listed requirements, we propose solutions with the following objectives.

- Examining the implications of the new law from the perspective of the data controller, especially in term of interpreting the types of personal data held by the controller.
- Developing a privacy risk assessment framework for data controllers to detect privacy risk in the data publishing phase.
- Proposing a technical solution for the data controller to better use and exploit personal data based on the multidisciplinary interpretation.
- Proposing a technical approach to outsource anonymisation service from the data controller to decentralised service providers in order to grant data subjects the ability to control who can have what access to his/her data and be informed that the data is used for the intended purposes.

1.3 Our Solution

In the first piece of our work (Chapter 2), we interpret three types of personal data or non-personal data and develop a risk-based approach to analyse the robustness of existing data anonymisation techniques against three re-identification risks. This work is vital for the data controllers as they are required by the GDPR to know what kind of personal data they are holding and whether the existing anonymisation technique will render the data within the scope of personal data or not. We discover that none of the existing anonymisation techniques can satisfy the strict new requirement of "anonymised data" in the GDPR except using the-state-of-the-art anonymisation technique - differential privacy (Dwork, 2006) for publishing statistic summaries of the data. This result further motivates our second part of work on privacy risk mining of sanitised data from a technical perspective.

The second piece of our work (Chapter 3) in this thesis focuses on helping data controllers assess the privacy risk of publishing sanitised data. To this end, we propose a two-stage risk mining framework for the data controller to estimate the risk of linkage attack in different heterogeneous datasets held by the controller. Using this framework, a data controller can predict the risk of linkage attack in the datasets prior to publication. We also perform empirical analyses to determine the efficacy of our framework to demonstrate that the predicted risk will show us the real risk. After helping data controllers better understanding the types of personal data they are holding and providing them with a framework of analysing the privacy risk in the dataset, we further move into developing novel solutions for the controllers to better utilise the collected data for the intended purpose and outsource the anonymisation service to a decentralised community formed by the data subjects in the third (Chapter 4) and fourth (Chapter 5) pieces of our work, respectively. Realising that differential privacy is the most

suitable GDPR-compliant anonymisation technique, our third piece of work (Chapter 4) focuses on improving the data utility of differential private data sharing and providing a transparency-by-design, privacy-budget-evident data processing mechanism for the data controller via the combination of blockchain technology and differential privacy.

To this aim, we introduce in Chapter 4, a new solution based on blockchain, an innovative technology that among other fascinating properties on data integrity ensures full decentralised control on data and code execution. Our approach utilises blockchain smart contracts to store, verify and adaptively allocate privacy budget consumptions depending on data owner's privacy and data utility requirements. Secure management of privacy budget is indeed the key to ensure privacy in the process. At the same time, we also modify the privacy budget allocation algorithm of differential privacy to support the reuse of the previously released results to answer new queries. In this way, we help the data controller get more utility out of the data. This part of our work not only improves the utility of differential private data sharing but also uses a blockchain-based approach to ensure that the controllers allocate the privacy budget correctly and honestly. The budget usage from the data controller is tracked and monitored by the data owners so as to avoid the controllers from violating differential privacy by sharing too much information.

In the fourth piece of our work (Chapter 5), we propose a framework that can help the data controller outsource anonymisation service to a decentralised community established by data owners. The traditional anonymisation process is performed either at the data owner side or at the data controller. In our approach, the data controller and data owner do not need to execute anonymisation service locally, which are now done through the nodes on the blockchain. Under the premise of ensuring sanitisation reliability, the transparency of the entire process is improved. The issue of trust between the data controller and the data owner is resolved. This framework also offers a solution to *multiple data controllers* to access data owners' data without putting too much overhead on the owners.

Technically, this framework combines blockchain and homomorphic encryption to enable new privacy protection capabilities, that is, outsourcing the anonymisation process and data sharing with multiple data controllers. The advantage of utilising homomorphic encryption is to protect the privacy of the data, meanwhile performing differential privacy mechanism on the encrypted data. This property ensures that data can be anonymised even when the original data and noise parameters are shared separately. At the same time, by encoding differential privacy data sharing policy and usage as smart contracts, the framework can allow data owners to control who can access to their data, and be able to maintain a trustworthy record of their data usage. When a data controller requests the use of the data, the data owner generates the noise parameter and transfer the encrypted noise to the anonymisation service provider in the blockchain. In this way, data owners do not need to implement the anonymisation by themselves. Still, they enjoy

the freedom of control the privacy to the degree they prefer, which also guarantee that the data is sanitised and used for the intended purposes approved by the data owner.

1.4 Key Contributions

The contributions of this thesis are summarised as below.

1. **A Risk-based Approach for Interpreting Three Types of Data in the GDPR:** We interpret three types of data states, which are related to data anonymisation in the GDPR, for data controllers to understand better what kind of data they are holding. We propose a granular risk-based approach to assess the robustness of existing data anonymisation techniques.

2. **Mining Privacy Risk for Data Anonymisation:** We further develop a two-stage clustering algorithm to identify potential linked records among heterogenous anonymised datasets, and utilise a privacy risk tree to quantify the risk of publishing new anonymised datasets from a technical perspective.

3. **Differential Private Data Sharing with Blockchain:** Realising that differential privacy is the most secure anonymisation technique for data publishing at the moment, we combine it with the blockchain technology to provide more utility for sharing and support transparent and immutable tracking of the privacy budget allocation. This work equips data controllers with a possible GDPR-compliant data anonymisation technique.

4. **Outsourcing Anonymisation with Blockchain and Homomorphic Encryption:** We build a data sharing scheme to outsource the differential privacy mechanism to a decentralised blockchain to eliminate the trust dilemma between data owners and data controllers, and enhance the privacy of the anonymisation process. This scheme also alleviates data owners' overhead of sharing data with multiple data controllers.

1.5 Thesis Structure

The rest of this thesis is organised as follows. We first provide a detailed interpretation of data anonymisation and the three types of data corresponding to different levels of anonymisation under the GDPR context in Chapter 2. In Chapter 3, we introduce our two-stage privacy risk mining framework. We then present two solutions for data controller: differential private data sharing with blockchain in Chapter 4, and outsourcing anonymisation with blockchain and homomorphic encryption in Chapter 5, respectively. In Chapter 6, we conclude this thesis and present future work.

Part I

Analysing the Risk

Chapter 2

Data Anonymisation Under the GDPR

In this chapter, we aim to figure out how different types of personal data or non-personal data introduced in the General Data Protection Regulation (GDPR) could be read in harmony with lots of existing anonymisation techniques. We offer a granular analysis of the three types of risks to be taken into account in order to assess the robustness of sanitisation techniques. The risks include *singling out*, *linkability* and *inference*, with linkability being split into local, global and domain linkability. We propose a classification of data sanitisation techniques and contextual controls in relation to the three categories of data found in the GDPR. This work is vital for data controllers as it servers as a cornerstone for them to gain a clearer understanding of the data they are holding so that the controllers can further decide more suitable actions to protect the data.

2.1 Personal Data in the GDPR

In recent years, the debate about personal data protection has intensified as a result of increasing demand for consistent and comprehensive protection of personal data leading to the adoption of new laws.

2.1.1 Changes of the Regulation

The current EU data protection legislation, Data Protection Directive 95/46/EC (DPD) (Directive, 1995), has been replaced by the General Data Protection Regulation (GDPR) from 25 May 2018, which, being a self-executing norm, is directly applicable in all the Member States in the European Union. This legislative reform has generated repeated discussions about its potential impact on business processes and procedures as the GDPR

contains a number of new provisions intended to benefit EU data subjects and comprises a strengthened arsenal of sanctions, including administrative fines of up to 4% of the total worldwide annual turnover of the preceding financial year, for non-compliant data controllers and processors.

One key question is to what extent the GDPR offers better tools than the DPD to frame or confine data analytics as well as data sharing practice. Addressing this issue requires, first of all, delineating the scope of data protection law. Second, it necessitates examining key compliance techniques, such as pseudonymisation, of which the reason is to enable data controllers to strike an appropriate balance between two distinct regulatory objectives: personal data protection and data utility maximisation.

2.1.2 The Relationship between Pseudonymisation and Personal Data

Within the new regulation, Articles 2 (2018) and 4 (2018) are starting points in order to demarcate the material scope of EU data protection law. Under Article 4(1), personal data means:

> any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person;

Recital 26 (2018) further expands upon the notion of identifiability and appears to draw a distinction between personal data and anonymous information, with anonymous being excluded from the scope of the GDPR. Not to be misleading, this key distinction was already present in the DPD. Nonetheless, the GDPR goes further than the DPD in that it indirectly introduces a new category of data as a result of Article 4, i.e. data that has undergone pseudonymisation, which we will name pseudonymised data (Stalla-Bourdillon and Knight, 2016), to use a shorter expression, although the former is more descriptive than the latter for it implies that the state of the data is not the only qualification trigger. Under Article 4(5) pseudonymisation means:

> the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable person;

While the final text of the GDPR does not seem at first glance to create an ad hoc regime with fewer obligations for data controllers when they deal with psedonymisaed data, Recital 29 (2018) specifies:

> In order to create incentives to apply pseudonymisation when processing personal data, measures of pseudonymisation should, whilst allowing general analysis, be possible within the same controller when that controller has taken technical and organisational measures necessary to ensure, for the processing concerned, that this Regulation is implemented, and that additional information for attributing the personal data to a specific data subject is kept separately.

A number of legal scholars have been investigating the contours of personal data under EU law, and have proposed refined categories, creating on occasion a spectrum of personal data, more or less complex (El Emam et al., 2013). The classifications take into account the intactness of personal data (including direct and indirect identifiers (Dalenius, 1986)) and legal controls to categorise data. For instance, with masked direct identifiers and intact indirect identifiers, data is said to become'protected pseudonymous data' when legal controls are put in place (El Emam et al., 2013). We argue in our study that these approaches still rely upon a pre-GDPR understanding of 'pseudonymisation' which should not be confused with GDPR Article 4 definition and thereby have not fully derived the implications of the new legal definitions emerging from the GDPR.

Furthermore, Article 11 (2018) of the GDPR is worth mentioning as it seems to treat with favours a third category of data, which we name Art.11 data for the sake of argument. Art.11 data under Article 11 of the GDPR, is data so that "the [data] controller is able to demonstrate that it is not in a position to identify the data subject". Examining the GDPR a couple of questions therefore emerges: whether and when pseudonymised data can become anonymised data and whether and when pseudonymised data can be deemed to be Art.11 data as well.

2.1.3 Importance of Evaluating Anonymisation Techniques under the GDPR

Article 29 Data Protection Working Party (Art.29 WP) did provide a comprehensive analysis of data anonymisation techniques in the light of the prescriptions of the DPD (Article 29 Data Protection Working Party, 2014). For this purpose, Art.29 WP identified three common risks and tested the robustness of data anonymisation techniques against these risks. However, as this work was done in 2014 against the background of the DPD and the relationship between these techniques and the data categories defined in the GDPR has not been analysed yet.

Therefore, the objective of this chapter is to fill the gap by expressly deriving the implications of the new legal definitions to be found more or less explicitly in the GDPR in order to enable data controllers and regulators to access the robustness of techniques and practices used to strike a compromise between personal data protection and data utility maximisation, as well as to inform the work of researchers, practitioners and data scientists. Consequently, the main contributions of the work are the followings:

- We offer a granular analysis of the three types of risks to be taken into account in order to assess the robustness of sanitisation techniques. The risks include singling out, linkability and inference, with linkability being split into local, global and domain linkability.
- We propose a classification of data sanitisation techniques and contextual controls in relation to the three categories of data found in the GDPR.
- We derive criteria for selecting sanitisation techniques and contextual controls, which data controllers could use to ensure legal compliance as well as regulators to assess legal compliance.

Structure of the chapter In Section 2, we give a brief overview of the new EU data protection legal framework, i.e. the GDPR, and of three risks identified by Art. 29 WP. In Section 3, we unfold our risk-based approach for interpreting the three types of data emerging from the GDPR. The classification of data sanitisation techniques and contextual controls is undertaken in Section 4, followed by our conclusions in Section 5.

2.2 The Three Types of Data in GDPR

As aforementioned, three types of data seem to emerge from the analysis of the GDPR. We introduce them in this section and interpret the underlying meanings of them.

2.2.1 The GDPR Definitions

The definitions presented in this section are derived from the GDPR, including Recital 26 for Anonymised data, Article 4 for pseudonymised data, and Article 11 for Art.11 data.

- ***Anonymised data*** means data that "does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable."

- ***Pseudonymised data*** means personal data that have processed "in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person."
- ***Art.11 data*** means data so that the data controller is "not in a position to identify the data subject" given such data.

The notions of '**identified**' and '**identifiable**' thus appear of paramount importance to distinguish the different types of data and determine whether a category should be considered **personal data**. An individual is usually considered identified if the data can be linked to a unique real-world identity. The term 'identifiable' refers to the capability to identify an individual, who is not yet identified, but is described in the data in such a way that if research is conducted using additional information or background knowledge she can then be identified. This explains why pseudonymised data is still (at least potentially) considered to be personal data. Recital 26 specifies that "personal data which have undergone pseudonymisation, which could be attributed to a natural person by the use of additional information should be considered to be information on an identifiable natural person."

While the two concepts of pseudonymised data and Art.11 data overlap, in order to test the extent to which they actually overlap, it is necessary to start by conceiving them differently. Besides, that Art.11 does not expressly refer to pseudonymisation.

We therefore suggest that in order to characterise data as pseudonymised data, one has to determine whether individuals are identifiable once the additional information has been isolated and separated from the dataset. Furthermore, to determine whether individuals are identifiable once the additional information has been isolated and separated from the dataset, only the dataset at stake should be considered. This is why, as it will be explained below, the concept of pseudonymised data is intimately linked to that of *local linkability* (Stalla-Bourdillon and Knight, 2016).

On the other hand, in order to characterise data as Art.11 data, one has to determine whether a data controller is in a position to identify individuals, i.e. whether individuals are identifiable given the data controller's capabilities, which require considering all the datasets in possession of the data controller; but the data controller's capabilities only (therefore to the exclusion of third parties' capabilities). This is the reason why we suggest that the concept of Art.11 data is intimately linked to that of domain linkability.

Consequently, following this logic, we argue that to characterise data as pseudonymised data or Art.11 data, it is not enough to point to the fact that the individuals are not directly identified within the dataset at stake. As a result, data controllers should not

be entitled not to comply with Articles 15 to 20 (2018) simply based on the fact that they have decided not to collect direct identifiers for the creation of the dataset at stake.

2.2.2 Additional Information

As hinted above, the concept of 'additional information' is closely related to that of pseudonymised data. Indeed, it can make data subjects identified or identifiable if combined with pseudonymised data. The GDPR requires it to be kept separately and be subject to technical and organisational measures. A typical example of additional information is the encryption key used for encrypting and decrypting data such as attributes: the encrypted data thus becomes pseudonymised data when the key is separated and subject to technical and organisational measures such as access restriction measures.

Two other important concepts related to additional information are that of 'background knowledge' and 'personal knowledge (Graham, 2012). In order to analyse re-identification risk properly, it is crucial to draw a distinction between additional information, background knowledge and personal knowledge.

As per GDPR Article 4, Additional information, is the information that can be kept separately from the dataset by technical and organisational measures, such as encryption key, hash function etc.

We distinguish additional information from background knowledge and personal knowledge. Background knowledge is understood as different in kind from additional information as it corresponds to knowledge that is publicly accessible to an average individual who is deemed reasonably competent to access it, therefore most likely including the data controller himself. It comprises information accessible through the Web such as news websites or information found in public profiles of individuals or traditional newspapers. While this kind of knowledge can potentially have a high impact on re-identification risks, it cannot be physically separated from a dataset. Therefore, we exclude it from additional information. However, and this is important, we take it into account when we analyse the three types of data by acknowledging that the potential existence of background knowledge makes it necessary to include singling out as a relevant risk for pseudonymised data within the meaning of the GDPR because as a result of a pseudonymisation process, the data shall not be attributable to an identifiable data subject as well. The same is true for Art. 11 data.

Personal knowledge is assessed through the means of a subjective test (as opposed to background knowledge, which is assessed through the means of an objective test) and varies from one person to another (Graham, 2012). It comprises information that is not publicly accessible to an average individual who is deemed reasonably competent to access it, but only to certain individuals because of their special characteristics. For example, a motivated intruder A has the knowledge that B is currently in hospital, as she

is B's neighbour, and she saw that B was picked up by an ambulance. When combined with anonymised data, this kind of subjective personal knowledge could obviously result in re-identification. However, for the purposes of this work we assume that the likelihood that a motivated intruder has relevant personal knowledge is negligible, which partly depends upon his/her willingness to acquire this relevant personal knowledge and his/her estimation of the value of the data at stake and thereby the degree of data sensitivity. We recognise, however, that further sophistication would be needed for scenarios in which the likelihood that a motivated intruder has relevant personal knowledge is high. In particular, this would mean considering with care the equivalence of sanitisation techniques and contextual controls. With this said, we note that Art. 29 WP wrote in 2007 that " a mere hypothetical possibility to single out the individual is not enough to consider the person as "identifiable" (Article 29 Data Protection Working Party, 2007).

2.2.3 Direct and Indirect Identifiers

As described in the ISO/TS document, direct identifier is "data that can be used to identify a person without additional information or with cross-linking through other information that is in the public domain" (International Organization for Standardization, 2008). Direct identifiers contain explicitly identifying information, such as names and social security numbers that are uniquely linked to a data subject. In contrast, sets of attributes which can be combined together to uniquely identify a data subject, are called indirect identifiers. They include age, gender, zip code, date of birth and other basic demographic information. No single indirect identifier can identify an individual by its own; however, the re-identification risks appear when combining indirect identifiers together, as well as, as aforementioned, when combining records with additional information or with background knowledge. Notably, the list of direct and indirect identifiers can only be derived contextually.

2.2.4 Data Sanitisation Techniques

Data sanitisation techniques process data in a form that aims to prevent re-identification of data subjects. Randomisation and generalisation are considered as two main families of sanitisation techniques (Article 29 Data Protection Working Party, 2014). There is a wide range of techniques including masking techniques, noise addition, permutation, k-anonymity, l-diversity and differential privacy, etc. Noise addition refers to general techniques that make data less accurate by adding noise usually bounded by a range, e.g., [-10, 10]. We differentiate it from differential privacy as the latter offers a more rigorous guarantee.

Masking or removal techniques are applied to direct identifiers to make sure the data subjects are not identified anymore and then additional techniques (including masking techniques) are then used to further process indirect identifiers. It is true that k-anonymity, l-diversity, and differential privacy are more commonly described as privacy models rather than techniques as such. However, as we built upon the Opinion on Anonymisation Techniques (2014) we use similar terminology to simplify the arguments.

2.2.5 Contextual Controls

Contextual controls comprise three sets of controls. First, legal and organisational controls such as obligations between parties and/or internal policies adopted within one single entity (one party) aimed at directly reducing re-identification risks, e.g. obligation not to re-identify or not to link. Second, security measures (including legal, organisational and technical controls) such as data access monitoring and restriction measures, auditing requirements as well as additional security measures, such as the monitoring of queries, all of them aimed at ensuring the de facto enforcement of the first set of controls. Third, legal, organisational and technical controls relating to the sharing of datasets aimed at ensuring that the first set of legal controls are transferred to recipients of datasets. They include obligations to share the datasets with the same set of obligations or an obligation not to share the datasets, as well as technical measures such as encryption to make sure confidentiality of the data is maintained during the transfer of the datasets.

These measures are used to balance the strength of data sanitisation techniques with the degree of data utility. In this sense, they are complementary to data sanitisation techniques. On the one hand, they reduce residual risks, which remain after implementing data sanitisation techniques; on the other hand, they make it possible to preserve data utility while protecting the personal data of data subjects.

In practice, the selection of contextual controls depends on specific data sharing scenarios.

2.3 A Risk-based Analysis of the Three Types of Data

In this section, we conceptualise the three types of risks identified by Art.29 WP (2014) to assess data anonymisation and masking techniques. We refine the concept of linkability and further specify the definitions of the three categories of data emerging from the GDPR using a risk-based approach.

2.3.1 Re-Identification Risks

The re-identification risks relate to ways attackers can identify data subjects within datasets. Art.29 WP's Opinion on Anonymisation Technique (Article 29 Data Protection Working Party, 2014) describes three common risks and, examines the robustness of data sanitization techniques against those risks. Underlying this risk classification is the premise that the means test is a tool to "assess whether the anonymisation process is sufficiently robust" (2014).

- ***Singling out***, which is the "possibility to isolate some or all records which identify an individual in the dataset."
- ***Linkability***, which is the "ability to link at least two records concerning the same data subject or a group of data subjects (either in the same database or in two different databases)."
- ***Inference***, which is the "possibility to deduce, with significant probability, the value of an attribute from the values of other attributes."

In cases in which there is background knowledge, singling out makes an individual identifiable. The connection between identifiability and linkability or inference is less straightforward. Adopting a restrictive approach one could try to argue that if background knowledge exists so that it is known that an individual belongs to a grouping in a dataset, the inferred attribute(s) combined with background knowledge could lead to identification or at the very least disclosure of (potentially sensitive) information relating to an individual.

Art.29 WP categorised data sanitisation techniques into "randomisation", "generalisation" and "masking direct identifiers" (Article 29 Data Protection Working Party, 2014), where randomisation and generalisation are viewed as methods of anonymisation but masking direct identifiers or pseudonymisation (to use the words of Art.29 WP) as a security measure. From now on it should be clear that the GDPR definition of pseudonymisation is more restrictive than merely masking direct identifiers. Masking direct identifiers is conceived as a security measure by Art.29 WP because it does not mitigate the three risks aforementioned; or rather, it simply removes/masks the direct identifiers of data subjects.

"Noise addition", "permutation" and "differential privacy" are included within the randomisation group as they alter the veracity of data. More specifically, noise addition and permutation are considered robust against the linkability and inference risks, but fail to prevent the singling out risk. Differential privacy is able to prevent all the risks but queries must be monitored and tracked when multiple queries are allowed

on a single dataset. As regards the generalisation category, "aggregation" and "k-anonymity" (Sweeney, 2002) are considered robust against singling out, but linkability and inference risks are still in presence. "l-diversity" (Machanavajjhala et al., 2007) is stronger than aggregation and k-anonymity as it prevents both the singling out and inference risks.

Although Art.29 WP has provided important insights for the selection of appropriate data sanitisation techniques, which are relevant in the context of personal data sharing, these techniques ought to be examined in the light of the GDPR. To be clear, the purpose of this work is not to question the conceptualisation of re-identification risks undertaken by Art.29 WP, but to deduce its implications when interpreting the GDPR in context.

2.3.2 Local, Global and Domain Linkability

Analysing in a more granular fashion the linkability risk defined by Art.29 WP, it is possible to draw a distinction between three scenarios. The first scenario focuses on a single dataset, which contains multiple records about the same data subject. An attacker identifies the data subject by linking these records using some additional information. In the second scenario, the records of a data subject are included in more than one datasets, but these datasets are held within one entity. An attacker links the records of a data subject if she can access all the datasets inside the entity, e.g. insider threat (Theoharidou et al., 2005). The third scenario also involves more than one datasets, but these datasets are not necessarily held within one entity. Based on these three scenarios, we distinguish between three types of linkability risks:

- ***Local Linkability***, which is the ability to link records that correspond to the same data subject within the same dataset.
- ***Domain Linkability***, which is the ability to link records that correspond to the same data subject in two or more datasets which are in the possession of the data controller.
- ***Global Linkability***, which is the ability to link records that correspond to the same data subject in any two or more datasets.

Based on this granular analysis of the linkability risk and assuming the concept of identifiability is used consistently across the GDPR, we suggest one way to derive the main characteristics of anonymised, pseudonymised and Art. 11 data within the meaning of the GDPR in the next section.

2.3.3 Privacy Risks Regarding Three Types of Data

2.3.3.1 Anonymised Data

Anonymised data, according to the GDPR definition, is a state of data for which data subjects are not identified nor identifiable anymore, taking into account all the means reasonably likely to be used by the data controller as well as third parties. While strictly speaking the legal test to be found in Recital 26 of the GDPR does not mention all of the three risks aforementioned (i.e. singling out, linkability and inference), we assume for the purposes of this work that for anonymised data to be characterised, singling out, local linkability, domain linkability, global linkability and inference should be taken into account. As aforementioned, whether the three reidentification risks should be re-conceptualised is a moot point at this stage. Suffice it note that not all singling out, linkability and inference practices lead to identifiability and identification. A case-by-case approach is therefore needed.

