A Thesis for the Degree of Ph.D. in Engineering

Advances in Formal Anonymization for Practical Applications

— Efficient and Secure Techniques for Structured and Unstructured Data —

September 2025

Graduate School of Science and Technology
Keio University

JING JIA

Abstract

Modern information systems and applications increasingly rely on large-scale data collection and sharing to provide intelligent services. However, data sharing carries significant privacy risks that can seriously compromise individual privacy. When sensitive information is not properly managed or used for unauthorized purposes, privacy issues such as leakage of users' personal data will arise, potentially discouraging users from sharing their data and thereby hindering the advancement of data-driven research.

This thesis presents three novel anonymization approaches that address the fundamental challenge of enabling secure data analysis while protecting sensitive information from untrusted entities. The proposed approaches focus on practical anonymization solutions that satisfy data owners' privacy requirements while preserving data utility for research and analytical purposes.

The first approach develops a central differential privacy scheme that allows for data analysis without revealing the private information of a user. Particularly, when computing the statistics of the original dataset, it cannot be determined if the original dataset contains a specific individual's data. It generates the perturbed dataset in the encrypted domain using homomorphic encryption. The unmodified data of each user is not exposed to the data curator; therefore, data users do not need a trusted data curator. Compared with existing models in which significant amounts of noise are locally added before sending data to the data curator, the proposed scheme adds considerably less noise while maintaining the same privacy requirements, resulting in enhanced data utility.

The second approach presents a privacy-preserving hierarchical k-anonymization framework over encrypted data. The framework divides the computational domain into two different domains: local and global, where different secure computational methods are conducted flexibly. Homomorphic encryption is conducted in the local domain with limited computational power, whereas secret sharing is conducted in the global domain with sufficient computational subjects. The hierarchical structure is achieved by sending the anonymized results of local domains to global domains, which can achieve higher-level anonymization. The experimental results show that connecting two domains can accelerate the anonymization process by more than two times, with the total execution time reduced by 61.7% while information loss increased by only 5.6%.

The third approach introduces a flexible two-stage k-anonymization framework for unstructured clinical records that adapts to various language models. The framework combines natural language processing methods and privacy-preserving techniques, where the first stage extracts sensitive entities from narrative clinical records according to identifiers predefined by existing privacy rules, and the second stage generates anonymized data that satisfies k-

anonymity. Fine-tuned Bidirectional Encoder Representations from Transformer (BERT) models and prompt-driven Large Language Models (LLMs) were developed and customized in this framework. Experimental results demonstrate that the framework achieves high F1-scores of over 90% across multiple entity types, and the two-stage structure allows for dynamic adjustment of entity categories and anonymization strategies to comply with various privacy regulations.

In summary, these contributions advance the field of privacy-preserving data anonymization by providing practical, deployable solutions that bridge the gap between theoretical privacy guarantees and real-world implementation requirements. The proposed approaches demonstrate that effective anonymization can be achieved without sacrificing data utility, enabling secure data sharing across diverse application domains including smart cities and healthcare systems.

Acknowledgments

Throughout the course of my Ph.D. studies, I had the privilege of receiving support, guidance, and encouragement from many individuals and institutions. I would like to express my heartfelt appreciation to all those who contributed to the successful completion of this work.

First and foremost, I wish to express my deepest gratitude to my supervisor, Prof. Hiroaki Nishi, for his insightful guidance, continuous support, and unwavering encouragement throughout the years. His mentorship has been instrumental both in my academic development and personal growth.

I would also like to extend my sincere thanks to Mrs. Yuko Nishi for her warmhearted support and kindness, which made my time at Keio University more comfortable and fulfilling. My appreciation goes to all the members of West Laboratory, who have helped me in numerous ways during my research and stay in Japan. Their friendship, advice, and technical discussions created an enriching environment. I am especially thankful for the support in both academic and daily matters.

I am also grateful to Professors Hiroshi Shigeno, Kunitake Kaneko, Ryogo Kubo, Valeriy Vyatkin, who participated in my preliminary review, pre-defense, final defense, and formal review. Your constructive feedback and valuable suggestions significantly improved the quality of this dissertation.

I gratefully acknowledge the China Scholarship Council (CSC) and the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan for their generous financial support throughout my doctoral studies. I would like to thank Keio University for providing additional funding through the Keio Leading-edge Laboratory (KLL) Research Grant and the Doctorate Student Grant-in-Aid program, which greatly supported the progress of my thesis work.

Finally, I would like to thank my family and close friends for their love, patience, and constant encouragement. Without their support, this journey would not have been possible.

JING JIA

September 2025

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

Privacy protection in data-driven environments has become one of the most critical challenges in modern digital society, as evidenced by the complex privacy threats emerging from contemporary data collection and analysis systems [1]. Modern digital systems rely on large-scale monitoring and analytics, which improve efficiency and services but also highlight the tension between data utility and individual privacy. These systems integrate diverse technologies including Internet of Things (IoT) sensors, cyber-physical systems, big data analytics, and real-time control mechanisms that continuously monitor user activities across multiple domains [2]. Information infrastructure forms the backbone of modern society, enabling seamless connectivity and intelligent services across diverse sectors as illustrated in Figure 1-1. It supports global communication through satellite and mobile networks, powers transportation systems with real-time data for safety and efficiency, enhances healthcare via digital records and remote monitoring, enables smart distribution in energy grids, facilitates rapid financial transactions, and streamlines public administration in government services. The broad scope of modern digital systems and information infrastructures creates new privacy risks by allowing seemingly harmless information to be aggregated and analyzed.

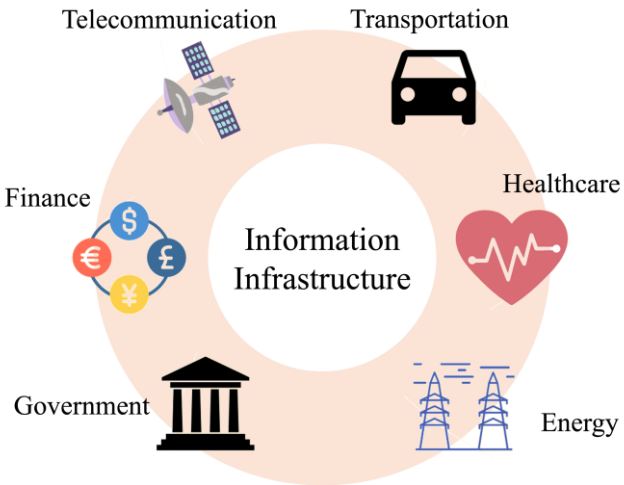


Figure 1-1 Information Infrastructure.

While this data-driven approach offers significant benefits for urban management and quality of life improvements, it simultaneously introduces complex privacy challenges that threaten individual autonomy and confidentiality [3]. The extensive sensor networks deployed across

smart cities create comprehensive infrastructures capable of tracking citizen trajectories, behaviors, and personal preferences [4]. When combined with data from multiple sources—transportation systems, utility networks, healthcare providers, and social media platforms—these datasets can reveal intimate details about individuals' daily routines, health conditions, financial status, and social relationships [5].

The privacy risks become particularly acute when considering the sensitive nature of healthcare data within these interconnected digital ecosystems. Electronic Health Records (EHRs) containing detailed medical histories, diagnostic information, treatment records, and genetic data represent some of the most sensitive personal information that requires protection. While the potential benefits of healthcare data sharing for medical research and personalized treatment development are substantial, the risks associated with unauthorized access, inference attacks, or misuse of this information pose significant threats to individual autonomy and confidentiality [6]. These concerns grow even more serious when healthcare data is combined with other data sources—transportation systems, utility networks, and social media platforms—enabling the reconstruction of intimate details about individuals' daily routines, health conditions, financial status, and social relationships [7].

Traditional privacy protection mechanisms typically rely on architectures where trusted third parties collect and process sensitive data before applying anonymization or privacy-preserving transformations [8]. However, these architectures introduce several fundamental vulnerabilities. Users must place complete trust in data curators who gain access to unmodified sensitive information during processing phases [9]. The limitations of trust-based privacy models have become increasingly apparent as high-profile data breaches continue to occur and regulatory frameworks such as the General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA) impose stricter requirements for data protection [10][11]. These developments necessitate innovative approaches to privacy-preserving data analysis that can eliminate or minimize trust requirements while maintaining data utility for legitimate analytical purposes.

1.2 Research Objectives

The primary challenge in modern data analytics involves balancing the need for high-quality data analysis with the requirement to protect individual privacy. Organizations require access to comprehensive, high-quality datasets to generate meaningful insights, develop innovative services, and advance scientific knowledge [12]; and individuals demand strong privacy guarantees that prevent unauthorized disclosure, misuse, or re-identification of their sensitive information [13]. Privacy protection can be ensured through various techniques, including de-identification, encryption, and anonymization.

Chapter 1. Introduction

De-identification is the removal or alteration of personal information from a dataset so that individuals can no longer be identified. It includes deleting parts of the data that could reveal confidential details. While re-identification methods are quick and straightforward to implement with relatively low computational overhead, they are vulnerable to re-identification attacks through data linkage with limited protection against sophisticated adversaries.

Encryption techniques use mathematical algorithms to transform data into unreadable ciphertext that requires a key to decrypt back to original form. They provide extremely strong protection when properly implemented, and data remains fully recoverable with the correct key. However, these methods are intended to keep information confidential and cannot be used to accelerate the provision of related services by making information public.

Anonymization, compared to re-identification and encryption, offers distinct advantages, particularly in contexts where the data sharing and analysis must balance utility and privacy concerns. Anonymization attempts to preserve the usefulness of the data as much as possible while protecting privacy. Once the data are anonymized, it can be made publicly accessible without the risk of revealing private or identifiable details, making anonymization methods particularly well-suited for efficient and secure data analysis and sharing.

Modern data-driven environments generate and process sensitive information across multiple forms, including structured tabular text data and unstructured narrative text. Traditional approaches typically assume homogeneous data structures, which are insufficient to address the heterogeneous nature of modern datasets. Furthermore, the absence of formal privacy guarantees in conventional privacy protection methods leaves them vulnerable to sophisticated attacks.

The aim of this work lies in establishing a theoretical and practical bridge between formal privacy models and real-world data publishing requirements through the development of provably secure anonymization techniques that adapt to diverse data characteristics. The research is guided by two complementary lines of inquiry:

- Query-Level Anonymization

  The first research direction focuses on scenarios where sensitive datasets remain centralized while external parties interact through controlled queries. Such cases involve designing noise injection mechanisms that satisfy formal privacy definitions while optimizing the utility-privacy trade-off. The objective is to design mechanisms that provide strong, quantifiable privacy guarantees for individual queries while minimizing the loss of utility for legitimate analytical tasks. This methodology enables organizations to provide data access without direct dataset exposure, thereby minimizing privacy risks while supporting legitimate research and analytical services.

- Dataset-Level Anonymization

The second research direction addresses direct dataset publication scenarios prevalent in open data initiatives, collaborative research consortiums, and regulatory compliance frameworks. This approach focuses on developing pre-release transformation techniques that protect individual privacy against linkage attacks and inference-based re-identification attempts. The research explores advanced transformation methodologies including semantic-preserving generalization, strategic data suppression, and learning-driven style transfer mechanisms. These techniques are designed to maintain the statistical validity and semantic coherence of datasets while providing quantifiable privacy protections.

The integration of these complementary approaches establishes a comprehensive framework for practical anonymization that emphasizes three critical performance dimensions: the provision of mathematically verifiable privacy guarantees, the preservation of data utility across heterogeneous data types, and computational scalability for large-scale real-world applications.

1.3 Problem Statement

Existing approaches have significant limitations that impede their practical adoption to develop practical anonymization methods:

- Informal Guarantees

Current anonymization techniques often rely on informal privacy guarantees that provide intuitive protection concepts but lack formal bounds on privacy leakage or quantifiable guarantees against re-identification attacks. Without formal mathematical frameworks, privacy practitioners must rely on subjective judgments about whether anonymization is sufficient, leading to inconsistent implementations and potential vulnerabilities. The absence of provable guarantees also complicates legal and regulatory compliance, as organizations cannot definitively demonstrate that their anonymization meets privacy requirements.

- Data Type Diversity

Traditional anonymization methods were primarily designed for structured, tabular datasets and struggle significantly when applied to the diverse data types prevalent in modern applications. Complex data formats like natural language text present unique re-identification risks that standard techniques cannot adequately address. This limitation forces organizations to either avoid anonymizing certain data types entirely, limiting their sharing potential, or apply inappropriate techniques that provide insufficient protection.

- Trust Requirements

Most anonymization approaches require data owners to trust the anonymization process implementer. This trust dependency creates significant barriers in multi-party scenarios where organizations are reluctant to share sensitive data without guarantees about how it will be handled post-anonymization. Some approaches, such as Local Differential Privacy (LDP) methods, provide formal privacy guarantees and eliminate trust requirements by adding noise to individual data points before transmission to data collectors [14]. However, this kind of approaches results in substantial utility degradation, with accuracy losses that can be orders of magnitude worse than centralized alternatives, leading to many impractical statistical analyses [15]. The noise requirements for achieving meaningful privacy guarantees in LDP can obscure the signal in individual data points, particularly when dealing with high-dimensional or sparse datasets.

- Data Utility Degradation

The fundamental tension between privacy protection and data utility remains a critical limitation that impedes practical anonymization adoption. Most anonymization techniques work by adding noise or generalizing values, all of which reduce the accuracy and granularity of the resulting dataset. Centralized Differential Privacy (CDP) offers superior data utility compared to LDP but requires users to trust data curators with their unmodified information during collection and processing phases [16]. This trust assumption becomes problematic in cloud computing environments where data may be processed by third-party services with potentially inadequate security measures or conflicting privacy policies [17]. The requirement for trusted entities fundamentally conflicts with the zero-trust security principles increasingly adopted by organizations to protect against both external threats and insider attacks. This utility-privacy trade-off is often difficult to optimize, leading to either over-anonymization that destroys data value or under-anonymization that provides inadequate privacy protection.

- Computational Overhead

Existing anonymization techniques often impose substantial computational burdens that make them impractical for large-scale, real-time, or resource-constrained applications. Complex privacy-preserving operations like differential privacy noise calibration, or multi-dimensional generalization can require significant processing power and memory, especially as dataset size and dimensionality increase. Unstructured data presents additional complexities that existing privacy-preserving techniques struggle to address effectively. Narrative clinical records, social media posts, sensor logs, and other text-based data sources contain sensitive information distributed across multiple words, phrases, and contextual relationships [18]. Unlike structured databases where privacy-preserving transformations can be systematically applied to well-defined attributes, unstructured data requires sophisticated natural language processing capabilities to identify and protect sensitive elements while

preserving semantic meaning and analytical value [19]. Computational efficiency and scalability constraints limit the practical applicability of many privacy-preserving technologies [20], particularly prohibitive in resource-constrained environments such as edge computing nodes, mobile devices, or healthcare facilities with limited computational infrastructure [21].

1.4 Contributions

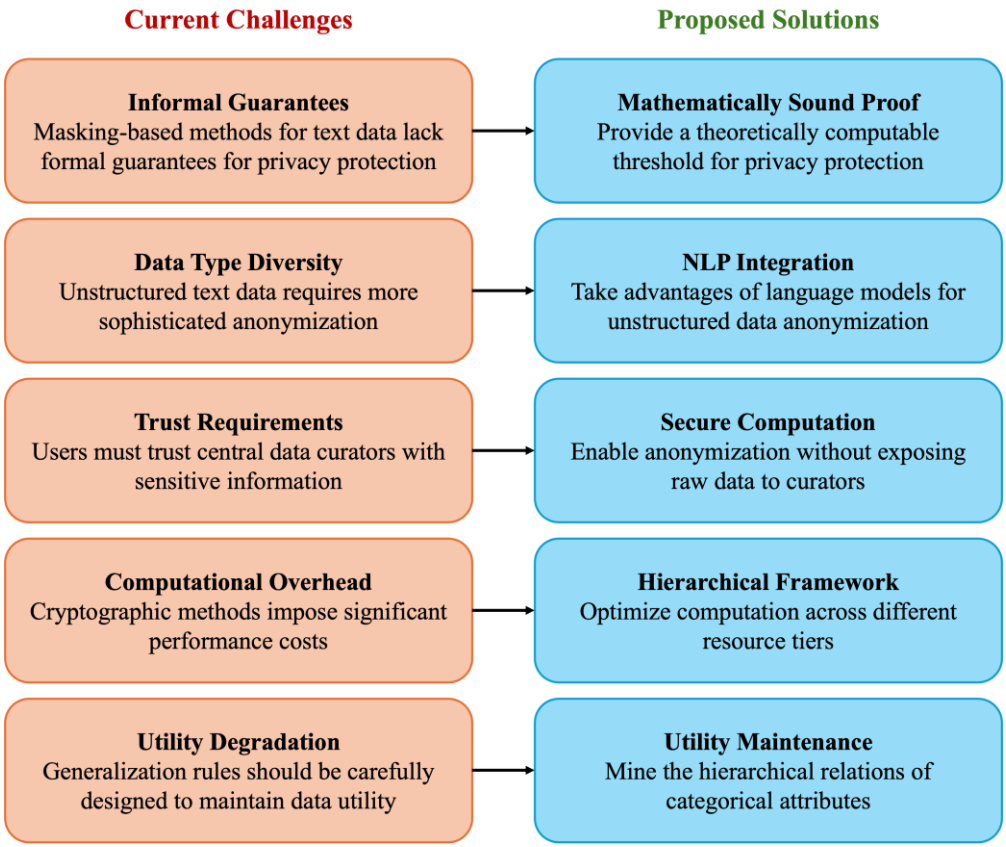


Figure 1-2 Current Challenges and Proposed Solutions.

This research addresses the identified challenges through the development of novel privacy-preserving frameworks that advance both theoretical understanding and practical applicability of secure data analysis techniques. Figure 1-2 shows the current challenges and the corresponding solutions proposed in this thesis. The overall design of this research is driven by two goals: providing formal privacy guarantees and supporting diverse data types. To achieve these, the proposed solutions address key existing challenges in practical anonymization, including trust requirements, computational overhead, and utility degradation.

- Ensuring Formal Privacy Guarantees with Controlled Information Leakage

This solution addresses the limitation of informal privacy guarantees by adopting rigorous mathematical frameworks, such as differential privacy and k-anonymity, to ensure provable

protection against re-identification. The focus is on mechanisms that allow for controlled privacy leakage, enabling fine-grained trade-offs between data utility and protection. By quantifying privacy risk in measurable terms, the proposed solutions ensure that anonymization processes remain transparent, reproducible, and adaptable to varying policy or regulatory requirements.

- Integrating NLP-Enhanced Anonymization for Diverse Data Types

To extend anonymization capabilities beyond structured data, this solution targets unstructured domains such as clinical narratives. It integrates advanced natural language processing methods to detect sensitive entities within narrative clinical records and then applies formal anonymization models to guarantee privacy [22]. Trade-offs between different entity recognition approaches, including fine-tuned transformer architectures and large language models, are analyzed with respect to accuracy, computational cost, and deployment feasibility, enabling privacy-preserving processing for diverse healthcare and information infrastructure datasets.

- Eliminating Trust Requirements through Secure Computation

To remove the dependency on trusted processing environments, this solution designs anonymization techniques that operate entirely within encrypted domains. Secure computation protocols are developed to perform statistical operations and data generalization without revealing raw sensitive data to potentially untrusted entities [23]. The challenge lies in achieving the same privacy guarantees while ensuring efficiency in handling complex generalization operations, thereby protecting inputs throughout the entire anonymization pipeline.

- Reducing Computational Overhead via Hierarchical Anonymization Frameworks

To address computational overhead in heterogeneous environments, this solution introduces hierarchical anonymization frameworks that adapt to available computing resources. Information infrastructures often span from constrained edge devices to high-performance cloud systems[24]. The proposed solution strategically assigns different privacy-preserving techniques to different nodes in the hierarchy, optimizing both computational efficiency and anonymization effectiveness. The framework dynamically selects methods based on local capacity and data protection requirements, supporting scalable, distributed deployments.

- Designing Utility-Preserving and Efficient Generalization Strategies

To prevent utility degradation caused by excessive data distortion, this solution focuses on creating efficient generalization rules that preserve analytical value while meeting privacy constraints. The solution designs algorithmic strategies to identify optimal generalization

hierarchies for structured data and applies domain-specific transformations for unstructured contexts. Optimization methods are incorporated to balance privacy guarantees with maximum data usability, ensuring that anonymized outputs remain suitable for real-world statistical analysis and decision-making [25].

Collectively, these five solutions addressed the overall design goals—formal privacy guarantees and support for diverse data types—while systematically mitigating the critical challenges of trust requirements, computational overhead, and utility degradation. Building upon these solutions, the thesis proposed several concrete contributions that advance the state of the art in privacy-preserving data publishing and processing, spanning theoretical foundations, algorithmic innovations, and deployment-oriented frameworks. The novel contributions are as follows:

The first major contribution introduces a differential privacy framework that utilizes secure computation methods to eliminate trust requirements while achieving data utility superior to local differential privacy approaches. Unlike conventional central differential privacy systems that require users to send unmodified sensitive information to trusted data curators, the proposed framework enables all statistical computations and noise addition operations to be performed entirely within encrypted domains using homomorphic encryption techniques [26]. This approach provides privacy guarantees equivalent to central differential privacy and ensures that sensitive data remains cryptographically protected throughout the entire processing workflow. Experimental results demonstrate that the framework adds significantly less noise than local differential privacy methods, resulting in substantially improved data utility for statistical analyses and machine learning applications.

The second contribution develops a hierarchical k-anonymization framework that strategically combines homomorphic encryption and secret sharing to optimize performance across heterogeneous computational environments. The framework recognizes that information infrastructures typically consist of multi-tier network architectures with varying computational capabilities, from resource-limited edge nodes to high-performance cloud computing centers. By intelligently selecting secure computation methods based on available resources—employing homomorphic encryption in local domains with limited computational entities and secret sharing in global domains with sufficient distributed participants-the framework achieves significant performance improvements while maintaining competitive privacy guarantees. Experimental evaluation shows execution time reductions exceeding 60% compared to homomorphic encryption-only implementations, with little impact on information loss metrics.

The third contribution presents a comprehensive two-stage anonymization framework specifically designed for narrative clinical records that integrates cutting-edge natural language processing capabilities with formal privacy models. The framework addresses unique challenges

in protecting sensitive information within unstructured medical text while preserving clinical utility for research and analytical applications. The first stage employs advanced language models, including domain-specific fine-tuned BERT models and large language models such as GPT-4o and Claude 3.5, to detect and extract sensitive entities according to regulatory frameworks like HIPAA. The second stage applies k-anonymization techniques to the extracted structured data using novel generalization methods that exploit hierarchical relationships between categorical attributes to minimize information loss. Comprehensive evaluation across multiple clinical datasets demonstrates entity detection F1-scores exceeding 90%, with the framework achieving superior anonymization quality compared to existing text de-identification approaches. In addition, detailed comparative analysis of different language model approaches for privacy-preserving applications is provided, offering practical guidance for deployment decisions in resource-constrained environments. Trade-offs between fine-tuned models and prompt-driven models across multiple performance dimensions are evaluated, including detection accuracy, computational requirements, memory utilization, inference latency, and deployment flexibility. The analysis reveals that while prompt-driven large language models achieve higher accuracy in entity detection tasks, fine-tuned models offer substantial advantages in computational efficiency, local deployment capabilities, and operational costs, making them more suitable for many healthcare applications. The research also demonstrates the effectiveness of Low-Rank Adaptation (LoRA) techniques for reducing memory requirements in resource-constrained deployment scenarios.

1.5 Dissertation Structure

The structure of the dissertation is illustrated in Figure 1-3. It is organized into six chapters that systematically investigate privacy-preserving data analysis frameworks, progressing from foundational secure computation techniques to sophisticated applications in clinical text processing. The organization reflects the evolution of the anonymization methods from basic privacy-preserving mechanisms to comprehensive, deployment-ready systems.

Chapter 2 establishes the theoretical foundation by surveying essential background concepts and related work across the research domains. The chapter begins with fundamental privacy-preserving data analysis concepts, including formal privacy models, anonymization techniques, and utility-privacy trade-offs. Differential privacy is examined in detail, covering central and local variants and composition properties. The chapter explores k-anonymity and its extensions, including ℓ-diversity, t-closeness, and m-Invariance, along with hierarchical generalization techniques for minimizing information loss. Secure computation methods, particularly homomorphic encryption and secret sharing, are discussed with emphasis on performance characteristics and practical considerations. The chapter concludes with an examination of information infrastructures and their unique privacy challenges.

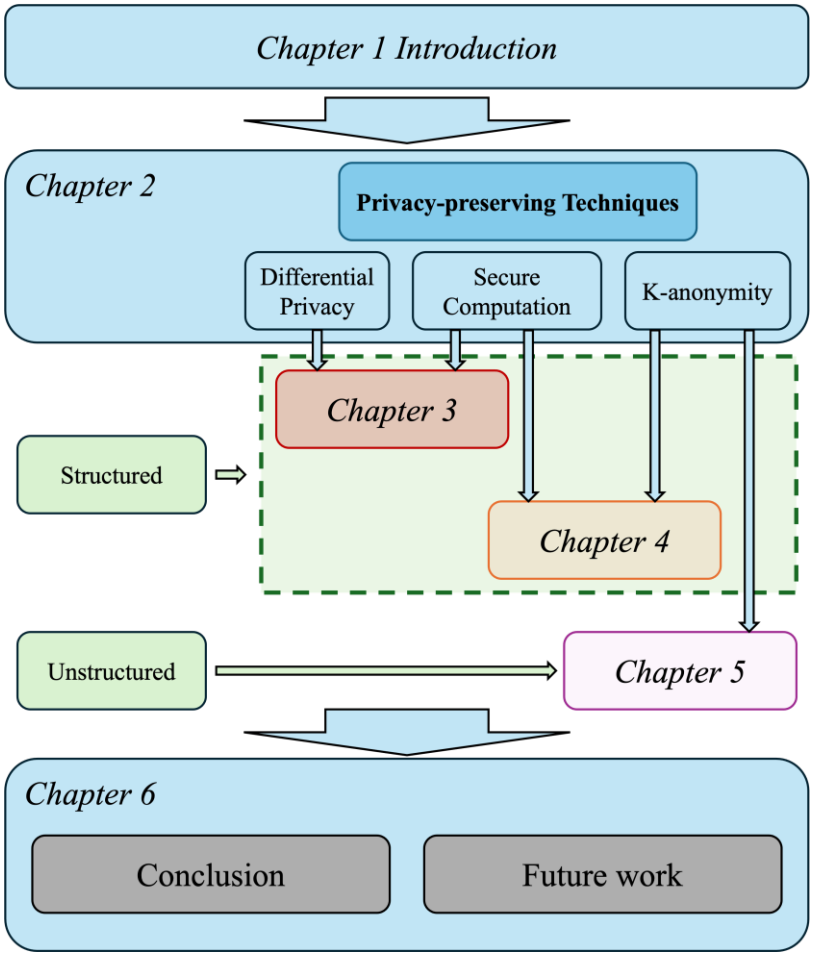


Figure 1-3 Dissertation Structure.

