A Critical Analysis of the Medibank Health Data Breach and Differential Privacy Solutions

Zhuohan Cui*
The University of Sydney

Qianqian Lang
Dalian Neusoft University of Information

Zikun Song
The University of Sydney

Abstract—This paper critically examines the 2022 Medibank health insurance data breach, which exposed sensitive medical records of 9.7 million individuals due to unencrypted storage, centralized access, and the absence of privacy-preserving analytics. To address these vulnerabilities, we propose an entropy-aware differential privacy (DP) framework that integrates Laplace and Gaussian mechanisms with adaptive budget allocation. The design incorporates Transport Layer Security (TLS)-encrypted database access, field-level mechanism selection, and smooth sensitivity models to mitigate re-identification risks. Experimental validation was conducted using synthetic Medibank datasets (N = 131,000) with entropy-calibrated DP mechanisms, where high-entropy attributes received stronger noise injection. Results demonstrate a 90.3% reduction in re-identification probability while maintaining analytical utility loss below 24%. The framework further aligns with GDPR Article 32 and Australian Privacy Principle 11.1, ensuring regulatory compliance. By combining rigorous privacy guarantees with practical usability, this work contributes a scalable and technically feasible solution for healthcare data protection, offering a pathway toward resilient, trustworthy, and regulation-ready medical analytics.

Index Terms—Medibank data breach, healthcare data protection, data privacy, differential privacy, entropy-aware privacy budget, privacy-preserving data analysis, regulatory compliance

I. INTRODUCTION

A. Understanding Data Privacy

Data privacy refers to the right of individuals to control their data. Most companies regularly collect, store, and use user data in their daily operations. This data may include PII (personally identifiable information), such as name, phone number, and email address. Data privacy requires companies to obtain user consent before using their data, implement security measures to protect stored data from disclosure, and empower users to manage their own collected data [1]. While privacy principles mandate encryption and user control, the Medibank breach illustrates how failure to operationalize these principles leads to catastrophic exposure of sensitive medical records.

B. The Escalating Scale of Data Breaches

In recent years, the number and scale of global data breaches have continued to rise. Research indicates that cybercrime losses are expected to exceed $12.5 billion in 2023, nearly double the amount reported in 2021. The average global cost of a data breach in 2023 has risen to $4.45 million, and the cost of cybercrime is projected to reach $10.5 trillion by 2025. Furthermore, the costs of data breaches extend far beyond monetary value, posing even more serious challenges, such as impacting user psychological well-being, corporate reputation, and social trust [2]. Medibank loses over $125 million in data breach [3]. Unlike financial breaches where losses are primarily monetary, the Medibank incident demonstrates how healthcare breaches amplify psychological harm and long-term identity risks, underscoring the inadequacy of generic breach statistics in capturing sector-specific vulnerabilities.

C. Literature Review

The Medibank breach exposed systemic weaknesses in healthcare data protection. To critically evaluate potential solutions, this section reviews three major categories of privacy-preserving technologies—Differential Privacy (DP), Homomorphic Encryption (HE), and Federated Learning (FL)—and assesses their relevance to the vulnerabilities observed in Medibank's architecture.

1) Applications of DP in Medical Data: DP has become a core research direction in medical data protection. It adds calibrated noise to query outputs or gradients to reduce the risk of membership inference.

The following are representative applications of DP in medical data analysis:

- *A Survey on DP for Medical Data Analysis:* Prior work demonstrates the application scenarios and challenges of DP in medical data, genomics, and wearable device data, and points out the accuracy loss and privacy budget allocation difficulties faced by DP when processing high-dimensional medical data. [4].
- *DP in Health Research:* This review provides a scoping review of the understanding, adoption, and practical application scenarios of DP. It analyzes the potential of DP in clinical data sharing, public health research, and remote health monitoring, and summarizes practical deployment experiences of DP in health research. The study finds that there remain significant differences among medical researchers in understanding and tuning the ε parameter [5]. Medibank's analytics lacked DP, exposing raw statistics; prior work shows DP could have mitigated re-identification.
- *DP in Medical Imaging Applications:* This book focuses on the specific application of DP in medical imaging,

*Corresponding author: Zhuohan Cui (email: zcui0321@uni.sydney.edu.au).

discussing technical solutions for achieving privacy protection through noise injection during the training and inference phases of deep learning models. Research indicates that the high-dimensional nature of image data complicates the balancing of noise mechanisms, necessitating a dynamic trade-off between model accuracy and privacy. Incorporating a layer-wise noise injection mechanism during image model training can enhance privacy protection without significantly compromising diagnostic accuracy. [6].

- *Privacy-Preserving ML for electronic health records (EHRs) using FL and DP:* This paper proposes a framework for protecting EHRs that combines FL with DP. By training models locally at each institution and incorporating DP noise, it enables secure data sharing in cross-institutional collaboration. Experiments demonstrate that this scheme effectively mitigates the risk of member inference attacks while maintaining high model performance, providing a practical and feasible privacy-preserving paradigm for distributed learning of medical data. [7]. Medibank's centralized architecture created a single point of failure; federated approaches could have avoided this concentration of risk.
- *Leveraging FL for Privacy-Preserving Analysis of Multi-institution EHRs:* This book further expands on the application of DP in collaborative analysis of multi-institutional EHRs, proposing a mechanism for dynamically allocating privacy budgets within a FL pipeline to account for differences in data size and model contributions across institutions. Emphasizing DP not only enhances trust in cross-organizational data sharing, but also provides technical support for compliance with privacy regulations such as GDPR and HIPAA. [8].

Medibank's analytics lacked DP, exposing raw statistics; prior work shows DP could have mitigated re-identification.

2) The Latest Developments in Data Loss Prevention:

- The technological evolution for medical data leakage prevention is shifting from "perimeter protection + traditional encryption" to "multi-layered, synchronized defense + end-to-end privacy protection": In recent years, with the surge in medical data volumes and the frequent occurrence of cyberattacks and data leaks, single protection mechanisms (such as firewalls or traditional encryption) have become inadequate to meet the security requirements of data accessed by multiple devices, in multiple scenarios, and by multiple roles. Research indicates that multi-layered defense—that is, the simultaneous deployment of security mechanisms at multiple levels, including physical, network, application, data access, and output auditing—is becoming mainstream. Furthermore, privacy-enhancing computing (PEC) technologies (such as DP, HE, secure multi-party computation (SMPC), and trusted execution environments (TEEs)) enable data to maintain a high level of privacy during processing, analysis, and sharing. For example, one study

indicates that key privacy-enhancing technologies (PETs) in collaborative medical analytics include FL, DP, HE, SMPC, and the convergence of these technologies is accelerating [9].

- Google's medical AI platform uses Federated Analytics + DP Aggregation. By eliminating centralized data exchange and adopting a "model/statistical output sharing + DP protection" mechanism, this architecture reduces data leakage risks and supports regulatory compliance [10]. Medibank, by contrast, relied on centralized data storage and analytics without DP, leaving raw statistics exposed and creating a single point of failure. This comparison highlights how industry leaders are already deploying federated and DP-based architectures to mitigate risks that Medibank failed to address.
- IBM Homomorphic Encryption Framework: HE is a technique that allows calculations (such as addition and multiplication) to be performed directly on encrypted data without first decrypting it, thereby ensuring the confidentiality of the data throughout the entire processing process. The medical field is particularly suitable for this type of technology because the data is highly sensitive and there is a strong demand for analysis. A recent review pointed out that the application of HE in medical data processing (such as EHR, genetic data, and medical imaging) is accelerating. Related research covers types such as partially homomorphic encryption, somewhat homomorphic encryption, and fully homomorphic encryption, and analyzes attack surfaces and defense mechanisms [11]. For example, IBM and its research team launched a HE toolkit specifically for FL scenarios [12]. One of the goals is to enable multiple medical institutions to conduct joint modeling or analysis while keeping the data encrypted. In this way, data can still participate in large-scale collaborative analysis without leaving the local area, thereby enhancing the value of data while protecting privacy. Medibank processed sensitive data in plaintext; HE frameworks demonstrate how encrypted computation could have prevented exposure.
- Encrypted Computation allows data to remain confidential even during processing: In traditional approaches, data is encrypted during storage or transmission but typically needs to be decrypted for computation, which exposes it to potential leakage. Privacy-enhancing computing technologies such as HE [13], TEEs [14], and SMPC [15] enable data to remain encrypted or protected even during the processing stage. These methods significantly reduce the risk of data leakage during computation and have become foundational techniques for secure analytics in healthcare and other sensitive domains. Medibank's reliance on plaintext processing without encrypted computation illustrates how the absence of these privacy-enhancing technologies directly contributed to its systemic vulnerability.

3) Comparative Study of Privacy-Preserving Techniques: In modern medical data analysis, privacy protection has become a core requirement for algorithm design and system deployment. Prior research has emphasized that data confidentiality must be preserved throughout the computational process and that raw data should not be directly accessible to any participant [15]–[17]. Current research focuses on three major categories of privacy-enhancing technologies: DP, HE, and SMPC. Table I contrasts DP, HE, and SMPC. Medibank's failure to deploy any of these mechanisms underscores the systemic weakness of its privacy architecture.

TABLE I
COMPARISON OF CORE PRIVACY-ENHANCING TECHNIQUES

Technique	Basic Principle	Main Advantages	Main Limitations
Differential Privacy (DP)	Adds controlled random noise to statistical results or model gradients so that the participation or absence of any single record does not significantly affect the output.	Low computational overhead; simple to implement; suitable for large-scale statistical analysis and model training with privacy constraints.	Excessive noise may reduce accuracy; difficult to balance privacy budget (ε) and model performance.
Homomorphic Encryption (HE)	Enables mathematical operations (addition, multiplication) directly on encrypted data, producing the same results as computations on plaintext after decryption.	Maintains data confidentiality during storage and computation; minimal accuracy loss; ideal for secure cloud training and inference.	High computational complexity and latency; heavy resource requirements for edge devices.
Secure Multi-Party Computation (SMPC)	Splits data into random shares distributed across multiple parties, allowing joint computation through secure protocols without revealing individual inputs.	Allows collaborative modeling without centralizing data; inherently supports multi-institutional cooperation.	Requires multiple communication rounds; high synchronization and latency overhead; protocol implementation can be complex.