2.3.3.2 Pseudonymised Data

Pseudonymised data, being the outcome of the pseudonymisation process defined by the GDPR in its Article 4, is a state of data for which data subjects can no longer be identified or identifiable when examining the dataset at stake (and only the dataset at stake). Nevertheless, the foregoing holds true on the condition that data controllers isolate the additional information and put in place "technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person."

As a result, it appears that pseudonymisation within the meaning of the GDPR is not tantamount to masking direct identifiers. In addition, although a number of studies stress the importance of legal controls (El Emam et al., 2013), there are different routes to pseudonymised data depending upon the robustness of the sanitisation technique implemented, as it is explained below.

One important element of the GDPR definition of pseudonymisation is the concept of additional information, which can identify data subjects if combined with the dataset. The definition specifies that such additional information is kept separately and safeguarded, so that the risks relating to the additional information can be excluded. This seems to suggest that in this context, the notion of identifiability should only relate to the dataset at stake. Based on this analysis, we define pseudonymised data as a data state for which the risks of singling out, local linkability and inference should be mitigated. At this stage, the domain and global linkability risks are not relevant and the data controller could for example be in possession of other types of datasets.

In order to mitigate the singling out, local linkability and inference risks at the same time, data sanitisation techniques must be selected and implemented on the dataset. As

aforementioned, Art. 29 WP has examined several sanitisation techniques in relation to re-identification risks (Article 29 Data Protection Working Party, 2014). We build on the upshot of the Opinion on Anonymisation Techniques, and find that K-anonymity, L-diversity and other stronger techniques can prevent these risks, but masking direct identifiers, noise addition, permutation alone is insufficient to reasonably mitigate the singling out, local linkability and inference risks.

The example below illustrates the mitigation of these three risks using k-anonymity.

Example. Table 2.1 shows a sanitised dataset with k-anonymity guarantee (k=4) released by hospital A in May. Suppose an attacker obtains relevant background knowledge from a news website that a famous actor Bob was recently sent to hospital A and that by checking the time it can be deduced that Bob is in the dataset at stake. Suppose as well that the attacker has no access to additional information (e.g. the raw dataset). Since each group of this dataset has at least four records sharing the same non-sensitive attribute values, the attacker cannot distinguish his target Bob from other records. This prevents the risks of singling out and local linkability. Moreover, the attacker is not able to infer the sensitive attribute of Bob because she is not sure to which group Bob belongs. Therefore, this dataset is pseudonymised within the meaning of the GDPR.

TABLE 2.1: An example of Pseudonymised data using k-anonymity (k=4)

	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Diagnosis
1	250**	<30	*	Cancer
2	250**	<30	*	Viral Infection
3	250**	<30	*	AIDS
4	250**	<30	*	Viral Infection
5	250**	3*	*	Cancer
6	250**	3*	*	Flu
7	250**	3*	*	Cancer
8	250**	3*	*	Flu

2.3.3.3 Art.11 Data

Art.11 data, by definition, focuses on the ability of a data controller to identify data subjects to the exclusion of third parties. More specifically, the data controller should be able to demonstrate that she is "not in a position to identify the data subject" (Art. 11, General Data Protection Regulation, 2018). First, this implies that direct identifiers (e.g. names, social security number, etc.) have been removed or have never been collected. In other words, Art.11 data is either de-identified by a certain process or originally non-identifying. Second, "not being in a position to identify the data subject" should also imply that the combination of indirect identifiers with relevant background knowledge accessible to the data controller does not lead to identification. There exist also situations

where data controller only collects indirect identifiers but a very rich of list of indirect identifiers for which arguably, and this is crucial, no accessible relevant background knowledge exists and the data controller is not in possession of other datasets which could be linked to the first one, e.g. dynamic IP addresses, browsed websites and search terms, transactions, etc. in order to create profiles and ultimately make decisions about individuals. We suggest that while an approach purely based on a re-identification risk approach would lead to exempting data controller from Article 15 to 20 in these situations, this would not necessarily be consistent with the spirit of the GDPR, which aims to strengthen the protection of data subjects in cases of profiling. As a result, in order to determine whether data is personal data and the full data protection regime applies two scenarios must be taken into account: 1) whether re-identification risks have been appropriately mitigated and 2) whether profiling and decisions about individuals are made.

Importantly, Art.11 definition requires that to determine whether the data is Art.11 data, all the means of the data controller should be considered to the exclusion of third parties' means. As a result, Art.11 data can be interpreted as a state of data for which there are no risks of singling out, domain linkability and inference. The protection applied to Art.11 data is therefore stronger than the protection applied to pseudonymised data because the former requires mitigating the domain linkability rather than local linkability risk. This does not mean that pseudonymised data cannot be transformed into Art.11 data. The example below illustrates the difference between Art. 11 and pseudonymised data.

Example. Suppose two hospitals H_1 and H_2 located in the same city publish patient data frequently, e.g., weekly. Table 2.2 is the dataset sanitised and published by H_1 using k-anonymity ($k = 4$). The dataset achieves the state of pseudonymised data as no record in the table can be attributed to a particular data subject without the use of additional information. Furthermore, H_1 claims that it is not able to identify any data subject using any other information within the domain/access of H_1. This other information could be the datasets previously published by H_1 and H_2. One week later, H_2 publishes its own patient dataset. It sanitises the data using k-anonymity ($k = 6$) and achieves the state of pseudonymised data, as shown in Table 2.3. Now H_2 wants to determine whether the dataset (Table 2.3 is also Art. 11 data. H_2 is in possession of other information (different from the concept of additional information) comprising Table 2.2, and background knowledge deriving from a news website (which has been read by many people in the city) saying that a 28-year-old celebrity living in zip code 25013 has been sent to both H_1 and H_2 to seek a cure for his illness. H_2 thus goes through the medical records of each patient. With the other information, H_2 knows that the celebrity must be one of the four records in Table 2.2 and one of the six records in Table 2.3. H_2 is therefore able to identify the celebrity by combining Table 2.2 and Table 2.3, because

only one patient was diagnosed with the disease that appears in both tables, i.e., cancer. As a result, H_2 can be sure that the celebrity matches the first record of both tables, and the celebrity has cancer. Therefore, Table 2.3 comprises pseudonymised data but not necessarily Art. 11 data.

TABLE 2.2: An example of 4-anonymous patient data from hospital H_1

	Non-Sensitive			Sensitive
	Zip code	Age	B_city	Diagnosis
1	250**	≤ 30	*	Cancer
2	250**	≤ 30	*	Viral Infection
3	250**	≤ 30	*	AIDS
4	250**	≤ 30	*	Viral Infection
5	250**	3*	*	AIDS
6	250**	3*	*	Heart Disease
7	250**	3*	*	Heart Disease
8	250**	3*	*	Viral Infection
9	250**	≥ 40	*	Cancer
10	250**	≥ 40	*	Cancer
11	250**	≥ 40	*	Flu
12	250**	≥ 40	*	Flu

TABLE 2.3: An example of 6-anonymous patient data from hospital H_2

	Non-Sensitive			Sensitive
	Zip code	Age	B_city	Diagnosis
1	250**	≤ 35	*	Cancer
2	250**	≤ 35	*	Tuberculosis
3	250**	≤ 35	*	Heart Disease
4	250**	≤ 35	*	Heart Disease
5	250**	≤ 35	*	Flu
6	250**	≤ 35	*	Flu
7	250**	≥ 35	*	Heart Disease
8	250**	≥ 35	*	Viral Infection
9	250**	≥ 35	*	Flu
10	250**	≥ 35	*	Flu
11	250**	≥ 35	*	Flu
12	250**	≥ 35	*	Flu

We summarise the three types of data based on the risks aforementioned in Table 2.4. Domain and global linkability are not applicable for pseudonymised data, which are denoted as "N/A". The inapplicability is due to the definition of pseudonymised data. When considering the risks of pseudonymised data, only the dataset at stake is considered. The definition specifies that additional information is kept separately and safeguarded. So the risks relating to the additional datasets in the controller's domain or globally available datasets can be excluded. Therefore, the domain and global linkability risks are not relevant to pseudonymised data.

Similarly, global linkability is not applicable for Art.11 data. The definition of Art.11 data specifies that the scope of the datasets needed to be considered is the datasets that are in possession of the data controller (which is what the "domain" means), excluding those globally available datasets. Therefore, global linkability is not relevant to Art.11 data.

TABLE 2.4: Risk-based interpretation for three types of data in the GDPR

	Singling out	Local linkability	Domain linkability	Global linkability	Inference
Anonymised data	✗	✗	✗	✗	✗
Art.11 data	✗	✗	✗	N/A	✗
Pseudonymised data	✗	✗	N/A	N/A	✗

2.4 The GDPR in Practice: Sanitisation Techniques and Contextual Controls

We now examine the robustness of practical data sanitisation techniques against the five types of re-identification risks. Taking into account data sharing contexts, we presents a hybrid assessment comprising both contextual controls and data sanitisation techniques.

2.4.1 Effectiveness of Data Sanitisation Techniques

We build upon the table of data sanitisation techniques presented by Art. 29 WP (2014) by splitting the linkability risk into local and global linkability. At this stage, domain linkability is not explicitly shown in the table as it is included in the global linkability. The table below summarises the results.

TABLE 2.5: Robustness of data sanitisation techniques against privacy risks

	Is singling out still a risk?	Is local linkability still a risk?	Is domain/global linkability still a risk	Is inference still a risk
Masking direct identifiers	Yes	Yes	Yes	Yes
Noise Addition	Yes	Yes	Yes	Yes
Permutation	Yes	Yes	Yes	Yes
Masking indirect identifiers	Yes	Yes	Yes	Yes
Aggregation or K-anonymity	No	No	Yes	Yes
L-diversity	No	No	Yes	No
Differential privacy	May not	May not	May not	May not

Note that domain linkability is in the same column as global linkability, because for both situations external datasets need to be taken into account and the listed data sanitisation techniques are not able to distinguish between different types of domains.

While one should revert to explanations provided by Art. 29 WP (2014) for the analysis of the singling out and inference risks, we then discuss the robustness of sanitisation techniques in relation to local, domain and global linkability risks.

Masking direct identifiers. Applying the techniques, such as encryption, hashing and tokenisation on direct identifiers, can reduce linkability between a record and the original identity of a data subject (e.g., name). However, it is still possible to single out data subjects' records with the pseudonymised attributes. If the same pseudonymised attribute is used for the same data subject, then records in one or more datasets can be linked together. If different pseudonymised attributes are used for the same data subject and there is at least one common attribute between records, it is still possible to link records using other attributes. Therefore, the local, domain and global linkability risks exist in both situations.

Noise Addition. This technique adds noise to attributes, making the values of such attributes inaccurate or less precise. However, this technique cannot mitigate local, domain and global linkability risks. Indeed, this technique only reduces the reliability of linking records to data subjects as the values of attributes are more ambiguous. Records may still be linked using inaccurate attribute values and linking those records together will further cause the sensitive attributes of the record being inferred.

Permutation. Permutation is a technique that consists in shuffling values of attributes within a dataset. More specifically, it swaps values of attributes among different records. It can be considered as a special type of noise addition though it retains the range and distribution of the values (2014). Therefore, it is still vulnerable to the local, domain and global linkability risks based on the shuffled values of attributes, although such linking may be inaccurate as an attribute value may be attached to a different subject.

Aggregation or K-anonymity. As the main technique of the generalisation family, aggregation and K-anonymity are applied to prevent singling out. They group a data subject with at least $k-1$ other individuals who share a same set of attribute values (Sweeney, 2002). These techniques are able to prevent local linkability, because the probability of linking two records to the same data subject is no more than $1/k$. However, they are not able to mitigate the domain and global linkability risks. As shown in our example of the two hospitals, records relating to the celebrity can be linked together via an intersection attack (Ganta, Kasiviswanathan, and Smith, 2008).

L-diversity. Compared with K-anonymity, the significant improvement of L-diversity is that it ensures the sensitive attribute in each equivalence class (i.e., the k group) has at least L different values (2007). Thus, it prevents the risk of inference to the probability of no more than $1/L$. However, like K-anonymity, it cannot prevent domain and global linkability because it is still possible to link records together if they have the same sensitive attribute values.

Differential privacy. Differential privacy is one of the randomisation techniques that can ensure protection in a mathematical way by adding a certain amount of random noise to the outcome of queries (Dwork, 2008). Differential privacy means that it is not possible to determine whether a data subject is included in a dataset given the query outcome. In the situation where multiple queries on one or more datasets are allowed, the queries must however be tracked and the noise should be tuned accordingly to ensure attackers cannot infer more information based on the outcomes of multiple queries. Therefore, "May not" is assigned for the risks depending on whether queries are tracked.

Masking indirect identifiers. As described before, encryption, hashing and tokenisation are the techniques for masking direct identifiers. They can also be implemented on indirect identifiers. We observe that these techniques are not able to mitigate the risks of local, domain and global linkability. Taking a dataset with three quasi-identifiers - gender, address and date of birth, for example, a hash function encrypts the combination of the three quasi-identifiers. If there are two records in the dataset (or different datasets) corresponding to a same data subject, then they will have the same hashed values for these three attributes.

We now combine our risk-based interpretation of three types of data in Table 2.4 with the foregoing analysis of the robustness of data sanitisation techniques in Table 2.5, in order to classify the output of different techniques into three types of data shown in Table 2.6.

TABLE 2.6: The results of data sanitisation techniques regarding three types of data

Techniques	Pseudonymised data	Art.11 data	Anonymised data
Masking direct identifiers	Not	Not	Not
Noise Addition	Not	Not	Not
Permutation	Not	Not	Not
Masking indirect identifiers	Not	Not	Not
Aggregation or K-anonymity	Not	Not	Not
L-diversity	Yes	Not	Not
Differential privacy	Maybe	Maybe	Maybe

As the first four techniques are not able to mitigate the risk of singling out, the outcome of these four techniques cannot be pseudonymised data, Art. 11 data, or anonymised data. For k-anonymity, it cannot produce any of these three data types because it only mitigates singling out and local linkability to the exclusion of inference when additional information is isolated and safeguarded. Notably, background knowledge is taken into account. Data after implementing l-diversity is pseudonymised data because it can

mitigate singling out, local linkability, and inference, but not domain linkability or global linkability. As for Art. 11 data, l-diversity does not mitigate against the fact that data controllers have within their domain other datasets, which can be used to link records together. Hence, "Not" is assigned. "Maybe" is assigned to differential privacy as it can guarantee Art. 11 data, pseudonymised data or anonymised data if a single query on one dataset is allowed or multiple queries are tracked. The uncertainty depends on whether the technique is appropriately implemented with control over the privacy budget consumption.

So far, we have classified data sanitisation techniques with respect to the three types of data. It is worth mentioning that data sanitisation techniques are often combined in practice. Table 2.6 derives the sanitisation outcome in situations where two or more techniques are implemented. For example, (k, l) - anonymity (Byun, T. Li, et al., 2009) combining k-anonymity and l-diversity, ensures that each equivalent class has at least k records, and their sensitive attributes have at least l different values. (k, l) - anonymity guarantees that there are no risks of singling out, local linkability and inference.

2.4.2 Improving Data Utility with Contextual Controls

Maintaining an appropriate balance between data utility and data protection is not an easy task for data controllers in practice. As discussed in Section 2.4.1, K-anonymity, L-diversity and differential privacy are the sole potential techniques that can render data pseudonymised, Art.11 or anonymised. However, these techniques could introduce undesired distortion on data, making data less useful for data analysis. Contextual controls are thus crucial to complement data sanitisation techniques and reduce risks (Leibniz Institute for Educational Trajectories (LIfBi), 2009). Obviously, the strength of the contextual control to add should depend upon the type of data sharing scenarios at hand.

In order to take into account the variety of data sharing scenarios, we distinguish between two types of contextual legal controls: "inter-party" and "internal controls". The former category comprises obligations between parties (i.e. data controller/data releaser and data recipient), and the latter comprises internal policies adopted within one entity, i.e. one party. As shown in Table 2.7, the top rows of controls are meant to directly address the re-identification risks. The middle rows list the controls used to ensure that the first set of controls are actually implemented. More specially, security measures are measures that relate to location of storage, access to data, training of staff and enforcement of internal policies. Additional security measures are associated with differential privacy only and are required to guarantee differential privacy mitigates all the risks. The third set of controls is essential when data are shared in order to make sure recipients of datasets put in place the necessary controls to maintain the dataset within the dataset within its initial category: depending upon the sensitivity of the data they take the form

of obligations/policies not to share the data or an obligation to share the data alike, i.e. with the same control. Technical measures, such as encryption, can complement these obligations to make sure confidentiality of the data is maintained during the transfer of the dataset to the recipient.

TABLE 2.7: Inter-party (obligation) and Internal (policies) controls

1. Mitigating risks directly	Singling out risk ➠Obligation/Policy to isolate info to de-mask directly identifiers with security measures in relation to location of storage, access to formula, training of staff and enforcement of rules ➠Obligation/Policy not to identify from indirect identifiers
	Local linkability risk ➠Obligation/Policy not to link records in the same dataset
	Domain linkability risk ➠Obligation/Policy not to link with other datasets within the same domain
	Global linkability risk ➠Obligation/Policy not to link with other datasets
	Inference risk ➠Obligation/Policy not to infer attributes from existing attributes
2. Enforcing the mitigation	Security measures ➠Obligation/Policy to implement security measures in relation to location of storage, access to dataset, training of staff and enforcement of internal policy rules
	Additional security measures ➠Obligation/Policy to monitor queries and query outcome after applying differential privacy
3. Transferring controls	➠Obligation/Policy not to re-share or to re-share with the same set of obligations ➠Obligation to share data in an encrypted state, e.g. through an encrypted communication channel

It is now time to combine data sanitisation techniques and contextual controls to determine when and how it is possible to maintain data utility. This is the objective of Tables 2.8 and 2.9. Two types of actors are distinguished to take into account the implications of data sharing scenarios: data collectors, who collect original data and transform the data in certain data types before sharing the data; and data recipients, who receive processed data and may have to implement controls in order to ensure the data remain within the desired data category. Table 2.8 only concerns data collectors. This is why no inter-party controls are considered.

In the first row of the table, data fall into the category of pseudonymised data when the singling out, local linkability and inference risks have been mitigated. When implementing a weak sanitisation technique only, i.e. masking direct identifiers, those risks still persist as explained above and contextual controls are therefore needed. Stronger

TABLE 2.8: Sanitisation options when data are in the hands of data collectors

Desired data type	**Sanitisation options**
Pseudonymised data	➠Masking direct identifiers + Policies on singling out, local linkability and inference risks + Security measures ➠K-anonymity + Policy on inference risk + Security measures ➠L-diversity + Security measures
Art. 11 data	➠Masking direct identifiers/Collecting only indirect identifiers + Policies on singling out, domain linkability risks + Security measures ➠K-anonymity + Policies on inference and domain linkability risks + Security measures ➠L-diversity + Policy on domain linkability risk + Security measures
Anonymised data	➠Masking direct identifiers + Policies on singling out, local, global linkability and inference risks + Security measures ➠K-anonymity + Policies on inference and global linkability risks + Security measures ➠L-diversity + Policies on global linkability risk + Security measures ➠Differential privacy + Security measures + Additional security measures

data sanitisation techniques, such as K-anonymity and L-diversity, mitigate more risks, which explains why fewer and/or weaker contextual controls are needed. For instance, when L-diversity is implemented, only security measures are required for achieving pseudonymised data. In the end the selection of data sanitisation techniques and contextual controls should depend on the type of data sharing scenario pursued (closed or open) given both the sensitivity and the utility of the data. Data in the second category, i.e. Art. 11 data, implies that the data controller is able to demonstrate that she is not in a position to identify data subjects. The listed options ensure that there are no singling out, domain linkability and inference risks. Data in the final category is anonymised data, which require the strongest protection, i.e. that no singling out, local and global linkability and inference risks exist. Differential privacy is one of the options, and only security measures are required when differential privacy is implemented.

Table 2.9 concerns data recipients. As for data recipients who receive processed data, they should take into account (i) the data sanitisation techniques that have been implemented on the received data, and (ii) the obligations imposed by data releasers.

TABLE 2.9: Sanitisation options when data are in the hands of data recipients

Desired data type	**Sanitisation techniques implemented on received data**	**Obligations imposed upon data recipients**	**Sanitisation options**
Pseudonymised data	Masking direct identifiers	Obligations on singling out, local linkability and inference risks + obligation on implementing security measures	➠Policies on singling out, local linkability and inference risks + Security measures ➠K-anonymity + Policy on inference risk + Security measures ➠L-diversity + Security measures
	K-anonymity	Obligation on inference risk + obligation on implementing security measures	➠Security measures ➠L-diversity + Security measures
	L-diversity	Obligation on implementing security measures	➠Security measures
Art. 11 data	Masking direct identifiers	Obligations on singling out, inference, local and domain linkability risks + obligation on implementing security measures	➠Policies on singling out, inference, local and domain linkability risks + Security measures ➠K-anonymity + Policies on inference, domain linkability risks + Security measures ➠L-diversity + Policy on domain linkability risk + Security measures

	K-anonymity	Obligations on inference and domain linkability risks + obligation on implementing security measures	➠Policies on inference and domain linkability risks + Security measures ➠L-diversity + Policy on domain linkability risk + Security measures
	L-diversity	Obligation on domain linkability risk + obligation on implementing security measures	➠Policy on domain linkability risk + Security measures
Anonymised data	Masking direct identifiers	Obligations on singling out, local, global linkability and inference risks + obligation on implementing security measures	➠olicies on singling out, local, global linkability and inference risks + Security measures ➠K-anonymity + Policies on inference and global linkability risks + Security measures ➠L-diversity + Policy on global linkability risk + Security measures ➠Differential privacy + Security measures + Additional security measures

	K-anonymity	Obligations on inference and global linkability risks + obligation on implementing security measures	➠Policies on global linkability and inference risks + Security measures ➠L-diversity + Policy on global linkability risk + Security measures ➠Differential privacy + Security measures + Additional security measures
	L-diversity	Obligation on global linkability risk + obligation on implementing security measures	➠Policy on global linkability risk + Security measures ➠Differential privacy + Security measures + Additional security measures
	Differential privacy	Obligation on implementing security measures	➠Security measures + Additional security measures

Table 2.9 provides a number of sanitisation options that data recipients can select to meet their data protection and utility requirements. We take pseudonymised data as an example. Suppose a data recipient receives data that were processed with k-anonymity techniques and she aims to keep the data in a pseudonymised state. The data recipient has thus two options. Either she does not change the data and simply adopt policies and security measures; or she further processes the data with l-diversity, and adopt different types of policies as well as security measures. Another consideration is worth mentioning. If the data collector keeps the original raw dataset, the original raw dataset should be conceived as falling within the category of additional information for the purposes of characterising personal data and within the category of the data controller's domain for the purposes of characterising Art. 11 data. As regards anonymised data, Art. 29 WP seems to suggest that as long as the raw dataset is not destroyed the sanitised dataset cannot be characterised as anonymised data (2014). Applying a risk-based approach of the type developed in this paper would lead to the opposite result. This said, and this is essential, this would not mean that the data controller transforming and releasing the raw dataset into anonymised data would not be subject to any duty anymore. It

would actually make sense to impose upon the data controller a duty to make sure recipients of the dataset put in place the necessary contextual controls. This duty could be performed by imposing upon recipients an obligation not to share the dataset or to share the dataset alike, depending upon data sensitiveness and data utility requirements. Ultimately, the data controller would also be responsible for choosing the appropriate mix of sanitisation techniques and contextual controls as the anonymisation process as such is still a processing activity governed by the GDPR. Data controllers could thus be required to monitor best practices in the field even after the release of the anonymised data.

Finally it should be added that the foregoing analysis implies a relativist approach to data protection law, which would require determining the status of a dataset on a case-by-case basis and thereby for each specific data sharing scenario.

2.4.3 Improving Data Utility with Dynamic Sanitisation Techniques and Contextual Controls

Re-identification risks are not static and evolve over time. This should mean that data controllers should regularly assess these risks and take appropriate measures when their increase is significant.