Chapter 3 presents the first major research contribution: a differential privacy framework utilizing secure computing methods. This chapter details the design and implementation of a system that achieves differential privacy without requiring trusted third parties. The chapter describes the three-entity architecture involving data owners, analysts, and data curators, explaining how homomorphic encryption enables statistical computations in encrypted domains. Comprehensive experimental evaluation demonstrates the framework's ability to provide stronger privacy guarantees than local differential privacy while achieving superior data utility through reduced noise requirements.

Chapter 4 extends the foundational work by introducing hierarchical anonymization frameworks that optimize performance across diverse computational environments. This chapter presents an innovative approach that combines homomorphic encryption and secret sharing within hierarchical architectures. The chapter details k-anonymization algorithms designed for encrypted domains, the hybrid secure computation methodology that adapts to available resources, and privacy guarantees maintained across different network tiers. Experimental

results demonstrate substantial performance improvements compared to single-method approaches, with practical validation through deployment in edge computing platforms.

Chapter 5 addresses the specialized challenges of anonymizing unstructured clinical text data through advanced natural language processing integration. This chapter introduces a comprehensive two-stage framework that first detects sensitive entities using state-of-the-art language models, then applies k-anonymization to provide formal privacy guarantees. The chapter provides detailed performance comparisons between fine-tuned models and prompt-driven models, evaluating accuracy, computational requirements, and deployment feasibility. The research offers practical guidance for selecting appropriate approaches based on specific application requirements and resource constraints.

Chapter 6 concludes the research contributions and discusses their implications for privacy-preserving data analysis. The chapter summarizes key innovations introduced across the three main research areas, discusses their collective impact on advancing the field, and identifies remaining limitations and challenges. The chapter concludes with directions for future research, including potential applications to emerging domains and opportunities for further optimization and standardization.

Chapter 2 Background and Related Works

2.1 Privacy-Preserving Data Analysis

Privacy-preserving data analysis has established itself as a fundamental research area addressing the inherent conflict between extracting valuable insights from data and protecting individual privacy rights. It includes diverse methodologies, theoretical constructs, and practical implementations designed to facilitate meaningful data analysis while safeguarding sensitive personal information from unauthorized exposure, inference, or exploitation [27]. The core challenge involves developing sophisticated mechanisms that can reveal aggregate patterns and trends within datasets without compromising the confidentiality of individual records.

The theoretical foundations of anonymization arose from the fundamental limitations of conventional privacy-preserving data analysis methodologies when applied to both structured and unstructured datasets. Traditional privacy-preserving techniques relied heavily on basic approaches such as direct identifier suppression or perturbation methods [28]. These generalization strategies failed to resist sophisticated re-identification attacks that exploited auxiliary information sources, probabilistic inference mechanisms, or multi-dataset linkage analysis to breach individual anonymity [29]. The failure of these elementary generalization techniques highlighted the critical need for more sophisticated anonymization frameworks that could effectively generalize sensitive attributes while maintaining data utility across diverse data structures and formats.

Contemporary privacy-preserving data analysis follows several fundamental principles that distinguish robust privacy protection from superficial data obfuscation techniques:

- The first principle emphasizes the requirement for mathematically rigorous privacy guarantees that can provide formal analysis and quantifiable bounds on information leakage [30]. This principle establishes that effective privacy protection must be based on solid mathematical foundations rather than heuristic approaches or assumptions about limited adversarial capabilities.
- The second principle addresses the importance of modeling sophisticated adversaries with extensive resources and knowledge. Unlike conventional security frameworks that may assume specific attack patterns or constrained adversarial capabilities, privacy-preserving data analysis must account for attackers possessing comprehensive background information, substantial computational resources, and access to multiple

auxiliary data sources [31]. This comprehensive adversarial modeling ensures that privacy protections remain effective even when facing threats that were not initially considered during system development.

- The third principle recognizes the fundamental conflict between privacy preservation and analytical utility. Effective privacy-preserving mechanisms must carefully balance the trade-off between protecting individual confidentiality and maintaining sufficient data quality for meaningful analysis [32]. This balance presents challenges because privacy protection mechanisms typically involve introducing uncertainty, reducing precision, or eliminating specific data elements, all of which can impact the effectiveness of subsequent analytical processes.

The landscape of privacy-preserving data analysis contains multiple distinct methodological approaches, each characterized by unique theoretical foundations, implementation requirements, and application contexts, as shown in Figure 2-1. Statistical disclosure control represents one of the earliest systematic frameworks for privacy-preserving data analysis, incorporating techniques such as data suppression, attribute generalization, and controlled perturbation to prevent sensitive information disclosure in published statistics [33]. These methodologies have been extensively refined within statistical research communities and continue to strengthen data release practices in numerous governmental and institutional contexts.

Differential privacy has emerged as a fundamental framework for privacy-preserving statistical analysis, enabling the release of meaningful aggregate statistics while providing formal privacy guarantees [34]. Traditional differential privacy implementations operate under a trusted curator model, where a central authority processes queries and adds calibrated noise to protect individual privacy [27]. However, this centralized approach introduces significant privacy vulnerabilities, as the curator must access raw sensitive data during computation, creating potential points of failure for privacy breaches[29]. To address these limitations, Chapter 3

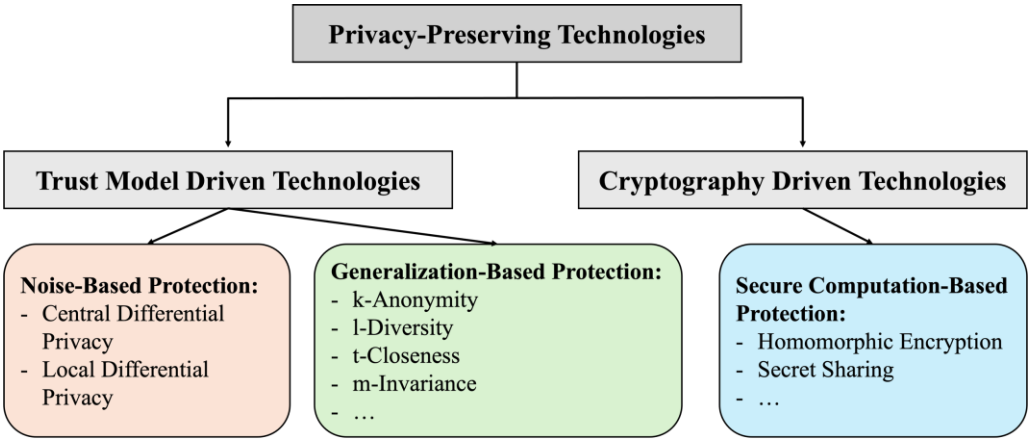


Figure 2-1 Privacy-Preserving Techniques.

proposes a mechanism integrating homomorphic encryption with differential privacy, enabling curators to perform statistical computations directly on encrypted data and return privacy-preserving results without ever accessing plaintext information.

Homomorphic encryption provides substantial advantages for privacy-preserving data analysis by allowing mathematical operations to be performed on ciphertext while maintaining the encrypted state of the data throughout the computational process [26]. This cryptographic approach enables secure outsourcing of data processing tasks to untrusted third parties, facilitating collaborative analysis across multiple organizations without compromising data confidentiality. However, homomorphic encryption schemes typically impose significant computational overhead, with encrypted operations requiring substantially more processing time and memory resources compared to their plaintext equivalents [35]. These performance limitations have motivated the integration of secure multi-party computation protocols, which distribute computational tasks across multiple parties to improve efficiency while maintaining privacy guarantees through cryptographic secret sharing [36].

Chapter 4 proposes an advanced privacy-preserving framework with a hierarchical architecture that combines multiple cryptographic primitives to achieve comprehensive data protection objectives. It integrates homomorphic encryption with secret sharing scheme to enable k-anonymity transformations within encrypted environments, allowing datasets to undergo generalization and suppression operations without requiring decryption. This hierarchical framework leverages the complementary strengths of different cryptographic techniques, enabling secure distributed anonymization processes where multiple parties can collaboratively transform datasets to achieve desired privacy levels while maintaining computational feasibility. The integration of homomorphic encryption and secret sharing provides robust protection against both external attackers and semi-honest participants in the anonymization process.

The anonymization of unstructured data represents more critical challenges in modern privacy-preserving data analysis, particularly given the increasing prevalence of textual information in healthcare, legal, and social media applications [37]. Natural language processing (NLP) techniques have become essential tools for identifying and processing sensitive information embedded within unstructured content, enabling automated detection of personal identifiers, location references, and other privacy-sensitive elements that traditional structured data methods cannot address [38]. The application of NLP techniques in privacy-preserving contexts requires sophisticated understanding of linguistic patterns, contextual relationships, and domain-specific terminology to ensure comprehensive identification of sensitive information while preserving the semantic integrity necessary for meaningful analysis [39].

Textual information contains complex semantic relationships, contextual dependencies, and

implicit identifiers that require advanced generalization techniques to achieve effective privacy protection [40]. The development of k-anonymization frameworks for unstructured clinical data requires careful integration of entity recognition, semantic analysis, and context-aware generalization procedures to maintain clinical utility while ensuring privacy compliance [41].

The necessity to preserve linguistic coherence and domain-specific meaning in clinical narratives, where medical terminology, contextual relationships, and diagnostic information must remain interpretable for healthcare professionals while ensuring patient privacy protection, has driven the development of specialized k-anonymization frameworks for narrative clinical records that integrate advanced natural language processing with context-aware generalization techniques to maintain clinical utility in anonymized healthcare documentation in Chapter 5.

2.2 Differential Privacy

Differential privacy stands as one of the most significant theoretical breakthroughs in privacy-preserving data analysis, providing the first mathematically rigorous framework for quantifying and controlling privacy leakage in statistical computations [34]. Introduced by Dwork in 2006, differential privacy revolutionized the field by establishing a formal definition of privacy that is independent of adversarial background knowledge and computational capabilities [8]. The framework addresses fundamental limitations of previous privacy models by providing provable guarantees that are robust against arbitrary side information and sophisticated inference attacks.

The mathematical foundation of differential privacy rests on the principle of indistinguishability between neighboring datasets that differ in exactly one individual's record:

Given two databases, D and D', drawn from the set of databases $\mathcal{D}$ and differ in exactly one record. In addition, there is a stochastic mechanism M that takes the databases as inputs to generate the output. If, no matter how databases D and D' are selected, the results of $M(D)$ and $M(D')$ are almost indistinguishable, then the stochastic mechanism M satisfies DP.

DEFINITION 1. *A mechanism M satisfies ϵ-DP, where $\epsilon > 0$ is a privacy parameter, if for any two neighboring databases D and D' such that $D = D' - t$ or $D' = D - t$, we have*

$$\forall S \subset \text{Range(M)}, \Pr[M(D) \in S] \leq e^{\epsilon} \Pr[M(D') \in S] \qquad (2-1),$$

where S is the set of all possible outputs of the randomized algorithm, and $\Pr(\cdot)$ is a probability density function.

THEOREM 1. *(Sequential Composition) If M_1 and M_2 are ϵ_1-DP and ϵ_2-DP algorithms with independent randomness, then releasing $M_1(D)$ and $M_2(D)$ on database D satisfies $(\epsilon_1 + \epsilon_2)$-DP.*

THEOREM 2. *(Postprocessing) Let* $M: D \rightarrow R$ *be a randomized algorithm that is* ϵ*-DP. Let* $f: R \rightarrow R'$ *be an arbitrary randomized mapping. Then* $f \circ M: D \rightarrow R'$ *is* ϵ*-DP.*

This definition ensures that the presence or absence of any individual's data has minimal impact on the output distribution, thereby protecting individual privacy while enabling aggregate analysis. The privacy parameter ϵ controls the strength of privacy protection, with smaller values providing stronger guarantees at the cost of reduced utility.

The theoretical elegance of differential privacy lies in its composition properties, which enable complex analytical workflows while maintaining formal privacy guarantees. *Sequential composition* allows multiple differentially private mechanisms to be applied to the same dataset, with the total privacy cost being the sum of individual privacy expenditures [42]. Parallel composition enables simultaneous analysis of separate data partitions while maintaining privacy costs that remain independent across partitions [43]. Advanced composition theorems provide more precise bounds for scenarios involving multiple queries, demonstrating that privacy degradation is often sublinear rather than linear in the number of queries [44].

The implementation of differential privacy typically relies on carefully calibrated noise addition mechanisms that preserve privacy while maintaining statistical utility. The Laplace mechanism adds noise drawn from a Laplace distribution with scale parameter proportional to the sensitivity of the query function and inversely proportional to the privacy parameter [27]. For queries with

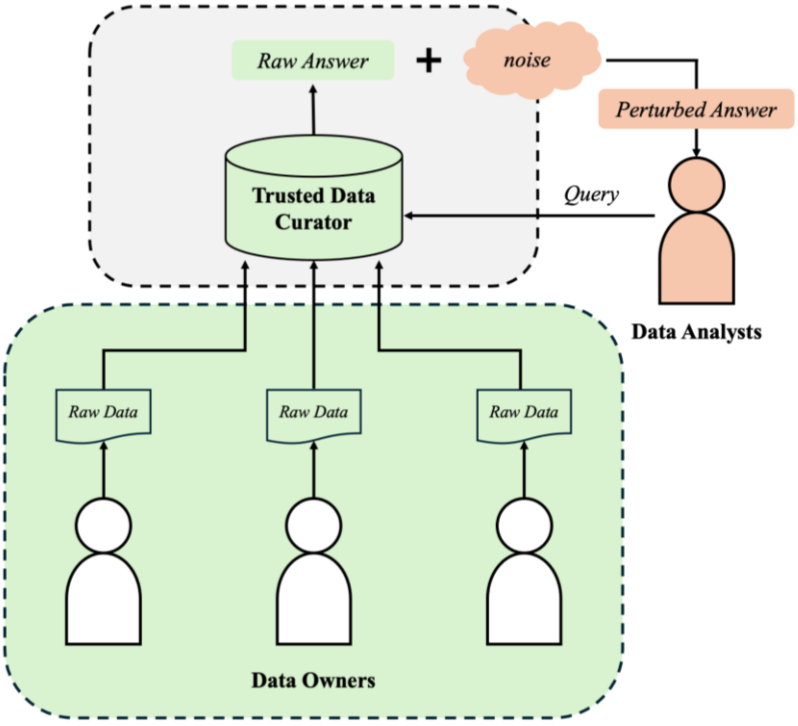


Figure 2-2 Central Differential Privacy Model.

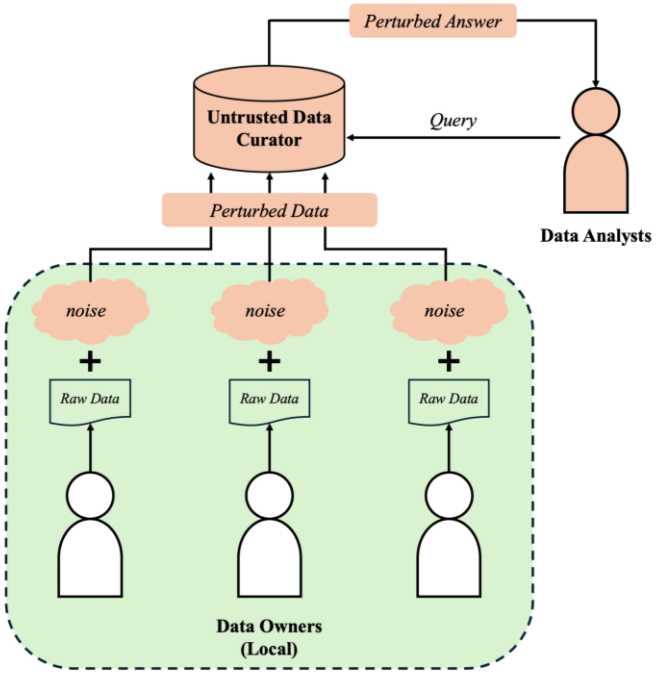


Figure 2-3 Local Differential Privacy Model.

bounded outputs, the exponential mechanism provides a general framework for selecting outputs with probabilities proportional to their utility while maintaining differential privacy [45]. The Gaussian mechanism offers computational advantages for certain applications by utilizing Gaussian noise calibrated to achieve approximate differential privacy guarantees [27].

Central differential privacy, shown in Figure 2-2, represents the traditional model where a trusted curator collects raw data and applies differentially private mechanisms before releasing results [46]. This approach enables high utility outcomes because noise is added only once during the analysis phase, allowing for sophisticated statistical computations and machine learning applications. However, central differential privacy requires complete trust in the data curator, who gains access to sensitive information during the collection and processing phases. This trust requirement has limited adoption in scenarios where data owners are reluctant to share unprotected information with external entities.

Local differential privacy, shown in Figure 2-3, addresses trust limitations by requiring individuals to perturb their own data before transmission to any data collector [30]. In this model, privacy protection is applied at the source, eliminating the need for trusted third-parties and ensuring that raw sensitive data never leaves individual control. Apple's implementation of local differential privacy for collecting usage statistics and Google's RAPPOR system for web analytics demonstrate the practical feasibility of local approaches [9]. However, local differential privacy suffers from significant utility degradation compared to central approaches, often requiring much larger datasets to achieve comparable accuracy.

The privacy-utility trade-offs in differential privacy have been extensively studied, revealing fundamental limits on what can be learned under privacy constraints. The sample complexity of differentially private learning often requires additional data proportional to the square root of the dimensionality for convex optimization problems [47]. For certain tasks, such as releasing high-dimensional statistics or training complex machine learning models, the utility cost of differential privacy can be prohibitive without substantial datasets [48]. These limitations have motivated research into more efficient algorithms and relaxed privacy models that can achieve better utility while maintaining meaningful privacy protection.

The practical deployment of differential privacy faces several challenges that continue to drive research and development efforts. Parameter selection remains a critical challenge, as choosing appropriate privacy parameters requires balancing privacy protection with analytical utility while considering the specific application context and regulatory requirements [49]. Implementing differential privacy correctly requires careful attention to potential sources of privacy leakage, including auxiliary information, timing attacks, and implementation vulnerabilities [50].

Industry adoption of differential privacy has accelerated significantly, with major technology companies integrating differential privacy into production systems for privacy-preserving analytics. The U.S. Census Bureau employed differential privacy for the 2020 decennial census, marking the first large-scale government deployment [51] of formal privacy protection mechanisms. However, this deployment also highlighted ongoing debates about the appropriate balance between privacy protection and data utility in official statistics [52]. Social media platforms and search engines have implemented differential privacy for user behavior analysis while maintaining service quality and advertising effectiveness.

The future evolution of differential privacy continues to address emerging challenges in modern data analysis environments. Integration with other privacy-preserving technologies, such as secure computation and synthetic data generation, promises to create comprehensive privacy protection frameworks that can address diverse application needs while maintaining practical efficiency [53].

2.3 k-Anonymity and its Extensions

k-Anonymity represents one of the earliest formal privacy models designed to protect individual identity in published datasets, establishing a foundational approach that significantly influenced how researchers and industry professionals implement data anonymization processes. Developed by Sweeney in 2002, k-anonymity addresses the critical vulnerability that individuals can be uniquely identified through combinations of seemingly non-sensitive attributes, even when explicit identifiers are removed [24]. The model establishes a quantitative framework for measuring and controlling re-identification risk by ensuring that each individual's record is

indistinguishable from at least $k-1$ other records within the dataset.

According to the data privacy terminology, data attributes can be divided into the following types: (1) direct identifiers can directly and uniquely identify an individual, such as name or ID number; (2) quasi-identifiers are those that can be combined with auxiliary information to reveal someone's identity, such as age, gender, and ZIP code; (3) sensitive attributes are information that individuals want to hide from others, such as salary and disease; and (4) attributes other than the above three types are non-sensitive attributes. The theoretical foundation of k-anonymity establishes on the concept of quasi-identifiers, which are attributes that, when combined, can potentially identify individuals even though they may appear non-sensitive separately. Quasi-identifiers [54] can create unique signatures that enable re-identification attacks. The k-anonymity model requires that for every combination of quasi-identifier values in the dataset, there must exist at least k records sharing identical values. A group of records with the same combination of quasi-identifiers is called a q^* block. All q^* blocks containing k or more records satisfy k-anonymity.

Achieving k-anonymity typically involves two primary transformation operations: generalization and suppression. Generalization replaces specific attribute values with more general representations, such as converting exact ages to age ranges or specific locations to broader geographic regions [55]. Suppression removes certain attribute values or entire records when generalization alone cannot achieve the desired anonymity level [56]. The challenge lies in selecting transformation strategies that minimize information loss while satisfying k-anonymity constraints, a problem that has been proven to be NP-hard for optimal solutions [57].

Various algorithmic approaches have been developed to address the computational complexity of k-anonymization while balancing privacy protection with data utility. The Mondrian algorithm employs recursive partitioning of the data space, creating rectangular regions that satisfy k-

Table 2-1 Original Disease Table.

	ZIP Code	Age	Disease
1	47677	29	Heart Disease
2	47602	22	Heart Disease
3	47678	27	Heart Disease
4	47905	43	Flu
5	47909	52	Heart Disease
6	47906	47	Cancer
7	47605	30	Heart Disease
8	47673	36	Cancer
9	47607	32	Cancer

Table 2-2 k-Anonymization ($k = 3$) of Table 2-1.

	ZIP Code	Age	Disease
1	476**	2*	Heart Disease
2	476**	2*	Heart Disease
3	476**	2*	Heart Disease
4	4790*	≥40	Flu
5	4790*	≥40	Heart Disease
6	4790*	≥40	Cancer
7	476**	3*	Heart Disease
8	476**	3*	Cancer
9	476**	3*	Cancer

anonymity requirements through systematic attribute generalization [58]. Incognito [59] provides a bottom-up approach that systematically explores the generalization space to find optimal solutions for smaller datasets. More recent approaches have incorporated machine learning techniques and optimization methods to improve scalability and solution quality for large-scale datasets[60]. Table 2-1 shows a table of original patients information and Table 2-2 shows the k-anonymized table.

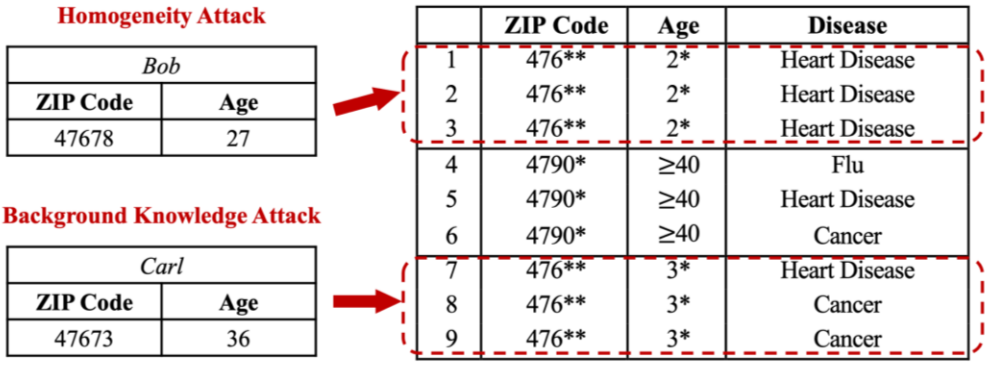


Figure 2-4 Homogeneity Attack and Background Knowledge Attack.

Despite its widespread adoption and intuitive appeal, k-anonymity suffers from several fundamental limitations that have motivated the development of enhanced privacy models as shown in Figure 2-4. The homogeneity attack exploits scenarios where all records within a k-anonymous group share identical sensitive attribute values, enabling attackers to infer sensitive information with certainty regardless of the anonymity level [61]. Background knowledge attacks leverage external information sources to narrow down possible identities within equivalence classes [62], potentially compromising privacy even when k-anonymity requirements are satisfied. Additionally, k-anonymity provides no protection against attribute disclosure, where

Table 2-3 Original Salary/Disease Information.

	ZIP Code	Age	Salary	Disease
1	47677	29	3K	Gastric Ulcer
2	47602	22	4K	Gastritis
3	47678	27	5K	Stomach Cancer
4	47905	43	6K	Gastritis
5	47909	52	11K	Flu
6	47906	47	8K	Bronchitis
7	47605	30	7K	Bronchitis
8	47673	36	9K	Pneumonia
9	47607	32	10K	Stomach Cancer

Table 2-4 ℓ-Diversity ($\ell = 3$) k-Anonymization ($k = 3$) of Table 2-3.