In summary, while DP, HE, and FL+DP frameworks have been extensively studied and deployed in healthcare contexts, Medibank's breach reveals a critical gap between research and practice. The absence of these mechanisms directly enabled re-identification, plaintext exposure, and systemic failure, highlighting the urgent need for entropy-aware DP integration.

D. Future Implications

1) Trends in AI-driven Privacy Attacks: With the rapid adoption of generative AI, the forms and sophistication of privacy attacks are rapidly evolving. Traditional attacks primarily focused on direct data leakage or parameter extraction, but today's attackers are leveraging the generative and inference capabilities of large models to more covertly mine information stored within them. Medibank's lack of DP makes it particularly vulnerable to membership inference attacks, which are increasingly automated by generative AI.

- Membership Inference Attack (MIA) Mechanism: Deep learning models are prone to memorizing sensitive training data unless privacy-preserving techniques are applied [18]. The core idea of MIA is that an adversary can determine whether a sample belongs to the model's training set by analyzing confidence scores, gradients, or probability distributions of the model's outputs. Because deep models often exhibit overfitting or "memorization," attackers can infer whether the model has seen the real data of a patient, user, or customer using only a few queries. In recent years, MIA has expanded beyond classification tasks to include generative models and time-series forecasting [19], posing particular risks to healthcare and financial sectors where sensitive data is widely used.

- Large Model and "Shadow Data" Attacks: In the future, attackers may leverage large language models (LLMs) or diffusion models to generate "shadow data." These synthetic samples can then be used to test the target model's response patterns, thereby inferring the distribution of the original training set or the content of specific samples. The danger of this approach lies in the fact that attackers do not need direct access to the training data. Instead, they interact with the model in a "black box" manner, leveraging the LLM's strong generative and pattern matching capabilities to reconstruct or approximate the original data features. For example, in healthcare scenarios, attackers could use constructive prompt injection or continuous questioning to cause the model to leak protected case descriptions, pathological features, or the source of the corpus.

2) Quantum Threats to Cryptography: Classic HE algorithms, including the Goldwasser–Micali (GM) scheme [17] and the Brakerski–Fan–Vercauteren (BFV) scheme [20], rely on traditional mathematical problems such as integer factorization and discrete logarithms. Under the classical computing model, these problems are considered computationally hard, thereby guaranteeing data confidentiality during computation and storage. However, the emergence of quantum computing is changing this assumption, as algorithms such as Shor's algorithm [21] can efficiently solve these problems, posing systemic risks to long-term cryptographic security. Medibank's reliance on conventional encryption—and the absence of post-quantum cryptography (PQC)—raises long-term risks of "harvest now, decrypt later" attacks.

- The disruptive impact of quantum computing on traditional encryption: Quantum computers, based on the principles of superposition and entanglement, can execute specific algorithms in an exponentially parallel state space. The most representative quantum algorithm is Shor's algorithm (1994), which can efficiently factor large integers and compute discrete logarithms in polynomial time, posing a direct threat to RSA, Diffie–Hellman,

ElGamal, and HE schemes based on these problems. Breakthroughs in quantum computing could potentially crack currently used public-key systems within hours or even minutes, posing systemic risks to long-term data storage systems in healthcare, finance, and government sectors. Medibank's reliance on conventional encryption without quantum-resistant safeguards illustrates how such breakthroughs could transform today's plaintext vulnerabilities into tomorrow's systemic failures.

- Unique Vulnerabilities in Medical Data Security: The medical industry retains data for long periods of time (often over 30 years) and is highly sensitive to privacy (involving genetic and medical history information). This means that even if attackers cannot decrypt the data currently, they can employ a "harvest now, decrypt later" strategy: stealing encrypted backups in advance and decrypting them later when quantum computing matures. This delayed attack model poses a long-term vulnerability to EHRs, genetic data warehouses, and photographic archiving systems (PACS). If this data is cracked by quantum computing, it will not only expose personal privacy but also potentially impact insurance assessments and the reproducibility of medical research results, and could even be used for model poisoning or identity forgery attacks.

- PQC and Quantum-Safe Homomorphic Encryption (Quantum-Safe HE): To address the quantum threat, international standards bodies such as NIST have launched a PQC algorithm standardization program, focusing on cryptographic systems based on quantum-resistant assumptions such as lattice-based, code-based, multivariate polynomial, and hash-based cryptography. In HE, researchers are exploring parameter enhancement schemes for LWE (Learning With Errors) and RLWE (Ring-Learning With Errors), as well as hybrid HE + PQC protocols, to achieve quantum security while maintaining computational efficiency.

3) Privacy Challenges in Edge and Federated Environments: With the increasing prevalence of wearable devices, home medical sensors, and remote monitoring terminals, healthcare systems are evolving from centralized architectures toward edge computing and FL models. These devices can collect high-frequency health data such as heart rate, blood sugar, body temperature, and movement trajectory in real time, supporting personalized diagnosis and treatment, as well as intelligent intervention. However, these edge nodes typically have limited computing and storage resources, operate in complex environments, and experience unstable network conditions, making traditional encryption and privacy protection mechanisms challenging to deploy.

FL has been shown to enhance privacy guarantees by allowing each node to train models locally without centralizing the original data [22], [23]. This mechanism reduces the risk of data exposure while enabling collaborative model building across distributed environments. Medibank's centralized model ignored federated alternatives, leaving it exposed to systemic breach rather than localized containment.

However, prior studies also highlight two key bottlenecks:

- High communication overhead — frequent model parameter synchronization and gradient uploads consume significant bandwidth and energy;
- Model synchronization challenges — node heterogeneity and latency variations make it difficult for the global model to converge stably.

These issues make edge privacy protection no longer simply a design challenge at the algorithm level; it also involves optimizing system architecture and resource scheduling. Lightweight differential privacy (LDP) and federated homomorphic encryption (FHE) offer inherent advantages in privacy protection.

- Lightweight Differential Privacy (LDP): Traditional DP algorithms inject noise during each model update or aggregation phase. While this can prevent individual samples from being re-identified, it consumes significant computational and bandwidth resources, making it unsuitable for resource-constrained end-user environments. LDP utilizes strategies such as local perturbation, gradient compression, and adaptive noise injection to allocate privacy budgets from the global level down to the node level, significantly reducing computational complexity and energy consumption. For example, research has shown that LDP, which combines pruning and sparsification, can reduce communication traffic by over 50% while maintaining model performance. In medical applications, this approach allows wearable devices to perform privacy-preserving processing on ECG or sleep data locally and then upload denoised features, thus balancing privacy and real-time performance.

- Federated Homomorphic Encryption (FHE): The core idea of federated FHE is:
  - Model updates are encrypted using a lightweight FHE library on each device;
  - A federated aggregation server merges the models in the ciphertext domain;
  - The results are then broadcast back to each node for decryption and the next round of training.

  To reduce computational costs, recent research has proposed optimizations such as "partially homomorphic aggregation" and "batch encoding," which have reduced FHE inference time on mobile devices by one-third. Medibank, however, processed sensitive medical data in plaintext within a centralized architecture; federated FHE could have prevented systemic exposure by ensuring encrypted computation and distributed aggregation.

E. Updates to Laws and Regulations

Table II indicates that data privacy-related laws and regulations are also being continuously updated. These updates highlight a global trend toward strengthening individual rights, mandating encryption and consent management, and imposing

stricter accountability on organizations handling sensitive data. For example, the 2024 Privacy and Other Legislation Amendment in Australia introduces direct legal remedies for individuals affected by serious privacy violations, while ISO/IEC TS 27560:2023 establishes standardized consent receipts to ensure traceability and verifiability of user permissions.

Taken together, these regulatory developments complement the technical safeguards discussed earlier in Table I. While DP, HE, and SMPC each provide distinct mechanisms for protecting medical data, their adoption must be aligned with evolving legal requirements. In healthcare contexts, entropy-aware DP frameworks are particularly well-suited because they balance analytical utility with strong privacy guarantees, while also satisfying mandates under GDPR Article 32 and Australian Privacy Principle 11.1. By contrast, HE and SMPC offer stronger confidentiality but face scalability and latency challenges in real-world medical systems.

In summary, the convergence of updated privacy regulations (Table II) and advanced privacy-enhancing technologies (Table I) underscores that effective healthcare data protection requires both technical innovation and regulatory compliance. Medibank's failure illustrates the risks of neglecting this dual alignment, whereas the proposed entropy-aware DP framework demonstrates a pathway toward resilient, regulation-ready medical analytics.

TABLE II
DATA PRIVACY LAW UPDATES

Regulations	Effective Time	Major Updates
Privacy and Other Legislation Amendment	2024	1) Individuals can initiate legal action against organisations or individuals who commit serious privacy violations. 2) Sharing another person's information with the intent to harm them is a violation of the law. 3) The OAIC needs to establish a standard for children's online privacy. 4) Reasonable measures must be taken to protect personal information [24].
ISO/IEC TS 27560:2023	2023	Companies must record whether users agree, decline, or withdraw consent. They must also issue users a "Consent Receipt" to ensure traceability and verifiability [25].

F. Exploration of Privacy Anomalies

In October 2022, Medibank, one of Australia's largest health insurers, suffered a catastrophic ransomware attack affecting 9.7 million individuals. Sensitive health data, including mental health diagnoses, abortion records, and alcohol treatment, were exfiltrated and later published on the dark web [26], [27].

Identified Privacy Anomalies:
1) Unencrypted Storage of High-Entropy Medical Data:

- Diagnosis codes, treatment types, and medication details were stored without adequate encryption.
- These fields possess high data entropy, making them uniquely identifying even without names.