Notably, adapting sanitisation techniques and contextual controls over time can help reduce re-identification risks. At least one dynamic sanitisation technique is worth mentioning here: changing pseudonyms over time for each use or each type of use as a way to mitigate linkability (Hintze and LaFever, 2017). Besides, techniques like k-anonymity and l-diversity can also be conceived as dynamic techniques as deploying k or l on the same dataset for new recipients can provide stronger protection when the data controller observes that re-identification risks increase.

At the same time, data recipients should be aware of the limits imposed upon the use of the data, even if the data is characterised as anonymised. This is a logical counterpart to any risk-based approach and necessarily implies that data controllers and data recipients are in continuous direct contact, at least when differential privacy is not opted for. Indeed, contextual controls put in place for mitigating risks directly (in order to preserve data utility) could be coupled with confidentiality obligations and/or confidentiality policy, be it relative (i.e. formulated as an obligation to share alike) or absolute (i.e. formulated as a prohibition to share). Importantly, taking confidentiality obligations seriously would then make it possible to then assess the likelihood of the singling out, linkability and inference risks leading to re-identification and could make certain types of singling out, linking and inferring practices possible, as long as the purpose of the processing is not to reidentify data subjects and there is not a reasonable likelihood that the processing will lead to reidentification. It is true, nevertheless that

the choice of confidentiality obligations coupled with weak sanitisation techniques can prove problematic if datasets are shared with multiple parties, even if each receiving party agrees to be bound by confidentiality obligations and adopt internal policies for this purpose. Obviously, access restrictions techniques and policies are a crucial means to make sure confidentiality obligations and policies are performed and/or implemented in practice.

Notably, while in the Breyer case of 2016 the CJEU interpreting the notion of "additional data which is necessary in order to identify the user of a website" considered the information held by the user's internet access provider, the CJEU recognised the importance of legal means in order to characterise personal data (2016). We suggest contractual obligations should be taken seriously into consideration in particular when they are backed up by technical measures such as measures to restrict access and dynamic measures to mitigate linkability.

2.5 Conclusion

The purpose of this study was to test the possibility of interpreting the GDPR and Art. 29 WP's Opinion on Anonymisation Techniques together, assuming the concept of identifiability has two legs (identified and identifiable), the three risks of singling out, linkability and inference are relevant for determining whether an individual is identifiable and the concept of identifiability is used consistently across the GDPR. On the basis of an interdisciplinary methodology, this study therefore builds a common terminology to describe different data states and derive the meaning of key concepts emerging from the GDPR: anonymised data, pseudonymised data and Art. 11 data. It then unfolds a risk-based approach, which is suggested to be compatible with the GDPR, by combining data sanitisation techniques and contextual controls in an attempt to effectively balance data utility and data protection requirements. The proposed approach relies upon a granular analysis of re-identification risks expanding upon the threefold distinction suggested by Art. 29 WP in its Opinion on Anonymisation Techniques. It thus starts from the three common reidentification risks listed as relevant by Art. 29 WP, i.e. singling out, linkability and inference and further distinguishes between local, domain and global linkability to capture the key concepts of additional information and pseudonymisation introduced in the GDPR and comprehend the domain of Article 11 as well as the implications of Recital 26. Consequently, the study aims to make it clear that even if a restrictive approach to re-identification is assumed, the GDPR makes the deployment of a risk-based approach possible: such an approach implies the combination of both contextual controls and sanitisation techniques and thereby the adoption of a relativist approach to data protection law. Among contextual controls, confidentiality obligations are crucial in order to reasonably mitigate re-identification risks. In Table 2.8 and 2.9, we claim that differential privacy combined with additional security measures can be

used to sanitise data to an "anonymised" level, where additional security measures are required to guarantee differential privacy mitigates all the risks. The requirements are obligations for data controllers from a legal perspective. In Chapter 4 and 5, we propose technical solutions to explain what kinds of additional security measures can be deployed and prove that the combination does work. [1]

[1]The full version of this paper can be accessed at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3034261

Chapter 3

Mining Privacy Risk for Data Anonymisation

In this Chapter, we expand the work in Chapter 2 and focus on solving the problem of mining privacy risk from a technical perspective aiming to help data controllers better assess the risk of data publishing. We consider the dynamic context in which heterogenous and homogenous datasets containing overlapping individuals are released continuously, and thus present possibilities for linking those data subjects. We develop a two-stage clustering framework. The first stage is to identify the global linkability among datasets through their common attributes; the second is to capture the local linkability among anonymised records via overlapped attribute values. The two types of linkability are used to estimate the privacy risk through a privacy risk tree. Our experiments on ten UCI datasets demonstrate the accuracy and efficiency of the proposed framework in detecting record linkage. The main target audience of this framework would be data controllers with sensitive datasets to release on a regular basis in the data publishing phase.

3.1 Background of Privacy Risk Assessment

3.1.1 Privacy Risk Mining in Dynamic Data Publishing

Privacy preserving data publishing has been broadly studied in recent years. Although a considerable number of techniques, such as k-anonymity (Samarati, 2001; Sweeney, 2002), l-diversity (Machanavajjhala et al., 2007) and t-closeness (N. Li, T. Li, and Venkatasubramanian, 2007) have provided valuable solutions that render data secure to publish, these models are limited to single data release (i.e. anonymisation is implemented on one dataset at stake). While this may be acceptable in an isolated sharing

environment, it remains a critical concern in the current big data era, when *heterogeneous* datasets (i.e. datasets about different topics, such as criminal data and patient data which may contain some overlapping individuals) or *homogenous* but time variant datasets (Byun, T. Li, et al., 2009) (i.e. a new dataset is generated by augmenting new records to previous datasets) are released regularly. Such dynamic data publishing context calls into question the security of current static anonymisation techniques, especially the concern of linking records corresponding to the same data subjects in these datasets to re-identify the person.

Privacy risk assessment has been increasingly promoted by researchers as well as regulations, such as the EU General Data Protection Regulation (GDPR) (Commission, 2016) due to the growing public sensitivity to privacy protection. Although risk assessment methodologies are helpful in identifying system vulnerabilities and suggesting best practice to mitigate risks, there is a lack of technical solutions for detecting re-identification risks, especially in dynamic data publishing contexts. This work advocates a more focused approach based on estimating the linkability between records and taking into account heterogeneous and homogenous datasets published previously. We suggest that using a 'distance' between datasets and records, we can capture such linkability and thus characterise the potential relation between data.

In this chapter, we propose a two-stage privacy risk mining framework. We harness the power of machine learning to formalise the re-identification analysis as an unsupervised clustering problem. More specifically, our approach clusters potentially linked datasets and records in corresponding groups: a k-means algorithm is chosen to model the topics of datasets and classify datasets by topics; a k-members algorithm is deployed to analyse the similarities between different records within each topic-based cluster. To evaluate the effectiveness of the framework, our experiments with 10 UCI datasets explore the selection of parameters, and how they affect the accuracy of identifying linked datasets and records.

The rest of the chapter discusses the notions of global and local linkability, and their calculation through machine learning algorithms. It subsequently illustrates how these two types of linkability are integrated to estimate the privacy risks using a privacy risk tree, and presents the experimental results. Finally, we conclude the work and discuss the potential improvements for privacy risk assessment in dynamic data publishing.

3.1.2 Related Work

Record linkage analysis (Christen, 2012) is the most relevant to the present work. It looks at how to recognise and match records that are associated with the same real-world individuals or objects from different datasets. However, most record linkage techniques

work on the detection of linkage from original datasets, which are not processed by any anonymisation techniques.

The study by Byun et al. (Byun, T. Li, et al., 2009) investigates the record linkage in the context of incremental data dissemination where the same data may be anonymised and released several times. Unlike our problem, these homogenous records have the same data structure and can be directly compared to discover the linkage between records. Abril et al. (Abril, Navarro-Arribas, and Torra, 2012) propose to use record linkage for estimating the disclosure risk of anonymised data. However, this proposal considers the worst case scenario where records matching only happens between the original data and its anonymised version. This assumption overlooks the re-identification risks caused by heterogenous datasets containing linked records; Besides, the linking status is assumed known beforehand, thus it focuses on calculating how close are the links between records. Our study does not assume that such information is available, which is more challenging.

3.2 Linkability Analysis

3.2.1 Global and Local Linkability

The main idea of the proposed privacy risk mining framework is to estimate the linkability between records from anonymised heterogeneous datasets, so as to discover if any two or more records belong to the same individuals. A possible solution to characterise such linkage is to calculate the distance between records' values. However, as heterogeneous datasets usually have different attributes, comparing records from these datasets by computing a distance between them is not feasible. In the light of this, we propose a novel framework that mines records linkability from heterogeneous datasets by splitting linkability into *global* and *local* linkability, as introduced in our paper (Runshan et al., 2017). The definitions of these two types of linkability are described as follows.

- **Global linkability** represents the possibility of linking datasets that contain records corresponding to the same data subjects.
- **Local linkability** represents the possibility of linkage appears on records level between two records corresponding to the same data subject.

Relying on this, our framework mines records' linkability in two stages: the first stage is to cluster datasets into different groups so that possible matching records are clustered together, while the second captures the local linkability at record level within the same group of datasets.

3.2.2 Measuring Global Linkability

In order to measure the global linkability, anonymised datasets need to be clustered into different groups, each of which contains datasets that may include records belonging to the same data subjects. The column attributes of each dataset contain the information which can be used to summarise the topic of a dataset. We assume here that the data scheme is not maliciously misleading. We believe this assumption is safe, as the data curator will ensure the correctness in order to detect linked records. We use the attributes to describe a dataset and form the clusters. Let n be the total number of datasets. The l-th ($1 \leq l \leq n$) dataset D_l is denoted by an attribute vector

$$(a_{l1}, a_{l2}, \ldots, a_{ln_l}; D_l),$$

where n_l is the number of attributes for dataset D_l. Each $a_{lj} (1 \leq j \leq n_l)$ represents the j-th attribute of dataset D_l and has a text value.

We choose the k-means algorithm for three main reasons. Firstly, the k-means algorithm is intuitive and straightforward. It geometrically partitions data points into clusters, and each of these clusters has a well-defined centroid. The clustering process visually shows how data points are grouped. Secondly, the k-means algorithm is efficient and easy to implement. The technique works by assigning a data point to its closest cluster and then adjusting clusters' centroids until all the data points reach convergence and no longer change their groups. This algorithm flow is easy to implement. Thirdly, the k-means algorithm has no particular requirements for the distribution or density of the data. In our scenario, the data points have no significant distribution or density characteristics, so the centroid-based approach is more suitable than the distribution-based and density-based clustering methods. Besides, in our two-stage clustering framework, the k-means technique is utilised to pre-cluster the data space into disjoint smaller sub-spaces where the second clustering algorithm that is k-members clustering algorithm can be applied. A constraint of the algorithm is that it works on numerical data, as it uses a Euclidean distance as a similarity measure (X. Chen et al., 2009). Since the column attributes of datasets are categorical rather than numeric, it is necessary to convert them to numerical values. Let m be the total number of semantically different attributes in n datasets, with $m \leq \sum_{l=1}^{n} n_l$. This is because the identical or semantically similar attributes from different datasets can be generalised. For instance, if there are three census datasets:

$$(age, sex, education, work\text{-}class; D_1),$$
$$(birthdate, gender, marital\text{-}status, occupation, salary; D_2),$$
$$(relationship, sex, wage\text{-}perhour; D_3).$$

After generalising semantically similar and identical attributes, the attribute vector for all the three datasets can be extracted as

$$(age, gender, education, marital\text{-}status, occupation, salary).$$

More specifically, *age* in D_1 and *birthdate* in D_2, *sex* in D_1 and *gender* in D_2 and *sex* in D_3, *work-class* in D_1 and *occupation* in D_2, *marital-status* in D_2 and *relationship* in D_3, *salary* in D_2 and *wage-per-hour* in D_3 are generalised into a single attribute value respecitively. We thus describe datasets D_1, D_2, and D_3 by 6-dimensional vectors:

$$\begin{aligned} x^1 &= \{1, 1, 1, 0, 1, 0\}, \\ x^2 &= \{1, 1, 0, 1, 1, 1\}, \\ x^3 &= \{0, 1, 0, 1, 0, 1\}, \end{aligned}$$

where 1 indicates that the dataset has the corresponding attribute, while 0 represents the attribute is not shown in the dataset. Consequently, every dataset can be represented by a binary numeric vector. We use x_j^l to denote the j-th element in the attribute vector for the l-th dataset.

Weights towards attributes. A limitation of k-means algorithm is that it treats all attributes equally, and does not respect different importance of the attributes. Indeed, the Euclidean distance involved in k-means algorithm does not consider the importance of each attribute in terms of re-identification risk. For example, the attribute DNA information is more capable of identifying a data subject than the postcode attribute as the former is more unique to capture a data subject's characteristics. Therefore, DNA information should be given higher importance or so-called weight. Let $W = [w_1, w_2, \cdots, w_m]$ denote the weight vector, and w_p denote the weight of the p-th ($p \in [1, m]$) attribute which is defined beforehand with domain knowledge. We have $w_p \geq 0$ and $\sum_{p=1}^{m} w_p = 1$.

Algorithm for datasets clustering. The input of the proposed clustering algorithm is a set of data points,

$$\mathbf{D} = \{x^1, x^2, \cdots, x^n\}$$

where each vector x^l is a m-dimensional binary vector representing the attribute information of each dataset l. The output is a set of k clusters, denoted by

$$\mathbf{O} = \{O^1, O^2, \cdots, O^k\},$$

and these clusters are depicted by k corresponding centroids

$$c = \{o^1, o^2, \cdots, o^k\}.$$

The algorithm starts by choosing k random points as the initial cluster centroids. Then each data point is assigned to the cluster whose centroid is the closest to the data point. Formally, data point x^i is assigned to cluster O^l if and only if:

$$\forall t \in [1..k], \quad t \neq l,$$
$$\sum_{j=1}^{m} w_j \cdot (x_j^i - o_j^l)^2 \leq \sum_{j=1}^{m} w_j \cdot (x_j^i - o_j^t)^2 .$$

After allocating data points, the clusters' centroids are updated by:

$$\forall t \in [1..k] \quad \forall j \in [1..m],$$
$$o_j^t \leftarrow \frac{\sum_{x^l \in O^t} x_j^l}{|O^t|}.$$

The algorithm operates iteratively by classifying data points into clusters and updating the centroids, until it reaches convergence. Finally, we use the normalised within-cluster sum of squares to denote the global linkability for each cluster:

$$GL(O^l) = \frac{1}{|O^l|} \sum_{x^i \in O^l} \|x^i - o^l\|^2 .$$

Intuitively, smaller global linkability indicates closer relation between datasets, thus it presents greater risk of having two or more records belonging to the same data subject within the cluster. We will discuss the selection of an effective parameter k in Section 3.4.

3.2.3 Measuring Local Linkability

Differently from global linkability which identifies content-similar datasets, local linkability focuses on the analysis of record level linkage in order to detect the records which are potentially associated with the same data subjects.

Record linkage analysis (Christen, 2012) is the most relevant to the measuring of local linkability. It is defined as a process of identifying records that corresponds to the same real-world entities across several datasets. It is widely used in data integration, data deduplication, etc. However, record linkage approaches (Abril, Navarro-Arribas, and Torra, 2012) have not been applied to data privacy scenario for analysing heterogeneous anonymised datasets which are anonymised by techniques, such as k-anonymity (Sweeney, 2002). We find that a small distance between records does not necessary indicates a high risk of de-anonymisation. This is because the anonymised datasets may have generalised records to guarantee that a data subject cannot be distinguished from other data subjects. Close records may be associated with more than

one data subject. Two examples illustrating the relationships between records distance and de-anonymisation risk are presented here.

- Small distance, low risk. Consider two records from two k-anonymised datasets where the parameter k is set as 3 and 5 respectively. If the distance between them is small, the risk can be still low because both records have 2 and 4 other records respectively sharing the same values that may belong to other data subjects. We denote this character as *group generality*;
- Big distance, high risk. Although two records may not share many common attribute values (i.e. low similarity on common attributes), if they have one identical attribute that is person-unique (such as DNA information, credit card number, etc.), the risk can be high. This character is denoted as *person uniqueness*.

We formalise the record linkage problem as an unsupervised clustering problem where records corresponding to the same entity form a cluster. The number of clusters can be very large if datasets contain a great number of data subjects. Moreover, each cluster can be generally very small, containing only few records if most records belong to different individuals. Therefore, instead of using k-means clustering, we deploy a k-members algorithm (Byun, Kamra, et al., 2007) to group the most likely linked k records together. Unlike the k-means algorithm, k-members requires that each cluster contains at least k records rather than specifying the number of clusters at the start of a clustering process.

As the datasets being grouped into a cluster after the first stage of clustering may still have different attributes, only overlapping attributes (i.e. identical or semantically similar attributes) are considered at the second stage of clustering. The rationale is that for the linkage analysis of heterogeneous records, the overlapped attributes are the main sources to connect records. We use the Euclidean distance to measure the similarity of two numeric attribute values:

$$d_N(v_1, v_2) = |v_1 - v_2|/|v_{max} - v_{min}|,$$

where v_1 and v_2 are two values in the numeric domain, v_{max} and v_{min} are the maximum and minimum in this domain respectively. For categorical values, we use the distance measure from (Byun, Kamra, et al., 2007):

$$d_C(v_1, v_2) = H(\Lambda(v_1, v_2))/H(\mathcal{T}_D),$$

where v_1 and v_2 are two categorical values in the categorical domain D, $\mathcal{T}_D$ is the hierarchy defined for D. $\Lambda(x, y)$ is the lowest common hierarchy level of x and y, and $H(\mathcal{T})$ represents the height of the hierarchy level $\mathcal{T}$.

weighted k-members. We modify the greedy algorithm proposed in (Byun, Kamra, et al., 2007) for k-members by changing the clustering criteria and introducing weights for variables. Given a set of overlapped attributes, suppose $\pi_{N_i}(i = 1, \cdots, p)$ is the index of an attribute with a numeric domain, $\pi_{C_j}(j = 1, \cdots, q)$ is the index of an attribute with a categorical domain, and a weight vector $W = [w_1, w_2, \cdots, w_{p+q}]$ for the overlapping attributes. Consider N_1 records from dataset D_1, N_2 records from dataset D_2. At the begining, a record r_i is randomly picked up from D_1 to form a cluster o_1. Then a record r_j from D_2 is selected to join o_1 by minimising $\sum_{r_i \in o_1} dist(r_i, r_j)$, where

$$dist(r_1, r_2) = \sum_{i=1,\cdots,p} w_{\pi_{N_i}} \cdot d_N(r_1[\pi_{N_i}], r_2[\pi_{N_i}]) + \sum_{j=1,\cdots,q} w_{\pi_{C_j}} \cdot d_C(r_1[\pi_{C_j}], r_2[\pi_{C_j}])$$

This process continues by selecting records from D_1 and D_2, until $|o_1|$ reaches k. When the process completes, another initial record which is the furthest from r_i is chosen to build another cluster in the same way. This clustering process is repeated until there are less than k records left. Finally, each remaining record will be examined and inserted into a cluster ensuring of which the distance sum is minimal. The choice of k will be discussed in Section 3.4. With respect to the *group generality*, we define the local linkability for two records r_i and r_j in the same cluster O as:

$$r_i \in D_1, \quad r_j \in D_2,$$
$$LL(r_i, r_j) = \frac{m_1 \cdot m_2 \cdot dist(r_i, r_j)}{|O|^2}$$

Note that m_1 and m_2 are the number of equivalent records which share the same attribute values with r_i and r_j in the cluster. $|O|$ is the size of the cluster. Intuitively, two records are more likely to be linked to the same data subject if the distance is smaller. However, this risk of re-identification is diluted through group generality by enlarging the distance using m_1 and m_2.

3.3 Privacy Risk Tree Model

To perform the privacy risk analysis, it is necessary to identify and define relevant elements, including risk source, privacy weaknesses, feared events and privacy harms, as well as to establish connections between them (De and Le Métayer, 2016). To conduct the risk analysis on publishing newly anonymised datasets, we formulate the risk sources as the previously published datasets, the privacy harm as the de-anonymisation of any data subject and illustrate privacy weakness and feared events in Table 3.1.

TABLE 3.1: Privacy weakness and feared events of dynamic data publishing

Types	Code	Description
Privacy weakness	V.1	overlapped attributes link datasets
	V.2	overlapped attributes are highly identifiable
	V.3	identical attribute values link records
	V.4	overlapped attribute values are person-unique
Feared events	FE.1	global linkability indicates a linkage between datasets
	FE.2	local linkability indicates a linkage between records

With these defined elements, we construct a risk tree as shown in Figure 3.1, which captures the de-anonymisation risk by considering both global and local linkability. We give a straightforward example to explain how the 'OR' and 'AND' rules are used in a risk tree. For example, if two datasets have a set of overlapped attributes birthdate, gender, marital-status, occupation, salary, the-last-four-digits-of-credit-card, this overlapping is considered as a privacy weakness (V.1) that results in the linkage of the two datasets. According to the 'OR' rule, the feared event FE.1 is triggered by V.1. Furthermore, if two record in the two datasets have an identical attribute value, for example, the same the-last-four-digits-of-credit-card, which is a V.3 and triggers the second feared event FE.2. At last, according to the 'AND' rule in the tree, since both FE.1 and FE.2 are presented, we can infer that the two records are most likely corresponding to the same individual.

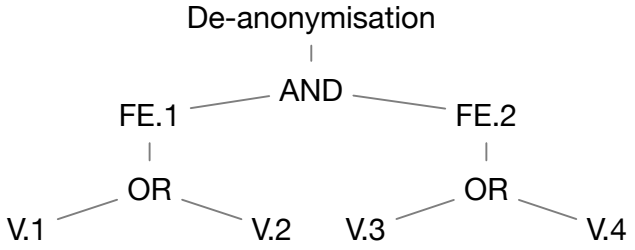


FIGURE 3.1: Privacy risk tree of de-anonymisating dynamically published datasets

We first convert the global and local linkability into the range of $[0, 1]$ as a likelihood metric. Formally,

$$p_{GL} = \frac{1}{1 + GL(O^l)},$$
$$p_{LL} = \frac{1}{1 + LL(r_i, r_j)}.$$

Then, the AND rule (De and Le Métayer, 2016) is applied to compute the de-anonymisation risk level, that is, the global and local linkability are integrated for the calculation. Formally, we have

$$p_{risk} = p_{GL} \times p_{LL}.$$

3.4 Experiments and Insights

3.4.1 Dataset Description

Our preliminary experiments with 10 datasets downloaded from the UCI machine learning repository (Dheeru and Karra Taniskidou, 2017) aim to evaluate the effectiveness of the proposed framework. The datasets include Student Performance Dataset (SUP), Open University Learning Analytics Dataset (OULA), Student Loan Relational Dataset (SLR), Adult Dataset (AUT), KDD Census-Income Dataset (CSI), Wholesale Customers Dataset (WSC), Online Retail Dataset (ORT), Haberman's Survival Dataset (HBS), Parkinson Disease Dataset (PKS), and Breast Cancer Wisconsin Diagnostic Dataset (WDBC). The reasons behind using the ten datasets for the experiments are two-fold. Firstly, these datasets are in the format of microdata, for example, healthcare data or census data. Such data is represented as a table where each row corresponds to one record of an individual. Each record has several attributes, which can be categorised as quasi-identifiers and sensitive attributes (e.g. salary and disease). Such dataset need to be anonymised before releasing and are suitable to be sanitised using k-anonymity. Secondly, these datasets have some common attributes and can show the pattern of clustering datasets according to their overlapped column attributes. Table 3.2 shows the number of column attributes and instances of each dataset.

Table 3.2: Ten UCI datasets for privacy risk mining

datasets	SUP	SLR	OULA	AUT	CSI	WSC	ORT	HBS	PKS	WDBC
#attributes	33	12	12	14	40	8	8	3	23	32
#instance	649	1000	30000	48842	299285	440	541909	306	197	569

3.4.2 Parameter Optimisation

We firstly extract 126 common attributes from these datasets, and represent each dataset using a 126-dimension binary vector. For the first-stage clustering where the k-means algorithm is used, the selection of an effective k is required before the start of the clustering. We utilise Silhouette coefficient (Rousseeuw, 1987) and Calinski-Harabaz index (Caliński and Harabasz, 1974) to determine whether a selected k is optimal. For both criteria, a higher score indicates better clusters that have small within-cluster dispersion and big between-clusters dispersion. In Figure 3.2, we utilise both Silhouette coefficient and Calinski-Harabaz index to determine an optimal k in the k-means algorithm. The value of k ranges from 2 to 9 as the number of datasets is 10. There is no point setting the value of k to 1 or 10 as that means no clustering happens, or each dataset forms an individual cluster. The Silhouette coefficient ranges from -1 to 1, where a higher value indicates that the clustering parameter k is more appropriate. In our experiment, the Silhouette coefficient gets its highest score when k is set to 4. The Calinski-Harabasz

index is the ratio of the between- clusters dispersion and inter-cluster dispersion. A high score means that clusters are dense and well separated. The Calinski-Harabasz index also gets its highest score when k is set to 4, which validates that the best value of k in our experiment is 4.

