Doctoral Dissertation

A study on Utility-aware Privacy-preserving Techniques

（有用性を考慮したプライバシ保護技術に関する研究）

January 2022

Graduate School of Engineering,
Osaka University

Tomoaki Mimoto

Supervisor: Professor Atsuko Miyaji

Abstract

Privacy is a fundamental human right and the constitutions of many countries mention the right to privacy. However, the development of computer science is threatening that right. Security and privacy measures are essential for the safe and secure utilization of data. The main difference between privacy and security is who is the attacker. As for security, the data needs to be protected from outsiders, and the correct data needs to be delivered to authorized recipients. Privacy, on the other hand, needs to take into account that even authorized recipients are attackers because data subjects have rights associated with their data. Therefore, it is important to strike a balance between privacy and utility, and technologies are needed to achieve both requirements.

Various privacy protection techniques have been proposed so far, and we focus on de-identification techniques. There are various directions of de-identification methods. In this dissertation, we broadly divide them into de-identification techniques for quasi-identifiers, de-identification techniques for sensitive attributes, and de-identification techniques based on information theory. k-anonymization is a typical de-identification method for quasi-identifiers, and has been adopted in various fields because it preserves overall trends in the data. k-anonymization can be achieved by various techniques such as generalization by hierarchical trees, swapping, clustering, and top/bottom coding, and a great number of k-anonymization algorithms have been proposed. However, this type of methods may be vulnerable to sensitive attributes. Therefore, de-identification methods for sensitive attributes such as l-diversification has been proposed. This type of methods are difficult to achieve for real data, and could significantly reduce its usefulness. Furthermore, when de-identified data is published, various attackers can be assumed, and de-identification methods and privacy metrics according to their attacker models are in disarray. The impact of combining de-identification techniques has not been well investigated, although it is believed that combining de-identification techniques would provide a better balance between privacy and utility. In fact, many publicly available data incorporate multiple de-identification techniques. In this context, k-anonymity is still widely accepted metric because its concepts and metrics are simple and straightforward, and privacy of the de-identified data will be protected by legal deterrents and systems that take privacy-by-design into account. On the other hand, researchers have proposed de-identification techniques based on information theory to achieve perfect privacy protection. Differential privacy is a prime example, and various mechanisms have been proposed to date. Differential privacy adds probabilistic noise to the output result, or all data, which may compromise the properties of the original data except

for necessary information, e.g., the results of a particular analysis. Therefore, it is necessary to design ad-hoc methods for different use cases. Privacy and utility are in the relationship of trade-off, and it is difficult to resolve them all completely.

In this study, we will examine privacy-preserving techniques that focus mainly on the utility of data and the understandability of privacy strength. Therefore, we consider the privacy strength as the probability that an individual will be re-identified. This is a generalization of the most widely known idea of k-anonymity. However, it is difficult to come up with a privacy metric that is consistent across formats and types of data. Privacy data can be divided into structured data and unstructured data. Furthermore, there are static and dynamic data for each type of data. Unstructured data such as documents are difficult to handle and there are few existing studies. Moreover, since dynamic data has more dimensions than static data, the curse of dimensionality problem is serious. Therefore, de-identification techniques also need to address the issues of privacy and usefulness in terms of data format.

Privacy-preserving techniques include not only de-identification techniques but also secure computation. Secure computation allows multiple data holders to compute a specific function while keeping each other's input secret. This allows for the integration and analysis of data across multiple institutions. It also allows an institution with data to outsource its analysis to another institution, which leads to secure data utilization. The main issue with secure computation is the amount of computation and communication. Furthermore, since the secure computation provides correct output, there is a possibility that privacy information can be leaked from the output value. Therefore, in secure computation, the privacy of the output must be protected, or the output must be available only to the data holder.

This dissertation presents *a study on utility-aware privacy preserving techniques*, and focuses on de-identification techniques. In particular, we aim for easy-to-understand metrics of the strength of privacy and usefulness in order to seek understanding from data subjects for data provision. First, we start with the basic idea of privacy in Chapter 1 and mention the need for privacy-preserving techniques. Then, in Chapter 2, we introduce the basic privacy-preserving techniques and related techniques used in our proposal. In Chapter 3, we describe several related studies on privacy-preserving techniques and discuss the remaining issues.

We propose de-identification methods for structured static, structured dynamic, and unstructured static data in terms of data format. We consistently consider the probability that an individual is re-identified as the strength of privacy in this paper, but give different privacy metrics depending on the format of the data.

In Chapter 4, we focus on structured static data, which is the most common type of data. We focus on three types of de-identification methods: generalization, noise addition, and sampling, and experimentally evaluate de-identified data combined with the various methods. It is easy to combine de-identification methods and is often done to generate flexible de-identified data. However, there is no metric to evaluate the privacy strength of such de-identified data. Since it is difficult to simply evaluate the privacy of de-identified data using a combination of methods, we conducted an evaluation using attack simulations. This evaluation is effective when the system is designed with privacy in mind and the data is properly managed. Since the proposed privacy metric is equivalent to k-anonymity, it is expected to facilitate data subjects' understanding of data use. In addition, we proposed a utility metric for de-identified data, assuming that the data will be used as training data for machine learning. This idea of a simulation-based privacy and utility metrics can be applied to any type of data. The results of our experiments show that data de-identified by the combined method retains utility compared to data de-identified by a single method when they have the same privacy.

In Chapter 5, we investigate de-identification methods for structured dynamic data. Dynamic data is need to consider time factor and balancing privacy and utility is an especially difficult problem. One of the directions of de-identification of dynamic data is the modification of pseudonymized attributes. This eliminates the association between data from different times of the same person. However, because individuals tend to behave similarly, it can be difficult to completely break the connection between data even if different pseudonyms are used. Therefore, we assume a re-identification attack and a linkability attack on dynamic data. We treat dynamic data as a matrix and propose a de-identification method using matrix manipulation in this chapter. As in the case of static data, we used a privacy metric equivalent to k-anonymity and a utility metric that takes into account actual use. In our experiment, we conducted an evaluation using actual web access log data and showed that the proposed method can maintain its utility compared to conventional de-identification methods. Furthermore, we showed that even with different pseudonymization processes at different times, i.e., treating the same person at different times as a different person, it is still possible to identify individuals with high probability. We also confirmed that the privacy risk against linkability attack can be significantly reduced with a very small amount of data processing.

In Chapter 6, we investigate de-identification techniques for unstructured data, mostly document. Since it is difficult to directly assess the privacy and utility of unstructured data, we start with the structuring of data through morphological analysis. We need to consider the possibility that anyone can be an attacker because documents can be publicly available, such as court documents or reports of accidents that occur in schools. Disturbing the data like differential privacy may not be appropriate since the documents need to be readable and understandable by people. Therefore, we proposed an attacker model and an attack algorithm that assumes a very powerful attacker with access to external databases. Documents are generally de-identified manually, and many existing studies have used how close to manually de-identified documents they can be as a privacy metric. On the other hand, we considered that privacy risks still lurk in manually de-identified documents. In our experiments, we applied the proposed attack algorithm to real manually de-identified documents, and confirmed that we could actually re-identify individuals from them. We found the proposed attack algorithm to be effective, and proposed an algorithm to process the risky words used in the attack that could lead to the identification of individuals. Our algorithm is able to prevent strong attacks using web search with minimal data processing, and thus maintain its utility. As mentioned above, we have shown that our simulation-based attack and countermeasure can be applied to various data formats with k-anonymization in mind, which is easy for data subjects to understand. All of our proposed de-identification methods maintain a high level of utility compared to simple de-identification methods.

In the following chapters, we investigate other privacy-preserving techniques toward further research.

In Chapter 7, we will focus on differential privacy mechanism. Differential privacy requires the calculation of sensitivity depending on the query. We defined a differential privacy model with dummy data in order to deal with the case where it is difficult for us to calculate sensitivity. The only difference from the original definition is that we do not consider arbitrary data, and various existing differential privacy mechanisms satisfy the proposed definition. Furthermore, we designed a differential privacy mechanism that uses the calculation of t-values in a t-test as a query as a concrete example. The impact of the proposed mechanism on the p-value is large, and the privacy parameter needs to be large to avoid the type I error. Alternatively, we need to replace the problem to reduce the sensitivity and get closer to the correct result instead of performing the t-test directly and this discussion is ongoing.

In Chapter 8, we focus on basic secure comparison protocols, which is well known as the millionaire's problem. Comparison protocol is a basic two-party protocol and is used to compare two values in secret, but it is very important because it is widely used in data mining and machine learning. Even in the comparison of two values, there are numerous classifications depending on who has the data, in what state (e.g., encrypted, shared, etc.), and who gets the output. We systematized this classification. We then describe conversions among these types of protocols. Many conversions have been dealt with in existing studies, so the types and transformations are dealt with in the chapters 2 and 3. These conversions allows us to construct a comparison protocol for one setting by converting an existing protocol from a different setting. Moreover, we proposed a base protocol; using this in combination with the above mentioned conversions allows to obtain an efficient comparison protocol for any of the configurations captured in our taxonomy. Finally, we implemented the proposed protocol and compared it with previous works. The results show that the proposed protocol outperforms the most efficient existing protocol in terms of computation time by taking advantage of its parallel processing capability.

Finally, we conclude by summarizing our results and future works, and provide a direction to preserve privacy.

Acknowledgments

I would like to thank everyone who has supported me to pursue my PhD.

First of all, I would like to express my gratitude to my supervisor, Professor Atsuko Miyaji.

She kindly accepted me as a working adult student, and gave me the opportunity to study on *utility-aware privacy-preserving techniques* as my research topic. I appreciate her invaluable support and guidance through the hard moments of graduate school.

I also would like to express my gratitude to the members of my examination committee, Professor Tetsuya Takine, Professor Noboru Babaguchi, Professor Kyo Inoue, Professor Seiichi Sampei, Professor Takashi Washio, Professor Akihiro Maruta, Professor Kazunori Komatani, for reviewing my dissertation and giving me valuable feedbacks. A very special thanks goes to Professor Jun Sakuma. He also took time to review my dissertation from his busy schedule and give me valuable feedbacks.

I would like to thank my superiors, Shinsaku Kiyomoto from KDDI Research, Inc. If he had not strongly recommended me to pursue my PhD, this dissertation might not have existed.

I am thankful to my superiors, colleagues, and subordinates in my workplace at KDDI Research, Inc. and Advanced Telecommunications Research Institute International. Thanks to their support and encouragement, I was able to pursue my PhD as a working adult student.

Finally, I am thankful to my family for their longtime support through my life. Especially, I deeply appreciate my wife Saori for understanding my dream and encouraging me every day.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Big data has become a hot topic in several fields, such as academia, IT industry, finance, and business [KYH$^+$14]. The amount of data created in digital world has increased excessively [Tan12], and there are many investments conducted in several fields. Particularly, personal data have great potential for building an efficient and sustainable society. Many companies like Amazon, Google, and Facebook provide value-added services through the use of personal information, such as location, purchase history, and health information. It is obvious that information systems and personal information are already inseparably. On the other hand, with interest in privacy problems being heightened, the laws concerning privacy are reviewed taking into account changes in society. The General Data Protection Regulation (GDPR), a regulation in EU law on data protection and privacy in the European Union (EU), came into force in May 2018. This regulation affected many national laws outside the EU, such as Japan.

Privacy is private life and affairs that we do not want others to see. For this reason, the subject of privacy varies from person to person. Researchers have been pursuing privacy metrics that are quantitative and acceptable to everyone, and privacy-preserving technologies that balance privacy and utility.

The difference between privacy and security is who can be the attacker. In security, we are primarily concerned with how to deliver data safely and accurately to authorized receivers. Privacy, on the other hand, needs to take into account that even authorized receivers are attackers because they may learn things about the individuals contained in the information they receive. Therefore, in order to protect the privacy of individuals, we must ensure that receivers do not get any information about individuals, while retaining the information needed for certain analyses. An overemphasis on privacy protection will reduce utility, as even simple analysis will not be possible. On the other hand, an overemphasis on utility increases the likelihood that personal privacy will be violated. Therefore, it is important to find a balance between privacy and utility.

Some definitions of privacy-related terms are given in ISO 25237: 2017 [ISO].

Anonymization is a *"process by which personal data is irreversibly altered in such a*

> *way that a data subject can no longer be identified directly or indirectly, either by the data controller alone or in collaboration with any other party."*

The word "irreversibly" is powerful and absolute, and in practice, it may be difficult to obtain data that is both anonymized and remains utility. For example, when a personal data is encrypted, the encryption key should be discarded to be recognized as an anonymized data. De-identification is also defined in [ISO] as follows:

> De-identification is a *"general term for any process of reducing the association between a set of identifying data and the data subject."*

Therefore, de-identification is needed as for balancing the privacy and utility of personal data. Furthermore, in Japan, personal data can be utilized without obtaining consent of the person if the data is de-identified enough, and even if the data is medical data, each medical institution can utilize the data by obtaining consents by opt-out. Researchers have been investigating ways to maintain the utility of datasets while protecting information about individuals. A number of privacy and utility metrics have been proposed for various situations, as well as de-identification techniques that satisfy these metrics. The scope of research is diverse, as different forms of data require different models of attackers and different methods of de-identification. On the other hand, some studies report de-identified personal data may be re-identified [NS06] and the problems of privacy have becomes tangible. For example, some research results [GCF10, Kor10] have suggested that Internet advertisements using private data involve the risk of leaking users' private information. Attack methods are becoming as diverse as de-identification methods, and research is continuing toward an ideal de-identification technology that can both protect privacy and maintain usefulness.

Privacy by design is a well known concept for balancing data privacy and utility, and it is required that privacy of data should be considered at the time of design. In order to make practical use of personal data while protecting privacy, it is necessary not only to improve privacy protection technology, but also to consider various aspects such as legal regulations and system design to isolate personal data from external databases. In addition, basically, the method and purpose of data handling should be explicitly presented to the data subject, and consent should be obtained when using personal data. For this reason, easy-to-understand privacy metrics such as k-anonymity are widely used.

Data utility depends on contexts and it is difficult to evaluate the general purpose utility. There are several types of de-identification operations, such as generalization, suppression, anatomization, permutation, and perturbation [EAZS18]. The common purpose is to contort data and ambiguate the existence of the data. Thus, by any measure, the amount of information in the data is reduced and the utility is diminished. Some utility metrics based on the distance between the original data and the de-identified one are proposed [XWP$^+$06b].However, if the purpose of the detailed analysis of the data is fixed, these utility metrics may not be appropriate. In an extreme case, data in which only the principal components are significantly processed, but other information is maintained, are judged to have high utility in distance-based utility indices. However, it is easy to imagine that the intended analysis cannot be performed with this data. It is important to consider the purpose of use of data and to define the utility metric as well as the privacy metric. However, as is often the case in actually, a data analyst sometimes can not decide the plan for data utilization until he/she obtains the data.

In this dissertation, we discuss de-identification techniques that aim to achieve both privacy and usefulness. In particular, we will consider privacy and utility metrics that are generally easy to understand, especially for obtaining convincing consent. First we consider the characteristics of the data according to the data format, and classify the data format into structured data and unstructured data. Furthermore, for each data format, we classify it into dynamic data and static data. Our approach to structured data is flexible data processing through a combination of de-identification methods. There have been several studies on the combination of de-identification methods. However, they are all methods based on differential privacy. In this dissertation, we consider the probability that an individual will be re-identified as the privacy definition. This is a generalization of the idea of privacy given by k-anonymity and is an intuitive definition. As mentioned earlier, k-anonymity is a very widely adopted and convincing privacy metric. Although the vulnerability of k-anonymity has been pointed out, it can protect privacy together with legal regulations and system design measures. Although k-anonymity can evaluate generalization and suppression, it cannot evaluate other de-identification methods such as perturbation. Therefore, we perform attack simulations on de-identified data that applied a combination of de-identification methods and obtain the probability that an individual is re-identified. This makes it possible to intuitively evaluate the privacy of complex anonymized data.

Our proposals to unstructured data, here dealing with document data, is a simulation attack algorithm with a powerful attacker model and its defense algorithm. Since document data is likely to be available to the public in some cases, attackers with various background knowledge can be assumed. Previous studies on documents were mainly aimed at approaching documents that were manually de-identified by experts. In other words, the utility of a de-identified document was evaluated by the distance from the manually de-identified document. However, we considered that there is still a risk in manually de-identified documents, and proposed an attack algorithm for documents. Our attack model uses an external database to attempt to identify the names of individuals associated with the target document data. Our privacy-preserving algorithm generalizes or removes only the words that pose a risk of re-identification of individuals. Thus, it is possible to guarantee stronger privacy while maintaining its usefulness.

Finally, we take a different approach: we investigate differential privacy and secure computation. Differential privacy is a privacy metric based on information theory, and some mechanisms to satisfy differential privacy have already been put to practical use. Differential privacy mechanisms can be easily constructed, but simple mechanisms greatly reduce their usefulness. Therefore, differential privacy mechanisms often need to be focused on a specific analysis and designed to have less impact on the results of the analysis. Privacy protection by differential privacy with the appropriate parameters can be powerful and has attracted the attention of many researchers. Secure computation can be used to evaluate a specific function while keeping the data secret. This makes it possible for institutions that possess the data but do not have the analytical methods to outsource the analysis. Furthermore, secure computation allows multiple institutions to conduct large-scale analysis using the data that each of them possesses. Thus, secure computation is a powerful technology for making effective use of data.

1.2 Contributions

This dissertation deals with a study on utility-aware privacy preserving techniques, and focuses particularly on de-identification techniques.

We first discuss the concept of privacy and the need for privacy protection in Chapter 1. Then, in Chapter 2, we introduce the privacy protection techniques that have been studied so far, roughly classified into k-anonymization-based, differential privacy-based, and secret computation. In Chapter 2, we also introduce the related research required for this dissertation. Then, in Chapter 3, we introduce related research on the main privacy protection techniques and consider the existing issues. Fig. 1.1 shows the overall picture of the privacy-preserving techniques covered in this dissertation. Input privacy-preserving technique often refers to technology that directly processes data to protect its privacy, and we mainly deals with this technique. This technology is mainly for data collected from data subjects to trusted institutions, and is mainly used by companies and organizations. We classify datasets into $2 \times 2 = 4$ categories based on their structure and data types. More specifically, we classify the structure of the dataset into structured data and unstructured data, and the type into static data and dynamic data. We consider privacy-preserving techniques in each category in Chapter 4 to 6. In Chapter 4, we focus on structured static data, which is the most widely studied data type. In Chapter 5, we discuss structured dynamic data. In Chapter 6, we discuss de-identification methods for unstructured data, and although Chapter 6 mainly deals with static data, this idea can be applied to dynamic data as well. In this way, each category of data will be exhaustively discussed in each chapter.

Differential privacy also aims primarily at protecting privacy information contained in data when that data is stored in a trusted institution. We will discuss differential privacy technique in Chapter 7. There is a very similar idea to differential privacy called local differential privacy, which is also an input privacy-preserving technique, and data subjects can use the technique to provide data to untrustworthy institutions.

As another technology to facilitate the use of data, we will also touch on secure computation in Chapter 8. This is a powerful tool for secretly integrating and analyzing distributed data, or outsourcing analysis, while protecting privacy.

Chapter 4: De-identification Technique for Static Data

Publications included in this chapter are international conferences paper as follows:

- Tomoaki Mimoto, Shinsaku Kiyomoto, Katsuya Tanaka, and Atsuko Miyaji. (p, N)-identifiability: Anonymity under Practical Adversaries. *2017 IEEE Trustcom/BigData*

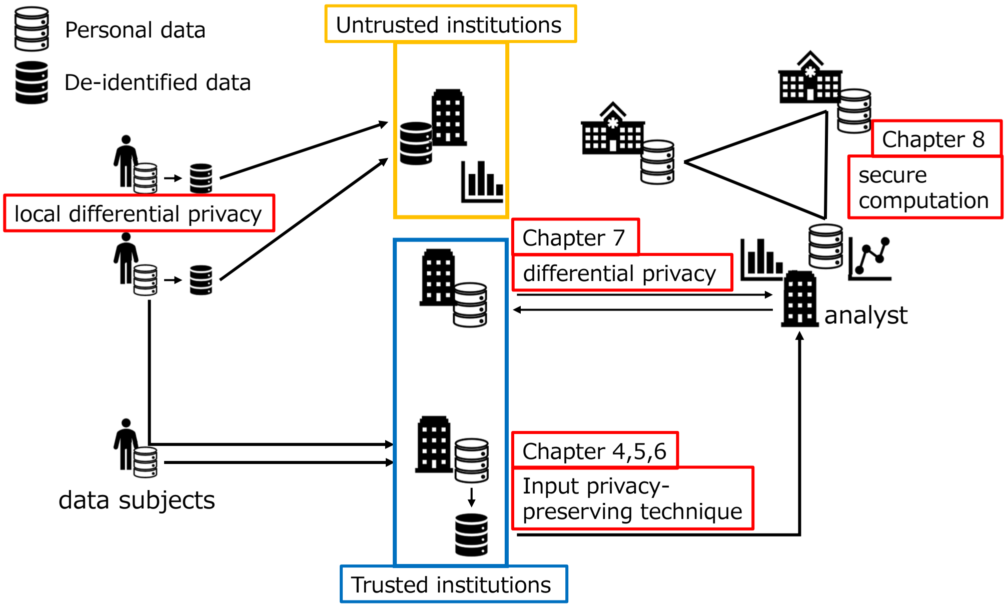


Figure 1.1: Overview of the dissertation.

SE/ICESS, 996-1003, 2017., (**Ref.** [MKTM17])

- Tomoaki Mimoto, Anirban Basu, and Shinsaku Kiyomoto. Towards Practical k-Anonymization: Correlation-based Construction of Generalization Hierarchy. *SECRYPT 2016*, 411-418, 2016., (**Ref.** [MBK16])

This chapter investigates de-identification techniques for structured static data. This kind of techniques have been studied for long time and many types of de-identification techniques are applied in combination in actual use-cases. However, the metric of the privacy risk for such the de-identification data is not established and the balance between privacy and utility is not clear.

We consider a practical-use case of de-identified datasets and define a realistic adversary model for such situations. In our model, we estimate the re-identification probability of each record and deal with de-identified datasets generated by various de-identification methods. The privacy risk should be as clear as k-anonymization because the risk must be understood by the data subjects who are the people in the dataset. We define the reciprocal of the number of candidates as the re-identification probability. We also define a utility metric based on machine learning and the utility of anonymized datasets in a practical-use case. We conduct experiments whose results suggest that a complex de-identification method is a practical and effective way to generate more useful de-identified datasets.

Chapter 5: Anonymization Technique for Dynamic Data

Publications included in this chapter are a journal paper and an international conference paper as follows:

- Tomoaki Mimoto, Seira Hidano, Shinsaku Kiyomoto, and Atsuko Miyaji. Anonymization

Technique based on SGD Matrix Factorization. *IEICE Trans. Inf. Syst.* 103-D(2): 299-308, 2020. (**Ref.** [MHKM20])

- Tomoaki Mimoto, Seira Hidano, Anirban Basu, Shinsaku Kiyomoto, and Atsuko Miyaji. The Possibility of Matrix Decomposition as Anonymization and Evaluation for Time-sequence Data. *16th Annual Conference on Privacy, Security and Trust (PST 2018)*, 1-7, 2018. (**Ref.** [MKH$^+$18])

This chapter investigates de-identification techniques for structured dynamic data. Dynamic data includes time-sequence data as opposed to static data and has high utility value. The high dimensional data has much more information than static data and it also presents a high risk leakage of private information even if some attributes are generalized or deleted. Therefore, it is necessary to decide the purpose of utilization of a time-sequence data and to select only the required information. For example, when car insurance companies calculate insurance premiums, the angle of the steering wheel and the acceleration of the car are important, not the position of the car; moreover a retail store developing some product analysis may consider the purchased goods and the number of the items more important than the purchase dates.

We propose matrix factorization as an de-identification methods and evaluate its effects on privacy risk. Dynamic data is high dimensional data and we consider privacy information and other information can be divided by dimension reduction techniques, such as principal component analysis (PCA) and singular value decomposition (SVD). With respect to the web access data, we claim that our method almost never destroys the correlation of web pages, and therefore, preserves the utility in the de-identified dataset. We compare our de-identification methods with existing methods by experiments and observe that matrix decomposition is a valid de-identification method. Our method can be easily combined with other de-identification methods as in the previous chapter, and its evaluation was also conducted. Since the proposed method is a type of perturbation, the privacy strength is defined as the probability that an individual will be re-identified, as in the previous chapter, and the privacy was evaluated by a simple re-identification simulation. Furthermore, we also consider the linkage risks between pseudonymized datasets. There are few studies about such risks and our experiments show that additive noise is one of the efficient measures against a linkage attack.

Chapter 6: Anonymization Technique for Unstructured Data

Publications included in this chapter are a journal paper and an international conference paper as follows:

- Tomoaki Mimoto, Masayuki Hashimoto, Shinsaku Kiyomoto, Koji Kitamura, and Atsuko Miyaji. Privacy Risk of Document Data and a Countermeasure Framework. *Journal of Information Processing*, vol.29, 778-786, 2021. (**Ref.** [MHK$^+$21])

- Tomoaki Mimoto, Shinsaku Kiyomoto, Koji Kitamura, and Atsuko Miyaji. A Practical Privacy-preserving Algorithm for Document. *IEEE 19th International Conference on*

Trust, Security and Privacy in Computing and Communications (TrustCom 2020), 1376-1383, 2020. (**Ref.** [MKKM20])

This chapter investigates de-identification techniques for unstructured static data. The data including privacy information is not always structured. Document data is a typical example and a huge number of documents such as news articles, public reports, tweets on SNS, and personal essays have been released on websites and social media. Once privacy-sensitive information is leaked, the effects are permanent, so great care must be taken with documents before they are released. In many cases, human experts perform redundancy and sanitization of documents before they are released, but this method can be inefficient in terms of cost and accuracy. Furthermore, such measures do not guarantee that significant privacy risks will be eliminated from the documents. In this work, we devise an algorithm to attack documents using web search engines and propose a framework to preserve privacy against such attacks. We assessed the privacy risk of actual accident reports from schools and court documents. Experiments on these datasets show that privacy risks exist even in artificially sanitized documents, and that the proposed method contributes to risk reduction. Furthermore, although we are focusing on static data, it can also be applied to dynamic data such as Twitter text.

Chapter 7: Differential Privacy Mechanism

Publication included in this chapter is an international conference paper as follows:

- Tomoaki Mimoto, Masayuki Hashimoto, Hiroyuki Yokoyama, Toru Nakamura, Takamasa Isohara, Ryosuke Kojima, Aki Hasegawa, and Yasushi Okuno. Differential privacy under incalculable sensitivity. *IEEE 6th International Conference on Cryptography, Security and Privacy (CSP 2022)*, 2022. (**Ref.** [MHY$^+$22])

This chapter investigates differential privacy mechanisms. In constructing a differential privacy mechanism, it is necessary to derive the query-specific sensitivities. The sensitivity is the impact of one record in a given dataset on the query. As an example of a query where the derivation of the sensitivity is the simplest, in the case of constructing a histogram, the impact of an arbitrary record on the result is at most 1. However, when considering complex queries, there are cases where deriving sensitivities is difficult. We attempt to solve this problem by adding dummy data to the dataset, and propose a definition accordingly.The only difference from the conventional definition of differential privacy is that the dummy data is fixed, and various existing differential privacy mechanisms satisfy the proposed definition. Furthermore, we focus on the t-test as a case study and propose a differential privacy mechanism that queries the t-value according to the defined model.

Chapter 8: Secure Computation

Publications included in this chapter are a journal paper and an international conference paper as follows:

- Nuttapong Attrapadung, Goichiro Hanaoka, Shinsaku Kiyomoto, Tomoaki Mimoto, Jacob C. N. Schuldt. A Taxonomy of Secure Two-Party Comparison Protocols and Efficient Constructions. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 102-A(9) 1048—1060, 2019. (**Ref.** [AHK+19])

- Nuttapong Attrapadung, Goichiro Hanaoka, Shinsaku Kiyomoto, Tomoaki Mimoto, Jacob C. N. Schuldt. A Taxonomy of Secure Two-Party Comparison Protocols and Efficient Constructions. *15th Annual Conference on Privacy, Security and Trust (PST 2017)*, 215-224, 2017. (**Ref.** [AHK+17])

This chapter investigates basic secure computation protocols. The first general solutions for the problem of secure computation were presented by Yao[Yao86] for the two-party case. This protocol is well known as the millionaire's problem and has been improved, expanded, and constructed in a variety of different ways.

In this chapter, we focus on the basic comparison protocol. There are 216 types of comparison protocols, and we describe conversions among these types. While these conversions are based on known techniques and have been explored previously, either explicitly or implicitly, the combination of these conversion techniques makes the two-party comparison protocol by Nergiz et al. [NNPC10] into a very efficient protocol in a configuration where the two parties hold shares of the values being compared, and obtain a share of the comparison result. This setting is commonly used in multi-party computation protocols, and hence in many privacy-preserving applications. Furthermore, we implement the protocol and measure its performance. The results show that in this input-output configuration, where no offline pre-computation is allowed, our protocol outperforms the previously proposed protocols.

1.3 Organization

The remainder of this dissertation is organized as follows:

- **Chapter 2** describes major privacy-preserving techniques and metrics. We introduce de-identification techniques and secure computations as preliminaries for our proposal and discussions.

- **Chapter 3** presents the major related studies. These are closely related to our work, and we will address the remaining issues in these studies or use them to compare the performance of new proposals.

- **Chapter 4** presents new results on de-identification technique for structured static data. We first define an adversary model and a privacy metric that can evaluate the privacy of a dataset even if the dataset is de-identificated by combined methods. We finally show our experimental results to confirm that combining de-identification techniques balances the privacy and utility.

- **Chapter 5** presents new results on de-identification technique for structured dynamic data. We first define adversary models for trajectory data, and then, propose a de-identification technique based on matrix factorization. We finally show our experimental results to confirm the effect of the proposal.

- **Chapter 6** presents new results on de-identification technique for unstructured data. We first propose an attack simulator using web search engine, and then, propose a privacy-preserving algorithm against the simulation attack. We finally show human-sanitized documents still contain privacy risks and that our algorithm can contribute to risk reduction.

- **Chapter 7** presents new results on differential privacy mechanism. We first consider a case where sensitivity is difficult to derive. Then, we design a differential privacy definition with dummy data to deal with such situations. Finally, as a concrete example, we take the t-test as an example and construct a differential privacy mechanism using the t-value as a query. The required noise variance depends on the number of records, the variance of the dataset, and the value range, and the experiments confirm that the implemented mechanism works correctly.

- **Chapter 8** presents new results on two-party computation protocol. We propose a new comparison protocol, which has shared inputs and shared output. Since we have summarized the classification and transformation of comparison protocols, they can be transformed into various formats based on the proposed protocol. We implement the protocol (and the conversion version) and measure the performance. The result shows the protocol outperforms the previous works.

- **Chapter 9** concludes this dissertation by summarizing our results and future works.

Chapter 2

Preliminary

In this chapter, we introduce previous research on basic privacy-preserving techniques. First, in Section 2.1, we introduce some privacy metrics. By quantitatively expressing how secure data is, it is possible to balance privacy and utility. In this section, we will introduce two types of privacy metrics: one derived from k-anonymity and the other based on information theory. Next, in Section 2.2, we introduce de-identification methods, i.e., methods to process data so that it satisfies the privacy metrics. Section 2.3 touches on secure computation as a different direction of privacy-preserving techniques. This dissertation deals with the most basic two-party comparison protocol, and we introduce the peripheral techniques.

2.1 Privacy Metrics

2.1.1 Privacy metrics for quasi-identifier

k-anonymity is a basic idea of de-identification technique and the definition is simple. A dataset is said to have k-anonymity if each record is indistinguishable from at least $k-1$ other records with respect to certain identifying attributes called quasi-identifiers.

Definition 2.1 (quasi-identifier [SS98]). *Let $D[a_1, ..., a_m]$ be a dataset and $A = \{a_1, ..., a_m\}$. A quasi-identifier of D is a set of attributes $\{a_1, ..., a_l\} \subseteq A$ the release of which must be controlled.*

Definition 2.2 (k-anonymity [SS98]). *Let $D[a_1, ..., a_m]$ be a dataset and QI be the quasi-identifers associated with it. D is said to satisfy k-anonymity iff for each quasi-identifier $qi \in QI$, each sequence of values in $D[qi]$ appears at least with k occurrences in $D[QI]$.*

This definition is simple and easy to understand, so that the privacy metric is widely referred in official documents[fHI10, Off12, OfCRS12, Ser13, NIS15]. There are several ways to achieve k-anonymity. Quasi-identifiers can be represented as hierarchies and one common way to satisfy the k-anonymity criterion is to reduce the detail of quasi-identifiers, i.e., to reduce the precision of the variables as they move up the hierarchy. For example, a less precise representation of “baseball” and “football” is “ballgame”. Numeric variables can be represented hierarchically,

Table 2.1: Example of a static data.

ID	Age	Gender	Address	Disease
1	18	M	Tokyo	AIDS
2	22	M	Saitama	AIDS
3	36	F	Kyoto	Diabetes
4	38	M	Osaka	Heart disease
5	88	F	Okinawa	Cancer

Table 2.2: Example of a k-anonymized data.

Age	Gender	Address	Disease
20	M	Kanto	AIDS
20	M	Kanto	AIDS
36	-	Kansai	Diabetes
36	-	Kansai	Heart disease

e.g., discrete ages can be converted to intervals such as [31-40], [41-45] etc. Furthermore, numeric variables can be changed into mean, median, and mode of a cluster.

Table 2.1 shows an example of personal data. In this example, ID is a identifier, which is information that explicitly identifies an individual, and age, gender, and address are quasi-identifiers. Disease is a sensitive attribute because we generally do not know the diseases that individuals are suffering from, but we need to set quasi-identifiers and sensitive attributes according to the situation.

Table 2.2 represents a k-anonymized data of Table 2.1. The identifier is removed and the quasi-identifiers are processed for all records. Age is changed to the average of clusters, and gender and address are generalized. Moreover, the outlier, which is ID: 5, is removed. This dataset has 2-anonymity because each cluster, which has the same set of quasi-identifiers, has at least two records.