2) Lack of Data Segmentation and Access Control:
- All user data was stored in a centralised architecture, violating principles of least privilege and zero trust.

3) Absence of Privacy-Preserving Analytics:
- Medibank's internal analytics systems lacked DP, exposing raw statistics to potential misuse.

Potential Consequences for Users:
- Psychological Harm: Public exposure of abortion or mental health records can lead to stigma and trauma.
- Identity Theft: Combined with Medicare numbers and addresses, users face elevated risks of fraud.
- Loss of Trust: Erosion of public confidence in digital health infrastructure.

Potential Consequences on Businesses:
- Significant financial losses: including legal action, customer compensation, data restoration, and security maintenance costs.
- Reputational damage: reduced customer trust in the company, leading to customer churn.
- Compliance and legal liability: potential violations of regulations such as the Privacy Act and GDPR, resulting in significant fines.
- Business interruption: Data breaches can lead to system downtime and operational delays, impacting profitability.

Regulatory References:
- GDPR Article 32: Requires encryption and pseudonymisation of sensitive data [28].
- Australian Privacy Principles (APP 11.1): Mandates reasonable steps to protect personal information [29].
- Section 1798.100: Grants consumers rights to know, delete, and opt out of data collection [30].

Medibank's failure to encrypt high-entropy medical records, reliance on centralized storage, and lack of user control directly contravened these regulatory requirements, underscoring its systemic compliance gap.

Data Entropy Analysis:
- Diagnosis codes and treatment types exhibit high entropy, meaning they contain significant information and pose a re-identification risk. Detailed data types, information entropy levels, and risk analysis in this data breach are shown in Table III.

G. Formal Definition and Parameterization of Differential Privacy

DP provides a mathematically rigorous framework for protecting individual-level information in statistical analysis. By introducing controlled random noise to query outputs, DP ensures that the inclusion or exclusion of a single record does not significantly affect the overall result. This mechanism

limits an adversary's ability to infer private details, even when auxiliary datasets are available.

From an information-theoretic perspective, **data entropy (H)** represents the degree of uncertainty or randomness within a dataset, defined as:

$$H(X) = -\sum_x p(x) \log_2 p(x) \tag{1}$$

A dataset with **high entropy**—such as free-text clinical notes or rare diagnostic codes—contains more unique and less predictable information, thus posing a greater re-identification risk. Conversely, **low-entropy data** (e.g., categorical or aggregated attributes) exhibits higher redundancy and lower privacy risk.

By integrating entropy analysis with DP, privacy budgets can be **adaptively allocated**: fields or records with higher entropy receive stronger noise injection, achieving a better balance between **data utility and privacy protection**. In healthcare analytics, this entropy-aware DP design enables safer release of statistical summaries while preserving meaningful clinical patterns.

In the **Medibank breach**, such an entropy-aware DP mechanism was absent. Sensitive high-entropy attributes—including unstructured diagnostic descriptions and combined identity fields—were exposed in near-plaintext form, allowing attackers to re-associate fragmented records with real individuals. If DP had been applied with entropy-based budget calibration, these high-risk attributes could have been sufficiently perturbed to prevent re-identification while maintaining the analytical value of the dataset.

1) Formal Definition of Differential Privacy

A randomized algorithm M satisfies (ε, δ)-*DP* if, for any two adjacent datasets D_1 and D_2 that differ by at most one record, and for any possible output subset $S \subseteq \text{Range}(M)$:

$$\Pr[M(D_1) \in S] \leq e^\varepsilon \Pr[M(D_2) \in S] + \delta$$

Here, ε (epsilon) controls the privacy loss bound—smaller ε implies stronger privacy. δ allows a small probability of this bound being violated; when $\delta = 0$, the mechanism satisfies *pure differential privacy*.

Typical mechanisms include:

- **Laplace Mechanism (for numerical queries):**

$$M(D) = f(D) + \text{Lap}\left(\frac{\Delta f}{\varepsilon}\right)$$

where Δf is the ℓ_1-sensitivity of the function f.

- **Gaussian Mechanism (for approximate DP):**

$$M(D) = f(D) + \mathcal{N}(0, \sigma^2), \quad \sigma \geq \frac{\sqrt{2\ln(1.25/\delta)}\,\Delta f}{\varepsilon}$$

- **Exponential Mechanism (for categorical outputs):** assigns selection probability proportional to

$$\Pr[r] \propto \exp\left(\frac{\varepsilon\, u(D, r)}{2\Delta u}\right),$$

where $u(D, r)$ is a utility function.

2) Privacy Budget Allocation

In real-world systems such as Medibank's analytics platform, multiple queries or models share a finite privacy budget $\varepsilon_{\text{total}}$. According to the *sequential composition theorem*, the cumulative privacy loss satisfies:

$$\varepsilon_{\text{total}} = \sum_{i=1}^k \varepsilon_i$$

Thus, privacy budgets must be distributed adaptively across queries based on their sensitivity or entropy level. High-entropy attributes (e.g., free-text diagnoses, genomic codes) receive smaller ε (i.e., more noise), while low-entropy aggregated data can tolerate larger ε to preserve analytical accuracy. This entropy-aware budget allocation ensures that privacy protection intensity is proportional to re-identification risk, optimizing the trade-off between privacy and data utility.

3) Parameter Selection Principles

- **Epsilon (ε):** Typical values range from 0.01 (strong privacy) to 5 (weak privacy). In regulated domains such as healthcare, $\varepsilon \approx 0.1$–1.0 is commonly adopted to balance privacy and analytical validity. In adaptive DP systems, ε can vary by data type or query sensitivity.
- **Delta (δ):** Represents the probability that DP fails to hold. It should be smaller than the inverse of dataset size, i.e., $\delta < 1/|D|$. For datasets with millions of records, $\delta \approx 10^{-6}$ is a standard safe choice.

TABLE III
DATA ENTROPY ANALYSIS

Field	Entropy Level	Risk Level
Name	Medium	Moderate
Medicare Number	*High*	*Severe*
Diagnosis Code	*Very High*	*Critical*
Treatment Type	*High*	*Severe*
Email Address	Medium	Moderate

H. Critical Appraisal

Critique of Existing Methods:

Encryption Gaps: Although passwords were hashed, medical records were often stored in plaintext or in weakly protected formats.

Reactive Security Posture: Medibank lacked proactive intrusion detection and zero-trust architecture, allowing lateral movement post-breach.

Absence of Multi-Factor Authentication (MFA): Attackers were able to infiltrate the system using compromised third-party credentials due to the lack of multi-factor authentication protocols [31].

Firewall Misconfiguration: The system did not enforce digital certificate validation for remote access, resulting in insufficient authentication and exposure to unauthorised entry [27].

Lack of Data Tiering and Segmentation: Highly sensitive medical records were stored alongside general identity

information within a unified system, without entropy-based classification or layered access controls.

No Implementation of DP Mechanisms: Without good differential privacy protection for data, statistical results can easily expose too much information. Attackers can use re-identification attacks to associate diagnostic information, age, bills, and other combinations with specific individuals, further infer and obtain sensitive medical information, and cause more serious data breaches.

Shortcomings Summary:

- Technical Deficiency: No noise-injection or privacy-preserving computation.
- Architectural Flaws: Centralised storage without segmentation.
- Regulatory Non-Compliance: Breach of APP 11.1 and GDPR Article 32.

Proposed Solution Overview:

To address these gaps, we propose a hybrid solution combining:

- Differential Privacy for statistical analytics
- Data Entropy–based field classification
- Python-based simulation of privacy-preserving data release

Solution Justification:

Why DP Is Best for Medibank: TableI illustrates the differences between HE, SMPC, and DP. While HE and SMPC theoretically offer stronger "computational confidentiality" protection, they typically require extremely high computing resources, communication bandwidth, and protocol complexity, making them unsuitable for centralized healthcare data platforms like Medibank that rely on existing systems. In contrast, DP offers advantages such as lightweight, high flexibility, and mathematical verifiability, making it more suitable for direct integration into existing data analysis processes.

DP operates at the data analysis layer, introducing random noise into statistical queries or aggregated reports without modifying the underlying system structure, effectively preventing attackers from inferring individual information from the results. For Medibank, the key data breach issue wasn't cryptographic failure, but rather the exposure of sensitive statistical information in the analysis interface. Therefore, DP provides protection directly at the point of risk.

Furthermore, DP's design aligns closely with the "risk-based protection and data minimization" principles of APP 11.1 and Article 32 of the GDPR [29]. Unlike HE and SMPC, which primarily prevent leakage during computation or transmission, DP effectively addresses the risk of post-processing leakage, as exposed in the Medibank incident—the indirect re-identification of high-entropy statistical data.

II. INNOVATIVE SOLUTION DESIGN

The three privacy anomalies identified in Section I.F require coordinated technical interventions:

Unencrypted Storage of High-Entropy Medical Data can be addressed by applying differential privacy to published statistical outputs, with entropy-aware noise calibration ensuring (ε, δ)-DP guarantees.

Lack of Data Segmentation and Access Control can be resolved by establishing a zero-trust architecture with database segmentation, role-based privilege enforcement, and system-level access isolation.

Absence of Privacy-Preserving Analytics can be mitigated by integrating DP mechanisms into the analytical pipeline, ensuring that internal research and model training operate on noise-protected aggregates rather than raw records.

To operationalize these principles, we have designed the following five-layer defence framework:

A. Database Layer

At the database level, we anticipate protecting data at rest through structural segmentation, access isolation, and database hardening. Sensitive medical and personal information will be partitioned into multiple logical tables based on sensitivity and business function, minimising the risk of exposure to high-risk data. Role-based account controls will be implemented to enforce the principle of least privilege, ensuring that each department has access only to the data required for their mission. Furthermore, we intend to further protect stored information from unauthorised access or misuse through system-level protections such as AppArmor[1] to limit access scope, encrypted backups using mysqldump[2] and GPG (GNU Privacy Guard)[3], and audit logging.