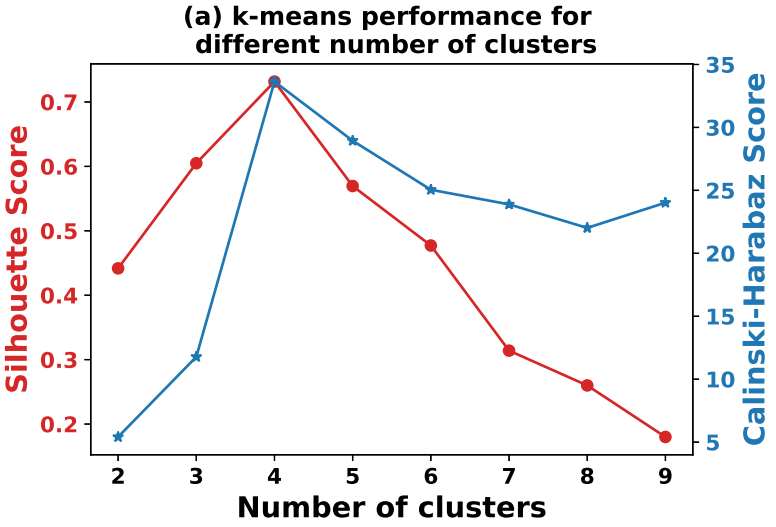


FIGURE 3.2: Parameter selection of first-stage k-means clustering

For the second-stage clustering, a weighted k-members algorithm has been implemented to analyse the record level linkability within a cluster. Note that as the algorithm works on any two datasets within a cluster, supposing n datasets have been clustered into the same group, the total time of executing k-members algorithm is $\mathcal{O}(\frac{n(n-1)}{2})$. We take the two datasets, SUP and SLR which are clustered into the same group for example. First, SUP and SLR are sanitised using 3-anonymity and 5-anonymity respectively. As there is no ground truth indicating if two records belong to the same individual or not, we artificially insert 100 synthetic records respectively into these two datasets, and evaluate the percentage of records clustered correctly into the same group with respect to different k. Figure 3.3 demonstrates how to decide an optimal k in the k-members clustering algorithm. As shown in the figure, the matching accuracy of records increases first when k ranges from 2 to 8 and drops when k ranges from 8 to 12. There is a trend that a value of k higher than 8 will cause the decline of the accuracy, so we stop at k=12. Therefore, we conclude that k = 8 achieves the highest percentage, that 80.1% of records are correctly clustered into the same groups. This is because if two datasets are protected by k_1- and k_2- anonymity separately, a match of records implies the matching of two equivalent classes. Indeed, the second-stage clustering locates the equivalent classes in which two linked records reside, but not the exact two records because of the implementation of k-anonymity. However, it is still helpful for data curators to pay more attention to the matched equivalent classes and with the 80.1% accuracy of identifying linked records, stronger protections can be applied to those equivalent classes in order to mitigate re-identification risks.

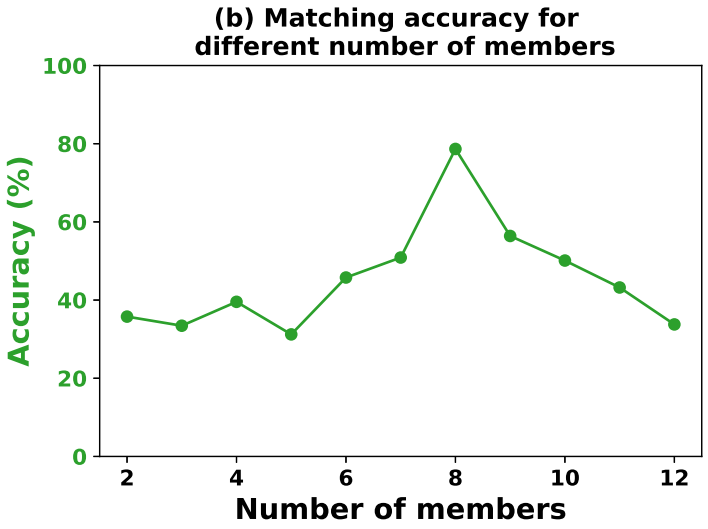


FIGURE 3.3: Matching accuracy of second-stage k-members clustering

The experimental analysis on ten datasets shows that the two-stage clustering framework can depict the pattern of the risk and the framework can scale to microdata since there are no specific requirements on the scheme of the dataset. Due to privacy concerns, microdata such as original demographic or healthcare data, are not readily available online. The ten datasets are the most suitable dataset in the UCI machine learning repository that we could have. The framework can scale to more datasets in the future when the data controller has more microdata to publish. The framework is practicable as long as the data scheme is not maliciously misleading. We believe this is safe, as the data curator will ensure the correctness to detect linked records. Also, the k-means and k-members algorithms are effective and can be easily implemented, which makes the framework easily deployed in reality.

3.5 Conclusion

This chapter presents a privacy risk mining framework to identify potential linked records among heterogenous anonymised datasets, based on the proposed two-stage clustering algorithm. As demonstrated by experiments, our framework is effective in locating equivalent classes that contain the records associated with same individuals. This empowers data curators to envisage the re-identification vulnerabilities and apply stronger measures. In future work, we plan to improve the matching accuracy by extracting more representative information of datasets in the first-stage clustering and investigating alternative machine learning algorithms. We also plan to conduct more comprehensive privacy risk analysis by refining the risk tree model and improving the calculation rule for risk likelihood level analysis.

Part II

Proposing the Solution

Chapter 4

Differential Private Data Sharing with Blockchain

Through the analysis of various anonymisation techniques in Chapter 2, we realize that differential privacy may be the most possible GDPR-compliant anonymisation technique for data sharing between data controllers and data owners. Anonymisation services that obfuscate sensitive datasets under differential privacy have been proved effective to support secure data-sharing among the two parties (McSherry, 2009). By providing an analysis result query interface to the data controller, the data owner can share the statistical information contained in the data set to data controller. Sharing obfuscated results guarantees the privacy of the dataset records.

Although differential privacy provides strict mathematical protection for privacy. The existing differential privacy management has deficiencies. For example, when the so-called privacy budget is used up, it simply stops data sharing. This is more compelling in a multi-query scenario, because privacy budget will be consumed quickly, and how to maintain privacy amounts to controlling the allocation of privacy budget is also a question that needs careful consideration.

In this Chapter, we focus on improving privacy management to maximize the data utility of differentially private data sharing and at the same time supporting a transparency-by-design, privacy-budget-evident data sharing mechanism via blockchain technology. The blockchain-based approach enables data owners to control the privacy preference and enhances the security of the anonymisation services. More specifically, the smart contract in the blockchain validates the usage of privacy budget of differential privacy and adaptively changes budget allocation, depending on the privacy requirements provided by data owners. The implementation with the Hyperledger permissioned blockchain validates our approach with respect to privacy guarantee and practicality.

4.1 Privacy Management of Data Controller

4.1.1 Traditional Privacy Management System

In a typical privacy preserving data sharing scenario, there are usually three kinds of participants, data owner, data controller and data consumer. Usually the data controller collects data from the data owners. These data may be a single individual record, or may be a data set generated by the data owner. These data owners are also data subjects under the GDPR context who enjoy various data privacy rights. The data controller is an intermediate party that implements data anonymisation services and provides the aggregated and anonymised data to the data consumer so that the data consumer can get desired information from the data. This process allows data controllers to collect data and provide a more accurate analytic result to data consumers based on a more comprehensive data resources. It builds up interconnectivity and cooperation among data owners, data controllers and data consumers, enabling them to achieve various business goals, such as controlled sharing of data, and optimisation of resources usage.

However, such a model poses a risk to the privacy of the data owner. When anonymisation is carried out on the data controller side, by outsourcing data protection to the controller, data owners lose control over their data, raising significant privacy management challenges, because the data owner cannot determine whether the data controller has implemented the prescribed anonymisation techniques. This kind of centralized anonymisation service centered on the data controller also faces the risk of single point of failure. When the data controller's anonymisation service has vulnerability, the privacy of all individuals included in the data set may be violated.

A second thought is whether the implementation of anonymisation service can be moved to the data owner side. However, when anonymisation is performed on the data owner side, not every data owner has the ability to provide sufficient anonymization protection for the data or aggregate enough data to provide sufficient data utility.

When implementing anonymisation service, a common method is to use the differential privacy mechanism to obfuscate the result of statistical queries towards a sensitive dataset, enabling its privacy-preserving sharing. We will describe the principle of differential privacy shortly in Section 4.2.1. A prominent problem with this approach is privacy management. Usually sharing will stop when the so-called privacy budget is used up. This termination of sharing undoubtedly limits the utility of the shared data. And this process requires monitor and allocate the privacy budget usage. This puts forward higher requirements on the privacy protection ability of the data controller.

Offering the anonymisation service has benefits for the data controller — accessing to multiple data sources and providing services to data consumers — but raises significant challenges for privacy management: sensitive datasets from multiple data owners each of

which has different privacy requirements, the anonymisation services may not be trusted by data owners. Traditional solutions for privacy management have insufficiency as-is in the hand of data controller when the controllers collect data from multiple data owners. Firstly, typical management of privacy and data utility requirements (Fung et al., 2010) must be extended to support multiple datasets and data owners. Data owners may have different privacy and data utility requirements on their own datasets. Existing proposals do not allow data owners to define such requirements and are not able to incorporate them and obfuscate data accordingly. Secondly, to protect anonymised data from degradation of privacy protection (e.g. due to dataset repeatedly queried and linking attacks), recent approaches (Kellaris et al., 2014; McSherry, 2009) proposed to verify *privacy budget* which determines the amount of noise produced in the obfuscation process, and control the amount of noise produced in the obfuscation process, stopping data sharing as soon as the budget is used up. However, stopping sharing data must be avoided as much as possible to not hamper the goal of making the most use of the data. More importantly, multiple data owners are not able to customizing its privacy requirements corresponding to the data they provided to the data controller's anonymisation services. Due to the lack of trust among data owners and data controllers, anonymisation services themselves should offer adequate guarantees for controlling and tracing privacy budget, otherwise they will end up being single point of attack to make multi-query de-anonymisation attacks possible (Dwork, 2006).

In the following, we introduce a motivating example to better illustrate the limitations of existing approaches and the privacy degradation of a multiple-query scenario.

4.1.2 Privacy Degradation in A Motivating Example

Assume that a dataset containing employees' absence information is collected from multiple data owners by the data controller for the sharing purpose. A data consumer sends a statistical data request (e.g., a Mean query) and receives obfuscated query result by the anonymisation services deployed by the data controller. While the controls on how data from different owners are integrated and accessed are outside the scope of this chapter, our focus here is on the privacy protection mechanism employed to anonymise the dataset.

The dataset, shown in Table 4.1, contains privacy-sensitive information, such as salary and number of absence, and the data owner wants to prevent the leakage of the sensitive information. To this aim, state-of-the-art anonymisation service based on Differential Privacy (Dwork, 2006) is used. Intuitively, it relies on an ϵ parameter setting the privacy budget of a given dataset. Based on the ϵ, it generates randomised noise to the query result.

TABLE 4.1: A motivating example of sensitive data - employee dataset

Employee ID	Date of Birth	Absence	Salary
1	08/11/1955	16	23112.30
2	22/07/1953	3	25388.43
3	01/01/1966	1	17303.11
...	...	...	...

Privacy requirements. Let us assume data consumers invoke data queries about the mean of employees' salary. The privacy budget ϵ can be set according to their privacy requirements, e.g. 0.1 or 0.5 for, respectively, stronger and weaker privacy protection that means different noise levels on obfuscated results. Figure 4.1 graphically shows the noise of obfuscated results over 20 queries (results correspond to the critical points of the lines in the figure). With ϵ set to 0.5 results (dotted line) are closer to the actual ones (solid line), while with ϵ set to 0.1 results (dashed line) differ more offering stronger protection, but less utility.

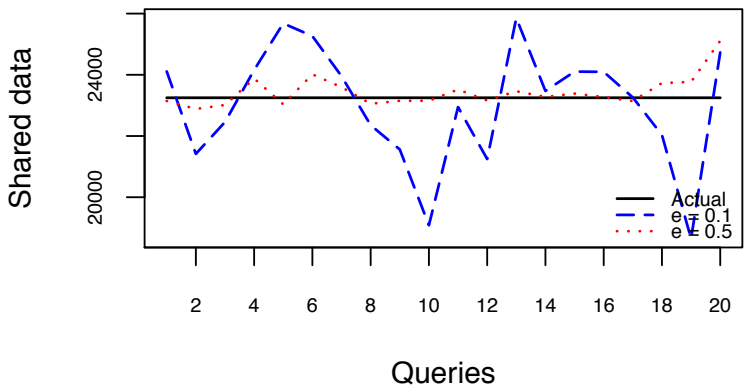


FIGURE 4.1: Obfuscated query results with different privacy requirements

Privacy degradation. Differential privacy suffers from privacy degradation as the number of queries increases. Sharing a single query result guarantees chosen privacy level (e.g., 0.1 or 0.5 in this example), making difficult to determine what the actual average salary is. However, if we combine multiple obfuscated results (i.e., each of the lines in the figure), the privacy level degrades by cumulating ϵ over queries (Dwork, 2006). For instance, executing 20 queries indicates 20ϵ-differential privacy. This can be visualised in Figure 4.1 as the sum distance between the points on the dotted line with the actual one tends to equalise, hence revealing the actual value. Furthermore, if more query types are allowed, that is, data consumers send not only *Mean* queries but also *Max, Min, 3rd Quatile, Sum* queries, then the data consumers can easily learn much more information about the dataset by sending queries continuously.

Recent proposals (Kellaris et al., 2014; McSherry, 2009) suggest to check the budget request (i.e., ϵ) for each query and reject queries when the consumed budget exceeds a threshold. However, in a setting with a high number of queries, this leads to stopping

sharing data quite soon. In order to serve more data consumers, data controllers are reluctant to halt the anonymisation service as stopping will prevent the realisation of business goals. A malicious data controller may keep sharing the data although the privacy budget is used up, which will result in the privacy breach. Additionally, as budget may be consumed by multiple data consumers and the data themselves refer to multiple data owner, the budget management cannot be entrusted to a single anonymisation service itself. Instead, it requires adequate integrity and accountability guarantees such that all involved parties can rely on it.

All in all, current privacy management solutions for differential privacy can be improved to enhance the integrity and accountability of the anonymisation service implemented by the data controller. Also, data owners should be given more freedom to choose their privacy requirements.

4.1.3 Blockchain-based Privacy Management

The main reason behind using the blockchain technology is to support a transparency-by-design, privacy-budget-evident differentially private data sharing mechanism. As the differential private anonymisation service is offered by the data controller, data owners cannot fully trust the anonymisation service, especially the management of privacy budget. Indeed, budget management must provide integrity and accountability guarantees. These guarantees enhance assurance on the anonymisation services and tracing of privacy degradation levels, i.e. budget consumptions. Due to the distribution and lack of trust among data owners and data controllers, centralised budget management does not offer adequate guarantees. Realising trustworthy decentralised management of the privacy budget is then of paramount importance to ensure privacy protection of sensitive datasets and, most of all, to enhance assurances on the anonymisation services. To this aim, we introduce here a new solution based on blockchain, an innovative technology that ensures full decentralised control on data and code execution. Thanks to the consensus algorithms and tamper-resistant transaction, blockchain enjoys some data integrity-related properties, for example, transparent and decentralised control of the data, non-repudiation and persistency of the public ledger.

In summary, the *blockchain-based approach for privacy-preserving data sharing* allows data owners to control the anonymisation process, such as defining their own privacy and data utility requirements, tracing the data-sharing activities, and enjoying a secure services supported by the outsourced anonymisation services.

Specifically, the main contributions of this approach are the followings:

- A blockchain-based data sharing approach to *store*, *verify* and *adaptively allocate* privacy budget consumptions via autonomous smart-contracts according to data owner privacy and data utility requirements.

- A high-level system architecture enabling the integration of any data anonymisation service with any smart-contract blockchain solution.
- Implementation and evaluation by means of the Hyperledger Fabric blockchain, and discussion on privacy and data utility enhancements.

4.2 Differential Privacy Meets Blockchain

4.2.1 Differential Privacy

Differential Privacy (Dwork, 2006) is proposed as a privacy technique for protecting individual records of statistical databases. It ensures that adding (or removing) a single record to (or from) a database does not significantly change the outputs of statistical queries. This is usually achieved by designing a mechanism that adds randomised noise to the query output, so that an adversary is not able to determine whether a targeted record is included in the database or not, no matter what side information the adversary might have. To present our approach, we first present the key concept and implementation mechanism of differential privacy.

Definition 4.1. (ϵ-Differential Privacy (Dwork, 2006)). A randomised mechanism $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{D}|}$ is ϵ-differentially private if for every set of outputs $\mathcal{S} \subseteq \mathit{Range}(\mathcal{M})$ and all databases $D, D' \in \mathbb{N}^{|\mathcal{D}|}$ that differ in one record,

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(D') \in \mathcal{S}] .$$

With any mechanism that generates differentially private query outputs, the probability of $\mathcal{S}$ being the output is nearly the same for D and D' such that an individual record is protected by including it in D but not D'. A popular technique that satisfies Definition 4.1 is the Laplace mechanism (Dwork, 2006), which generates noise using Laplace distribution. The Laplace distribution is centered at zero and has a scale parameter b depending on the l_1-norm sensitivity of queries. The l_1-norm sensitivity of a query is the maximum difference of the query results based on two databases which differ in one record.

Definition 4.2. (l_1-Norm Sensitivity (Dwork, 2006)). The l_1-norm sensitivity S_q of a query $q : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ for all databases D and D' which differ in one record is:

$$S_q = \max_{D,D'} ||q(D) - q(D')||_1$$

where $|| \cdot ||_1$ denotes l_1 norm, and q represents a numeric query function, which maps databases to k real numbers. Without loss of generality, we focus on $k = 1$ situations.

With the scale parameter b computed as S_q/ϵ, the Laplace mechanism can be defined as follows.

Definition 4.3. (Implementing ϵ-Differential Privacy: The Laplace Mechanism). Given any query q, the Laplace mechanism is $q(D) + y$ where y is a random variable drawn from the Laplace distribution with scale parameter $b = S_q/\epsilon$ where S_q represents the l_1-norm sensitivity (Dwork, 2006) of the query q, and location parameter $\mu = 0$.

$$f(y) = \frac{1}{2b} \exp\left(-\frac{|y|}{b}\right).$$

The variable y expresses how much noise should be added to the query outcome. The smaller the ϵ or the greater S_q, the greater noise generated for achieving ϵ-differential privacy. We use $Lap(\epsilon)$ to denote the randomised noise generated by the Laplace mechanism.

An important property of differential privacy is the *composition* property, which shows how privacy degrades when the number of queries on the same database increases.

Lemma 4.4. *(Composition (Dwork, 2006)). If $\mathcal{M}_1$ is ϵ_1-differentially private, and $\mathcal{M}_2$ is ϵ_2-differentially private, then let $\mathcal{M}$ be another mechanism that executes $\mathcal{M}_1$ and $\mathcal{M}_2$ independently on a database, $\mathcal{M}$ is $(\epsilon_1 + \epsilon_2)$-differentially private.*

According to Lemma 4.4, the privacy level degrades linearly from ϵ to $n\epsilon$ when executing n queries on the same database if ϵ-differential privacy is guaranteed for each query. The privacy level $n\epsilon$ represents how much information is leaked after n queries. The total allowed leakage is commonly viewed as *privacy budget*, and each query consumes part of the budget (e.g., ϵ in this example).

Another important property is that differential privacy is immune to *post-processing*.

Lemma 4.5. *(Post-Processing (Dwork, 2006)). If $\mathcal{M}$ is ϵ-differentially private, then $g \circ \mathcal{M}$ is also ϵ-differentially private for any g where g is an arbitrary randomised mapping.*

This property allows the noised outcome of a query being processed by data analysts, and guarantees that the level of privacy will not degrade if the data analysts have no additional knowledge about the private database.

4.2.2 Blockchain and Smart-Contracts

Blockchain is a novel technology that recently came to prominence when used as a public ledger for the Bitcoin cryptocurrency. It consists of consecutive chained blocks, replicated and stored by the nodes of a peer-to-peer network. Blocks are created in a

decentralised fashion by means of a consensus algorithm, which can range from expensive proof-of-work mechanism, e.g., Bitcoin's, to lightweight Byzantine consensus algorithm, e.g., Hyperledger's (www.hyperledger.org). The use of consensus algorithms enables several data integrity related properties in blockchain, such as distributed control of the data on the chain, non-repudiation and persistency of transactions and data provenance.

Differently from Bitcoin, new types of blockchains have recently appeared featuring *smart-contract*s, that is, programs deployed and autonomously executed on the blockchain. Being part of blockchain makes contracts and their executions *immutable* and *irreversible*. The state-of-the-art smart-contract blockchains are Ethereum (www.ethereum.org) and Hyperledger Fabric. Our implementation relies on the latter due to its performance and flexible architecture. Hyperledger Fabric is a permissioned blockchain that can jointly unite a consortium of recognised participants, that is data owners and data consumers in our cases, each of whom does not necessarily trust the others. The network is private, and each node must be approved to become a member of the consortium. Unlike a public blockchain, such as Bitcoin, that allows anyone to join the network anonymously, nodes in permissioned blockchain are not anonymous but with identities approved by the membership service. Once entities in the blockchain network, especially data controllers, take any malicious activities, the membership service can take actions to prevent these malicious entities from participating in the blockchain. A permissioned blockchain is necessary to eliminate the trust issues among data owners and data controllers where both of them need to know the identities of each other. A permissioned blockchain also guarantees that only approved entities can involve in the data sharing scenario. We utilise Hyperledger Fabric to implement our framework as it is a widely-used and actively-studied permissioned blockchain system which does not require a native cryptocurrency to support the running of the system. Instead of employing expensive consensus protocol such as proof-of-work, Fabric runs a fast byzantine-fault-tolerant consensus algorithm to decide on the next chaining block.

4.3 Blockchain-based Data Sharing

The objectives of our blockchain-based data sharing approach are two-folds: allowing data owners to control their own privacy requirement and helping data controllers scrutinize the anonymisation processes. In particular, the privacy levels can be customized when using third-party anonymisation services especially to protect against multi-query attacks. As discussed in the motivating example, secure management of privacy budget is the key to ensure privacy. Our approach utilises blockchain smart-contracts to store, verify and adaptively allocate privacy budget consumptions depending on data owner's privacy and data utility requirements. We consider the goal of adversary is to degrade privacy, and the adversary has the capability of collecting all query outcomes.

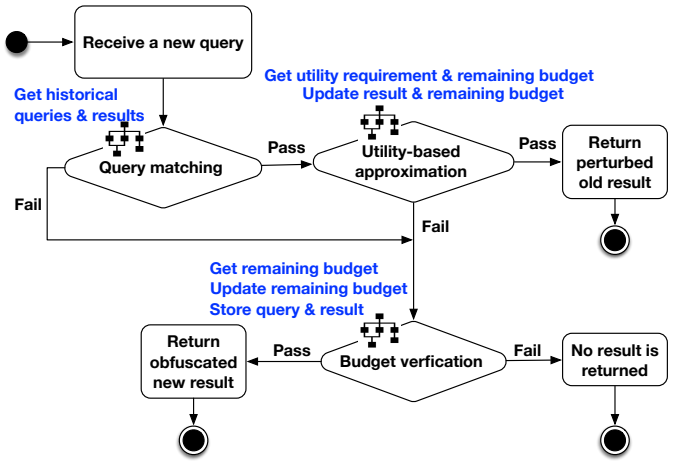


FIGURE 4.2: Overview of the runtime mechanism $\mathcal{M}_i$ (diamond boxes relies on smart contracts)

Our approach is based on two phases: *Setup*, setting up requirements for privacy and specifications for data utility, and *Runtime*, to dynamically allocating privacy budgets and tuning the data sharing process. Blockchain smart-contracts are used to store, evaluate and keep track of *historical queries* and *privacy budget*.