It is important to note that the terms "anonymization" and "de-identification" defined in ISO 25237: 2017 [ISO] and k-anonymity is a "de-identification" technique based on those definitions. A generalization algorithm that transforms a dataset into one with k-anonymity naturally involves the loss of information in that dataset. Therefore, minimizing this information loss is a challenging problem in the design of generalization algorithms. This optimization problem is called the k-anonymity problem; Meyerson reported that optimal generalization in this respect is an NP-hard problem [MW04]. Furthermore, Aggarwal et al. proved that it is NP-hard to find an optimal table containing more than three attributes [AFK+05]. Nevertheless, k-anonymity has been widely studied because of the simplicity of the concept [AF05, MGK06, MGK07, WLFW06, TV06, SWL+08].

2.1.2 Privacy metrics for sensitive attribute

k-anonymity is a privacy metric for quasi-identifiers and has vulnerability against sensitive attributes. Homogeneity attack and background knowledge attack are well known attacks for k-anonymized datasets [MGK06].

- Homogeneity attack: Assume a k-anonymized dataset that has a sensitive attribute. k-anonymization algorithm processes the quasi-identifiers to have clusters whose size are at least k, and maintains the sensitive attribute for analyze. If the sensitive attribute of a cluster is the same, for instance AIDS, and an attacker know the quasi-identifiers of a person and his data is included in the cluster, the attacker infers the person has AIDS.
- Background knowledge attack: Quasi-identifiers and sensitive attributes sometimes have associations. Assume a cluster of a k-anonymized dataset has more than one sensitive data, such as virus and heart disease. Furthermore, assume a quasi-identifier is race, and a person who is Japanese is in the cluster. An attacker can not infer whether the person has a virus or heart disease without additional information, but it is well known that Japanese have an extremely low incidence of heart disease. Hence, the attacker can infer the person has a virus.

To overcome the problem, many privacy metrics are proposed [WF06, NCN08, MGK07], and the most representative metric is l-diversity.

Definition 2.3 (l-diversity principle[MGK06])**.** *An equivalence class is said to have l-diversity if there are at least l "well-represented" values for the sensitive attribute. A table is said to have l-diversity if every equivalence class of the table has l-diversity.*

[MGK06] also defines what are "well-represented" values for sensitive attributes in the following ways: entropy l-diversity, recursive (c, l)-diversity, positive disclosure-recursive (c, l)-diversity, and negative/positive disclosure-recursive (c_1, c_2, l)-diversity toward. Entropy l-diversity, for example, defines entropy of an equivalent class and requires every class has entropy above a threshold.

2.1.3 Privacy metrics based on information theory

In addition to the privacy metrics for quasi-identifiers and sensitive attributes, privacy metrics based on information theory are also proposed.

Dalenius [Dal77] discussed information theoretic security on similar databases, but Dwork [Dwo06] showed it is unrealizable. Dwork proposed the definition of differential privacy on the basis of the discussion. Differential privacy is a promising data perturbation technique based on the statistical distance between two database tables that differ by at most one record. The basic idea is that an adversary who has access to a dataset, regardless of background knowledge, will draw the same conclusions regardless of whether his or her data is included in the dataset. Differential privacy can also be used as a kind of de-identification method, but it has been studied mainly in connection with perturbation methods in interactive settings. The definition of differential privacy is as follows.

Definition 2.4 (ϵ-differential privacy[Dwo06]). *Let ϵ be a positive real number and M be a randomized mechanism that takes a dataset as input. The mechanism M is said to enjoy ϵ-differential privacy if, for all datasets D and D' that are neighbor, which is the hamming distance of them $d(D, D') = 1$, and all subsets $\mathcal{D}$ of image of M, $\mathcal{D} \subseteq Range(M)$.*

$$Pr[M(D) \in \mathcal{D}] \leq e^{\epsilon} \cdot Pr[M(D') \in \mathcal{D}]. \tag{2.1}$$

When a mechanism enjoys differential privacy, the information of a record does not affect the output of M intuitively. However, some types of queries are affected by a record and the noise to be added may be too large. Furthermore, researchers have studied about the problem that some queries are affected by a record become large. The effect of a record is called sensitivity and the definition of the sensitivity is as follows.

Definition 2.5 (global sensitivity [Dwo06]). *Let $f : \mathcal{D}^{|\chi|} \rightarrow \mathbb{R}^k$ be a query and $|| \cdot ||_p$ be a L_p norm function over the range of f. The sensitivity GS_f is defined as*

$$GS_f = \max_{d(D,D')=1} ||f(D) - f(D')||_p. \tag{2.2}$$

Sensitivity is the value of the noise criterion that is added. The smaller it is, the less influence a record has on that query. When the query asks for a count of records that satisfy a certain condition for D, the effect of a record is at most 1. However, when the query is calculating the max of D and the value range of records $r \in D$ is $r \in [0, m]$, that of a record is at most m, and the sensitivity becomes too large. Researchers have studied about the problem that the sensitivity become large, and Nissim et al. proposed the smooth sensitivity framework. The difference between the local sensitivity and the global sensitivity is D is given as input.

Definition 2.6 (local sensitivity [NRS07]). *Assume a query $f : \mathcal{D}^{|\chi|} \rightarrow \mathbb{R}^k$, a norm function $|| \cdot ||_p$, and D are given. The local sensitivety LS_f is defined as*

$$LS_f(D) = \max_{D':d(D,D')=1} ||f(D) - f(D')||_p. \tag{2.3}$$

If we use the definition directly to differential mechanism, attackers may infer whether the dataset is D or D'. Therefore, the smooth sensitivity framework smooths the scale of noise across neighboring datasets.

Definition 2.7 (smooth sensitivity [NRS07]). *Assume a query $f : \mathcal{D}^{|\chi|} \rightarrow \mathbb{R}^k$, D, and a number β are given. The β smooth sensitivity of f is defined as*

$$S_{f,\beta}(D) = \max_{AnyD'} \exp\left(-\beta \cdot d(D, D')\right) \cdot LS_f(D'). \tag{2.4}$$

Other privacy frameworks, such as [CM06], were proposed and these definition are included in (ϵ, δ)-differential privacy [DKM$^+$06].

Definition 2.8 ((ϵ, δ)-differential privacy[DKM⁺06]). *Let ϵ and δ be a positive real number and M be a randomized mechanism that takes a dataset D as input. The mechanism M is said to enjoy (ϵ, δ)-differential privacy if, for all datasets D and D' that are neighbor, and all subsets S of image of M.*

$$Pr[M(D) \in S] \leq e^{\epsilon} \cdot Pr[M(D') \in S] + \delta. \tag{2.5}$$

Local differential privacy [DJW13] is a kind of differential privacy. Each data object adds noise to their data. Data controller is untrusted and each data objects send their data to the controller in differential privacy model. This model is already in use such in Google, Apple [EPK14, CJK⁺18].

Definition 2.9 (ϵ-local differential privacy[DJW13]). *Let ϵ be a positive real number and M be a randomized mechanism that takes a record as input. The mechanism M is said to provide ϵ-local differential privacy if, for all records r, $r' \neq r$, and*

$$Pr[M(r) \in S] \leq e^{\epsilon} \cdot Pr[M(r') \in S]. \tag{2.6}$$

2.2 De-identification Techniques

2.2.1 k-anonymization

k-anonymization[SS98, Sam01, Swe02a] is one of the well-known de-identification techniques. The property of k-anonymity is that in a given dataset, there are at least k more records with the same quasi-identifier value combination.

There are several types of de-identification operations, such as generalization and suppression, anatomization, permutation, and perturbation [EAZS18].

- Generalization: It is a method of generalizing data based on a hierarchical tree constructed mainly for each attribute. Quasi-identifiers can be represented as hierarchies and one common way to satisfy the k-anonymity criterion is to reduce the detail of quasi-identifiers, i.e., to reduce the precision of the variables as they move up the hierarchy. For example, a less precise representation of “baseball” and “football” is “ballgame”. Numeric variables can be represented hierarchically, e.g., discrete ages can be converted to intervals such as [31-40], [41-45] etc. Furthermore, top/bottom coding is also known as a generalization technique. These generalize outliers that are too large or too small for the value of each attribute. For example, in the case of age, all records that are 80 years old or older are set to 80 years old.

- Suppression: This is a technique to remove some attribute values or records. It is used for outliers that reduce the amount of information significantly when generalized. For example, assume the case where most of the records have values between 50 and 60, and only one record x has a value of 80. In this case, rather than generalizing each record to [50-80], it is more reasonable to remove x and generalize each record to [50-60].

- Anatomization: This is a method of managing quasi-identifiers and sensitive attributes in separate tables. This does not directly perform k-anonymization, but it can reduce the risk of attacks on sensitive attributes.

- Permutation: This is also known as data swapping, which is a method of replacing attribute values between records. The correlation between attributes will be distorted, but the percentage of attribute values in the total data will not change.

- Perturbation: It is a method of adding noise to attribute values. Micro-aggregation, which converts the data to the mean or median of each cluster, can be considered as one of these methods. This method is often used in the context of local differential privacy, but its utility is drastically reduced if the noise level is not well controlled.

For example, some techniques based on space division [IN07, LDR06] and on clustering [BKBL07a, HCC$^+$12, LW08] have been proposed to achieve k-anonymity. In space division techniques, records are represented as points in a multidimensional space, and the space is divided so that every space has at least k records. kd-tree [FBF09] or R-tree [Gut84] are usually used as methods to divide the space. These methods are fast, but they do not take into account the distance between points, so points that are some distance apart may be placed in the same space, leading to information loss. In clustering-based methods, the distance between records is taken into account, but every group must contain more than k records. Therefore, the cluster area is expanded and information loss may occur. In particular, the Incognito algorithm [LDR05] and the Mondrian algorithms [LDR06] are well known as k-anonymization algorithms. The Incognito algorithm uses taxonomy trees to generalize attributes, and the Mondrian algorithm averages or replaces representative values of the original data to achieve k-anonymization. However, these methods mainly deal with generalization and suppression, and cannot perform other de-identification methods.

2.2.2 Noise addition

Noise addition works by adding or multiplying a probabilistic or random number to sensitive data [Miv13]. The idea is simple and is also well-known as a re-identification technique. The first work on noise addition was proposed by Kim[Kim86] and the idea was to add noise ϵ with a distribution $\epsilon \sim N(0, \sigma^2)$ to original data. Additive noise is uncorrelated noise, which means that the mean and covariance of the original data are maintained, but the correlation coefficient and variance are not. Another variation of additive noise is correlated additive noise, which preserves the mean of the original data and allows the correlation coefficient to persist [YJ07].

The most famous technique for adding noise is the differential privacy mechanism. It is known that adding noise to the answer of a query satisfies the definition of differential privacy. The Laplacian mechanism [DMNS06] is a popular ϵ-differentially private mechanism for queries f.

Theorem 2.1 ([Dwo06]). *Let query $q : \mathcal{D}^{|\mathcal{X}|} \to \mathbb{R}^k$ and $L(\mu, v)$ be the Laplacian noise with mean μ and variance v, then the following mechanism M_q satisfies $(\epsilon, 0)$-DP.*

$$M_q(D) = q(D) + L\left(0, \frac{GS_q}{\epsilon}\right). \tag{2.7}$$

Moreover, it is also proven the following mechanism achieves (ϵ, δ)-differential privacy.

Theorem 2.2 ([NRS07]). *Let query $q : \mathcal{D}^{|\mathcal{X}|} \to \mathbb{R}^k$. When $\alpha = \frac{\epsilon}{5\sqrt{2\ln 2/\delta}}, \beta = \frac{\epsilon}{4(k+\ln 2/\delta)}$, the following mechanism M_q satisfies (ϵ, δ)-DP.*

$$M_q(D) = q(D) + \frac{S_{q,\beta}(D')}{\alpha} \cdot N(0, 1). \tag{2.8}$$

Furthermore, there are some approaches to reduce the sensitivity [RN10, KP13, VSBH13]. In [RN10], the differential privacy mechanism is adapted to Fourier coefficient, and in [KP13], the differential privacy mechanism is adapted to clusters that is adjusted. One major approach is to convert a dataset into other data formats in this way, and there are many differential privacy mechanism fused with machine learning model in the same way [ZZX+12, VSBH13, WLD+20]. Particularly, [WLD+20] adopts differential privacy to federated learning model and the sensitivity is considered on loss function.

In order to build more effective mechanisms, several useful theorems for differential privacy have been proposed. [McS09] provides sequential and parallel theorem and the techniques are applied such in [BS15].

Theorem 2.3 ([McS09]). *Let M_i be a (ϵ_i, δ_i)-differential privacy mechanism and f be any function. $M(D) = f(M_1(D), M_2(D, M_1(D)), \cdots, M_n(D, M_1(D), M_2(D), \cdots, M_{n-1}(D)))$ satisfies $(\sum_{i=1}^n \epsilon_i, \sum_{i=1}^n \delta_i)$-differential privacy.*

Theorem 2.4 ([McS09]). *Let M_i be a ϵ-differential privacy mechanism and a partition of a dataset D be D_i. $M(D = \cup D_i) = g(M_1(D_1), M_2(D_2), \cdots, M_n(D_n))$ satisfies ϵ-differential privacy.*

Furthermore, the sampling theorem [LQS12] is also widely known and used such in [KP13].

Theorem 2.5 ([LQS12]). *Assume a mechanism M^β gives (ϵ, δ)-differential privacy, where M^β denotes the mechanism to first sample with probability β, and then apply a mechanism M. Any mechanism M^β satisfies (ϵ', δ')-differential privacy, where $\epsilon' = \ln(1 + \beta(e^\epsilon - 1))$, and $\delta' = \beta\delta$.*

Differential privacy is the state-of-the-art in research area and it is turned to practical use [JNS18].

2.2.3 Sampling

Sampling is a widely used technique to preserve privacy in the statistics field. A sampled dataset preserves privacy to a greater extent than a non-sampled dataset because unique records included in a sampled dataset may not be unique in the original dataset. Therefore, an attacker cannot link the records to the target completely even if there are unique records in an anonymized dataset. In the statistics field, the relationship between an original dataset and a sampled dataset has been researched for a very long time. Greenberg et al. [GZ92] proposed the probability model of re-identification and, in the model, the probability that a unique record is also unique in the population is taken into account. Furthermore, the relationship between differential privacy and sampling, as described in [LQS12] and [CM06], is theoretically evaluated.

2.2.4 Matrix factorization

Matrix factorization is an essential technique for data analysis, and is widely used in text data mining and acoustic analysis. Matrix factorization converts one matrix into two matrices by solving an optimization problem. Therefore, it is not possible to completely recover the original matrix from the resulting matrix. This can be thought of as a kind of data perturbation. We use a matrix factorization as a de-identification technique, so we present the overview of a matrix factorization here.

Let an unknown rank r matrix be $M \in \mathbb{R}^{n\times m}$, of which we know set $\Omega \subset [n] \times [m]$ of elements. $P_\Omega(M) \in \mathbb{R}^{n\times m}$ is defined below:

$$P_\Omega(M) = \begin{cases} M_{ij} & \text{if}(i,j) \in \Omega, \\ 0 & \quad \text{otherwise.} \end{cases} \tag{2.9}$$

The purpose of matrix factorization is to find two matrices $U \in \mathbb{R}^{r\times n}$ and $V \in \mathbb{R}^{r\times m}$ which approximate the original matrix $M_{ij} \approx X_{ij} \quad \text{s.t.} \quad \forall M_{ij} \in \Omega(M)$ with lower dimensionality $r \ll \min(n,m)$. Here, $X = U^\mathsf{T} V$. This is defined to solve the following optimization problem:

$$\min_{u^*,v^*} \sum_{(i,j)\in P_\Omega(M)} (M_{ij} - u_i^\mathsf{T} v_j)^2 + \lambda(||u_i||^2 + ||v_j||^2), \tag{2.10}$$

where u_i is a user-factors vector and v_j is an item-factors vector. When u_i and v_j are variables, this function is not a convex set, so that the problem described above cannot be solved. The equation is widely known and you can find details in some references (e.g. [BK07]).

Several methods have been proposed to solve this problem, for example, the gradient descent method is a basic method to find the local minimum [NW06]. However, it is necessary to update the vectors iteratively to obtain the optimal solution, and using the gradient descent method is computationally expensive, so the stochastic gradient descent (SGD) method is widely used, such as KDDCup 2011 [DKKW11] and Netflix Prize [BK07]. There are several studies on speeding up SGD-based matrix factorization such as [RRWN11, GNHS11, ZCJL13, OHYJ15] and each algorithm updates the matrices in parallel or in a distributed manner. Furthermore, there are many derivative version of matrix factorization [LS99, Kor08, Kor09, LZXZ14]. Non-negative matrix factorization (NMF) [LS99] is a matrix factorization especially used in image processing and the values of each vector are non-negative. Many values in the real world are non-negative and NMF is suitable in these cases. NMF is defined to solve the following optimization problem.

$$U, V = \min_{U,V} ||M - UV^{\mathsf{T}}||_F^2, \quad \text{s.t.} \quad \forall i, j, k, \;\; U_{ik}, V_{jk} \geq 0, \tag{2.11}$$

where $M \in \mathbb{R}_{n\times m}, U \in \mathbb{R}_{n\times r}, V \in \mathbb{R}_{m\times r}$. $||A||_F^2$ means the Frobenius norm of A and $||A||_F^2 = \Sigma_{i,j} A_{ij}^2$. As same as equation (2.10), when U and V are variables, $||M - UV^{\mathsf{T}}||_F^2$ is not a convex set, so that the problem described above can not be solved. Lee et al. [LS99, LS01] use Jensen's inequality and Taylor expansion to transform the formula so that it can be differentiated. Then, U and V are alternately optimized to obtain a local optimal solution. The NMF algorithm is described in Algorithm 1, with further details to be found in [LS99, LS01].

Algorithm 1 (M, r, I): Alternating minimization

```
Input: Original dataset $M$, rank $r$, and the number of iterations $I$.
$t = 0$
Construct $U_t \in \mathbb{R}_{n\times r}$ and $V_t \in \mathbb{R}_{m\times r}$ randomly
while $t < I$ do
    $U_{t+1} = U_t \cdot \frac{MV_t^{\mathsf{T}}}{U_t V_t V_t^{\mathsf{T}}}$
    $V_{t+1} = V_t \cdot \frac{U_{t+1}^{\mathsf{T}} M}{U_{t+1}^{\mathsf{T}} U_{t+1} V_t}$
    $t = t + 1$
end while
return $U_{t+1}, V_{t+1}$
```

The inputs M, r, and I represent the original dataset, rank, and the number of optimization iterations, respectively. Since the difference between $X = UV^{\mathsf{T}}$ and M increases with decreasing rank, we can see that rank is an important parameter. The number of iterations affects the accuracy of X as well as the rank, and is usually set so that X and M converge. As for the number of iterations, there are studies on how to increase the efficiency. Cichocki et al. proposed to update each vector [CZA07] and Hsieh et al. proposed to update by selecting important variables [HD11].

2.3 Secure Comparison

2.3.1 Multi-party computation

Multi-party computation (MPC) is a powerful cryptographic tool that is often used to obtain privacy-preserving applications such as privacy-preserving data mining (PPDM). In general, MPC allows a set of parties to jointly compute a function of their private inputs. That is, for a function f, n parties with private inputs x_i $(i = 1, \ldots, n)$, can compute $f(x_1, \ldots, x_n)$ without revealing their private inputs x_i. The security guarantee provided by MPC is so strong that a party j who obtain $f(x_1, \ldots, x_n)$ as part of the protocol will learn nothing about the inputs x_i for $i \neq j$, except the information that can be derived from $f(x_1, \ldots, x_n)$ and x_j. In short, MPC guarantees maximum input privacy, which is a highly desirable property in PPDM. However, MPC will not protect the outputs of the function. MPC provides the actual results and it may cause the information leakage. For example, Alice, Bob, and Charlie want to take a majority vote. They input $x_i \in \{X, Y\}$, which is the preference of each parson. Assume Alice and Bob input X and Charlie inputs Y. MPC evaluates the result without opening their input, but if Charlie gets the results, he obtains the information that Alice and Bob input X. Therefore, MPC and de-identification techniques such as k-anonymization and differential privacy need to distinguish depending on use-case. For example, assume a data owner has a large personal data but does not have an analysis ability. In this situation, the data owner may ask an analytical institution to analyze the data. The data is privacy information and the data owner does not trust the analytical institution. The data owner first gives the encrypted data to the analytical institution. The analytical institution then analyzes it without knowing any information about the data and returns the encrypted analysis results to the data owner. Finally, the data owner use the private key to obtain the analysis results. In this way, the data owner can analyze the data without giving any information about the data to anyone. MPC is expected to be used in outsourcing services. Moreover, combining MPC and other de-identification techniques enables to handle large scale data because MPC protects the user's input data and de-identification techniques protect the contents of the data and the analyzing results of the input data. Although MPC is theoretically feasible for arbitrary functions f [Yao86, GMW], the resulting protocols are often complex and inefficient, and difficult to use in practice. This has led to the development of custom-made MPC protocols that efficiently implement specific functions required by various privacy-preserving applications. For example, Bunn and Ostrovsky [BO07] proposed a two-party k-means algorithm in which two servers, each holding a separate data set, compute a combined k-means clustering of their data sets without disclosing the individual data points to the other server. Note that k-means clustering is a very popular method in data mining and machine learning, which is explained in [ORSS13] for example. Bost et al. [BPTG15] proposed various protocols for privacy-preserving machine learning classification, in which a server with model M and a client with input x, can jointly classify x according to M without the server revealing M to the client or the client revealing x to the server. Other examples include privacy-preserving biometric data matching [EFG+09, BFL+11] and privacy-preserving recommender systems [EVTL12]. Furthermore, the latest research proposes to combine oblivious transfer, garbled circuits, and secret sharing to secretly build and run various machine learning models [MZ17, LJLA17, RRK18, WGC19]. A common approach to the design of these protocols is to use a modular design that combines efficient sub-protocols for specific low-level operations.

2.3.2 Homomorphic encryption

There are various building blocks for implementing MPC. Additive homomorphic encryption is a commonly used building block. The syntax consists of the following algorithms. The key generation algorithm outputs a public key pk and a secret key sk. Here, we write $(pk, sk) \leftarrow \mathsf{KGen}$. The encryption algorithm applies the encryption process to a message $X \in \mathcal{M}$ and generates a ciphertext $C = Enc_{pk}(X)$. Here, a ring $\mathcal{M}$ is the message space. In the decoding algorithm, C is decoded into X using sk. Additive homomorphism allows us to homomorphically add the encrypted message as $Enc_{pk}(M_1) + Enc_{pk}(M_2)$ and decrypt it to get $M_1 + M_2$. Also, decoding $r \cdot Enc_{pk}(M)$ multiplied by a known value r gives $r \cdot M$.

2.3.3 Taxonomy of secure comparison protocols

Secure comparison is an important sub-protocol for configuring multi-party computation (MPC). In particular, we focus on secure two-party computation for comparison functionalities. We label the two parties as Alice and Bob. The correctness and security are completely defined by the inputs and the outputs of Alice and Bob. In particular, the security requires that each party learn no more than can be inferred from his or her own inputs and outputs and views of interactions with the other party. We consider the semi-honest adversary model, assuming a computationally limited adversary that tries to learn additional information from the messages it sees during the execution of the protocol. In contrast to the more powerful malicious adversary, the semi-honest adversary is not allowed to deviate from the protocol.

Research on secure comparison protocols have a vast literature, e.g., [Yao86, NO07, CDH10, GSV07, DGK08], and in particular, a survey published by Veugen et al. [VBdHE15] provides a detailed overview. There is a noteworthy related work by Bost et al. [BPTG15] on the "categorization" of comparison protocols, in which the syntax and schemes for 5 configurations, consisting mainly of encrypted input/output settings, are described.

We exhaustively describe the setting of the comparison protocol in the following.

Semantics. In a comparison protocol, we have two integer inputs x, y, where we assume $0 \leq x, y < n$ and an output value $\delta \in \{0, 1\}$ defined as

$$\delta = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{if } x < y \end{cases}.$$

Here, the integer n denotes the upper bound of x and y, and we usually assume n to be a power of 2, say $n = 2^\ell$.

We categorize types of comparison protocols by the following criteria:

- semantics of Alice's and Bob's input regarding x
- semantics of Alice's and Bob's input regarding y
- semantics of Alice's and Bob's output regarding δ

More specifically, for each of the values x, y, and δ, we will consider the following scenarios describing the data held by Alice and Bob, respectively:

ES: (Encrypted data, Secret key),

SE: (Secret key, Encrypted data),

SS: (Shared data, Shared data),

PN: (Plain data, None),

NP: (None, Plain data),

PP: (Plain data, Plain data).

That is, for a value z (which can be either x, y, δ), the considered scenarios are as follows.

- ES:

  $$\text{Align: } E_{pk}(z), \qquad\qquad \text{Bob: } sk,$$

  where $(pk, sk) \leftarrow \mathsf{KGen}$ is generated by Bob.

- SS:

  $$\text{Align: } z^A, \qquad\qquad \text{Bob: } z^B,$$

  where $z^A, z^B \in \mathbb{Z}_N$ and $z^A + z^B \equiv z \bmod N$. Here, N is a known integer. For inputs x, y, we assume N to be much larger than n. For an output δ, we use $N = 2$ (hence, the addition then becomes XOR).

- PN: Alice holds z, while Bob holds nothing (about z).

- PP: Alice and Bob both hold z.

The scenarios SE and NP are defined analogously to ES and PN.

Categorization. Using the above, we can systematically write a protocol type as XX-YY-DD, where each $\mathrm{XX}, \mathrm{YY}, \mathrm{DD} \in \mathcal{S} := \{\mathrm{ES}, \mathrm{SE}, \mathrm{SS}, \mathrm{PN}, \mathrm{NP}, \mathrm{PP}\}$ represents the semantics of the date held by Alice and Bob regarding x, y, δ, respectively. *Therefore, overall we obtain* $6^3 = 216$ *protocol types.*

For concreteness, we list the *commonly used* types:

- Type 1: SS-SS-SS,
- Type 2: SE-SE-SE,
- Type 3: PN-NP-SE,
- Type 4: PN-NP-SS,

- Type 5: PN-NP-NP,

where, for example, Type 1 is considered in [BO07, NO07, GSV07, DGK08, CDH10] , Type 2 in [DGK08, BPTG15], Type 3 in [BPTG15], while Type 5 is the original Yao's millionaire protocol [Yao82] and is also considered in [NNPC10], who consider Type 4 as well.

We also consider some type which is somewhat more specific and/or has not been considered in the literature before such as:

- Type 6: PN-SE-SS,
- Type 7: PP-SS-ES,

and give their explicit descriptions below.

We will touch on another type of protocol. Type 8 is a close relative of Type 2, but instead of encrypting the entire x, we consider encrypting each bit in the bit representation of x separately. This is the type of protocol considered in [GHJR15], and is the subject of our comparison in Chapter 8.

Type 1: Shared inputs/ Shared output.

Type 1

	Alice	Bob	Constraints
Input	x^A, y^A	x^B, y^B	$x^A + x^B \equiv x \bmod N$, $y^A + y^B \equiv y \bmod N$
Output	δ^A	δ^B	$\delta^A \oplus \delta^B = \delta$, $\delta = (x \geq y)$

Type 2: Encrypted inputs/ Encrypted output.

Type 2

	Alice	Bob	Constraints
Input	sk	$E_{pk}(x), E_{pk}(y)$	$(pk, sk) \leftarrow \mathsf{KGen}$
Output		$E_{pk}(\delta)$	$\delta = (x \geq y)$

Type 3: Plain inputs/ Encrypted output.

Type 3

	Alice	Bob	Constraints
Input	x	y	
Output	sk	$E_{pk}(\delta)$	$\delta = (x \geq y)$, $(pk, sk) \leftarrow \mathsf{KGen}$

Type 4: Plain inputs/ Shared output.

Type 4

	Alice	Bob	Constraints
Input	x	y	
Output	δ^A	δ^B	$\delta^A \oplus \delta^B = \delta$, $\delta = (x \geq y)$

Type 5: Plain inputs/ Plain output.

Type 5

	Alice	Bob	Constraints
Input	x	y	
Output		δ	$\delta = (x \geq y)$

Type 6: Plain, Encrypted inputs/ Shared output.

Type 6

	Alice	Bob	Constraints
Input	x, sk	$Enc_{pk}(y)$	$(pk, sk) \leftarrow \mathsf{KGen}$
Output	δ^A	δ^B	$\delta^A \oplus \delta^B = \delta$, $\delta = (x \geq y)$

Type 7: Plain, Encrypted inputs/ Encrypted output.

Type 7

	Alice	Bob	Constraints
Input	x, sk_1	$x, Enc_{pk_1}(y)$	$(pk_1, sk_1) \leftarrow \mathsf{KGen}$
Output	$Enc_{pk_2}(\delta)$	sk_2	$(pk_2, sk_2) \leftarrow \mathsf{KGen}$, $\delta = (x \geq y)$

Type 8: Encrypted inputs (One is bit representation)/ Encrypted output.

Type 8

	Alice	Bob	Constraints
Input	sk	$Enc_{pk}(y)$, $\{E_{pk}(x_i)\}_{i \in [1,l]}$	$(pk, sk) \leftarrow \mathsf{KGen}$
Output		$Enc_{pk}(\delta)$	$\delta = (x \geq y)$, $\mathsf{bit}(x) = (x_1, \cdots, x_l)$

Furthermore, we consider the conversion of comparison protocols with various types of inputs and outputs. Specifically, we consider building a protocol of type XX-YY-DD from another protocol of type XX′-YY′-DD′, i.e., the former uses the latter as subroutines. In this case, we write XX′-YY′-DD′ ⇒ XX-YY-DD. We also use ⇔ if a conversion in both directions is possible. Here we present an almost self-explanatory and easy to build conversion protocol, i.e., conversion protocol between plain data and shared data.

Conversion 1 (Plain → Shared *Input*).

PN → SS

	Alice	Bob	Constraints
Input	x		
Output	x^A	x^B	$x^A + x^B \equiv x \bmod N$

1. **Alice**:
   - Randomly choose $x^A \in \mathbb{Z}_N$.
   - Send $x^B = x - x^A \bmod N$ to Bob.

Conversion 2 (Shared → Plain *Output*).

SS → PN

	Alice	Bob	Constraints
Input	δ^A	δ^B	$\delta^A \oplus \delta^B = \delta$
Output	δ		

1. **Bob**:
   - Send δ^B to Alice.
2. **Alice**:
   - Compute $\delta = \delta^A \oplus \delta^B$.

Note that these capture the usual “sharing” and “reconstruction” procedures in 2-out-of-2 secret sharing schemes. However, if we rephrase them in our terminology, we get the conversions shown in the following Lemma.

Lemma 2.1. *For any XX, YY, DD* $\in \mathcal{S}$*, and any TT* $\in$ *{PN,NP,PP}, we have*

- *SS-YY-DD ⇒ TT-YY-DD.*
- *XX-SS-DD ⇒ XX-TT-DD.*
- *XX-YY-SS ⇒ XX-YY-TT.*

The first two lines of the transformation are obtained from Conversion 1. For example, if the type we are interested in is PN-YY-DD, i.e., Alice has plain x as input, then simply having Alice share x (as x^B) with Bob and then running the protocol of type SS-YY-DD will construct a protocol for this type. The conversion of the last line is done via Conversion 2. For example, if the type of interest is XX-YY-PN, we run XX-YY-SS to get the shared output, Bob sends the share to Alice, and Alice reconstructs δ. It is also straightforward to see that XX-YY-PN ⇒ XX-YY-PP by just letting Alice send her output to Bob.

Chapter 3

Related Works

In order to achieve both privacy and utility of the data, it is possible to combine the de-identification techniques introduced in Chapter 2. In this chapter, we first introduce existing research on combining de-identification techniques. In addition, we will introduce existing research on analysis by matrix factorization. Matrix factorization has not been used as a de-identification technique so far, and it is mainly used for recommendation algorithms. We will use matrix factorization as a combination of de-identification techniques.

Then, we introduce privacy protection techniques for unstructured data. We broadly classify data into four types according to their structure and type. In particular, there is little research on unstructured data, and there are many challenges. Finally, we present related work on differential privacy and secret computation for future research.

3.1 Data De-identification

3.1.1 Combination of de-identification methods

Some privacy metrics for combining de-identification methods have been proposed, and the relationships among existing metrics have been studied. Chaudhuri et al. proposed (c, ϵ, δ)-privacy, which indicates the risk of sampling in a manner similar to differential privacy [CM06]. Li et al. considered the relationship of sampling, differential privacy, and k-anonymity and proposed $(\beta, \epsilon, \delta)$-differential privacy for sampling [LQS12]. Soria-Comas et al. proposed an insensitive algorithm and correlated k-anonymity with differential privacy [SCDFSM14]. Many of these have proposed their own or differential privacy-based indicators.

In the dissertation, we focus on k-anonymity as an easy-to-understand privacy metric. However, k-anonymity cannot evaluate the impact of sampling and perturbation, which are major privacy-preserving techniques. Therefore, we consider the privacy of de-identified data as the most basic idea, which is the probability of re-identification of an individual, and propose a privacy risk evaluation by simulation attack. This is a general extension of k-anonymity and the privacy risk of a k-anonymized dataset is $1/k$. Thus, the re-identification probability of an individual can be easily understood as the strength of privacy.

Simulation attacks require an attacker model. We assume a real use case in structured data, an attacker who does not have access to the external database, but on the other hand we need to assume a strong attacker within the limitations, which is like semi-honest model. In [DFM16], the attacker is assumed to have background knowledge of the original data and the de-identification algorithm, and we follow the model.

3.1.2 Applying matrix computation to dynamic data

Several studies have used matrices in time-sequence datasets; Zheng et al.[ZZM+11, ZZXM09] proposed to predict users' interests in unvisited locations. They assumed users' GPS trajectory as a user-location matrix and each value of the matrix means the number of visits of a user to a location. The matrix is very sparse, as each user has visited only a few locations, and a collaborative filtering model is applied for prediction. Zheng et al. [ZZXY10] built a location-activity matrix M with missing values. M is decomposed into two low-rank matrices U and V. The missing values can be filled by $X = UV^{\mathsf{T}} \simeq M$, that is, for people who have not visited a certain location, it is possible to recommend that location based on the trends of other users. Chawla et al.[CZH12] constructed a graph from trajectories of the cab and converted the graph to a matrix. The authors in [Zhe15] proposed a method to identify the traffic flows that cause an anomaly between two regions.