	ZIP Code	Age	Salary	Disease
1	476**	2*	3K	Gastric Ulcer
2	476**	2*	4K	Gastritis
3	476**	2*	5K	Stomach Cancer
4	4790*	≥40	6K	Gastritis
5	4790*	≥40	11K	Flu
6	4790*	≥40	8K	Bronchitis
7	476**	3*	7K	Bronchitis
8	476**	3*	9K	Pneumonia
9	476**	3*	10K	Stomach Cancer

attackers can infer sensitive characteristics without necessarily identifying specific individuals. ℓ-Diversity was introduced to address the homogeneity vulnerability by requiring that each equivalence class contains at least l distinct values for sensitive attributes [61]. Table 2-3 shows a table of original salary/disease information of patients, and

Table 2-4 shows the 3-diversity anonymization of Table 2-3. This extension ensures that sensitive attribute values are well-represented within each group, preventing trivial inference attacks based on attribute homogeneity. Several variants of ℓ-diversity have been proposed, including entropy ℓ-diversity, which considers the distribution of sensitive values, and recursive (c, ℓ)-diversity, which provides stronger protection against probabilistic inference attacks [63]. However, ℓ-diversity can be difficult to achieve when sensitive attributes have biased distributions or when multiple sensitive attributes must be protected simultaneously.

t-Closeness provides an alternative approach by requiring that the distribution of sensitive attribute values within each equivalence class closely approximates the distribution in the overall

dataset [63]. This model addresses limitations of ℓ-diversity by considering the global distribution of sensitive values and preventing attackers from gaining information through distributional differences between groups and the entire population. t-Closeness uses distance measures such as Earth Mover's Distance to quantify distributional similarity and ensure that equivalence classes do not reveal information through their statistical properties [64].

The practical implementation of k-anonymity and its extensions faces several challenges that continue to drive research and development efforts. Scalability becomes a significant concern when dealing with high-dimensional datasets or large volumes of data, as the number of possible generalizations grows exponentially with the number of attributes [65]. Multi-dimensional generalization requires sophisticated algorithms that can handle complex attribute relationships and dependencies while maintaining computational efficiency [66]. Dynamic scenarios, where datasets are continuously updated with new records, present additional challenges for maintaining anonymity properties over time [67].

m-Invariance is a privacy model designed to address the limitations of k-anonymity and ℓ-diversity in dynamic data publishing scenarios, where data is released over multiple time periods and adversaries may exploit temporal correlations to re-identify individuals or infer sensitive attributes [68]. The core idea is to ensure that for any individual whose data appears in multiple releases, their sensitive attribute values remain confined within a stable set—referred to as the "signature"—across all published versions, and that this signature contains at least m distinct sensitive values. By enforcing this invariance property, m-Invariance mitigates the risk of sensitive attribute disclosure due to record linkage or value inference attacks over time, even when adversaries have access to multiple snapshots of the dataset. This approach provides a stronger and more sustainable privacy guarantee in longitudinal data publishing, where maintaining anonymity across time, rather than within a single release, is essential for protecting individual privacy without excessively sacrificing data utility.

Subsequent developments have focused on integrating k-anonymity with other privacy-preserving techniques to create more robust protection mechanisms. Personalized privacy models allow different individuals to specify their own privacy requirements, leading to (α, k)-anonymity and other customizable frameworks [69]. Distributed k-anonymization enables privacy-preserving data integration across multiple organizations without centralizing sensitive information [70]. The combination of k-anonymity with differential privacy has produced hybrid models that provide both group-level protection and statistical privacy guarantees.

The evaluation of k-anonymity and its extensions requires comprehensive metrics that assess both privacy protection and data utility preservation. Information loss measures quantify the amount of precision sacrificed during the anonymization process, typically considering both numerical and categorical attribute transformations [71]. Privacy risk assessment involves

analyzing potential attack scenarios and measuring the residual re-identification risk after anonymization [41]. Utility evaluation must consider the specific analytical tasks that will be performed on the anonymized data, as different applications may be more or less sensitive to particular types of information loss [72].

Modern applications of k-anonymity extend beyond traditional relational databases to encompass diverse data types and emerging application domains. Location privacy applications use k-anonymity principles to protect individual movement patterns while enabling location-based services and urban planning applications [73]. Social network anonymization applies k-anonymity concepts to graph-structured data, protecting both node identity and relationship information [74]. Healthcare applications have extensively utilized k-anonymity for protecting patient privacy in medical research datasets while enabling biomedical discovery and epidemiological studies [75].

The future evolution of k-anonymity continues to address critical performance and applicability challenges in diverse data environments. Computational efficiency remains a fundamental concern, particularly for large-scale datasets where traditional k-anonymity algorithms exhibit prohibitive time complexity that limits practical deployment in resource-constrained environments. Recent advances have focused on developing optimized algorithms and distributed computing frameworks that significantly reduce the computational overhead of k-anonymization processes, enabling scalable privacy protection for massive datasets while maintaining acceptable processing times for real-world applications. The extension of k-anonymity principles to unstructured clinical records represents another vital direction for future development, addressing the growing need for privacy protection in healthcare text mining and clinical research applications. Traditional k-anonymity frameworks designed for structured tabular data cannot directly handle the semantic complexity, contextual dependencies, and linguistic variability inherent in clinical narratives. Developing specialized k-anonymization methods that integrate natural language processing techniques with generalization-based anonymization approaches, enables effective privacy protection for clinical text while preserving the medical semantic content essential for healthcare analytics and research applications.

2.4 Secure Computation

Secure computation represents a transformative paradigm in cryptography that enables computational operations on encrypted or secret-shared data without revealing the underlying information to computing parties [76]. This revolutionary approach addresses fundamental privacy concerns in distributed computing environments by allowing multiple parties to collaboratively compute functions over their private inputs while maintaining complete confidentiality throughout the computational process. The theoretical foundations of secure

computation emerged from the seminal work of Yao in the 1980s [77], which demonstrated that any polynomial-time computable function could be evaluated securely using cryptographic protocols.

The conceptual framework of secure computation involves several distinct but related cryptographic techniques, each offering unique advantages and trade-offs for different application scenarios. The unifying principle across all secure computation approaches is the ability to perform meaningful computations while preserving input privacy, output correctness, and protocol security against various adversarial models [78]. These properties enable secure computation to serve as a foundational technology for privacy-preserving data analysis, collaborative machine learning, and distributed analytics across untrusted networks.

2.4.1 Homomorphic Encryption

Homomorphic encryption stands as one of the most significant achievements in modern cryptography, providing the capability to perform arbitrary computations on encrypted data without requiring decryption. The theoretical possibility of fully homomorphic encryption was first envisioned by Rivest, Adleman, and Dertouzos in 1978, but practical constructions remained challenging for over three decades until Gentry's breakthrough work in 2009 [20]. Homomorphic encryption schemes enable data owners to encrypt their sensitive information and outsource computations to untrusted cloud providers while maintaining complete confidentiality and obtaining encrypted results that can be decrypted only by authorized parties.

Table 2-5 An Overview of Homomorphic Encryption Methods.

Type	Features	Representative Models
Additive HE	Enable addition operations on ciphertexts	Paillier
Multiplicative HE	Support multiplication operations on ciphertexts	ElGamal, RSA
Somewhat HE	Enable both addition and multiplication operations on ciphertexts by limited rounds	BGN
Fully HE	Support arbitrary polynomial-time computations on ciphertexts without operational limitations	BGV, BFV, CKKS

The mathematical foundation of homomorphic encryption relies on algebraic structures that preserve certain operations when applied to encrypted data. Table 2-5 lists an overview of different types of homomorphic encryption methods. Additive homomorphic encryption schemes, such as the Paillier cryptosystem, enable addition operations on encrypted values while

maintaining semantic security against chosen-plaintext attacks [79]. Multiplicative homomorphic schemes, such as the RSA encryption system under specific parameter choices, support multiplication operations on ciphertexts [80]. Somewhat homomorphic encryption schemes enable both addition and multiplication operations on encrypted data, though they are constrained by a limited number of operations before noise accumulation makes decryption infeasible [81].

Additive homomorphic encryption has a homomorphism for addition as the following equation:

$$Dec\big(Enc(x_1) \oplus Enc(x_2)\big) = x_1 + x_2 \qquad (2-2)$$

where $\oplus$ represents an addition operator between ciphertexts, and $Enc(\cdot)$ and $Dec(\cdot)$ denote an encryption function and a decryption function, respectively.

Multiplicative homomorphic encryption has a homomorphism for multiplication as the following equation:

$$Dec\big(Enc(x_1) \otimes Enc(x_2)\big) = x_1 \times x_2 \qquad (2-3)$$

where $\otimes$ represents a multiplication operator between ciphertexts.

Fully homomorphic encryption constitutes the most advanced form of homomorphic encryption capabilities, supporting arbitrary polynomial-time computations on encrypted data without operational limitations. Gentry's original construction utilized ideal lattices and a technique called bootstrapping to manage noise growth during homomorphic operations [82]. Subsequent improvements have focused on enhancing computational efficiency, reducing ciphertext expansion, and developing more practical parameter settings for real-world applications [83]. Modern fully homomorphic encryption schemes, such as BGV, BFV, and CKKS, offer different trade-offs between security assumptions, computational efficiency, and supported data types [84].

The practical implementation of homomorphic encryption faces several significant challenges that continue to drive ongoing research and development efforts. Computational overhead remains a primary concern, as homomorphic operations typically require orders of magnitude more time and memory compared to equivalent plaintext computations [85]. Ciphertext expansion, where encrypted data occupies significantly more storage space than plaintext equivalents, presents challenges for bandwidth-constrained applications and large-scale data processing [86]. Noise management in homomorphic encryption schemes requires careful parameter selection and algorithmic design to ensure computational correctness while maintaining security guarantees [87].

2.4.2 Secret Sharing

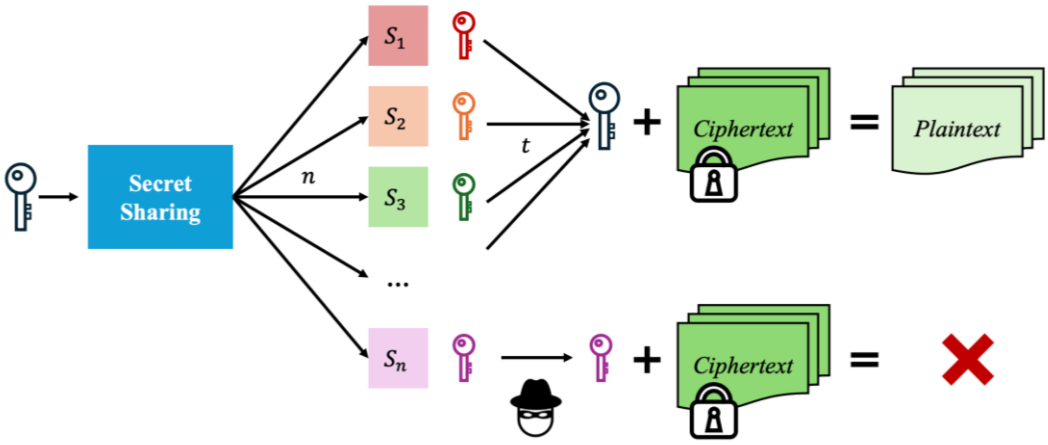


Figure 2-5 Secret Sharing.

Secret sharing, as shown in Figure 2-5, provides an alternative approach to secure computation that distributes sensitive data across multiple parties in a way that prevents any individual party from learning the original information [36]. Shamir's secret sharing scheme, based on polynomial interpolation over finite fields, enables a secret S to be divided into n shares such that any $t(1 \leq t \leq n)$ shares can reconstruct the original secret, while any $t-1$ or fewer shares reveal no information about the secret S [88]. This threshold property provides inherent fault tolerance and enables secure computation protocols that can withstand the compromise of up to $t-1$ participating parties.

The efficiency advantages of secret sharing-based secure computation include lower computational overhead compared to homomorphic encryption and the ability to leverage parallel processing across multiple computing parties [89]. Secure multi-party computation protocols combine secret sharing techniques with cryptographic primitives to enable complex computational workflows while maintaining privacy and correctness guarantees [90]. Information-theoretic security models assume computationally unbounded adversaries but require honest majorities among participating parties [91]. Cryptographic security models allow for dishonest majorities but rely on computational assumptions such as the hardness of discrete logarithm or factoring problems [78]. Practical implementations must balance security requirements with performance considerations and deployment constraints specific to each application domain.

Recent developments in secure computation have focused on improving practical efficiency and expanding application domains beyond traditional cryptographic settings. Emerging trends in secure computation continue to address scalability challenges and expand compatibility with modern computing infrastructures. The future evolution of secure computation promises continued improvements in efficiency, usability, and application scope. The convergence of

secure computation with other privacy-preserving technologies, such as differential privacy and k-anonymity, creates opportunities for comprehensive privacy protection frameworks that can address diverse security and privacy requirements.

2.5 Summary

The extensive review of privacy-preserving data analysis literature reveals several critical gaps that need innovative research approaches to address emerging challenges in data protection. These gaps represent fundamental limitations in current methodologies that hinder the practical deployment of privacy-preserving technologies in real-world environments where data utility requirements must be balanced against stringent privacy protection demands.

The first significant gap concerns the fundamental trust assumptions underlying existing differential privacy implementations. Current central differential privacy approaches require data owners to place complete trust in data curators who gain unrestricted access to sensitive information during collection and processing phases, whereas local differential privacy eliminates trust requirements but introduces substantial utility degradation that renders many statistical analyses impractical for real-world applications. The lack of effective solutions that can provide central differential privacy utility without requiring trust in data curators represents a fundamental limitation that impedes widespread adoption of formal privacy protection mechanisms.

The second critical research gap is the absence of adaptive frameworks that can optimize privacy-preserving computations across heterogeneous computing environments. Information infrastructures typically feature hierarchical network architectures with varying computational capabilities, from resource-constrained edge devices to powerful cloud computing centers. The lack of intelligent frameworks that can dynamically select and combine different secure computation methods based on available resources represents a significant limitation that prevents optimal utilization of distributed computing infrastructures for privacy-preserving analytics. In addition, the computational efficiency limitations of existing secure computation approaches represent a substantial barrier to practical deployment in resource-constrained environments. Current homomorphic encryption implementations impose computational overhead that can increase processing times by several orders of magnitude compared to plaintext operations, making them impractical for many real-time applications. The lack of hybrid approaches that can intelligently combine different secure computation methods to optimize both security and efficiency limits the practical applicability of privacy-preserving technologies.

While privacy-preserving techniques for structured data have achieved significant development through well-established frameworks like differential privacy, k-anonymity, and secure computation, the transition to unstructured data processing presents fundamentally

different challenges that existing methodologies cannot adequately address. Structured data anonymization benefits from clearly defined schemas, standardized data types, and predictable attribute relationships that enable systematic application of generalization and suppression techniques. However, unstructured textual data, particularly in healthcare domains, contains complex linguistic patterns, contextual dependencies, and implicit semantic relationships that resist traditional anonymization approaches designed for tabular formats.

Current approaches to unstructured data anonymization, particularly clinical text processing, suffer from fundamental inadequacies that limit their practical applicability in healthcare environments. Existing text de-identification systems primarily focus on simple identifier removal without providing formal privacy guarantees or considering the complex semantic relationships in narrative clinical records. The integration of advanced natural language processing capabilities with rigorous privacy models remains largely unexplored, creating a significant research gap. This gap is particularly critical given the increasing importance of clinical text analytics for medical research and the strict regulatory requirements governing healthcare data protection.

The accelerating digitization of healthcare systems and the expansion of data-driven applications create opportunities for improving public services and health outcomes through data analytics. The emergence of advanced machine learning capabilities creates additional urgency for developing effective privacy-preserving solutions. Large language models and sophisticated analytics systems can extract subtle patterns and correlations from data that were previously undetectable, creating new privacy risks while also offering new opportunities for privacy-preserving data analysis. The practical significance of these research gaps and the inability of current approaches to address these real-world requirements represents a critical barrier to realizing the full potential of data-driven innovation in these essential domains.

Chapter 3 Differential Privacy Framework on Untrusted Servers

3.1 Motivation

The rapid development of data-driven systems has completely changed how organizations collect, process, and utilize data to enhance quality of life across multiple domains including healthcare analytics, financial services, telecommunications, and digital governance. These technological advancements make use of sophisticated sensing networks, advanced analytics, and real-time decision-making systems to provide more responsive and personalized services. However, the widespread data collection methods involved in these applications have introduced significant privacy challenges that threaten individual autonomy and undermine public trust in data-sharing initiatives, creating barriers to realizing the full potential of collaborative data analytics.

Privacy breaches in information systems have demonstrated the critical importance of robust privacy protection mechanisms. Historical incidents, such as the Netflix Prize dataset de-identification attack, have revealed that traditional techniques—including the removal of personally identifiable information—are insufficient to prevent sophisticated privacy attacks that take advantage of auxiliary information and statistical correlations. These incidents revealed the urgent need for privacy-preserving technologies that can provide mathematically rigorous guarantees against privacy violations while maintaining data utility for legitimate analytical purposes.

Differential privacy has emerged as the industry standard for privacy-preserving data analysis, offering provable protection against the strongest possible adversaries with arbitrary background knowledge. This framework quantifies privacy loss through carefully calibrated noise injection, ensuring that the presence or absence of any individual's data cannot be reliably determined from analytical outputs. The differential privacy paradigm encompasses two primary deployment models: central differential privacy (CDP) and local differential privacy (LDP), each presenting distinct trade-offs between privacy guarantees, data utility, and trust assumptions.

Central differential privacy relies on trusted data curators to collect raw data and apply privacy-preserving transformations during query processing. While this approach produces superior data utility compared to local alternatives, it requires users to entrust their unmodified data to potentially untrusted third parties—a significant barrier to adoption in many real-world

scenarios. Conversely, local differential privacy eliminates trust requirements by having users apply privacy mechanisms before data transmission but resulting in substantially reduced analytical accuracy due to increased noise requirements.

This section addresses the fundamental tension between privacy protection and data utility by proposing a novel framework that combines the privacy advantages of cryptographic techniques with the analytical benefits of central differential privacy. This hybrid approach represents a significant advancement in privacy-preserving data analytics for data-driven applications and other sensitive data processing scenarios.

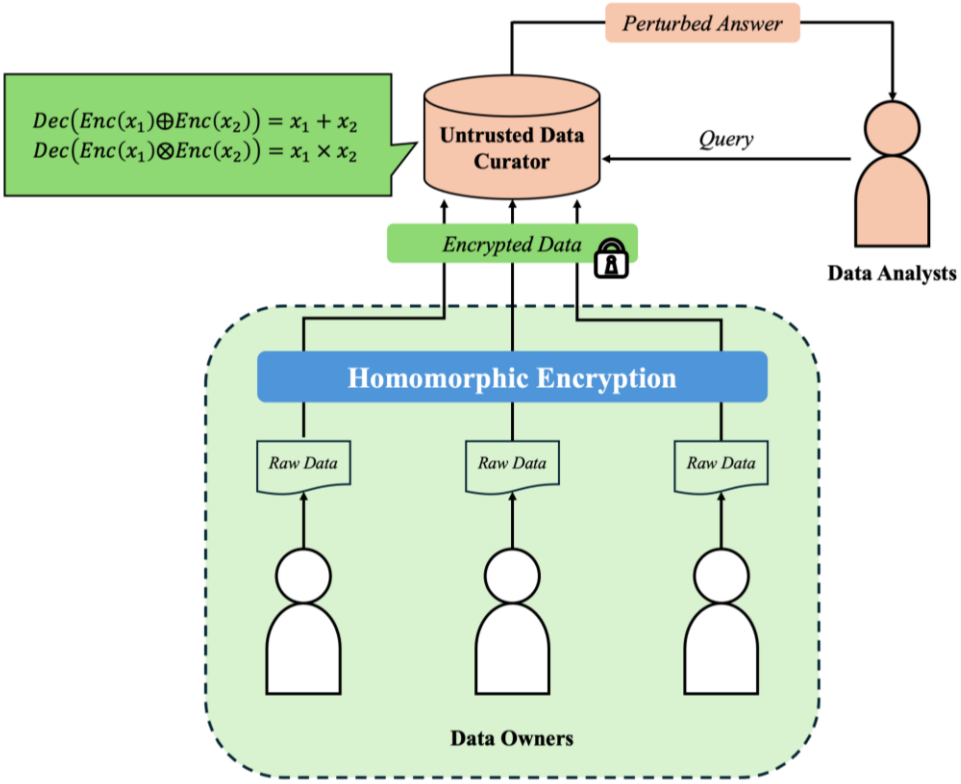


Figure 3-1 Differential Privacy Model on Untrusted Servers.

3.2 System Model and Threat Model

Our proposed system operates within a multi-party computational environment consisting of three primary entities, each with distinct roles, capabilities, and security assumptions:

- Data Owners represent individual users or organizations that possess sensitive datasets requiring protection during analytical processes. These entities are assumed to be honest but privacy-conscious, meaning they are willing to participate in collaborative analytics but require strong guarantees that their raw data remains confidential. Data Owners have limited computational resources compared to centralized servers but can perform basic

cryptographic operations such as encryption and decryption. They maintain exclusive control over their private keys and have the capability to verify the integrity of received results through cryptographic protocols.

- **Data Curator** serves as the central computational entity responsible for performing privacy-preserving analytics on encrypted datasets. This entity is considered semi-honest or honest-but-curious, meaning it follows the prescribed protocol correctly but may attempt to infer sensitive information from observed data patterns, intermediate computations, or metadata. The data curator possesses substantial computational resources and advanced cryptographic capabilities, including homomorphic encryption operations and differential privacy mechanisms. Importantly, the curator never gains access to plaintext data, working exclusively with encrypted inputs and producing encrypted outputs.

- **Data Analysts** represent legitimate third parties seeking to extract statistical insights from aggregated datasets for research and service optimization purposes. These entities are assumed to be honest and operate within the bounds of approved analytical queries. Analysts receive differentially private results that provide statistical utility while maintaining individual privacy guarantees. They do not have direct access to raw data or intermediate computational states.

The system architecture, as shown in Figure 3-1, is designed to facilitate privacy-preserving statistical analysis while maintaining computational efficiency and practical deploy ability in real-world scenarios.

The threat model assumes that adversaries may include compromised data curators, malicious analysts attempting to exceed their authorized access, or external attackers seeking to breach the system through various attack vectors including collusion between multiple parties. Our security model specifically addresses scenarios where the data curator cannot be fully trusted, representing a significant advancement over traditional centralized approaches.

3.3 The Proposed Scheme

3.3.1 Design Goals

(1) Privacy Protection with Formal Guarantees

The fundamental design principle of our scheme is that the system must provide mathematically provable privacy protection that remains robust against sophisticated adversaries with extensive background knowledge. We achieve this through the combination of differential privacy mechanism that ensures statistical indistinguishability of datasets differing by

a single record, and cryptographic protection of raw data throughout the computational pipeline.

The privacy framework addresses both input privacy, which protects individual data records from exposure to untrusted parties, and output privacy ensuring that analytical results do not reveal sensitive information about specific individuals. Input privacy is achieved through homomorphic encryption technique that enables additional and multiplicative operations on encrypted data without requiring decryption at the curator level. Output privacy is guaranteed through carefully calibrated noise injection following the Laplace mechanism, with noise parameters determined by sensitivity analysis and privacy budget allocation.

(2) Maximizing Data Utility Under Privacy Constraints

Our design seeks to optimize the privacy-utility trade-off by leveraging the superior noise characteristics of central differential privacy while eliminating the trust requirements typically associated with such approaches because excessive noise injection can render analytical results useless for practical applications which represents a critical challenge in privacy-preserving systems.

The utility optimization strategy involves several key components: efficient sensitivity calculation for common statistical queries, adaptive noise scaling based on dataset characteristics, and post-processing techniques that preserve differential privacy guarantees while improving result accuracy. By performing computations on aggregated encrypted data rather than individual encrypted records, our approach achieves noise levels comparable to trusted centralized systems while maintaining cryptographic security.

3.3.2 Scheme Implementation

The data flow of the process is shown in Figure 3-2 and corresponds to the following steps:

1. The data owners encrypt confidential information.

2. The data owners send ciphertext and a public key to the data curator.

3. The data curator performs secure computing in the encrypted domain. During this process, the data curator may request computational support several times and use the results of the computational support.

4. The data curator sends the perturbed result (ciphertext) to the data owners.

5. The data owners decrypt the perturbed result.

6. The data owners send the perturbed result (plaintext) to the data curator.

7. The data curator stores the perturbed result in a database.

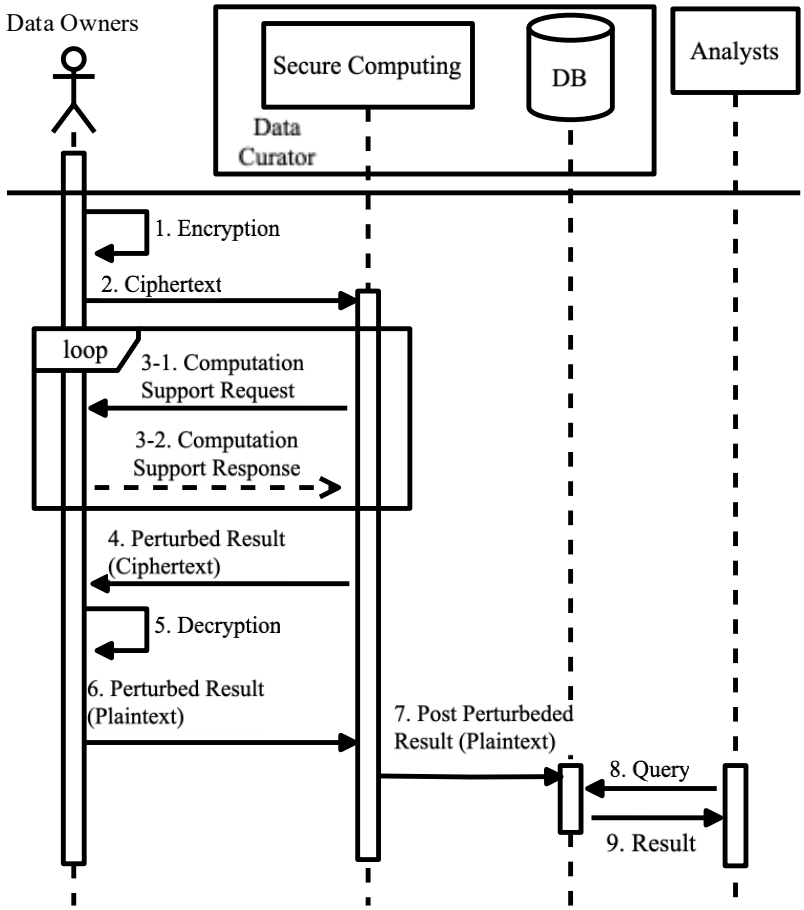


Figure 3-2 Data Flow of the Proposed Scheme.