In the following, we discuss first the high-level activities integrating blockchain and data sharing, then outline the proposed system architecture.

4.3.1 Main Components and Phases

At the Setup phase, data owners specify their privacy and data utility requirements and then store them in the smart-contract. The privacy requirement is represented by the privacy budget ϵ_0, which represents the maximum amount of budget allowed on sharing data. According to data owner preferences, the budget can be associated to one or many datasets, or even to single columns of a single dataset. Data utility requirement is represented by a numerical variable, denoted by $u \in \mathbb{R}_{\geq 0}$, representing the maximum amount of noise allowed on the actual query result, thus to maintain adequate data utility.

At the Runtime phase, data queries are managed returning, when allowed by the privacy budget and requirements, anonymised results. Indeed, our approach $\mathcal{M}$ consists of an unbounded sequence of mechanisms $\mathcal{M}_1, \mathcal{M}_2, \cdots$, where $\mathcal{M}_i$ operates when the i-th query is received. Figure 4.2 illustrates the activities involved in each mechanism $\mathcal{M}_i$. Logically, it can be decomposed into three main test activities (i.e. the diamond boxes in the figure): *Query matching*, *Utility-based approximation* and *Budget verification*. The activity flow is reported in the figure and relies on data and computation offered by smart-contract.

4.3.1.1 Query Matching

This activity aims at determining whether a newly received query has been executed before. Smart-contract checks the sharing history stored on the blockchain. Formally, the sharing history amounts to the following tuple

$$(DsetId, \epsilon_r, [(qry_1, res_1), \cdots])$$

where *DsetId* is a reference to a dataset (or a column of it), while ϵ_r is the remaining associated privacy budget. The following list of tuples forms the sharing history. Each tuple (qry_1, res_1) has as first element the *function type* of the query—e.g. sum, average, max, min— and as second the corresponding previously released result. Thus, res_i are just the latest released results for each query type i, hence lightweight information whose limited size makes them suitable for blockchain storage.

The query matching compares a newly received query *qry* on the dataset referred by *DsetId* with the corresponding history. The query is denoted by the tuple $(DsetId, qry, \epsilon)$, where the parameter ϵ denotes the requested budget for executing the query. The value of ϵ can be provided by data consumer, or pre-defined as a fixed value by anonymisation services. Without loss of generality, we assume function types fixed and comparable by names; additional comparison parameters can be set as well. Namely, given a *DsetId*, the test is passed when *qry* is equal to a qry_i part of the history. Notably, to keep queries private to all the members part of the blockchain, the history data can be stored hashed. The comparison will be then on hash texts.

4.3.1.2 Utility-based Approximation

This test aims at checking whether a previous released result can approximate the result to return for the current query. The test pseudocode is reported in Algorithm 1. Intuitively, it checks if dataset changes—e.g. new or deleted records— affect the utility of previous results. Indeed, it decides if the actual query result *res* can be approximated with the previously released res_{old}, given matching data utility requirements u, and enough remaining privacy budget ϵ to compute the approximation test at the cost σ.

Firstly, it checks whether the remaining budget is enough for executing the test (Line 1), If yes, it produces an obfuscated version of the old result res_{old} using a very small amount σ of the privacy budget (Line 2). Otherwise, it returns *false* (Line 11) stating the approximation test failed. The computed obfuscated result is compared by a smart-contract with the actual one with respect to the threshold u (Line 3). If the approximation test passes, the new obfuscated result is set as the last returned result of such query (Line 4), the budget is updated accordingly (Line 5) and the approximated

Algorithm 1 Utility-based approximation Pseudocode

Input: res, res_{old}: actual and previous query results
u: data utility requirement; ϵ_r: remaining budget
σ: a small budget for performing approximation test.
Output: res' if passes; boolean value *false* otherwise.

```
1: if ε_r ≥ σ then                                  ▷ Checking budget for the test.
2:     res' = res_old + Lap(σ);                     ▷ Obfuscating old result.
3:     if |res' − res| ≤ u then
4:         res_old = res'                           ▷ Updating history in blockchain.
5:         ε_r = ε_r − σ                            ▷ Updating budget in blockchain.
6:         return res'.
7:     else
8:         ε_r = ε_r − σ                            ▷ Updating budget in blockchain.
9:         return false.
10:    end if
11: else
12:    return false.
13: end if
```

result is returned (Line 6). Otherwise, only the budget is updated (Line 7, 8) to keep tracking that σ was consumed by the approximation test.

When this approximation test succeeds and res_{old} is used, the consumed budget σ is significantly less than that (i.e., requested budget ϵ) used for returning the actual *res*. The obfuscation added to res_{old} aims at adding randomness to the utility test. This permits dealing with the fact that adversaries may know how the test works and attempt to gain knowledge about the actual result *res* from the test result.

4.3.1.3 Budget Verification

The budget verification test is triggered if there has been no same query executed (i.e., the query matching test failed), or the query result cannot be approximated (i.e., the approximation test failed). Thus, a new result has to be computed, as long as the remaining privacy budget is enough.

This test is carried out on a smart-contract that, given a query tuple $(DsetId, qry, \epsilon)$, compares the remaining budget ϵ_r of the dataset *DsetId* with the requested budget ϵ. If the test succeeds, the anonymisation service generates randomised noise under differential privacy to add to the actual query result consuming the requested budget. Otherwise, the query is rejected because it would violate the defined data privacy requirement.

According to Lemma 4.4, the ensured privacy level degrades as the number of queries increases if the noise is generated independently over queries. The activity of budget verification ensures the satisfaction of pre-defined privacy requirement ϵ_0 as it makes sure the consumed budget does not exceed ϵ_0. Formally, we have the following result

Theorem 4.6. *Our approach $\mathcal{M}$ satisfies ϵ_0-differential privacy.*

The mechanism updates the remaining budget, query function type (if the received query has not been stored before), as well as the generated new result in the blockchain.

4.3.2 System Architecture

To implement the proposed approach, we propose a generic system architecture for blockchain-based data sharing. Specifically, an Anonymisation Interface (AI) is realised to integrate pluggable differential privacy component with blockchain smart-contracts.

As illustrated in Figure 4.3, federated datasets and anonymisation services (denoted by ANM) are integrated via AIs, which act as mediator with blockchain smart-contracts realising the control flow in Figure 4.2 previously described. Data consumers interact with any AI to query datasets. Then blockchain smart-contracts execute the test activities, i.e., Query matching, Utility-based approximation and Budget verification, to ensure privacy protection.

Data owners federating their sensitive datasets to a data controller can then *trust third-party anonymisation services* due to the principled exploitation of blockchain smart-contracts. They store and evaluate sharing history, while enforcing utility and data privacy requirements. Non-repudiable evidences of privacy budget consumption and released query results enhance the security guarantees on privacy-preserving data sharing processes. In particular, blockchain smart-contracts carry out the secure management of privacy budget and carry out the test activities. The third-party anonymisation services only execute when there is no previously released result that can be used. This prevents attacks of altering, deleting budget consumptions, and improves the availability of anonymisation services.

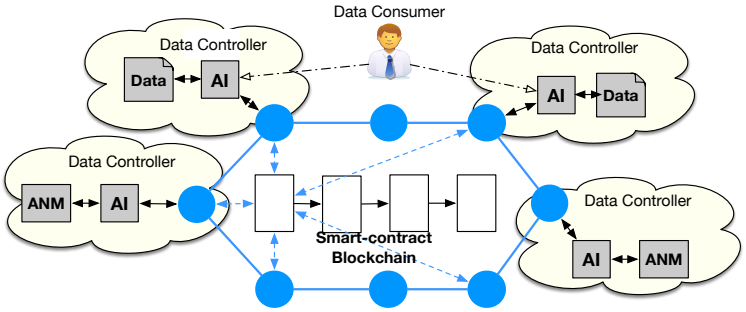


FIGURE 4.3: Blockchain data sharing System among Data Controllers (AI stands for *Anonymisation interface*, while ANM stands for *Anonymisation service*)

4.4 Experimental Evaluation

We prototyped our blockchain-based data sharing approach by the Hyperledger Fabric smart-contract blockchain and a traditional implementation of differential privacy using Laplace mechanism. A real-world dataset from the Italian Ministry of Economy and Finance is used, which contains employees' salary information as exemplified in Table 4.1.

Our implementation is in Hyperledger Fabric V0.6 on a 2.6 GHz 4 core Intel Xeon laptop running Ubuntu 14.04.5. The dataset identifiers are set on single columns, for which data owners specify privacy and utility requirements.

The experiments aim at evaluating, on the one hand, privacy and data-utility guarantee and, on the other hand, blockchain practicality. Specifically, the query function types are four—i.e. sum, average, max and min. The total privacy budget ϵ_0 is set to 10, and the requested budget ϵ for each query is fixed at 0.5. In reality, data owners can set ϵ according to their privacy requirements, e.g. 0.1 or 0.5 for, respectively, stronger and weaker privacy protection that means different noise levels on obfuscated results. For simplicity and without losing generality, we fixed ϵ to 0.5 for a better data utility. Note that, this setting does not violate any differential privacy protection, but only affect the total number of queries that can be requested by data consumers. The total privacy budget ϵ_0 is set to 10, which is a typical value used to support multiple queries and achieve a balance on privacy protection and data utility. The data utility requirement u is set to 1500. The value depends on the utility preference of data consumers for the scenario of salary query. Queries are simulated continuously and randomly by uniformly choosing a query type from those four. The compared baseline approach is the standard differential privacy mechanism that generates randomised noise independently for each query.

4.4.1 Privacy

In this chapter, the privacy of the data, that is the confidentiality of the data, is protected during data sharing process. However, the privacy of the user, that is the anonymity of the user is not considered and beyond the scope of this chapter. As proved in Theroem 4.6, our approach always provides ϵ_0-differential privacy. That is, the consumed privacy budget does not exceed ϵ_0 that the pre-defined privacy requirement is satisfied. In this experiment, we focus on how the budget is consumed over queries. Figure 4.4 shows the budget consumption when our and the baseline approach are implemented. It is clear that the remaining budget decreases linearly in the baseline approach, so that the budget is used up after 20 queries. The remaining budget in our approach decreases slower than in the baseline approach. Specifically, for the first query, it drops the same in both approaches as there has been no sharing history and no result can be approximated. From the second query, the decrease slows down as historical sharing

results become available for approximation at some queries. More specifically, after receiving 6 queries, the historical sharing tuple stored in the blockchain becomes

$$
\begin{aligned}
&(DsetId = EmployeeDset, \epsilon_r = 7.93,\\
&[(qry_1 = \max, res_1 = 28643.57),\\
&(qry_2 = average, res_2 = 23147.29)])\\
&(qry_3 = \min, res_3 = 16127.25)])\\
&(qry_4 = sum, res_4 = 578106.25)])
\end{aligned}
$$

where all four query types have been received and stored for future approximation.

We now change the number of query types from four to two (i.e., max and average) representing the situation where two query types are allowed. The consumption of privacy budget is plotted together with the situation of four query types. Indeed, the fewer query types, the less the budget is consumed: it is more likely that historical sharing results can be used to approximate new results. Therefore, our approach is able to allow more queries executed and is more effective when there are fewer query types.

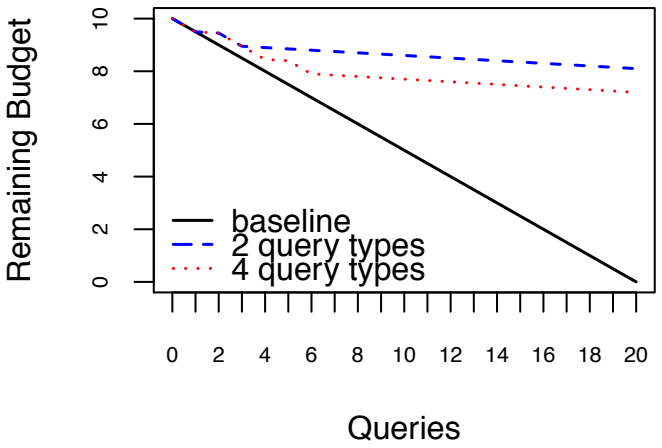


FIGURE 4.4: Budget consumption as the number of queries increases, where "2 query types" means that just max and *average* queries are allowed, while "4 query types" also includes min and *sum* queries.

4.4.2 Data Utility

We plot the noise generated at each query in Figure 4.5. Our approach introduces less noise compared with the baseline approach after receiving more queries, as the approximation test takes into account the utility requirement, guaranteeing the amount of generated noise is bounded by $u = 1500$. We compute the mean of the absolute noise over 20 queries, and have 43331.823 for the baseline approach, 3864.97 and 3806.47 for our approach with respectively four and two query types. Therefore, our approach

provides slightly better data utility, and the number of query types does not affect the data utility.

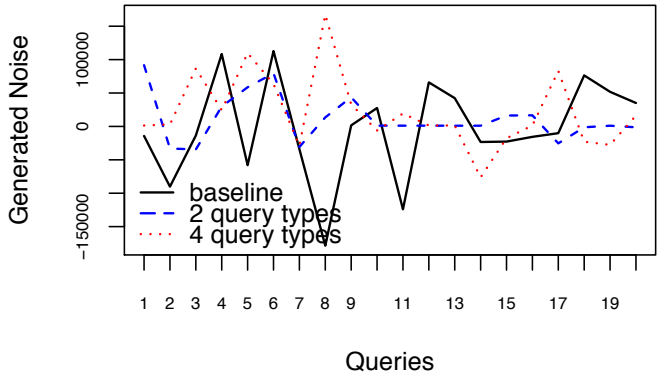


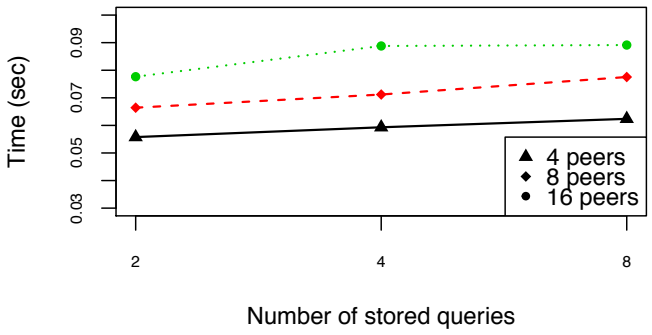
FIGURE 4.5: Generated noise over 20 queries in baseline approach and our approach.

4.4.3 Blockchain Practicality

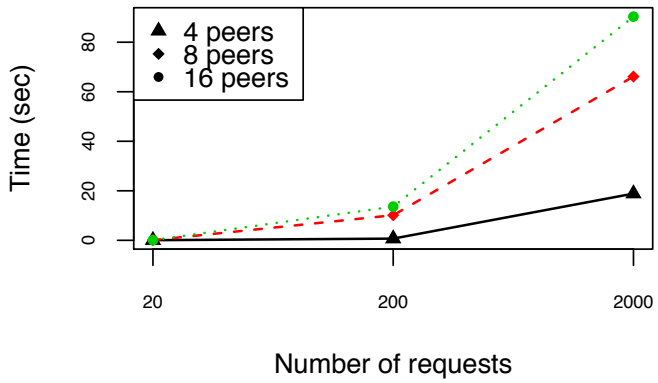
Storage capacity. As the data sharing history stored on the blockchain records only the latest released result for each query type, rather than a full list of release results, the size of the tuple that gathers such sharing history is suitably small and can be optimised grouping by query types. Tuples can be illustrated, e.g., as $(DsetId, \epsilon_r, [(qry_1, res_1), (qry_2, res_2)])$ if two query types are allowed. Therefore, the design of history tuple is light and suitable for blockchain storage as testified by the extensive tests.

Blockchain performance. The performance of smart-contract computation is shown in Figure 4.6a when the number of stored query types changes from 2 to 4 and 8. The computation on Hyperledger Fabric is very efficient as the maximum time is 0.09 second. There is a slightly increase in time when the size of the sharing history tuple increases. As the number of peers deployed in the Hyperledger blockcahin increases, the computation time increases. This is because it takes more time to allow all peers to confirm the computation result (i.e., adding a new block). We now simulate more query requests at a single timestamp, particularly from 20 to 200 and 2000 requests. As shown in Figure 4.6b, the time increases and the maximum time becomes 90 seconds when there are 2000 requests received at the same time. Therefore, implementing our approach in Hyperledger Fabric offers good performance, and is able to handle great number of requests.

Permissioned blockchain. Our prototype implementation relies on Hyperledger Fabric, which enables us to deploy a *private* blockchain with control on operating users. Data owners are the default users who are allowed to access the blockchain but only manage the anonymissation process of their own datasets. Additional access rules can be negotiated with data owners supported by the function of smart-contract.



(A) number of allowed query types



(B) workload

FIGURE 4.6: Smart-contracts performance regarding different number of stored queries and requests.

4.5 Related Work

A considerable body of research has been devoted to address the data privacy issues in the hand of the data controller. Because of the openness and not fully trusted characteristic of data controllers in the data sharing process, traditional privacy-preserving approaches, such as anonymisation techniques (M. YANG, Sassone, and O'Hara, 2012) by their own cannot ensure the protection of personal data. Cryptographic approaches (Esposito, Castiglione, and Choo, 2016; Wang et al., 2010) have been proposed to encrypt data before sharing to the data controller, and data can only be decrypted by authorised data consumers. These approaches rely on novel access control models to support various access request from data consumers (D. CHEN and Zhao, 2012).

In order to equip data owners with more control and accountability over data protection, blockchain-based proposals (Ekblaw et al., 2016; Zyskind et al., 2015) utilise blockchain to store data and control data sharing as a data management platform. More specifically, Enigma (Zyskind et al., 2015), a peer-to-peer network supports different parties to jointly store and run computations on data while guaranteeing the privacy of data. This proposal combines blockchain with multi-party computation techniques and examines a mobile application data sharing scenario. The other proposals, such as (Ekblaw

et al., 2016) aim to protect patients health records, and ensures the immutable, quick access, confidential properties of such data storage and access. While these approaches focus on storing sensitive data directly on blockchain, our solution stores the process of anonymisation services which provides stronger data privacy guarantee and requires only light configuration for implementing our solution.

4.6 Conclusion

Our blockchain-based data sharing approach allows data owners to control the privacy protection of their datasets while enjoying the anonymisation services provided by the data controller. Future work includes examining practical deployment issues of the data controller, integrating with security components (e.g., access control) and developing an effective user interface to support the control of the anonymisation services.

Chapter 5

Outsourcing Differential Privacy Sanitisation using Blockchain and Encryption

In the previous chapter, we provide a differential privacy management system which enables data owners to control their privacy preferences by themselves. The system focuses on outsourcing the privacy budget allocation service of differential privacy to the data owners' nodes in the blockchain. A group formed by data owners manages the privacy management of the differential privacy mechanism. Therefore, the risk of privacy management breach due to single-point-of-failure is eliminated. Besides, the built-in differentially private data sharing mechanism is optimised for better data utility.

However, in the previous system, the sanitisation service needs to be performed off-chain by the data controller as blockchain transactions are public. The input data of smart contracts are visible to all the nodes in the blockchain, the private data to be anonymised has to be processed locally. The anonymisation process is centralised in the previous system. In this chapter, we try to resolve this problem by outsourcing the anonymisation process to the blockchain, which is a challenging task as the privacy requirement of private data processing and the transparency of blockchain transactions are contradictory by nature. To solve the conflict, we apply a two-layer model using blockchain and homomorphic encryption. Intuitively, the blockchain infrastructure and smart contract residing on it form the privacy management layer. Meanwhile, homomorphic encryption supports a secure data transferring and computing layer among three kinds of entities: data owners, anonymisation service providers and data controllers.

In this chapter, we employ differential privacy as our primary data anonymisation technique. Other Privacy Enhancing Techniques, for example, k-anonymity, l-diversity, are not considered. The terms "anonymisation" and "sanitisation" are used interchangeably

in this chapter. We adopt UCI Adult dataset and sample different numbers of records from the dataset for designed experiments.

Chapter Organisation The rest of this chapter is organised as follows. Some preliminary knowledge and a framework overview are given in Section 5.1 and Section 5.2 respectively. In Section 5.3, the detailed protocol and implementation are discussed. Then, the evaluation of the experiments are presented in Section 5.4. Finally, we conclude our work in Section 5.6.

5.1 Preliminaries

5.1.1 Motivating Scenario

Nowadays most online service require interactions between two entities, namely service providers and client users or as refer to in the GDPR context as data controller and data subject, respectively. Data controllers are organisations or companies with more powerful computation capabilities. Data owners (DOs) are willing to share their data with these controllers hoping to get more utility by leveraging the computation advantage or delicate algorithms offered by these controllers. Specially, we consider the multiple-data-owner and multiple-data-controller scenario as a data controller usually collects data from a large group of data subjects. Meanwhile, data owner's data can be requested by multiple data controllers for different purposes. In some cases, data controllers request different parts of the data. That is why it is necessary to introduce the concept of *vertically partitioned data* to better illustrate the scenario when different parts of the dataset, or so called different attributes of the dataset are used by data controllers for various purposes. Therefore, we present the concept of vertically partitioned data to describe the data sharing scenario.

Vertically Partitioned Data. Data are said to be vertically partitioned when different attributes of information for the same set of entities are split by the data owner and shared with several data controllers (Vaidya, 2009). Thus, vertical partitioning of data can be formally defined as follows: First, consider a data owner who has a dataset D in terms of the entities from whom the data are collected and the information that is collected for each entity. Let $D = (E, I)$ be the tuple representing the dataset, where E is the entity set for whom information is collected and I is the feature set that is collected. Assume that there are k different data controllers, $C_1, ..., C_k$ requesting different parts of the dataset $D_1 = (E_1, I_1), ..., D_k = (E_k, I_k)$ respectively. Therefore, data is said to be vertically partitioned if $E = \cup_i E_i = E_1 \cup ... \cup E_k$, and $I = \cup_i I_i = I_1 \cup ... \cup I_k$ (Vaidya, 2009). In general, data can be distributed to data controllers in an arbitrary fashion. This means that different data controllers may own partial information about different sets of entities. While such arbitrary partitioning is possible, in practice,

it rarely happens. Data is said to be vertically partitioned when different controllers collect different features of data for the same set of entities. Integrating controllers' local datasets gives the global dataset. Vertically partitioned data occurs naturally in multiple-data-controller situation, and protecting the privacy of individuals in it requests more consideration.

Figure 5.1 demonstrates an example of multiple data controllers requesting vertical partitioning of data from data owners. There are two data controllers - a hypothetical hospital and an insurance company who want to request different attributes of the dataset from the data owners. The hospital collects medical records such as the type of brain tumor and diabetes. On the other hand, the insurance company collects other information such as age, sex, and occupation etc.

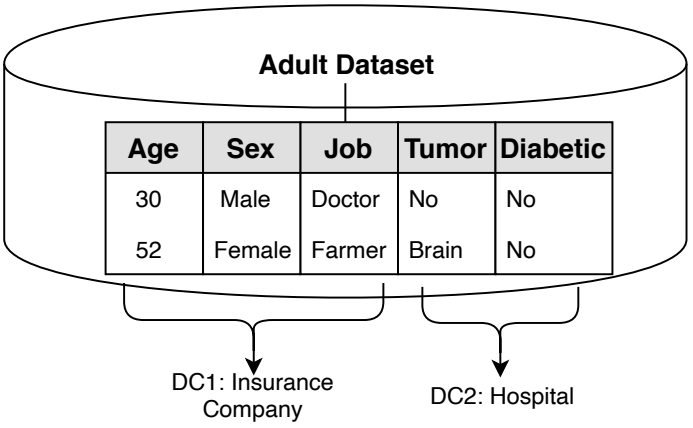


FIGURE 5.1: An example of vertically partitioned data with two data controllers

When the data owner shares the dataset with multiple data controllers, s/he has to add different types of noise to the original dataset and send those noisy data to corresponding data controllers separately. The noise addition and transferring of the noisy data causes a large amount of computational and communicational overhead for the data owner when the number of requests from data controllers is large. This problem motivates us to employ the solution of combining of homomorphic encryption and differential privacy, which avoids the transferring of whole-size noisy data to data controllers each time but only transferring a small-size of noise. This improvement helps to save a considerable amount of communicational bandwidth for the data owner.