As described above, many existing studies treat matrix factorization as a tool for data analysis. On the other hand, matrix factorization compresses the actual matrix by a given rank and distorts the data, so it can be considered as a kind of de-identification method. However, its evaluation has not been done so far.

3.1.3 De-identification for unstructured data

Unstructured data is difficult to deal with, and there has not been much research on de-identification for this type of data. As far as we have surveyed, de-identification based on k-anonymization [CGRM08, BCG+09], and de-identification based on information content [SB16] have been proposed.

[CGRM08] defines a document as consisting of entities and terms, with each entity being established based on the associated term. For example, in the case of a compendium of diseases, each disease is an entity, and the items related to that context, such as its symptoms and the drugs used to treat it, are terms. The entity to be protected is determined in advance. Under this assumption, [CGRM08] proposes the idea of K-security, which is similar to k-anonymity, and an algorithm to realize K-security. The de-identification algorithm sanitizes the terms so that at least K entities are candidates from the terms. In another work [BCG+09], a sanitization tool was proposed. This tool has two main functions: one is to understand the content of sentences using linguistic content analysis, and the other is to understand the sensitivities of the overall content using inference detection algorithms. [BCG+09] argued, through interviews and feedback, that sanitization is an alternative approach to revising documents to hide sensitive content while maintaining as much coherence and utility as possible. Another study [SB16] predefines high-risk words and sanitizes words that are highly related to those words. Different from other approaches [CG11, ACJ$^+$12], this study proposed a method to automatically detect terms that may disclose sensitive data and ensure their protection.

In these previous studies, pre-defined high-risk words and words that are highly related to high-risk words are checked, and the experimental results show that the algorithm tries to mimic manual sanitization, although it is less accurate than manual sanitization. In addition, the combination of common words that seem to have nothing to do with each other can sometimes lead to privacy leaks. For example, two common words, "soccer class" and "flood," may lead to the re-identification of the victim of an accident. Accidents in a soccer class and those caused by flooding can sometimes occur independently. However, accidents caused by flooding during soccer classes are very rare, and manual sanitization may be overlooked because the words "soccer class" and "flood" are very common. Therefore, algorithms that mimic manual sanitization may not be sufficient in terms of de-identification. Furthermore, if a sanitizer only has access to partially sanitized documents, traditional algorithms may not work effectively because they cannot accurately evaluate pointwise mutual information.

3.1.4 Differential privacy mechanism

The relationship between difference privacy and statistical analysis has been investigated in several ways. For example, it has been proposed to add noise to the aggregate (or marginal table) to guarantee differential privacy and then calculate the test statistic [WLK15, GLRV16]. Duch et al. [DJW18] proposed a t-test mechanism using a local differential privacy mechanism. The proposal exploits the property that the data to which the local differential privacy mechanism is applied follows a Bernoulli distribution and reduces the type I error.

The local differential privacy mechanism is constructed based on the upper and lower bounds of the input data's domain of definition. This is because the definition of local differential privacy needs to take into account arbitrary input data. On the other hand, the differential privacy mechanism is built based on the impact, that is, the sensitivity, of any one record on the query. Therefore, the differential privacy mechanism becomes more difficult to construct depending on the complexity of the query. Especially in statistical processing, such as t-test or derivation of correlation coefficient, considering arbitrary data sets may cause the query results to diverge, making it impossible to build the mechanism directly.

We describe the t-test here because it will be specifically treated as a case study. The t-test is a general term for tests using the t-distribution, which is a test method for the population mean. The t-distribution is used when the population variance of the normal distribution is unknown. There are not many cases where the population variance is known at the time of data analysis, and the t-test is one of the most basic tests that can be used in such a situation. There are three types of t-tests: one-sample t-test, paired t-test, and unpaired t-test. We focus especially on the most commonly used unpaired t-test, called Welch's t-test. The unpaired t-test considers two independent populations. The Welch's t-test tests whether there is a dominant difference between the means of certain attributes of two populations when the variances of the two samples are not necessarily equal.

In the following, for dataset D, each record $r_i \in D$ is contained in one of the sets A, B, i.e., $r_i \in \{A, B\}$. Each set is denoted by $A = \{a_1, \cdots, a_{n_A}\}$ and $B = \{b_1, \cdots, b_{n_B}\}$, and $\max(r_i) = m$. Furthermore, the mean and unbiased variance of records in the set $X \in \{A, B\}$ are denoted by μ_X, s_X^2. The unbiased variance of the set X can be obtained from the following equation.

$$s_X^2 = \frac{1}{n_X - 1} \cdot \sum_{i=1}^{n_X} (\mu_X - x_i)^2 \tag{3.1}$$

For the t-test, we first sets the null hypothesis $H_0 : \mu_A - \mu_B = d_0$ and the alternative hypothesis $H_1 : \mu_A - \mu_B \neq d_0$, and then determine whether to reject H_0 at the significance level α. Specifically, the t-value and p-value are calculated in this order, along with the degrees of freedom. The p-value represents the probability of the test statistic under the null hypothesis, and if the p value is less than α, the null hypothesis is rejected.

The α is generally given as 0.05 or 0.03. The t-value and the degrees of freedom df are obtained from the following equations, respectively.

$$t = \frac{\mu_A - \mu_B}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}}, \tag{3.2}$$

$$df = \frac{(s_A^2/n_A + s_B^2/n_B)^2}{\frac{(s_A^2/n_A)^2}{n_A - 1} + \frac{(s_B^2/n_B)^2}{n_B - 1}}. \tag{3.3}$$

It should be noted that although differential privacy can be satisfied by using data to which the local differential privacy mechanism is applied, the utility cannot be guaranteed by a general-purpose mechanism.

3.2 Secure Computation

In Section 2.3, we presented the setting of the comparison protocol and a part of conversion protocol. However, other types of conversion protocols are needed to design a comparison protocol with arbitrary inputs and outputs. Here we present protocols that have been introduced in existing research [Veu14, VBdHE15].

We describe a conversion from a state of shared data to an equivalent state of encrypted data, and vice versa. We assume that $\mathcal{M} = \mathbb{Z}_N$. These conversions seem to be implicitly used in the literature.[1]

Conversion 3 (Shared → Encrypted data).

[1] However, we cannot find a reference that does exactly this. A closely related conversion can be found in, e.g., [KSS14], though.

SS → SE

	Alice	Bob	Constraints
Input	x^A	x^B	$x^A + x^B \equiv x \bmod N$
Output	sk	$E_{pk}(x)$	$(pk, sk) \leftarrow \mathsf{KGen}$

1. **Alice**:
   - Generate $(pk, sk) \leftarrow \mathsf{KGen}$.
   - Send pk and $E_{pk}(x^A)$ to Bob.
2. **Bob**:
   - Compute $E_{pk}(x) = E_{pk}(x^A) + E_{pk}(x^B)$.

Conversion 4 (Encrypted → Shared data).

SE → SS

	Alice	Bob	Constraints
Input	sk	$E_{pk}(x)$	$(pk, sk) \leftarrow \mathsf{KGen}$
Output	x^A	x^B	$x^A + x^B \equiv x \bmod N$

1. **Bob**:
   - Randomly choose $x^B \in \mathbb{Z}_N$.
   - Compute $E_{pk}(x^A) = E_{pk}(x) - E_{pk}(x^B)$ and send to Alice.
2. **Alice**:
   - Decrypt $E_{pk}(x^A)$ to x^A.

Security of the conversion follows if each party does not learn more about x at the output than at the input state. Conversion 3 is secure due to the semantic security of encryption, while Conversion 4 is secure due to the randomness of x^B.

From these conversions, we have the following lemma.

Lemma 3.1. *The following conversions can be constructed in a similar way for any XX, YY, DD* $\in \mathcal{S}$.

- *SE-YY-DD ⇔ SS-YY-DD ⇔ ES-YY-DD.*
- *XX-SE-DD ⇔ XX-SS-DD ⇔ XX-ES-DD.*
- *XX-YY-SE ⇔ XX-YY-SS ⇔ XX-YY-ES.*

We describe the most sophisticated conversion so far [Veu14]: it is for PN-NP-SE ⇒ SE-SE-SE. That is, it converts Type 3 protocol to Type 2.

Consider $a, b \in \mathbb{Z}_N$, and $n << N$ [2], we denote $a \operatorname{div} n = \lfloor a/n \rfloor$, hence we have $a = (a \operatorname{div} n) \cdot n + (a \bmod n)$. Define

$$\beta_{a,b,n} := \begin{cases} 1 & (a+b) \bmod n < b \bmod n \\ 0 & (a+b) \bmod n \geq b \bmod n \end{cases}.$$

Lemma 3.2 ([Veu14])**.** $(a+b) \operatorname{div} n = a \operatorname{div} n + b \operatorname{div} n + \beta_{a,b,n}$.

The idea of the protocol is to use the fact that $0 \leq x, y < n$, and when considering $a = n + (x - y)$, we have the identity:

$$\begin{aligned} x \geq y \;&\Leftrightarrow\; a \operatorname{div} n = 1, \\ x < y \;&\Leftrightarrow\; a \operatorname{div} n = 0. \end{aligned} \tag{3.4}$$

The value $a \operatorname{div} n$ is then securely computed as

$$a \operatorname{div} n = (a+b) \operatorname{div} n - b \operatorname{div} n - \beta_{a,b,n},$$

while $a + b$ and b will be known privately to Alice and Bob, respectively. (Hence, in particular, the values mod n and div n are also computable privately by each.) The encrypted value of $\beta_{a,b,n}$ is obtained by running Type 3 protocol.

Conversion 5 (protocol of Type 2 from Type 3) [Veu14].

1. **Bob**:
   - Denote $a = n + (x - y)$ (this is unknown to Bob).
   - Randomly choose $b \in \mathbb{Z}_N$.
   - Set $y' = b \bmod n$.
   - Compute $E_{pk}(a + b) = E_{pk}(n) + E_{pk}(x) - E_{pk}(y) + E_{pk}(b)$ and send to Alice.
2. **Alice**:
   - Decrypt $E_{pk}(a + b)$ to $a + b$.
   - Set $x' = (a + b) \bmod n$.
   - Compute $E_{pk}((a + b) \operatorname{div} n)$ and send to Bob.
3. **Alice** and **Bob**
   - Together run protocol of Type 3 using inputs x' (from Alice) and y' (from Bob).

[2] That is, N is much larger than n. In particular, we will only need $4n < N$ to ensure that, in Conversion 5, we have $a < N/2$.

- As a result, Bob will obtain $E_{pk}(1-\beta_{a,b,n})$. (This holds by the correctness of the Type 3 protocol and the definition of $\beta_{a,b,n}, x', y'$.) This can be used to obtain $E_{pk}(\beta_{a,b,n})$.

4. **Bob**:

- Compute $E_{pk}(a \operatorname{div} n)$ as

$$E_{pk}((a+b) \operatorname{div} n) - E_{pk}(b \operatorname{div} n) - E_{pk}(\beta_{a,b,n}).$$

- From Lemma 3.2 and Eq.(3.4), we have $E_{pk}(a \operatorname{div} n) = E_{pk}(\delta)$.

The intuition is given above, while more details can be found in [Veu14, VBdHE15]. We state this result into the following lemma:

Lemma 3.3. *PN-NP-SE ⇒ SE-SE-SE.*

Chapter 4

De-identification Technique for Static Data

Structured static data is a basic data structure. Organizations have possessed static datasets so far to store a lot of information, such as customer information and inventory information. Customer information is personal data and invariant or almost unchanged information, such as gender, preference, and medical history, is included. Once this type of data is re-identified, the privacy breaches are serious and may be uncontrollable. Furthermore, if the data is sensitive information, the impact of the damage is incalculable. There are some models to evaluate the risk of privacy leakage [JOm, Rom16], and static and sensitive information tend to be estimated to have high value.

Combining data processing methods is one way to generate a high-quality de-identified dataset [Zay07]. However, few studies have evaluated such datasets because the adversary models and the situations differ. Several studies have evaluated the relationship between de-identification methods [WLK15, GLRV16], but mainly on the basis of differential privacy. Differential privacy requires noise that protects privacy against a powerful attacker who has knowledge of everything but the desired information. As a result, the individual information in the resulting de-identified data is heavily distorted. This model is similar to the malicious model and is reasonable in situations where de-identified data is publicly available and in the hands of a malicious attacker. However, the ability of adversaries can be restricted in actual business situations. An de-identified dataset is distributed among organizations by secure channels, and only authorized personnel can access it in a secure room. Therefore, the attacker is limited to being an authorized person who cannot access other databases in the secure room. In this chapter, we focus on a typical business situation and consider a combination of generalization, perturbation, and sampling as an de-identification method and propose an analysis method that yields risk estimation in real settings with real adversaries. More precisely, we estimate the re-identification probability of each record. The privacy risk should be as clear as k-anonymization because the risk should be understood by the data subjects who are the people in the dataset. Therefore, we define the reciprocal of the number of candidates as the re-identification probability. We also define a utility metric based on machine learning and the utility of de-identified datasets in a practical-use case. We conduct experiments whose results suggest that a complex de-identification method is a practical and effective way to generate more useful de-identified datasets. Our proposal can adapt to various data formats such as vectors and matrices. We believe that an easy-to-understand privacy metric is a critical requirement to obtain consent to use data. Our metrics are intuitive and satisfy the requirement.

The remainder of this chapter is organized as follows: Section 4.1 defines the concept of privacy and typical usefulness indicators in this dissertation. Section 4.2 presents a simulation attack for risk assessment against structured static data. Section 4.3 presents the evaluation experiments using a sample dataset. Section 4.4 concludes this chapter.

4.1 Privacy and Utility Metrics

When we intend to publish a de-identified dataset, defining the adversary's knowledge is difficult. When publishing trajectory data, for example, self-disclosures are possible by SNSs and it is becoming increasingly more complicated to assume an awareness of the adversary's knowledge. Therefore, we need to define how powerful the adversaries are. If the definition is not sufficient, the privacy may leak. Massachusetts governor William Weld is a typical example of re-identification by an unexpected attacker [Swe02b], and recently, an attack on the Netflix Prize dataset by an attacker who gained access to the Internet Movie Database is reported [NS06].

Differential privacy [Dwo06], which was defined in 2006, ensures privacy of each person mathematically. However, a de-identified dataset that provides protection by such mechanisms may not retain utility. In actual use, a previous experiment [EPK14] implies that when N data are de-identified by differential privacy, the volume of information that the data have is as same as that of raw $\sqrt{N}/10$ data, and such techniques may not be appropriate for small-scale enterprises. Therefore, to preserve utility, we need to define adversary models and evaluate their risks in relation to each situation. We must design a use case of a de-identified dataset and protect its privacy from both the technical and operational sides. As a typical business scenario, consider the following situation. Let S and C be organizations. S has a large-scale personal dataset, and C has a customer's dataset. S and C conclude a contract, and S de-identifies the dataset to meet both C's requirements and to provide a certain level of privacy. Finally, S sends a de-identified dataset to C and a person from the data analysis department of C analyzes it using machine learning (Fig. 4.1).

For example, if C is an automobile dealer that wants to advertise luxury cars to wealthy buyers, one requirement is that the income attribute in the dataset is not processed excessively. Such typical business situations as the one described above limit attacker knowledge by such legal restrictions as the General Data Protection Regulation (GDPR). For example, authorized personnel who deals with a de-identified dataset are prohibited from accessing other databases or using re-identification algorithms by conducting a privacy impact assessment (PIA) [ISO17] beforehand. In practice, the following operations may be taken. First, the data processor considers how to de-identify each attribute in the data set containing privacy information, and

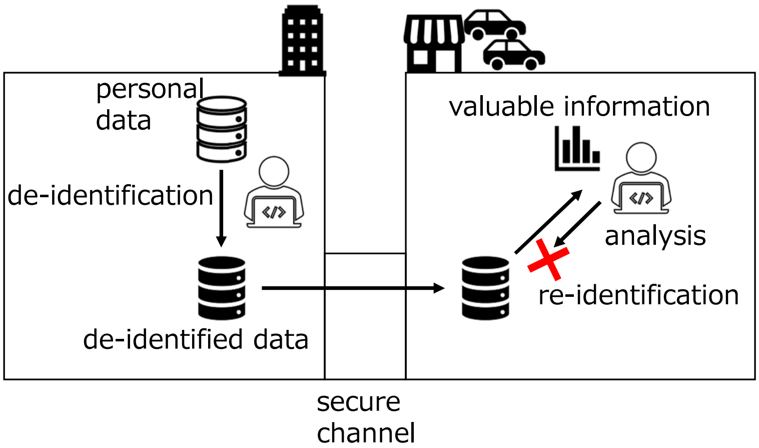


Figure 4.1: Typical use case of a de-identified dataset.

actually evaluates the sample data. If it is determined that the proposed de-identification method is sufficient to protect privacy, the method is implemented on the actual data. In such a situation, one of the most conceivable attacks is re-identifying or getting additional information of an acquaintance of the personnel of C. We define adversary models and privacy metric under such a situation in Section 4.1.1. One thing to keep in mind is that we are dealing with a dataset that can be processed manually. In reality, there are datasets with thousands of attributes, such as genome information or compound information, which cannot be handled by our de-identification method due to the curse of dimensionality [Agg05]. For such data, it is considered necessary to extract only the truly necessary information before responding to it, or to respond using a different de-identification method.

4.1.1 Privacy metric

First, and most importantly, we define "privacy" as the inability to know that some data belongs to a particular individual, regardless of the type or format of the data. This philosophy is a generalization of the most widely known k-anonymity concept.

We define two adversary models: a re-identifying adversary model and a revealing adversary model. We assume an attacker has a record that is his target and a de-identified dataset. Furthermore, an attacker generates de-identification simulator f_{sim} and tries to re-identify the target or uncover its sensitive information of the target from the de-identified dataset using f_{sim}.

Definition 4.1 (attacker's knowledge)**.** *An attacker has quasi-identifiers of original record* $r_i^0[QI]$*, de-identified dataset* D^1*, and de-identification simulator* f_{sim}*.*

Definition 4.2 (de-identification simulator f_{sim})**.** *Let* D^0 *be a dataset, let* D^1 *be a de-identified* D^0*, and let* $r_i^x[QI], r_i^x[SI]$ *be a set of quasi-identifiers or sensitive information of* r_i^x*, which is the i-th record of* D^x*. For input* $r_i^0[QI]$ *and* D^1*, de-identification simulator* $f_{sim}(r_i^0, D^1)$ *outputs* C_i*, which is the set of candidates for* r_i^0*.*

In this chapter, we address a typical business situation like Fig. 4.1 and assume that an attacker, who is in charge of a data analysis department, is restricted from accessing datasets except de-identified ones. Thus, we assume that an attacker can only access a de-identified dataset and the information of acquaintances and simulator f_{sim}, generated by checking the de-identified dataset. For example, if an attacker has acquaintance record $p = \{baseball\}$ and de-identified records $q_1 = \{sports\}, q_2 = \{ballgames\}, q_3 = \{indoorsports\}$, he generates a simulator so that $f_{sim}(p, D^1) = C_p = \{q_1, q_2\}$. The input candidates vary depending on the de-identification simulators and we explain some details of simulator f_{sim} in the following section. An attacker has one record as a target in our model, although we need to evaluate all the records of a dataset. Therefore, we define a de-anonymizer that simulates all attackers below.

Definition 4.3 (de-anonymizer for de-identified datasets, DA). *Given D^0 with n_0 records, D^1, and f_{sim}, the de-anonymizer DA runs f_{sim} against for $\forall r_i^0$ and outputs $C = \{C_1, \cdots, C_{n_0}\}$. Here, C_i represents the set of r_j^1 that are candidates for r_i^0.*

The attack on r_i^0 fails when $r_i^1 \notin C_i$, so the size of C_i can be ignored that does not include r_i^1. We define the privacy metric as Definitions 4.4 and 4.5.

Definition 4.4 (privacy metric against re-identifying adversary). *A re-identified adversary is assumed to have D_0, D_1, f_{sim}, and DA. This adversary executes DA and obtains C_i. Let n_q be the minimum number of $C_i[QI]$ containing $r_i^1[QI]$, then the privacy risk to the re-identified adversary is defined to be $1/n_q$.*

Definition 4.5 (privacy metric against revealing adversary). *The revealing adversary is assumed to have D_0, D_1, f_{sim}, and DA. This adversary executes DA and obtains C_i. Let n_s be the minimum number of types of $C_i[SI]$ containing $r_i^1[SI]$, then the privacy risk to the revealing adversary is defined to be $1/n_s$.*

The re-identifying adversary is an adversary model against quasi-identifiers and the revealing adversary is an adversary model against sensitive information. When a dataset has k-anonymity or l-diversity, and f_{sim} is the de-identification method, the risk denoted by the re-identifying adversary is $1/k$ and the risk denoted by the revealing adversary is $1/l$. This means our privacy metric does not diverge from conventional metrics. In this chapter, we address the re-identifying adversary model. Note that, if we assume that the minimum number of $r_j^1[SI] \in C_i[SI]$ s.t. $r_i^0[SI] = r_j^1[SI]$ is n_s, and the privacy risk for revealing adversary is the minimum value of $n_s/|C_i[SI]|$, then the suspicion of the sensitive information of r_i^0 can be expressed probabilistically. For example, if $r_i^0[SI] = X$ and $C_i[SI] = \{X, X, X, Y, Z\}$, then by Definition 4.5 the privacy risk to the revealing adversary of r_i^0 is 1/3, but it can also be taken as 3/5. It is necessary to think appropriately according to actual data and use cases.

4.1.2 Utility metric

We define the utility metric here. When datasets are de-identified, the volume of information and utility generally decreases. Most of the previous studies have been based on the distance between the original and de-identified datasets or the amount of information loss as a measure of utility [XWP+06a, XWP+06b, BKBL07a]. However, these metrics do not necessarily match the actual utility because the utility depends on the situation (i.e., context, use-case) and there are no general utility metrics. We consider use-case scenarios for de-identified data, such as the one we are interested in here, and propose a utility definition for the scenarios. Specifically, we consider a case in which a de-identified dataset is used as training data for a machine learning algorithm, and define the utility of the de-identified dataset as F-measure in the learning model.

Definition 4.6 (utility metric)**.** *Let $F(D, E)$ be the F-measure of a machine learning model, where the training data is D and the test data is E. The utility metric of a de-identified dataset $A(D)$ is defined as follows:*

$$Uti(A(D)) = \frac{F(A(D), E)}{F(D, E)}, \tag{4.1}$$

where $A(\cdot)$ is a de-identification algorithm.

The F-measure $F(D, E) \in [0, 1]$, along with AUC, is a widely adopted evaluation metric for machine learning algorithms, and is calculated using precision P and recall R. The accuracy P is obtained by dividing the number of true positives TP by the total number of true positives and false positives FP. That is,

$$P = \frac{TP}{TP + FP}. \tag{4.2}$$

This represents the percentage of correct predictions out of all predictions. Recall R is obtained by dividing the number of true positives by the total number of true positives and false negatives FN. That is,

$$R = \frac{TP}{TP + FN}. \tag{4.3}$$

This is the percentage of actual positives that were correctly predicted to be positive. The F-measure $F(D, E)$ represents the harmonic mean of the precision P and the reproduction rate R.

$$F(M, E) = \frac{2 \cdot R \cdot P}{R + P}. \tag{4.4}$$

We show the overview of the utility evaluation in Fig. 4.2.

We define a utility metric using F-measure for machine learning models that are mainly used for classification problems. In a similar manner, we can consider the utility of de-identified data as the ratio of typical evaluation values such as mean square error or logarithm loss, depending on the use case.

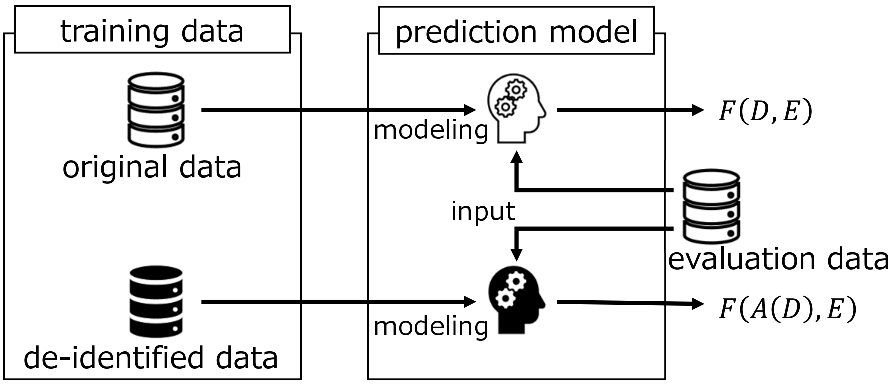


Figure 4.2: Overview of the utility evaluation.

4.2 De-identification Simulator

4.2.1 Overview of an de-identification simulator

When the original dataset is handled properly, as shown in Fig. 4.1, the malicious actions of attackers can be restricted. In such a situation, one of the most likely attacks is obtaining information about an attacker's acquaintance. Therefore, we define privacy risk as the maximum probability of re-identification or revealing each original record, as defined in Definitions 4.4 and 4.5. Because we cannot infer which record is the attacker's acquaintance, we need to evaluate all the records of D^0 and define the worst case as the privacy risk. An overview of our risk evaluation method is shown in Fig. 4.3.

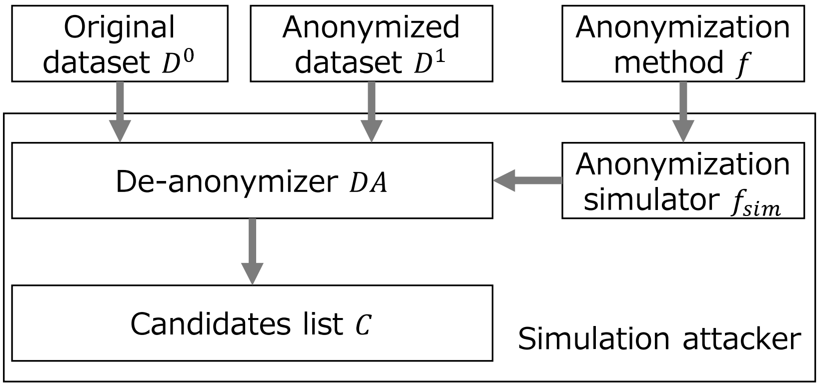


Figure 4.3: Overview of the simulation attack.

First, we input original dataset D^0 and de-identification method f. As a de-identification method, we can use generalizations, including top/bottom coding, noise addition, sampling, and combinations of them. Second, an attack simulator generates de-identification simulator f_{sim}, runs de-anonymizer DA, and obtains candidate set C. Finally, privacy risk parameter n_q or n_s is calculated from C.

The pseudocode of an attack simulator against the re-identification attacker is described below. The output is the only difference between the simulator against the re-identification adversary and that against the revealing adversary. The simulator against the revealing adversary calculates $n_s \leftarrow \min(|C_1[SI]|, \cdots, |C_{n_0}[SI]|)$ and outputs $1/n_s$.

Algorithm 2 $(D^0, D^1, f_{sim}, T, k, \alpha)$: The privacy risk evaluation algorithm

```
Input: Original dataset $D^0$, de-identified dataset $D^1$, and attack simulator $f_{sim}$
while $\forall r_i^0 \in D^0$ do
    $C_i \leftarrow f_{sim}(r_i^0, D^1)$
    if $r_i^1 \notin C_i$ then
        $1/|C_i| \leftarrow 0$
    end if
    if $T' < |(C_i^H \cup C_i^L) \cap D^1|$ then
        $|C_i[QI]| = |C_i[QI]|/\alpha$
    end if
end while
$n_q \leftarrow \min(|C_1[QI]|, ..., |C_{n_0}[QI]|)$
return $1/n_q$
```

In some countries, such as Japan, publishing de-identification methods is illegal. Therefore, an actual attacker does not know de-identification rules f and generates f_{sim} by inferring from a de-identified dataset. But in our simulation, we assume an attacker can simulate f for two reasons. First, as stated by Kerckhoffs' principle, even if de-identification methods are leaked, privacy must be preserved. This concept is also adopted in existing studies such as [DFM16]. Second, when a simple de-identification method is used, an attacker can infer the method. The idea stems from a security proof in cryptology where an attacker is assumed to be able to simulate a de-identification method.

We provide attack simulators for each de-identification. In our model, an attack simulator replicates the actual de-identification methods, although some methods are probabilistic. Thus, we introduce the confidence of an attack. When record r_i^1 is linked to r_j^0 with high probability, the confidence of an attack on the record is either "high" or "low". When the confidence of an attack against an actual target is high, we label all the high confidence records as target candidates. On the other hand, we define both high and low confidence records as target candidates when the confidence of an attack against an actual target is low. For example, we assume an attacker has r_i^0 whose candidates are r_1^1, r_2^1, and r_3^1. Let the confidence of r_1^1 and r_2^1 be high, and that of r_3^1 be low. If r_1^1 or r_2^1 is actual de-identified r_i^0, a simulation attacker regards $C_i^H = \{r_1^1, r_2^1\}$ as high confidence candidates and $C_i^L = \{r_3^1\}$ as a low confidence candidate. Hence, when an actual target is in C_i^H, $f_{sim}(r_i^0, D^1) = C_i = C_i^H = \{r_1^1, r_2^1\}$, and the privacy risk of r_i^0 is $1/2$. On the other hand, when an actual target is in C_i^L, $f_{sim}(r_i^0, D^1) = C_i = C_i^H \cup C_i^L = \{r_1^1, r_2^1, r_3^1\}$, and the privacy risk of r_i^0 is $1/3$. If an actual target is not included in either C_i^H and C_i^L, the privacy risk of r_i^0 is 0. As mentioned above, understanding privacy risks should be easy for data subjects, and this definition simplifies the evaluation of them. This is why we introduce the confidence of an attack. Moreover, all a simulated attacker needs to do is to count the number of candidates, which accelerate simulation attacks.

4.2.2 Simple de-identification methods

In this subsection, we present attack simulators for generalization, perturbation, and sampling and propose to combine them. The combined attack simulators are easily constructed under our adversary models.

Generalization is a de-identification technique that reduces the risk of re-identification by abstracting values. We assume hierarchical tree structures for each attribute, and top/bottom coding and suppression are regarded as special cases of generalization. For example, suppression can be achieved by setting all the attributes to the highest levels in the trees. Generalization using tree structures is a basic concept and several studies on de-identification have been conducted using them.

Incognito [LDR05] is a well-known k-anonymization algorithm using tree structures. An attacker can infer tree structures because she can check a de-identified dataset and the tree structures are usually constructed based on common sense. For a transportation attribute, for example, it is easy to say that $r^0[qi_i] = \{train\}$ is generalized to $r^1[qi_i] = \{public\, transport\}$. We argue that the confidence of the de-identified records is high since all of their attributes belong to their ancestor nodes of the target. More precisely, let $A(r_i[qi_j])$ be the set of ancestors of $r_i[qi_j]$, including $r_i[qi_j]$. Generalization simulator $f_{gen}(r^0_i, D^1[qi_j])$ outputs a set of records $r_{\hat{i}}$, where $\forall j, r_{\hat{i}}[qi_j] \in A(r^0_i[qi_j])$ as high confidence candidates C^H_i. A generalization algorithm is deterministic with no probability to be a candidate for other records, and so $f_{gen}(r^0_i, D^1[qi_j])$ outputs no records as C^L_i. Finally, attack simulator $f_{sim}(r^0_i, D^1)$ outputs $C_i = C^H_i \cap D^1$ because generalization is a deterministic algorithm and the actual target is in C^H_i.

Noise addition, microaggregation, and data swapping are widely known techniques that are collectively referred to as perturbation. Noise addition is a simple de-identification technique where such noise, as Gaussian or Laplacian noise, is added to each cell. This technique is mainly applied to quantitative attributes, although some research has addressed noise addition for categorical data [MT07]. Noise addition for categorical data changes the data to another type of categorical data with a certain probability, and such data swapping resembles a kind of noise addition for categorical data.

In our model, an attacker has a de-identified dataset and can check the distribution of each bit of data. Therefore, perhaps the attacker can infer the type and parameter of the noise and simulate an de-identification technique. Let $N(r^0_i[qi_j])$ be the noise added to $r^0_i[qi_j]$, and let $F(r^1_{\hat{i}}[qi_j]) = Pr[N(r^0_i[qi_j])] \leq |r^1_{\hat{i}}[qi_j] - r^0_i[qi_j]|$ be the cumulative distribution function of the noise. We believe that the confidence of the de-identified records is high when the added noise is less than parameter T. Perturbation simulator $f_{per}(r^0_i[qi_j], D^1)$ sets $r^1_{\hat{i}}$ s.t. $T \leq \prod_j F(r^1_{\hat{i}}[qi_j])$ as high confidence candidates C^H_i. On the other hand, $r^1_{\hat{i}}$ s.t. $T > \prod_j F(r^1_{\hat{i}}[qi_j])$ are regarded as low confidence candidates C^L_i. Here T is a threshold, and we fix $T = 0.9$ in the following experiments. Finally, attack simulator $f_{sim}(r^0_i, D^1)$ outputs $C_i = C^H_i \cap D^1$ if an actual target is in C^H_i and $C_i = (C^H_i \cup C^L_i) \cap D^1$ when the target is in C^L_i. When this algorithm is actually performed, the simulator records the upper and lower limits and a de-identified record included in the range is classified in C^H_i. Note that the number of combinations of the range of each attribute increases as the number of attributions increases, and it causes reducing the processing speed.

One thing to note is that the impact of data swapping on re-identification needs to be considered separately. For example, suppose a dataset D with 10 attributes, r_1^0 with ID 1, and r_2^0 with ID 2. Suppose that one attribute of ID 1 and ID 2 are exchanged by data swapping. In this case, an attacker with the information of ID 1 can follow the data swapping algorithm and may list the candidate records. However, suppose that all the attributes of ID 1 and ID 2 are exchanged by data swapping. In this case, the de-identified ID 1 record is the same as the original ID 2 record, namely $r_1^1 = r_2^0$, and if the data swapping algorithm is to exchange records randomly, the privacy risk against re-identification adversary will be virtually eliminated. However, the record of each ID itself will remain in its original form.