8. The analysts send queries to the data curator.

9. The data curator responds with perturbed results.

The secure computing process consists of the following steps, as illustrated in Figure 3-3.

First, the data curator receives encrypted datasets from data owners. Then, the data curator calculates the statistics of the datasets (mean and standard deviation) in the encrypted domain using secure-computing methods. Using the mean μ_x and standard deviation σ_x of variate X, all datasets can be standardized using the following equation:

$$X_i' = \frac{X_i - \mu_x}{\sigma_x} \tag{3-1}$$

Then, the data curator adds Laplace noise to perturb the datasets.

Laplace distribution is defined as follows:

$$Lap(x|b) = \frac{1}{2b} e^{-\frac{|x|}{b}} \tag{3-2}$$

where b is a scale parameter.

l_1 sensitivity Δ_f is defined as follows:

$$\Delta_f = max_{D,D'} \parallel f(D) - f(D') \parallel_1 \tag{3-3}$$

When the privacy budget ϵ is fixed, the greater the sensitivity Δ_f, the larger the scale parameter b; and the larger the scale parameter b, the greater the added noise.

In the Laplace mechanism, when b is set to be $\frac{\Delta_f}{\epsilon}$, ϵ–DP is satisfied.

The output value is $f(D) + r$, where r is Laplace noise. To generate Laplace noise, the process requires the number of input data and the domain of attributes to calculate the sensitivity. After obtaining the encrypted Laplace noise Lap_{μ_x} and Lap_{σ_x} of the mean μ_x and standard deviation σ_x of variate X, the perturbed output can be calculated using the following equation:

$$Y = X' \times \left(\sigma_x + Lap_{\sigma_x}\right) + \left(\mu_x + Lap_{\mu_x}\right) \tag{3-4}$$

Considering the accuracy of the Laplace mechanism, when y is an output of the Laplace mechanism, the following inequality exists:

$$\forall \delta \in (0,1], \Pr\left(\parallel y - f(D) \parallel_1 > \frac{\Delta_f}{\epsilon} ln \frac{1}{\delta}\right) \leq \delta \tag{3-5}$$

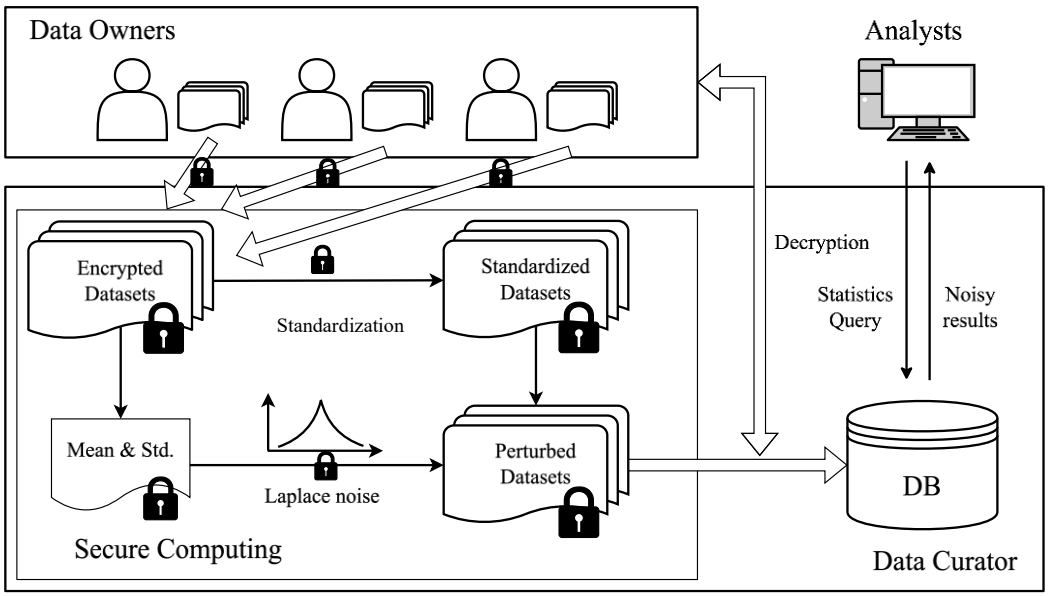


Figure 3-3 An Overview of the Proposed Scheme.

After generating the perturbed datasets in the encrypted domain, the data curator sends them back to the data owners that hold the private key. The data owners then decrypt the ciphertext and outsource it to the database of the data curator for sharing and analysis.

3.4 Experimental Evaluation

3.4.1 Experimental Settings

The experiments were run on a 1.4 GHz Intel i5 Core CPU processor with 16 GB of memory using the macOS operating system. It simulates the process of the data curator to perturb datasets to achieve DP in the encrypted domain, so that the key generation and encryption steps on the data users' side are performed in advance. In addition, the Laplace noise is generated in the plain domain.

	Age	Count	Noisy Result
0	36	1348	1349.997471
1	35	1337	1338.247839
2	33	1335	1335.977242
3	23	1329	1332.566639
4	31	1325	1325.431502
5	34	1303	1304.153552
6	28	1280	1282.697891
7	37	1280	1280.500949
8	30	1278	1276.860269
9	38	1264	1266.332147
10	32	1253	1252.912741
11	41	1235	1234.531724
12	27	1232	1233.753200
13	29	1223	1226.013263
14	24	1206	1209.036285

Figure 3-4 The Result of Query for Count.

Suppose the original dataset is a numerical dataset, the number of data records in the dataset is n, and the range of numerical data records is N. According to Formula (3-2), scale parameter b is indispensable for generating Laplace noise. In Laplace mechanism, b is set to be $\frac{\Delta_f}{\epsilon}$, and different query function has different l_1 sensitivities. For example, the l_1 sensitivity of the mean Δ_{mean} can be calculated as follows:

$$\Delta_{mean} = \frac{N}{n} \tag{3-6}$$

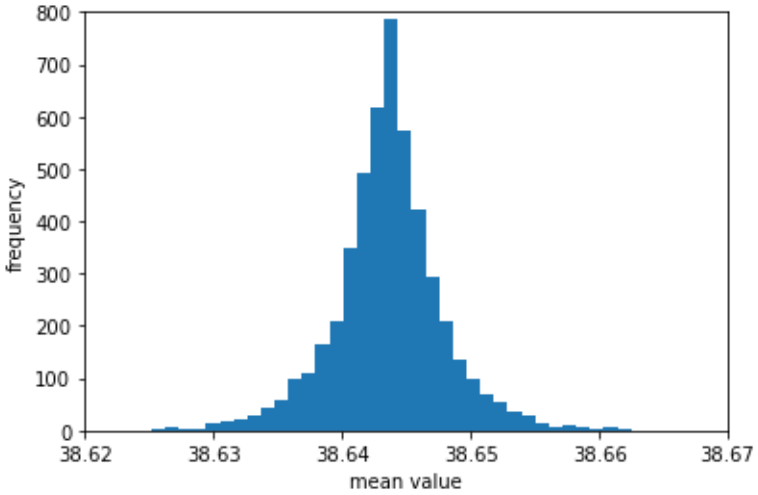


Figure 3-5 Distribution of Mean Values ($\epsilon = 0.5$).

Given privacy parameter ϵ, Laplace noise can be generated in advance. Then, the ciphertext of these data is sent to the data curator to perturb the encrypted dataset to satisfy differential privacy.

3.4.2 Performance of Privacy Protection

In this section, the experimental results show that the proposed scheme satisfies DP. The original dataset contains the age of 48,842 people, and the age range is 17-90 years.

Figure 3-4 shows an example of a count query of the dataset (only showing the first 15 records). If the analysts query the number of people of each age, the original dataset returns the exact number, and the proposed scheme returns the perturbed results with Laplace noise so that it can resist any background attacks.

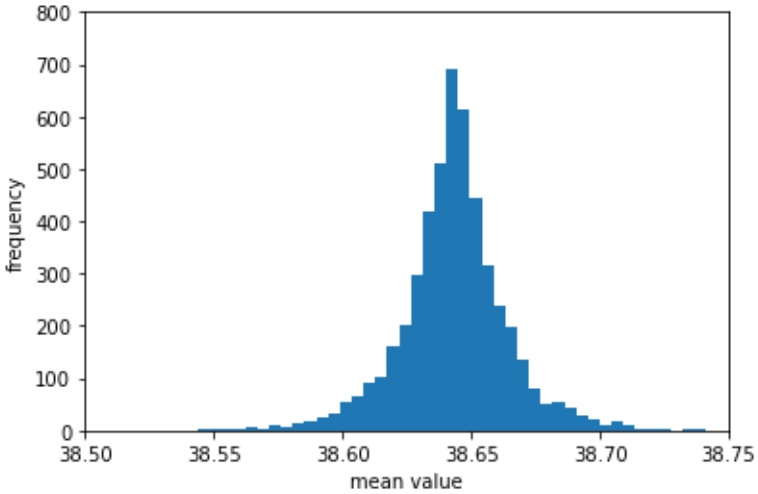


Figure 3-6 Distribution of Mean Values ($\epsilon = 0.1$).

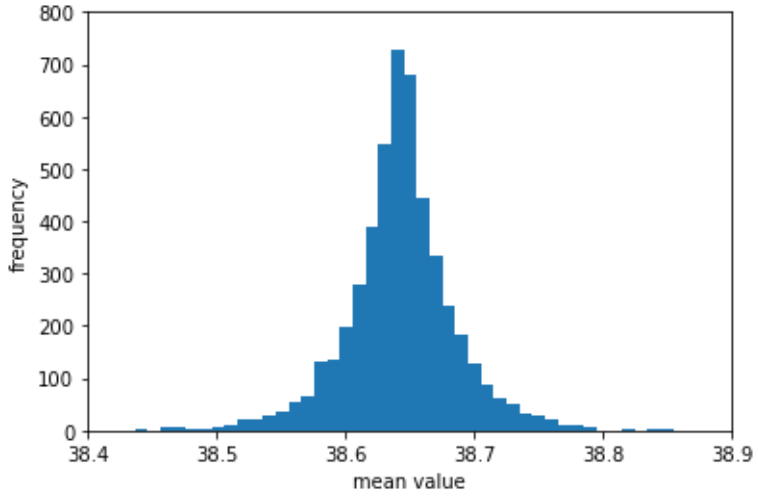


Figure 3-7 Distribution of Mean Values ($\epsilon = 0.05$).

To visually evaluate the privacy preservation performance, the privacy parameter ϵ was set to different values and Laplace noise was added to the mean of the original dataset using secure-computing methods. The process was run 5000 times and Figure 3-5 to Figure 3-8 show histograms that represent the distribution of mean values under the Laplace mechanism. The mean value of the original dataset was 38.644, and the dataset consists of 48,842 records. As shown in the histograms, the smaller the privacy parameter ϵ, the more the added noise, resulting in higher privacy protection.

3.4.3 Performance of Utility Maintenance

In practical applications, according to different scenarios and requirements, setting a reasonable value for privacy parameters to achieve a balance between privacy protection and data utility is a key issue in the application of the DP technology. This section shows that the

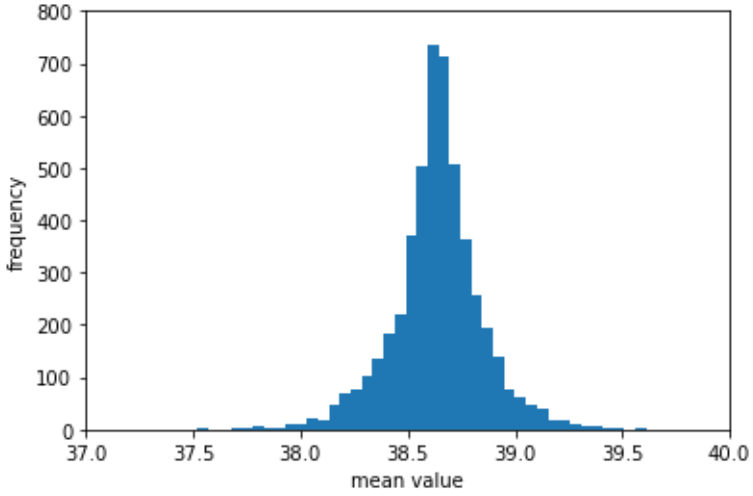


Figure 3-8 Distribution of Mean Values ($\epsilon = 0.01$).

proposed scheme has high data utility.

As introduced before, the more the noise added, the higher the degree of privacy protection that is guaranteed, which results in a decrease in data utility. Therefore, the data utility can be evaluated by calculating the Laplace noise. In the proposed scheme, a central data curator adds Laplace noise to the datasets using secure computing. By contrast, in LDP models, data users add noise separately before aggregation. Therefore, the original dataset was divided into different number of parts to simulate the LDP models and compared them with the proposed scheme.

The original dataset contains 48,842 records, which were separated into two, four, and eight pieces. The process was run 5000 times with the same privacy parameter $\epsilon = 0.5$. Figure 3-9 to Figure 3-12 show the amount of Laplace noise added under different settings. It can be seen that with an increase in data pieces, it is necessary to add more noise to maintain the same

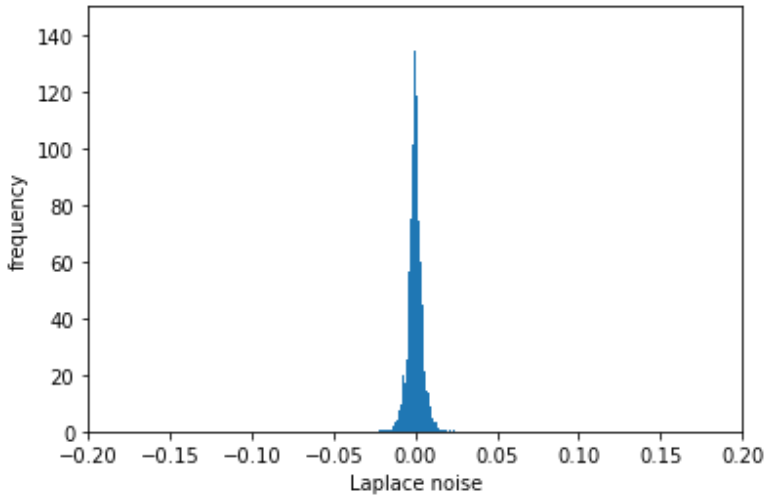


Figure 3-9 Distribution of Laplace (one data user).

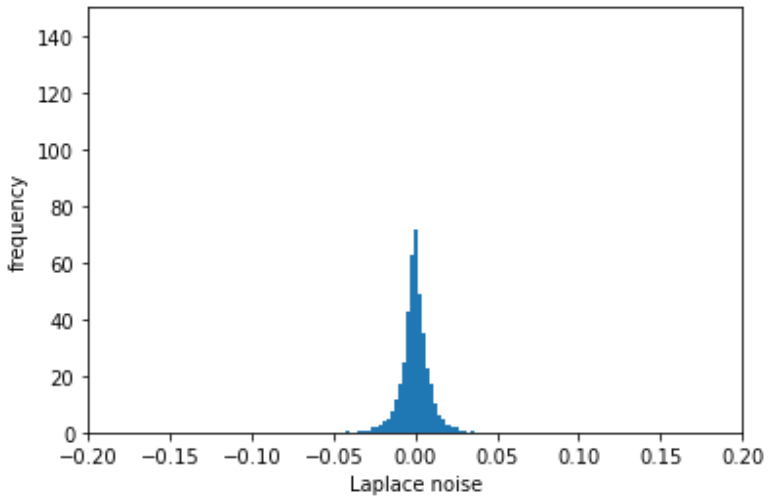


Figure 3-10 Distribution of Laplace (two data users).

degree of privacy preservation, which leads to the deterioration of data utility.

3.5 Summary

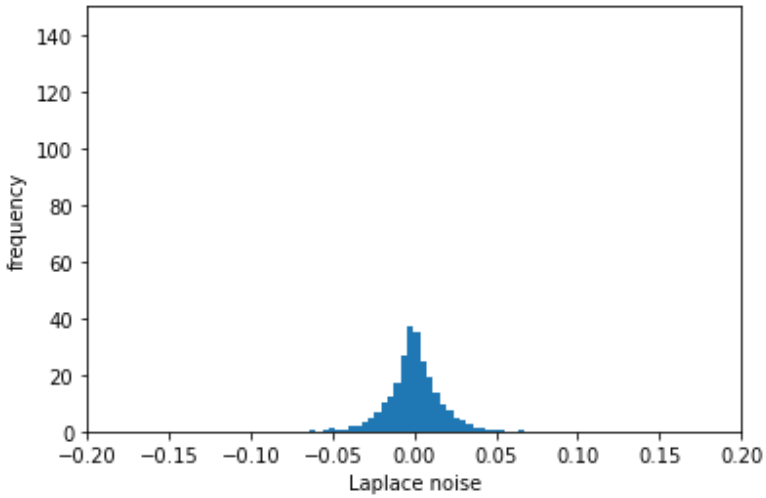


Figure 3-11 Distribution of Laplace (four data users).

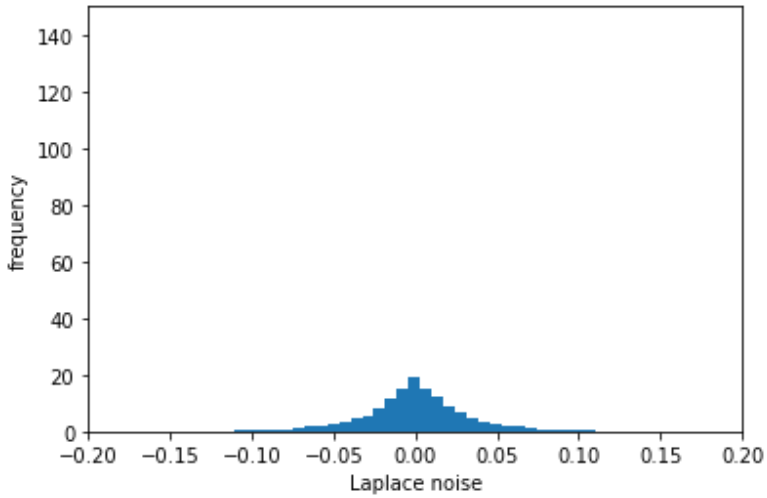


Figure 3-12 Distribution of Laplace (eight data users).

This chapter presented a novel privacy-preserving framework that combines cryptographic techniques with differential privacy to enable secure statistical analysis on sensitive datasets without requiring trusted third parties. The proposed scheme takes advantage of homomorphic encryption to perform differential privacy computations entirely in the encrypted domain, achieving the utility benefits of centralized approaches while eliminating trust requirements that have historically limited practical adoption.

The experimental evaluation validates both the privacy guarantees and utility advantages of the framework, demonstrating superior performance compared to local differential privacy approaches while maintaining rigorous cryptographic security. The results confirm that the

scheme provides a practical solution for privacy-preserving analytics in data-driven applications and other sensitive data processing scenarios, representing a significant advancement in the field of privacy-preserving data analysis.

Chapter 4 Hierarchical k-Anonymization over Encrypted Data

4.1 Motivation

The development of IoT and information infrastructures has generated volumes of sensor data that hold tremendous value for urban planning, public safety, healthcare monitoring, and intelligent service delivery [92][93]. These systems rely on extensive networks of sensors, mobile devices, and monitoring equipment that continuously collect personal and environmental data, creating rich datasets to enhance quality of life. However, the meaningful sharing and utilization of such data is severely constrained by privacy concerns and regulatory requirements that demand strong protection of individual privacy [94][95].

The emergence of edge computing architectures, as a crucial complement to traditional cloud computing infrastructure, demonstrates enormous potential. Edge computing architectures push computational capabilities, data storage, and processing power closer to data sources rather than relying exclusively on remote cloud data centers [21][96]. This distributed approach offers significant advantages including reduced network latency, improved system responsiveness, enhanced bandwidth efficiency, and better support for real-time applications. However, the hierarchical nature of edge computing environments also introduces new challenges for privacy-preserving data processing, particularly when data needs to be anonymized before sharing across different computational tiers.

Traditional data anonymization approaches typically assume trusted processing environments requiring access to plaintext data during the anonymization process. Sensitive information remains exposed to potential attacks or unauthorized access in such methods. In edge computing scenarios, this assumption becomes problematic due to the distributed nature of data sources and the heterogeneous computational capabilities across different network layers.

The multi-tiered architecture inherent in edge computing systems presents both opportunities and challenges for privacy-preserving data processing. At the device layer, sensors and IoT devices generate continuous streams of potentially sensitive data but possess extremely limited computational resources. The edge layer, consisting of local edge servers and gateways, provides intermediate processing capabilities that can handle data preprocessing and initial anonymization tasks while maintaining proximity to data sources. The cloud layer offers substantial computational resources suitable for complex anonymization operations and higher-level data

analytics. This hierarchical structure creates an opportunity to design efficient privacy-preserving frameworks that leverage the unique characteristics of each layer while maintaining end-to-end data protection.

Current anonymization techniques face significant limitations when applied to edge computing environments. The challenge becomes more acute when considering the need for k-anonymity, one of the most widely adopted privacy models that requires grouping records into clusters where each cluster contains at least k indistinguishable records [24][58]. Achieving meaningful k-anonymity typically requires access to substantial datasets and significant computational resources for clustering and generalization operations. In edge computing environments, individual edge nodes may not have access to sufficient data volumes to achieve effective k-anonymization locally, while transmitting raw data to cloud layers for centralized processing violates fundamental privacy principles.

Recent advances in cryptographic techniques offer promising foundations for addressing these challenges. Homomorphic encryption enables computation over encrypted data without requiring decryption, while secret sharing allows distributed computation across multiple parties without revealing sensitive information to individual participants [26][97]. However, the practical integration of these techniques in hierarchical edge computing environments remains an open research challenge that requires careful consideration of computational constraints, network topology, and security requirements across different system layers.

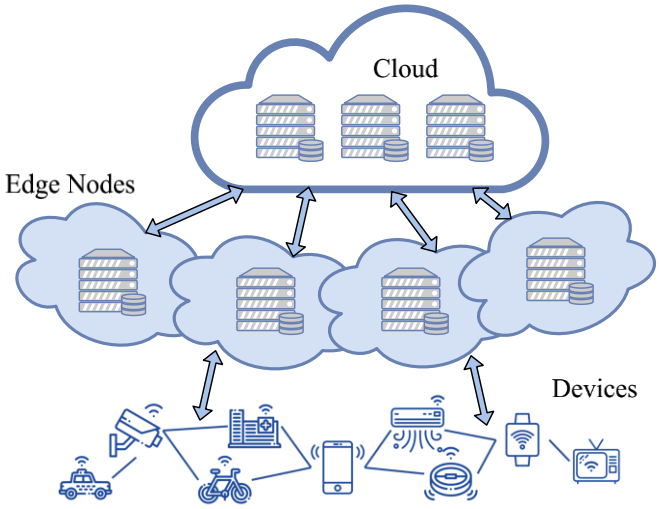


Figure 4-1 Edge Computing Environment.

This research is motivated by the critical need to develop practical and efficient privacy-preserving anonymization frameworks specifically designed for hierarchical edge computing environments as shown in Figure 4-1. Such frameworks must effectively leverage the unique characteristics of each computational tier—from resource-constrained edge devices to powerful

cloud infrastructure—while maintaining strong privacy guarantees and computational efficiency. By addressing these challenges, we aim to enable secure and privacy-preserving data sharing in IoT and smart city applications while preserving the performance benefits of edge computing architectures.

4.2 System Model and Threat Model

4.2.1 System Model

This study assumes an environment in which computational resources are arranged hierarchically, and secret computations are used to perform k-anonymization processing based on k-member clustering. The topology of the network is hierarchical, and computational resources for anonymization services exist at each level. The assumed environment includes the following areas and entities:

- Global domain: A network domain in which several computational entities can perform anonymization processes, and there is sufficient space to distribute shares when secret sharing is used. For example, a cloud domain can be treated as a global domain.
- Local domain: A network domain in which few computational entities can perform anonymization processes and insufficient entities to distribute shares when secret sharing is used. The edge domain corresponds to this domain.
- Data Owners: Data owners possess confidential information and provide data for various services. Data owners enter into a contract in advance with a service provider that stipulates the type of data to be provided, the type of anonymization, and privacy standards for anonymization results.
- Computation Units: Computation units provide large storage and assist data owners in anonymizing encrypted datasets.

In the local domain, the number of computational resources is small, and there are insufficient distributed locations for computation when secret sharing is employed. The required computational resources are not very large because the amount of data is similar but not so large. Therefore, homomorphic cryptography is used for secret computations. However, in the global domain, the number of computational operators is large and widely spread. Because the data are gathered and their size increases, the required computational power also increases significantly in this domain. Secret sharing, which requires less processing time per operation than homomorphic cryptography, can reduce the time and cost, and is more feasible because the amount of data is larger than that in the local domain. Thus, the two secret computations were used together to achieve secure k-anonymization and reduce the overall computational

cost.

The degree of generalization is based on the number of computational resources in a domain. Therefore, it is assumed that the local and global domain anonymization processes can be linked. Consequently, anonymization results from the local domain are collected in the global domain and anonymized more robustly than anonymized data in the local domain. This allows each service provider to use data that meet the appropriate privacy criteria. In addition, the processing time can be reduced compared with the time required for anonymization in the global domain.