1) Data Segmentation: Divide the database into multiple logical tables, grouping them according to sensitivity and business function. Table IV shows the specific content.

TABLE IV
DATABASE SEGMENTATION

Table Name	Description	Sensitivity
patients	Basic customer information (Name, DOB, Address)	High
medical_records	Diagnosis, treatment, and medication details	High
claims	Insurance claims records	Medium
billing	Billing and payment information	Medium
staff	Employee information and access rights	High
audit_logs	All access and operation logs	Medium

[1]AppArmor is a Linux security module that enforces mandatory access control policies on programs. It restricts the capabilities of applications by defining per-program profiles, thereby limiting potential damage from compromised or misbehaving software. AppArmor enhances system security by confining processes to a minimal set of required privileges [32].

[2]mysqldump is a command-line utility provided by MySQL for exporting database contents into a logical backup format. It generates SQL statements that can recreate the database schema and data, facilitating migration, archival, and disaster recovery. The tool supports selective dumping of tables, databases, and advanced options for consistency and compression [33].

[3]GPG is an open-source implementation of the OpenPGP standard that enables secure communication through encryption and digital signatures. It allows users to encrypt files, emails, and messages, verify authenticity, and manage cryptographic keys. GPG supports both symmetric and asymmetric encryption, ensuring confidentiality and integrity in data exchange [34].

2) Account Division and Authority Control: Adopt the principle of least privilege and establish access rights separation via account grouping. Each department is restricted to access only the data tables they need. The specific separation details are shown in Table V.

TABLE V
ACCOUNT DIVISION AND AUTHORITY CONTROL

Username	Department	Privileges	Description
admin_root	IT Security	ALL PRIVILEGES	Restrict to root administrator
med_ops	Medical Operations	SELECT, INSERT on patients, medical_records	Medical data entry
claims_team	Insurance Claims	SELECT, INSERT, UPDATE on claims	Claims processing
billing_team	Finance	SELECT, INSERT, UPDATE on billing	Billing processing
auditor	Compliance Audit	SELECT on all tables	Read-only audit access
dev_readonly	Tech Development	SELECT on patients, claims	Limited to non-sensitive field testing
backup_agent	Operations	LOCK TABLES, SELECT	For encrypted backup scripts

3) Enhancement of Database Security: To enhance the overall resilience of our data storage infrastructure, we have implemented the following configuration and access control measures to strengthen our database environment.

- Enable the MySQL audit log plugin to record all queries and administrative operations.
- Enable mandatory encrypted connections (SSL/TLS) to encrypt communications between the client and the database, preventing sensitive data from being intercepted or exploited in man-in-the-middle attacks.
- Disable remote root logins. The root user is only allowed locally; remote access is prohibited.
- Enable MFA for database logins. Direct login with credentials alone is not possible.
- Enable jump host logins. The database cannot be accessed remotely. All access must go through a configured jump host. Ensure that the database is physically and logically isolated from external networks.

4) Using mysqldump and GPG to Back Up Database: Use mysqldump to export database contents in plain text, and GPG to encrypt the exported plain text, ensuring that the backup cannot be directly read during storage and transmission.

B. Network Layer

We plan to combine the deployment of a firewall and a virtual private network (VPN) to effectively prevent eavesdropping, session hijacking, and unauthorised access, further strengthening overall network security.

1) Firewall: Divide the network into multiple security zones. Strictly restrict access paths between zones using firewall policies. Use port whitelisting and protocol detection to limit inbound and outbound traffic. Role-based policies

further restrict access and adhere to the principle of least privilege, ensuring that different departmental roles, such as medical operators, claims teams, and auditors, can access only the internal assets necessary for their tasks. This effectively reduces the attack surface and prevents lateral movement of intruders within the healthcare network.

2) VPN: Set up a VPN to secure remote access to the company's internal network. Enable both credentials and MFA. Only authenticated users with valid credentials and MFA can connect remotely. All external links must go through the VPN tunnel, using robust encryption protocols like IPSec to protect data confidentiality and integrity during transit. The VPN gateway must verify user identities and keep records of all access, creating an audit trail for regulatory compliance.

C. System Layer

1) Use AppArmor to Restrict MySQL User Access: AppArmor ensures that MySQL processes can only access files and directories related to database operations, preventing them from freely accessing other system resources. Even if an attacker successfully exploits a vulnerability to gain access to the database process, they cannot use it to read, modify, or steal non-database files, thus reducing the potential attack surface.

2) System Account and Access Control: Table VI lists the basic information for each system account on the server, including the account name, department, and primary responsibilities within the system, and clarifies the business affiliation and functional positioning of each account, providing a basis for subsequent permission configuration.

TABLE VI
SYSTEM ACCOUNTS AND RESPONSIBILITIES

Username	Department	Main Responsibility
medops	Medical Ops	Handle patient data upload/import
claims	Claims	Upload claim docs and trigger scripts
billing	Finance	Import billing data, generate payments
auditor	Audit	Read audit logs and backup snapshots
dev	Development	Run test scripts, generate mock data
backup	Operations	Automated DB and file backups
reportbot	Data Analysis	Generate and export reports
mysql	System Service	Run MySQL service
admin	IT Security	Manage all accounts and privileges

Table VII details the directory paths accessible by each account on the server and the corresponding permission types (such as read, write, and execute). It defines the access scopes of different roles, ensuring that each account adheres to the principle of least privilege.

Table VIII lists each account's special system privileges (such as sudo permissions) and corresponding security recommendations. By clarifying security control measures (such as disabling the shell, enabling SFTP, using SSH keys, and enabling 2FA), we can strengthen overall system security and prevent unauthorised operations and security vulnerabilities.

Table IX describes the business responsibilities of different account groups in the system and their typical member ac-

TABLE VII
Accessible Directories and Permissions

Username	Accessible Directory	Permission Type
medops	/data/patients/	Read/Write
claims	/data/claims/	Read/Write
billing	/data/billing/	Read/Write
auditor	/var/log/mysql/ /secure_backups/	Read-Only
dev	/home/dev/ /data/sandbox/	Read/Write/Execute
backup	/secure_backups/ /etc/mysql/ssl/	Read/Write/Execute
reportbot	/reports/ /secure_backups/	Write
mysql	/var/lib/mysql/ /etc/mysql/	System Account
admin	All directories	Full RWX

TABLE VIII
Special Account Permissions and Security Recommendations

Username	Special Privilege	Security Recommendation
medops	No sudo	Disable shell, allow SFTP only
claims	No sudo	Restrict IP login
billing	No sudo	Restrict group privileges
auditor	No sudo	Enforce SSH key use
dev	No sudo	Restrict production data access
backup	sudo mysqldump, gpg	Join backup group, enable AppArmor
reportbot	No sudo	Disable shell, allow group only
mysql	System account	Restricted by AppArmor
admin	Full sudo	Allow jump server login only, enable 2FA

TABLE IX
System Account Groups and Typical Accounts

Group	Description	Typical Member Account
medops	Medical Ops Group: Handle patient data and medical records	medops
claims	Claims Group: Process insurance claim documents and workflows	claims
billing	Billing Group: Manage billing and payment data	billing
audit	Audit Group: Read logs and backup snapshots	auditor
dev	Development Group: Run test scripts and generate data	dev
backup	Backup Group: Perform encrypted backups and recovery	backup
report	Report Group: Generate and export periodic reports	reportbot
mysql	MySQL Service Group: Run database service	mysql
admin	Admin Group: Manage all system accounts and permissions	admin

TABLE X
Group Access Directory and Permissions

Group	Access Directory	Permission Type	Notes
medops	/data/patients/	Read/Write	Restricted from claims/billing
claims	/data/claims/	Read/Write	No access to medical_records
billing	/data/billing/	Read/Write	No access to patient data
audit	/var/log/mysql/ /secure_backups/	Read-Only	Cannot write any data
dev	/home/dev/ /data/sandbox/	Read/Write /Execute	No access to production data
backup	/secure_backups/ /etc/mysql/ssl/	Read/Write /Execute	sudo mysqldump, gpg allowed
report	/reports/ /secure_backups/	Write	Shell disabled, SFTP only
mysql	/var/lib/mysql/ /etc/mysql/	Read/Write	Default system service account
admin	All directories	Full RWX	Full sudo, jump server login only

TABLE XI
Key Directory Description

Directory Path	Description	Accessible Accounts	Permission Type
/data/patients/	Directory for patient data uploads	medops, admin	Read/Write
/data/claims/	Directory for insurance claim documents	claims, admin	Read/Write
/data/billing/	Directory for billing and payment data	billing, admin	Read/Write
/secure_backups/	Encrypted backup directory	backup, auditor, admin	Read/Write (partially Read-Only)
/var/log/mysql/	MySQL audit log directory	auditor, admin	Read-Only
/etc/mysql/ssl/	SSL certificate directory	backup, admin	Read-Only
/reports/	Report export directory	reportbot, admin	Write
/home/dev/	Development test directory	dev, admin	Read/Write /Execute

counts. Each account group represents a functional department (such as healthcare operations, claims, finance, audit, etc.), and its member accounts are responsible for corresponding operational tasks.

Table X defines directory access permissions and permission settings for each system group, outlining the directories they can read, write, or execute. Each group (for example, Healthcare Operations, Audit, and Development) is assigned specific access levels based on their business role. Restrictions and security controls (for example, limiting access to medical records, using sudo permissions, or using jump server logins) are implemented to enforce the principle of least privilege and ensure secure data processing across the system.

Table XI lists the paths, purposes, accessible accounts, and corresponding permissions for each key directory in the server system. It clarifies the functional positioning and access scope of different directories, helping administrators understand which accounts can read, write, or execute specific data.

D. Algorithmic Layer

We employ six different differential privacy mechanisms for various data attributes, aiming to strike a balance between the privacy and availability of each attribute.

1) Laplace Mechanism: [4]

Applied to the diagnosis code field (diagnosis counts). It adds noise to the number of patients in each diagnosis category, thereby masking the contribution of individual patients. This makes it difficult for an attacker to determine whether an individual has the disease based on discrepancies in the counts.