Furthermore, if the five attributes of ID 1 and ID 2 are exchanged, the privacy information of ID 1 will exist across r_1^1 and r_2^1, and it is difficult to evaluate privacy risk with re-identification probability. Therefore, when we deal with excessive data swapping, we should assume a revealing adversary. The revealing adversary model assumes that each record has sensitive information. For example, if all the attributes except for the sensitive information of ID 1 and ID 2 are exchanged, the revealing adversary with ID 1 will try to obtain the sensitive information of ID 1. If the sensitive information of ID 1 and ID 2 are different, the privacy of the data subject represented by ID 1 is protected. However, for such de-identified datasets, the connection between the quasi-identifier and the sensitive information is weak, and the accuracy of the analysis of the sensitive information is low. On the other hand, if the sensitive information of ID 1 and ID 2 are the same, the privacy of the data subject represented by ID 1 will be exposed with high probability.

Sampling is a widely used technique to preserve privacy in the statistics field. A sampled dataset preserves privacy to a greater extent than a non-sampled dataset because the unique records included in the former may not be unique in the original dataset. Therefore, an attacker cannot completely link the records to the target even if there are unique records in a de-identified dataset. In the statistics field, the relationship between an original dataset and a sampled dataset has been researched for a long time. Greenberg et al. [GZ92] proposed the probability model of re-identification that considered the probability that a unique record is also unique in the population. However, this may be not realistic because it assumes the existence of prior information about cell frequencies in the population.

We focus on outliers and argue that since they are risks to the dataset during sampling, we need to actually evaluate their risk. To put it simply, privacy risks are reduced when sampling rate α is lowered because an attacker ’ s confidence is reduced. However, if an outlier is sampled, an attacker who knows that the outlier is in the dataset suspects the data are sampled and the record ’ s privacy risk is not reduced. Therefore, we construct the following attack simulator of sampling. First, an attacker who has record r_i^0 judges whether it is an outlier or not. The only information the attacker can use is the original record and a de-identified dataset so that an attack simulator judges whether the original record is an outlier by counting candidates $(C_i^H \cup C_i^L) \cap D^1$. If the number of the candidates is less than parameter T', the record is regarded as an outlier. Here T' is a threshold as well as T, and we fix $T' = k \times \alpha$, where k is a fixed number and α is the sampling rate. This means that a record is not regarded as an outlier if there are k records near it before the sampling was applied. When record r_i^0 is not regarded as an outlier, the privacy risk decreases according to α.

4.2.3 Combination of de-identification methods

Although such de-identification methods as those denoted above can be combined easily, few studies have evaluated the privacy risk when they are combined. The privacy risk of combined de-identified data can be evaluated using our adversary models and simulators. The idea is simple, and the privacy risk can be calculated by combining the simulators. Each de-identification simulator outputs records that are given a confidentiality level. We define the confidence of the attack as high if all simulators have high confidence for a given record.

Example: Assume a dataset with two quasi-identifiers, qi_1 and qi_2. A data processor adds noise to qi_1 and generalizes the noise added to qi_i based on a hierarchical tree. The data processor generalizes and adds noise to qi_2 and performs simple 80% random sampling for de-identification. Here let $f_{per}^{qi_1}$ be a function that adds noise to qi_1 where the size of the noise is 0 or ±1 with 35% probability and ±2 to 4 with 10% probability, and let $f_{per}^{qi_2}$ be a function that swaps the categories of $f_{gen}(qi_2)$ with 20% probability. Let f_{gen} be a function that generalizes one level based on a hierarchical tree and let $f_{sam}^{D^0}$ be a simple 80% random sampling function. Tables 4.1 and 4.2 show both the original and de-identified datasets and Table 4.3 shows the tree structures of qi_1 and qi_2. Then the data processor runs attack simulators for each record r_i^0. The simulators replicate $f_{sim}(r_i^0[qi_1, qi_2], D^1)$, which we denote as $f_{sam}^{D^0}(f_{gen} \circ f_{per}^{qi_1}(r_i^0[qi_1], D^1[qi_1]) \cap f_{per}^{qi_2} \circ f_{gen}(r_i^0[qi_2], D^1[qi_2]))$, and output set of candidates C_i^H and C_i^L. Here, we consider a simulator that has record r_1^0. For qi_1, the simulator knows $r_1^0[qi_1] = 1$ and guesses that the data may be 1 or 2 with 70% probability and 3 or 4 with 20%. The simulator also knows f_{gen} and believes $r_1^1[qi_i] \in \{[1, 2], [3, 4]\}$ with 90%. For qi_2, the simulator knows $r_1^0[qi_2]$ becomes "sports" by generalization and guesses that $r_1^1[qi_i] = \{Culture\}$ with 20%. The simulator calculates the cumulative probability that r_1^1 can be and puts the records into C_1^H when the probability exceeds threshold T. For example, the simulator includes r_1^1 in C_1^L because the probability of $r_1^1[qi_1] \leq 4$ is 90% and the probability of $r_1^1[qi_2] \neq \{Sports\}$ is 20%, and so that the probability that r_1^1 becomes r is 18%. We illustrate an example of candidates of combination of generalization and noise addition in Fig. 4.4. The right figure represents candidates when a data is generalized and added noise, and the other represents candidates when a data is added noise and generalized. The red dots denote original data, the blue dots denote the generalized candidates, and the yellow dots denote the noise added candidates. In this example, the simulator outputs $C_1^H = \{r_2^1, r_4^1\}$ and $C_1^L = \{r_1^1, r_7^1\}$ and checks whether r_1^0 is an outlier. When the parameter is $k = 2$, for example, $T' = 2 \times 0.8 = 1.6$ and $|(C_1^H \cup C_1^L) \cap D^1| \leq T'$. Therefore, the simulator considers the risk to be low that r_1^0 is unique in a population. Finally, the simulator check where the actual r_1^1 is. In this example, $r_1^1 \in C_1^L$ and the simulator outputs $C_1 = C_1^H \cup C_1^L$. If $C_1 = \min\{C_1, \cdots, C_n\}$, the data processor returns

$1/n_q = 1/(|C_1|/0.8) = 1/5$ as the privacy risk against re-identifying adversary of the dataset. The privacy risk of r_i^0 can be explained intuitively. When $r_i^0 \in C_i^H$ and $C_i = C_i^H$, there are at least $|C_i^H|$ records where r_i^0 is a process to with more than probability T. On the other hand, when $r_i^0 \in C_i^L$ and $C_i = C_i^H \cup C_i^L$, there are at least $|C_i^H \cup C_i^L|$ records where r_i^0 can happen by de-identification.

Table 4.1: Original dataset.

record	qi_1	qi_2
r_1^0	1	Football
r_2^0	3	Baseball
r_3^0	4	Swimming
r_4^0	1	Basketball
r_5^0	7	Art
r_6^0	7	Chorus
r_7^0	8	Brass band

Table 4.2: De-identified dataset.

record	qi_1	qi_2
r_1^0	$[3, 4]$	Culture
r_2^0	$[1, 2]$	Sports
r_4^0	$[1, 2]$	Sports
r_5^0	$[9, 10]$	Culture
r_7^0	$[5, 6]$	Sports

Table 4.3: Tree structures of qi_1 and qi_2.

Level 1	Level 2	Level 3
$\{[1,2],[3,4],[5,6],[7,8],[9,10]\}$	$\{[1,4],[5,6],[7,10]\}$	*
$\{Outdoor, Indoor, Music, Others\}$	$\{Sports, Culture\}$	*

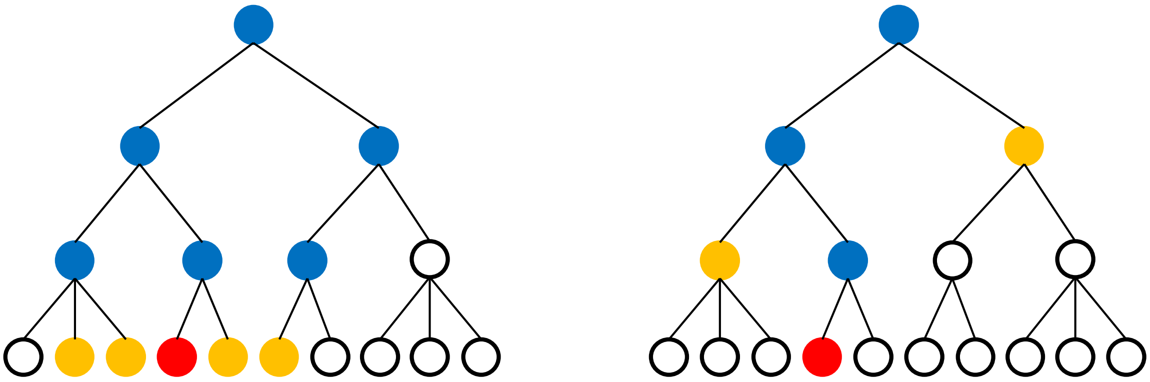


Figure 4.4: Candidates of combination of generalization and noise addition.

4.3 Experiment

4.3.1 Dataset

We anticipate that a combined de-identification method can retain greater utility than a simple one when the privacy risks are identical. We conducted experiments to confirm our expectation and de-identified a dataset using several methods; generalization using tree structures, noise addition to numerical data, and random sampling.

We used the $32,561$ records from the UCI Adult dataset[1] for the following experiments. We assigned age, hours-per-week, workclass, and marital-status as quasi-identifiers QI and income as sensitive information SI. For generalization, we constructed tree structures (Table 4.4) for each attribute and denoted a de-identification function as $f_x(r[qi_i]) = f_x(r_i^0[qi_i], D^1)$ when we used a de-identification technique x to qi_i.

Table 4.4: Relationship among noise, sampling, and risk.

Attribute	Level 1	Level 2	Level 3
age	$\{[27], [28, 35], [36, 47], [48\}$	$\{35], [36\}$	*
workclass	$\{government, private, other\}$	*	*
martial-status	$\{married, notmarried\}$	*	*
education	$\{[8], [9, 9], [10, 10], [11, 12], [13\}$	*	*

We also denoted a combined de-identification function as $f_x^{par_x}(r[qi_i]) \circ f_y^{par_y}(r[qi_j])$ when we used de-identification techniques x with parameter par_x and y with parameter par_y to qi_i and qi_j. For example, when we added $\epsilon = 1.0$ noise to age and generalized age and $workclass$ to level 1 and performed 90% sampling, we denoted the de-identification functions as $f_{sam}^{\alpha=0.9}(f_{per}^{\epsilon=1}(r[age]) \circ f_{gen}^{l=1}(r[age, workclass]))$. We applied random sampling and Laplace noise for sampling and noise addition to the numerical data in the algorithm. The sampling rate was α, and noise was generated with distribution $Lap(\delta_j/\epsilon)$, where δ_j is the minimum unit of qi_j. We also applied exponential mechanism [MT07] for noise addition toward the categorical data. We defined the distance between the categorical data based on the number of edges in the hierarchical tree, and the exponential mechanism selects $r^1[qi_j]$ based on probability $Pr[r^1[qi_j]] \propto e^{\epsilon H(D^0, r^1[qi_j])/2}$, where $H(D^0, r^1[qi_j])$ is the distance between $r^1[qi_j]$ and the median in $D^0[qi_j]$. Note that, this noise mechanism achieves differential privacy only when the query toward the dataset is concerning with a person.

4.3.2 Privacy risk evaluation

We evaluate the effects of each de-identification technique and confirm whether our privacy metric is valid. We obtained a 4-anonymized dataset (dataset 1) by generalizing each attribute to level 1, $\forall i, f_{gen}^{l=1}(r[qi_i])$. Our privacy metric is identical to k-anonymity when the dataset is de-identified by k-anonymization, and so its privacy risk is $1/n_q = 1/4$. We next generated two de-identified datasets. We added noise to age and education, generalized workclass and martial-status, and performed sampling (dataset 2). In the other dataset, we generalized age and education, added noise to education, workclass and martialstatus, and performed sampling (dataset 3). Each de-identification simulator f_{sim} is denoted as $f_{sam}^{\alpha}(f_{per}^{\epsilon}(r[age, education]) \circ f_{gen}^{l=1}(r[workclass, martialstatus]))$ and $f_{sam}^{\alpha}(f_{gen}^{l=1}(r[age, education]) \circ f_{per}^{\epsilon}(r[education, workclass, martialstatus], D^1))$. We evaluated the privacy risk of datasets 2 and 3 ten times and took the average scores because sampling and noise addition are probabilistic algorithms, and

[1] https://archive.ics.uci.edu/ml/datasets/adult

our privacy risk is affected by outliers. The risks of combining de-identification techniques are shown in Tables 4.5 and 4.6. The privacy risk decreases as the noise increases, showing that our privacy metric is intuitively correct. On the other hand, the sampling rate only slightly affects the privacy risk. This is because, the probability of all the outliers is not sampled is $(1-\alpha)^l$, where l is the number of outliers in a dataset, and the probability is not very high generally. Applying only simple sampling may be inefficient when there are many outliers. Our algorithm evaluates a dataset ’ s risk as high when an outlier remains in it. However, when the sampling rate is reduced, it is difficult for an attack simulator to judge whether a record is an outlier. Therefore, even if an outlier remains, the privacy risk is reduced. In our model, even though an attacker has an original record, and the privacy risk is subject to attackers who have an outlier. Attackers struggle to infer the target when noise is added, sampling is conducted, and the target is excluded. We generalized some quasi-identifiers and added noise to the other quasi-identifiers to generate dataset 2. Its result shows that combining noise addition and sampling can achieve the same privacy level as the k-anonymized dataset. We added noise to the categorical data and generalized the numerical quasi-identifiers to generate dataset 3. Its result shows that combining generalization, noise addition, and sampling achieved a higher privacy level than the k-anonymized dataset. This is because the candidates of a record increase explosively when categorical data are added to the noise. This result shows that privacy improves even if the noise is very small and the sampling rate is very high.

Table 4.5: Privacy risk n_q of dataset 2.

Privacy risk(n_q)		Noise parameter(ϵ)					
		1.0	0.9	0.8	0.7	0.6	0.5
Sampling rate (α)	0.9	1.8	2.2	2.5	2.4	3.1	3.0
	0.8	1.5	1.8	2.2	2.6	2.9	3.2
	0.7	1.9	2.1	2.1	2.6	2.8	3.1
	0.6	2.2	2.4	3.0	3.1	3.4	3.4
	0.5	2.8	2.6	3.2	3.8	3.6	3.8

Table 4.6: Privacy risk n_q of dataset 3.

Privacy risk(n_q)		Noise parameter(ϵ)					
		1.0	0.9	0.8	0.7	0.6	0.5
Sampling rate (α)	0.9	6.8	7.6	6.7	8.1	8.6	9.8
	0.8	7.2	7.5	7.9	9.6	12.1	14.1
	0.7	7.7	8.0	8.5	9.6	14.1	16.9
	0.6	8.3	8.6	9.4	11.3	16.8	18.7
	0.5	8.1	9.2	9.6	14.1	18.3	19.3

4.3.3 Utility evaluation

The main aim of our proposal is combining de-identification methods and improving the utility of de-identified datasets. We adopted the privacy metric and the utility metric defined in Sec 4.1.

We evaluated the utility of some de-identified datasets, which have identical privacy levels, and confirmed that a dataset processed by combined de-identification methods is more useful than a dataset processed by a single de-identification method. In our use case (Fig. 4.1), organization C receives a de-identified dataset from S and delivers advertisements to its customers. More precisely, C first performs a machine learning algorithm for a de-identified dataset and constructs a learning model. Then it inputs its customer's data into the model obtains prediction results, and finally provides improved services based on those prediction results. In this experiment, we employed SVM (RBF kernel) for the machine learning model.

We generated several datasets in our previous experiment denoted in Section 4.3.2 by changing the noise and sampling rate parameters, some of which have almost identical privacy levels as dataset 1. Moreover, we generated a dataset with 22-anonymity by generalizing age to level 2 and the other attributes to level 1 (dataset 4) and a dataset with 10-anonymity by removing records from dataset 1 (dataset 5). We randomly chose 10% of the records from the original dataset as test data and the remainder were used as training data. The utility of each dataset is shown in Table 4.7. The combining de-identification methods is efficient based on the result; the performance of dataset 3 ($\epsilon = 1.0, \alpha = 0.5$) is the highest. The utilities of datasets 1 and 5 are almost the same and suggest that outliers barely affect utility, which we defined above. Similarly, when we focus on dataset 3 ($\epsilon = 0.5, \alpha = 0.9$) and dataset 3 ($\epsilon = 0.5, \alpha = 0.5$), the sampling effect is less than the noise addition. Focusing on the datasets 1 and 4, generalization improves the privacy, although the utility is reduced drastically. Our experiment result shows that combining de-identification methods can be handled flexibly to achieve privacy as effectively as k-anonymization.

Table 4.7: Utility evaluation.

Privacy risk (n_q)	Dataset	$Uti(D^1)$
4	Dataset 1	0.658
	Dataset 2 ($\epsilon = 0.5, \alpha = 0.5$)	0.633
10	Dataset 5	0.647
	Dataset 3 ($\epsilon = 0.5, \alpha = 0.9$)	0.636
	Dataset 3 ($\epsilon = 1.0, \alpha = 0.5$)	0.695
22	Dataset 4	0.588
	Dataset 3 ($\epsilon = 0.5, \alpha = 0.5$)	0.650

4.4 Conclusion

We assumed an actual use case of a de-identified dataset and defined adversary models for such a situation. Since personal datasets are sometimes de-identified by combined de-identification methods, evaluating their privacy risks is difficult. Our adversary models enabled the evaluation of the risk of combined de-identified datasets. We also defined a utility metric for typical business situations and conducted two experiments using privacy and utility metrics. In the first, we evaluated the relationship among de-identification parameters and confirmed that generalization is an efficient method to preserve privacy; applying only small noise additions or simple sampling are less efficient on outliers. We conducted an additional experiment to confirm whether a complex de-identification is valid. Its results show that when de-identification processes are done properly, noise addition and sampling greatly protect privacy. In addition, noise addition and sampling can retain the overall features and datasets obtained using a complex de-identification method can preserve far greater utility than a simple method.

Chapter 5

De-identification Technique for Dynamic Data

We clarified that a combined pseudonymized dataset can preserve greater utility compared to a single pseudonymized dataset in Chapter 4. This chapter handles dynamic data. Dynamic data is fluid and usually includes the time axis. Daily electric power consumption, trajectory data, purchase history, and treatment progress data are categorized as dynamic data. Analyzing dynamic data is a social necessity, although data processors have to deal with more complex de-identification methods compared to static data. Dynamic personal data is essential for building an efficient and sustainable society, but it is also sensitive and must be handled with care. Time-sequence data, such as purchase and movement history, has attractive values at a macroscopic level. For example, vehicle trajectories can be used to find the cause of traffic jams, and purchase histories can be used to develop marketing strategies. In contrast to security, the key challenge of preserving privacy in personal data is that the attacker may be a legitimate user who is the recipient of the de-identified data. Therefore, it is important to maintain a balance between privacy and utility, and several techniques [Zhe15] have been studied to achieve this balance for dynamic data.

There are static and dynamic attributes in time sequence data. Static attributes are identifiers or quasi-identifiers such as name, age, and gender. Dynamic attributes are characteristic of time sequence data and include information about the time sequence. Examples of dynamic attributes are pseudonym ID, time stamp, location, direction of vehicle travel, engine speed, number of purchases, etc.

Dynamic attributes are particularly important and valuable, but there is a high risk of personal information leakage even if some of the information is generalized or deleted. In addition, prior research [Kru07] has shown that the amount of processing required to protect the privacy of time-sequence data is surprisingly large.

Most of the existing research [ABN08] believe that privacy is leaked when de-identified records are linked to the original records. However, especially in time-sequence data, there is also a privacy risk in linking pseudonymized records to each other. Therefore, in many cases, the primary key is changed every certain period of time to make one user's record appear to be another user's record. However, there is a tendency for people to behave in a similar manner. For example, there are people who follow the same trajectory at the same time, such as commuting to work, or people who buy their favorite products in the same way over time. Therefore, there is a risk of linking between records that represent the same user, even if the primary keys are different. Linking records may lead to the disclosure of some other information. We define linkage attack in Section 5.1 to address the privacy and evaluate the effect in Section 5.3.

Some time-sequence data such as location data can be denoted as a matrix and we consider a matrix data in this chapter. There is some research that manages a time-sequence data as a matrix [ZZXM09, ZZXY10, CZH12, Zhe15] and our main proposal is to maintain the utility and the privacy of a dataset using matrix operations.

Matrix factorization is a fundamental step in data analysis. In particular, matrix $M \in \mathbb{R}^{n\times m}$ is decomposed into $U \in \mathbb{R}^{r\times n}$ and $V \in \mathbb{R}^{r\times m}$. U and V represent properties of rows and columns respectively. The matrix $X = U^{\mathsf{T}}V$ is an approximation of M and rank r affects the accuracy. We observe in our later evaluations that low-rank matrix factorization is useful for de-identification, i.e., a low-rank matrix is more likely to withstand re-identification and linkage attacks. Since V, the feature matrix of items, does not carry any personal information, we propose to de-identify only U, the feature matrix of users, in order to maintain practicality. We evaluate the effect of our proposal method in an actual time-sequence dataset in Section 5.3.

In summary, there are two main contributions in this chapter. The first one is that we regard a matrix factorization technique as a de-identification method and evaluate its effectiveness on real data. Furthermore, we propose a de-identification algorithm that combines matrix factorization with other de-identification methods. We apply the algorithm to real data and evaluate its privacy risk and the utility. The strength of the proposed algorithm is that it can flexibly adjust the privacy risk by changing the rank r, compared to the conventional de-identification algorithms. Another contribution is to define the linkage attack, which is a privacy risk specific to time series data, and to evaluate its risk in real time-sequence data.

The remainder of this chapter is organized as follows: Section 5.1 introduces privacy and utility metrics. Section 5.2 presents the de-identification method based on matrix factorization. Section 5.3 summarizes the results of each experiment and shows the effectiveness of the proposed method. Section 5.4 concludes this chapter.

5.1 Privacy and Utility Metrics

5.1.1 Privacy metric

We define two types of attack models for time-sequence datasets. The first one, re-identification attack, is a general attack model in which the attacker has information about the original dataset M and tries to re-identify it in the de-identified dataset $A(M)$. This model assumes that an attacker has maximum information about the original dataset. However, on the other hand, as in Chapter 4, we assume that it is properly managed and that the attacker has limited external knowledge. This model is same as that of k-anonymization where even if an attacker has an original dataset, the probability of re-identification of a k-anonymized dataset is $1/k$. Here, we define two types of attacks.

Definition 5.1 (re-identification attack). *Let an attacker have matrix $M_{t_1} \in \mathbb{R}^{n \times m}$ and a de-identified matrix $A(M_{t_1}) \in \mathbb{R}^{n \times m}$. M_{t_1} represents a time-sequence data, which is observed during t_1, and n is the number of records and m is the number of items. A re-identification attack against a record r_i succeeds if record $r_i \in M_{t_1}$ is linked to record $r'_j \in A(M_{t_1})$, where r'_j is the de-identified r_i or belongs to the cluster which includes the de-identified r_i.*

The linkable attack, which is the attack of an authorized user, is that an attacker tries to obtain some information from the given datasets $A(M_{t_1})$ and $A(M_{t_2})$. $A(M_{t_1})$ and $A(M_{t_2})$ are assumed to be included the same users but the primary keys are different. In this model, the attacker only has a de-identified data set, which is assumed to be one of the more realistic attacks. There are few studies on this issue, and in this chapter we evaluate the danger using a real data set.

Definition 5.2 (linkable attack). *Let an attacker have two de-identified matrix $A(M_{t_1}) \in \mathbb{R}^{n \times m}$ and $A(M_{t_2}) \in \mathbb{R}^{n \times m}$. M_T represents a time-sequence data, which is observed during T, n is the number of records, m is the number of items and both M_{t_1} and M_{t_2} include the same users and items. A linkable attack against a record r_i succeeds if record $r'_i \in A(M_{t_1})$ is linked to record $r''_j \in A(M_{t_2})$, where r'_i and r''_j represent the same user or r''_j belongs to the cluster which includes the same user of r'_i.*

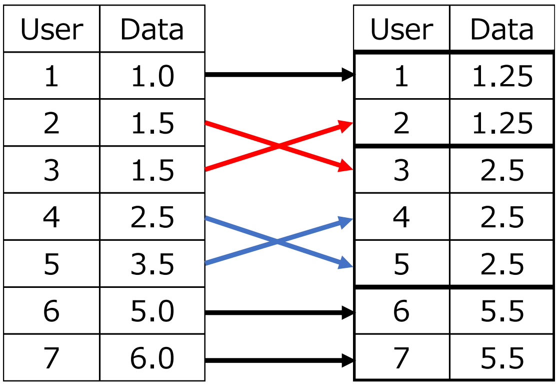
We next define privacy metric as follows:

Definition 5.3 (privacy metric against adversaries of dynamic data). *Let n be the total number of users of a dataset M and n' be the number of users which are attacked succesfully. The privacy risk of M is defined n'/n.*

This definition represents the same thing as definitions 4.4 and 4.5. Both of these definitions represent the probability that an individual will be re-identified. Def. 4.4 refers to re-identification attacks with quasi-identifiers, and Def. 4.5 refers to re-identification attacks that consider even sensitive information. We define here a more general privacy risk that takes into account linkable attacks.

On the other hand, we also need to consider the attack method: in Chapter 4, we assumed a situation where the attacker has the original data set, the de-identified data set, and the de-identification method. In this chapter, we assume a relatively simple attack method using existing matching algorithms. Although rigorous privacy evaluation of the de-identified data is difficult, it is sufficient to analyze the trend of the effectiveness of the proposed method. In addition, our method combines matrix factorization with other de-identification methods, and especially when combined with k-anonymization, the de-identified dataset has k-anonymity at least.

Example: We consider the attacks to be the same as those to solve an assignment problem. An assignment problem is to find the task assignment properly when there are n users and tasks and the Hungarian algorithm [Kuh55] solves the assignment problem in such a way that the entire cost is minimal. We apply the algorithm as re-identification and linkable attacks and consider when an attacker assigns the same user, the attack succeeds. Although the privacy evaluation in this attack is not rigorous, the experimental results are sufficient to see the effect our proposed algorithm produces. When a dataset is k-anonymized, there are at least $k-1$ identical records. Thus, if a record is assigned to a cluster to which the correct record belongs, it is considered correctly assigned, even if the assigned record is not actually correct. Furthermore, since probability is the proportion of clusters that are correctly assigned, we define privacy as the result of multiplying the probability by $1/k$. Figure 5.1 shows an example of risk evaluation. The left side is the original dataset and the right side is the de-identified dataset. Arrows indicate assignment results. For example, user 2 in the original dataset is assigned to user 3 in the de-identified dataset, so the attack on user 2 fails. When noise addition is used as the de-identification method, users 2, 3, 4, and 5 are assigned to the wrong users and the privacy risk is 3/7. On the other hand, with k-anonymization, in this case $k=2$, users 4 and 5 are assigned to the wrong user (blue arrows), but to the same cluster as the correct user. Therefore, we can assume that the attacks against users 4 and 5 were successful. The only unsuccessful attacks are against users 2 and 3 (red arrows), and the privacy risk is $5/7 \times 1/2 = 5/14$.



User	Data
1	1.0
2	1.5
3	1.5
4	2.5
5	3.5
6	5.0
7	6.0

User	Data
1	1.25
2	1.25
3	2.5
4	2.5
5	2.5
6	5.5
7	5.5

Figure 5.1: Example of an attack simulation using Hungarian algorithm.

5.1.2 Utility metric

As we mentioned in Section 4.1, the utility depends on the situation (i.e., context, use-case), and metrics based on either the distance or the amount of information loss do not necessarily match the actual utility.

We also adopt Definition 4.6 here and consider that de-identified data are used for machine learning model. Specifically, we consider a use-case in which a de-identified dataset is used as training data for a machine learning algorithm. In the case of a web access log dataset, for example, a client, who is a developer of an anti-virus software, may generate a machine learning model from a de-identified dataset and predict whether their user will access a phishing website.

We consider an actual use-case and generate machine learning model as opposed to such as [SB16]. We first generate two machine learning models; one is from an original dataset and the other is from its de-identified dataset. An item is randomly chosen as a objective variable and the remainders are explanation variables. And then, we use these models and predict an attribute of each record of an evaluation dataset that has the same attributes as the original dataset. This operation is performed several times while an objective variable is changed. The utility is defined as the average of the ratio of the F-measure of a model of the de-identified dataset to that of a model of the corresponding original dataset. In this chapter, we apply logistic regression as the machine learning algorithm and predict for fifty attributes.

5.2 De-identification based on Matrix Factorization

Matrix factorization is the technique of decomposing a single matrix into two by solving an optimization problem. The accuracy of matrix factorization depends on the rank r, and the smaller r is, the lower the accuracy becomes. This can be viewed as a de-identification technique with rank r as a parameter. In addition, since matrix factorization generates matrices, it can easily be combined with different de-identification methods. Therefore, we consider a combination of de-identification methods as in some existing studies [FLS+11, MKTM17]. Specifically, we combine matrix factorization with other de-identification methods *ano*. Although *ano* can be applied to any de-identification method for matrices, in particular, we consider k-anonymization and noise addition. We introduced several matrix factorization algorithms in Section 2.2, and our proposal can be implemented for any matrix factorization algorithm. We explain the case of simple SGD[1] technique here. The basis matrix U and the weighting matrix V can be assumed as row and column characteristics, respectively, and for the dataset we are dealing with, U can be viewed as the user characteristics matrix. Therefore, we propose to de-identify only U and keep V, thus de-identifying only the user's features and preserving the characteristics of the domain.

In this algorithm, the dataset M is first divided into U and V, and U is de-identified. Then, V is optimized once and recombined with the de-identified U, as in the usual matrix factorization procedure. $A_{ano}(M)$ represents the application of a de-identification algorithm to the matrix M, and $A_{(ano,r)}(M)$ combines matrix factorization of rank r with the de-identification algorithm *ano* and is denoted as follows.

$$A_{(ano,r)}(M) = (A_{(ano)}(U))^{\mathsf{T}} V, \text{ where } U \in \mathbb{R}^{r\times n}, V \in \mathbb{R}^{r\times m}. \tag{5.1}$$

The pseudo code of our proposed method is shown in Algorithm 3. In the algorithm, we optimize the quantity represented by Eq. (2.10) and use the SGD method to update the matrix X as $Update(X)$. We also denote *ano* as containing the de-identification method and its parameters.

As mentioned earlier, in our experiments we use the Hungarian algorithm to evaluate the attack success probability, but when we combine matrix factorization with k-anonymization, we can guarantee privacy quantitatively as in the following theorem.

[1] We denoted the case of NMF in [MKH+18].

Algorithm 3 (M, r, I, ano): De-identification using Matrix Factorization

```
Input: Original dataset $M$, rank $r$, de-identification function and the parameter $ano$, and the
    number of iteration $I$.
$t = 0$
Construct $U_t \in [0,1]^{n \times r}$ and $V_t \in [0,1]^{m \times r}$ randomly
while $t < I$ do
    $U_{t+1} = Update(U_t)$
    $V_{t+1} = Update(V_t)$
    $t = t + 1$
end while
$U'_{t+1} = A_{(ano)}(U_{t+1})$
return $X = U'^{\mathsf{T}}_{t+1} V_{t+1}$
```

Theorem 5.1. *If ano is a k-anonymization method, then $A_{(ano,r)}(D)$ satisfies k-anonymity.*

Proof. The record after de-identification $r_i \in A_{(k,r)}(D)$ is represented by $r_i = A_k(u_i)^{\mathsf{T}} v_j$. Here, $A_k(u_i)$ is k-anonymized and there are at most k kinds of data. Therefore, it can be said that r_i is also k-anonymized. □

Table 5.1: Dataset Format.

ID ($= i$)	Date	URL ($= j$)
x_{t_1} ($= 1$)	2016-12-01 16:13:48	www.google.com ($= 1$)
y_{t_1} ($= 2$)	2016-12-01 16:15:14	mail.google.com ($= 2$)
x_{t_1}	2016-12-01 16:17:13	www.youtube.com ($= 3$)
z_{t_1} ($= 3$)	2016-12-01 16:19:01	www.facebook.com ($= 4$)
x_{t_2} ($= 1$)	2016-12-01 16:21:15	www.youtube.com
x_{t_2}	2016-12-01 16:22:42	www.google.com
z_{t_2} ($= 3$)	2016-12-01 16:25:01	www.youtube.com

5.3 Experiment

5.3.1 Dataset

We use an actual web access log dataset as a time-sequence dataset. The dataset consists of an ID, a time-stamp, and the access domain as shown in Table 5.1. This dataset is converted into a matrix as follows.

$$M_T = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix}. \tag{5.2}$$

Here, T is the observation time. If a user with ID i accesses domain j during time T, then $r_{ij} = 1$, otherwise $r_{ij} = 0$. For example, the dataset in Table 5.1 is denoted as follows.

$$M_{t_1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.3}$$

$$M_{t_2} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{5.4}$$

Here, t_1 is the 10-minutes span between 2016-12-01 16:10:00 and 2016-12-01 16:19:59, and t_2 is the similar 10-minutes span between 2016-12-01 16:20:00 and 2016-12-01 16:29:59. The IDs are different between t_1 and t_2 but x_{t_1} and x_{t_2}, and z_{t_1} and z_{t_2} represent the same users. In the following experiment, we randomly selected 200 users and 1,000 domains from actual web access logs and had them change their pseudonym IDs every T for a specified time.

Table 5.2: Linkable attack against original dataset.

Observation Time	Risk against Linkable attack
$2h$	0.51
$4h$	0.64
$8h$	0.80

Table 5.3: Attacks against matrix factorization.

Rank	Risk against Re-identification attack	Risk against Linkable attack
10	0.98	0.31
20	1.00	0.45
30	1.00	0.54
40	1.00	0.58

5.3.2 Preliminary experiment

First, we evaluate whether a linkable attack as defined in Definition 5.2 can occur. The observation time t_1 was set to 2, 4, and 8 hours starting at 16:00 on a weekday, and the observation time t_2 was set to the same time on another weekday. The probability of a linkable attack between M_{t_1} and M_{t_2} is shown in Table 5.2.