5.1.2 Technical Approach and Objectives

We introduce a two-layer framework to separate the privacy management and secure data transferring. Similar to the framework in Chapter 4.6, the blockchain infrastructure and smart contracts are in charge of privacy management, which means that the smart contracts are responsible for verifying and recording the privacy budget of the differential privacy mechanism. Meanwhile, all the actions performed on the data are written down

as key-value transactions in the blockchain ledger, which enables data owners to be aware that their data will be shared and processed according to their own preferences. The smart contract residing on the blockchain is utilised as an interface to handle different utility requirements from multiple data controllers in a transparent manner and manage the privacy budget in a trustworthy and tamper-proof way.

Along with these features, the blockchain network is different from the one introduced in Chapter 4.6. This network contains three kinds of nodes, that is, data owners, anonymisation service providers and data controllers. The Anonymisation Service Provider (ASP) is newly introduced into the blockchain as a new kind of role for carrying out the anonymisation service.

Thanks to anonymisation service providers, data owners only need to transfer the encrypted data to the provider once and reduce further computation on noise addition. This outsourcing reduces the computation and communication overhead of the data owner. The anonymisation service provider can be seen as honest-but-curious. There are two reasons for this. Firstly, the provider has to obey the smart contract honestly otherwise s/he will not be trusted by other nodes thanks to the blockchain's consensus algorithm. The chaining technique solves the trust problem among these three kinds of nodes. Secondly, since the private data is transferred between the data owner and the ASP in the data transferring layer, an encryption scheme with additive homomorphic property is deployed to compute noise for differential privacy mechanism, which enables the ASP to perform noise addition without seeing the data in clear. Therefore, the confidentiality of the data is preserved even though the ASP is curious about the sensitive information in the data s/he received. Considering the identity of the anonymisation service providers, they can be data broker companies, for example, those who aim at providing a middleware solution for data owners and data controllers. They can also be some machines/nodes set up by the data owners on some popular cloud computing platforms, e.g. AWS, Azure, instead of data owners' local machine.

Next we present our proposed framework, including architecture design and protocol description. We also evaluate the proposed scheme in a multiple-data-controller scenario using different parts of the UCI dataset to prove the feasibility and effectiveness of our system.

The benefits of the framework introduced in this chapter are two-fold.

- Firstly, it transfers data owners' overhead of anonymisation computation. As presented in Section 5.3.3, the data owner only needs to encrypt the original data s/he wants to share with the data controller and transfer the encrypted data to the anonymisation servicer provider once. When the data owner agrees on specific preferences to share the data with the data controller, s/he uploads the privacy parameters, for example, Laplace noise in differential privacy mechanism (Dwork,

Roth, et al., 2014), to the anonymisation service provider in the encrypted form. The anonymisation service provider will then add the received privacy parameters to the original data using additive homomorphic encryption. In this way, the anonymisation service provider cannot observe either the original data or the noise but only performs the anonymisation computation, which protects the anonymisation service provider from snooping into user's privacy even when the anonymisation service providers are honest-but-curious. As such, the data owner does not need to implement the anonymisation process but focus on defining their privacy preferences s/he wants to enforce based on the data controller's request. Note that, all the processing related to the data will be recorded and monitored on the blockchain, including data controllers' request of a specific portion of the data, the privacy budget consumption and the data owners' consent to share the data. The design of the world-state of the blockchain and smart contract will be presented in Section 5.3.2.

- Secondly, the benefit of outsourcing anonymisation service is even more significant under a scenario of data sharing among multiple data controllers. When the anonymisation service are managed by data owners themselves, in order to adapt to various applications and utility preferences of multiple data controllers, data owners need to add different kinds of noise to different parts of the data, as formalised as *Vertically Partitioned Data*. This inevitably incurs enormous computation and communication burden for data owners. If data owners want to keep the sanitised data for future auditing purpose, a lot of storage space needs to be consumed as well (J. Li et al., 2018). Data owners cannot entrust data controllers for anonymising the data as there are no guarantees from technical aspect that the data controllers will perform sufficient sanitisation honestly. Therefore, it is helpful for data owners to have a reliable anonymisation service which implements the sanitisation and does not collude with the data controllers. Our framework is able to achieve this goal as both the privacy management and the sanitisation process are outsourced and decentralised to the blockchain network. Also, under the GPDR, existing techniques needs to be reconsidered for multiple data controllers scenario. Our framework is a good practice showing how a practical privacy preserving technique, i.e. differential privacy, can be well-customised for data sharing with multiple data controllers.

5.2 Framework Overview

In this section, we justify the use of encryption and a permissioned blockchain for data sanitisation outsourcing. We give a high-level overview of the framework's design and advantages (implementation details followed in Section 5.3).

5.2.1 Mitigating Privacy Management Issues in Direct-Sharing Systems

First, assume a straightforward architecture for private data sharing, in which the data owners, directly release their data to one or more data controllers. Figure 5.2 illustrates such an architecture along with the interactions between the participating entities, with solid arrows representing requests made to the data owners and dotted arrows representing the data owners' responses.

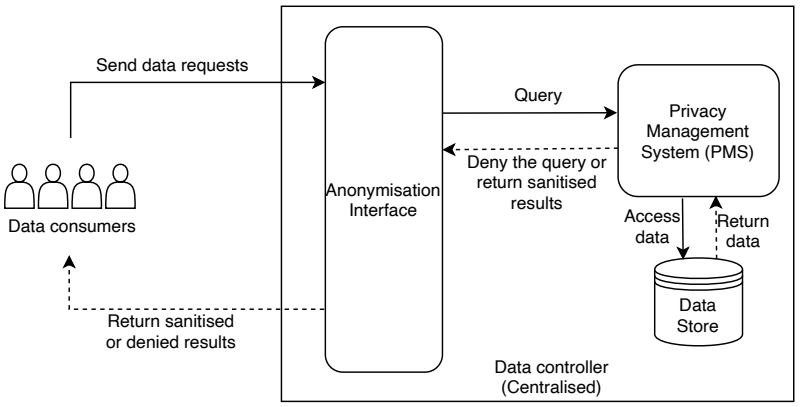


FIGURE 5.2: Centralised sanitisation model in direct-sharing system

In this design, data owners rely on themselves to manage and implement the sanitisation cautiously and correctly before sharing data with data controllers. As a result, the following shortcomings exist:

Q(1) Data owners must have their own privacy management system to keep track of their sharing history and to allow sharing only to authorised data controllers.

Q(2) There is no membership service managing the identities of data controllers.

Q(3) Data owners need to generate various versions of sanitised data in order to satisfy the requirements from different data controllers, meanwhile ensure these sanitised data will not breach the privacy guarantee when these data are released together.

Q(4) The volume of computation and the difficulty of privacy management increase dramatically when the number of data controllers increases.

Q(5) Data controllers' purpose and usage of the data cannot be monitored.

As described in Chapter 4, we observed that blockchain can mitigate the privacy management issues, i.e. Q(1) and Q(5). In order to deal with the other two shortcomings, i.e. Q(3) and Q(4), we introduce a new functionality into our framework, which is an outsourced anonymisation service that performs anonymisation tasks for data owners. In order to enable this service, a new role - anonymisation service provider (ASP) is

introduced into the blockchain. When introducing anonymisation service provider into the design, we assume the provider is curious-but-honest, which means that s/he follows the protocol honestly but is curious about data owners' private data. We explain why the engagement of the anonymisation service provider can solve those two issues by first describing the interactions between the entities.

Data owners (DOs) possess the data and want to share datasets with data controllers to get valuable knowledge from the data. Data controllers provide data analysis service to the data owners but are not trusted by the DOs. Therefore, the DOs outsource the sanitisation process to the ASP instead of directly sharing the data to data controllers. To protect the privacy of the data, the data will be encrypted before sending to the ASP. An anonymisation service provider (ASP) receives the encrypted data from DOs, performs anonymisation service on these data and then sends the noisy datasets to DCs.

The use of encryption protects the data from being observed by the ASP, however it also prevents ASP to sanitise the data. Notice that the typical differential privacy mechanism, i.e. Laplace mechanism, only requires addition operation, so we utilise the additive homomorphic encryption to perform the noise addition on the encrypted data with acceptable computation overhead (J. Li et al., 2018). Before noise generation, data controllers publish the functions they want to perform on a specific dataset. The corresponding data owner calculates the global sensitivity of those function and send the noise generation parameter to the ASP. The noise addition procedure is performed by the ASP using additive homomorphic encryption (Paillier, 1999). After that, ASP sends the noisy dataset to the DCs. Our scheme allows DCs to decrypt the encrypted noisy data, which means that data are encrypted by DC's public key. Since the only data sent to the DC is encrypted noisy data, although they can be decrypted for further analysis, the privacy of these data are still guaranteed by differential privacy. Our design uses homomorphic encryption to encrypt the transferred data meanwhile supporting computations on the ciphertext.

Similar to the design in previous chapter, we deploy a private blockchain network including three kinds of entities. These entities participate as three kinds of nodes in the blockchain, namely, the data owners nodes (DOs), the anonymisation service provider ndoes (ASP), and the data controllers nodes (DCs). Besides, we choose a permissioned blockchain in order to manage the memebership (i.e. to identify data owners, anonymisation service providers and data controllers).

At a high level, a blockchain is a distributed, immutable and tamper-proof transaction log maintained by a network of nodes. We introduce blockchain network into the design to monitor the behaviour of data controllers, at the same time providing a sharing history logging system for data owners. To do so, data owners, anonymisation service providers and data controllers all become nodes of the blockchain system to maintain a copy of the log, and a consensus protocol is used to agree on the state of the log. Introducing the

blockchain network also solves the trust issue among these three entities by integrating them into an autonomous system.

Also, the communication between the DO and the ASP will utilise Fabric's private data collection, which indicates only the hash of the encrypted data is recorded in the distributed ledger. The actual data are only visible to the ASP and the DO. Therefore, though data controllers are in the same blockchain channel as DOs and ASP, they can only see the transaction that a dataset has been transferred but not any actual data.

A critical new feature in this design is the decoupling of privacy management and data transferring, similar to decoupling the control plane from the data plane in computer network (Agarwal et al., 2019). Figure 5.3 shows the design in this decoupled environment. We define the privacy management layer, which provides the guarantee of privacy and include functionalities such as allocating privacy budget and recording data usage. In contrast, the data transferring layer includes transferring encrypted data. As shown in the figure, privacy management is now jointly managed by all parties with a blockchain back end, without having to be handled by a single data owner or to be entrusted to a third party. Data controllers can submit transactions to request access to a data resource for a specific purpose. Data owners can decide whether or not to share the data with data controllers according to their privacy preferences.

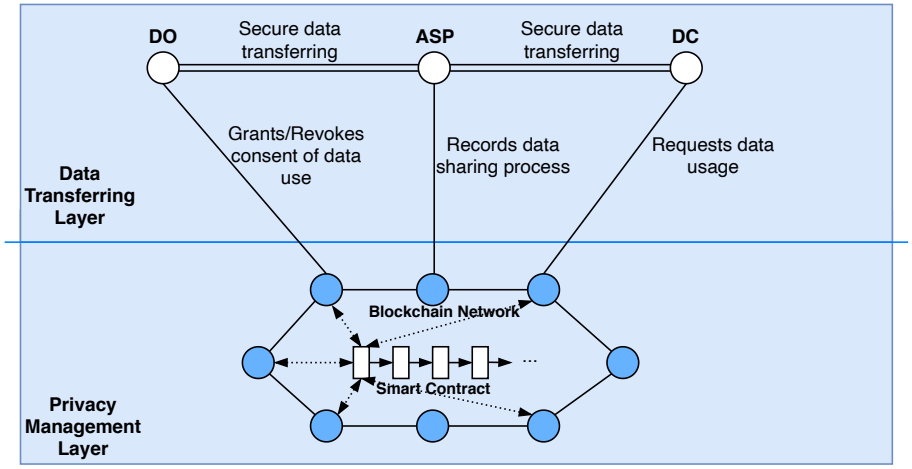


FIGURE 5.3: A decoupled framework with privacy management layer and data transferring layer

5.2.2 Managing Identities of Data Controllers using Membership Service

One main difference of using blockchain in this design compared with that in the previous chapter is the focus of Membership Service (Cachin, 2016). In order to illustrate how the Membership Service is used in a blockchain network. We first introduce the difference

between permissionless blockchain and permissioned blockchain, and why a permissioned blockchain is needed in our framework.

Permissionless blockchain and permissioned blockchain In a permissionless blockchain system such as Bitcoin, the network is public and anyone can join anonymously, making consensus difficult. Expensive methods such as *Proof of Work* are used to decide on the next block of transactions, which are then validated (e.g., to ensure there is no double-spending) by all the participating nodes. In proof of work, each node may independently validate incoming transactions and place them into an ordered block. Nodes then compete to solve a cryptographic puzzle that requires substantial computing resources. These competing nodes are referred to as miners. The first miner that solves the puzzle wins the right to append its proposed next block to the chain, as well as a reward in bitcoin. The winner sends a copy of its block to the other nodes, which again validate the transactions, and sequentially execute and commit them to update their copy of the blockchain.

On the other hand, permissioned blockchain systems are usually jointly owned by a consortium of know participants who may not necessarily trust each other. Here, nodes are not anonymous, and each node must be approved to join the network by a *membership service* run by the consortium. Thus, if any actor is found to engage in malicious activities, the membership service can take appropriate action. Instead of using proof of work, permissioned systems delegate consensus to a subset (or to all) of the participating nodes that run byzantine or crash fault tolerant consensus protocol to decide on the next block of transactions. Permissioned systems have been used in a wide range of applications requiring a tamper-proof transaction log, and are usually not backed by a native cryptocurrency.

Permissioned and permissionless systems ensure immutability and tamper resistance via full replication and hash pointers to the previous block stored in the next block (any attempted changes to the blockchain after a new block has been committed invalidate the pointers).

We utilise Hyperledger Fabric to implement the framework. Fabric introduces a modular and extensible architecture to provide a resilient, flexible, scalable and confidential permissioned blockchain for general purpose. The implementation of Fabric deploys an execute-order-validate model to execute distributed smart contracts in a trust-less environment. The model contains modular components to proceed with the transaction in three stages. Firstly, the smart contract execution component manages the execution of the smart contract within the isolated container, then checks the correctness of the generated transaction and thereby endorse it; Secondly, orderers broadcast the state updates to all the peers and provide an ordering service to establish an agreement on the order of transactions among peers via a consensus protocol; Thirdly, transactions are validated

against the endorsement policy specified by the application before they are committed to the ledger. Besides the execute-order-validate model, Fabric also implements a scalable dissemination component to disseminate the output blocks of the ordering service to all peers using a gossip protocol. Each peer in Fabric locally maintains a ledger in the form of the append-only chaining blocks and stores the state updates in a key-value database.

Membership Service As mentioned in Section 5.2.1, one shortcoming of a direct-sharing system is the lack of membership service managing the identities of data controllers. One advantage of permissioned blockchain is that it inherently has a membership service, which manages identities of the participants in the network. In our framework, the blockchain network consists of three kinds of entities: data owner, anonymisation service provider and data controller. The main benefit of introducing the membership service is to manage the identity of data controllers. The data owner can know who requests their data and makes the decision whether or not to share certain data according to the data controller's identity and credibility.

Since Hyperledger Fabric is used in our framework, we herewith illustrate how the Membership Service in Fabric is deployed to achieve our goal. First of all, for an identity to be verifiable, it must come from a trusted authority. This authority is called membership service provider (MSP) in Fabric. An MSP provides all nodes in the blockchain network with identities as well as credentials for the purposes of authentication and authorisation. We first configure Fabric to use the default MSP implementation and adopt a standard Public Key Infrastructure (PKI) hierarchical model (Perlman, 1999). Then we set up the built-in certification authoritie (CA), called Fabric-CA, to generate and distribute X.509 certificates (2016) to nodes for authentication purpose. Note that, this process is configured off-line before the blockchain network runs.

5.2.3 Private Data Collection Mechanism for Data Transferring

Another important feature deployed in our framework is the privacy data collection mechanism in permissioned blockchain (2016). In Fabric's design, a *channel* is a private subnet for communication between two or more organisation members. After each peer joins a channel, it will have its own identity, which is provided by the membership services provider (MSP) and used to authenticate itself to other peers and services in the channel. Each transaction is executed on a channel, and each party must be authenticated and authorised to execute transactions on the channel. Although a peer can join multiple channels and maintain multiple ledgers, ledger data cannot be transferred from one channel to another. This separation between ledgers implemented through channels is achieved through chaincode configuration, the membership service and the gossip data dissemination protocol (2016). This separation of peers and ledger data through

channels allows private and confidential transactions between members. If a group of organisations on a channel needs to keep data private from other organisations on the channel, using the channel mechanism, they can choose to create a new channel that contains only the organisations that need to access the data. However, each time this situation is encountered, creating a separate channel adds additional management overhead (maintaining chaincode versions, policies, MSPs, etc.). This method can not support all channel participants to see all transaction records while ensuring that part of the data is private (Hyperledger Fabric, 2016).

This is why we use the private data collection mechanism provided by Fabric, which enables the defined subset of organisations on the channel to endorse, commit, or query private data without creating a separate channel (2016). In our scenario, this means that the data owner can send encrypted data to anonymisation service provider under the same channel without letting the data controller observe the transmitted data, while recording the transaction about the fact of data transmission in the channel where all three are present.

A private data collection mainly contains two elements (Hyperledger Fabric, 2016):

- **The actual private data**, sent peer-to-peer via gossip protocol (Barger et al., 2017) to only the organisations(s) authorised to see it. This data is stored in a private state database on the peers of authorised organisations (sometimes called a "side" database, or "SideDB"), which can be accessed from chaincode on these authorised peers.
- **A hash of that data**, which is endorsed, ordered, and written to the ledgers of every peer on the channel. The hash serves as evidence of the transaction and is used for state validation and can be used for audit purposes.

While unauthorised peers will not have the private database synced and will only be able to see the hash on the ledger. Since hashes are irreversible, these peers will not be able to see the actual data. Using the private data collection of Hyperledger Fabric, the encrypted data transferred between data owner and anonymisation service provider are not visible to the data controller, therefore maintain the confidentiality of the original data.

In general, blockchain provides a decentralised platform to integrate three kinds of parties together and mitigates the trust issues among them. Data owner nodes collaborate together to provide an anonymisation service for themselves without relying on any single party. As such, privacy budget consumption are recorded and verified by data owners together, which enhance the security of data sharing with multiple data controllers by eliminating single-point-of-failure. Data controllers publish the functions they want to perform on the data in the blockchain ledger as well. Data owners can observe what

kinds of operation will be implemented on their data and decide the privacy budget for each function with data controllers. This enables data owners have more control of the usage of their data. Last but not least, since every data use will be recorded as a transaction in the blockchain, the shared ledger will provide an evidence for data controller to show to regulatory organisations that the use of the data are GDPR-compliant.

5.2.4 Differential Privacy with Homomorphic Encryption

In this section, we are not going to repeat the definition of differential privacy, but focus on how differential privacy can be combined with homomorphic encryption and is customised for outsourcing sanitisation process between entities. The combination of homomorphic encryption and differential privacy supports a secure outsourcing of the anonymisation process to the ASP. More importantly, the solution saves data owners' computation and communication overhead of anonymising a dataset multiple times when the dataset are shared with various data controllers with different data utility requirements. Traditional differential privacy solutions for data sharing with multiple data controllers require the data owner to add different types of noise to the original dataset and send those noisy data to those data controllers separately. The transfer causes a large amount of communicational overhead for the data owner when the number of data controllers is large. With the combination of homomorphic encryption, the data owner now can send the encrypted original data to the ASP only once and then send different types of encrypted noise according to data controllers' request. This combination saves communicational bandwidth for the data owner as the original dataset do not need to be transferred multiple times, and only the small-size noise are transferred multiple time.

Additive Homomorphic Encryption An encryption scheme is said to be additive homomorphic if it conforms to the following properties (J. Li et al., 2018):

Definition 5.1. (Additive homomorphic encryption)

Let m_1 and m_2 be two plaintexts, let $\mathcal{A}$ be an encryption algorithm that outputs the corresponding ciphertexts $\|m_1\|$ and $\|m_2\|$, and let $\mathcal{B}$ be an operation performed on the two ciphertexts. For any two ciphertexts, additive homomorphic encryption has the following property:

$$\mathcal{B}(\mathcal{A}(m_1), \mathcal{A}(m_2)) = \mathcal{B}(\|m_1\|, \|m_2\|) = \|m_1 + m_2\|$$

On the one hand, the homomorphic encryption properties allow us to run computations on encrypted data. Specifically, additive homomorphic encryption enables us to compute an encrypted sum of a set of encrypted values without decrypting them first, and hence nothing about the individual values is disclosed. On the other hand, differential privacy

has steadily become the de-facto standard for achieving strong privacy guarantees in data analysis. Laplace mechanism is one of the most-widely used differential privacy mechanism. The primary step of the Laplace mechanism is the addition of Laplacian noise to original data. We therefore apply additive homomorphic encryption to the Laplace mechanism. After the combination, the addition can be performed on the encrypted noise and original data. Therefore, the process can be outsourced to the ASP without disclosing the original data nor the added noise. We deploy Pallier cryptosystem Paillier, 1999 in our implementation as Pallier scheme is a public-key encryption scheme with additive homomorphic properties. We denote it as $(Setup, KeyGen, Enc, Dec, Add)$, which consists of the following steps (J. Li et al., 2018).

- ***Setup***(1^l): A membership service provider (MSP) uses a security parameter l to generate the public parameters (pp) and the main secret key (msk).
- ***KeyGen***(msk, pp, uid): The MSP uses a user's identity uid as input to generate a pair of keys (pk, sk) for that user.
- ***Enc***(pk, m): The user uses his public key pk to encrypt a plaintext record m, generating the ciphertext $\|m\|$ as output.
- ***Dec***$(sk, \|m\|)$: The user uses his secret key sk to decrypt a ciphertext record $\|m\|$ into the corresponding plaintext m.
- ***Add***$(\|m_1\|, \|m_2\|)$: Two ciphertext $\|m_1\|$ and $\|m_2\|$ are inputs, and the result $\|m_1 + m_2\|$ is output.

Recall that, to share data that satisfy ϵ-DP when a query function f is applied, the principal approach is to perturb the data by adding random noise based on Δf and the privacy budget ϵ. For example, for the Laplace mechanism, let $Lap(\lambda)$ denote the Laplace probability distribution with mean zero and scale λ. The Laplace mechanism achieves DP by adding Laplace noise to an original dataset M. In concrete, we describe the *Global Sensitivity* calculated by data owners and Encrypted Laplace Mechanism used by anonymisation service provider as follows.

Definition 5.2. (Global sensitivity)

Let f be a function that maps a database to a database to a fixed-size vector of real numbers. For all neighboring databases D_1 and D_2, the global sensitivity of f is defined as

$$\Delta(f) = max_{D_1,D_2}||f(D_1) - f(D_2)||$$

where $||\cdot||$ denotes the L_1 norm.

As data owners have the original dataset and the function from data controllers since data controllers make the request and have it recorded on the blockchain, data owners will calculate the global sensitivity, encrypted it and sent it the anonymisation service provider in the following steps. Thereafter, the ASP uses the encrypted Laplace mechanism to perform the anonymisation.

Definition 5.3. (Encrypted Laplace mechanism)

Let m be a record in database M ($m \in M$), let η be a random variable such that $\eta \sim Lap(\Delta f/\epsilon)$, and let $\mathcal{A}$ be an encryption algorithm with additive homomorphic property. The encrypted Laplace mechanism is defined as follows:

$$\mathcal{A}(m') = \mathcal{A}(m) + \mathcal{A}(\eta)$$

Note that, the data record m and Laplace noise η are encrypted with the encryption algorithm $\mathcal{A}$ by the data owner. Then the encrypted data are sent to the ASP for further processing, i.e. addition. The main rationale behind the outsourcing is to preserve more utility of the data. If the data are anonymised by the data owner before sending them to the data controller, data utility is limited and cannot be customised by the data controller for multiple purposes. An advantage of uploading the encrypted data and noise to the ASP separately is that different kinds of noise can be added to the original data for various utility. First, the data controller can publish the set of functions $\mathbf{F} = \{f_1, f_2, ..., f_n\}$ s/he wants to compute on the data on the blockchain. If the data owner is interested in the data analysis and willing to share the data (i.e. functions in $\mathbf{F}$ the data controller proposed), s/he can decide the privacy budget $\epsilon = \{\epsilon_1, \epsilon_2, ..., \epsilon_n\}$ for each function in $\mathbf{F}$ and calculate the corresponding differential privacy noise $\frac{\Delta f_i}{\epsilon_i} (i = 1, 2, ..., n)$. Then the data owner will encrypt the original data and the noise, then upload them to the ASP. The ASP will use the additive homomorphic encryption to calculate the sum of the noise and the original data. Finally, the ASP will send the noisy data to the DC.