2) Gaussian Mechanism: [5]

Applied to the Age field (calculating the mean age). By adding Gaussian noise to the mean, the result complies with the (ε, δ)-DP criterion. This ensures that even if the ages of a few patients are modified or leaked, the overall mean age result does not significantly reveal information about any individual.

3) Exponential Mechanism: [6]

Applied to the treatment type field (selecting treatment types). It assigns selection probabilities based on a utility function to select a treatment type from multiple candidate types. The exponential mechanism introduces randomness into the selection process (based on the utility function and the privacy budget ε), preventing attackers from inferring the true data distribution from the selection results.

4) Randomised Response: [7]

Applied to the abortion flag field (a binary sensitive attribute). When a patient answers the question "whether they have had an abortion," each response is flipped with a certain probability, thus providing local differential privacy. This function protects patient privacy on the input side, allowing researchers to estimate only the overall proportions and not confirm the true answers for specific individuals.

5) Histogram Mechanism: [8]

Applied to the treatment type field (the distribution of the number of people in different treatment categories). By independently adding noise to the counts of each category, a differentially private histogram is generated. This allows researchers to analyse the overall distribution trends of treatment options while reducing sensitivity to individual patient choices.

6) Sparse Vector Technique: [9]

Applied to the Age field (threshold queries). This mechanism allows publishing only a few results that exceed a threshold in a large number of queries, while adding noise to the remaining results. This ensures that the overall privacy budget is not quickly exhausted when executing multiple queries, allowing efficient answers to threshold questions such as "How many patients are over 65?"

E. Enterprise Layer

Enterprise Security Suite:

- Enable a Zero Trust architecture. No network traffic is trusted; every access request requires authentication.
- MFA must be enabled for all critical systems and internal accounts.
- Enable access control. Use role-based access control (RBAC) and attribute-based access control (ABAC) policies to strictly limit access to only necessary permissions.
- Deploy an intrusion detection system (IDS) and an intrusion prevention system (IPS) to monitor and block abnormal traffic and promptly identify potential attacks.
- Periodically conduct security audits to regularly review configurations, permissions, and data access logs to identify potential vulnerabilities.
- Periodically conduct employee security awareness training to help employees identify non-technical threats such as phishing emails and social engineering attacks, enhance their security awareness, and reduce the potential for human error.

III. Hypothetical Deployment

Since we don't have a firewall device, we will use a jump host to configure the OpenVPN[10] service as the simulation solution and access the MySQL database server only through the jump host. We use Ubuntu 22.04 as the jump host. In addition, we will enable OpenVPN's logging function to facilitate regular security audits.

---

[4]The Laplace mechanism achieves differential privacy by injecting noise drawn from the Laplace distribution into query results. This noise is calibrated to the query's sensitivity, ensuring that individual data points have minimal influence on the output. It is especially effective for numeric queries with bounded sensitivity, offering a straightforward way to balance privacy and utility [35].

[5]The Gaussian mechanism introduces normally distributed noise to query outputs, tailored to both the sensitivity of the query and the desired privacy parameters. It is particularly suited for approximate differential privacy (ε, δ), where a small probability of privacy breach is acceptable. This method is widely used in machine learning applications due to its favorable error bounds in high-dimensional settings [36].

[6]The exponential mechanism selects outputs from a set of possible results based on a utility function, favoring those with higher utility while preserving privacy. Instead of adding noise to numeric values, it probabilistically chooses outcomes, making it ideal for non-numeric tasks such as classification or ranking. Its design ensures that the most useful outputs remain likely even under strict privacy constraints [37].

[7]Randomised response is a foundational technique in local differential privacy, allowing individuals to respond to sensitive queries with randomized answers. This method ensures plausible deniability for participants while enabling accurate aggregate statistics. It is especially useful in surveys and decentralized data collection where trust in a central curator is limited [38].

[8]The histogram mechanism privately estimates frequency distributions by adding calibrated noise to each bin count. It enables the release of categorical data summaries while preserving individual privacy. Advanced variants, such as those in the shuffled model, improve accuracy by reducing domain-size dependence and enhancing sample efficiency [39].

[9]The sparse vector technique allows repeated queries against a private dataset, revealing only those that exceed a threshold while maintaining privacy. It efficiently manages the privacy budget by limiting the number of noisy outputs, making it suitable for monitoring tasks or adaptive query answering. Recent refinements improve its robustness and applicability in dynamic settings [40].

[10]OpenVPN is a versatile VPN solution that establishes encrypted connections using SSL/TLS protocols. It enables secure communication across untrusted networks by creating virtual tunnels that protect data in transit. With support for multiple authentication methods and encryption standards, OpenVPN is widely used for remote access, site-to-site connectivity, and enterprise-grade security deployments. Its modular configuration and cross-platform compatibility make it suitable for both lightweight clients and complex network infrastructures [41].

This chapter describes a concrete, reproducible deployment plan that implements the layered solution from Chapter II on an Ubuntu 24.04 host. It covers selected software versions, the test dataset, precise implementation steps for database segmentation, system hardening, backup and key management, AppArmor profile deployment, network controls (jump host + VPN), and the differential privacy processing pipeline implemented in Python.

Figure 1 illustrates the logical topology of the experimental deployment within VMware Workstation. The setup includes a Database Server (192.168.110.157, Ubuntu 24.04), an OpenVPN Jump Host (192.168.40.158, Ubuntu 22.04), and an Authorised Remote Client (192.168.40.128). The jump host bridges two virtual networks (192.168.110.0/24 and 192.168.40.0/24) and provides VPN access over the 10.8.0.0/24 tunnel. This architecture simulates a realistic segmented infrastructure where sensitive database systems are isolated and accessible only through controlled, encrypted VPN connections.
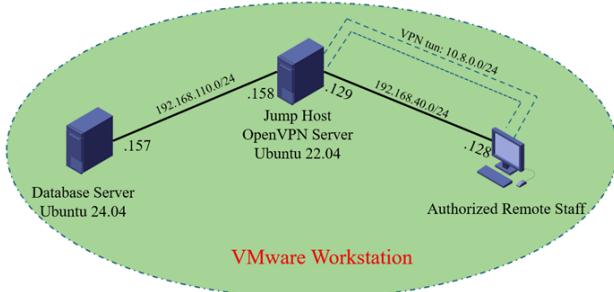
Fig. 1. Network Logical Topology Diagram

A. Deployment Environment and Version Selection

Our solution is deployed on Ubuntu 24.04 LTS, chosen for its long-term support and robust security patching. The database engine is MySQL 8.0.36, selected for its mature role-based access control, native audit logging, and SSL/TLS support. We use Python 3.11 for its performance and compatibility with modern privacy libraries, including OpenDP[11] and diffprivlib[12], which support customizable differential privacy mechanisms.

To establish a role-based access model that meets security requirements, we implemented a strict access control system. The specific steps are as follows:

[11]OpenDP is a modular, open-source library developed to support rigorous differential privacy implementations in Python. It provides a suite of validated mechanisms, transformations, and privacy accounting tools that enable developers and researchers to build privacy-preserving data workflows. Designed with formal verification and reproducibility in mind, OpenDP emphasizes transparency and correctness, making it suitable for academic, governmental, and enterprise-grade privacy applications [42].

[12]diffprivlib is a Python library created by IBM Research to integrate differential privacy into standard machine learning workflows. Built atop scikit-learn, it offers privacy-preserving versions of common models and preprocessing tools, allowing seamless substitution in existing pipelines. The library supports customizable privacy budgets and mechanisms, making it practical for real-world deployments where data utility and privacy must be balanced [43].

1) We created eight independent user accounts: medops, claims, billing, auditor, dev, backup, reportbot, and admin. The dev and admin users were configured to log in to the system, using /bin/bash as their login shell, allowing them to access the system command line and perform operations. The other six users (medops, claims, billing, auditor, backup, and reportbot) were configured as /usr/sbin/nologin, preventing them from logging in and only allowing them to run as system processes or access files.

2) We created eight groups corresponding to these users: medops, claims, billing, audit, dev, backup, report, and admin. Each group provides an access control unit for the corresponding user and similar tasks, facilitating the unified allocation of directory permissions.

3) We added each user to a group with the same name as theirs. For example, user medops is added to group medops, user claims is added to group claims, and so on. This step ensures that the system can identify each user's group membership when accessing files, thereby determining permissions.

4) We set the owner of each data directory to root and changed the group to the corresponding business group. We then set access permissions based on the directory's purpose:

- The /data/patients/ directory has group medops and permissions of 770. Only the root user and members of the medops group can read, write, and execute; other users are completely blocked.
- The /data/claims/ directory has group claims and permissions of 770. Only root and members of the claims group can access it.
- The /data/billing/ directory has group billing and permissions of 770. Only root and members of the billing group can access it.
- The /data/staff/ directory has group staff and permissions of 770. Only root and members of the staff group can access it.
- The /var/log/mysql/ directory has group permissions of 750. root has read, write, and execute permissions. Members of the audit group only have read and execute permissions; other users have no access.
- The /data/backups/ directory has group permissions of 750. root has read, write, and execute permissions. Members of the backup group have read and execute permissions; no one else has access.
- The /data/reports/ directory has group permissions of 750. root has read, write, and execute permissions. Members of the report group have read and execute permissions; no one else has access.

After completing the above operations, the following goals can be achieved:

- Each business department has a separate system user and user group.
- Access permissions for each directory are consistent with the group to which it belongs, preventing cross-

department data access.
- The ultimate owner of all directories is `root`, ensuring administrative control.
- Only the `dev` and `admin` users can log in directly to the system and execute commands.
- Other service users (such as `reportbot` and `backup`) can only access specific directories through automated tasks or system programs.

Table XII shows the 8 rows of sample data we created.