This matrix only contains information on whether or not the domain has been accessed, and even if the observation time is 2 hours, the linkable attack probability, i.e., the risk of identifying the same user with a different temporary ID, is very high (more than 50 %). Moreover, the longer the observation time, the more pronounced the user's tendencies become; therefore, the longer the observation time, the higher the risk. The results show that people's web access patterns have surprisingly consistent characteristics. Therefore, it is necessary to pay attention not only to re-identification attacks but also to linkable attacks to prevent privacy leakage.

Next, we evaluate the effectiveness of the matrix factorization as a de-identification technique. In the following experiment, we fixed the observation time t_1 and t_2 as 8 hours from 16:00 on weekdays. The inputs of matrix factorization are original dataset $M \in \mathbb{R}^{200\times 1000}$, number of iterations I, and rank r. Furthermore, λ and γ, which are the hyper parameters. When $\gamma = 0.05$, and $\lambda = 0.01$, the relationship between error, namely $\sum_{ij} |(M_{t_1})_{ij} - (X_{t_1})_{ij}|$, and iteration is shown in Fig. 5.2. Hence, we fix $I = 100$, which is enough to converge. Since the accuracy of the dataset $X = UV^{\mathsf{T}}$ depends on the rank r and can be treated as a parameter of de-identification by matrix factorization, r was set as a parameter of the algorithm, $r = 10, 20, 30, 40$. We set larger values in the experiments in [MKH+18] but the results of the case $r > 40$ are saturated. The probabilities of re-identification and linkable attack are shown in Table 5.3.

The results show that matrix factorization itself is not very effective against re-identification attacks. Note that the privacy risk of re-identification attack is not reduced much by the matching algorithm because the matrix factorization can preserve the relative positions between records. When the rank is small enough, $r = 10$, the positional relationship is broken and the

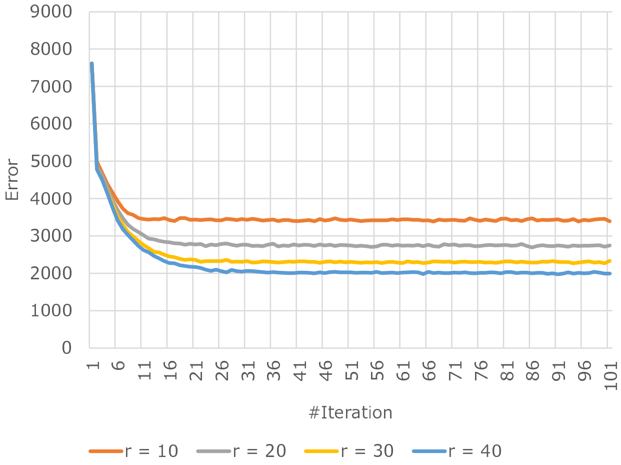


Figure 5.2: The relationship between the error and the number of iteration.

privacy risk is lowered slightly.

On the other hand, compared to re-identification attacks, rank affects the linkable attack probability between $A_r(M_{t_1})$ and $A_r(M_{t_2})$. This is because the relationship between the records of M_{t_1} and M_{t_2} is weaker than the relationship between M_{t_1} and $A_r(M_{t_1})$. In this experiment, we confirmed that the data set with 8 hours of observation time and $r = 30$ has almost the same privacy level as the data set with 2 hours of observation time.

5.3.3 Privacy risk evaluation

We evaluate our proposal, algorithm 3, in the following experiments. The overview of the expriments is shown in Fig. 5.3. In the following experiments, we apply [BKBL07b] as k-anonymization and Laplace noise as a noise addition. When noise addition is applied, noise $\epsilon \sim Lap(0, 2\phi^2)$ is added to each element and the parameter is ϕ.

1. Evaluate the privacy risk of re-identification attack between $A_k(M_{t_1})$ and M_{t_1}, and linkable attack between $A_k(M_{t_1})$ and $A_k(M_{t_2})$.

2. Evaluate the privacy risk of re-identification attack between $A_\phi(M_{t_1})$ and M_{t_1}, and linkable attack between $A_\phi(M_{t_1})$ and $A_\phi(M_{t_2})$.

3. Evaluate the privacy risk of re-identification attack between $A_k(U_{t_1})^\top V$ and M_{t_1}, and linkable attack between $A_k(U_{t_1})^\top V$ and $A_k(U_{t_2})^\top V$.

4. Evaluate the privacy risk of re-identification attack between $A_\phi(U_{t_1})^\top V$ and M_{t_1}, and linkable attack between $A_\phi(U_{t_1})^\top V$ and $A_\phi(U_{t_2})^\top V$.

The evaluation of the re-identification attack in Experiments 1 and 2 is almost identical to what has been done in many previous studies. The difference is the privacy metric (see Definition 5.3) and these results are used for comparison with experiments 3 and 4, which are the evaluations of our algorithm. There are few studies on linkable attacks, and their evaluation is one of our contributions.

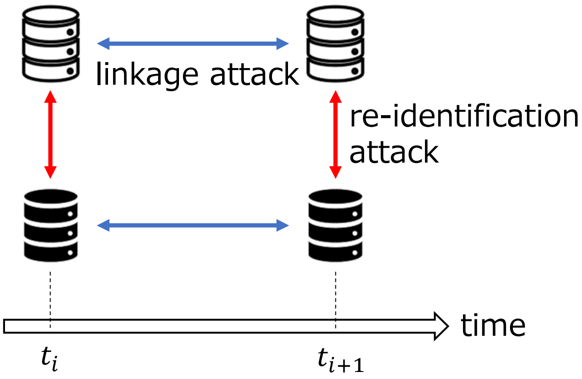


Figure 5.3: Overview of the experiment.

Table 5.4: Experiment 1: The privacy risk of a k-anonymized data.

k	Re-identification Attack	Linkable Attack
2	0.500	0.185
4	0.250	0.050
6	0.167	0.038
8	0.125	0.027
10	0.098	0.023

The evaluation of re-identification attack in experiment 1 (Fig. 5.4) is simple and the result is almost the same as k-anonymization. However, the attack success probability is slightly different from $1/k$ because our privacy assessment is slightly different from k-anonymity. This is because we use the Hungarian algorithm to optimize the overall distance, but we think it is sufficient to know the privacy trend. Note that $A_k(X)$ is naturally k-anonymous. The result of the linkable attack (compare Fig. 5.2 with Fig. 5.4) also shows that k-anonymization can significantly improve the privacy of linkable attack and 2-anonymization can reduce the privacy lisk by 77% ($0.8 \rightarrow 0.185$).

The evaluations in Experiment 2 are shown in Table 5.5. The privacy risk for re-identification attacks decreases for $\phi \geq 0.9$, and the score appears to get better as phi increases. However, if, for example, $\phi = 1.5$, most of the records have changed by more than 1 due to noise. However, the original values of M are 0 or 1 respectively, i.e. $M_{ij} \in \{0, 1\}$. Therefore, it is easy to imagine that the noise is too large to maintain the utility. We conclude that simple noise addition is not a good de-identification method from the viewpoint of utility preservation. On the other hand, for linkable attacks, we have interesting results. For linkable attacks, privacy is improved even when the noise is very small, which means that adding even small noise is an effective countermeasure against linkable attacks.

In Experiment 3, we evaluated the effectiveness of the proposed algorithm combining matrix factorization and k-anonymization. Table 5.6 is the result of the re-identification attack. In the experiments, we did not find much effect of matrix factorization, but we can see that privacy tends to improve slightly as r increases. This may be because k-anonymization has a dominant

Table 5.5: Experiment 2: The privacy risk of a noise added data.

ϕ	Re-identification Attack	Linkable Attack
0.3	1.00	0.33
0.6	1.00	0.10
0.9	0.95	0.01
1.2	0.81	0.03
1.5	0.62	0.00

Table 5.6: Experiment 3: The privacy risk of a data applied with Algorithm 1 (SGD + k-anonymization) for Re-identification attack.

k	$r = 10$	$r = 20$	$r = 30$	$r = 40$
2	0.44	0.50	0.50	0.50
4	0.21	0.24	0.25	0.25
6	0.12	0.14	0.15	0.16
8	0.10	0.11	0.11	0.12
10	0.08	0.08	0.08	0.08

effect on the re-identification risk and the effect of matrix factorization is hard to see.

The results of linkable attack in Experiment 3 are shown in Table 5.7. The experiment does not provide any new insight into the effect of matrix factorization. When datasets observed at different times are sufficiently de-identified by k-anonymization, the relationship between the same users in each dataset may be almost completely destroyed and only some outliers may be linked.

In Experiment 4, we evaluate the impact of our method, which is a combination of matrix factorization and noise addition. The evaluation results of the re-identification attack are shown in Table 5.8. The user's feature U is added with noise and multiplied by U^{T} and V. Therefore, we cannot simply compare with experiment 2. but we can see that the effect of matrix factorization is large and that the privacy risk changes smoothly by combining matrix factorization and a little noise. This result shows that matrix factorization can be used to construct a de-identified dataset flexibly from the viewpoint of privacy. For example, it was confirmed that $A_{(\phi=0.15,r=20)}(M_{t_1})$ and $A_{(\phi=0.20,r=40)}(M_{t_1})$ are approximately as secure as $A_{(k=2)}(M_{t_1})$ and $A_{(\phi=1.5)}(M_{t_1})$.

The results of the linkable attack in Experiment 4 are described in Table 5.9. This is the same trend as the re-identification attack, and matrix factorization is compatible with noise addition. We present the details of the results of the re-identification attack and the linkable attack in Figs. 5.4 and 5.5.

Table 5.7: Experiment 3: The privacy risk of a data applied with Algorithm 1 (SGD + k-anonymization) for Linkable attack.

k	$r = 10$	$r = 20$	$r = 30$	$r = 40$
2	0.11	0.15	0.15	0.15
4	0.05	0.07	0.08	0.07
6	0.04	0.03	0.03	0.04
8	0.03	0.03	0.03	0.03
10	0.02	0.02	0.02	0.02

Table 5.8: Experiment 4: The privacy risk of a data applied with Algorithm 1 (SGD + noise addition) for Re-identification attack.

ϕ	$r = 10$	$r = 20$	$r = 30$	$r = 40$
0.05	0.75	0.95	0.97	1.00
0.10	0.42	0.72	0.85	0.86
0.15	0.25	0.50	0.61	0.70
0.20	0.18	0.28	0.40	0.49

5.3.4 Utility evaluation

Next, we evaluate the utility of the de-identified data set. Following the experiments in Chapter 4, we evaluate the utility of a data set based on machine learning. A logistic regression[2] is applied in the following experiment and the parameters are default setting. One of the uses of access logs is to predict malicious sites and inform web browser users of them. Therefore, we use machine learning algorithms to predict whether each user will access a malicious site or not. We generate training models using the original and de-identified datasets, and input the same test dataset to each model. The utility score is defined in Definition 4.6 and the F-measure of the model of the original dataset was 0.763. Each result of the evaluation is shown in Tables. 5.10, 5.11, 5.12, and 5.13.

1. Evaluate the utlity of $A_{(k)}(M_{t_1})$ for $k = 2, 4, 6, 8, 10$.

2. Evaluate the utlity of $A_{(\phi)}(M_{t_1})$ for $\phi = 0.3, 0.6, 0.9, 1.2, 1.5$.

3. Evaluate the utlity of $A_{(k=2,r)}(M_{t_1})$ for $r = 10, 20, 30, 40$.

4. Evaluate the utlity of $A_{(\phi,r)}(M_{t_1})$ for $\phi = 0.1, 0.15$ and $r = 10, 20, 30, 40$.

In Experiment 1, the results show that the usefulness is maintained even as $k = 10$. This is because the dataset used is $M_{ij} \in \{0, 1\}$, which allows us to determine whether each element is 0 or non-zero even when k-anonymized, resulting in the maintenance of a high level of usefulness. However, the utility of k-anonymized datasets is expected to decrease when it comes to complex datasets that cannot be expressed as 0 or 1, such as image data.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Table 5.9: Experiment 4: The privacy risk of a data applied with Algorithm 1 (SGD + noise addition) for Linkable attack.

ϕ	$r = 10$	$r = 20$	$r = 30$	$r = 40$
0.05	0.21	0.34	0.34	0.50
0.10	0.12	0.15	0.14	0.20
0.15	0.07	0.11	0.09	0.10
0.20	0.03	0.03	0.03	0.02

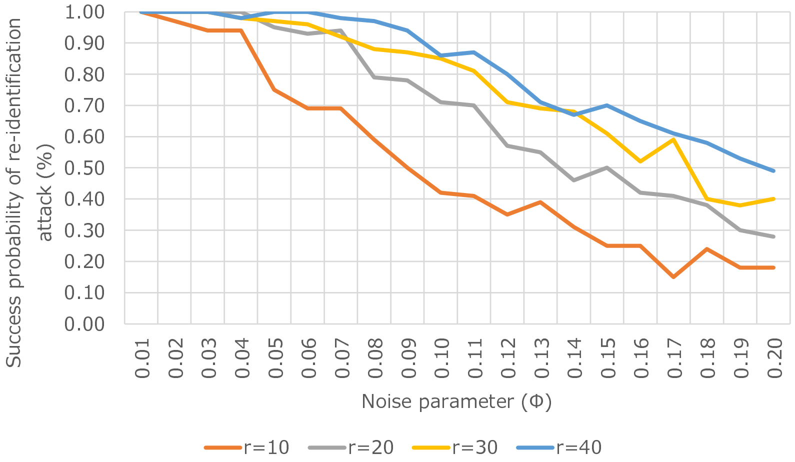


Figure 5.4: The re-identification risk of combination of matrix factorization and noise addition.

The results of the Experiment 2 shows that the utility of the dataset decreases as the noise increases. As shown in Section 5.3.3, each element in the original dataset is either 0 or 1, so when the noise parameter is large, such as $\phi = 1.5$, the utility deteriorates rapidly.

When k-anonymization is combined with matrix factorization, the effect of matrix factorization is small on utility as well as on privacy risk. In the Experiment 3, we can see that the effect of k-anonymization is large and that the effect of matrix factorization is relatively small.

In Experiment 4, we evaluated the combination of noise addition and matrix factorization, and obtained good results (Tables 5.13 and 5.14). When each de-identified dataset has the same privacy level, the dataset generated by the combination of matrix factorization and noise addition is more useful than the dataset generated by noise addition.

In the experiments, we can say that our proposal algorithm has at least three strengths. First, the proposed algorithm can flexibly control the privacy risk using the parameter r. For example, Fig. 5.4 shows that the privacy improves as the rank r decreases. Second, the matrix factorization itself is efficient when considering a linkable attack model. The privacy against linkable attacks can be easily improved by using matrix factorization, since the relationship between records in M_{t_1} and M_{t_2} is weak. Finally, the proposed algorithm can improve the privacy of the dataset while maintaining the utility of the dataset, especially when noise addition is combined with matrix factorization. For example, the privacy risk and the utility of $A_{(\phi=1.5)}(M_{t_1})$ are 0.62 and 0.744. On the other hand, those of $A_{(\phi=0.15,r=30)}$ are 0.61 and 0.907. This means our proposal algorithm can improve the utility maintaining the privacy of the dataset.

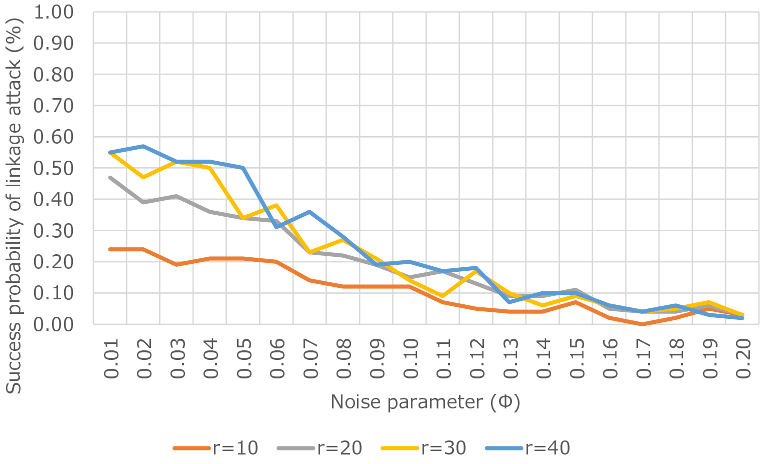


Figure 5.5: The linkable attack risk of combination of matrix factorization and noise addition.

Table 5.10: Utility Evaluation 1: The utility of k-anonymized data.

Dataset D	Precision	Recall	F-measure	$Uti(D)$
$A_{(k=2)}(M_{t_1})$	0.780	0.720	0.749	0.981
$A_{(k=4)}(M_{t_1})$	0.741	0.688	0.714	0.936
$A_{(k=6)}(M_{t_1})$	0.755	0.691	0.721	0.946
$A_{(k=8)}(M_{t_1})$	0.737	0.659	0.696	0.913
$A_{(k=10)}(M_{t_1})$	0.748	0.677	0.711	0.932

5.4 Conclusion

In this chapter, we proposed a de-identification method using matrix factorization. The matrix factorization can be easily combined with other de-identification methods, and we specifically focus on k-anonymization and noise addition. The datasets de-identified by the de-identification method combining k-anonymization and matrix factorization have k-anonymity. We conducted experiments and showed that the de-identification method combining matrix factorization and noise addition can maintain higher utility than noise addition alone. Furthermore, we discussed the risk of linkable attacks between pseudonymized identical records. The experimental results show that when the de-identification is insufficient, the risk against linkable attacks remains, but the risk is reduced even with very small distortions provided by matrix factorization.

Table 5.11: Utility Evaluation 2: The utility of noise added data.

Dataset D	Precision	Recall	F-measure	$Uti(D)$
$A_{(\phi=0.3)}(M_{t_1})$	0.780	0.664	0.717	0.941
$A_{(\phi=0.6)}(M_{t_1})$	0.738	0.610	0.668	0.876
$A_{(\phi=0.9)}(M_{t_1})$	0.719	0.541	0.618	0.810
$A_{(\phi=1.2)}(M_{t_1})$	0.652	0.507	0.571	0.748
$A_{(\phi=1.5)}(M_{t_1})$	0.625	0.520	0.567	0.744

Table 5.12: Utility Evaluation 3: The utility of data applied with Algorithm 1 (SGD + k-anonymization).

Dataset D	Precision	Recall	F-measure	$Uti(D)$
$A_{(k=2,r=10)}(M_{t_1})$	0.686	0.735	0.710	0.930
$A_{(k=2,r=20)}(M_{t_1})$	0.699	0.767	0.731	0.959
$A_{(k=2,r=30)}(M_{t_1})$	0.695	0.773	0.732	0.960
$A_{(k=2,r=40)}(M_{t_1})$	0.712	0.786	0.747	0.980

Table 5.13: Utility Evaluation 4: The utility of data applied with Algorithm 1 (SGD + noise addition).

Dataset D	Precision	Recall	F-measure	$Uti(D)$
$A_{(\phi=0.10,r=10)}(M_{t_1})$	0.742	0.650	0.693	0.909
$A_{(\phi=0.10,r=20)}(M_{t_1})$	0.752	0.688	0.719	0.943
$A_{(\phi=0.10,r=30)}(M_{t_1})$	0.736	0.703	0.719	0.943
$A_{(\phi=0.10,r=40)}(M_{t_1})$	0.737	0.735	0.736	0.965

Table 5.14: Utility Evaluation 5: The utility of data applied with Algorithm 1 (SGD + noise addition).

Dataset D	Precision	Recall	F-measure	$Uti(D)$
$A_{(\phi=0.15,r=10)}(M_{t_1})$	0.718	0.614	0.662	0.868
$A_{(\phi=0.15,r=20)}(M_{t_1})$	0.748	0.655	0.698	0.915
$A_{(\phi=0.15,r=30)}(M_{t_1})$	0.704	0.680	0.692	0.907
$A_{(\phi=0.15,r=40)}(M_{t_1})$	0.716	0.711	0.713	0.935

Chapter 6

De-identification Technique for Unstructured Data

An enormous amount of documents, including news articles, public reports, and personal essays, are published on websites and social media. In addition, governments, public offices, and corporations exchange and publish a vast amount of documents, including personal information. For example, some medical institutions are using medical data for epidemiological research and disease prevention. Once privacy-related information is released, the impact of privacy violations can be enormous, so document review prior to release needs to be done very carefully. In many cases, human experts sanitize documents before they are published, but this can be inefficient in terms of cost and accuracy because no systematic rules for automatic processing have been developed. Furthermore, such measures do not guarantee that critical privacy risks are eliminated from the documents. However, organizations attempting to release data need to define appropriate protection measures due to the sensitivity and importance of this information. One possible solution to these problems is to incorporate privacy risk analysis methods and document sanitization algorithms for publishing documents that contain privacy-sensitive information.

There are many privacy-preserving techniques for structured data (see Chapter 2), however, document data are unstructured and contain risk words, where the risk depends on the contexts of the document data. Consequently, a new approach to preserving privacy is needed to reduce the risk posed by publishing document data.

In this chapter, we focus on privacy risk analysis of document data. An attacker has a target record (or a dataset), a processed dataset, an attack algorithm, and a risk evaluation subroutine. The goal of the attacker in this chapter is to re-identify the person associated with the document. Once a person is re-identified, the attacker can link that person to additional information in the document, causing a serious invasion of privacy.

Our proposal provides the following contributions: First, we define an actual adversary model for document data and propose a realistic web-based attack algorithm. In comparison with other models [SB16, CGRM08], the aim of an attacker in this model is to discover words that the sanitizer misses and to re-identify a person associated with the document. In our proposal, in contrast to conventional methods, there is no need to define or specify sensitive words, nor is there a need to evaluate the relationship between sensitive words and words in the document. Therefore, any kind of document can be handled without considering the length of the article. Second, we propose a privacy-preserving algorithm. It uses a web-based attack algorithm as a subroutine. The attack algorithm searches for informative words and uses those words as keywords to find articles related to documents on the Internet. If the article contains additional information, the privacy-preserving algorithm sanitizes the keywords. This algorithm can reduce the privacy risk by finding words that may be at risk of re-identification by combining with other words. In addition, since the proposed algorithm generalizes or removes only risky words, it does not reduce the amount of information excessively, and as a result, it can maintain its utility. Finally, we applied the attack algorithm to two real Japanese document datasets, which had previously been manually sanitized, and confirmed that the algorithm worked effectively in both cases. The experimental results show that manual sanitization is insufficient and that the proposed algorithm reduces the risk of re-identification by sanitizing words that are at risk of re-identification.

The remainder of this chapter is organized as follows: Section 6.1 introduces a framework for de-identification of document data. Section 6.2 follows the proposed framework to evaluate privacy risks for actual document data and to check the effectiveness of our proposal. Section 6.3 concludes this chapter.

6.1 Framework for Document De-identification

6.1.1 Framework for document de-identification

In this section, we provide a framework for document sanitization. We assume that users who are reporters or users with sanitizing authority (hereafter, we refer to users as “sanitizers” as in previous studies) will adopt this framework when checking the privacy of documents.

There are some de-identification techniques such as generalization and data deletion, and we call the de-identification techniques “sanitization” collectively. This framework consists of three parts: preprocessing, simulated attack, and sanitization. We propose a privacy preserving algorithm that assumes an attacker on document data. In the previous chapters, the scope of data publication was limited and the attacker was assumed to have no external knowledge. However, in this chapter, we assume that the documents will be made public, so we need to assume a strong attacker.

The first step in the pre-processing is to structure the data. Documents are not structured, and thus it is difficult to handle them. Therefore, we divide documents d into word sets w. Furthermore, this part calculates the amount of information of each word for the next part. Pre-processing is used to classify the input data set in a complementary manner. Then, a simulated attacker accesses the internet and searches for documents related to a target document. Therefore, the risk of attack depends on the degree of interest in the target document. For example, since major disasters are covered in the news, documents about major disasters can be found on the Internet more easily than those about small accidents. Pre-processing tags the target dataset, and the attack part utilizes this information. If a sanitizer is not a sanitization proxy, i.e., if a sanitizer is a data holder and has an additional dataset D^{**}, the dataset is available as optional input. In the attack part, a simulated attack on the target document d_i is performed, and a list of retrieved documents related to the target document is created. A simulated attacker searches the internet using words with a large volume of information and outputs a document list $List_i$, which can be found on the web. The sanitization part first evaluates the documents in $List_i$. If the document contains information known only to the data holder, the keywords used in the web search are considered to have privacy risks. We refer to the words that have risks of re-identification including identifiers as "risk words". Finally, the risk words are sanitized and the sanitized document d'_i is output. The pseudo-code of the proposed framework is described below.

Algorithm 4 `DocumentSanitization`$(d, (D^{**}))$: Document sanitization framework.

```
Input: A target document dataset $d = \mathcal{A}(D)$ (and a document dataset $D^{**} \supseteq D$.)
Output: A sanitized document dataset $d'$
1: $(w, I(w), Label) \leftarrow$ Preprocessing$(d, (D^{**}))$
2: $List \leftarrow$ Attack$(w, I(w), Label)$
3: $d' \leftarrow$ Sanitization$(d, List)$
4: return $d'$
```

Before we move on to the description of each sub-protocol, we summarize the assumptions of the proposed framework. This framework is assumed to be used when a sanitizer, who is the reporter or who has sanitization authority, checks the document privacy. A sanitizer is assumed to have target documents $\mathcal{A}(D) = d = \{d_1, ..., d_n\}$. D is an original dataset and $\mathcal{A}(D) = d$ is a set of manually sanitized documents d_i, and $d_i = \{A_i, g(R_i)\}$. Here, $g(\cdot)$ is a generalization function, R_i is a set of information containing re-identification risk that may include such information as the name of the person associated with document d_i, and A_i is the other information about d_i. We first regard a sanitizer as having d and $D \subseteq D^{**}$, namely we consider the case that the sanitizer is the data owner. The case where the data holder and sanitizer are not the same will be discussed in the following section. The sanitizer may use another dataset $X \subseteq D^{**}$ to calculate the information content of the words and tag the documents. On the other hand, an attacker is assumed to have a sanitized dataset d and access to D^*. D^* is an ideal dataset and includes a part of R_i. We assume that D^* is a document dataset on the web, and a simulated attack $\mathcal{A}^*$ is used for linking d_i to $d^* \in D^*$ and obtaining R_i from d^*. This attack is valid when a target document d_i is related to events that can be known publicly, and we focus on documents related to accidents that occur at schools and court documents. It is easy for human beings to judge whether a word in d^* is r_i, so, the evaluation function can be very simple. If the attack succeeds, namely, an article about d_i exists on the web and includes additional information such as the name of the victim, $E(\mathcal{A}^*(d_i, D^*), R_i) = 1$, and 0 otherwise. However, the evaluation must be mechanical and exclude human factors. When a sanitizer has D, a simulated attacker knows that $R_i \in D_i$ and $E(\mathcal{A}^*(d_i, D^*), R_i)$ can be evaluated. However, a sanitizer may not have D because some data owners commission an outside agency to evaluate the risk of sanitized datasets. Therefore, we need to consider a more flexible evaluation function with caution. The evaluation function is discussed in Section 6.1.2.

6.1.2 Details of functions

The framework for document sanitization is consists of three parts; `Preprocessing`, `Attack`, and `Sanitization`. In this subsection, we explain the details of each function.

Documents are not structured, and some processing is thus needed to handle them. In the pre-processing part, we first perform morphological analysis and define the document as a set of words according to the conventions of previous studies [CGRM08, SB16]. More precisely, we define a document i including m words as $d_i = w^i = \{w^i_1, ..., w^i_m\}$ (denoted as $w^i_j = w_j$ for simplicity). Some words have risk of re-identification, and without loss of generality, we denote $A_i = \{w_1, ..., w_l\}, R_i = \{w_{l+1}, ..., w_m\}$. The pre-processing algorithm, then, runs $CalculationI$ to calculate the amount of information of each word $I(w) = \{I(w^1_1), ..., I(w^n_m)\}$. $CalculationI$ requires word set $w^i = \{w_1, ..., w_m\}$ and d and calculates the volume of information of each word $w_j \in d_i$. The volume of information of w_j is defined as

$$I(w_j) = -\log P(w_j) + \epsilon(w_j). \tag{6.1}$$

Here, $P(w_j)$ is the appearance probability of w_j, and $\epsilon(w_j)$ is a moderator variable. $P(w_j)$ is calculated as $P(w_j) = \frac{\#d_i(w_j)}{\#d}$, where $\#d$ is the number of words in d and $\#d_i(w_j)$ is the number of $w_j \in d_i$. Furthermore, since the sanitizer may have D^**, we define $P(w_j)$ as follows

$$P(w_j) = \frac{\#d_i(w_j)}{\#D^{**} + \#d}, \tag{6.2}$$

where $\#D^{**}$ is the number of words in D^{**} and $\epsilon(w_j)$ is the parameter for selecting appropriate words. In general, natural language processing (NLP) requires tuning based on data, which is expressed here as $\epsilon(w_j)$. More precisely, the emphasis is on parts of speech. Prepositions, conjunctions, and adjectives are rarely used in web search, so we put more weight on nouns and verbs. In addition, nouns and verbs that have a similar meaning to the word are given negative weight. For example, we consider a document and find that $-\log P(w_j)$ of $w_j =$"first" and $w_j =$"second" are high. They may be regarded as candidates of risk words when $I(w_j) = -\log P(w_j)$, but they have similar meanings. Another important adjustment is to reduce the weight of words that are not included in D_i but are included in d_i. This is because the word is considered to be a modification and generalization of D_i by the sanitizer. Hence, $I(w_j)$ of one of them decreases by adjusting $\epsilon(w_j)$ and is excluded from the candidates. For the other words, we adjust $\epsilon(w_j)$ so as to find words that have different vectors. Furthermore, the pre-processing part tags each document d_i using the classification protocol. The classification is arbitrary, but has advantages in terms of computational cost and accuracy. The classification protocol classifies documents from the standpoint of interest, and documents with a high interest label is attacked intensively for efficiency. These tasks require d, and the accuracy is expected to improve by using $D^{**} \supseteq D$. We focus on reports of accidents at a school and court documents, and when a target document is sensational, such as a murder, a greater weight assigned to the document.

Algorithm 5 `Preprocessing`$(d, (D^{**}))$: Preprocessing for document sanitization.

```
Input: A target document dataset $d = \mathcal{A}(D)$ (and a document dataset $D^{**} \supseteq D$.)
Output: A word set $w$, information content $I(w)$, and $Label$ for each document.
for $i < n$ do
    $w^i \leftarrow MorphologicalAnalysis(d_i)$
    $I(w^i) \leftarrow CalculationI(w^i, d)$
    $Label_i \leftarrow Labeling(d_i, d)$
end for
return $w, I(w), Label$
```

The attack part follows the pre-processing part. The attack algorithm in this part inputs a word set $w = \{w^1, ..., w^n\}$ and additional information $I(w) = \{I(w^1), ..., I(w^n)\}$. w^i is a word set of document d_i, and $I(w^i) = \{I(w_1), ..., I(w_l), I(g(w_{l+1})), ..., I(g(w_m))\}$ represents the amount of information of words. The attack algorithm, `Attack`, inputs a word set $w = \{w^1, ..., w^n\}$ with the volume of information $I(w)$ and $Label$. Here, $Label = \{Label_1, ..., Label_n\}$ is the tag set of documents, and when the pre-processing algorithm does not run the classification protocol, $Label_i = \phi$. Moreover, some parameters are actually required, but we omit them here for simplicity. The attack algorithm calls $SetKeywords$ and $WebSearchAttack$. $SetKeywords$ requires w^i, $I(w^i)$, and $Label_i$ and outputs a set of words that have a large volume of information KW_i. Subsequently, the attack algorithm calls $WebSearchAttack$. $WebSearchAttack$ searches for documents related to d_i using $kw \in KW_i$ and returns $List_i = \bigcup(List_i(kw), kw)$. $List_i(kw)$ is the set of documents found using kw as keywords. Note that we have to limit the number of documents to search for and the number of kw sets. Regarding the document search, the number of documents has little effect on the run time and the evaluation result. We fix $\#List(\cdot) = 10$, which represents the number of web pages that are displayed in a web browser at one time. Moreover, kw generates the $2^{|KW_i|} - 1$ combinations, and the run time of the algorithm strongly depends on $|KW_i|$. This must be a parameter of the algorithm, and we set $|KW_i| = 3$ in the following experiments. After that, we only need to check whether $r_{ij} \in R_i$ is included in the $List_i$ to confirm that the attack has succeeded. If $r_{ij} \in R_i$ is included in $List_i$, document i is at risk and an attacker may obtain additional information about d_i.

In this chapter, we focus on documents about accidents that occurred at a school. One document includes the exact date and time of the accident, the place, gender, grade, and the name of the student victim, his/her medical history, the compensation value of the accident, etc., in addition to a report on the accident situation. The details of the dataset are stated in subsection 6.2.1. The names of the victims of each accident was removed from the documents we received. Accidents in schools often include the names of students in news documents. Therefore, we assume that D^* represents news articles on the web, and define an attack as a search on the internet for a document containing the victim's name. Metadata, which is information associated with a document, such as medical history, has practical value, but if the name of a student associated with a document is revealed, the metadata is tied to the student, resulting in a serious privacy violation.

Algorithm 6 `Attack`$(w, I(w), Label)$: Web-search attack algorithm.

Input: A word set w with the volume of information $I(w)$, and Label
Output: A searched document set $List$

```
for $i < n$ do
    $KW_i \leftarrow SetKeywords(w^i, I(w^i), Label_i)$
    $List_i \leftarrow WebSearchAttack(KW_i)$
    $List = List \cup List_i$
end for
return $List$
```

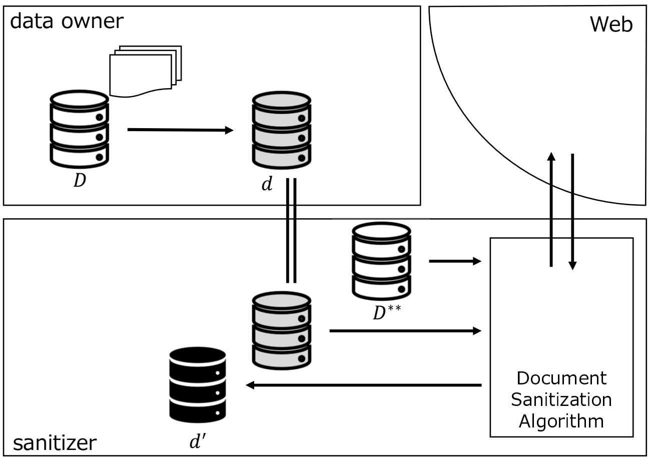


Figure 6.1: Overview of the model.