5.3 Protocol and Implementation

In this section, we discuss the implementation of our framework using the Hyperledger Fabric and homomorphic encryption. We first analyse the threat model and security requirement of the system. Then we illustrate the transaction and world states design in the blockchain system. Since the framework is split into two layers, we give a detailed illustration on the interaction and protocols between the two layers.

5.3.1 Trust Relation and Threat Model

We first analyse the trust relation among these three kinds of entities. Firstly, the anonymisation service provider (ASP) is honest-but-curious, which means the provider will follow the protocol honestly but is curious about the sensitive information in the data. Therefore, the data owner will use encryption to encrypt the data before sending them to the ASP. The ASP is not able to see the clear data. Secondly, the ASP do not collude with the data controller. Since the encrypted data is encrypted using the data controller's public key, the ASP cannot send the original encrypted data to the data controller, but only send the noisy encrypted data to the data controller. Also, data controllers will not give his secret key to ASP, otherwise ASP can decrypt the encrypted dataset to get the original data. This requirement is satisfiable in reality. For example, the ASP can be Google and the data controller can be Facebook. They have a competitive business relationship so will not collude with each other (J. Li et al., 2018). Last but not least, the data owner does not trust the data controller. The data owner hopes to use data controller's service, mainly the data analytics algorithms and computation resource. Rather than sending clear data to data controllers, the data owner would only like to delegates the anonymisation service to ASP and only allow ASP to send noisy data.

5.3.2 World States Design

There are two important components in a blockchain system: the world state and the smart contract. The former is a key-value store to maintain some application-defined state information. In our system, the privacy management deals with data owners (identified by $data\_owner\_ID$) and their data resources (identified by $data\_set\_ID$ and $data\_attribute\_ID$), data controllers (identified by $data\_controller\_ID$) and their request (identified by $request\_function$ and $budget\_request$). To allow vertically partitioned data sharing, data sets are divided based on the attributes set identified by $data\_attribute\_id$.

The key-value world state is a critical data structure maintained by Fabric to process transactions. In simple financial applications, the world state is straightforward: the key is an account ID and the value is the current balance in that account. Smart contracts that move funds from one account to another must verify that there is enough money in the sender's account, subtract funds from the sender's account, and add funds to the recipient's account (Agarwal et al., 2019). This can be done by reading from and writing to the world state, followed by appending the corresponding transaction to the blockchain; it is not necessary to scan the blockchain (which may be very long) in order to commit a transaction.

In the design of our framework, the privacy management challenge of data anonymisaiton outsourcing is to transform the requests from data controllers and the responses from data owners into a key-value world state. We want to ensure high transaction throughput to scale to very large deployments: many data owners, many datasets vertically partitioned into many attributes sets, many data controllers and various data requests, etc.

To explore the space of world state designs, we observe that the three main entities in anonymisation outsourcing management are data owners, data, data controllers. This suggests three designs, explained below and illustrated in key-value format in Listing 16.

- **Data-Controller-Oriented world state** (DC-WS) groups data controllers that request the same data resource together. A key is a concatenation of data owner ID, data set ID, data attribute ID, the request function and privacy budget, and a value is a list of data controllers that were given access to the data specified in the key.
- **Data-Owner-Oriented world state** (DO-WS) A key is a concatenation of data controller ID, data set ID, data attribute ID, the request function and privacy budget, and a value is the corresponding data owner ID.
- **Data-Resource-Oriented world state** (DR-WS) A key is a concatenation of data owner ID, data controller ID, the request function and privacy budget, and a value is a concatenation of data set ID and data attribute ID.

```
1   DC-WS
2   {do_id | dset_id | dattr_id | req_func | bud_req :
3           [dc_id_1, dc_id_2, ..., dc_id_n]
4   }
5
6   DO-WS
7   {dc_id | dset_id | dattr_id | req_func | bud_req :
8           [do_id]
9   }
10
11  DR-WS
12  {dset_id | dattr_id | do_id | dc_id:
13          [ req_func | bud_req]
14
15  }
```

Listing 5.1: Three kinds of world state designs in key:value format

5.3.3 Smart Contracts and Workflow

As discussed in the previous section, there are three kinds of world state that are used by three different smart contracts. In the pseudocode, we use the following functions to interact with the key-value world state.

GET(k): returns the value corresponding to key k
PUT(k,v): write value v to key k and increase the version number (or creates a new key with version number 1 if the key does not exist)

Data Controller Request Data Usage: Algorithm 2 shows the pseudocode for the smart contract invoked by data controllers to request data usage on a specific data resource. Remember that the Data-Resource-Oriented-World State is used in this smart contract. It first assembles keys by concatenating the data owner ID, data controller ID, data set ID and data attribute ID (Line 2). It then fetches a list of request history for the key (Line 3). If the requested privacy budget is smaller than the remaining budget and request history list is empty, it creates an entry for this key and store the requested functions and corresponding budgets (Line 4-6). If an entry already exists and also the requested budget does not exceed the remaining budget, it updates the request history list by concatenating the new functions and budgets (Line 7-9). Finally, the world state is updated with the new value (Line 10). The smart contract always checks if the remaining budget has been used up before updating the request list, in order to make sure the guarantee of differential privacy has not been violated.

Algorithm 2 The Data Controller Requests Data Usage

Input: do_id, dc_id, req_func, bud_req, dset_id, dattr_id, bud_remaining
1: **procedure** DATA CONTROLLER REQUEST DATA USAGE
2: *key* ← do_id | dc_id | dset_id | dattr_id
3: *req_list* ← GET(*key*)
4: **if** bud_req <= bud_remaining **then**
5: **if** *req_list* == ∅ **then**
6: *req_list* ← [req_func | bud_req]
7: **else**
8: *req_list* ← *req_list* ∪ [req_func | bud_req]
9: **end if**
10: PUT(*key*, *req_list*)
11: **end if**
12: **end procedure**

Data Owner Grants/Revokes Consent of Data Use: Algorithm 3 shows the pseudocode for the smart contract which allows consent modification for a data resource assigned to data controllers. Remember that the Data-Controller-Oriented-World-State is used in this smart contract. It first assembles keys by concatenating the data owner ID, data set ID, data attribute ID, and the parameters for differential privacy mechanism, i.e. requested functions and corresponding privacy budgets (Line 2). It then fetches a list of consenting data controllers for the key (Line 3). If the list is empty and the action is Grant, it creates an entry for this key and store the data controller ID (Line 4-6). If an entry already exists and if the action is Revoke, it updates the value by deleting the data controller id from the list (Line 7-10). If the action is Grant and the data controller

id does not exist in the value, the id is added (Line 11-13). Finally, the world state is updated with the new value (Line 15).

Algorithm 3 The Data Owner Grants/Revokes Consent of Data Use

```
Input: do_id, dset_id, dattr_id, req_func, bud_req, dc_id, action
 1: procedure DATA OWNER GRANTS/REVOKES CONSENT OF DATA USE
 2:     key ← do_id | dset_id | dattr_id | req_func | bud_req
 3:     dc_id_list ← GET(key)
 4:     if dc_id_list == ∅ and action == 'Grant' then
 5:         dc_id_list ← [dc_id]
 6:     end if
 7:     if dc_id_list ≠ ∅ then
 8:         if dc_id ∈ dc_id_list and action == 'Revoke' then
 9:             dc_id_list ← dc_id_list \ dc_id
10:         end if
11:         if dc_id ∉ dc_id_list and action == 'Grant' then
12:             dc_id_list ← dc_id_list ∪ dc_id
13:         end if
14:     end if
15:     PUT(key, dc_id_list)
16: end procedure
```

Anonymisation Service Provider Records Data Sharing Process: Algorithm 4 shows the pseudocode for the smart contract invoked by the ASP when the data sharing process has been completed. Remember that the Data-Owner-Oriented-World State is used in this smart contract as this helps data owners to audit the integrity of the data sharing process. The contract performs the actions if and only if the data sharing is successful. It assembles keys by concatenating the data controller ID, data set ID, data attribute ID, the requested functions from data controllers and privacy budgets used (Line 3). The value for the key is the data owner ID of the data resource (Line 3). If the data sharing process succeeds, this key-value pair will be written to the world state (Line 5), otherwise the smart contract will not do anything.

Algorithm 4 Anonymisation Service Provider Records Data Sharing Process

```
Input: dc_id, dset_id, dattr_id, req_func, bud_req, do_id, action
1: procedure ANONYMISATION SERVICE PROVIDER RECORDS DATA SHARING
2:     if action == 'Success' then
3:         key ← dc_id | dset_id | dattr_id | req_func | bud_req
4:         value ← do_id
5:         PUT(key, value)
6:     end if
7: end procedure
```

Based on these smart contracts, we are now able to describe the overall process of outsourced differential private data sharing as follows:

Step 1 - Blockchain Network Setup. In the blockchain network, three parties will participate in the same Hyperledger Fabric network within one channel. Also, the data owner and the anonymisation service provider set up a private data collection communication for encrypted data transfer. The membership service provider manages the membership for each party and generates public-private key pair for data controllers. For example, let (pk, sk) be a key pair for the data controller, where pk is the public key and sk is the secret key.

Step 2 - Data Controller Requests Data Usage. In this step, the data controller invokes the smart contract 2 to request a data usage. The request will be published on the blockchain so that the data owner knows how their data will be used. In the request, the parameter req_func represents the possible set of functions, denoted as $F = (f_1, f_2, \cdots, f_m)$, that will be used in the data analysis tasks by the data controller. For example, these functions can be $max, min, sum, average$ for a statistical analysis task. The parameter bud_req denotes the corresponding privacy budget for each function in the Laplace mechanism which determines the amount of noise to be added to the original data. As shown in the smart contract, this set of functions and budgets is invoked as a transaction and written to the world state on the blockchain so that the other two entities - the data owner and the anonymisation service provider knows how strong the protection has been implemented on the data.

Step 3 - Data uploading. If the data owner is willing to share the data with the data controller according to his data usage request, firstly the data owner will invoke the smart contract 3 to give his/her consent. Then, the data owner fetches the data controller's public key pk from the blockchain, encrypt his dataset $M = (m_1, m_2, \cdots, m_n)$ using the $Enc(pk, M)$ algorithm, and uploads the resulting ciphertexts
$C = (||m_1||, ||m_2||, \cdots, ||m_n||)$ to the anonymisation service provider through the private data collection. Besides encrypted data uploading, the data owner needs to upload the corresponding noise parameter as well. As the data controller has already published the functions $f_i (i = 1, 2, \cdots, m)$ and privacy budget $\epsilon_i (i = 1, 2, \cdots, m)$ in Step 2, firstly the data owner is able to calculate the function sensitivities $\Delta F = (\Delta f_1, \Delta f_2, \cdots, \Delta f_m)$, and furthermore the parameter vector $\mathbf{b}$ for noise generation (e.g. for the Laplace mechanism), which depends on $\mathbf{b} = (b_1, b_2, \cdots, b_m) = \frac{\Delta F}{\epsilon} = (\frac{\Delta f_1}{\epsilon_1}, \frac{\Delta f_2}{\epsilon_2}, \cdots, \frac{\Delta f_m}{\epsilon_m}))$. Then the data owner encrypts the parameter vector $\mathbf{b}$ with the same encryption scheme $Enc(pk, M)$ and transferred the encrypted vector to the ASP via the private channel. Note that, there is a maximum value limitation of total privacy budget $\epsilon = \sum_{i=1}^{m} \epsilon_i$ to protect the privacy, for example, $\epsilon = 20$ is often used in practice, which is defined by the data owner relative to his/her privacy preferences.

Step 4 - Noise addition. After receiving the parameter vector $\mathbf{b}$ from the data owner, the anonymisation server provider generates the actual noise η in the Laplace mechanism where $b_i = \frac{\Delta f_i}{\epsilon_i}$ is used as the parameter to define the Laplace distributions from which to randomly draw noise, and encrypts the noise using $Enc(pk, \eta) = ||\eta||$. Then, the

anonymisation service provider uses the homomorphic addition $Add(||M||, ||\eta||)$ to add the generated noise to the data owner's encrypted data and only sends the resulting noisy data to the data controller. The success of performing these steps by the anonymisation service provider means that s/he has the consent from the data owner and sanitises the data according to the data owner's privacy requirement. The ASP then invoke the smart contract 4 to record the evidence.

Step 5 - Data analysis. The data controller first decrypts the received ciphertexts using $Dec(sk, ||M + \eta||)$ to obtain all of the noisy data. Based on these data, the data controller can perform other algorithm in order to derive more information from the data. The details of the sharing process are shown in Algorithm 5

Algorithm 5 Outsourced Differential Private Data Sharing Using Blockchain and Encryption

DC: invokes smart contract 2 to request data usage
if data owner grants consent of data use **then**
    DO: invokes smart contract 3 with $action = Grant$
    DO: calculates the function sensitivities $\Delta F = (\Delta f_1, \Delta f_2, \cdots, \Delta f_m)$
    DO: fetches privacy budget ϵ from the blockchain
    DO: calculates the parameter vector $\mathbf{b} = \frac{\Delta F}{\epsilon}$
    DO: $||M||_{pk} = Enc(M, pk)$, $||\mathbf{b}||_{pk} = Enc(\mathbf{b}, pk)$, encrypts original data and the parameter vector
    DO: transfer $\{||M||, ||\mathbf{b}||\}$ to the anonymisation service provider
    ASP: generates encrypted noise $||\eta||$ obeying Laplace distribution $||\eta|| \sim Lap(||\mathbf{b}||)$
    ASP: $||M + \eta||_{pk} = Add(||M||, ||\eta||)$, calculate the noisy dataset
    ASP: sends $||M + \eta||$ to DC;
    DC: $M + \eta = Dec(||M + \eta||, sk)$, decrypts the ciphertexts
end if

5.4 Evaluation

In this part, we evaluate the performance of our solution with regards to encryption overhead, transmission overhead, and blockchain efficiency, focusing on testing whether our solution has good scalability in the context of multiple data owners and multiple data controllers. The experiments are carried out on a computer running MacOS with a 2.3 GHz 8-core Intel-i9 CPU and 32GB of memory. We use Golang as the development language for the chaincode in Hyperledger Fabric, and choose the Laplace mechanism as the main mechanism to provide differential privacy protection.

5.4.1 Encryption Overhead

One of the key goals in the experiment is to explore the impact of the introduction of encryption algorithms on the efficiency of data sharing, that is, to analyse the computational overhead of encryption algorithms during data sharing. Note that in the data sharing scheme, there are mainly three types of encryption computation, namely encryption on the data owner side, decryption on the data controller side, and homomorphic addition on the anonymisation service provider side. We explored the time spent in each of these three computations in the experiment and analyzed the relationship between them.

We use the UCI Adult data (*UCI Adult Data* 1996) as the experimental data set, and the Paillier scheme (Paillier, 1999) which is a commonly used homomorphic encryption library in our implementation. The UCI Adult data set contains 45222 records and 6 numeric attributes, which are used as data to be shared. We divide the data into vertically partitioned data sets by attributes. These attributes are *age, education-num, capital-gain, capital-loss, hours-per-week, final-weight*. The data controller can then request different partitioned data sets based on their interest in these attributes. Without loss of generality, in our experiments, these attributes are randomly selected by the data controller.

In order to simulate the data sharing scenario with multiple data controllers, we set the number of data controllers to 10. Besides, we let the number of requests from each data controller vary by 20, 60, 180, 540, 1620 and 4860, so that the total number of requests, in other words, the number of data sharing, vary by 100, 300, 900, 2700, 8100, and 24300. This range is rational to explore the impact of the increase in the number of data requests on sharing efficiency, which is the main focus of the experiment. We assume that the function requested by the data controller is *max* in the differential privacy mechanism. Here we do not consider other functions, because different functions will only lead to differences in shared value, and will not affect the encryption process. On the other hand, we set the number of data owners to 5 to simulate multiple data owners. Since the focus is to test the scalability of data sharing with multiple data controller, there is no need to vary the number of the data owner. We set the number of data records owned by each data owner to 100 as we suppose the data records held by the data owner are not changing dynamically. With these parameters, we perform homomorphic encryption and calculations on the data shared each time. Then we compare the influence of encryption, decryption and homomorphic addition on the efficiency of data sharing.

As shown in Figure 5.4, the overall computing cost increases almost linearly with the increase in the number of data sharing in logarithmic scale. Among them, the computation time of homomorphic encryption and decryption is in the same order of magnitude, but the time needed for homomorphic addition is three orders of magnitude lower than the former two. For example, for 8100 times of data sharing, the encryption operation

requires 645.21 seconds, the decryption operation requires 183.01 seconds, and the homomorphic addition operation requires only 0.37 seconds. This means that the data owners need to spend more time on encryption than data controllers and anonymisation service providers. However, in differentially private data sharing, the data owner only needs to encrypt and upload the original data once. In subsequent sharing to different data controllers, if the noise parameters are the same, the data owner does not need to regenerate the noise and re-encrypted them, which can save a lot of computation time. In this experiment, we assume that the data owner generates a new noise every time, so the encryption time required will be longer than in the actual situation. Besides, since the original data does not change frequently, data owners can use their spare time to encrypt the data in advance and upload it to anonymisation service provider to achieve better utilisation of their resources.

It is worth noting that, since the time required for homomorphic addition is short, it is helpful for anonymisation service provider to use the Laplace mechanism to quickly generate anonymised data, which means that homomorphic addition has little effect on data sharing efficiency. This advantage is more obvious when there are many data controllers or many requests. The data controller usually has more computing resources than the data owner, so the decryption operation does not cost much to it. Moreover, after the ciphertext is transmitted from the anonymisation service provider to the data controller, it will not affect the subsequent data sharing process, so the data controller can also use the idle time for decryption. In summary, the computational overhead introduced by homomorphic encryption has little effect on the entire data sharing process, and it introduces more protection without significantly reducing the sharing efficiency.

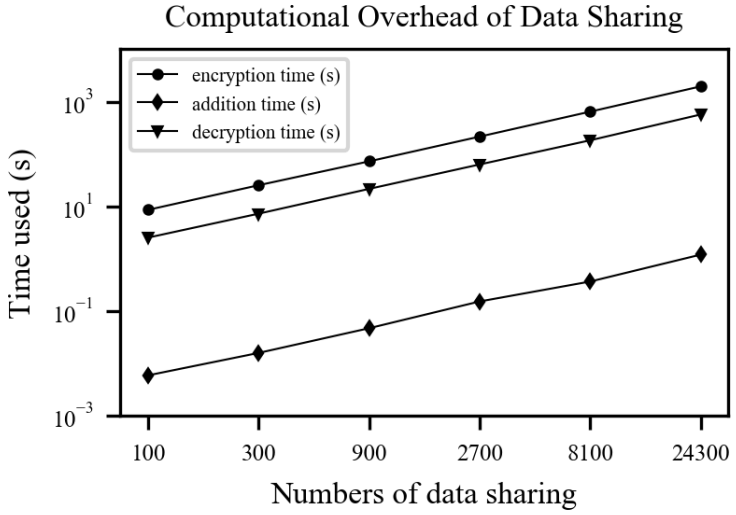


FIGURE 5.4: Computational overhead of outsourcing differential privacy sanitization

5.4.2 Communicational Overhead

After the introduction of ASP, the straightforward two-party data transfer between the data owner and the data controller changes to the data transfer among the three parties. Among them, there are two main phases that will incur communication costs, including data uploading from DO to ASP and data downloading from ASP to DC.

In order to theoretically analyse the differences between the data transmission in the new scheme and the straightforward direct transmission between the data owner and the data controller, we first introduce the symbols in Table 5.1 to represent the various variables in the transmission process. In the data uploading phase, the size of the message sent in our scheme is $ndc + k'x$ bytes, where ndc represents the size of the encrypted dataset that needs to be transmitted only once, and $k'x$ represents the size of the noise transmitted x times. In contrast, the message size in the direct transmission is $nd(p + k)x$ bytes. In this scheme, not only the noise but also the data need to be transmitted each time the sharing happens. As shown in the formula, with the increase in the number of data sharing, in the direct transmission scheme, the communication cost increases linearly. Meanwhile, in the new scheme, when sampling 45 records, the original data size is 1445 bytes, while the encrypted data size increases to 2881 bytes, indicating that encryption results in a nearly double increase of the size. As shown in Figure 5.5, once the number of data sharing is more than twice, our scheme will ensure lower communication costs. In the data downloading phase, compared with the $ndpx$ in the direct scheme, the total size of the message sent by the ASP to the DC is $ndc'x$, where c' is the size of the encrypted noisy data. In general, the size of the data sent from the ASP to the DC in the downloading phase in our scheme is about twice the size of the data in the direct sharing scheme. However, when the amount of data sharing becomes larger, the communication cost saved in the data uploading phase of our scheme is more significant than the additional cost in the data downloading phase.

TABLE 5.1: Symbols used in data transferring phase

Symbols	**Meanings**
n	the size of the dataset
d	the number of attributes
c	the size of each ciphertext record
p	the size of each plaintext record
k'	the size of each encrypted noise
k	the size of each plaintext noise
x	the number of data sharing

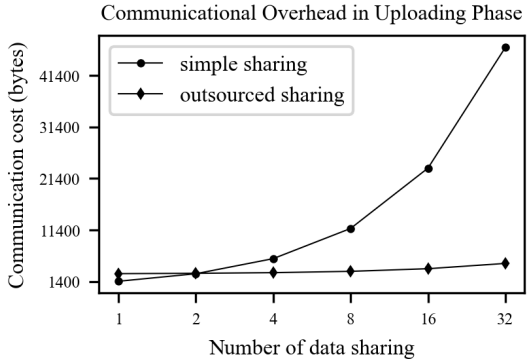


FIGURE 5.5: Communicational overhead in data uploading phase

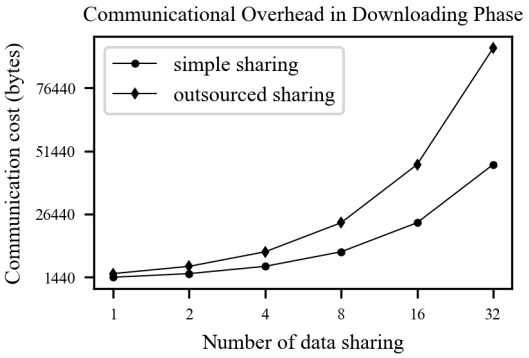


FIGURE 5.6: Communicational overhead in data downloading phase

5.4.3 Blockchain Practicality

Blockchain Storage Capacity The data sharing history stored on each block is only the metadata about the released results, and it is the latest one, not the complete list of released results. Meanwhile, neither the original dataset nor encrypted dataset will be stored on the blockchain. The world state stored on the blockchain is three kinds of key-value pairs introduced in Sec. 5.3.2, designed for three different smart contract functionalities. The design of these history tuples is lightweight and suitable for blockchain storage and query.

Blockchain Performance - Benchmarking of World State The key to test the performance of using Blockchain platform in our framework is to measure the throughput when reading and writing to the world state. Since the world state is in the form of *key-value* pairs, we explore the effect of both the key and the value on throughput by increasing the size of them respectively. The other purpose of this experiment is because the design of the three kinds of world states in the framework cause different changes in the key and value size. First, when a data resource is requested by multiple data controllers, the *dc_id* field of the value in the *data-controller oriented world state (DC-WS)* will continue to increase. Second, when the resource is requested by many data controllers or when the requested functions and budgets vary, the *data-owner-oriented world state (DO-WS)* design will produce a event where many different keys have the same value *do_id*. Third, when a data resource is requested multiple times and each request consumes a different privacy budget, the value corresponding to the key in the *data-resource oriented world state (DR-WS)* design will be frequently added with new requests. In response to the these three different situations induced by three kinds of world state designs, we designed three different sets of experiments.