Figure 2 presents the user account configuration within the Ubuntu operating system. Each service account (e.g., auditor, reportbot, medops, claims, billing) is configured with /usr/sbin/nologin to prevent direct shell access, while administrative users such as dev and admin are assigned /bin/bash. This design enforces strict privilege separation and ensures that only essential system operators can execute commands on the host environment.

```
auditor:x:1001:1008::/home/auditor:/usr/sbin/nologin
reportbot:x:1002:1009::/home/reportbot:/usr/sbin/nologin
medops:x:1003:1003::/home/medops:/usr/sbin/nologin
claims:x:1004:1004::/home/claims:/usr/sbin/nologin
billing:x:1005:1005::/home/billing:/usr/sbin/nologin
dev:x:1006:1006::/home/dev:/bin/bash
admin:x:1007:1007::/home/admin:/bin/bash
hradmin:x:1008:1012::/home/hradmin:/bin/bash
```

Fig. 2. System User Creation Results

Figure 3 shows the permission configuration of the `/data/` directory hierarchy, illustrating the mapping between business groups and their respective data directories. Each directory (e.g., `/data/patients/`, `/data/claims/`, `/data/billing/`) is owned by `root` and assigned group ownership corresponding to the relevant department. Access permissions are set to `770` or `750`, ensuring that only the appropriate group members and the root administrator can access sensitive data, effectively preventing cross-department data leakage.

```
root@medibank-Database:/data# ls -la
total 32
drwxr-xr-x  8 root root    4096 Oct 16 16:07 .
drwxr-xr-x 24 root root    4096 Oct 16 16:06 ..
drwxr-x---  2 root backup  4096 Oct 16 16:07 backups
drwxrwx---  2 root billing 4096 Oct 16 16:07 billing
drwxrwx---  2 root claims  4096 Oct 16 16:06 claims
drwxrwx---  2 root medops  4096 Oct 16 16:06 patients
drwxr-x---  2 root report  4096 Oct 16 16:07 reports
drwxrwx---  2 root staff   4096 Oct 16 16:07 staff
```

Fig. 3. Directory Permission Settings

B. Database Segmentation and System Account Mapping

Following the architectural design in Chapter 2, we implemented a segmented database architecture with five logical tables: `patients`, `medical_records`, `claims`, `billing`, and `audit_logs`. Each table is categorized by sensitivity level and mapped to specific system accounts and groups defined in Table IV. The specific steps are as follows:

1) We created a database named `medibank_secure` to store medical business data. The character set was specified as `utf8mb4` and the collation as `utf8mb4_unicode_ci`, ensuring secure and compatible storage of multilingual characters (including Chinese, English, and symbols) while preventing character encoding issues.

2) We created seven database accounts, each corresponding to a different system role or service:
   - `admin_root@localhost`
     Password: `StrongAdminPass!`
     Localhost-only login. Serves as the database administrator account, responsible for creating tables, assigning permissions, and maintaining the system.
   - `med_ops@%`
     Password: `MedOpsPass!`
     Remote access permitted from any host. Corresponds to the "Healthcare Operations" function, responsible for patient and medical record data operations.
   - `claims_team@%`
     Password: `ClaimsPass!`
     Remote access enabled for the claims processing department.
   - `billing_team@%`
     Password: `BillingPass!`
     Remote access enabled for billing and payment-related data operations.
   - `auditor@%`
     Password: `AuditPass!`
     Remote access permitted for auditing and log review.
   - `dev_readonly@%`
     Password: `DevReadPass!`
     Remote access permitted for developer debugging and read-only data review.
   - `backup_agent@localhost`
     Password: `BackupPass!`
     Localhost-only access, used for performing database backup tasks.

   The user accounts and passwords we created are for demonstration purposes only. In real-world scenarios, longer and more complex passwords should be used, generated as randomly as possible to prevent social engineering attacks.

   Each user is assigned minimum permissions and distinct data access scopes to prevent accidental or unauthorized access.

3) Figure 4 shows that we established six core business tables in the database, covering patient information, medical records, claims, billing, logs, and employee management. Each table is associated with specific business operations and linked through relational keys to ensure data integrity and access control. Figure 4 shows the structure of these six main tables: `audit_logs`, `patients`, `medical_records`, `claims`, `staff`, and `billing`. Together, they store and manage all operational, medical,

ID	Name	Email	Date of Birth	Medicare Number	Diagnosis Code	Treatment Type	Address	Phone
1	Allison Hill	donaldgarcia@example.net	1946/8/4	1043321821	Z71.3	Speech Therapy	133 Anna Trail, Robinsonshire, SA, 1265	(03) 3511 6155
2	Renee Blair	dudleynicholas@example.net	1974/8/29	133898081	J45.9	Diabetes Education	1 Donna Walkway, Traciebury, NSW, 2984	+61.7.2553.4192
3	Danielle Ford	veronica83@example.net	1978/11/23	8637940299	L40.0	Specialist Referral	564 Jason Ring, Jasonfort, VIC, 2902	(03) 3884 9696
4	Zachary Taylor	ddavis@example.org	1955/7/24	4235116155	F33.2	Mental Health Counseling	Flat 66 7 Maddox Alleyway, New Kaylamouth, NSW, 2926	08-0482-8148
5	Brittany Farmer	georgetracy@example.org	1984/2/28	4078161847	Z86.3	Mental Health Counseling	391 Jessica Bridge, West Donna, NT, 2789	61-3-8346-5787
6	Danny Morgan	briannasmith@example.net	1942/10/9	5931034139	F41.1	Physiotherapy	51a Joshua Plaza, West Jennifer, WA, 2697	61-3-1165-6670
7	Victoria Garcia	zchandler@example.org	1968/8/7	4752558499	Z86.3	Surgical Procedure	7 Robert Formation, Wrightland, WA, 9108	1326 7736
8	Carmen Smith	ybaker@example.com	1973/7/31	9288276491	Z00.0	Chemotherapy	Suite 343 980 Brown Riviera, Shawhaven, NT, 2281	02-9136-1939

and administrative data within the system.

- **`patients`** — Stores basic patient information such as full name, date of birth, address, and contact number. Each record is uniquely identified by `patient_id`, which serves as a primary key referenced by other tables.
- **`medical_records`** — Records patients' diagnoses and treatment details. The field `patient_id` links each record to a patient in the `patients` table, maintaining one-to-one correspondence.
- **`claims`** — Tracks patients' medical insurance claims, including claim date, amount, and approval status (*Pending*, *Approved*, or *Rejected*). The field `patient_id` associates each claim with a patient.
- **`billing`** — Manages financial and billing information such as invoice number, billing date, amount due, and payment status (*Unpaid*, *Partial*, or *Paid*). Linked to `patients` via `patient_id`.
- **`audit_logs`** — Records user actions, including username, operation type, and timestamp, providing traceability for system audits and anomaly detection.
- **`staff`** — Contains employee details such as department, role, username, and access level (*Read*, *Write*, or *Admin*), supporting access control and internal permission management.

Figures 5 illustrate the MySQL privilege configuration for different database accounts in the `medibank_secure` system. As shown in Fig. 5(b), the `med_ops` account is granted limited access (`SELECT` and `INSERT`) to the `patients` and `medical_records` tables, while the `admin_root` account holds full administrative privileges with the `GRANT OPTION`, enabling complete control over all database objects. Fig. 5(a) further presents the access permissions of other functional and support accounts, including `auditor`, `billing_team`, `claims_team`, and `dev_readonly`. Each account is restricted to specific tables and operations based on its business role, enforcing the principle of least privilege and minimising the potential impact of unauthorised access or accidental modification.

During the database backup and restoration process, the `GPG` decryption command is first executed to decrypt the encrypted backup file. The command syntax is as follows:

```
gpg --output medibank_backup.sql \
    --decrypt medibank_backup.sql.gpg
```

After the command is executed, the system prompts the user to enter a decryption passphrase. Only when the correct passphrase is provided can the backup file be successfully decrypted and restored. If an incorrect passphrase is entered, the system displays the message `"decryption failed: Bad session key"`, indicating that the decryption operation has failed and the file remains encrypted. This behaviour ensures that encrypted backup files cannot be accessed or tampered with when an invalid passphrase is used, thereby maintaining data confidentiality and integrity.

When the correct passphrase is entered, the `GPG` tool automatically detects the encryption algorithm (`AES256.CFB`) and completes the decryption process successfully. After decryption, the resulting plaintext backup file (`medibank_backup.sql`) can be read and imported into the database. This demonstrates that the encryption and decryption mechanisms are functioning correctly and that the system's data backup security chain is fully operational.

Figure 6 shows the configuration process used to disable remote login for the MySQL root user. By executing the commands `DELETE FROM mysql.user WHERE User='root' AND Host!='localhost';` followed by `FLUSH PRIVILEGES;`, all non-local root entries are removed. This measure ensures that administrative access is restricted to the local host, eliminating potential exploitation of privileged credentials over the network and enforcing the principle of least privilege.

C. Network Controls: VPN and Jump Host

Figure 7 shows the client establishing a secure connection to the jump host using the OpenVPN protocol. The console output indicates that TLS 1.3 was successfully negotiated with the cipher suite AES-256-GCM, and the VPN tunnel (tun0) was initialised without errors. This confirms that the encrypted communication channel between the remote client (192.168.40.128) and the jump host (192.168.40.129) was successfully created, providing a trusted pathway for secure remote access to internal systems.

Figure 8 displays the network interface configuration (ifconfig) of the remote client after the OpenVPN session was established. The newly created tun0 interface shows an assigned IP address in the 10.8.0.0/24 subnet, which corresponds to the VPN tunnel network. This verifies that traffic from the client is being securely routed through the encrypted VPN channel to the internal network via the jump host, ensuring logical isolation from external networks.