The rules for separating the global domain from the local domain depend on the data to be anonymized and the number of data owners of the computing resources. These rules must be defined in advance. An example of this is as follows: Let N_D be the number of owners of the computing resources that can be anonymized in the domain. In this case, the following criteria were used to select the secret computation method:

(1) Homomorphic encryption is used when

$N_D = 1$;

$N_D \geq 2$ and most of the calculated resources are possessed by the same entity.

(2) Secret sharing is used in cases other than those described above.

When $N_D = 1$, it is impractical to distribute to multiple computing entities with secret sharing. And when $N_D \geq 2$ but most of the calculated resources are possessed by the same entity, it is inappropriate to apply secret sharing in terms of resistance to collusion attacks. Therefore, the criteria for choosing which secure computation method to use is defined above.

4.2.2 Threat Model

In this privacy-preserving hierarchical anonymization framework, a multi-tier threat model that reflects the distributed nature of edge computing environments is defined as follows:

- It is assumed that data owners at the device layer are trusted entities who possess sensitive data and perform encryption operations in secure environments. However, computational entities at both the local domain (edge layer) and global domain (cloud layer) are considered semi-honest adversaries who follow the prescribed protocols but may attempt to infer sensitive information from the encrypted data they process. These computational units may collude within their respective domains but are assumed not to collude across domain boundaries due to administrative and organizational separation.
- External adversaries who may intercept network communications or attempt to

compromise computational infrastructure through various attacks are also considered. The threat model assumes that cryptographic primitives (homomorphic encryption and secret sharing) provide computational security against polynomial-time adversaries, and that a threshold number of computational units in the global domain remain honest to ensure the security of secret sharing protocols.

The proposed framework aims to protect against these threats while maintaining the utility and efficiency required for practical deployment in edge computing environments.

4.3 The Proposed Scheme

4.3.1 Workflow

This subsection describes the workflow in the proposed anonymization process. A sequence diagram of a data owner and computation subject using homomorphic encryption is shown in Figure 4-2.

The workflow of the anonymization process shown in Figure 4-2 corresponds to the following steps:

1. The data owner sends data records to the computation unit.

2. The computation unit determines the secret computation.

3. The computation unit sends the secret computation context to the data owner.

4. The data owner performs key generation, etc.

5. The data owner encrypts confidential information.

6. The data owner sends ciphertext and a public key to the computation unit.

7. The computational unit performs the anonymization process. During this process, the computational unit may request computational support several times and use the results of the computational support for the anonymization process.

8. The computation unit sends the anonymization result (ciphertext) to the data owner.

9. The data owner decrypts the anonymization result.

10. The data owner sends the anonymization result (plaintext) to the computation unit.

11. The computation unit registers the anonymization result in a database.

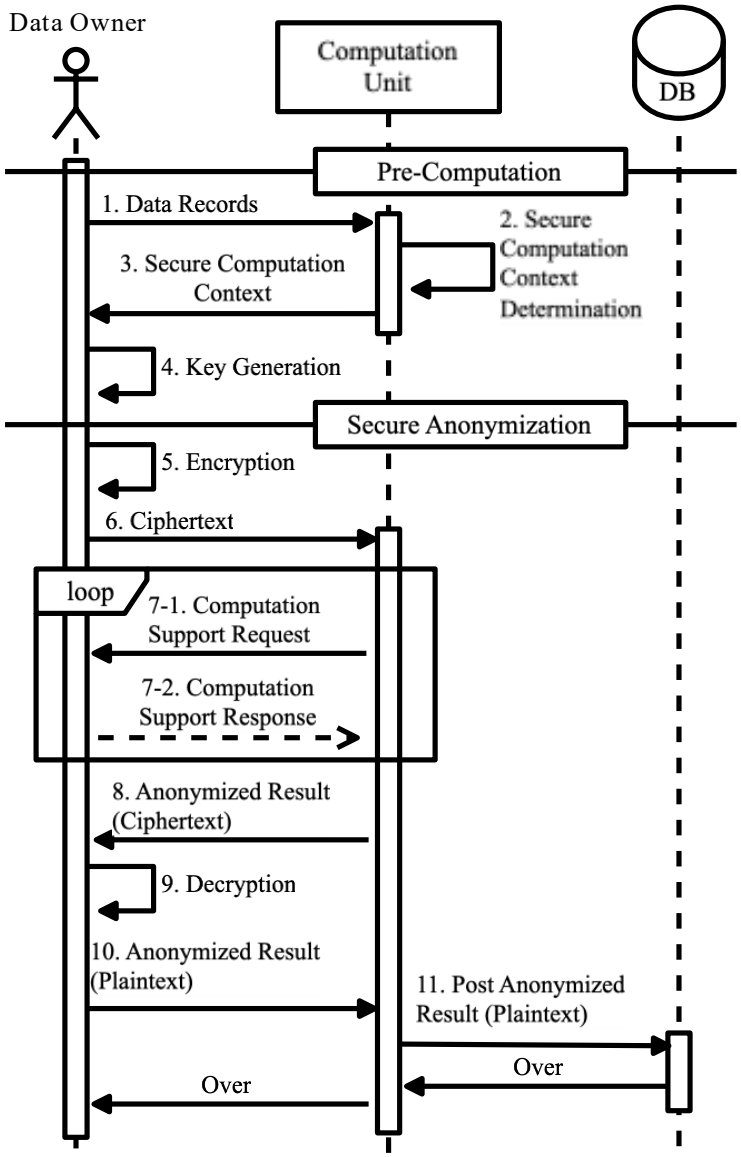


Figure 4-2 Workflow using Homomorphic Encryption.

Figure 4-3 shows the sequence diagram of the anonymization process using secret sharing. Compared with anonymization using homomorphic encryption, there is a controller that manages secret sharing computation. The data owner sends the data records to the controller to generate a secret computation context. The secret computation context includes an ID that identifies the secret computation process and destination of the share. Then, the data owner sends the secret computation context to each computation unit, and the following flow is the same as that using homomorphic encryption. When the local and global domains are linked, the computational entity of the local domain becomes the data owner of the global domain.

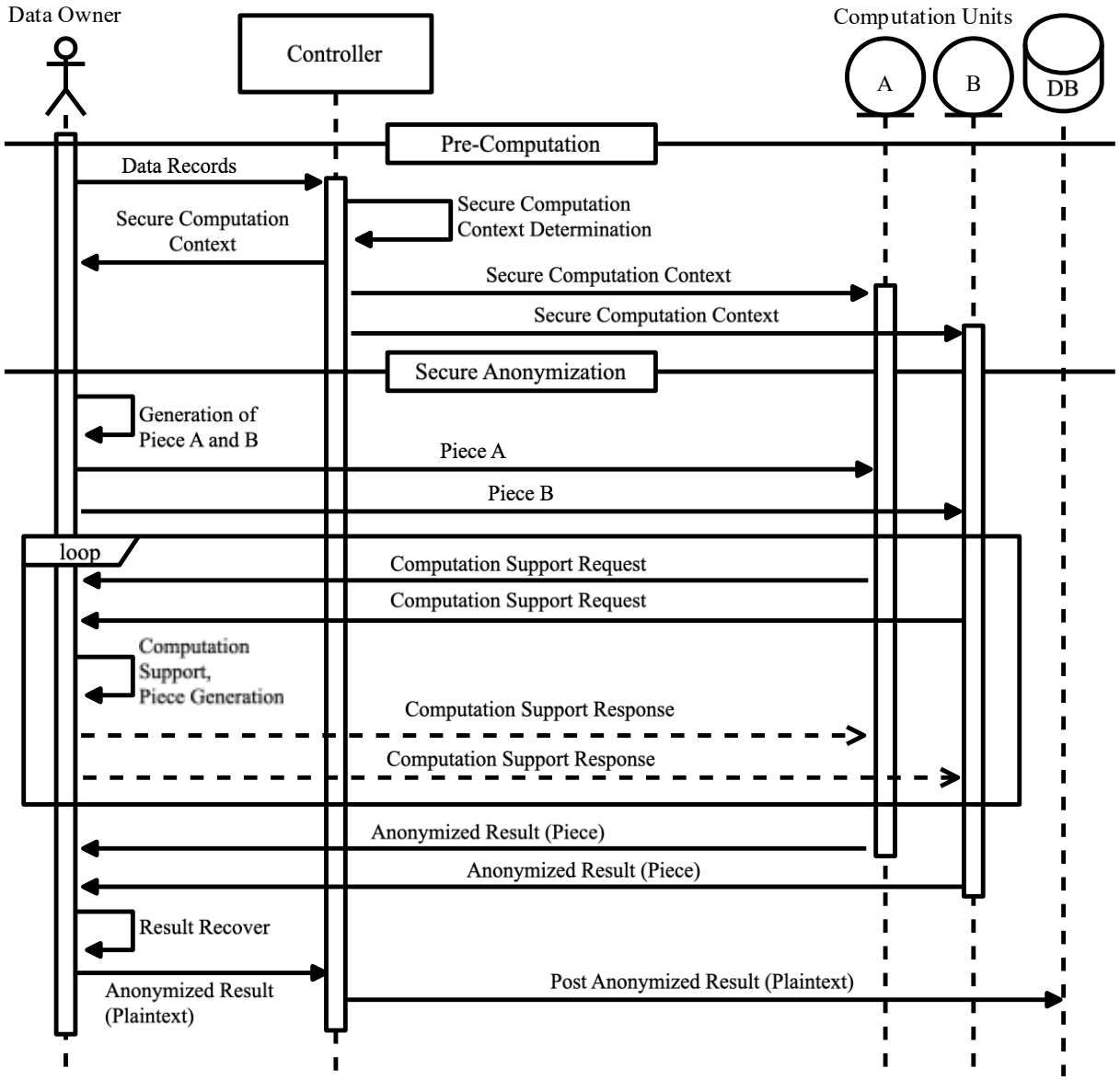


Figure 4-3 Workflow using Secret Sharing.

4.3.2 k-Anonymization Implementation

This section introduces the details of the k-anonymization implementation for structured tabular datasets.

The number of records in a dataset is denoted as N, and the number of quasi-identifiers is denoted as M. In the anonymization process, identifiers are deleted, and quasi-identifiers are transformed into a range. Records in the same q* block have the same quasi-identifiers as in

the definition of k-anonymity.

4.3.2.1 K-member Clustering

First, the data owner transforms and normalizes the numerical data values of the quasi-identifiers such that the original data values fall within the range of [0,1]. Specifically, the value x is converted to x' and normalized using the following formula:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4-1}$$

where x_{max} represents the maximum value of this dataset and x_{min} represents the minimum value of this dataset.

```
Function k_member_clustering(S, k)
Input: Array of data records S, anonymization parameter k
Output: Clusters each of which contains at least k records
1.  Initialize array clusters
2.  cluster_counts = floor (S, k)
3.  N = len(S)
4.  Initialize array used with the number of records N and
    value false
5.  rand_array = [0, .., N-1]
6.  Shuffle rand_array
7.  Initialize array rests
8.  for i, v in rand_array:
9.      if used[v]:
10.         continue
11.     used[v] = true
12.     Initialize array c and insert v
13.     record_list = find_best_record_index (S, used, v, k-1)
14.     Insert all elements of record_list to c
15.     Insert c into clusters
16.     if len(clusters) == cluster_counts:
17.         rests = rand_array[i+1:]
18.         break
19. for v in rests:
20.     if used[v]:
21.         continue
22.     used[v] = true
23.     reverse_used = reversed value of used
24.     r = find_best_record_index(S, reverse_used, c, 1)
25.     cluster = cluster containing r
26.     Insert v into cluster
27. return clusters
```

Figure 4-4 Pseudo Code of k_member_clustering.

The data owner encrypts the normalized data and sends them to the computation unit. The computational unit performs k-member clustering using the received data (ciphertext). Figure 4-4 shows the pseudo-code for k_member_clustering in the proposed method. First, the order of records is randomly determined. Subsequently, based on the order, if the chosen record is

unclassified, it is selected as a core record and added to the cluster. A record that is close to the core record is then selected and added to the cluster to generate a cluster containing k records after $k-1$ rounds. The maximum number of clusters that can be created when N records are classified to satisfy anonymity is $\lfloor N/k \rfloor$ records. Therefore, once $\lfloor N/k \rfloor$ clusters are created, the creation of new clusters is stopped. Then, unclassified records must identify the closest classified record and add to the cluster to which they belong.

Function find_best_record_index (*S*, *used*, *r*, *count*)

Input: Array of encrypted data records *S*, clustered flag array of records *used*, encrypted core record *r*, number of records to extract *count*
Output: Index of nearest *count* records to *r*

1. Initialize array *d* with size *N* and value 0
2. for i, v in *S*:
3. for m in range(*M*):
4. $d[i] = d[i] +_{enc} (dist(r[m], v[m]) \times_{enc} \frac{1}{M})$
5. $d[i] = Enc(1) -_{enc} d[i]$
6. If *used*[i] == *true*, then *d*[i] = 0
7. Initialize array *top_index*
8. for i in range(*count*):
9. *best_index* = *max_index*(*d*)
10. Insert *best_index* into *top_index*
11. *used*[*best_index*] = *true*, *d*[*best_index*] = 0
12. return *top_index*

Figure 4-5 Pseudo Code of find_best_record_index.

4.3.2.2 Find Closest Record

The target records must be selected to be added to the cluster. This process corresponds to find_best_record_index in Figure 4-5. First, the distance between the core record and each candidate record is calculated. The distance is calculated as the square of the difference between records for each quasi-identifier and the sum of the squares. In other words, the distance $dist(u,v)$ between records u and v is defined by the following equation:

$$dist(u,v) = \sum_{i=0,..,M-1} \left(a_{u,i} - a_{v,i}\right)^2 \quad (4\text{-}2)$$

In the fourth line of Figure 4-5, the distance is divided by M (multiplied by $1/M$) because it is necessary to set $0 \le d_i \le 1$ $(0 \le i \le M-1)$ for the convenience of the max_index in the ninth line.

Because the secret computation encrypts the result of the distance between the core record and each record, the smallest value is unknown. Therefore, this study used a calculation aid to obtain the minimum value in the array. However, if an array of distances is given to the private key holder for computational support, the original data may be exposed. max_index uses the

following conversion formula to convert the distances from x_i to y_i. The more iterations t is set, the closer the value approaches 0, 1.

$$y_i = \frac{x_i^{2^t}}{\sum_j^N x_j^{2^t}} \sim \begin{cases} 1\ (x_i = \max x) \\ 0\ (\text{otherwise}) \end{cases} \tag{4-3}$$

When the value of the distance is $0 \le d_i \le 1\ (0 \le i \le M-1)$, the index of the record with the smallest distance can be obtained by computing $x_i = 1 - d_i$. Because find_best_record_index extracts the closest records from a single core record, the distance to each record is calculated only once. Subsequently, max_index is executed with a mask applied to the classified record to prevent it from being selected as the record with the smallest distance, and the result is repeated to obtain the result with calculation support. This allowed us to extract the nearest records from the core records.

4.3.2.3 Generate Output Data

First, the range of quasi-identifiers in each q* block is determined to generate the output data, and confidential information is transformed into a list form. The range of quasi-identifiers in a q* block is determined as follows. Because the number of q* blocks is $\lfloor N/k \rfloor$, the number of values subject to computation support is $2M\lfloor N/k \rfloor$. During the process, several loops of computation requests and responses among data owner and all computation units were executed. These computation support steps must be performed in sequence because the intermediate result depends on the result of each computation support, whereas only one communication is required for output data generation because it allows for the independent calculation of all values. The method for converting the confidentiality information into a list format is to compile the confidentiality information of all records in the q* block into a list format. Subsequently, the output data of the anonymization process were generated.

4.3.2.4 Decryption and Recovery of Anonymized Results

Because all calculation results from secret calculations are encrypted, it is necessary to decrypt the anonymized results. Therefore, as shown in Figure 4-2, in steps 8 to 10, the anonymization results in encrypted form are sent to the data owner for decryption. However, if the data owner sends fake records back to the computation unit, it is unaware that the data owner has tampered with the result. This section discusses the proposed methods for detecting tampering during decryption.

In homomorphic cryptography, the computation unit can probabilistically detect tampering by adding dummy data to the anonymized cipher result. Instead of combining q* blocks into a single record, a dummy record was generated for each record in the original data table. The encryption method used for homomorphic encryption was a public-key cryptosystem; therefore, there were

no security problems even if the public key used for encryption was made public. The computation unit generated encrypted dummy data, inserted it into the anonymization result before the decryption request, and passed it to the private key holder. If the value of the dummy data was rewritten, it was certain that the data had been tampered. The random numbers added to the data records also prevented the data owner from guessing which data were dummy data from the decryption results. If only some of the records in a q* block had been tampered, k-anonymity was no longer satisfied when noise was removed from the decryption results and tampering could be detected.

For secret sharing, results can be restored if a certain number of shares are collected. Therefore, multiple computation units and data owners can share pieces after the anonymization process, enabling multiple people to recover the results. Each entity then sends its decryption results to the controller, which checks them against each other to determine if they have been tampered.

4.3.3 Privacy Guarantees

For the threat model, Data Owners are regarded as trusted parties who perform key generation, encryption, and decryption in a trusted environment, and Computation Units are regarded as untrusted parties that assist Data Owners in anonymizing encrypted data. Our proposed framework aims to protect data privacy when untrusted parties perform an anonymization process. In particular, it aims to ensure the privacy of sensitive data from the point at which the Data Owners send it to the Computation Units to the point at which any results are returned to the Data Owners.

Consider a malicious attacker may have access to encrypted data and even the infrastructure where computations are being performed but does not have the decryption key, or actively tries to disrupt the protocol, either by modifying the shares, providing incorrect information during the share distribution or reconstruction phase, or attempting to reconstruct the secret without authorization.

For the former scenario, HE ensures that computations on encrypted data do not reveal any information about the plaintext data apart from what is intended through the computation results. For the latter, a well-designed secret sharing scheme ensures that as long as the number of colluding curious parties is below a certain threshold (less than the minimum number required to reconstruct the secret), any information about the secret cannot be revealed.

4.4 Experimental Evaluation

The performance evaluation uses a personal PC to conduct the experiments to simulate the

Table 4-1 Experimental Setup.

OS	Ubuntu 20.04.3 LTS
CPU	Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz(20 cores)
RAM	DDR-4 395 GB
Homomorphic Encryption	Microsoft SEAL [98]
Secret Sharing	CrypTen [99]

local domain with limited computing resources and the global domain with sufficient computational subjects. The experimental environment is listed in Table 4-1.

4.4.1 Evaluation Criterion

The evaluation criteria are the execution time of the anonymization process and the information loss of the anonymization result. The following formula is used to calculate the information loss where the number of records is N, and the number of quasi-identifier attributes is M:

$$IL = \frac{1}{N} \sum_{i=1,\dots,N} \sum_{j=1,\dots,M} \frac{a_{i,j,max} - a_{i,j,min}}{|N_j|} \tag{4-4}$$

where $a_{i,j,max}, a_{i,j,min}$ are the upper and lower limits of the ranged values of the i-th record, j-th quasi-identifier. $|N_j|$ is the size of the domain of the j-th quasi-identifier.

4.4.2 Execution Time

This section describes the case in which only homomorphic cryptography is used (local

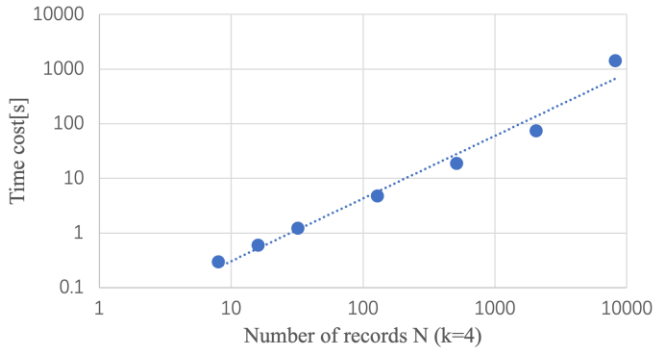


Figure 4-6 Execution Time with Different N (using Homomorphic Encryption).

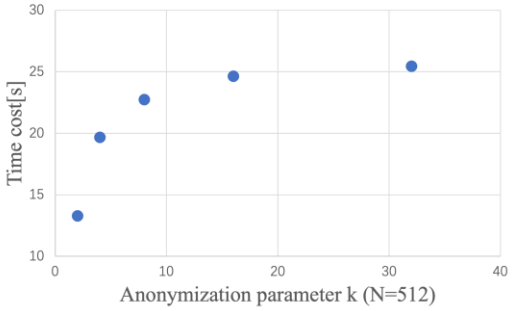


Figure 4-7 Execution Time with Different k (using Homomorphic Encryption).

domain) and only secret sharing is used (global domain). The execution time was measured from the time when encryption of the data table values began after key generation to the time when the anonymization process ended after decryption.

4.4.2.1 Execution Time using Homomorphic Encryption

First, the case of only using homomorphic cryptography was evaluated. Figure 4-6 shows the execution time for a different number of records when $k = 4$ is fixed, and Figure 4-7 shows the execution time for a different anonymization parameter k when $N = 512$ is fixed.

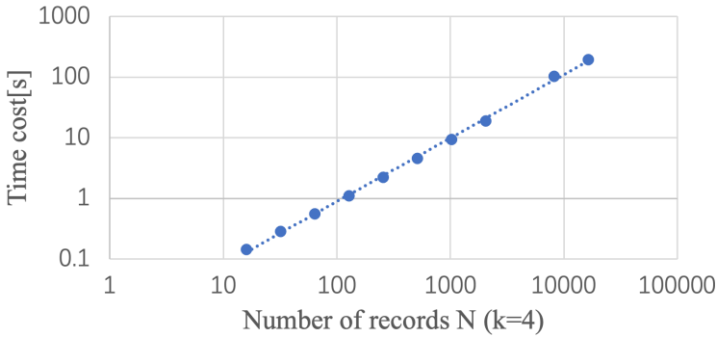


Figure 4-8 Execution Time with Different N (using Secret Sharing).

The execution time increased linearly with the number of records. In the anonymization process, one record was randomly selected from the unclassified records as the core record, and the process of determining a record that is close to the core record was executed $k - 1$ times. The number of times a core record is selected is equal to the number of q* blocks and is $[N/k]$ times, which is approximately N/k times. Therefore, because each q^* block generation involves $k - 1$ record selection operations and generates q^* blocks approximately N/k times, the time complexity of the anonymization process can be expressed using the following equation:

$$O\left(\frac{N}{k} \cdot (k-1)\right) = O\left(N - \frac{N}{k}\right) \quad (4\text{-}5)$$

Regarding the increase in execution time with different k, it can be observed that the execution time increases as k increases; however, the increase in execution time becomes milder. This can be explained by the fact that the time complexity of the anonymization process can be expressed by Equation (4-5).

4.4.2.2 Execution Time using Secret Sharing

Figure 4-8 shows the change in execution time for the anonymization process using secret sharing under different values of N when $k = 4$ was fixed, and Figure 4-9 shows the change in execution time under different k when $N = 512$ was fixed. It can be observed that the execution time increases with $O(N - \frac{N}{k})$, as in the case of homomorphic encryption. Compared to the homomorphic encryption method, the secret sharing method is generally 3.9 to 4.3 times faster than the homomorphic encryption method.

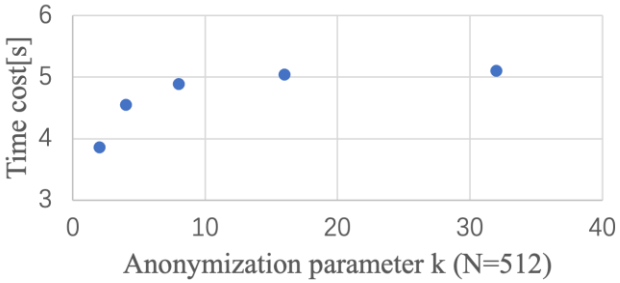


Figure 4-9 Execution Time with Different k (using Secret Sharing).

4.4.3 Information Loss

Figure 4-10 shows the information loss with a different number of records N when $k = 4$, and Figure 4-11 shows the information loss with a different k when $N = 512$. The information loss with the number of records shows that the information loss decreased as the number of records

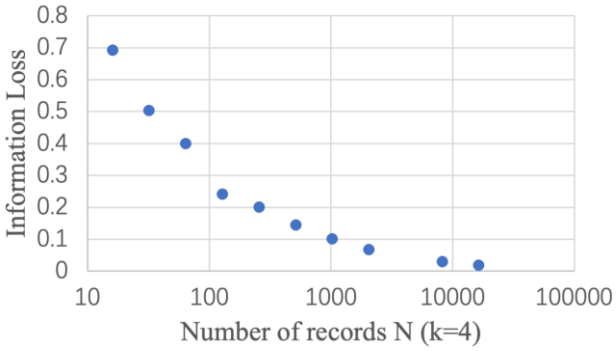


Figure 4-10 Information loss with different N.

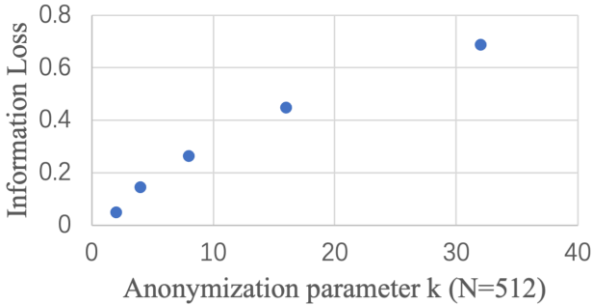


Figure 4-11 Information loss with different k.

increased and increased as k increased.