Finally, we construct an algorithm to sanitize risk words by using the `Attack` algorithm. The sanitization algorithm requires a target document dataset d and $List$. $List_i \in List$ includes keywords kw and documents found on the web. In the sanitization algorithm, $RiskEvaluation$ and $Reconstruct$ are executed. $RiskEvaluation$ compares d_i with the documents included in $List_i$ and then extracts risk words. $Reconstruct$ removes or generalizes the words in d_i and the sanitized document d'_i is output. Subsequently, the privacy risk of each document is calculated by $RiskEvaluation$, the function of which is to evaluate the privacy risk of kw. When a document $d_{search} \in List_i(kw)$ includes words $r_{ij} \in R_i$, kw violates the privacy of d_i and $RiskEvaluation$ returns 1. The kw is input in $RiskWords_i$, and finally this algorithm runs $Reconstruct$ and outputs a sanitized document d'_i. An overview of our model is shown in Fig. 6.1.

Algorithm 7 `Sanitization`$(d, List, (D^{**}))$: Privacy-preserving algorithm for documents using attack simulator.

```
Input: A target document dataset $d$ and a document set $List$.
Output: A sanitized document dataset $d'$.
for $i < n$ do
    if $RiskEvaluation(d_i, List_i, (D^{**})) == 1$ then
        $RiskWords_i = RiskWords_i \cup kw$
    end if
    $List \leftarrow List - List(kw)$
end for
$d'_i \leftarrow Reconstruct(d_i, RiskWords_i)$
return $d' = \bigcup d'_i$
```

6.2 Experiment

6.2.1 Dataset

The first target document to be considered in this chapter is the school accident document. We perform morpheme analysis of the documents using Mecab[1] and define all words that appear in the results as A_i. These documents are owned by the Japan Sport Council (JSC), and we can use (almost) original document data D and sanitized data $\mathcal{A}(D) = d$. The original data D have the exact date and time of the accident, the location, the diagnosis (or cause of death), the gender and grade of the student victim, his/her medical history, and the compensation value of the accident, etc., in addition to a report on the accident situation. The name of the victim is clearly included in the original data D, but the dataset we received d does not include such information. In addition, d is sanitized in multiple ways to reduce privacy risks. Not only are the document data sanitized, but metadata are also generalized and deleted, such as the date and time of the accident, medical history, and the compensation value of the accident. Note that some sanitized data are open to the public by JSC[2], and we can confirm that there is no confidential information. The published data do not include sensitive information, such as medical history and the compensation value of the accident, so even if the name of a victim is revealed by the attack (with this information being publicized by a news report), the attacker cannot obtain additional information from the open data. However, the lack of information leads to a lack of value of the data. For instance, some research institutes review these accident data, analyze the scale of the accident, and use the results to prevent a future accident. In this case, the medical history information and the compensation value of the accident could be useful. Therefore, the linkability between the data and the name of the victim should be reduced to protect the victim's privacy, and at the same time, the utility of the data should be maintained. We have approximately 700 original fatal accident documents (OADs) and more than 4,000 sanitized accident documents (SADs). All sanitized versions of the OADs are included in the SADs. We reviewed the original documents on a contractual basis for research purposes, and they are not published. On the other hand, the sanitized documents were manually sanitized

[1] https://github.com/neologd/mecab-ipadic-neologd

[2] https://www.jpnsport.go.jp/anzen/anzen_school/anzen_school/tabid/822/Default.aspx

by staff members of the JSC and are disclosed online.

The other target in this chapter is court documents in Japan. In particular, murder cases were used for the experiment because they tend to be broadcasted/published as news in the public domain such as the internet. Accordingly, our attack algorithm can easily capture the information related to the documents. As in the experiments on the school accident documents, we performed a morpheme analysis of the documents using Mecab and defined all words in the overview section as A_i. We collected these documents from a website[3].In contrast to the accident documents, suspects, victims, and other information such as their age are de-identified, but the degree of de-identification of these documents is lower than that of the school data. We denote all court documents as sanitized court documents (SCDs). Different from accident documents, court documents do not have metadata, so we define the top 20 words with a large volume of information as metadata. For experiments, we downloaded 30 documents concerning murder and 1,000 documents randomly to calculate the volume of information.

6.2.2 Attack simulation

It is necessary to define risk words in order to estimate the privacy risk of a document. Here, the goal of the attackers of the documents is to reveal the names of people who are vulnerable to privacy violations. In previous chapters, we have referred to privacy risk as the re-identification of individuals, and we continue that idea in this chapter. R_i is a set of words w_j that link to websites of accidents and incidents related to d_i. If the sanitizer has the original data set containing the victim's name, then R_i can be considered the victim's name. However, as mentioned earlier, the sanitizer may not know the name of the victim associated with d_i. In other words, a simulated attacker is assumed to have a sanitized dataset d and access to D^*. The risk words of d are $g(R)$ and may not include the name of the victim. The school accident data we have this time also had the names of individuals erased. Therefore, We first manually checked each document and searched for the names of the people relevant to our experiments by using $SetKeywords$ and extracting the candidates keywords. Then, we searched the internet to identify the victim's name and set it as R_i. It was also manually judged whether the name is correct. In the first experiment, $RiskEvaluation$ returns 1 when a document $d \in List(kw)$ contains words $r_{ij} \in R_i$, implying that it also contains the tagged name of a relevant person, which is defined as r_{ij}. On the other hand, if a sanitizer does not have the name of a relevant person, this process takes huge computational time as well as a manual search of the risk documents. Consequently, we need some indices for automation. We focus on the number of words that are included in the metadata of each document. We assume if the number of words appearing in both a searched document and the metadata increases, the possibility that the two documents indicate the same accident will also increase. In the case of court documents, no document has metadata, but there are many words including a large volume of information, and thus these words are handled as metadata. After calculating the volume of information of each word and setting kw, we set words with a high volume of information other than kw as

[3] https://www.courts.go.jp/app/hanrei_jp/search1

R_i. The validity of this assumption is considered in the following.

In our experiments on accidental documents, we used both original and sanitized documents. First, we analyze the risk of OAD and then apply our algorithm to SAD to make sure that there is still a privacy risk of victim identification in SAD. We set 700 OADs as D, the corresponding 700 SADs as d, the other 1,600 SADs as D^{**}, and the victim's actual name as R_i. `Preprocessing` applies the morpheme analysis to d and evaluates $I(w_j)$ for each word. In this experiment, we obtained $\#d \fallingdotseq 150,000$ as the result of the morpheme analysis. The words with the highest value are $\#d_i(w_j) = 1$ and $I(w_j) = 23.80$, followed in order by 22.79 and 22.21. $SetKeywords$ outputs KW_i, which is the set of words $w_j \in d_i$, where it has a large volume of information. We focus on the words having the top three volumes of information, namely the words w_j s.t. $I(w_j) \geq 22.21$. In the experiment, words with the top three informative values were named potential risk words (PRWs) in order to compare OAD and SAD. We classify the words w_j into three classes: (1) words that appear only in $D_i \in D$; (2) words that appear in both D_i and d_i, i.e., that are not sanitized manually; and (3) words that have a risk and appear in d_i, that is, words that are keywords leading to the acquisition of the name of the victim by web search (Fig. 6.2). There are $93 + 40 + 36 = 169$ PRWs in the OADs (1), and we find $35 + 20 + 21 = 76$ PRWs remaining in the corresponding SADs (2). This result shows that $169 - 76 = 93$ PRWs are sanitized in SADs, but nevertheless many PRWs remain in the SADs. Then, we run $SetKeywords$ and launch $WebSearchAttack$ using the PRWs in the `Attack` algorithm. KW_i is input to $WebSearchAttack$, and it outputs a set of searched documents $List_i$. $RiskEvaluation$ is populated with the set of retrieved documents, and outputs 1 if the attack is successful, and the keyword $RiskWords$ is updated. Here, $RiskWords$ contains risk words (e.g., specific location names). In the experiment, the $RiskEvaluation$ algorithm found that 12 out of 76 PRWs are associated with the victim's name. $RiskWords_i$ is the set of words classified in class (3).

The experimental results focusing on PRWs are listed in Table 6.1. Here, we can see 93(=Class (1) \ Class (2)) PRWs are sanitized manually. However, the other 76(=Class(2)) words are not sanitized and remain. We found that in OADs, proper nouns such as the name of the facility were often considered PRWs, while in SADs, they were all sanitized. However, some events such as "plane accidents" that are PRWs are not sanitized even in the SADs and are published. The attack simulator in the algorithm exploits PRWs, 12 of which (=Class (3)) link to actual articles containing the victim's name. This revealed that an attacker could re-identify the person associated with a document and link other information, such as medical history, to the re-identified person. In conclusion, there is still a risk of identification in manual sanitization, and our algorithm can efficiently detect the risk words missed by manual sanitization. Finally, $Reconstruct$ removes or generalizes the 12 words from d and outputs d'.

Our algorithm performs a simulation attack and removes only the informative words that cause the re-identification of people associated with a document. In this experiment, there were 700 documents that were sanitized manually and we found 12 words that caused re-identification among them. As mentioned before, we focus on the risk of re-identification, which is a critical issue for privacy, and we can minimize the decrease of utility due to deletion or generalization.

Table 6.1: Relationship between $I(w_j)$ and the class of words (I).

$I(w_j)$	Class (1)	Class (2)	Class (3)
23.80	93	35	9
22.79	40	20	3
22.21	36	21	0
#PRWs	169	76	12

Sanitized words can restore the linkage between people and documents, which must be a critical issue. Thus, even if an analyst seeks to maintain utility, at least sanitization of those words is necessary. Therefore, we can say our algorithm preserves privacy while maintaining utility. Furthermore, the experiment showed that almost all of the words classified in (3) are the top three words in $I(w_j)$, i.e., $I(w) \leq 22.21$. Therefore, we can see that a large number of KW_i does not necessarily mean strong privacy protection, and re-identification attacks can be sufficiently prevented even if $|KW_i|$ is not large. In general, when people search the web, the number of keywords is about 3 at most, so $|KW_i| = 3$ is reasonable.

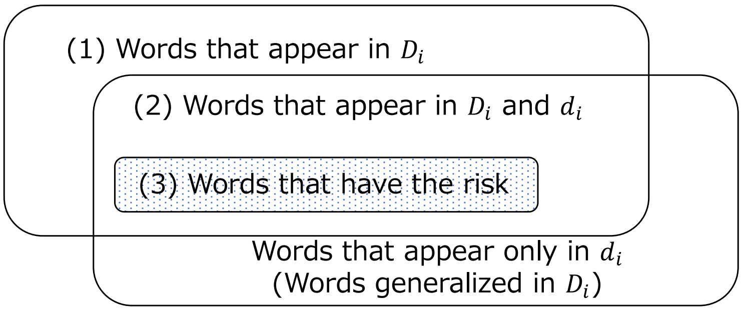

(1) Words that appear in D_i
(2) Words that appear in D_i and d_i
(3) Words that have the risk
Words that appear only in d_i
(Words generalized in D_i)

Figure 6.2: Relationship of each word class.

In the case of court documents, we collected sanitized documents on the web, but due to contractual issues, we could not obtain the original documents D. This is not a special case, and a sanitizer needs to define $RiskEvaluation$ carefully. We perform an experiment to see if the algorithm can successfully attack SCDs. As mentioned above, we set d for the 30 SCDs related to the murder, D^{**} for the other 1,000 SCDs, and R_i for the name of the person associated with d_i. We also set $KW_i = 3$ so that the algorithm would select three words from each document as the words of class (2). $WebSearchAttack$ searches the articles in $List_i$ using these words. The words in $List_i$ were compared with R_i, and the words linked to the corresponding person were assigned to class (3). The results are shown in Table 6.2. Informative words have a higher risk of being identified, and risk words have been identified in documents that have been manually sanitized as well as in school accident documents.

Note that the parameter $\epsilon(w_j)$ was changed from the previous experiment due to the difference between accident documents and court documents. One major change was an adjustment to reduce the priority of names of people, since OADs emphasize names of people, but in SCDs, court documents include names of judges. Furthermore, the importance of age in SCDs was in-

Table 6.2: Relationship between $I(w_j)$ and the class of words (II).

$I(w_j)$	Class (1)	Class (2)	Class (3)
31.50	–	41	14
31.21	–	28	8
30.83	–	15	0
#PRWs	–	84	22

Table 6.3: Labeling result.

	Positive	Negative
True	650	646
False	10	6

creased. All of the victims in the school accident documents were students, and even if their ages were included, this information would not be of much value from a re-identification perspective. On the other hand, the age of suspects and victims in articles might be effective identifiers of the relevant person. Thus, some tuning is necessary for different types of documents, and the sanitizer can easily optimize the policy of parameter setting d.

6.2.3 Expansion

`Preprocessing` provides a labeling option. Since the attack accuracy depends on the interest of the target document, we assign an interest label to the target document. In this section, we check the effectiveness of the labeling option. We use FastText [JGB+16], which is published as open source by Facebook AI Research, to classify documents. FastText handles words as a vector, as does Word2Vec [Chu17], and classifies documents at high speed. Of our 2,300 SADs, we set half of them as training data. We assigned two labels to the training data, fatal and non-fatal accidents, and inferred the labels for the other half of the documents. The results (Table 6.3) show that we could classify with high accuracy whether the documents were related to fatal or non-fatal accidents. In our experiments, we have confirmed that attacks on non-fatal accidental documents do not succeed, and $SetKeywords$ was able to change the parameter KW_i by labels. The labeling mechanism varies depending on the type of document, but introducing variable parameters based on the label is expected to improve the utility and processing speed of the algorithm.

In this chapter, we generated an evaluation function that links victim names to corresponding documents in advance, and outputs 1 if the function finds the relevant name in the retrieved document, and 0 otherwise. However, this function can be generated by a sanitizer who knows the victim's name, i.e., when the sanitizer and the data holder are the same, and we need to consider the case where the sanitizer does not know this name. This case is common. For example, if the data owner does not have sanitization skills, the owner may sign a contract with an organization that does. In this case, the data owner can remove the identifier before sending the data set to the proxy sanitizer. Therefore, it is necessary to consider a more flexible evaluation function. In other words, it is necessary to create an index for privacy risk of documents without using the victim's name. As a way to deal with this problem, we focus on metadata. Let w be the number of metadata that are also present in the retrieved documents. Intuitively, the larger w is, the more likely it is that the retrieved document and the actual document show the same content. We manually checked this relationship between w and risk. In the experiment, we considered all documents to be 3,000 SADs, d, and applied $WebSearchAttack$ to the data, focusing on documents that satisfy $w \geq 4$. The horizontal and vertical axes in Fig. 6.3 represent w and `Attack` success probability, respectively. The attack here, as before, is to check whether the retrieved documents contain personal names. For example, we can see that 50% of the documents with $w = 16$ are at risk of being identified, and that there are four documents with $w = 16$ that have been successfully attacked. It is worth mentioning here that as w increases, the probability of a successful attack increases, supporting the hypothesis that there is a correlation between w and risk. In other words, if a document contains many of the same words that are found in another document, it is likely that the documents represent the same thing. The results also show that the number of w may be used for privacy risk assessment. If the sanitizer does not know the correct answer to the data to be evaluated, in this case the victim's name, the following response is possible. $RiskEvaluation$ outputs w instead of 0 or 1, and if w exceeds the threshold, it either moves to $Reconstruct$ or aborts and rejects the attack and immediately outputs the document. In addition, $WebSearchAttack$ and $RiskEvaluation$ are not independent and run alternately in actual operation, and $WebSearchAttack$ attacks a document using $kw \subseteq KW_i$ is the bottleneck. Therefore, in order to improve the processing speed, the idea can be applied.

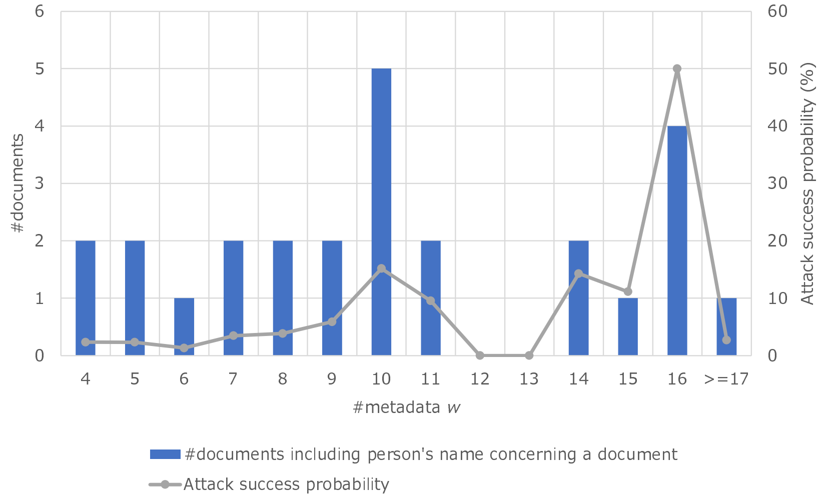


Figure 6.3: Relationship between the number of w and attack success probability.

6.3 Conclusion

We proposed a document sanitization framework and applied it to public accident reports from schools and court documents. First, based on the generalized adversarial model for re-identification, we applied it to the privacy problem on documents. As in Chapters 4 and 5, we assume that the privacy index is the personal identification probability and the attacker treats the results of web search as external knowledge. This eliminated the need to define in advance what is sensitive information, and thus reduced the cost compared to the previous approach. In addition, since there is no need to calculate PMI, it can be used for long documents such as court documents.

We considered that web search engines are used for simulation attacks, implemented an attack algorithm based on the model, and confirmed in experiments that attacks on sanitized documents actually occur. As a result, we found that there are still documents that are at risk of being re-identified even after manual sanitization, proving that manual sanitization may not be sufficient to defeat attackers who use web search engines.

Furthermore, we proposed a sanitization algorithm for this attack, which accurately sanitizes or generalizes only the words that cause re-identification. The situation assumed in our experiments is realistic, and our framework can be applied even when documents do not have metadata, by setting informative words as metadata.

Chapter 7

Differential Privacy Mechanism

We have discussed privacy metrics based on k-anonymity as metrics that are easy for data subjects to understand. However, these are mainly semi-honest models and cannot guarantee privacy when data is passed on to malicious users. Therefore, we believe that information-theoretic privacy-preserving techniques that maintain their utility will be necessary in the future. At present, differential privacy is one of the most promising privacy-preserving techniques. In this chapter, we discuss the differential privacy mechanism to develop the research. In particular, we discuss sensitivity, which is necessary in proposing a differential privacy mechanism, and focus on t-tests as a case study. (For the t-test, see Chapter 3.)

7.1 Differential Privacy under Incalculable Sensitivity

7.1.1 Queries for which sensitivity is difficult to calculate

We consider queries for which the derivation of sensitivity is difficult. As in Definitions 2.5 and 2.6, sensitivity is the difference between the output values for a query on any neighboring dataset D, D', which is the noise criterion. Considering an arbitrary dataset leads to a guarantee of the privacy of all records. Now, we consider the t-value again, the denominator is $\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}$. However, if we assume that both A and B contain records with all the same values, then $s_A^2 = s_B^2 = 0$, and the t-value cannot be calculated. It is also impossible to calculate $S_{q,\beta}(D)$ in smooth sensitivity, where the dataset is given as input, since there exists D' such that $s_m^2 = 0$. Thus, when we consider arbitrary datasets, we can see that it is difficult to calculate sensitivity for queries that are undefined or divergent.

7.1.2 Differential privacy definition with dummy data

The reason why sensitivity is difficult to derive is that we need to consider arbitrary datasets. On the other hand, if we do not consider a particular record, we cannot obtain a privacy guarantee for that record. Therefore, we define differential privacy by adding dummy data, which does not require privacy guarantee.

Definition 7.1 ((ϵ', δ')-differential privacy with dummy data)**.** *A mechanism M^r satisfies (ϵ', δ')-differential privacy (DP) with dummy data if it satisfies the following for any neighboring datasets D, D' and $r = \{r_1, \cdots, r_w\}$.*

$$Pr[M^r(D \cup r) \in \mathcal{D}] \leq e^{\epsilon'} \cdot Pr[M^r(D' \cup r) \in \mathcal{D}] + \delta' \tag{7.1}$$

Similarly, we define the global sensitivity with dummy data.

Definition 7.2 (global sensitivity with dummy data)**.** *The global sensitivity GS^r_q with dummy data for query $q : \mathcal{D}^{|\mathcal{X}|} \to \mathbb{R}^k$ is defined as follows.*

$$GS^r_q = \max_{\forall D, D' : H(D, D')=1} \|q(D \cup r) - q(D' \cup r)\|_p \tag{7.2}$$

In Definitions 7.1 and 7.2, we guarantee the privacy of any data in D, but not for r. However, since r is dummy data, there is no privacy problem. In addition, the conventional differential privacy mechanism satisfies Definitions 7.1 and 7.2. As a concrete example, we consider the Laplace mechanism.

Theorem 7.1. *Let $q : \mathcal{D}^{|\mathcal{X}|} \to \mathbb{R}^k$ be the query and $L(\mu, v)$ be the Laplacian noise with mean μ and variance v, then the following mechanism M^r_q satisfies $(\epsilon', 0)$-DP with dummy data.*

$$M^r_q(D \cup r) = q(D \cup r) + L\left(0, \frac{GS^r_q}{\epsilon'}\right) \tag{7.3}$$

Proof. Assume $D \cup r$ and $D' \cup r$ for neighboring dataset D and D'. Let $q : \mathcal{D}^{|\mathcal{X}|} \to \mathbb{R}^k$ be the query. If p_x is the probability density function of $M^r_q(x)$, then for any $z \in \mathbb{R}^k$ the following holds.

$$\begin{aligned}
\frac{p_{(D \cup r)}(z)}{p_{(D' \cup r)}(z)} &= \prod_{i=1}^{k} \frac{\exp(-\frac{\epsilon'|q(D \cup r)_i - z_i|}{GS^r_q})}{\exp(-\frac{\epsilon'|q(D' \cup r)_i - z_i|}{GS^r_q})} \\
&= \prod_{i=1}^{k} \exp\left(\frac{\epsilon'(|q(D' \cup r)_i - z_i| - |q(D \cup r)_i - z_i|)}{GS^r_q}\right) \\
&\leq \prod_{i=1}^{k} \exp\left(\frac{\epsilon'|q(D' \cup r)_i - q(D \cup r)_i|}{GS^r_q}\right) \\
&= \exp\left(\frac{\epsilon'||q(D' \cup r) - q(D \cup r)||_1}{GS^r_q}\right) \\
&\leq \exp(\epsilon').
\end{aligned} \tag{7.4}$$

We can consider the symmetry for $\frac{p_{(D \cup r)}}{p_{(D' \cup r)}} \geq \exp(-\epsilon')$. □

As can be seen from the proof, the only difference between Definitions 7.1 and 7.2 and Definitions 2.5 and 2.8 is the consideration of arbitrary data, which does not affect the rest of the calculation results. In Definitions 7.1 and 7.2, the privacy of r is not guaranteed, but since

r is dummy data, it does not affect the privacy. Similarly, smooth sensitivity with dummy data can be defined as Definitions 7.3 and 7.4.

Definition 7.3. *Assume dataset $D \cup r$ is given. The local sensitivity LS^r_q with dummy data for query $q : \mathcal{D}^{|\mathcal{X}|} \to \mathbb{R}^k$ is defined as follows.*

$$LS^r_q(D \cup r) = \max_{\forall D' : H(D,D')=1} \|q(D \cup r) - q(D' \cup r)\|_p \tag{7.5}$$

Definition 7.4. *Assume $\beta > 0$ and the dataset $D \cup r$ are given. The smooth sensitivity S^r_q with dummy data for the query $q : \mathcal{D}^{|\mathcal{X}|} \to \mathbb{R}^k$ is defined as follows.*

$$S^r_{q,\beta}(D \cup r) = \max_{anyD'} (LS^r_q(D' \cup r) \cdot e^{-\beta H(D,D')}) \tag{7.6}$$

Theorem 7.2. *Let query $q : \mathcal{D}^{|\mathcal{X}|} \to \mathbb{R}^k$. When $\alpha = \frac{\epsilon'}{5\sqrt{2\ln 2/\delta'}}$ and $\beta = \frac{\epsilon'}{4(k+\ln 2/\delta')}$, the following mechanism satisfies (ϵ', δ')-DP with dummy data.*

$$M_q(D \cup r) = q(D \cup r) + \frac{S^r_{q,\beta}(D' \cup r)}{\alpha} \cdot N(0,1) \tag{7.7}$$

As shown in [NRS07], in order to prove Theorem 2.2, we need to consider a data set where $H(D, D') = 1$. Here, $H(D, D') = H(D \cup r, D' \cup r)$ and it allows us to prove Theorem 7.2 as well as Theorem 2.2. The only difference in the proof from [NRS07] is that the record r in $D \cup r$ is fixed, so the privacy of D, which is arbitrary data, is guaranteed. At this time, the privacy of r is not guaranteed, but since r is dummy data, it has no effect on each data.

7.1.3 Differential private t-test with dummy data

We use Definitions 7.3 and 7.4 and Theorem 7.2 to construct a t-test mechanism that satisfies differential privacy. In the following, we denote the value corresponding to D in D' by $()'$. For example, the set corresponding to A in D' is denoted by $(A)'$ and the mean value of $(A)'$ is denoted by $(\mu_A)'$.

Theorem 7.3. *The local sensitivity with dummy data to the query q for t-value is obtained by the following equation.*

$$LS^r_t(D) \leq \max\left(2C_1 \cdot \frac{|\mu_A - \mu_B| + m/\min(n_A, n_B)}{\sqrt{s^2_m}}, C_2 \cdot \frac{|\mu_A - \mu_B| + 2m/\min(n_A, n_B)}{\sqrt{s^2_m}}\right), \tag{7.8}$$

where $C_1 = \sqrt{\frac{n_A n_B}{n_A + n_B}}$, $C_2 = \frac{\sqrt{\frac{n_A+n_B}{(n_A-1)(n_B+1)}} + \sqrt{\frac{n_A+n_B}{n_A n_B}}}{\sqrt{\frac{n_A+n_B}{(n_A-1)(n_B+1)} \cdot \frac{n_A+n_B}{n_A n_B}}}$. Also $s^2_m = \min(s^2_A, s^2_B, (s^2_A)', (s^2_B)')$.

Proof. Assume D and its neighbor D'. There are two possible cases in which different records belong to the same set or different sets, i.e., $(|(A)'|, |(B)'|) = ((n_A)', (n_B)') = \{(n_A, n_B), (n_A - 1, n_B + 1)\}$.

First, we consider $(|(A)'|, |(B)'|) = (n_A, n_B)$. It does not lose generality if we consider a_{n_A} and $(a_{n_A})'$ to be different.

The t-value in D' is obtained by the following equation.

$$(t)' = \frac{(\mu_A)' - (\mu_B)'}{\sqrt{\frac{(s_A^2)'}{n_A} + \frac{(s_B^2)'}{n_B}}}. \tag{7.9}$$

From $(B)' = B$, we have $(\mu_B)' = \mu_B, (s_B^2)' = s_B^2$. If we denote $(\mu_A)' - \mu_A = \Delta\mu_A, (s_A^2)' - s_A^2 = \Delta s_A^2$, then the difference between t-values of neighboring datasets is expressed as

$$\begin{aligned} |(t)' - t| &= \left| \frac{(\mu_A)' - (\mu_B)'}{\sqrt{\frac{(s_A^2)'}{n_A} + \frac{(s_B^2)'}{n_B}}} - \frac{\mu_A - \mu_B}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}} \right| \\ &= \left| \frac{\mu_A + \Delta\mu_A - \mu_B}{\sqrt{\frac{s_A^2 + \Delta s_A^2}{n_A} + \frac{s_B^2}{n_B}}} - \frac{\mu_A - \mu_B}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}} \right|. \end{aligned} \tag{7.10}$$

$|(t)' - t|$ is maximal when the first term in Eq. (7.10) is maximal and the second term is minimal. We denote $\mu_M = \max(\mu_A, \mu_B, (\mu_A)'), \mu_m = \min(\mu_A, \mu_B, (\mu_A)')$ and

$$\begin{aligned} \max(|(t)' - t|) &\leq \frac{\mu_M - \mu_m}{\sqrt{\frac{s_m^2}{n_A} + \frac{s_m^2}{n_B}}} - \frac{\mu_m - \mu_M}{\sqrt{\frac{s_m^2}{n_A} + \frac{s_m^2}{n_B}}} \\ &= 2 \cdot \frac{\mu_M - \mu_m}{\sqrt{\frac{n_A + n_B}{n_A n_B} \cdot s_m^2}} \\ &= 2C_1 \cdot \frac{\mu_M - \mu_m}{\sqrt{s_m^2}}. \end{aligned} \tag{7.11}$$

Then we have

$$\mu_M - \mu_m \leq |\mu_A - \mu_B| + |\Delta\mu_A| + |\Delta\mu_B|. \tag{7.12}$$

Since $\Delta\mu_B = 0$ here, we only need to consider $|\Delta\mu_A|$, where $|\Delta\mu_A| \leq m/n_A$. Therefore, the following results can be obtained.

$$\max(|(t)' - t|) \leq 2C_1 \cdot \frac{|\mu_A - \mu_B| + m/n_A}{\sqrt{s_m^2}}. \tag{7.13}$$

Next, we consider $(|(A)'|, |(B)'|) = (n_A - 1, n_B + 1)$. It does not lose generality if we consider a_{n_A} and b_{n_B+1} are different in D and D'.

We have

$$(t)' = \frac{(\mu_A)' - (\mu_B)'}{\sqrt{\frac{(s_A^2)'}{n_A - 1} + \frac{(s_B^2)'}{n_B + 1}}}. \tag{7.14}$$

and the difference of t-values in neighboring data sets is obtained by

$$\begin{aligned}|(t)'-t| &= \left|\frac{(\mu_A)'-(\mu_B)'}{\sqrt{\frac{(s_A^2)'}{n_A-1}+\frac{(s_B^2)'}{n_B+1}}}-\frac{\mu_A-\mu_B}{\sqrt{\frac{s_A^2}{n_A}+\frac{s_B^2}{n_B}}}\right| \\ &= \left|\frac{\mu_A+\Delta\mu_A-(\mu_B+\Delta\mu_B)}{\sqrt{\frac{(s_A^2)'}{n_A-1}+\frac{(s_B^2)'}{n_B+1}}}-\frac{\mu_A-\mu_B}{\sqrt{\frac{s_A^2}{n_A}+\frac{s_B^2}{n_B}}}\right|.\end{aligned} \tag{7.15}$$

$|(t)'-t|$ is maximal when the first term of Eq. (7.15) is maximal and the second term is minimal. If we assume $s_m^2=\min(s_A^2,s_B^2,(s_A^2)',(s_B^2)')$, we have the following equation.

$$\begin{aligned}\max(|(t)'-t|) &\le \frac{\mu_M-\mu_m}{\sqrt{\frac{s_m^2}{n_A-1}+\frac{s_m^2}{n_B+1}}}-\frac{\mu_m-\mu_M}{\sqrt{\frac{s_m^2}{n_A}+\frac{s_m^2}{n_B}}} \\ &= \frac{\sqrt{\frac{n_A+n_B}{(n_A-1)(n_B+1)}}+\sqrt{\frac{n_A+n_B}{n_An_B}}}{\sqrt{\frac{n_A+n_B}{(n_A-1)(n_B+1)}\cdot\frac{n_A+n_B}{n_An_B}\cdot s_m^2}}\cdot(\mu_M-\mu_m) \\ &= C_2\cdot\frac{\mu_M-\mu_m}{\sqrt{s_m^2}}.\end{aligned} \tag{7.16}$$

Now we consider $|\mu_M-\mu_m|$.

$$\begin{aligned}|\mu_M-\mu_m| &\le |\mu_A-\mu_B|+|\Delta\mu_A|+|\Delta\mu_B| \\ &\le |\mu_A-\mu_B|+2m/\min(n_A,n_B).\end{aligned} \tag{7.17}$$

Then,

$$\max(|(t)'-t|)\le C_2\cdot\frac{|\mu_A-\mu_B|+2m/\min(n_A,n_B)}{\sqrt{s_m^2}}. \tag{7.18}$$

□

From Theorem 7.3, we can show the upper bound of local sensitibity with dummy data for an input dataset D if we can obtain s_m^2. The s_m^2 depends on whether D and the neighboring D' contain different records in the same set.

Theorem 7.4. *In D and the neighboring D', $s_m^2=\min((s_A^2)',(s_B^2)')$. $(s_A^2)'$ is given by*

$$(s_A^2)'=s_A^2-\max\left(\frac{n_A}{(n_A-1)^2}\cdot d_{n_A}^2,\frac{1}{(n_A-1)(n_A-2)}\cdot\left\{-(n_A-1)\cdot d_{n_A}^2+\sum_{i=1}^{n_A-1}(\mu_A-a_i)^2\right\}\right), \tag{7.19}$$

where $d_{n_A}=\max(|\mu_A-a_i|)$. $(s_B^2)'$ is given by

$$(s_B^2)'=s_B^2-\frac{1}{n_B(n_B-1)}\sum_{i=1}^{n_B}(\mu_B-b_i)^2. \tag{7.20}$$

Proof. We first consider the case of $(|(A)'|,|(B)'|) = (n_A, n_B)$. It does not lose generality if we consider a_{n_A} and $(a_{n_A})'$ to be different.