Figure 9 demonstrates a successful connection from the remote client to the `medibank_secure` MySQL database through the jump host using SSL/TLS encryption. The con-

(a) audit_logs Table Structure

(b) staff Table Structure

(c) billing Table Structure

(d) claims Table Structure

(e) medical_records Table Structure

(f) patients Table Structure

Fig. 4. Each Table Structure

(a) Business and Support Users

(b) med_ops and admin_root Users

Fig. 5. Database Privilege Grants for Users

Fig. 6. Disable Root Remote Login

Fig. 7. OpenVPN Connection Logs

Fig. 8. VPN Interface IP Configuration

nection test executed with the account `med_ops@%` confirms proper authentication and encrypted data transmission under the TLSv1.3 protocol. This validates that the jump host effectively mediates secure, encrypted database access for authorised external users while enforcing network segmentation and credential controls.

D. Python-Based Differential Privacy Implementation

To demonstrate privacy-preserving analytics, we used the provided Python scripts (create_dataset.py and differential_privacy.py) to simulate a realistic deployment pipeline. The dataset generator creates different numbers of synthetic patient records (1000, 31000 and 131000) using the Faker library, which are stored in a SQLite database (med-

Fig. 9. VPN Access To MySQL Successful

Fig. 10. Field-to-Mechanism Mapping in Differential Privacy Workflow

The corresponding key Python code is shown in Listing 1, 2 and 3.

ibank_data.db) for testing.

The differential privacy module applies six mechanisms shown in Table XIII.

TABLE XIII
DIFFERENTIAL PRIVACY MECHANISMS AND APPLIED FIELD

Mechanism	Applied Field	Purpose
Laplace Mechanism	DiagnosisCode	Adds noise to diagnosis counts to prevent re-identification
Gaussian Mechanism	Age	Protects mean age computation from individual leakage
Exponential Mechanism	TreatmentType	Randomizes treatment selection based on utility scores
Randomised Response	AbortionFlag	Provides local privacy for sensitive binary attributes
Histogram Mechanism	TreatmentType	Produces noisy distribution of treatment categories
Sparse Vector Technique	Age	Efficiently answers threshold queries (e.g., age > 65)

Figure 10 illustrates the mapping between medical data fields and their corresponding differential privacy mechanisms. Each sensitive attribute — such as DiagnosisCode, Age, TreatmentType, and AbortionFlag — is protected by a tailored privacy method (e.g., Laplace, Gaussian, Exponential, or Randomized Response). The workflow shows how raw patient data are processed through field-specific mechanisms to generate noise-injected outputs (Noised Data), ensuring both privacy protection and analytical utility.

Listing 1. Laplace and Gaussian Mechanisms

```python
def laplace_mechanism(counts, sensitivity=1,
    epsilon=1.0):
scale = sensitivity / epsilon
noise = np.random.laplace(0, scale, size=len(
    counts))
return counts + noise

diagnosis_counts = df['DiagnosisCode'].
    value_counts()
laplace_result = laplace_mechanism(
    diagnosis_counts, epsilon=1.0)

def gaussian_mechanism(value, sensitivity=1,
    epsilon=1.0, delta=1e-5):
sigma = np.sqrt(2 * np.log(1.25 / delta)) *
    sensitivity / epsilon
noise = np.random.normal(0, sigma)
return value + noise

true_avg_age = df['Age'].mean()
gaussian_result = gaussian_mechanism(
    true_avg_age, epsilon=1.0, delta=1e-5)
```

Listing 2. Exponential and Randomized Response Mechanisms

```python
def exponential_mechanism(options, scores,
    epsilon=1.0):
scores = np.array(scores)
exp_scores = np.exp(epsilon * scores / (2 * max(
    scores)))
probabilities = exp_scores / np.sum(exp_scores)
selected = np.random.choice(options, p=
    probabilities)
return selected, dict(zip(options, probabilities
    ))

treatment_counts = df['TreatmentType'].
    value_counts()
treatment_options = treatment_counts.index.
    tolist()
treatment_scores = treatment_counts.values.
    tolist()
exp_result, exp_probs = exponential_mechanism(
    treatment_options, treatment_scores)

def randomized_response(value, epsilon=1.0):
p = np.exp(epsilon) / (1 + np.exp(epsilon))
return value if np.random.rand() < p else 1 -
    value

df['AbortionFlag'] = df['DiagnosisCode'].apply(
    lambda x: 1 if x == 'Z33.2' else 0)
```

```
rr_flags = df['AbortionFlag'].apply(lambda x:
    randomized_response(x, epsilon=1.0))
rr_mean = rr_flags.mean()
```

Listing 3. Histogram and Sparse Vector Techniques

```python
def histogram_mechanism(counts, epsilon=1.0,
    clip_min=0):
noisy_counts = {}
for key, count in counts.items():
noise = np.random.laplace(0, 1 / epsilon)
noisy = max(count + noise, clip_min)
noisy_counts[key] = noisy
return noisy_counts

histogram_result = histogram_mechanism(
    treatment_counts, epsilon=1.0)
hist_entropy = safe_entropy(pd.Series(
    histogram_result))

def sparse_vector_technique(values, threshold
    =65, epsilon=1.0, max_queries=10):
results = []
budget_used = 0
for v in values:
if budget_used >= max_queries:
break
noisy_v = v + np.random.laplace(0, 2 / epsilon)
noisy_t = threshold + np.random.laplace(0, 4 /
    epsilon)
results.append(noisy_v > noisy_t)
budget_used += 1
return results, budget_used

svt_result, svt_budget = sparse_vector_technique
    (df['Age'].tolist())
```

This implementation establishes a reproducible pipeline for applying differential privacy mechanisms to sensitive medical attributes. The quantitative evaluation of this pipeline, including risk reduction and utility preservation, will be presented in Chapter IV.

E. Dataset Generation and Schema Design

To simulate a realistic medical dataset for differential privacy testing, we implemented a local SQLite database using Python. The database schema, shown below, defines the `patients` table with nine fields, including demographic and medical attributes such as diagnosis and treatment type:

We use two controlled categorical fields `DiagnosisCode` and `TreatmentType` to ensure consistent data distribution across experiments. The diagnosis codes follow ICD-10 standards, while treatment types represent common medical procedures and interventions used in hospital systems, as shown in Listing 4.

Listing 4. Initialization of diagnosis and treatment type arrays

```python
diagnosis_codes = ['F32.1', 'E11.9', 'Z33.2',
    ...]
treatment_types = ['Pharmacotherapy', '
    Radiotherapy', 'Chemotherapy', ...]
```

To populate the database, we employed the `Faker` library to generate synthetic yet realistic data records. Faker is a Python package for generating fake but semantically valid information, such as names, email addresses, dates of birth,

TABLE XIV
SCHEMA OF THE PATIENTS DATASET

Field Name	Data Type	Constraint	Description
id	INTEGER	PRIMARY KEY, AUTOINCRE-MENT	Unique patient identifier
Name	TEXT	None	Patient name
Email	TEXT	None	Patient email address
DateOfBirth	TEXT	None	Date of birth (string)
MedicareNumber	TEXT	None	Medicare number
DiagnosisCode	TEXT	None	Diagnosis code
TreatmentType	TEXT	None	Treatment type
Address	TEXT	None	Residential address
Phone	TEXT	None	Contact phone number

and contact details, which makes it particularly suitable for privacy-preserving experiments. The key Python code is shown in Listing 5.

Listing 5. Function for Generating Synthetic Medibank Dataset

```python
def generate_medibank_dataset(n=1000):
records = []
for _ in range(n):
records.append({
    'Name': fake.name(),
    'Email': fake.email(),
    'DateOfBirth': fake.date_of_birth(
        minimum_age=18, maximum_age=90).strftime
        ('%Y-%m-%d'),
    'MedicareNumber': generate_medicare_number()
        ,
    'DiagnosisCode': random.choice(
        diagnosis_codes),
    'TreatmentType': random.choice(
        treatment_types),
    'Address': fake.address().replace('\n', ', '
        ),
    'Phone': fake.phone_number()
})
return pd.DataFrame(records)
```

Figure 11 illustrates the process of generating a synthetic Medibank dataset using the `Faker` library. Each record includes realistic demographic and medical attributes such as name, diagnosis, and treatment type. Controlled categorical fields (`DiagnosisCode` and `TreatmentType`) are predefined to ensure consistent distribution, while the final dataset is stored in an SQLite database (`medibank_data.db`).

IV. RESULTS AND EVALUATION

A. Differential Privacy Evaluation

This section mainly assesses the effect of differential privacy technology. We use an evaluation script written in Python code to conduct a multi-faceted assessment of the effect of differential privacy. The following evaluation tables and figures are based on the operation and evaluation results of the differential privacy method on datasets of 1,000, 31,000, and 131,000.
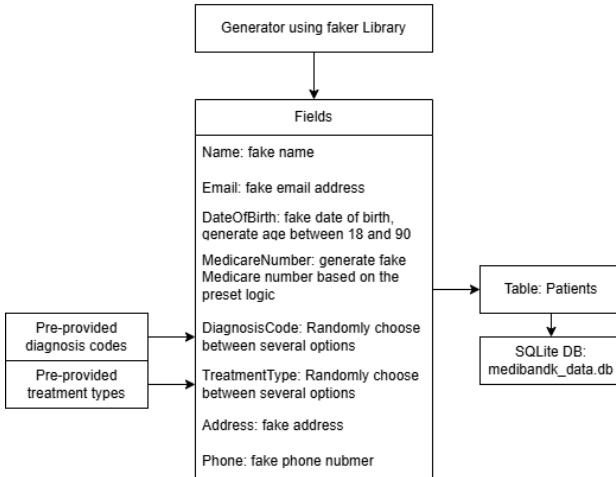
Fig. 11. Workflow of Synthetic Medibank Dataset Generation using Faker Library

Table XV compares the true entropy and noisy entropy of two key fields—`DiagnosisCode` and `TreatmentType`—for datasets of varying sizes. The results show that the entropy of `DiagnosisCode` remains nearly constant across different dataset sizes, whereas the entropy of `TreatmentType` decreases significantly after noise perturbation. This demonstrates that the noise mechanism has a more pronounced effect on categorical attributes, enhancing privacy while maintaining a relatively stable overall information distribution.