4.4.4 Hierarchical Connection of Two Domains

As mentioned, the experiments were conducted assuming an environment in which anonymization processes were coordinated in the local domains as towns and global domains as cities, as described in Section 4.3. The following is a description of the assumed environment for the experiment. Assume that there were eight blocks (local domains) that did not have sufficient share distribution destinations, using homomorphic encryption to generate $k = 4$ anonymity, and there were sufficient operators in the entire city (global domain) that could perform secret sharing to satisfy $k = 16$ anonymity.

Table 4-2 Comparation of Time Costs.

Global Domain	Homomorphic Encryption	**Secret Sharing**	Secret Sharing
Local Domain	Homomorphic Encryption	**Homomorphic Encryption**	Homomorphic Encryption
Connection	✓	✓	×
Time(seconds)	1.53×10^2	$\mathbf{1.19 \times 10^2}$	3.11×10^2

As shown in Table 4-2, when using homomorphic encryption in the local domain and using secret sharing in the global domain, the total execution time for our proposed hierarchical scheme was 1.19×10^2 seconds by sending the anonymized results of local domains to the global domain as input. Conversely, the process took 1.53×10^2 seconds when using homomorphic cryptography in both domains, indicating that secret sharing accelerated the process. In addition, when the results of local domains were not used in the global domain, the process took 3.11×10^2 seconds, indicating that the connection of local domains and a global domain sped up the anonymization process more than two times.

Moreover, when homomorphic encryption was used in local domains and secret sharing was used in the global domain, the information loss per record was 0.113 if the global domain used the anonymized results of local domains, and the information loss per record without using the anonymized results of local domains was 0.107. The increase in information loss owing to the connection between the two domains was 5.6%. However, the processing time decreased by 61.7%, benefiting from this connection. Therefore, in practical use, an increase in information loss is acceptable.

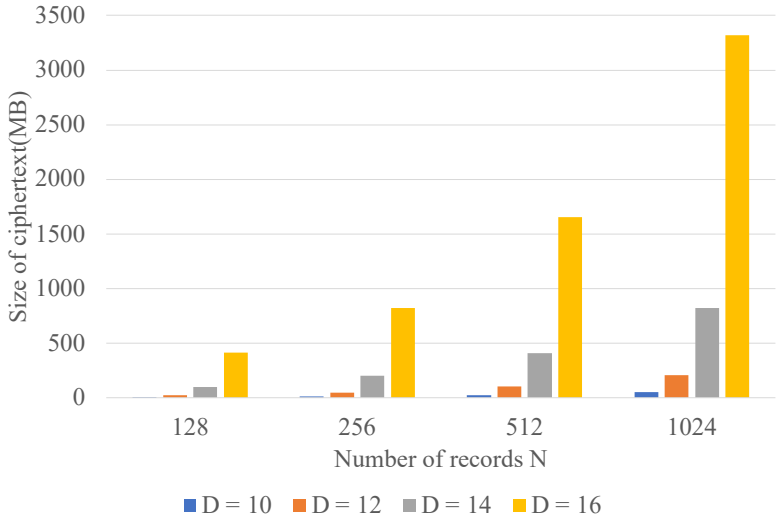


Figure 4-12 Size of ciphertext using CKKS with different polynomial degrees and numbers of records.

4.5 Limitations and Discussion

Combining homomorphic encryption and secret sharing to achieve k-anonymization over encrypted data leverages the strengths of both cryptographic techniques to enhance privacy. However, as shown in Section 4.4, the execution time increases linearly with the number of records, which may cause potential limitations when managing large records or frequent updates. In addition, the required memory usage of the two techniques is also a significant concern.

The CKKS implemented in the Microsoft SEAL [98] library is designed for performing efficient approximate arithmetic on encrypted data, making it well-suited for applications requiring complex computations on floating-point numbers. However, utilizing CKKS within SEAL introduces memory and bandwidth overheads, primarily due to the nature of homomorphic encryption. HE requires larger key size and ciphertexts for complex operations than traditional encryption methods. In addition, CKKS performs arithmetic on approximate numbers rather than exact integers and involves scaling of ciphertexts to control the precision and manage the growth of noise. These specific characteristics of the CKKS scheme typically require more memory

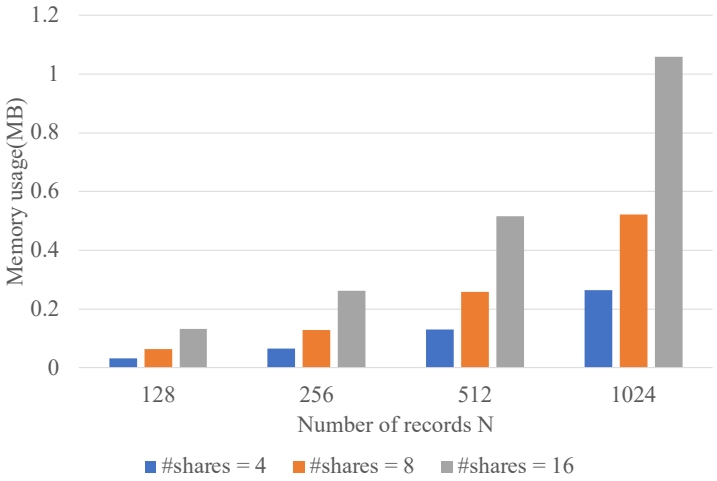


Figure 4-13 Memory usage of using Secret Sharing with different numbers of shares and records.

compared to simpler forms of homomorphic encryption. The increase in polynomial degree 2^D can lead to exponential growth of ciphertext size, as shown in Figure 4-12. Moreover, the precision of computations in the CKKS scheme largely depends on the polynomial modulus size and the noise budget. This means that increasing the polynomial degree can improve the accuracy of the results from arithmetic operations performed on encrypted data. Therefore, it's important to balance this with the increased time cost and memory usage.

Secret sharing requires less memory than homomorphic encryption. The secret is divided into multiple shares stored separately. The total memory usage grows linearly with the increase in the number of records and shares, but for every single party involved, the required memory usage is approximately as small as the original data, as shown in Figure 4-13.

Compared with only using Homomorphic Encryption, the proposed framework introduces additional memory overhead caused by Secret Sharing, but the additional memory required is so minimal that it can be considered negligible. Despite this, the proposed framework significantly improves computational speed, as discussed in Section 4.4.

The proposed framework achieves k-anonymization over encrypted data and allows for outsourcing computational tasks to cloud environments without ever needing to decrypt it, thereby preserving the privacy of the underlying data throughout the processing phase. Some organizations, such as medical and financial institutions, place a higher priority on user privacy and can accept the associated computational overhead and time costs that come with it.

In addition, the proposed framework can also integrate with other privacy-preserving technologies for data publishing and analysis, such as Differential Privacy. DP is basically limited to numerical data; however, it guarantees that the output of a statistical query will be essentially the same whether any single individual's data is included in the input dataset. k-Anonymization

modifies the dataset before release to make records indistinguishable, while DP focuses on the analysis or query process, ensuring that outputs are insensitive to any single individual's data. The choice between them depends on the specific privacy requirements and use cases.

Dynamic k-anonymization, which requires real-time data processing capabilities, is more challenging. Dealing with frequently updated datasets can be computationally intensive. Based on the computational cost and memory use discussed in Sections 4.4 and 4.5, the proposed hierarchical framework can be more scalable for large datasets and offers promising advantages for achieving dynamic k-anonymization than HE-based schemes.

4.6 Summary

Privacy-preserving outsourcing and data sharing will become increasingly important with the development of information infrastructures. However, data sharing may cause leakage of sensitive information, and existing anonymization methods that rely on untrusted third parties still expose unmodified data.

This chapter proposed a privacy-preserving hierarchical framework for k-anonymization using two secret computation methods: homomorphic encryption and secret sharing. The network was divided into global and local domains. The experimental results showed that the execution time increased linearly with the number of records and could be reduced by sending the anonymized results of a local domain to a global domain to realize hierarchical anonymization. More specifically, using secret sharing in the global domain could reduce over 20% of the processing time, and the connection between the local and global domains could reduce the processing time by 61.7%, whereas the information loss only increased by 5.6%. Hence, the framework proposed in this chapter can achieve secure anonymization of encrypted data and speed up the process, benefiting from the hierarchical structure.

The proposed method was piloted in a smart city data platform in Saitama City. Its implementation makes each individual's information unidentifiable, thus protecting the privacy of the individuals in the data set while maintaining the effectiveness of the service. For example, it can be used to protect the healthcare records of patients or the education records of students before they are shared for research or analysis. The proposed hierarchical structure can also be integrated with Differential Privacy technique to calculate the output of a statistic query, which may play an important role in various fields such as finance, retail, and public administration of citizens.

Chapter 5 Adaptive *k*-Anonymization for Unstructured Clinical Data

5.1 Motivation

The medical industry is an important field that relates to people's life and health, which generates and uses a large amount of personal information. With the development and application of information technology, clinical data is of great value for improving medical quality, promoting scientific research, and achieving precision in medicine. The professional processing and reuse of medical and health data is of great importance for monitoring health status, disease prevention, and analyzing health trends [100]. For healthcare providers, medical data sharing provides a more comprehensive view of a patient's medical history, including diagnoses and treatments, enabling better clinical decisions and personalized treatment plans. These data can reduce the risk of errors and adverse drug interactions and improve treatment outcomes. In addition, access to a broad medical dataset allows researchers to identify patterns, trends, and correlations that are impossible to distinguish from smaller, isolated data sets. This can accelerate the discovery of disease mechanisms, treatment effectiveness, and potential side effects, leading to the development of better treatments. Sharing medical data can improve public health monitoring and enable health authorities to track disease outbreaks and allocate resources more effectively.

However, this digital transformation has simultaneously introduced significant privacy challenges that demand sophisticated technical solutions to protect sensitive patient information while preserving data utility for legitimate medical purposes.

Privacy in the context of medical data sharing can be protected through various techniques, such as de-identification, cryptography, and anonymization.

De-identification is the removal or alteration of personal information from a dataset so that individuals can no longer be identified. In the context of privacy and data protection, re-identification is often used to ensure that documents or datasets comply with legal and regulatory requirements for privacy and confidentiality. While re-identification plays a critical role in removing information that can directly identify an individual, there is a risk that data may be re-identified, especially when combined with external data sources [101].

Cryptography techniques such as homomorphic encryption are promising for data protection. Homomorphic encryption enables secure data processing without exposing the actual information,

making it useful for scenarios like cloud computing, where sensitive data must be processed by third parties. However, these methods are intended to keep information confidential and cannot be used to accelerate the provision of related services by making information public. Cryptography-based methods in particular are problematic due to their high computational cost, which limits their practical use.

Anonymization, compared to re-identification and encryption, offers distinct advantages, particularly in contexts where the data sharing and analysis must balance utility and privacy concerns. Re-identification typically involves removing sensitive information from documents or datasets, which can significantly limit the usefulness of the data for further analysis or research. In contrast, anonymization attempts to preserve the usefulness of the data as much as possible while protecting privacy. Encryption protects the data by making it unreadable without the corresponding decryption key. This means that the data must be decrypted before it can be used or passed on. On the other hand, anonymization modifies the data itself to minimize the risk of disclosing personal information, even if unauthorized parties access the data. Once the data are anonymized, it can be made publicly accessible without the risk of revealing private or identifiable details. Unlike encrypted data, anonymized data can be made publicly available and used for analysis or research purposes without decryption.

Unlike structured databases where personal identifiers are confined to specific fields, clinical text can contain sensitive information dispersed across multiple words, phrases, or contextual references that are difficult to identify systematically. The linguistic complexity of medical text further compounds these challenges.

Recent advances in artificial intelligence and machine learning have created new opportunities for addressing these challenges. The integration of these models into the anonymization process allows for more accurate and context-aware detection of sensitive information, making them highly effective for protecting privacy in medical applications.

This chapter addresses these challenges by developing a comprehensive framework for clinical text anonymization that combines advanced natural language processing techniques with formal privacy guarantees. By addressing both the technical challenges of clinical text processing and the practical constraints of healthcare environments, this work contributes to the broader goal of enabling safe and ethical medical data sharing in the digital age.

5.2 System Model

This section presents the overall architecture and core design principles of the proposed anonymization framework, which comprises two main modules: an Entity Detector and an Anonymizer, as shown in Figure 5-1.

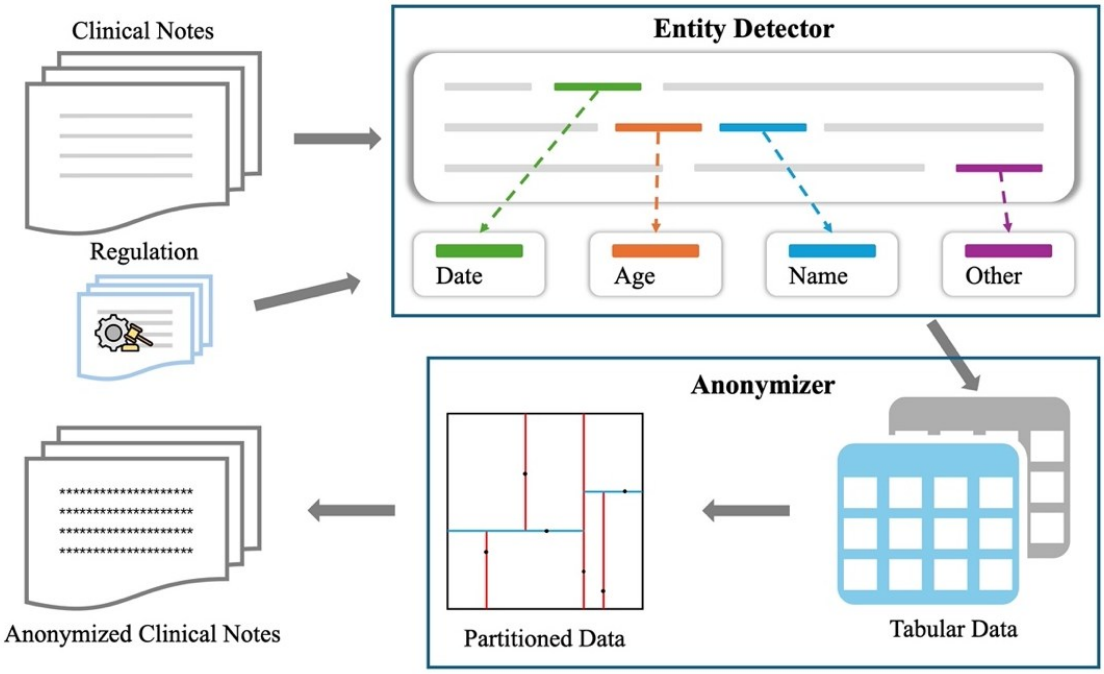


Figure 5-1 Overview of the proposed framework
There are two main modules in the framework: an Entity Detector and an Anonymizer. The Entity Detector extracts entities from clinical notes, and the Anonymizer partitions the detected entities and generalize them to meet k-anonymity.

The design is presented in the following sections: (1) The Entity Detector in Section 5.2.1. Different types of language models have been developed to extract entities from narrative medical records. (2) The Anonymizer is described in Section 5.2.2. After entity detection, the extracted identifiers are stored in a structured manner so that the Anonymizer can partition the records into q^* blocks that satisfy k-anonymity. Subsequently, different generalization rules were used to output the anonymized data.

5.2.1 Entity Detector

5.2.1.1 Fine-tuning based Model for Entity Detection

Figure 5-2 illustrates the architecture of a BERT model. The BERT model was fine-tuned using specific datasets and predefined categories. After fine tuning, it can be applied to detect the desired entities.

There are several layers to extract the named entities from the narrative text.

Tokenizer Layer:

The tokenizer splits the input text “Radiograph dated 01/02/1993” into tokens that BERT can understand. As shown in Fig. 2, the text was split into sub word tokens, and [CLS] and

[SEP] were added at the beginning and end, respectively. Tokenized text with special tokens are [“[CLS]”, “radiograph”, “dated”, “01”, “/”, “02”, “/”, “1993”, “[SEP]”].

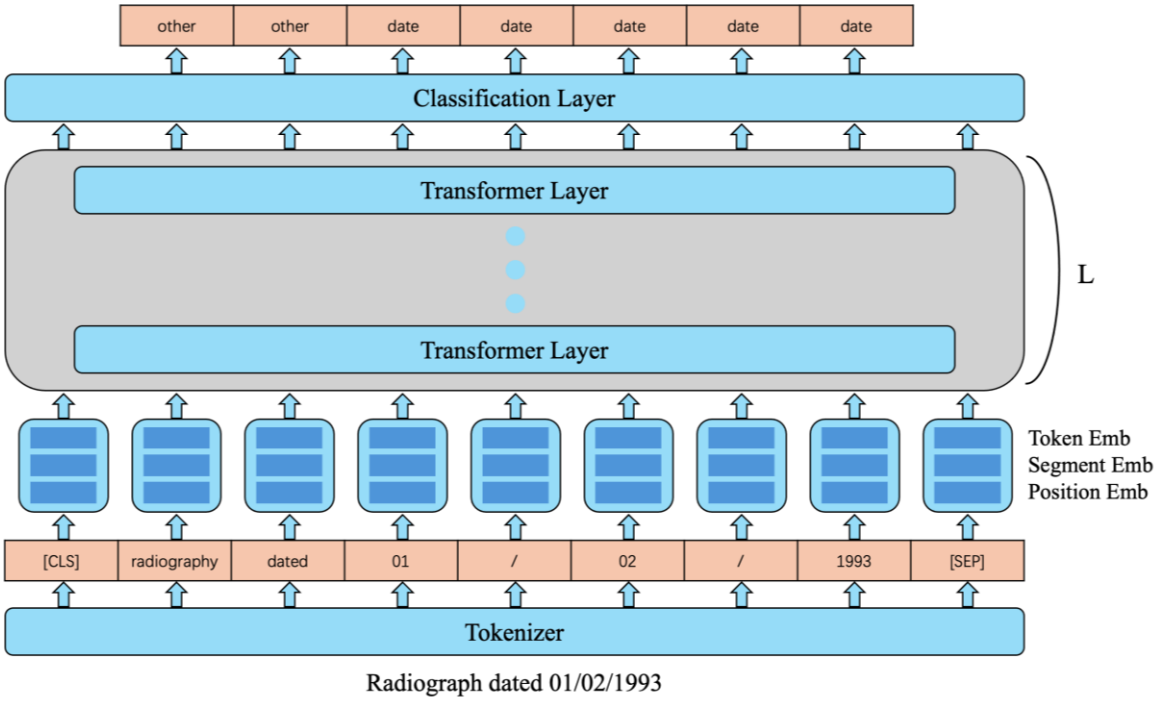


Figure 5-2 An Example of BERT Model Detecting Entities.

Contextual Embedding Layer:

The contextual embedding layer of BERT generates context-aware embeddings for each token in a given input sequence, including Token Embedding, Segment Embedding and Position Embedding.

Transformer Layers:

The model consists of L Transformer layers (only two are depicted for simplicity) [102]. Each layer consists of several self-attention heads that process the input tokens in parallel. These are the core features of BERT and allow the model to weight the importance of different tokens within a sentence to better understand the context and relationships between words.

Classification Layer:

The output of the final Transformer layer is passed to a classification layer that predicts the tag for each token. The classification layer assigns probabilities to each category to which the token belongs. In this example, the model predicts the categories for each token. For example, the tokens corresponding to the date “01/02/1993” are classified with high confidence as “date.”

Output Layer:

Figure 5-2 shows the predicted categories for each token. The model has correctly identified "Radiograph" and "dated" as "other" (meaning they are not entities of interest in this context) and the components of the date "01/02/1993" as "date."

5.2.1.2 Prompt based Model for Entity Detection

Advancements in LLMs, such as GPTs, Llama, and Claude, have revolutionized text processing. These models can easily be applied to the detection of entities using prompts. The prompts designed to detect entities that comply with specific privacy regulations will be introduced in Section 5.3.2.

```
Function k-MemberPartitioning (D, k)
Input: Dataset D, anonymization parameter k
Output: Partitioned dataset D'
# Step 1: Choose the splitting attribute
ranges = CalculateRanges(D)
split_attri = FindLargestAttributeRange(ranges)
# Step 2: Sort the dataset based on the chosen attribute
sorted_dataset = SortByAttribute(D, split_attri)
# Step 3: Split the dataset at the median
median_index = size(sorted_dataset) // 2
partition1 = sorted_dataset[:median_index]
partition2 = sorted_dataset[median_index:]
# Step 4: Recursively apply the partitioning to each partition
result1 = k-MemberPartitioning(partition1, k)
result2 = k-MemberPartitioning(partition2, k)
# Step 5: Combine the results
D'= result1 + result2
Return D'
```

Figure 5-3 Pseudo Code of k-MemberPartitioning.

5.2.2 Anonymizer

5.2.2.1 Manually Coded Rules

This section will introduce the anonymization process executed by manual code, which are partitioning and generalization.

After entity detection, the identifiers in the narrative medical records were extracted and stored in a tabular dataset D. The number of records in dataset D is denoted by N, and the anonymization parameter is denoted by k. During the partitioning process, the records are

divided into several q^* blocks and each block contains at least k records.

Figure 5-3 shows the pseudocode for partitioning a given dataset. First, the algorithm considers the dataset D and a k-anonymity value as inputs. k-Anonymity ensures that each record is indistinguishable from at least $k-1$ other records with respect to certain identifying attributes. The dataset was then recursively split based on the attributes. The attribute with the largest range is usually chosen for the split to ensure a balanced partitioning. For numeric attributes, the range was calculated as the difference between the maximum and minimum values and for categorical attributes it is counted as the number of unique values. Once the attribute with the largest range has been chosen, the values of the chosen attribute are sorted, and the dataset is split at the median value. In this step, two partitions are created. The aim is to ensure that each partition contains at least k records. The splitting process was applied recursively to each partition until all partitions contained at least k records. The algorithm has a time complexity of $O(NlogN + Np)$, where p is the number of partitions, and a space complexity of $O(N)$ for storing the sorted records. The greedy approach may not achieve the globally optimal partition and produce unbalanced partitions when element weights have high variance and disparate weights.

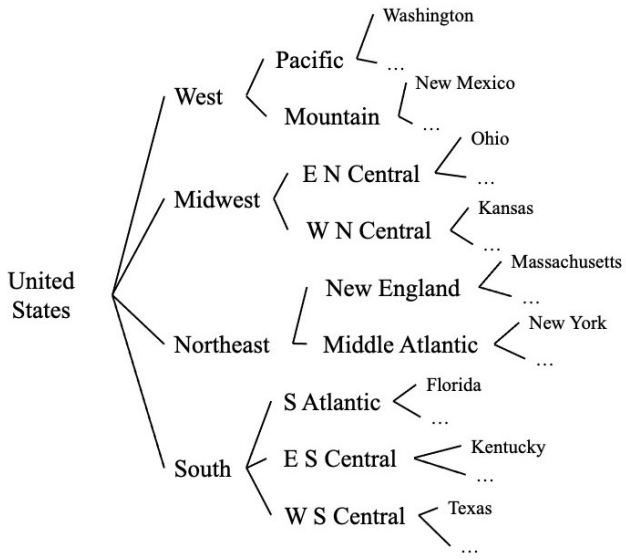


Figure 5-4 Multiway Tree.

Once the partitions are formed, the values within each partition must be generalized to meet the requirement of k-anonymity. For numeric attributes, this could mean replacing the values with a range of values within a partition. For categorical attributes, this could mean that the values are replaced by a common representation or a set of categories.

The elements of an address, such as state, city, and street, have been identified as locations. However, they are not on the same hierarchical level and may have containment relationships.

Simply using a set of values to generalize the results does not provide accurate results. Therefore, a multiway tree is constructed to represent the more precise containment relationships within a "LOCATION" shown in Figure 5-4.

According to the multiway tree structure, a common parent node can be selected to generalize categorical attributes instead of their set. Figure 5-5 shows the algorithm to find a common parent for multiple nodes.

```
Function FindCommonParent (root, nodes)
Input: Multiway tree root, attribute values nodes
Output: Common parent
if root is null or nodes is empty:
    return null
if containsAllChildren(root, nodes):
    return root
commonParents = []
for child in root.children:
    commonParent =
FindCommonParent(child, nodes)
    if commonParent is not null:

commonParents.append(commonParent)
if len(commonParents) == len(nodes):
    return root
elif len(commonParents) == 1:
    return commonParents[0]
else:
    return null
```

Figure 5-5 Pseudo Code of FindCommonParent.

Finally, the entities originally identified in the medical records were replaced by generalized entities. This replacement ensured that no sensitive information was exposed, while preserving the overall structure and context of the records.

5.2.2.2 Prompt Template for Anonymization

Prompt templates designed for anonymization that allows LLMs to partition and generalize a given dataset will be introduced in detail in the next section.

5.3 Scheme Implementation

This section provides the detailed implementation of the components introduced in Section 5.2. It elaborates on the practical aspects, such as dataset processing, model configurations, training procedures, and prompts for LLMs to achieve the design goals.

5.3.1 Implementation of Fine-tuning-based Model

5.3.1.1 Datasets

In this study, the Informatics for Integrating Biology and the Bedside (i2b2) [103] datasets were used in their entirety. Both datasets consist of annotated clinical texts that have been used extensively in research and development. Figure 5-6 shows an example record.

```
<?xml version="1.0" encoding="UTF-8" ?>
<deIdi2b2>
<TEXT><![CDATA[
...
CLINICAL NOTE
...
]]></TEXT>
<TAGS>
<DATE id="P0" start="16" end="26" text="xxx" TYPE="DATE" comment="" />
<NAME id="P1" start="38" end="46" text="xxx" TYPE="PATIENT" comment="" />
<DATE id="P2" start="89" end="97" text="xxx" TYPE="DATE" comment="" />
<NAME id="P3" start="1666" end="1684" text="xxx" TYPE="DOCTOR" comment="" />
<NAME id="P4" start="1691" end="1694" text="xxx" TYPE="DOCTOR" comment="" />
<NAME id="P5" start="1695" end="1701" text="xxx" TYPE="DOCTOR" comment="" />
<DATE id="P6" start="1708" end="1716" text="xxx" TYPE="DATE" comment="" />
<DATE id="P7" start="1721" end="1729" text="xxx" TYPE="DATE" comment="" />
<DATE id="P8" start="1734" end="1742" text="xxx" TYPE="DATE" comment="" />
</TAGS>
</deIdi2b2>
```

Figure 5-6 An Example Record of i2b2 Corpus.