First, we gradually increase the number of data controllers involved in the value of the *DC-WS* design from 20k to 1000k, which means that the size of the value corresponding to the key will also increase. At the same time keep the size of each key unchanged. In

this experiment, we only perform world state reading (i.e. GET requests). Figure 5.7 shows the transaction throughput in this case.

Next, we consider the *DO-WS* design. In the experiment corresponding to Figure 5.8, we fixed the value space as one data owner, and then increased the key space, that is, the number of data controllers corresponding to that data owner and the different number of data requests, from 1 to 10k. Similarly, we only perform GET requests.

Finally, Figure 5.9 shows, in the *DR-WS* design, the impact of different number of requests in the value on the write throughput of the world state (i.e., PUT requests). We keep the size of the key unchanged and change the number of $[req\_func \mid bud\_req]$ item in the value corresponding to the key.

As shown from Figures 5.7 to 5.9, increasing the number of keys or increasing the number of items contained in the value field does not reduce the throughput of blockchain read and write. Instead the throughput only fluctuates up and down. We deduce that this is the normal blockchain read and write throughput fluctuation. Therefore, we conclude that the design of the three kinds of world states is effective for sharing scenarios with multiple data owners and multiple data controllers, and the read and write throughput of the blockchain are neither affected by the number of keys nor the number of items in each value.

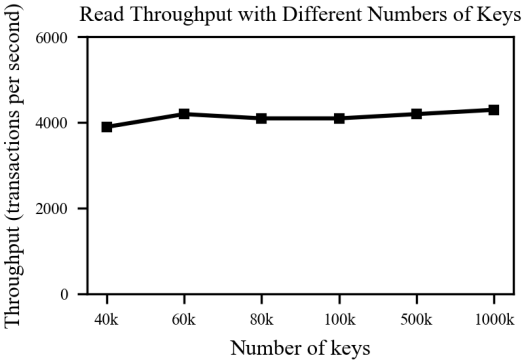


FIGURE 5.7: Read throughput with different numbers of keys

5.5 Related Work

Data anonymisation model Traditional data anonymisation models, for example, k-anonymity, l-diversity, and DP, contribute a lot to data privacy community. However, these techniques are originally proposed for single dataset anonymisation. They can be improved or combined with other techniques, for instance, homomorphic encryption, to entail better privacy guarantees or to be suitable for data sharing with multiple parties. This is increasingly important as there is a trend that personal data are shared with more and more data controllers under the complicated big data era.

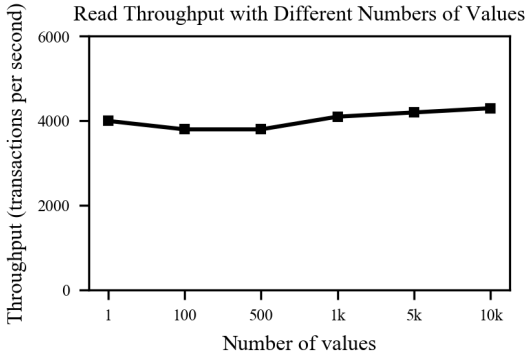


FIGURE 5.8: Read throughput with different numbers of values

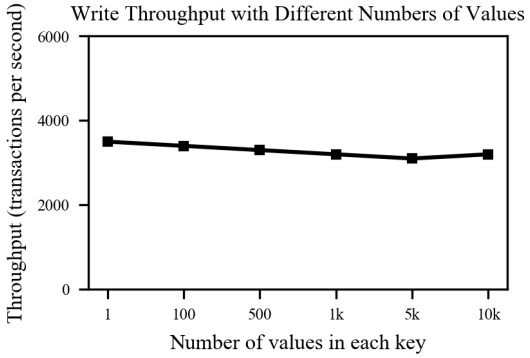


FIGURE 5.9: Write throughput with different numbers of values

Differential Privacy with Blockchain MU YANG et al. (2018) proposed a blockchain-based technique that has proved to be feasible and efficient in providing a secure layer of privacy budget management for differential privacy thanks to the main blockchain's features, namely decentralisation, transparency and tamper-proof. Yang's work focuses on answering interactive queries with differential privacy in a cloud federation environment, where a centralised anonymisation service is still needed besides the blockchain. In our scheme, the idea of using blockchain for privacy management is inspired by Yang's work. However, our scheme deals with a different data sharing scenario, i.e. non-interactive data publication with multiple data controllers. Also, the anonymisation service provider in our scheme is part of the blockchain network, which means that anonymisation procedures are under the audit of data owners and data controllers as well.

Encryption with Blockchain Our work is also inspired by Li's work (J. LI et al., 2018). Li's proposal outsource the differential privacy procedure to a cloud service provider (CSP) and also use additive homomorphic encryption for noise addition. However, Li's work greatly relies on the CSP's honesty of implement the sanitisation service. If the CSP is compromised or collude with the data receiver, the privacy of the whole scheme is broken. Our work use the blockchain to provide a decentralised anonymisation

and privacy budget management service, which is more reliable, trustworthy and not vulnerable to single-point-of-failure. The combination of blockchain with encryption has also proved to provide privacy guarantees in decentralised systems according to previous work (L. Chen et al., 2019; Manzoor et al., 2019; Rahulamathavan et al., 2017; Zyskind et al., 2015). Zyskind et al. (2015) design a decentralised personal data management system to allow user possessing and controlling their data. The method utilises a symmetric encryption scheme to protect the confidentiality of the data transferred on the network. Manzoor et al. (2019) propose a secure IoT data sharing system, which combines a proxy re-encryption scheme and smart contracts residing on the blockchain to allow that the data sharing is only visible between the owner and the intended person. Rahulamathavan et al. (2017) instead combine attribute-based encryption with blockchain to provide a decentralised data sharing system with both confidentiality and access control. L. Chen et al. (2019) develop a blockchain-based sharing system for electronic health record with searchable encryption to support complex query on the data. In our work, homomorphic encryption is integrated with blockchain to provide data confidentiality, meanwhile supporting the computation on encrypted data.

5.6 Conclusion

In this chapter, we proposed a novel privacy-preserving data sharing scheme with differential privacy, blockchain and encryption. Our contributions can be summarised as follows.

- A blockchain-based data publication approach to manage privacy budget consumption as well as data controllers' analytic functions through autonomous smart contracts according to data owner privacy and data-utility requirements.
- A decentralised anonymisation service provider performing anonymisation service for data owners with additive homomorphic encryption to guarantee the confidentiality of data.
- Implementation and evaluation by means of the Hyperledger Fabric blockchain, as well as discussion on data privacy and utility trade-off.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we proposed a roadmap for data controllers to process data in accordance with the new data protection regulation GDPR. We first presented a risk-based approach to investigate personal data types in a data anonymisation scenario from a combination of both legal and technical perspectives. Subsequently, a privacy-risk mining framework based on machine learning is further proposed to identify the potential risk of linkage attack among heterogenous sanitised datasets. Following the risk analysis, we introduced a key mechanism of differential private data publishing with blockchain technology, i.e. using blockchain for tamper-proof privacy budget allocation. Then we presented another solution that combines blockchain and homomorphic encryption to outsource the differential private sanitisation process from centralised data controller to a decentralised network organised by data owners, which enable data owners to have full control of the sanitisation process.

We built a common terminology to describe three types of data states for anonymisation under the GDPR. More specifically, we investigate the key meanings of three types of data in the GDPR, i.e. pseudonymised data, anonymised data and Art.11 Data. Then we proposed a risk-based approach to translate the legal language into technical analysis. The proposed approach relies upon a granular analysis of three common re-identification risks, i.e. singling out, linkability and inference and further distinguishes between local, domain and global linkability to capture the key concepts of additional information and pseudonymisation introduced in the GDPR. Consequently, the study examined the robustness of practical data anonymisation techniques against these types of re-identification risk, and then classified the output of different techniques into the three types of data. To effectively balance data utility and data privacy requirements, we combined the most reliable data anonymisation technique, i.e. differential privacy with transparent privacy budget allocation scheme and a monitoring system.

We further refined our risk-based approach by exploiting the power of machine learning, proposing a two-stage clustering framework for mining privacy risk under dynamic data publishing context. More specifically, the first stage is to identify the global linkability among datasets through their common attributes; the second is to capture the local linkability among anonymised records via overlapped attribute values. The two types of linkability are used to estimate the privacy risk through a privacy risk tree. This privacy risk mining framework is useful for data controllers to uncover the risk of publishing new sanitised datasets. As such, data controllers are capable of taking the appropriate measures toward data privacy in accordance with the GDPR.

In addition, we also considered a data sharing scenario when a data owner outsource their data to data controller and could potentially lose control over the shared data. We introduced a key mechanism that integrates differential privacy and blockchain to improve the utility of data sharing for benefiting the controllers, meanwhile supporting transparency-by-design, privacy-budget-evident tracking and monitoring of the sharing procedure. Our approach relies on blockchain to validate the usage of privacy budget and adaptively change its allocation via smart contracts, depending on the privacy preferences provided by data owners. The transparency provided by the blockchain enables the data owners to control the anonymisation process and hence enhance the overall security of the system.

In order to resolve the trust issue between the data controller and the data owner, we proposed an anonymisation outsourcing framework to empower data owners with full privacy control of the anonymisation process. This framework offers a solution to the data controller to outsource the differential privacy anonymisation process to a decentralised blockchain network. In particular, this framework combines blockchain and homomorphic encryption technologies to enable new privacy protection capabilities, i.e decentralised anonymisation service and secure anonymisation on encrypted data. This work also provide a solution for secure data-sharing among multiple data controllers, where data owners customise the sanitisation of the data amounts to the utility requirements from multiple parties. To validate our proposed approach, we implemented a prototype using the permissioned blockchain Hyperledger Fabric. This prototype was further evaluated with respect to computational and communicational overheads.

In summary, the work in this thesis involved two main parts in a progressive way. We first supported data controllers understand the essence and privacy risks of the data they are holding. Then, we provided one solution for the controllers to better utilise the data they collected, and to process data in a transparent and tamper-proof way. We proposed another solution for the data controllers to outsource the anonymisation service to a blockchain network in order to further decentralise the anonymisation process and enable data owners to have full privacy control of their data.

6.2 Future work

In this thesis, we focus on the different definitions of personal data and the anonymisation terminologies in the GDPR. We interpret three types of data newly emerged in the GDPR with regard to existing anonymisation techniques. The GDPR also highlights many other rights of data subject, for example, the right to consent, right to forgotten, right to rectification, and right to restriction of processing etc. A very important consideration is the ability to guarantee the rights of data subject from a technical perspective rather than only rely on the legal requirements. Therefore, it is an important avenue for future research and development to interpret these rights using technical languages and build solutions based on the combination of legal and technical understanding.

Privacy risk assessment is important for data controllers to understand the potential risk of privacy leakage in the data they hold and the loss caused to corresponding individuals because of the leakage. This work focuses on the risk of personal data leakage when publishing heterogeneous datasets. The risk of leakage occurs when there are connections between multiple datasets that are released dynamically. Although each dataset has been anonymised separately, when these datasets are analysed together, it is still possible to locate overlapping individuals from them. Therefore, we try to find out possible connections between anonymised datasets through machine learning. In addition of focusing on the possibility of leakage amongst published datasets, a future research direction may focus on the loss caused by linkage attacks. Because each linkage may cause different degrees of damage to user's privacy. It is essential to establish a reasonable model to explain the different levels of loss to both the data controller and data subject when the linkage occurs.

In order to equip data controllers with technical solutions to request and exploit the data from data owners, we proposed two solutions. The first solution utilises the blockchain technology to build a decentralised system that manages the privacy budget allocation of differential privacy mechanism. Then the second solution is proposed to outsource the differential privacy sanitisation to a decentralised network by adopting blockchain and homomorphic encryption. An important direction we like to further investigate is to further develop sophisticated protocols that can support more complex data analysis tasks based on these two frameworks.

Bibliography

Abril, Daniel, Guillermo Navarro-Arribas, and Vicenc Torra (2012). "Improving record linkage with supervised learning for disclosure risk assessment". *Information Fusion* 13.4, pp. 274–284. ISSN: 1566-2535.

Agarwal, Rishav Raj, Dhruv Kumar, Lukasz Golab, and Srinivasan Keshav (2019). "Consentio: Managing consent to data access using permissioned blockchains". *arXiv preprint arXiv:1910.07110*.

Art. 11, General Data Protection Regulation (2018). *Processing which does not require identification*. https://gdpr-info.eu/art-11-gdpr/. Accessed: 2020-06-01.

Art. 2, General Data Protection Regulation (2018). *Material scope*. https://gdpr-info.eu/art-2-gdpr/. Accessed: 2020-06-01.

Art. 28, General Data Protection Regulation (2018). *Processor*. https://gdpr-info.eu/art-28-gdpr/. Accessed: 2018-12-18.

Art. 4, General Data Protection Regulation (2018). *Definitions*. https://gdpr-info.eu/art-4-gdpr/. Accessed: 2020-06-01.

Article 29 Data Protection Working Party (2007). "Opinion 04/2007 on the concept of personal data".

– (2014). "Opinion 05/2014 on Anonymisation Techniques".

Bainbridge, David and Graham Pearce (1998). "Data controllers and the new data protection law". *Computer Law Security Review* 14.4, pp. 259–264. ISSN: 0267-3649.

Barger, Artem, Yacov Manevich, Benjamin Mandler, Vita Bortnikov, Gennady Laventman, and Gregory Chockler (2017). "Scalable communication middleware for permissioned distributed ledgers". In: *Proceedings of the 10th ACM International Systems and Storage Conference*, pp. 1–1.

Byun, Ji-Won, Ashish Kamra, Elisa Bertino, and Ninghui Li (2007). "Efficient k-anonymization using clustering techniques". In: *International Conference on Database Systems for Advanced Applications*. Springer, pp. 188–200.

Byun, Ji-Won, Tiancheng Li, Elisa Bertino, Ninghui Li, and Yonglak Sohn (2009). "Privacy-preserving incremental data dissemination". *Journal of Computer Security* 17.1, pp. 43–68.

Cachin, Christian (2016). "Architecture of the hyperledger blockchain fabric". In: *Workshop on distributed cryptocurrencies and consensus ledgers*. Vol. 310. 4.

Caliński, Tadeusz and Jerzy Harabasz (1974). "A dendrite method for cluster analysis". *Communications in Statistics-theory and Methods* 3.1, pp. 1–27.

Chapter 3, General Data Protection Regulation (2018). *Rights of the data subject*. https://gdpr-info.eu/chapter-3/. Accessed: 2020-06-01.

Chen, Deyan and Hong Zhao (2012). "Data Security and Privacy Protection Issues in Cloud Computing". In: *Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering - Volume 01*, pp. 647–651.

Chen, Lanxiang, Wai-Kong Lee, Chin-Chen Chang, Kim-Kwang Raymond Choo, and Nan Zhang (2019). "Blockchain based searchable encryption for electronic health record sharing". *Future Generation Computer Systems* 95, pp. 420–429.

Chen, Xiuguo, Wensheng Yin, Pinghui Tu, and Hengxi Zhang (2009). "Weighted k-means algorithm based text clustering". In: *2009 International Symposium on Information Engineering and Electronic Commerce*. IEEE, pp. 51–55.

Christen, Peter (2012). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media.

Dalenius, Tore (1986). "Finding a needle in a haystack or identifying anonymous census records". *Journal of official statistics* 2.3, p. 329.

De, Sourya Joyee and Daniel Le Métayer (2016). "Privacy harm analysis: a case study on smart grids". In: *Security and Privacy Workshops (SPW), 2016 IEEE*, pp. 58–65.

Dheeru, Dua and Efi Karra Taniskidou (2017). *UCI Machine Learning Repository*.

Directive, EU (1995). "95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data". *Official Journal of the EC* 23.6.

Dwork, Cynthia (2006). "Differential Privacy". In: *Proceedings of the 33rd International Conference on Automata, Languages and Programming*, pp. 1–12.

– (2008). "Differential privacy: A survey of results". In: *International conference on theory and applications of models of computation*. Springer, pp. 1–19.

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith (2006). "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 265–284. ISBN: 978-3-540-32732-5.

Dwork, Cynthia, Aaron Roth, et al. (2014). "The algorithmic foundations of differential privacy." *Foundations and Trends in Theoretical Computer Science* 9.3-4, pp. 211–407.

Ekblaw, Ariel, Asaph Azaria, John D Halamka, and Andrew Lippman (2016). "A Case Study for Blockchain in Healthcare: "MedRec" prototype for electronic health records and medical research data". In: *Proceedings of IEEE Open & Big Data Conference*. Vol. 13, p. 13.

El Emam, Khaled, Eloise Gratton, Jules Polonetsky, and Luk Arbuckle (2013). "The Seven States of Data: When is Pseudonymous Data Not Personal Information?" *Journal of Science & Technology* 24.1.

Esposito, Christian, Aniello Castiglione, and Kim-Kwang Raymond Choo (2016). "Encryption-based solution for data sovereignty in federated clouds". *IEEE Cloud Computing* 3.1, pp. 12–17.

EU:C:2016:779 (2016). *CJEU, C-582/14, Patrick Breyer v Bundesrepublik Deutschland*. http://curia.europa.eu/juris/liste.jsf?num=C-582/14. Accessed: 2020-06-01.

Commission, European (2016). "General Data Protection Regulation". *Official Journal of the European Union* L119, pp. 1–88.

Ferdous, Md Sadek, Andrea Margheri, Federica Paci, Mu Yang, and Vladimiro Sassone (2017). "Decentralised runtime monitoring for access control systems in cloud federations". In: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, pp. 2632–2633.

Fung, Benjamin CM, Ke Wang, Rui Chen, and Philip S Yu (2010). "Privacy-preserving data publishing: A survey of recent developments". *ACM Computing Surveys (Csur)* 42.4, pp. 1–53.

Ganta, Srivatsava Ranjit, Shiva Prasad Kasiviswanathan, and Adam Smith (2008). "Composition attacks and auxiliary information in data privacy". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 265–273.

GDPR Basics: What is the difference between a data controller and a data processor? (2018). https://www.dporganizer.com/gdpr-data-controller-vs-processor/. Accessed: 2018-12-18.

General Data Protection Regulation (2018). https://gdpr-info.eu. Accessed: 2020-06-01.

Graham, Christopher (2012). "Anonymisation: managing data protection risk code of practice". *Information Commissioner's Office*.

Hintze, Mike and Gary LaFever (2017). "Meeting Upcoming GDPR Requirements While Maximizing the Full Value of Data Analytics". *Available at SSRN 2927540*.

Hyperledger Fabric (2016). *Private Data*. https://hyperledger-fabric.readthedocs.io/en/release-2.0/private-data/private-data.html. Accessed: 2020-07-01.

ICO (2018). *Data sharing code of practice*. https://ico.org.uk/media/for-organisations/documents/1068/data_sharing_code_of_practice.pdf. Accessed: 2020-06-01.

International Organization for Standardization (2008). *ISO/TS 25237:2008 Health Informatics – Pseudonymization*. https://www.iso.org/standard/42807.html. Accessed: 2020-06-01.

Kellaris, Georgios, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias (2014). "Differentially Private Event Sequences over Infinite Streams". *Proceedings of the VLDB Endowment* 7.12.

Leibniz Institute for Educational Trajectories (LIfBi) (2009). *Star Ng Cohort 6: Adults (SC6) SUF Version 7.0.0 Anonymiza On Procedures Tobias Koberg*. https://www.

nepsdata.de/Portals/0/NEPS/Datenzentrum/Forschungsdaten/SC6/7-0-0/SC6_7-0-0_Anonymization.pdf. Accessed: 2020-06-01.

Li, Jin, Heng Ye, Wei Wang, Wenjing Lou, Y Thomas Hou, Jiqiang Liu, and Rongxing Lu (2018). "Efficient and secure outsourcing of differentially private data publication". In: *European Symposium on Research in Computer Security*. Springer, pp. 187–206.

Li, Ninghui, Tiancheng Li, and Suresh Venkatasubramanian (2007). "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity". In: *2007 IEEE 23rd International Conference on Data Engineering*, pp. 106–115.

Machanavajjhala, Ashwin, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam (2007). "l-diversity: Privacy beyond k-anonymity". *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1, 3–es.

Manzoor, Ahsan, Madhsanka Liyanage, An Braeke, Salil S Kanhere, and Mika Ylianttila (2019). "Blockchain based proxy re-encryption scheme for secure IoT data sharing". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, pp. 99–103.

McSherry, Frank (2009). "Privacy Integrated Queries: An Extensible Platform for Privacy-preserving Data Analysis". In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. SIGMOD '09. Providence, Rhode Island, USA: ACM, pp. 19–30. isbn: 978-1-60558-551-2.

Nakamoto, Satoshi (2008). *Bitcoin: A peer-to-peer electronic cash system*.

Paillier, Pascal (1999). "Public-key cryptosystems based on composite degree residuosity classes". In: *International conference on the theory and applications of cryptographic techniques*. Springer, pp. 223–238.

Pantlin, Nick, Claire Wiseman, and Miriam Everett (2018). "Supply chain arrangements: The ABC to GDPR compliance—A spotlight on emerging market practice in supplier contracts in light of the GDPR". *Computer Law Security Review* 34.4, pp. 881–885.

Perlman, Radia (1999). "An overview of PKI trust models". *IEEE network* 13.6, pp. 38–43.

Politou, Eugenia, Alexandra Michota, Efthimios Alepis, Matthias Pocs, and Constantinos Patsakis (2018). "Backups and the right to be forgotten in the GDPR: An uneasy relationship". *Computer Law Security Review* 34.6, pp. 1247–1257. issn: 0267-3649.

Rahulamathavan, Yogachandran, Raphael C-W Phan, Muttukrishnan Rajarajan, Sudip Misra, and Ahmet Kondoz (2017). "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption". In: *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, pp. 1–6.

Recital 29, General Data Protection Regulation (2018). *Pseudonymisation at the Same Controller*. https://gdpr-info.eu/recitals/no-29/. Accessed: 2020-06-01.

Rousseeuw, Peter (1987). "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis". *J. Comput. Appl. Math.* 20.1, pp. 53–65. ISSN: 0377-0427.

Runshan, Hu, Sophie Stalla-Bourdillon, Mu Yang, Valeria Schiavo, and Vladimiro Sassone (2017). *Bridging Policy, Regulation, and Practice? A Techno-Legal Analysis of Three Types of Data in the GDPR*. Hart Publishing.

Samarati, Pierangela (2001). "Protecting respondents identities in microdata release". *IEEE transactions on Knowledge and Data Engineering* 13.6, pp. 1010–1027.

Stalla-Bourdillon, Sophie and Alison Knight (2016). "Anonymous Data v. Personal Data-False Debate: An EU Perspective on Anonymization, Pseudonymization and Personal Data". *Wisconsin International Law Journal* 34, p. 284.

Sweeney, Latanya (2002). "k-anonymity: A model for protecting privacy". *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05, pp. 557–570.

Theoharidou, Marianthi, Spyros Kokolakis, Maria Karyda, and Evangelos Kiountouzis (2005). "The insider threat to information systems and the effectiveness of ISO17799". *Computers & Security* 24.6, pp. 472–484.

UCI Adult Data (1996). https://archive.ics.uci.edu/ml/datasets/adult. Accessed: 2020-05-01.

Vaidya, Jaideep (2009). "Vertically Partitioned Data". In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, pp. 3263–3265. ISBN: 978-0-387-39940-9.

Wang, Cong, Qian Wang, Kui Ren, and Wenjing Lou (2010). "Privacy-preserving public auditing for data storage security in cloud computing". In: *Infocom, 2010 proceedings ieee*, pp. 1–9.

X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks (2016). https://www.itu.int/rec/T-REC-X.509. Accessed: 2020-07-01.

Yang, M., V. Sassone, and K. O'Hara (2012). "Anonymisation: managing data protection risk code of practice". *UK Information Commissioner's Office*.

Yang, Mu, Andrea Margheri, Runshan Hu, and Vladimiro Sassone (2018). "Differentially private data sharing in a cloud federation with blockchain". *IEEE Cloud Computing* 5.6, pp. 69–79.

Zhang, Ning, Jin Li, Wenjing Lou, and Y Thomas Hou (2018). "PrivacyGuard: Enforcing private data usage with blockchain and attested execution". In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, pp. 345–353.

Zyskind, Guy et al. (2015). "Decentralizing Privacy: Using Blockchain to Protect Personal Data". In: *Proceedings of the 2015 IEEE Security and Privacy Workshops*, pp. 180–184.