We consider Δs_A^2. Since Δs_A^2 is the difference of the unbiased variances of the A-groups of D, D', we have

$$\begin{aligned}\Delta s_A^2 &= (s_A^2)' - s_A^2\\ &= \frac{1}{n_A-1}\cdot\left\{\sum_{i=1}^{n_A-1}(\mu_A+\Delta\mu_A-a_i)^2+(\mu_A+\Delta\mu_A-(a_{n_A}+\Delta a_{n_A}))^2\right\}\\ &-\frac{1}{n_A-1}\cdot\left\{\sum_{i=1}^{n_A-1}(\mu_A-a_i)^2+(\mu_A-a_{n_A})^2\right\}\\ &= \frac{1}{n_A-1}\cdot\left\{2(a_{n_A}-\mu_A)\Delta\mu_A+(n_A-1)(\Delta\mu_A)^2+2(\mu_A-a_{n_A})(\Delta\mu_A-\Delta a_{n_A})\right.\\ &\left.+(\Delta\mu_A-\Delta a_{n_A})^2\right\}.\end{aligned} \tag{7.21}$$

Here, from $\Delta\mu_A = \frac{\Delta a_{n_A}}{n_A}$, we get the following equation.

$$\Delta s_A^2 = \frac{1}{n_A-1}\cdot\left\{\frac{n_A-1}{n_A}\cdot(\Delta a_{n_A})^2+2(a_{n_A}-\mu_A)\cdot\Delta a_{n_A}\right\} \tag{7.22}$$

Δs_A^2 is a quadratic function of Δa_{n_A} and is convex below. The Δs_A^2 is minimized when $(\Delta a_{n_A}, \Delta s_A^2) = (\frac{n_A(\mu_A-a_{n_A})}{n_A-1}, -\frac{n_A}{(n_A-1)^2}(a_{n_A}-\mu_A)^2)$ and a_{n_A} is the value at which $|\mu_A - a_{n_A}|$ is maximized, namely $a_{n_A} = \arg\max_{a_i}(|\mu_A - a_i|)$. Furthermore, since $\Delta a_{n_A} = \frac{n_A(\mu_A-a_{n_A})}{n_A-1}$, we have $(a_{n_A})' = \frac{n_A}{n_A-1}\cdot\mu_A - \frac{1}{n_A-1}\cdot a_{n_A}$. $a_{n_A} > \mu_A$ should be considered symmetric.

Next, we consider the case of $(|(A)'|,|(B)'|) = (n_A-1, n_B+1)$. Even if a_{n_A} and b_{n_B+1} are considered differently, they do not lose generality. We consider Δs_A^2.

$$\begin{aligned}\Delta s_A^2 &= (s_A^2)' - s_A^2\\ &= \frac{1}{n_A-2}\cdot\left\{\sum_{i=1}^{n_A-1}(\mu_A+\Delta\mu_A-a_i)^2\right\}\\ &-\frac{1}{n_A-1}\cdot\left\{\sum_{i=1}^{n_A-1}(\mu_A-a_i)^2+(\mu_A-a_{n_A})^2\right\}\\ &= \frac{1}{(n_A-1)(n_A-2)}\cdot\left\{\sum_{i=1}^{n_A-1}(\mu_A-a_i)^2+2(n_A-1)(a_{n_A}-\mu_A)\cdot\Delta\mu_A\right.\\ &\left.\qquad+(n_A-1)^2\cdot(\Delta\mu_A)^2-(n_A-2)(\mu_A-a_{n_A})^2\right\}.\end{aligned} \tag{7.23}$$

Since $\Delta\mu_A = \frac{1}{n_A-1}\sum_{i=1}^{n_A-1}a_i - \frac{1}{n_A}(\sum_{i=1}^{n_A-1}a_i + a_{n_A}) = \frac{1}{n_A(n_A-1)}\sum_{i=1}^{n_A-1}a_i - \frac{a_{n_A}}{n_A} = \frac{\mu_A-a_{n_A}}{n_A-1}$, we have

$$\begin{aligned}\Delta s_A^2 &= (s_A^2)' - s_A^2\\ &= \frac{1}{(n_A-1)(n_A-2)} \cdot \left\{ \sum_{i=1}^{n_A-1} (\mu_A - a_i)^2 + 2(n_A-1)(a_{n_A} - \mu_A) \cdot \Delta\mu_A \right.\\ &\qquad\qquad \left. + (n_A-1)^2 \cdot (\Delta\mu_A)^2 - (n_A-2)(\mu_A - a_{n_A})^2 \right\}\\ &= \frac{1}{(n_A-1)(n_A-2)} \cdot \left\{ -(n_A-1) \cdot (\mu_A - a_{n_A})^2 + \sum_{i=1}^{n_A-1} (\mu_A - a_i)^2 \right\}.\end{aligned} \tag{7.24}$$

Δs_A^2 is a quadratic function of a_{n_A} and is upwardly convex and is maximized when $(a_{n_A}, \Delta s_A^2) = (\mu_A, \sum_{i=1}^{n_A-1}(\mu_A - a_i)^2)$. Therefore, Δs_A^2 is minimized when a_{n_A}, where $|\mu_A - a_{n_A}|$ is maximized, and b_{n_B+1} are different in the neighboring datasets D, D'. In this case, $s_m^2 = \min((s_A^2)', (s_B^2)')$.

Similarly, considering the difference of unbiased variances of $B, (B)'$, we get the following equation.

$$\begin{aligned}\Delta s_B^2 &= (s_B^2)' - s_B^2\\ &= \frac{1}{n_B} \cdot \left\{ \sum_{i=1}^{n_B} (\mu_B + \Delta\mu_B - b_i)^2 + (\mu_B + \Delta\mu_B - b_{n_B+1})^2 \right\}\\ &- \frac{1}{n_B - 1} \cdot \left\{ \sum_{i=1}^{n_B} (\mu_B - b_i)^2 \right\}\\ &= \frac{1}{n_B(n_B-1)} \cdot \left\{ (n_B+1)(n_B-1) \cdot (\Delta\mu_B)^2 + 2(n_B-1)(\mu_B - b_{n_B+1}) \cdot \Delta\mu_B \right.\\ &\qquad\qquad \left. + (n_B-1)(\mu_B - b_{n_B+1})^2 - \sum_{i=1}^{n_B} (\mu_B - b_i)^2 \right\}.\end{aligned} \tag{7.25}$$

Since $\Delta\mu_B = \frac{1}{n_B+1}(\sum_{i=1}^{n_B} b_i + b_{n_B+1}) - \frac{1}{n_B}\sum_{i=1}^{n_B} b_i = \frac{b_{n_B+1} - \mu_B}{n_B+1}$, we have

$$\begin{aligned}\Delta s_B^2 &= (s_B^2)' - s_B^2\\ &= \frac{1}{n_B(n_B-1)} \cdot \left\{ (n_B+1)(n_B-1) \cdot (\Delta\mu_B)^2 + 2(n_B-1)(\mu_B - b_{n_B+1}) \cdot \Delta\mu_B \right.\\ &\qquad\qquad \left. + (n_B-1)(\mu_B - b_{n_B+1})^2 - \sum_{i=1}^{n_B} (\mu_B - b_i)^2 \right\}\\ &= \frac{1}{n_B(n_B-1)} \cdot \left\{ \frac{n_B(n_B-1)}{n_B+1} \cdot (\mu_B - b_{n_B+1})^2 - \sum_{i=1}^{n_B} (\mu_B - b_i)^2 \right\}.\end{aligned} \tag{7.26}$$

Δs_B^2 is a quadratic function of b_{n_B+1} and is convex below. In this case, the vertex is $(b_{n_B}, \Delta s_B^2) = (\mu_A, -\frac{1}{n_B(n_B-1)} \cdot \sum_{i=1}^{n_B}(\mu_B - b_i)^2)$. Thus, when a_{n_A} and $b_{n_B+1} = \mu_B$ are different in the neighboring dataset D, D', $(s_B^2)' = s_B^2 - \sum_{i=1}^{n_B}(\mu_B - b_i)^2$, and we can say $s_m^2 = \min((s_A^2)', (s_B^2)')$. □

When we consider queries for the p-value, we also need to consider the degrees of freedom. If we consider df as a bivariate function with $X = s_A^2/n_A$ and $Y = s_B^2/n_B$, then df has a maximum difference Δdf when $(s_A^2)', (s_B^2)'$ is minimum or maximum, and these can be easily obtained from Theorem 7.4 . If B is the beta function, sensitivity can be derived by finding the difference of $p = 2\int_t^{\infty} f_{df}(t)$, where $f_{df}(t) = \frac{1}{\sqrt{df}B(df/2, 1/2)} \cdot (1 + t^2/df)^{-\frac{df+1}{2}}$ is the probability density function of the t-distribution of the degrees of freedom df.

Next, we consider smooth sensitivity with dummy data for the t-value. We need to consider the local sensitivity with dummy data for an arbitrary $D'_k : H(D, D'_k) = k$. We consider the neighboring D' such that $\Delta s_A^2 = (s_A^2)' - s_A^2, \Delta s_B^2 = (s_B^2)' - s_B^2$ is minimized for D. In this case, the unbiased variance of the set A', B' of D' is minimized. Therefore, the unbiased variance in D'_k is minimized when k changes of records that minimize $\Delta s_A^2, \Delta s_B^2$ for D are recursively performed. From the above, when smooth sensitivity with dummy data is used for t-test, neighboring datasets that minimize s_m^2 for D are recursively constructed $n_A + n_B$ times, and $LS_q^r(D') \cdot e^{-\beta H(D, D')}$ is calculated for each execution. The maximum value should be $S_{q,\beta}^r$.

Therefore, the following algorithm can be used to perform a t-test satisfying the differential privacy. Note that, we do not perform the r_1, r_2 operation in lines 4-7.

Algorithm 8 $(D, \epsilon', r_1, r_2(\neq r_1))$: Differential private t-test mechanism with dummy data

```
Input: dataset $D = A \cup B$, privacy parameter $\epsilon'$, and random records $r_1, r_2$
1: $D_0 \leftarrow D$
2: $A \leftarrow A \cup \{r_1, r_2\}$
3: $B \leftarrow B \cup \{r_1, r_2\}$
4: for $i = 0; i < n_A + n_B; i++$ do
5:     Generate $D_{i+1}$, whose $(s_A^2)'$ or $(s_B^2)'$ is equal to $s_m^2$
6:     Calculate $s_m^2$ and $LS_q^r(D_i) \cdot e^{-\beta H(D_0, D_i)}$
7: end for
8: $S_{q,\beta}^r(D) = \max(LS_q^r(D_i) \cdot e^{-\beta H(D_0, D_i)})$
9: return $M_q^r(D) = q(D) + \frac{S_{q,\beta}^r(D)}{\alpha} \cdot N(0, 1)$
```

7.2 Experiment

We generated a dataset and evaluated the magnitude of the given variance $\frac{S_{q,\beta}^r(D)}{\alpha}$. If $n = n_A + n_B$ and the unbiased variance of the dataset is s^2, the variance of the added noise is $O(\frac{m}{\epsilon' \cdot \sqrt{n \cdot s^2}})$. We fixed $m = 200, \epsilon' = 5$ and evaluated the variance with n, s^2 as variables. The evaluation results show that, as in the case of a direct t-test using the local differential privacy mechanism [DNLA18] used in [DKY17], a rather large noise is added. Therefore, the probability of type-I error will be high.

Figure 7.1 shows the probability of the type-I error when $n = 10000, m = 200$, and $s_D^2 = 200$ are fixed and ϵ' is a variable. As the number and variance of records and the privacy parameter

Table 7.1: Variance $(\epsilon', \mu_d) = (5, 0)$.

#records	s_D^2		
n	20	200	2000
100	256.04	33.07	10.32
1000	78.72	13.06	7.23
10000	30.33	8.98	7.70

increase, the variance of the noise decreases, but a significant number of samples and variance are required to keep the type I error below α.

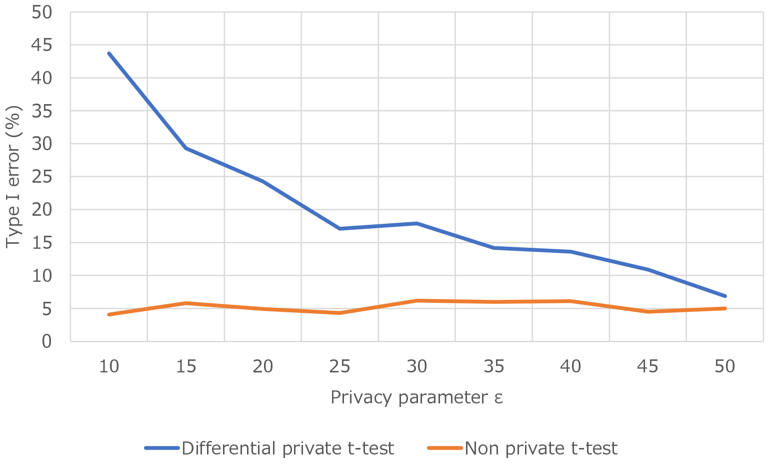


Figure 7.1: The probability of type I error.

7.3 Conclusion

In this chapter, we have shown that the construction of a probabilistic mechanism that satisfies differential privacy is complicated even for a simple analysis such as a t-test, and furthermore, a large amount of data is required in order to obtain correct analysis results. In the t-test, which is the case study, it is necessary to take measures such as replacing the analysis process with another problem as shown in [DNLA18], and even then, a large amount of data is still required for the correct analysis. We can say that differential privacy is effective in capturing the general trend of the entire data, but it is not suitable for capturing the detailed characteristics of the data.

On the other hand, our main contribution in this chapter is that we have shown that by adding dummy data to the data set, it is possible to construct a probabilistic mechanism even in analyses where it is difficult to derive sensitivity in a straightforward manner.

We suppose that the privacy parameters ϵ' in our definition do not strictly correspond to the privacy parameters ϵ in conventional differential privacy. This is because of the following reasons. Adding dummy data to the dataset reduces the impact, i.e., sensitivity, of arbitrary data on the output results of the query. However, we do not mention the privacy of the dummy data, and if we assume that the attacker has the dummy data as background knowledge, the privacy in our definition will naturally be weaker than the traditional differential privacy.

If we consider the addition of dummy data as a kind of de-identification method and assume that the added dummy data is indistinguishable from the actual data, we can quantitatively evaluate the impact of adding dummy data on privacy by clarifying the relationship between ϵ' and ϵ.

Chapter 8

Secure Computation

Although secure computation does not directly process data and protect privacy, the use of encryption technology allows for the outsourcing of analysis without exposing the data. Thus, secure computation is an important technology for making effective use of privacy data, and it is indispensable for achieving both privacy and utility. In this chapter we propose a very efficient two-party comparison protocol. This is a basic protocol and can be a component of any protocol, including machine learning. Furthermore, by using the existing techniques in Chapters 2 and 3, any type of comparison protocol can be constructed.

8.1 Comparison Protocol

8.1.1 Conversions

We have described the types of comparison protocols and their conversion in Chapters 2 and 3. There are 216 comparison protocols ($6^3 = 216$) due to the differences in the input and output of Alice and Bob.

A few of the protocol types are trivial to construct in the sense that output value δ can be computed (by either Alice or Bob) without any interaction (and, as a result, carry no practical meaning of secure computation). We list these types in the following lemma.

Lemma 8.1. *Consider any choice of DD $\in \mathcal{S}$. We have that PN-PN-DD, PP-PN-DD, PN-PP-DD, NP-NP-DD, PP-NP-DD, NP-PP-DD, PP-PP-DD are trivial.*

The proof of the lemma is straightforward. For example, PN-PN-DD is trivial since Alice knows both x, y. The other cases can be argued similarly.

From the Lemmata 2.1, 3.1, and 3.3, we can construct any of 174 non-trivial protocol types. We will describe efficient constructions for Type 4 and Type 5 in the following section.

8.1.2 Tree-based structure

We describe two protocols originally proposed by Nergiz et al. [NNPC10]. Protocol 1, which is a Type 5 protocol in our categorization, is exactly the same as the main protocol in [NNPC10].

Protocol 2, which is a type-4 protocol in our categorization, is a slight variant of the extended protocol in [NNPC10], of which outputs are shared data; the difference is that in the extended protocol in [NNPC10], the sharing is done modulo 2, while in our setting, the sharing is done modulo N.

The main technique in these protocols is to use a classical method called dyadic range which expresses a range (i.e., a set of consecutive integers) efficiently using nodes in a complete binary tree[1].

We describe some intuition first. A naive way to compare two integers is to represent both values as binary strings and use the standard Boolean circuit for comparison, which firstly compares the most significant bit, and if equal, simply compares the next bit, and so on. This means that it is required that each bit is processed in a *sequential* manner. On the other hand, in the dyadic range method, integers are represented in such a way that a comparison can be done by comparing each bit in an *independent* manner (and simply OR all the results). Contrasting the two methods, the latter has much shallower circuit; this makes it more efficient than the former. Moreover, in the latter method, only secure equality checking will be required as a sub-routine, and we can straightforwardly use additive homomorphic encryption to implement a secure protocol for this.

We firstly describe our own terminology and notations for tree-based structure.

Let $\mathbb{T}_n$ be the complete binary tree that has leaves corresponding to each index in $[1, n]$. Let $\mathbb{S}_n$ be the set of all nodes in $\mathbb{T}_n$ that are labeled in a systematic way. For a node $w \in \mathbb{S}_n$, let parent(w) denote its parent node in $\mathbb{T}_n$. Consider node $w, y, z \in \mathbb{S}_n$; z is an *ancestor* of w if z is on the path from w to the root (including w); y is a *descendant* of w if y is on a path from w moving away from the root (including w). For any node w, we define its *layer* as the distance from its leaves. (Hence, in particular, the layer of any leaf is 0, and the layer of the root is $\log_2 n$.) We label each node as a pair (i, j) where i is its layer and j is its number in that layer from the left of tree (starting from 1). See how we label each node in e.g., Fig. 8.1, where we omit the comma in the figure.

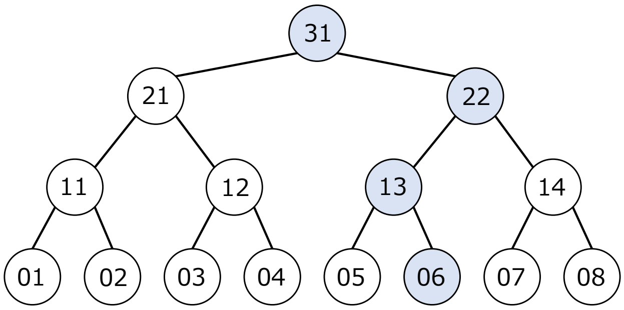


Figure 8.1: Example of Point Encoding, for $x = 6$. Here, $\mathsf{pointEnc}(6) = \{(0, 6), (1, 3), (2, 2), (3, 1)\}$.

[1] The method is referred to as *dyadic range* in [GKMS02, CM05], *segment tree* in [Ben77, DBVKOS97], and was rediscovered in [NNPC10, SBC+07], and has also been used in [GMW15, AHO+18].

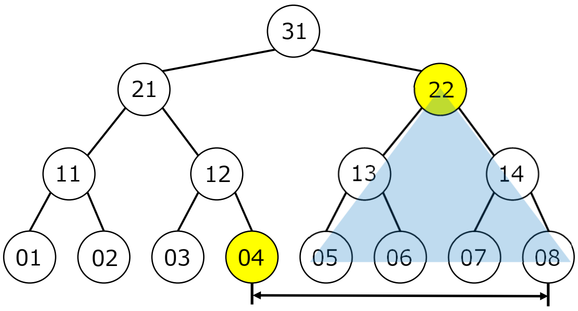


Figure 8.2: Example of Range Encoding, for $R_1 = [4, 8]$. Here, $\mathsf{rangeEnc}([4, 8]) = \{(0, 4), (2, 2)\}$.

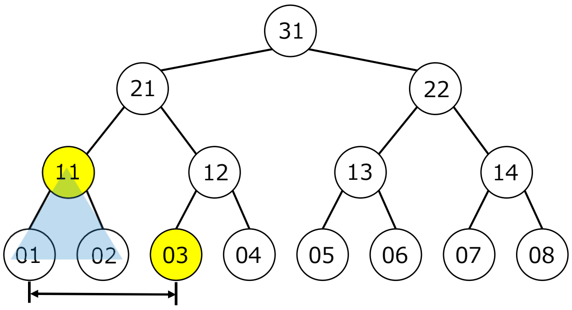


Figure 8.3: Example of Range Encoding, for $R_2 = [1, 3]$. Here, $\mathsf{rangeEnc}([4, 8]) = \{(0, 3), (1, 1)\}$.

We let

$$\begin{aligned} \mathcal{D}_n &:= \{[u, v] | 1 \le u \le v \le n\}\,, \\ \mathcal{L}_n &:= \{[1, v] | 1 \le v \le n\}\,, \\ \mathcal{R}_n &:= \{[u, n] | 1 \le u \le n\}\,. \end{aligned}$$

That is, $\mathcal{D}_n$ is the set of all ranges, while $\mathcal{L}_n$ and $\mathcal{R}_n$ fix the start point and end point to 1 and n, respectively. For any range $R \in \mathcal{D}_n$, a node $w \in \mathbb{S}_n$ is called a *cover node* of R, and we write $w \in \text{cover}(R)$, if all the leaves that are descendants of w are in R. Let $2^{\mathbb{S}_n}$ be the collection of all subsets of $\mathbb{S}_n$. We then define two encoding functions:

- **Range Encoding.** $\mathsf{rangeEnc} : \mathcal{D}_n \to 2^{\mathbb{S}_n}$. For $R \in \mathcal{D}_n$, define

$$\mathsf{rangeEnc}(R) := \{\, w \in \mathbb{S}_n | w \in \text{cover}(R), \text{parent}(w) \notin \text{cover}(R)\,\}\,. \tag{8.1}$$

- **Point Encoding.** $\mathsf{pointEnc} : [1, n] \to 2^{\mathbb{S}_n}$. For $x \in [1, n]$, define $\mathsf{pointEnc}(x)$ as the set of all ancestors of x in $\mathbb{T}_n$.

Lemma 8.2. *For $R \in \mathcal{L}_n \cup \mathcal{R}_n$, we have that* $\mathsf{rangeEnc}(R)$ *contains* at most one node *from each layer.*

Lemma 8.3. *For $x \in [1, n]$, we have that* $\mathsf{pointEnc}(x)$ *contains* exactly one node *from each layer.*

From these lemmata, we obtain that

$$\begin{aligned} \mathsf{rangeEnc}(R) &= \{\, (i, a_i) \mid i \in W_R \,\}, & (8.2) \\ \mathsf{pointEnc}(x) &= \{\, (i, b_i) \mid i \in [1, \log n] \,\}, & (8.3) \end{aligned}$$

for some a_i, b_i, where we let $W_R \subseteq [0, \log n]$ be the set of layers in which there exists a node in $\mathsf{rangeEnc}(R)$.

Lemma 8.4. *For any $R \in \mathcal{L}_n \cup \mathcal{R}_n$, and any $x \in [1, n]$,*

$$|\mathsf{rangeEnc}(R) \cap \mathsf{pointEnc}(x)| = \begin{cases} 1 & \textit{if } x \in R \\ 0 & \textit{if } x \notin R \end{cases}.$$

From this lemma, Eq.(8.2), and Eq.(8.3), we have the following corollary.

Corollary 8.1. *For any $R \in \mathcal{L}_n \cup \mathcal{R}_n$, and any $x \in [1, n]$,*

$$x \in R \ \Leftrightarrow \ \textit{There exists a unique } i \in W_R \textit{ s.t. } a_i = b_i.$$

where W_R, a_i, b_i are defined in Eqs.(8.2) and (8.3).

Example. Let $n = 8$. Consider $x = 6$, $R = [4, 8] \in \mathcal{R}$, and $L = [1, 3] \in \mathcal{L}$. We have

$$\begin{aligned} \mathsf{pointEnc}(6) &= \{(0, 6), (1, 3), (2, 2), (3, 1)\}, \\ \mathsf{rangeEnc}([4, 8]) &= \{(0, 4), (2, 2)\}, \\ \mathsf{rangeEnc}([1, 3]) &= \{(0, 3), (1, 1)\}, \end{aligned} \tag{8.4}$$

and $W_{[4,8]} = \{0, 2\}$, $W_{[1,3]} = \{0, 1\}$. Now since $6 \in [4, 8]$, we have that Lemma 8.4 holds with an intersection node $(2, 2)$, while Corollary 8.1 holds at $i = 2$, and $a_2 = b_2 = 2$. On the other hand, since $6 \notin [1, 3]$, we can verify that the intersection of their encodings is empty.

8.1.3 Tree-based comparison protocols

We are now ready to describe the protocol of Type 5 and 4 by Nergiz et al. [NNPC10]. Let $H : 0, 1^* \to \mathbb{Z}_N$ be a bijective function (or collision-resistant hash).

Protocol 1 (for Type 5) [NNPC10].

1. **Alice**:
   - Compute $E_{pk}(H(i, b_i))$ for all $(i, b_i) \in \mathsf{pointEnc}(x)$ and send to Bob, together with pk.
2. **Bob**:

- Set $R = [y, n]$. (Hence, $R \in \mathcal{R}_n$.)
- Compute $\mathsf{rangeEnc}(R)$ as Eq.(8.2).
- For all $i \in [0, \log n]$, randomly choose $r_i \in \mathbb{Z}_N^*$ and compute

$$V_i := \begin{cases} E_{pk}\Big(r_i\big(H(i,b_i) - H(i,a_i)\big)\Big) & \text{if } i \in W_R \\ E_{pk}(r_i) & \text{if } i \notin W_R \end{cases} \tag{8.5}$$

- Randomly shuffle all V_i to V_i', and send all to Alice.

3. **Alice**:

- Decrypt all V_i' and output $\delta = 1$ (meaning $x \geq y$) if there is exactly one plaintext being zero. Otherwise, output $\delta = 0$ (meaning $x < y$).

Intuition for Correctness/Security. We sketch the intuition as follows. Suppose $x \geq y$. We have $x \in [y, n]$. Hence, due to Corollary 8.1, there is exactly one layer where the encrypted node label is the same from Alice ' s point encoding and Bob ' s range encoding. Therefore, exactly one ciphertext in the list of V_i' will be 0. As for security, Bob will not be able deduce any information due to the security of the encryption scheme. Alice will not know the range R (the private input of Bob), due to the shuffle of the V_i values and due to the inclusion of the random values r_i, which is used for randomizing each non-zero value to a *random* non-zero element.

Protocol 2 (for Type 4). This is a slight variant of the extended protocol in [NNPC10]. The difference is that in [NNPC10], the output sharing is done modulo 2, while in our setting, the sharing is done modulo N. To share in modulo N, we have an additional step, namely Step 4 below. The intuition is described below.

1. **Alice**:

- Compute $E_{pk}(H(i, b_i))$ for all $(i, b_i) \in \mathsf{pointEnc}(x)$ and send to Bob, together with pk.

2. **Bob**:

- Randomly choose $s^B \in \{0, 1\}$ and set

$$R = \begin{cases} [1, y-1] & \text{if } s^B = 1, \\ [y, n] & \text{if } s^B = 0. \end{cases}$$

  (Hence, $R \in \mathcal{L}_n \cup \mathcal{R}_n$.)

- Compute $\mathsf{rangeEnc}(R)$ as Eq.(8.2).

- For all $i \in [0, \log n]$, randomly choose $r_i \in \mathbb{Z}_N^*$ and compute

$$V_i := \begin{cases} E_{pk}\Big(r_i\big(H(i,b_i) - H(i,a_i)\big)\Big) & \text{if } i \in W_R, \\ E_{pk}(r_i) & \text{if } i \notin W_R. \end{cases} \tag{8.6}$$

- Randomly shuffle all V_i to V_i', and send all to Alice.

3. **Alice**:
   - Decrypt all V_i' and set $s^A = 1$ if there is exactly one plaintext being zero. Otherwise, output $s^A = 0$.
   - Send $E_{pk}(s^A)$ to Bob.

4. **Bob**:
   - pick $\delta^B \in \mathbb{Z}_N$. Compute $C = E_{pk}(s^A) + E_{pk}(s^B) - 2s^B E_{pk}(s^A) - E_{pk}(\delta^B)$.
   - Send C to Alice.

5. **Alice**:
   - Set δ^A as the decryption result of C.

Intuition for Correctness/Security. We first verify that $\delta^A \oplus \delta^B = \delta$. By the same correctness argument as for Protocol 1, we have that Protocol 2 ensures that $s^A = 1$ iff $x \in R$. The trick here is then to vary R in the two cases: $R = [1, y-1]$ and $R = [y, n]$ (corresponding to the case where $s^B = 1$ and $s^B = 0$, respectively, by our construction). From these, we have

- if $s^B = 1, s^A = 1$, then $x \in [1, y-1]$.
- if $s^B = 1, s^A = 0$, then $x \in [y, n]$.
- if $s^B = 0, s^A = 1$, then $x \in [y, n]$.
- if $s^B = 0, s^A = 0$, then $x \in [1, y-1]$.

By definition, $\delta = 1$ iff $x \in [y, n]$, and hence we have that $s^A \oplus s^B = \delta$, as claimed.

Next we use the fact that $\delta = s^A \oplus s^B = s^A + s^B - 2s^A s^B$. By our construction, $C = E_{pk}(\delta - \delta^B)$, which implies that $\delta^A + \delta^B \equiv \delta \bmod N$, as required.

As for security, since s^B is hidden from Alice, she will not know which case is being tested.

Improvement. We provide a further improvement for Protocol 1 and 2 above. This reduces the communication cost by one ciphertext for each of Step 1 and 2 (hence, two ciphertexts overall). This can be done by first observing that since the root node, namely, the node with label $(\log n, 1)$, is always in the point encoding of *any* point in $[1, n]$, Alice can omit sending $E_{pk}(H(\log n, 1))$ to Bob in the first pass (in Step 1), and simply letting Bob compute by himself. This already reduces one ciphertext in Step 1. The next observation is that $V_{\log n}$ encrypts 0 if and only if $y = 1$. (Since, $\log n \in W_R$ iff $R = [1, n]$, the full range.) Hence, we just treat only the case of $y = 1$ specifically while simply omitting $V_{\log n}$ altogether. This will reduce one ciphertext in Step 2. To enable this, in Step 2, we can let Bob compute as follows.

- If $y > 1$, then compute V_i as usual (Eqs. (8.5), (8.6)), albeit for only $i \in [0, \log n - 1]$. (That is, $V_{\log n}$ will not be used.) Shuffle V_i to V_i' for all $i \in [0, \log n - 1]$ and send back to Alice these $\log n$ ciphertexts.

- If $y = 1$, then simply generate $\log n$ ciphertexts with only one message being 0 and the others are random.

In Step 3, Alice does as usual, i.e., to check if one plaintext is zero, albeit among all the $\log n$ ciphertexts (instead of $\log n + 1$ ciphertexts as before). The security trivially follows from the security of the basic protocols.

Parallel Time Complexity. One of the main reason that we choose these protocols is that all the heavier procedures, such as encryption, decryption, homomorphic valuation, can be run in parallel. In particular, in Step 2, all the calculation for V_i can be done independently for each i, which means that we can compute them in parallel. The same is true for Steps 1 and 3. Hence the parallel time complexity is almost *constant* regardless of the number of layers, $\ell + 1$, where we recall that $n = 2^\ell$. (Indeed, ℓ is the bit length of the compared numbers x, y.) It is *almost* constant since the Range encoding and the Point encoding will depend on ℓ, but these computations are much lightweight since they operated on plain data and are comparable to bit decomposition (of a plain value). Indeed, Point encoding is identical to bit decomposition, just phrased in term of a tree-based structure.

8.1.4 Our implemented protocol

We construct a new comparison protocol of Type 1 (shared inputs/shared output), which is the type that can be used as a sub-protocol in generic two-party computation. Our protocol is simply a combination of previous results: we base on Protocol 2 (of Type 4), and use the necessary conversions, described in Section 3.2, to convert it to Type 1.

Our Implemented Protocol (for Type 1).

1. **Alice** and **Bob**:

    - Run Conversion 1 to convert shared inputs to encrypted inputs (for both values: x and y, and in parallel). As a result, Alice obtains sk, while Bob obtains $E_{pk}(x), E_{pk}(y)$.

2. **Alice** and **Bob**:

    - Run Conversion 5 (for protocol type 2) using the following protocol for Type 3 (with plain input x' and y') as a subroutine.

        - Run Protocol 2 (for Type 4) on input x' and y'. As a result, Alice obtains $(\delta')^A$ and Bob obtains $(\delta')^B$, which are the shares of $\delta' = (x' \geq y')$.

- Run Conversion 3 to convert Shares of δ' to encrypted forms. As a result, Alice obtains sk, while Bob obtains $E_{pk}(\delta')$. This is fed back to the remaining part of Conversion 5.

3. **Alice** and **Bob**:

- Run Conversion 4 to convert encrypted outputs to shared outputs. As a result, Alice obtains δ^A, while Bob obtains δ^B.

8.2 Experiment

We implemented our main protocol and conversion protocols for a comparison with existing comparison protocols: Type 1[BO07] and Type 2[GHJR15], [VBdHE15] protocols. We used the Paillier cryptsystem with 1024-bit cryptographic keys and implemented these protocols on a PC with an Intel Core i7-4790 3.6 GHz CPU and 16.0 GB of memory.

The total computation time and communication cost of our protocol for Type 1 are shown in Table 8.1. Our main protocol is constructed by sub-protocols, so we show the computation time for each sub-protocol in Table 8.2. We varied the input size to 5, 10, 25, 50, and 100 bits, ran each experiment 10 times, and averaged the various results.

Table 8.1: The total computation time and communication cost of our protocol for Type 1.

Cost	5 bit	10 bit	25 bit	50 bit	100 bit
Computation (ms)	47.731	47.700	50.301	56.419	65.459
Communication (KB)	5.294	8.399	17.715	33.243	64.296

Table 8.2: Computation Time (ms).

Protocol	5 bit	10 bit	25 bit	50 bit	100 bit
Step 1 (Conversion 2)	0.011	0.011	0.011	0.012	0.012
Step 2 (Conversion 5 including subroutines)	47.350	47.679	50.279	56.396	65.435
Step 2 (Type 3 protocol as a subroutine)	38.626	38.678	38.220	38.030	35.000
Step 2 (Conversion 5 except subroutines)	8.724	9.001	12.059	18.366	30.435
Step 3 (Conversion 4)	0.010	0.010	0.011	0.011	0.011

The results show that conversions only consumes little time regardless of the bitsize. The amount of encrypted data of protocol 2 increases proportionally to the bit size, so the protocol is the bottleneck regarding both computation time and communication cost.