The 2006 i2b2 Corpus comprises approximately 889 records (669 records in the training set and 220 records in the testing set). The annotations focus on identifying and categorizing various PHI elements such as names, dates, addresses, and other identifiers that need to be obscured or removed according to the Health Insurance Portability and Accountability Act of 1996 (HIPAA) regulations [11].

The 2014 i2b2 Corpus contains approximately 1,304 patient records (790 records in the training set and 514 records in the testing set), each of which contains multiple documents reflecting various aspects of the patient's medical history.

5.3.1.2 Compliance with Privacy Regulations

HIPAA is a major regulation in the United States that provides data privacy protection for medical information. It provides guidelines for healthcare providers, health plans, and other entities dealing with PHI to ensure privacy and security.

All forms of PHI in specific datasets must be correctly identified and handled in accordance with privacy laws, such as HIPAA regulations. There are 18 identifiers that HIPAA considers PHI, including name, address, date, and other entities.

The semantic similarities between each HIPAA identifier and category were calculated to

match the HIPAA identifiers with the PHI categories in specific datasets. If the similarity score exceeded a predetermined threshold, the identifier was mapped to the category with the highest similarity. If the highest similarity score does not meet the threshold, the identifier is categorized under an "Others" group. This strategy ensures the precise mapping of HIPAA identifiers to the appropriate categories and can be applied to different privacy laws.

To align with HIPAA regulations, the HIPAA identifiers are matched to the specific PHI categories of the i2b2 datasets: age, contact, date, location, ID, name, and profession. Unannotated entities have been labeled as "other," resulting in C = 8 possible classes for each token.

5.3.1.3 Fine-tuning

When initializing the framework, pretrained BERT models were used as the backbone. The BERT models were fine-tuned using Adam optimizer with learning rates of 3e-5. All the weights in the model were fine-tuned with a batch size of 32 for five epochs using the Google Colab GPU.

5.3.2 Prompt Engineering for LLMs

Compared to fine-tuning-based language models, prompt-driven LLMs can easily understand data and adapt to different tasks. LLMs can process narrative medical records and generate anonymous data. This section presents several prompts that LLMs can use to achieve anonymization and improve their performance.

5.3.2.1 Anonymization using Zero-shot Prompts

Zero-shot learning refers to the ability of a model to perform tasks without explicit training on specific examples related to the tasks. Zero-shot prompts can be designed to instruct LLMs to identify and effectively redact sensitive information, even without direct examples of what constitutes sensitive data in the training set.

Figure 5-7(a) shows an implicit prompt $IP.1$ to anonymize clinical records. However, this prompt cannot be easily customized for specific datasets or use cases, limiting the applicability of LLMs in diverse clinical settings with unique anonymization requirements. To increase the effectiveness of zero-shot prompts in anonymizing narrative clinical records, specific rules can be applied to identify sensitive data, providing an additional layer of accuracy. Figure 5-7(b) gives an example of a specific rule $R.1$ and Figure 5-7(c) shows an explicit prompt $EP.1$. Figure 5-7(d) gives an example of a specific rule $R.2$ and Figure 5-7(e) shows an explicit prompt $EP.2$.

Ensuring the usability of data during anonymization is critical to preserving the value of narrative clinical records for research, analysis, and clinical purposes. Instead of complete redaction, data utility can be improved by identifying and categorizing entities in clinical

narratives. In addition, some guidelines, such as the GDPR and HIPAA, have specific privacy requirements that can be met with a prompt design to obtain higher customization. Figure 5-7(f) shows an explicit anonymization prompt that can be complied with given regulations.

```
prompt = f"""
Please anonymize the following clinical note and replace all the following information with the term "[redacted]".

Clinical note: <{text}>
"""
```

(a) Implicit Prompt $IP.1$

```
rules = f"""
Redact any strings that might be a name or acronym or initials, patients' names, doctors' names, the names of the M.D. or Dr., medical staff names, redact any strings that might be a location or address, redact any strings that look like "something years old" or "age 37", redact any dates and IDs and record dates, redact clinic and hospital names, redact professions, redact any contact information.
"""
```

(b) Specific Rule $R.1$

```
prompt = f"""
Please anonymize the following clinical note and replace all the following information with the term "[redacted]" based on the specific rules.

Rules: <{rules}>
Clinical note: <{text}>
"""
```

(c) Explicit Prompt $EP.1$ with Specific Rule $R.1$

```
rules = f"""
1. Redact any strings that might be a name or acronym or initials, patients' names, doctors' names, the names of the M.D. or Dr.: NAME
2. Redact any strings that look like "something years old" or "age 37": AGE
3. Redact any strings that might be a date, such as "01/01/2024" or "January": DATE
4. Redact any contact information: CONTACT
5. Redact any IDs and numbers: ID
6. Redact any strings that might be a location or address, such as "3970 Longview Drive": LOCATION
7. Redact professions such as "manager": PROFESSION
8. Redact any other unique identifiers: OTHER
"""
```

(d) Specific Rule $R.2$

```
prompt = f"""
Please anonymize the following clinical note and replace all the following information with the category of named entities based on the specific rules.

Rules: <{rules}>
Clinical note: <{text}>
"""
```

(e) Explicit Prompt $EP.2$ with Specific Rule $R.2$

```
prompt = f"""
Please anonymize the following clinical note and replace all the following information with the category of named entities based on the provided regulations.

Regulation: <{regulation}>
Clinical note: <{text}>
"""
```

(f) Explicit Prompt $EP.3$ Complied with Given Regulations

Figure 5-7 Examples of Zero-shot Prompts that Comply with Specific Rules.

5.3.2.2 Anonymization using Few-shot Prompts

Large language models have shown remarkable zero-shot capabilities but are inadequate for more complex tasks when using the zero-shot setting. Few-shot prompting uses a small number of example inputs to demonstrate how a task should be performed. Few-shot prompting can be

used to enable contextual learning that drives the models to achieve higher performance by providing in-prompt demonstrations.

Figure 5-8(a) and Figure 5-8(b) show an example of a prompt to anonymize clinical records by adding an in-prompt demonstration to enhance the performance of the LLMs.

```
prompt = f"""
Please anonymize the following clinical note and replace all the following information with the category of named entities based on the provided regulations.

Regulation: <{regulation}>
Clinical note: <{text}>

Examples of clinical note and extracted named entities are given for reference:
Example clinical note: <{example_text}>
Example result: <{example_entity}>
"""
```

(a) A Few-shot Prompt

```
example_text = f"""
Record date: 2069-04-07
.....
#1:  Inflammatory arthritis - possibly RA - with response noted to Hydroxychloroquine along with Prednisone.  He has stopped the Prednisone, and I would not restart it yet.
#2:  New onset of symptoms suspicious for right-sided carotid disease.  Will arrange for carotid ultrasound studies.  Patient advised to call me if he develops any worsening symptoms.  He has been taking 1 aspirin per day prophylaxis long-term, and I stressed that he continue to do so.  He will follow-up with me shortly after
the ultrasound study.

Xzavian G. Tavares, M.D.
XGT:holmes

DD: 04/07/69
DT: 04/15/69
DV: 04/07/69
"""

example_entity = f"""
{
  "Name": ["Villegas", "Xzavian G. Tavares", "XGT", "holmes"],
  "Date": ["2069-04-07", "November", "04/07/69", "04/15/69", "04/07/69"]
}
"""
```

(b) An In-prompt Demonstration

Figure 5-8 An Example of A Few-shot Prompt Enhanced by An In-prompt Demonstration.

5.3.2.3 Anonymization using Chain-of-Thought Prompts

Chain-of-thought (CoT) prompts are a powerful technique that guides a model through a multistep reasoning process to arrive at a solution. By breaking down the task into sequential steps and providing clear examples, this method improved the accuracy and consistency of the model to identify and anonymize sensitive information.

Figure 5-9 outlines the structured approach to anonymizing narrative clinical records using an agent-based system. The entire anonymization task can be divided into different subtasks and performed step by step:

1. Agent Initialization:

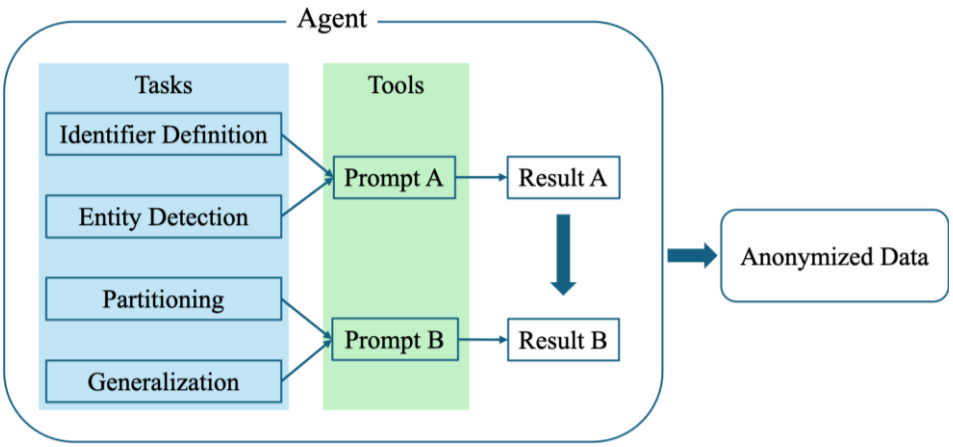


Figure 5-9 Anonymization Flow with Task-Tool Mapping.

An agent was responsible for planning the anonymization process. This agent supervises the execution of various tasks and the application of prompts to achieve anonymization.

2. Task Execution:

The agent performs predefined tasks to identify and process sensitive information within clinical records, including Identifier Definition, Entity Detection, Partitioning, and Generalization.

3. Tool Application:

The tasks feed into different tools that use specific prompts to process the data. The tools and prompts work together to execute the anonymization.

Prompt A: This prompt is used to process the outputs from the task Identifier Definition and Entity Detection. The tool uses Prompt A to identify and anonymize specified entities.

Prompt B: This prompt is used to process the Partitioning and Generalization tasks. This tool uses Prompt B to ensure that the text is appropriately segmented and generalized.

4. Final Anonymized Data:

The combined anonymized output is then ready for further use, such as research, analysis, or sharing with stakeholders, while ensuring privacy compliance and data integrity.

Prompt A, designed for the Entity Detector, and prompt B, for the Anonymizer, are shown in Figure 5-10. Compared to zero-shot and few-shot prompts, more specific generalization rules are executed to achieve higher data utility. The evaluation of this method compared to manually

coded anonymization rules is shown in Figure 5-11 in Section 5.4.

```
prompt = f"""
Based on the provided regulations, identify and extract all named entities in the given clinical notes. Return the results as
a JSON object with the following structure:
{
  "Name": [],
  "Age": [],
  "Date": [],
  "Contact": [],
  "Location": [],
  "ID": [],
  "Profession": [],
  "Other": []
}
Regulation: <{regulation}>
Clinical note: <{text}>
"""
```

(a) A Prompt for Entity Detection

```
regulation = f"""
1. Names
2. All geographical subdivisions smaller than a State (including street address, city, county, precinct, zip code, and their equivalent geocodes,
with specific rules for zip codes)
3. All elements of dates (except year) for dates directly related to an individual (including birth date, admission date, discharge date, date of
death) and all ages over 89
.....
18. Any other unique identifying number, characteristic, or code (excluding codes assigned by the investigator to code the data)
"""
```

(b) HIPAA Regulation

```
text = f"""
Record date: 2069-04-07
.....
#1:  Inflammatory arthritis - possibly RA - with response noted to Hydroxychloroquine along with Prednisone.  He has stopped the Prednisone, and I
would not restart it yet.
#2:  New onset of symptoms suspicious for right-sided carotid disease.  Will arrange for carotid ultrasound studies.  Patient advised to call me if he
develops any worsening symptoms.  He has been taking 1 aspirin per day prophylaxis long-term, and I stressed that he continue to do so.  He will follow-
up with me shortly after
the ultrasound study.

Xzavian G. Tavares, M.D.
XGT:holmes

DD: 04/07/69
DT: 04/15/69
DV: 04/07/69
"""
```

(c) An Original Clinical Record

```
prompt = f"""
Given a structured tabular dataset containing attributes of patients, achieve k-anonymization and return the anonymized
data step by step.

Step 1: identify which columns contain quasi-identifiers;
Step 2: set the value of k to 4;
Step 3: generalize and suppress certain values in the quasi-identifier columns;
Step 4: use ranges for numerical quasi-identifiers and use broader categories for categorical quasi-identifiers;
Step 5: apply the generalization rules to the dataset and show the modified dataset;
Step 6: verify that the dataset now meets the k-anonymity requirement.

Quasi-identifiers: <{quasi_identifiers}>
Structured records: <{structured_records}>
"""
```

(d) A Prompt for Anonymization

Figure 5-10 Example of prompt compliance with HIPAA regulations.

5.4 Experimental Evaluation

As presented in Section 5.3, the proposed framework consists of two main components: an Entity Detector and Anonymizer. Therefore, this section first evaluates the performance of

different models, such as the Entity Detector, and then compares the anonymization performance.

5.4.1 Performance of Entity Detection

5.4.1.1 Evaluation Criteria

Evaluating the performance of a model as an Entity Detector involves measuring how well it can identify and classify named entities (such as the names of people, organizations, and locations) in a given text. Common metrics and methods for evaluating NER performance include the following.

Precision: The ratio between the correctly identified named entities and the total number of entities identified by the model. This measures the accuracy of the model's predictions.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{5-1}$$

Recall: Ratio between the correctly identified named entities and the total number of entities present in the ground truth. It measures the ability of a model to identify all relevant entities.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{5-2}$$

F1-Score: The harmonic mean of the precision and recall provides a single metric that balances both aspects.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5-3}$$

5.4.1.2 Models

As presented in Section 5.3, two types of language models can be applied to the proposed framework: locally fine-tuned and prompt-driven models.

For the former, the pre-trained weights were downloaded from the Hugging Face and these models were initialized locally. BERTbase [104], BERTlarge [104], RoBERTa [105], and ClinicalBERT [106] were developed using the training dataset of the i2b2 corpus with the same hyperparameters and they were evaluated on the test set.

BERTbase is a smaller version of the BERT model and has 12 transformer layers, whereas BERTlarge has 24 transformer layers, doubling the depth of BERTbase.

ClinicalBERT was pre-trained using clinical notes extracted from MIMIC-III. RoBERTa is an improved variant of the BERT model developed by Facebook AI, which is trained with larger batch sizes and higher learning rates and removes the next sentence prediction.

For the prompt-driven methods, various prompts were sent to the GPT-4o [107], Llama 3.1 [108], and Claude 3.5 [109] models to evaluate their performance. We used the publicly available GPT-4o model released by OpenAI in May 2024, the 8B parameter version of LLaMA 3.1, released by Meta in April 2024, and the Claude 3.5 Sonnet, released by Anthropic in June 2024. The GPT-4o and Claude 3.5 models can only be accessed via online APIs, whereas the llama 3.1 8B model can be applied locally.

5.4.1.3 Entity Detection Performance

Table 5-1 lists a comprehensive overview of the model evaluation results for the i2b2 dataset, including the Precision, Recall, and F1-score. The first four models were fine-tuned locally with the same hyperparameters. Of the four BERT-based models, the fine-tuned RoBERTa achieved the highest performance, whereas all four fine-tuned models achieved F1-scores above 90%.

Table 5-1 Performance Comparison of Entity Detection.

Dataset	model	P	R	F1
i2b2 2006	BERTbase	0.897	0.905	0.901
	BERTlarge	0.924	0.934	0.929
	RoBERTa	**0.944**	**0.948**	**0.946**
	ClinicalBERT	0.942	0.940	0.941
	GPT-4o	**0.965**	**0.981**	**0.973**
	Claude 3.5	0.958	0.966	0.962
	Llama 3.1	0.955	0.961	0.958
i2b2 2014	BERTbase	0.891	0.895	0.893
	BERTlarge	0.909	0.907	0.908
	RoBERTa	**0.927**	**0.927**	**0.927**
	ClinicalBERT	0.923	0.919	0.921
	GPT-4o	0.948	**0.991**	0.969
	Claude 3.5	**0.954**	0.989	**0.971**
	Llama 3.1	0.947	0.957	0.95

In contrast, prompt-driven large language models were evaluated using the prompts presented in Section 5.3.2, which comply with specific privacy regulations. It can be observed that these prompt-driven models all outperformed the fine-tuned BERT-based model, indicating the powerful capability of LLMs. However, it should be noted that the performance of these models is highly dependent on the quality and structure of the prompts provided. The prompts should provide the context that the model needs to generate responses. If a prompt is unclear, the model may give irrelevant or incorrect responses. LLMs may detect more entities than are actually present in a text if they are not properly regulated. Such an over detection can result

in lower precision in entity-detection tasks.

In addition, Table 5-2 shows the evaluation results, including the Precision, Recall, and F1-score, of each entity type for i2b2 2014 corpus, using the RoBERTa model and GPT-4o model, which are the best-performing model of fine-tuned models and prompt-driven models respectively.

Table 5-2 Performance Comparison of Entity Detection for Each Entity Type.

Entity Type	Model	P	R	F1
AGE	RoBERTa	0.923	0.921	0.922
	GPT-4o	0.949	0.996	0.972
CONTACT	RoBERTa	0.922	0.925	0.923
	GPT-4o	0.946	0.993	0.969
DATE	RoBERTa	0.932	0.932	0.932
	GPT-4o	0.952	0.990	0.970
LOCATION	RoBERTa	0.916	0.919	0.917
	GPT-4o	0.937	0.990	0.963
ID	RoBERTa	0.921	0.924	0.922
	GPT-4o	0.944	0.987	0.965
NAME	RoBERTa	0.929	0.927	0.928
	GPT-4o	0.951	0.994	0.972
PROFESSION	RoBERTa	0.914	0.916	0.915
	GPT-4o	0.931	0.988	0.959

5.4.2 Performance of Anonymization

5.4.2.1 Evaluation Criteria

To evaluate the performance of anonymization techniques, Information loss is calculated as a metric. By measuring the extent to which the original data are altered or removed during the anonymization process, information loss helps determine how effectively the data retains its utility while protecting sensitive information.

The following formula is used to calculate the information loss for numerical attributes, where the number of records is denoted as N, and the number of quasi-identifier attributes is denoted as M.

$$IL_{num} = \frac{1}{N} \sum_{i=1,\dots,N} \sum_{j=1,..,M} \frac{a_{i,j,max} - a_{i,j,min}}{|N_j|} \quad (5\text{-}4)$$

where $a_{i,j,max}$ and $a_{i,j,min}$ represent the upper and lower limits of the range values of the i-th record and j-th quasi-identifier, respectively. $|N_j|$ denotes the domain size of the j-th quasi-identifier.

As presented in Section 5.2.2, when generalizing categorical data are within a q^* block, the values are replaced by their least common ancestors. The information loss of the categorical attributes was calculated based on the extent to which the original data were generalized. Specifically, it is measured by the height of the node to which the data were generalized in the hierarchical tree and normalized by the height of the entire tree:

$$IL_{cat} = \frac{H\big(A(c)\big)}{H(root)} \tag{5-5}$$

where $H(A(c))$ is the height of the least common ancestor, and $H(root)$ is the total height of the tree. This ratio indicates the extent to which the data have been generalized, with a higher value indicating a greater loss of information.

5.4.2.2 Models

After entity detection, the same k-anonymization process was applied across all locally fine-tuned models using specific generalization rules. The main difference between these models lies in their entity detection capabilities, whereas the subsequent anonymization step remains consistent. The anonymization process was performed with manually coded generalization rules. The fine-tuned model with best performance in Entity Detection process was chosen and the combination of this model and our proposed generalization rules was named as "Anon_manual".

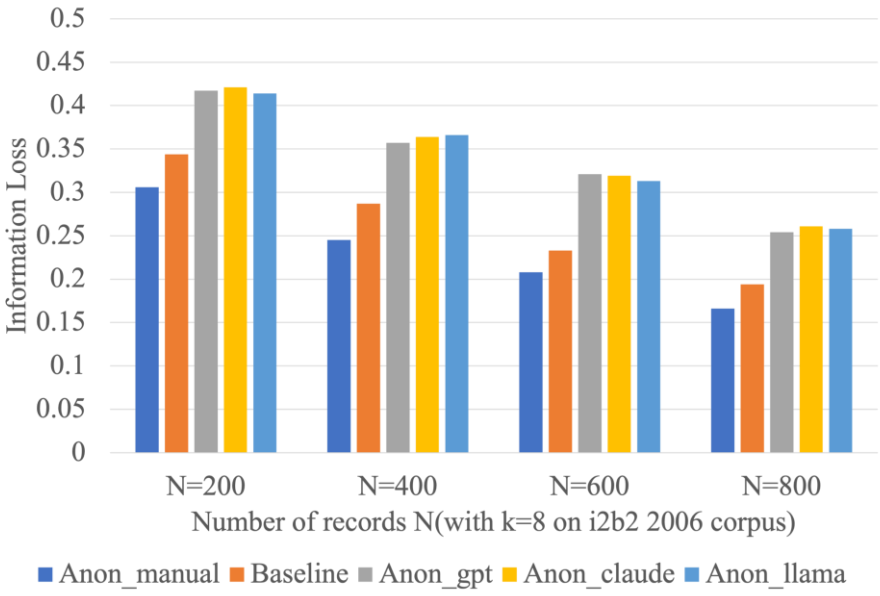
However, for the prompt-driven LLMs, the prompts followed the same generalization rules designed to guide them in performing the anonymization process as the anonymizers, as presented in Section 5.3. These models were named as "Anon_gpt", "Anon_claude", "Anon_llama", respectively.

In addition, the above models were compared with a baseline model [110]. The "Baseline" anonymizer used the same entity detection results with "Anon_manual". The information loss of different methods performing k-anonymization was compared over the same structured dataset obtained in the entity detection step.

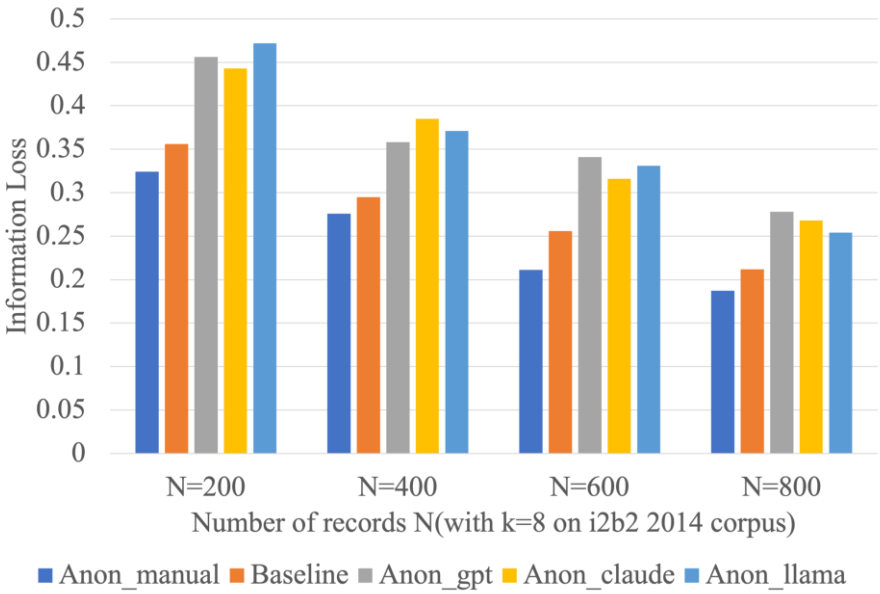
5.4.2.3 Anonymization Performance

Figure 5-11 shows the information loss with a different number of records N when $k = 8$ for the 2006 i2b2 and 2014 i2b2 corpora. Figure 5-12 shows the information loss with a different k

when $N = 800$ for the 2006 i2b2 and 2014 i2b2 corpora.



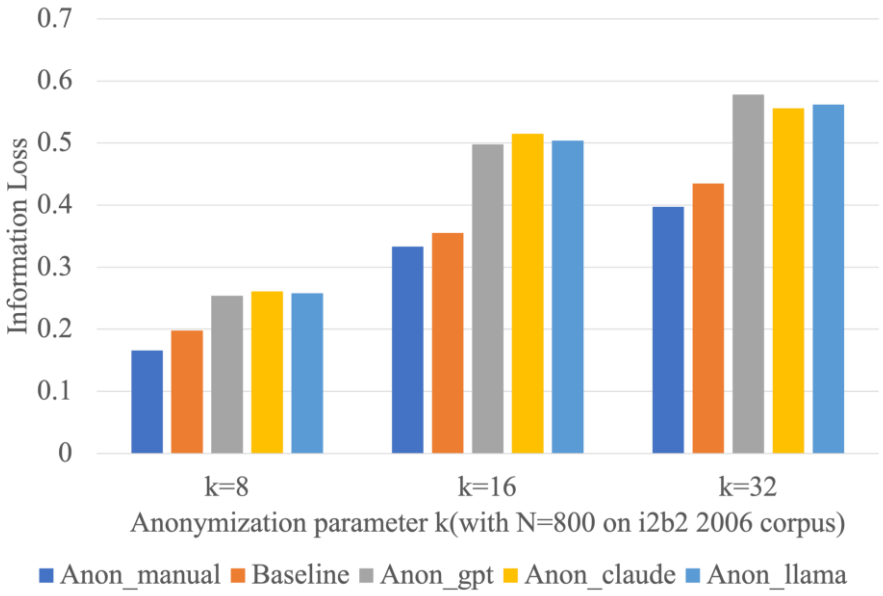
(a) Information Loss with Different Number N on i2b2 2006 Corpus



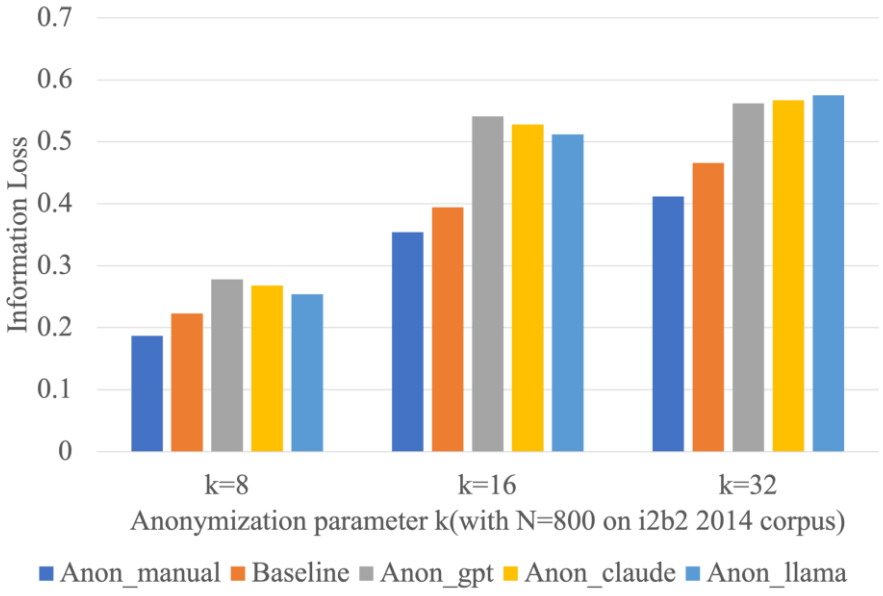
(b) Information Loss with Different Number N on i2b2 2014 Corpus

Figure 5-11 Information Loss with Different Number of Records N.

It is evident that the manually coded anonymization process outperformed the prompt-driven LLMs. Prompt-driven LLMs often face the challenge of achieving the most precise generalization necessary to satisfy k-anonymity. Unlike manually coded anonymization, which enables carefully controlled and targeted generalization according to predefined rules, LLMs may overgeneralize data because they are not inherently designed to optimize the minimal scope



(c) Information Loss with Different Parameter k on i2b2 2006 Corpus



(d) Information Loss with Different Parameter k on i2b2 2014 Corpus

Figure 5-12 Information Loss with Different Privacy Parameter k.

required to preserve privacy. An example of the intermediate prompt reasoning step for the anonymization step is shown in Figure 5-13, where the generalized age range is set to a fixed value of 10 years. This lack of precision in the generalization can result in a higher degree of information loss, as the model may apply broader generalizations than necessary, thereby reducing the usefulness of the anonymized data.

```
Step 5: Applying generalization rules to the dataset:
-  Generalizing 'age' column:
-  - Original values: [68, 60, 78, 64, 54, ...]
-  - Generalized values: [60-69, 60-69, 70-79, 60-69, 50-59, ...]
-  Generalizing 'date' column:
-  - Original values: ['20930428', '20880521', '21351215', '20810906', '20821030', ...]
-  - Generalized values: ['20900000', '20800000', '21000000', '20800000', '20800000', ...]
-  Generalizing 'location' column:
-  - Original values: ['BMC', 'Allina Medical Clinic At Hinckley', 'PERDUE MEDICAL CENTER', 'Chesterfield', 'CHH', ...]
-  - Generalized values: ['Other', 'Other', 'Other', 'Other', 'Other', ...]
```

Figure 5-13 An Example of An Intermediate Reasoning Step.

In addition, as described in Section 5.2.2, the proposed framework built a multiway tree structure to represent the hierarchical relations between attributes so that it has less information loss than the baseline method.

5.5 Comprehensive Discussion

In this study, a framework for anonymizing narrative clinical records was proposed, which includes two important steps: detecting entities within the records and then applying k-anonymization to ensure privacy. The framework was developed with two distinct implementations to evaluate the performance and trade-offs between the different approaches. The first implementation used fine-tuned BERT-based models for entity detection, followed by manually coded generalization rules to achieve k-anonymity. The second implementation allows the LLMs to detect entities and then perform k-anonymization according to predefined generalization rules.

Entity detection is a critical component in the framework, as accurate identification of sensitive information such as names, dates, and locations is essential for effective anonymization. In the first implementation, fine-tuned BERT-based models were used to detect these entities, and they all achieved an F1-score of over 90%. In comparison, the second implementation used LLMs to detect entities, and the best-performing model achieved an F1-score of 97.3%.

However, storage and time costs should be considered when comparing these methods. As shown in Table 5-3, LLMs require significantly more computational resources than fine-tuned BERT models. In contrast, BERT-based models are more efficient as they require less computing power. For example, using low-rank adaptation (LoRA) [25] instead of full fine-tuning for the large RoBERTa model can significantly reduce memory consumption, bringing it down to less than 18 GB on the GPU, which is acceptable for resource-constrained environments. In terms of time latency, LLMs can identify entities using few-shot prompts or even zero-shot prompts, whereas BERT-based models require time-consuming fine-tuning. However, once fine-tuned,

BERT-based models typically achieve inference speeds that are 10 to 50 times faster than those of LLMs.

Table 5-3 Comparison of Language Models.

Model	Scale	Time latency		Open source	Deployment method	Memory
		Fine-tuning	Entity detection per record			
BERTbase	110M	37min	Less than 200ms	Yes	Fine-tuning (LoRA)	6.76GB
BERTlarge	340M	116min	Less than 500ms	Yes	Fine-tuning (LoRA)	16.80GB
RoBERTa	355M	155min	Less than 500ms	Yes	Fine-tuning (LoRA)	17.65GB
Clinical BERT	110M	34min	Less than 200ms	Yes	Fine-tuning (LoRA)	6.98GB
GPT-4o	N/A	N/A	Over 5s	No	API(cloud)	N/A
Claude3.5	N/A	N/A	Over 5s	No	API(cloud)	N/A
Llama3.1	8B	N/A	Over 10s	Yes	API(local)	6.55GB

The second critical step in the framework is k-anonymization, which is applied to the structured tabular data generated from the detected entities. The goal of k-anonymization is to ensure that any given record is indistinguishable from at least $k-1$ other records, thus protecting the privacy of the individuals in the dataset. In the first implementation, manually coded generalization rules were used to achieve k-anonymity. As shown in Figure 5-11 and Figure 5-12, the manually coded anonymization process clearly outperformed the prompt-driven LLMs and resulted in less information loss. Conversely, the prompt-driven LLMs, which tend to apply broader generalizations, struggled to achieve the precision of manually coded anonymizations, resulting in greater information loss.

The results in this section highlight the strengths and limitations of both fine-tuned BERT models and LLMs for anonymizing narrative clinical records. Although LLMs improve the accuracy of entity detection, fine-tuned BERT models remain highly competitive, especially when considering domain-specific optimization. In addition, some LLMs are not open-source and can only be accessed via external APIs or cloud services. The use of such models may conflict with privacy regulations or raise additional concerns. In contrast, fine-tuned BERT models can be deployed locally, providing greater control over the data and reduce the dependance on external services.

In general, the proposed framework allows for the use of different language models. Therefore,

the choice of implementation can be flexibly adapted to the specific requirements of each scenario, allowing users to select the most suitable model based on factors such as computational resources, data complexity, and the desired balance between accuracy and privacy.

5.6 Summary

In this chapter, a flexible and adaptive framework for anonymizing narrative clinical records was proposed, allowing for the integration of various language models according to the specific requirements of the application scenario. Two primary implementations were evaluated: one that uses locally fine-tuned BERT models for entity detection and manually coded generalization rules for k-anonymization and the other that uses prompt-driven LLMs to accomplish both tasks.

The results of this study showed that while LLMs obtained higher performance in entity detection, reaching an F1-score of over 95% compared to the 90% achieved by fine-tuned BERT models, manually coded anonymization rules significantly outperformed the LLM-driven approaches in the anonymization step.

Furthermore, the analysis indicated the importance of considering computational resources and accessibility when selecting a suitable implementation. While LLMs are powerful, they require substantial computational resources and are often dependent on external services that can pose additional challenges in terms of data privacy and network reliability. In contrast, fine-tuned BERT models with local deployment options offer a practical and effective solution for resource-constrained healthcare systems.

In summary, the proposed framework provides a robust and adaptable approach for anonymizing narrative clinical records, ensuring a balance between accuracy, privacy, and computational efficiency. Fine-tuned BERT models with manually coded generalization rules can anonymize data with limited resources and achieve satisfactory accuracy. Future studies could investigate the integration of additional models and techniques to further improve the framework's capabilities.

Chapter 6 Conclusion

This thesis has presented three novel anonymization frameworks that address critical challenges in privacy-preserving data analysis for both structured and unstructured data. The research contributes to the advancement of practical privacy protection mechanisms that can be deployed in real-world environments while maintaining the balance between privacy protection and data utility. The proposed approaches collectively demonstrate that effective anonymization can be achieved without requiring trust in third-party entities, while adapting to diverse computational environments and regulatory requirements.

6.1 Summary of Contributions

The primary contribution of this research lies in the development of practical anonymization frameworks that bridge the gap between theoretical privacy guarantees and real-world deployment requirements. Each proposed approach addresses specific limitations in existing privacy-preserving technologies while maintaining focus on techniques that can be effectively implemented across diverse application domains.

The secure differential privacy framework represents a significant advancement in eliminating trust requirements for centralized data processing. By enabling statistical computations to be performed entirely within encrypted domains using homomorphic encryption, the framework ensures that sensitive data remains protected throughout the analytical pipeline. The experimental evaluation demonstrates that this approach achieves superior data utility compared to local differential privacy methods while maintaining equivalent privacy guarantees. The framework adds significantly less noise than existing local approaches under identical privacy budgets, resulting in substantially improved accuracy for statistical analyses. This contribution addresses the fundamental limitation of existing centralized differential privacy systems that require complete trust in data curators, making privacy-preserving analytics accessible to organizations with strict security requirements.

The hierarchical k-anonymization framework provides an innovative solution for optimizing privacy-preserving computations across heterogeneous computing environments. By strategically combining homomorphic encryption in resource-constrained local domains with secret sharing in distributed global domains, the framework achieves substantial performance improvements while maintaining rigorous privacy guarantees. The experimental evaluation reveals execution time reductions exceeding 60% compared to single-method approaches, with minimal impact on information loss metrics. This hierarchical approach recognizes the reality of

modern computing infrastructures, particularly in smart city environments, where computational capabilities vary significantly across network tiers. The framework's ability to adapt secure computation methods based on available resources represents a practical advancement that enables efficient deployment of privacy-preserving technologies in real-world scenarios.

The adaptive two-stage anonymization framework for clinical text data addresses the growing need for privacy protection in unstructured healthcare data. By integrating advanced natural language processing capabilities with formal k-anonymization techniques, the framework provides a comprehensive solution for protecting sensitive information in narrative medical records. The research demonstrates that both fine-tuned BERT models and large language models can achieve high accuracy in entity detection, with F1-scores exceeding 90% across multiple clinical datasets. The framework's modular design enables dynamic adaptation to various privacy regulations and deployment constraints, making it suitable for diverse healthcare environments. The comparative analysis between different language model approaches provides practical guidance for organizations seeking to balance accuracy requirements with computational and deployment constraints.

Beyond these primary contributions, the research introduces several methodological innovations that enhance the practical applicability of privacy-preserving technologies. The development of multiway tree structures for categorical attribute generalization reduces information loss compared to traditional set-based approaches. The creation of comprehensive evaluation methodologies that assess both privacy protection and data utility across diverse application scenarios provides valuable tools for future research and practical deployment decisions.

6.2 Limitations and Future Research Directions

While this research demonstrates significant advances in anonymization, several limitations suggest directions for future investigation. The computational overhead associated with secure computation techniques, while reduced through the proposed optimizations, remains higher than plaintext operations. Future research could explore additional optimization strategies, including hardware acceleration and algorithmic improvements, to further reduce computational costs.

The scalability of the proposed frameworks to extremely large datasets requires additional investigation. While the hierarchical approach demonstrates improved efficiency compared to single-method alternatives, the fundamental scalability limitations of cryptographic operations may become prohibitive for certain applications. Research into more efficient cryptographic primitives and distributed computing strategies could address these limitations.

The evaluation of privacy protection effectiveness relies primarily on theoretical guarantees

and simulated attack scenarios. Future research could benefit from more comprehensive empirical evaluation using real-world attack datasets and adversarial testing to validate the practical security of the proposed frameworks under diverse threat models.

The integration of the proposed frameworks with emerging technologies such as federated learning, blockchain systems, and quantum computing presents opportunities for future research. These integrations could extend the applicability of formal and practical anonymization to new domains while addressing evolving security and privacy challenges.

References

[1] S. E. Bibri and J. Krogstie, "Smart sustainable cities of the future: An extensive interdisciplinary literature review," *Sustain. Cities Soc.*, vol. 31, pp. 183-212, 2017.

[2] T. Yigitcanlar, M. Kamruzzaman, L. Buys, G. Ioppolo, J. Sabatini-Marques, E. M. da Costa, and J. J. Yun, "Understanding 'smart cities': Intertwining development drivers with desired outcomes in a multidimensional framework," *Cities*, vol. 81, pp. 145-160, 2018.

[3] M. Batty *et al.*, "Smart cities of the future," *Eur. Phys. J. Spec. Top.*, vol. 214, pp. 481-518, 2012.

[4] R. Kitchin, "The real-time city? Big data and smart urbanism," *GeoJournal*, vol. 79, pp. 1-14, 2014.

[5] S. Zuboff, *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. New York: PublicAffairs, 2019.

[6] J. Adler-Milstein and A. K. Jha, "HITECH Act drove large gains in hospital electronic health record adoption," *Health Aff.*, vol. 36, no. 8, pp. 1416-1422, 2017.

[7] I. Keshta and A. Odeh, "Security and privacy of electronic health records: Concerns and challenges," *Egypt. Inform. J.*, vol. 22, no. 2, pp. 177-183, 2021.

[8] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, Berlin, Heidelberg: Springer, 2008, pp. 1-19.

[9] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. 2014 ACM SIGSAC Conf. Comput. Commun. Security*, 2014, pp. 1054-1067.

[10] European Parliament and Council, "Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data," *Off. J. Eur. Union*, vol. L119, pp. 1-88, 2016.

[11] US Department of Health and Human Services, "Health Insurance Portability and Accountability Act of 1996," Public Law 104-191, 1996.

[12] D. Boyd and K. Crawford, "Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon," *Inf. Commun. Soc.*, vol. 15, no. 5, pp. 662-679, 2012.

[13] D. J. Solove, "Introduction: Privacy self-management and the consent dilemma," *Harv. Law Rev.*, vol. 126, no. 7, pp. 1880-1903, 2013.

[14] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. 2013 IEEE 54th Annu. Symp. Found. Comput. Sci.*, 2013, pp. 429-438.

[15] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1376-1385.

[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, vol. 3876, Berlin, Heidelberg: Springer, 2006, pp. 265-284.

[17] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1351-1362, 2016.

[18] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.

[19] D. Sánchez and M. Batet, "C-sanitized: A privacy model for document redaction and sanitization," *J. Assoc. Inf. Sci. Technol.*, vol. 67, no. 1, pp. 148-163, 2016.

[20] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 169-178.

[21] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637-646, 2016.

[22] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557-570, 2002.

[23] A. Roy Chowdhury, C. Wang, X. He, A. Machanavajjhala, and S. Jha, "Cryptϵ: Crypto-assisted differential privacy on untrusted servers," in *Proc. 2020 ACM SIGMOD Int. Conf. Manage. Data*,

2020, pp. 603-619.
[24] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450-465, 2018.
[25] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022, vol. 1, no. 2, p. 3.
[26] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Found. Secure Comput.*, vol. 4, no. 11, pp. 169-180, 1978.
[27] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211-407, 2014.
[28] L. Sweeney, "Simple demographics often identify people uniquely," *Health*, vol. 671, p. 1-34, 2000.
[29] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. 2008 IEEE Symp. Security Privacy*, 2008, pp. 111-125.
[30] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM J. Comput.*, vol. 40, no. 3, pp. 793-826, 2011.
[31] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proc. 22nd ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.*, 2003, pp. 202-210.
[32] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 351-360.
[33] M. Templ, A. Kowarik, and B. Meindl, "Statistical disclosure control for micro-data using the R package sdcMicro," *J. Stat. Softw.*, vol. 67, pp. 1-36, 2015.
[34] C. Dwork, "Differential privacy," in *Int. Colloq. Automata Lang. Program.*, Berlin, Heidelberg: Springer, 2006, pp. 1-12.
[35] Y. Lindell, "Secure multiparty computation for privacy preserving data mining," in *Encyclopedia of Data Warehousing and Mining*, IGI Global, 2005, pp. 1005-1009.
[36] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612-613, 1979.
[37] S. M. Meystre, F. J. Friedlin, B. R. South, S. Shen, and M. H. Samore, "Automatic de-identification of textual documents in the electronic health record: a review of recent research," *BMC Med. Res. Methodol.*, vol. 10, pp. 1-16, 2010.
[38] I. Neamatullah *et al.*, "Automated de-identification of free-text medical records," *BMC Med. Inform. Decis. Mak.*, vol. 8, pp. 1-17, 2008.
[39] A. Stubbs, C. Kotfila, and Ö. Uzuner, "Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1," *J. Biomed. Inform.*, vol. 58, pp. S11-S19, 2015.
[40] A. E. Johnson, L. Bulgarelli, and T. J. Pollard, "Deidentification of free-text medical records using pre-trained bidirectional transformers," in *Proc. ACM Conf. Health Inference Learn.*, 2020, pp. 214-221.
[41] K. El Emam *et al.*, "A globally optimal k-anonymity method for the de-identification of health data," *J. Am. Med. Inform. Assoc.*, vol. 16, no. 5, pp. 670-682, 2009.
[42] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proc. 2009 ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 19-30.
[43] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 371-380.
[44] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proc. 2010 IEEE 51st Annu. Symp. Found. Comput. Sci.*, 2010, pp. 51-60.
[45] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci.*, 2007, pp. 94-103.
[46] C. Dwork *et al.*, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology-EUROCRYPT 2006*, vol. 4004, Berlin, Heidelberg: Springer, 2006, pp. 486-503.
[47] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proc. 2014 IEEE 55th Annu. Symp. Found. Comput. Sci.*, 2014, pp. 464-473.
[48] C. Dwork and V. Feldman, "Privacy-preserving prediction," in *Conf. Learn. Theory*, 2018, pp. 1693-1702.

References

[49] J. Hsu *et al.*, "Differential privacy: An economic method for choosing epsilon," in *Proc. 2014 IEEE 27th Comput. Security Found. Symp.*, 2014, pp. 398-410.

[50] I. Mironov, "Rényi differential privacy," in *Proc. 2017 IEEE 30th Comput. Security Found. Symp.*, 2017, pp. 263-275.

[51] J. M. Abowd, "The US Census Bureau adopts differential privacy," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 2867-2867.

[52] S. Ruggles and D. Van Riper, "The role of chance in the census bureau database reconstruction experiment," *Popul. Res. Policy Rev.*, pp. 1-8, 2022.

[53] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[54] P. Golle, "Revisiting the uniqueness of simple demographics in the US population," in *Proc. 5th ACM Workshop Privacy Electron. Soc.*, 2006, pp. 77-80.

[55] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, pp. 1010-1027, 2001.

[56] L. Willenborg and T. De Waal, *Elements of Statistical Disclosure Control*, vol. 155. Springer Science & Business Media, 2012.

[57] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," in *Proc. 23rd ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.*, 2004, pp. 223-228.

[58] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Proc. 22nd Int. Conf. Data Eng.*, 2006, pp. 25-25.

[59] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *Proc. 2005 ACM SIGMOD Int. Conf. Manage. Data*, 2005, pp. 49-60.

[60] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *Proc. 21st Int. Conf. Data Eng.*, 2005, pp. 217-228.

[61] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, p. 3-es, 2007.

[62] D. J. Martin *et al.*, "Worst-case background knowledge for privacy-preserving data publishing," in *Proc. 2007 IEEE 23rd Int. Conf. Data Eng.*, 2007, pp. 126-135.

[63] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. 2007 IEEE 23rd Int. Conf. Data Eng.*, 2007, pp. 106-115.

[64] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vision*, vol. 40, pp. 99-121, 2000.

[65] C. C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *Proc. VLDB*, vol. 5, 2005, pp. 901-909.

[66] B. C. Fung, K. Wang, and S. Y. Philip, "Anonymizing classification data for privacy preservation," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 5, pp. 711-725, 2007.

[67] J. W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient k-anonymization using clustering techniques," in *Int. Conf. Database Syst. Adv. Appl.*, Berlin, Heidelberg: Springer, 2007, pp. 188-200.

[68] X. Xiao and Y. Tao, "M-invariance: towards privacy preserving re-publication of dynamic datasets," in *Proc. 2007 ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 689-700.

[69] X. Xiao and Y. Tao, "Personalized privacy preservation," in *Proc. 2006 ACM SIGMOD Int. Conf. Manage. Data*, 2006, pp. 229-240.

[70] S. Zhong, Z. Yang, and R. N. Wright, "Privacy-enhancing k-anonymization of customer data," in *Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.*, 2005, pp. 139-147.

[71] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 279-288.

[72] G. Loukides, J. C. Denny, and B. Malin, "The disclosure of diagnosis codes can breach research participants' privacy," *J. Am. Med. Inform. Assoc.*, vol. 17, no. 3, pp. 322-327, 2010.

[73] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mobile Syst. Appl. Services*, 2003, pp. 31-42.

[74] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Proc. 2008 IEEE 24th Int. Conf. Data Eng.*, 2008, pp. 506-515.

[75] B. Malin and L. Sweeney, "How (not) to protect genomic data privacy in a distributed network:

using trail re-identification to evaluate and design anonymity protection systems," *J. Biomed. Inform.*, vol. 37, no. 3, pp. 179-192, 2004.

[76] O. Goldreich, *Foundations of Cryptography*, vol. 2. Cambridge: Cambridge University Press, 2004.

[77] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. Symp. Found. Comput. Sci.*, 1982, pp. 160-164.

[78] S. Micali, O. Goldreich, and A. Wigderson, "How to play any mental game," in *Proc. 19th ACM Symp. Theory Comput.*, New York: ACM, 1987, pp. 218-229.

[79] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Int. Conf. Theory Appl. Cryptographic Tech.*, Berlin, Heidelberg: Springer, 1999, pp. 223-238.

[80] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[81] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM J. Comput.*, vol. 43, no. 2, pp. 831-871, 2014.

[82] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, Berlin, Heidelberg: Springer, 2011, pp. 129-148.

[83] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Trans. Comput. Theory*, vol. 6, no. 3, pp. 1-36, 2014.

[84] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology–ASIACRYPT 2017*, vol. 10624, Cham: Springer, 2017, pp. 409-437.

[85] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proc. 3rd ACM Workshop Cloud Comput. Security Workshop*, 2011, pp. 113-124.

[86] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Des. Codes Cryptogr.*, vol. 71, pp. 57-81, 2014.

[87] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," in *Advances in Cryptology–CRYPTO 2014*, vol. 8616, Berlin, Heidelberg: Springer, 2014, pp. 297-314.

[88] G. R. Blakley, "Safeguarding cryptographic keys," in *Managing Requirements Knowledge, Int. Workshop*, IEEE Computer Society, 1979, pp. 313-313.

[89] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[90] Y. Lindell, "Secure multiparty computation," *Commun. ACM*, vol. 64, no. 1, pp. 86-96, 2021.

[91] T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," in *Proc. 21st Annu. ACM Symp. Theory Comput.*, 1989, pp. 73-85.

[92] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787-2805, 2010.

[93] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22-32, 2014.

[94] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Inf. Syst. Front.*, vol. 17, pp. 243-259, 2015.

[95] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414-454, 2014.

[96] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30-39, 2017.

[97] D. Evans, V. Kolesnikov, and M. Rosulek, "A pragmatic introduction to secure multi-party computation," *Found. Trends Privacy Security*, vol. 2, no. 2-3, pp. 70-246, 2018.

[98] Microsoft SEAL, "Microsoft SEAL Release 4.1," 2023. [Online]. Available: https://github.com/microsoft/SEAL. [Accessed Jan. 2023].

[99] B. Knott *et al.*, "CrypTen: Secure multi-party computation meets machine learning," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 4961-4973, 2021.

[100] M. Whaiduzzaman *et al.*, "A privacy-preserving mobile and fog computing framework to trace and prevent COVID-19 community transmission," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 12, pp. 3564-3575, 2020.

References

[101] Z. Kraljevic *et al.*, "Validating transformers for redaction of text from electronic health records in real-world healthcare," in *Proc. 2023 IEEE 11th Int. Conf. Healthcare Inform.*, 2023, pp. 544-549.

[102] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[103] i2b2 Dataset. [Online]. Available: https://www.i2b2.org.

[104] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Am. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 1, 2019, pp. 4171-4186.

[105] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[106] E. Alsentzer *et al.*, "Publicly available clinical BERT embeddings," *arXiv preprint arXiv:1904.03323*, 2019.

[107] OpenAI, "GPT-4 technical report," 2023. [Online]. Available: https://openai.com/research/gpt-4.

[108] Meta AI, "LLaMA 3.1," 2024. [Online]. Available: https://ai.facebook.com/research/llama.

[109] Anthropic, "Claude 3.5," 2024. [Online]. Available: https://www.anthropic.com/index/claude.

[110] J. Andrew, J. Karthikeyan, and J. Jebastin, "Privacy preserving big data publication on cloud using mondrian anonymization techniques and deep neural networks," in *Proc. 2019 5th Int. Conf. Adv. Comput. Commun. Syst.*, 2019, pp. 722-727.