We compared our protocol for Type 1 with [BO07], [GHJR15], and [VBdHE15]. The protocol in [BO07] is a protocol for Type 1 and used as a sub-protocol for k-means clustering. The protocols in [GHJR15] and [VBdHE15] are protocols for Type 2. Therefore, we converted their protocols to Type 1 in the experiment. The performance of the protocol in [VBdHE15] was measured in the original paper, so that we refer to these results in the comparison The environment for the experiment was different from ours and the experiment was conducted using from 1 to 25 bits, so we inferred that the values were directly proportional to bit size and extrapolated these (drawn as dotted lines)

Fig. 8.4 shows the comparison results for computation time. The result shows that the computation time is directly proportional to the bit size but the slope on the graph of our protocol is gentler than the others except for [VBdHE15]. The construction of [GHJR15] is simple but the protocol use a equal-to-zero sub-protocol that needs $\log n$ communication rounds. On the other hand, our protocol for Type 1 works using only 6 communication rounds regardless of the bit length.

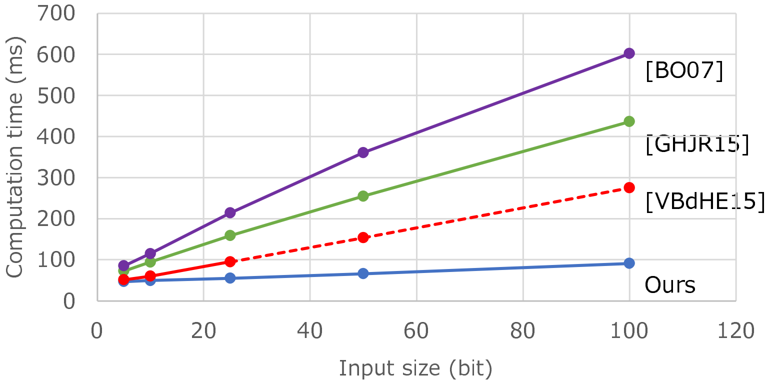


Figure 8.4: Computation time.

Fig. 8.5 shows the comparison of communication cost. As well as computation time, communication cost is in direct proportion to the bit size, and our protocol is more efficient than the others except for [VBdHE15].

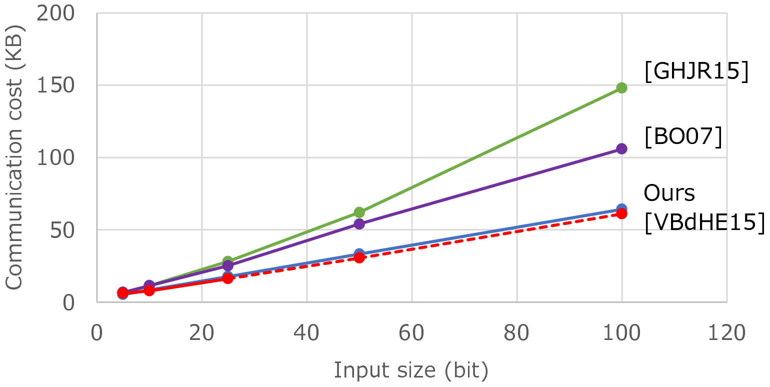


Figure 8.5: Communication cost.

Table 8.3 shows the initialization time for each protocol. Our protocol does not need key exchanges and hence, the time required for the initialization step, which consists of creating and sharing keys and variables to be used in the protocols, is shorter than the other protocols which need multiple keys.

Table 8.3: Initialization time.

Protocol	Initialization time (ms)
Ours	89.124
[BO07]	89.124
[GHJR15]	612.375
[VBdHE15]	132.501

8.3 Conclusion

In this research, we firstly described a taxonomy of secure two-party comparison protocols which allows us to describe the different configurations used for these protocols in a systematic manner. There are a total of 216 types of comparison protocols, and we discussed these types of conversions. These conversions are based on known techniques and have been explored previously, either explicitly or implicitly. We then showed that by combining these conversion techniques, a two-party comparison protocol can be converted into a very efficient protocol in a configuration where the two parties have a share of the values to be compared and a share of the comparison result as output. This setting is often used in multi-party computation protocols, and hence in many privacy-preserving applications as well. We furthermore implemented the protocol and measured its performance. In this input/output configuration, the measurements suggest that our protocol is superior to the previously proposed protocols when offline precomputation is not allowed.

Chapter 9

Conclusion and Future Works

9.1 Summary of our results

The data sets to be de-identified can be divided into structured and unstructured data. We further classified them into static data and dynamic data, and proposed de-identification techniques for each type of data. We assumed the connection between the data and the individual as a privacy risk for any type of data.

In Chapter 4, we examined the most basic data type, namely structured static data. Although it is possible to combine multiple de-identification methods to de-identify data to balance the privacy and utility, such de-identified data could not be evaluated for privacy strength with an easily understandable metric such as k-anonymity. This is because k-anonymity does not support probabilistic de-identification methods like perturbation. Therefore, we proposed a risk assessment by simulation attack for a combination of de-identification methods. Specifically, we design a semi-honest attacker model for a realistic situation and evaluate the privacy risk. The privacy risk is defined as the probability that an individual will be re-identified, which can be evaluated even when de-identification methods are combined. Furthermore, this metric is easy for data subjects to understand and will help promote the obtaining of consent for data utilization.

In Chapter 5, we focused on structured dynamic data and investigated the potential of matrix factorization as a de-identification method. Matrix factorization can be easily combined with other de-identification methods, and we have presented a combination with k-anonymization and random noise assignment. Matrix decomposition separates the matrix containing privacy information from the other matrices, and de-identification can be performed while maintaining the correlation of attributes that are not related to privacy. Therefore, compared to the simple de-identification method, the de-identified dataset using the proposed method maintains higher utility even for the same privacy risk. We consider privacy risk as the probability that an individual will be re-identified. Since this experiment deals with probabilistic de-identification methods, we evaluate the risk by simulation attacks, but the combination of k-anonymization and matrix factorization ensures at least k-anonymity. From the experimental results, we confirmed that we can flexibly control the privacy risk and utility by combining matrix factorization and noise addition.

In Chapters 4 and 5, we proposed to represent the privacy strength of data that is de-identified by multiple de-identification methods through attack simulations. Furthermore, we showed that it is possible to balance privacy and utility by combining de-identification methods.

In Chapter 6, we dealt with unstructured static data, especially document data. We mainly dealt with static data, but dynamic data such as Twitter document can be handled in the same way. Since structured data held by companies are often used for data analysis, there are few situations in which it is released to the public, but there are possible situations in which document data are released to the public. As before, we considered privacy risk to be linked to document data and individuals, and examined risk assessment using simulation attacks. The difference with the handling of structured data was that the data could be exposed to the outside world, and it was necessary to assume a powerful attacker. On the other hand, document data may be required to be readable, so de-identification methods such as perturbation are not suitable. In fact, generalization and suppression have often been used for document data. We then assumed a powerful attacker who can access to the Web, and proposed to detect words with high privacy risk by simulation attacks. Furthermore, we proposed a privacy-preserving algorithm that breaks the connection between document data and individuals by generalizing and suppressing the detected words with high privacy risk. Our simulation attack was actually able to link individuals with manually de-identified document data. Therefore, the similarity with manually de-identified document data is sometimes treated as the utility of the de-identified document data, but it is not adequate. On the other hand, our de-identification algorithm detects and suppresses only words that can lead to the re-identification of an individual, so our proposal achieves both privacy and utility.

In Chapters 7 and 8, we broadened the discussion of de-identification and turned our attention to other techniques for both privacy protection and utility.

Differential privacy is an information-theoretic privacy metric that does not require a situation-specific attacker model. In recent years, differential privacy mechanisms have begun to be put to practical use and are a promising de-identification method. We discussed sensitivity which is a criterion for the amount of noise added by the differential privacy mechanism. While differential privacy does not require the assumption of an attacker model, it does require the assumption of a worst-case scenario, which tends to increase sensitivity. Furthermore, in some cases, it is impossible to derive sensitivity. Therefore, we defined a privacy metric that can be used to derive sensitivity in any situation by adding dummy data. We also constructed a differential privacy mechanism for a t-test as a case study.

Secure computation performs specific functions while keeping the data secret. Therefore, it can be applied to the analysis of distributed data and the outsourcing of data analysis. We specifically addressed comparison protocol in the dissertation. We first covered the patterns of input and output in the two-party comparison protocols, and then organized their conversion protocols. This makes it possible to construct any type of comparison protocol. We further proposed a base protocol based on a tree structure and compared its performance with existing comparison protocols in the setting of their input/output types. The experimental results show that our proposed protocol has the lowest computation time and the lowest communication cost among the comparison protocols proposed so far.

9.2 Future works

We proposed privacy protection methods for various forms of data with a generalized privacy concept of k-anonymity. Especially for structured data, we thought we could control privacy and usefulness by combining de-identification methods. Therefore, we first studied how to evaluate the privacy of a dataset to which multiple de-identification methods were applied, and conducted an evaluation using a simulation attack. We also proposed the use of matrix factorization as a de-identification method, and confirmed that it is possible to maintain usefulness at the same level of privacy strength compared to conventional methods. For unstructured data, we proposed a privacy protection algorithm using attack simulation as a sub-protocol, assuming a more powerful attacker model. Our proposal assumes real-world use and help from laws and system design.

On the other hand, information-theoretic de-identification methods, such as differential privacy, do not require an attacker model and are more privacy friendly. However, their utility is severely degraded, and detailed mechanisms need to be designed for specific use cases to ensure that only the necessary information is retained. Research on differential privacy has progressed in recent years, and methods that do not impair its usefulness for specific purposes have been proposed. It is also expected that understanding of consent will increase as differential privacy mechanisms become more widely used. Furthermore, there is no sufficient amount of data for a single institution, and the need for data analysis using secure computation is beginning to emerge. In order to respond to these changes in the situation, it is necessary to promote research on differential privacy and secure computation.

List of Publications

Books

1. Atuko Miyaji, Tomoaki Mimoto. Security infrastructure technology for integrated utilization of big data: applied to the living safety and medical fields. *Springer 2020*, ISBN 978-981-15-3653-3.

Journals

1. Tomoaki Mimoto, Masayuki Hashimoto, Shinsaku Kiyomoto, Koji Kitamura, Atsuko Miyaji. Privacy Risk of Document Data and a Countermeasure Framework. *Journal of Information Processing*, volume 29, pages 778-786, 2021. (**Ref.** [MHK$^+$21])

2. Tomoaki Mimoto, Seira Hidano, Shinsaku Kiyomoto, Atsuko Miyaji. Anonymization Technique Based on SGD Matrix Factorization. *IEICE Trans. Inf. Syst.*, 103-D(2): 299-308, 2020. (**Ref.** [MHKM20])

3. Nuttapong Attrapadung, Goichiro Hanaoka, Shinsaku Kiyomoto, Tomoaki Mimoto, Jacob C. N. Schuldt. A Taxonomy of Secure Two-Party Comparison Protocols and Efficient Constructions. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 102-A(9): 1048—1060, 2019. (**Ref.** [AHK$^+$19])

International Conferences

1. Tomoaki Mimoto, Masayuki Hashimoto, Hiroyuki Yokoyama, Toru Nakamura, Takamasa Isohara, Ryosuke Kojima, Aki Hasegawa, Yasushi Okuno. Differential privacy under incalculable sensitivity. *2022 6th International Conference onCryptography, Security and Privacy (CSP 2022)*, 2022. (**Ref.** [MHY$^+$22])

2. Tomoaki Mimoto, Shinsaku Kiyomoto, Koji Kitamura, Atsuko Miyaji. A Practical Privacy-Preserving Algroithm for Document. *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2020)*, pages 1376-1383, 2020. (**Ref.** [MKKM20])

3. Tomoaki Mimoto, Shinsaku Kiyomoto, Seira Hidano, Anirban Basu, Atsuko Miyaji. The Possibility of Matrix Decomposition as Anonymization and Evaluation for Time-sequence Data. *16th Annual Conference on Privacy, Security and Trust (PST 2018)*, pages 1-7, 2018. (**Ref.** [MKH+18])

4. Nuttapong Attrapadung, Goichiro Hanaoka, Shinsaku Kiyomoto, Tomoaki Mimoto, Jacob C. N. Schuldt. A Taxonomy of Secure Two-Party Comparison Protocols and Efficient Constructions. *15th Annual Conference on Privacy, Security and Trust (PST 2017)*, pages 215-224, 2017. (**Ref.** [AHK+17])

5. Tomoaki Mimoto, Shinsaku Kiyomoto, Katsuya Tanaka, Atsuko Miyaji. (p, N)-identifiability: Anonymity under Practical Adversaries. *IEEE 16th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2017)*, pages 996-1003, 2017. (**Ref.** [MKTM17])

6. Tomoaki Mimoto, Anirban Basu, Shinsaku Kiyomoto. Towards Practical k-Anonymization: Correlation-based Construction of Generalization Hierarchy. *13th International Conference on Security and Cryptography (SECRYPT 2016)*, pages 411-418, 2016. (**Ref.** [MBK16])

Domestic Conferences

1. Tomoaki Mimoto, Shinsaku Kiyomoto, Koji Kitamura, Atsuko Miyaji. Attack Algorithms and Risk Assessment for Document Data. In *the 37th Symposium on Cryptography and Information Security (SCIS 2020)*, 1C2-3, 2020.

2. Tomoaki Mimoto, Shinsaku Kiyomoto, Atsuko Miyaji. Approximate Matching of Hierarchical Data Using Bloom Filters. In *the 40th Joint Conference on Medical Informatics*, 2-F-2-03, 2020.

3. Tomoaki Mimoto, Shinsaku Kiyomoto, Koji Kitamura, Atsuko Miyaji. Privacy for Documents and Prototype of Risk Evaluation Tool. In *Special Interest Group on Computer Security (CSEC 2019)*, pages 1-6, 2019.

4. Tomoaki Mimoto and Shinsaku Kiyomoto. Anonymously Processed Information Series Seminars. at Japan Data Communications Association, 2019.

5. Tomoaki Mimoto, Shinsaku Kiyomoto, Jacob C. N. Schuldt, Nuttapong Attrapadung, Goichiro Hanaoka. A Privacy-Preserving Clustering Protocol Using Quasi-Homomorphic Cryptography. In *the 36th Symposium on Cryptography and Information Security (SCIS 2019)*, 2A1–3, 2019.

6. Tomoaki Mimoto. The Potential of Matrix Factorization as a Privacy Protection Technique. In *the 176th Risk Engineering Research Meeting*, Tsukuba University, 2018.

7. Tomoaki Mimoto, Shinsaku Kiyomoto, Atsuko Miyaji. Risk assessment of anonymized data by simulation. In *Computer Security Symposium 2017 (CSS 2017)*, 3F2-2, 2017.

8. Tomoaki Mimoto, Shinsaku Kiyomoto, Atsuko Miyaji. A Study on a Safety Index of Sampling. In *Computer Security Symposium 2016 (CSS 2016)*, 1A4-4, 2016.

9. Tomoaki Mimoto, Seira Hidano, Shinsaku Kiyomoto. A Study on Providing Privacy-Conscious Data Using Game Theory. In *IEICE Society Conference 2016*, D19-12, 2016.

10. Tomoaki Mimoto, Anirban Basu, Shinsaku Kiyomoto. A proposal for Fast privacy risk analysis methods. In *IEICE Society Conference 2016*, NBS-1-9, 2016.

11. Tomoaki Mimoto, Toru Nakamura, Shinsaku Kiyomoto, Atsuko Miyaji. CREST - Initiatives of KDDI R&D Laboratories. In *the 40th Joint Conference on Medical Informatics*, 2015.

12. Tomoaki Mimoto, Anirban Basu, Shinsaku Kiyomoto. A Proposal for Privacy Risk Analysis Methods. In *Computer Security Symposium 2015 (CSS 2015)*, 1B4-3, 2015.

References

[ABN08] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. 2008.

[ACJ$^+$12] Balamurugan Anandan, Chris Clifton, Wei Jiang, Mummoorthy Murugesan, Pedro Pastrana-Camacho, and Luo Si. t-plausibility: Generalizing words to desensitize text. *Trans. Data Privacy*, 5(3):505–534, 2012.

[AF05] S. S. Al-Fedaghi. Balanced k-anonymity. In *Proc. of WASET*, volume 6, pages 179–182, 2005.

[AFK$^+$05] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *Proc. of ICDT 2005, LNCS*, volume 3363, pages 246–258, 2005.

[Agg05] Charu C Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, volume 5, pages 901–909, 2005.

[AHK$^+$17] Nuttapong Attrapadung, Goichiro Hanaoka, Shinsaku Kiyomoto, Tomoaki Mimoto, and Jacob CN Schuldt. A taxonomy of secure two-party comparison protocols and efficient constructions. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pages 215–21509. IEEE, 2017.

[AHK$^+$19] Nuttapong Attrapadung, Goichiro Hanaoka, Shinsaku Kiyomoto, Tomoaki Mimoto, and Jacob CN Schuldt. A taxonomy of secure two-party comparison protocols and efficient constructions. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 102(9):1048–1060, 2019.

[AHO$^+$18] Nuttapong Attrapadung, Goichiro Hanaoka, Kazuto Ogawa, Go Ohtake, Hajime Watanabe, and Shota Yamada. Attribute-based encryption for range attributes. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 101(9):1440–1455, 2018.

[BCG$^+$09] Eric Bier, Richard Chow, Philippe Golle, Tracy Holloway King, and Jessica Staddon. The rules of redaction: Identify, protect, review (and repeat). *IEEE Security & Privacy*, 7(6):46–53, 2009.

[Ben77] Jon Louis Bentley. Solutions to klee ' s rectangle problems. *Unpublished manuscript*, pages 282–300, 1977.

[BFL$^+$11] Mauro Barni, Pierluigi Failla, Riccardo Lazzeretti, Ahmad-Reza Sadeghi, and Thomas Schneider. Privacy-preserving ecg classification with branching programs and neural networks. *IEEE Transactions on Information Forensics and Security*, 6(2):452–468, 2011.

[BK07] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *SiGKDD Explorations*, 9(2):75–79, 2007.

[BKBL07a] J.-W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymity using clustering technique. In *Proc. of the International Conference on Database Systems for Advanced Applications*, pages 188–200, 2007.

[BKBL07b] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. Efficient k-anonymization using clustering techniques. In *International Conference on Database Systems for Advanced Applications*, pages 188–200. Springer, 2007.

[BO07] Paul Bunn and Rafail Ostrovsky. Secure two-party k-means clustering. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 486–497, 2007.

[BPTG15] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. Machine learning classification over encrypted data. In *NDSS*, volume 4324, page 4325, 2015.

[BS15] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 127–135, 2015.

[CDH10] Octavian Catrina and Sebastiaan De Hoogh. Improved primitives for secure multiparty integer computation. In *International Conference on Security and Cryptography for Networks*, pages 182–199. Springer, 2010.

[CG11] Chad Cumby and Rayid Ghani. A machine learning based system for semi-automatically redacting documents. In *Twenty-Third IAAI Conference*, 2011.

[CGRM08] Venkatesan T Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh K Mohania. Efficient techniques for document sanitization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 843–852, 2008.

[Chu17] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[CJK^{+}18] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658, 2018.

[CM05] Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[CM06] Kamalika Chaudhuri and Nina Mishra. When random sampling preserves privacy. In *Annual International Cryptology Conference*, pages 198–213. Springer, 2006.

[CZA07] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Hierarchical als algorithms for nonnegative matrix and 3d tensor factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 169–176. Springer, 2007.

[CZH12] Sanjay Chawla, Yu Zheng, and Jiafeng Hu. Inferring the root cause in road traffic anomalies. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 141–150. IEEE, 2012.

[Dal77] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15(429-444):2–1, 1977.

[DBVKOS97] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Schwarzkopf. Computational geometry. In *Computational geometry*, pages 1–17. Springer, 1997.

[DFM16] Josep Domingo-Ferrer and Krishnamurty Muralidhar. New directions in anonymization: permutation paradigm, verifiability by subjects and intruders, transparency to users. *Information Sciences*, 337:11–24, 2016.

[DGK08] Ivan Damgard, Martin Geisler, and Mikkel Kroigard. Homomorphic encryption and secure comparison. *International Journal of Applied Cryptography*, 1(1):22–31, 2008.

[DJW13] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[DJW18] John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

[DKKW11] Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. The yahoo! music dataset and kdd-cup'11. In *Proceedings of the 2011 International Conference on KDD Cup 2011-Volume 18*, pages 3–18. JMLR. org, 2011.

[DKM$^+$06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. *arXiv preprint arXiv:1712.01524*, 2017.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[DNLA18] Bolin Ding, Harsha Nori, Paul Li, and Joshua Allen. Comparing population means under local differential privacy: with significance and power. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Dwo06] C. Dwork. Differential privacy. In *Proc. of ICALP 2006, LNCS*, volume 4052, pages 1–12, 2006.

[EAZS18] Can Eyupoglu, Muhammed Ali Aydin, Abdul Halim Zaim, and Ahmet Sertbas. An efficient big data anonymization algorithm based on chaos and perturbation techniques. *Entropy*, 20(5):373, 2018.

[EFG$^+$09] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *International symposium on privacy enhancing technologies symposium*, pages 235–253. Springer, 2009.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

[EVTL12] Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE transactions on information forensics and security*, 7(3):1053–1066, 2012.

[FBF09] J. H. Freidman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. In *ACM Transactions on Mathematical Software*, volume 16 (5), pages 670–682, 2009.

[fHI10] Cnadian Institute for Health Information. 'best practice' guidlines for managing the disclosure of d-identified health information, 2010.

[FLS$^+$11] Michael Freiman, Jason Lucero, Lisa Singh, Jiashen You, Michael DePersio, and Laura Zayatz. The microdata analysis system at the us census bureau. In *Proceedings of the Joint Statistical Meetings, American Statistical Association Section on Survey Research Methods*, 2011.

[GCF10] Saikat Guha, Bin Cheng, and Paul Francis. Challenges in measuring online advertising systems. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 81–87, 2010.

[GHJR15] Craig Gentry, Shai Halevi, Charanjit Jutla, and Mariana Raykova. Private database access with he-over-oram architecture. In *International Conference on Applied Cryptography and Network Security*, pages 172–191. Springer, 2015.

[GKMS02] Anna C Gilbert, Yannis Kotidis, S Muthukrishnan, and Martin J Strauss. How to summarize the universe: Dynamic maintenance of quantiles. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pages 454–465. Elsevier, 2002.

[GLRV16] Marco Gaboardi, Hyun Lim, Ryan Rogers, and Salil Vadhan. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In *International conference on machine learning*, pages 2111–2120. PMLR, 2016.

[GMW] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Annual ACM Symposium on Theory of Computing.*

[GMW15] Romain Gay, Pierrick Méaux, and Hoeteck Wee. Predicate encryption for multi-dimensional range queries from lattices. In *IACR International Workshop on Public Key Cryptography*, pages 752–776. Springer, 2015.

[GNHS11] Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–77. ACM, 2011.

[GSV07] Juan Garay, Berry Schoenmakers, and José Villegas. Practical and secure solutions for integer comparison. In *International Workshop on Public Key Cryptography*, pages 330–342. Springer, 2007.

[Gut84] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, volume 14, page 47, 1984.

[GZ92] Brian V Greenberg and Laura V Zayatz. Strategies for measuring risk in public use microdata files. *Statistica Neerlandica*, 46(1):33–48, 1992.

[HCC+12] X. He, H. Chen, Y. Chen, Y. Dong, P. Wang, and Z. Huang. Clustering-based k-anonymity. In *Advances in Knowledge Discovery and Data Mining SE*, volume 7301, pages 405–417. Springer-Verlag, 2012.

[HD11] Cho-Jui Hsieh and Inderjit S Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1064–1072. ACM, 2011.

[IN07] T. Iwuchukwu and J. F. Naughton. K-anonymization as spatial indexing: Toward scarable and incremental anonymization. In *Proceeding of the 33rd International Conference on Very Large Data Bases, VLDB*, pages 746–757, 2007.

[ISO] ISO 25237:2017(en) Health informatics - Pseudonymization. https://www.iso.org/obp/ui/#iso:std:iso:25237:ed-1:v1:en.

[ISO17] ISO ISO. Iso/iec 29134: 2017" information technology–security techniques–guidelines for privacy impact assessment. *International Organization for Standardization Std*, 2017.

[JGB+16] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[JNS18] Noah Johnson, Joseph P Near, and Dawn Song. Towards practical differential privacy for sql queries. *Proceedings of the VLDB Endowment*, 11(5):526–539, 2018.

[JOm] JNSA Damage Operation Model for Individual Information Leak. https://www.jnsa.org/result/incident/2018.html.

[Kim86] Jay J Kim. A method for limiting disclosure in microdata based on random noise and transformation. In *Proceedings of the section on survey research methods*, pages 303–308. American Statistical Association, 1986.

[Kor08] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[Kor09] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, 2009.

[Kor10] Aleksandra Korolova. Privacy violations using microtargeted ads: A case study. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 474–482, 2010.

[KP13] Georgios Kellaris and Stavros Papadopoulos. Practical differential privacy via grouping and smoothing. *Proceedings of the VLDB Endowment*, 6(5):301–312, 2013.

[Kru07] John Krumm. Inference attacks on location tracks. In *International Conference on Pervasive Computing*, pages 127–143. Springer, 2007.

[KSS14] Florian Kerschbaum, Thomas Schneider, and Axel Schröpfer. Automatic protocol selection in secure two-party computations. In *International Conference on Applied Cryptography and Network Security*, pages 566–584. Springer, 2014.

[Kuh55] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

[KYH$^+$14] Nawsher Khan, Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Zakira Inayat, Waleed Kamaleldin Mahmoud Ali, Muhammad Alam, Muhammad Shiraz, and Abdullah Gani. Big data: survey, technologies, opportunities, and challenges. *The scientific world journal*, 2014, 2014.

[LDR05] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proc. of SIGMOD 2005*, pages 49–60, 2005.

[LDR06] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proc. of the 22nd International Conference on Data Engineering (ICDE '06)*, pages 25–35. IEEE, 2006.

[LJLA17] Jian Liu, Mika Juuti, Yao Lu, and Nadarajah Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–631, 2017.

[LQS12] Ninghui Li, Wahbeh Qardaji, and Dong Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 32–33, 2012.

[LS99] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[LS01] DD Lee and HS Seung. Algorithms for non-negative matrix factorization advances in neural information processing 13 (proc. nips* 2000), 2001.

[LW08] J.-L. Lin and M.-C. Wei. An efficient clustering method for k-anonymization. In *Proc. of the 2008 international workshop on Privacy and anonymity in information society (PAIS '08)*, pages 46–50. ACM, 2008.

[LZXZ14] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284, 2014.

[MBK16] Tomoaki Mimoto, Anirban Basu, and Shinsaku Kiyomoto. Towards practical k-anonymization: Correlation-based construction of generalization hierarchy. In *SECRYPT*, pages 411–418, 2016.

[McS09] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009.

[MGK06] A. Machanavajjhala, J. Gehrke, and D. Kifer. l-diversity: Privacy beyond k-anonymity. In *Proc. of ICDE'06*, pages 24–35, 2006.

[MGK07] A. Machanavajjhala, J. Gehrke, and D. Kifer. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proc. of ICDE'07*, pages 106–115, 2007.

[MHK$^+$21] Tomoaki Mimoto, Masayuki Hashimoto, Shinsaku Kiyomoto, Koji Kitamura, and Atsuko Miyaji. Privacy risk of document data and a countermeasure framework. *Journal of Information Processing*, 29:778–786, 2021.

[MHKM20] Tomoaki Mimoto, Seira Hidano, Shinsaku Kiyomoto, and Atsuko Miyaji. Anonymization technique based on sgd matrix factorization. *IEICE TRANSACTIONS on Information and Systems*, 103(2):299–308, 2020.

[MHY$^+$22] Tomoaki Mimoto, Masayuki Hashimoto, Hiroyuki Yokoyama, Toru Nakamura, Takamasa Isohara, Ryosuke Kojima, Aki Hasegawa, and Yasushi Okuno. Differential privacy under incalculable sensitibity. In *IEEE 6th International Conference on Cryptography, Security and Privacy (CSP 2022)*, 2022.

[Miv13] Kato Mivule. Utilizing noise addition for data privacy, an overview. *arXiv preprint arXiv:1309.3958*, 2013.

[MKH$^+$18] Tomoaki Mimoto, Shinsaku Kiyomoto, Seira Hidano, Anirban Basu, and Atsuko Miyaji. The possibility of matrix decomposition as anonymization and evaluation for time-sequence data. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–7. IEEE, 2018.

[MKKM20] Tomoaki Mimoto, Shinsaku Kiyomoto, Koji Kitamura, and Atsuko Miyaji. A practical privacy-preserving algorithm for document data. In *2020 IEEE 19th*

International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pages 1376–1383. IEEE, 2020.

[MKTM17] Tomoaki Mimoto, Shinsaku Kiyomoto, Katsuya Tanaka, and Atsuko Miyaji. (p, n)-identifiability: Anonymity under practical adversaries. *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 996–1003, 2017.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[MW04] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proc. of PODS 2004*, pages 223–228, 2004.

[MZ17] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE, 2017.

[NCN08] Mehmet Ercan Nergiz, Christopher Clifton, and Ahmet Erhan Nergiz. Multirelational k-anonymity. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1104–1117, 2008.

[NIS15] NIST. De-identification of personal information, 2015.

[NNPC10] Ahmet Erhan Nergiz, Mehmet Ercan Nergiz, Thomas Pedersen, and Chris Clifton. Practical and secure integer comparison and interval check. In *2010 IEEE Second International Conference on Social Computing*, pages 791–799. IEEE, 2010.

[NO07] Takashi Nishide and Kazuo Ohta. Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In *International Workshop on Public Key Cryptography*, pages 343–360. Springer, 2007.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[NS06] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

[NW06] Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[OfCRS12] U.S. Department of Health Office for Civil Rights and Human Services. Guidance regarding methods for de-identification of protected health information in accordance with the health insurance portability and accountability act (hipaa) privacy rule, 2012.

[Off12] Information Commissioner's Office. Anonymisation: managing data protection risk code of practice, 2012.

[OHYJ15] Jinoh Oh, Wook-Shin Han, Hwanjo Yu, and Xiaoqian Jiang. Fast and robust parallel sgd matrix factorization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 865–874. ACM, 2015.

[ORSS13] Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):1–22, 2013.

[RN10] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 735–746, 2010.

[Rom16] Sasha Romanosky. Examining the costs and causes of cyber incidents. *Journal of Cybersecurity*, 2(2):121–135, 2016.

[RRK18] Bita Darvish Rouhani, M Sadegh Riazi, and Farinaz Koushanfar. Deepsecure: Scalable provably-secure deep learning. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.

[RRWN11] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701, 2011.

[Sam01] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[SB16] David Sánchez and Montserrat Batet. C-sanitized: A privacy model for document redaction and sanitization. *Journal of the Association for Information Science and Technology*, 67(1):148–163, 2016.

[SBC$^+$07] Elaine Shi, John Bethencourt, TH Hubert Chan, Dawn Song, and Adrian Perrig. Multi-dimensional range query over encrypted data. In *2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 350–364. IEEE, 2007.

[SCDFSM14] Jordi Soria-Comas, Josep Domingo-Ferrer, David Sánchez, and Sergio Martínez. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *The VLDB Journal*, 23(5):771–794, 2014.

[Ser13] National Health Service. Anonymisation standard for publishing health and social care data specification, 2013.

[SS98] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *Proc. of PODS 1998*, page 188, 1998.

[Swe02a] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. In *J. Uncertainty, Fuzziness, and Knowledge-Base Systems*, volume 10(5), pages 571–588, 2002.

[Swe02b] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[SWL$^+$08] X. Sun, H. Wang, J. Li, T. M. Truta, and P. Li. (p^+, α)-sensitive k-anonymity: a new enhanced privacy protection model. In *Proc. of CIT'08*, pages 59–64, 2008.

[Tan12] Colin Tankard. Big data security. *Network security*, 2012(7):5–8, 2012.

[TV06] T. M. Truta and B. Vinay. Privacy protection: p-sensitive k-anonymity property. In *Proc. of ICDE'06*, pages 94–103, 2006.

[VBdHE15] Thijs Veugen, Frank Blom, Sebastiaan JA de Hoogh, and Zekeriya Erkin. Secure comparison protocols in the semi-honest model. *IEEE Journal of Selected Topics in Signal Processing*, 9(7):1217–1228, 2015.

[Veu14] Thijs Veugen. Encrypted integer division and secure comparison. *International Journal of Applied Cryptography*, 3(2):166–180, 2014.

[VSBH13] Jaideep Vaidya, Basit Shafiq, Anirban Basu, and Yuan Hong. Differentially private naive bayes classification. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 571–576. IEEE, 2013.

[WF06] Ke Wang and Benjamin CM Fung. Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 414–423, 2006.

[WGC19] Sameer Wagh, Divya Gupta, and Nishanth Chandran. Securenn: 3-party secure computation for neural network training. *Proc. Priv. Enhancing Technol.*, 2019(3):26–49, 2019.

[WLD$^+$20] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[WLFW06] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proc. of ACM SIGKDD'06*, pages 754–759, 2006.

[WLK15] Yue Wang, Jaewoo Lee, and Daniel Kifer. Differentially private hypothesis testing, revisited. *arXiv preprint arXiv:1511.03376*, 1, 2015.

[XWP$^+$06a] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu. Utility-based anonymization for privacy preservation with less information loss. *SIGKDD Explor. Newsl.*, 8(2):21–30, 2006.

[XWP$^+$06b] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, and Ada Wai-Chee Fu. Utility-based anonymization using local recoding. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 785–790, 2006.

[Yao82] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167. IEEE, 1986.

[YJ07] Ting Yu and Sushil Jajodia. *Secure data management in decentralized systems*, volume 33. Springer Science & Business Media, 2007.

[Zay07] Laura Zayatz. Disclosure avoidance practices and research at the us census bureau: An update. *Journal of Official Statistics*, 23(2):253, 2007.

[ZCJL13] Yong Zhuang, Wei-Sheng Chin, Yu-Chin Juan, and Chih-Jen Lin. A fast parallel sgd for matrix factorization in shared memory systems. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 249–256. ACM, 2013.

[Zhe15] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):29, 2015.

[ZZM$^+$11] Yu Zheng, Lizhu Zhang, Zhengxin Ma, Xing Xie, and Wei-Ying Ma. Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, 5(1):5, 2011.

[ZZX$^+$12] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *arXiv preprint arXiv:1208.0219*, 2012.

[ZZXM09] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.

[ZZXY10] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pages 1029–1038. ACM, 2010.