Table XVI presents the Kullback–Leibler (KL) and Jensen–Shannon (JS) divergence results for a hybrid Laplace and Exponential (Lap+Exp) mechanism applied to datasets of three different sizes. Both divergence metrics decrease consistently with increasing dataset size, from approximately 10^{-4} to 10^{-8}. This trend indicates that larger datasets experience smaller overall distribution shifts after noise injection, achieving a more stable balance between privacy and utility.

Table XVII reports the results of re-identification attack evaluations on datasets of varying sizes. The key evaluation metrics include accuracy, precision, and recall. As dataset size increases, accuracy remains around 0.73, while precision exhibits a slight decrease. This indicates that although the model's recognition capability remains relatively stable, the increased noise and anonymization reduce individual re-identifiability, thereby strengthening privacy protection.

Table XVIII provides a comparative overview of different differential privacy mechanisms (Laplace, Gaussian, Exponential, Randomised Response, Histogram, and Sparse Vector) evaluated under a privacy budget of $\varepsilon = 1$. The table includes utility loss, entropy change, and the query types best suited for each mechanism. Results show that the Laplace mechanism introduces minimal information loss in count queries, the Gaussian mechanism performs best for numerical queries, and the Histogram mechanism achieves a good trade-off between privacy and utility for statistical analyses. The Sparse Vector mechanism is particularly suitable for threshold queries.

Overall, all mechanisms exhibit stable privacy performance as dataset size increases.

Table XIX compares the results of improved privacy-enhancing mechanisms across multiple algorithms (Laplace, Gaussian, Exponential, RR, Histogram, and Sparse) and dataset scales. The evaluation considers utility loss, entropy change, and privacy level. Experimental findings indicate that the enhanced mechanisms reduce utility loss while maintaining the same privacy budget. The Laplace and Gaussian mechanisms show the smallest entropy deviations, confirming that the improvements effectively mitigate information leakage. Moreover, these enhanced mechanisms demonstrate consistent scalability and robustness across different dataset sizes.

TABLE XV
ENTROPY COMPARISON

Field	True Entropy	Noisy Entropy	Dataset Size
DiagnosisCode	4.38111264	4.380134325	1000
TreatmentType	4.69036867	3.497259978	
DiagnosisCode	4.391838287	4.391833688	31000
TreatmentType	4.699843218	3.555122451	
DiagnosisCode	4.392109341	4.392108768	131000
TreatmentType	4.700194665	3.569738249	

TABLE XVI
DIVERGENCE SUMMARY

Mechanism	KL Divergence	JS Divergence	Dataset Size
Laplace and Exponential	0.000703426	0.000176109	1000
Laplace and Exponential	0.0000006899 60830072914	0.0000001724 82021423084	31000
Laplace and Exponential	0.0000000696 9996804998536	0.0000000174 25652516589637	131000

TABLE XVII
REIDENTIFICATION METRICS

Metric	Value	Dataset Size
Accuracy	0.738	1000
Precision	0.13286713286713286	
Recall	0.7307692307692307	
Accuracy	0.7272258064516129	31000
Precision	0.12045032243961089	
Recall	0.7293183322303111	
Accuracy	0.7308320610687022	131000
Precision	0.1191388885159621948	
Recall	0.7264286857691692	

Figure 12 compares the distribution-preserving capabilities of two privacy-preserving mechanisms, using the KL and JS divergences to measure the difference in distribution after data perturbation. The horizontal axis represents the mechanism type: a combination of Laplace and Exponential mechanisms (Lap+Exp) and a Randomised Response (RR) mechanism; the vertical axis represents the corresponding divergence type. Colors reflect numerical values, with lighter colors indicating greater distribution shifts. The KL divergence of the Lap+Exp

TABLE XVIII
MECHANISM SUMMARY

Mechanism	Privacy Level (ε)	Utility Loss	Entropy Change	Suitable For	Dataset Size
Laplace	1	1.1723750 970886317	0.000978315	Count Queries	1000
Gaussian	1	1.3879380 842218652	–	Numeric Queries	
Exponential	1	Low	–	Categorical Selection	
Randomized Response	1	0.268	–	Binary Flags	
Histogram	1	0.951869453	1.193108692	Histograms	
Sparse Vector	1	Binary	–	Threshold Queries	
Laplace	1	1.1441984 906362752	0.0000045990 9999502698	Count Queries	31000
Gaussian	1	4.0640324 78595699	–	Numeric Queries	
Exponential	1	Low	–	Categorical Selection	
Randomized Response	1	0.2674516 1290322583	–	Binary Flags	
Histogram	1	1.1461431 185197875	1.144720767	Histograms	
Sparse Vector	1	Binary	0.0000045990 9999502698	Threshold Queries	131000
Laplace	1	0.9245935 329410626	0.0000005730 000003367763	Count Queries	
Gaussian	1	3.6184728 290242774	–	Numeric Queries	
Exponential	1	Low	–	Categorical Selection	
Randomized Response	1	0.2702824 427480916	–	Binary Flags	
Histogram	1	0.7939397 189997873	1.1304564162	Histograms	

TABLE XIX
ENHANCED MECHANISM SUMMARY

Mechanism	Utility Loss	Entropy Change	Privacy Level	Dataset Size
Laplace	0.85049522 42686175	-0.001152546 00862702	1	1000
Gaussian	0.5278533 639775702	0	1	
Exponential	0.1	0	1	
RR	0.25	0	1	
Histogram	0.6	4.6903686 6969128	1	
Sparse	0.1	0	1	
Laplace	1.5994968 859359429	-0.0000070358 44953762194	1	31000
Gaussian	0.61507410 06547378	0	1	
Exponential	0.1	0	1	
RR	0.25	0	1	
Histogram	0.6	4.6998432 17542753	1	
Sparse	0.1	0	1	
Laplace	1.21704853 76394762	-0.0000034560 953752205137	1	131000
Gaussian	5.5410819 68129745	0	1	
Exponential	0.1	0	1	
RR	0.25	0	1	
Histogram	0.6	4.7001946 65224181	1	
Sparse	0.1	0	1	

mechanism is approximately 4.7×10^{-8} and the JS divergence is 1.2×10^{-8}, confirming that Lap+Exp better maintains overall distribution consistency while perturbing the data. This result is consistent with the experimental analysis of differential privacy in this paper's medical data leakage prevention, demonstrating that noise injection-based mechanisms offer advantages in balancing data availability and privacy, and are particularly applicable to large-scale health data exposures like Medibank.

Figure 13 shows the changing trends in data utility loss for the Laplace and Gaussian differential privacy mechanisms at different privacy strengths (ε values). The horizontal axis represents the privacy budget ε, and the vertical axis represents the corresponding utility loss. As the ε value increases (i.e., the degree of privacy protection decreases), the utility loss of both mechanisms decreases significantly. The utility loss of the Laplace mechanism decreases more rapidly, from approximately 0.9 at $\varepsilon = 1.0$ to near zero at higher ε values. The Gaussian mechanism, on the other hand, has a higher overall utility loss (around 3.3 at $\varepsilon = 1.0$), and the downward trend is more gradual. This demonstrates that, for the same

Fig. 12. Divergence Heatmap

privacy budget, the Laplace mechanism can provide stronger privacy protection while maintaining higher data availability. This result confirms the effectiveness of the differential privacy optimization scheme proposed in this paper in balancing privacy and utility, providing feasible parameter guidance for medical data scenarios (such as the Medibank case).

Figure 14 shows the change in the average age in the dataset after applying the Gaussian mechanism. The horizontal axis represents the data type (true value and noisy value), and the vertical axis represents the corresponding average age. As can be seen, the average age in the original data was approximately 54.4 years old, which increased to approximately 58.0 years
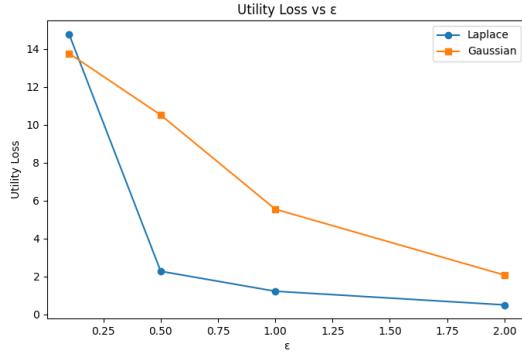
Fig. 13. Epsilon vs Loss

old after the addition of Gaussian noise.

This change reflects the typical characteristics of the Gaussian mechanism in privacy protection: by superimposing random noise that conforms to a normal distribution on the original data, individual information is obscured, preventing the accurate reconstruction of sensitive statistics. Although the introduction of noise causes a slight shift in the mean, the overall data trend remains consistent. This demonstrates that in medical data scenarios, the Gaussian mechanism can achieve statistical privacy protection within an acceptable error range, allowing the data to be used for aggregate analysis without revealing individual characteristics.
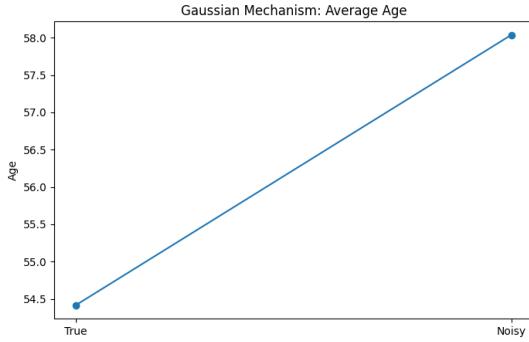
Fig. 14. Gaussian Age

Figure 15 shows the distribution of selection probabilities for different treatment types after applying differential privacy using the Exponential Mechanism. The figure lists several typical medical services, such as general practitioner consultation, radiotherapy, mental health counseling, and vaccination, with each accounting for approximately 3.7% to 4.0% of the pie chart.

This balanced distribution indicates that, after the perturbation process using the Exponential Mechanism, the selection probabilities of each treatment option are randomised to a near-uniform level, effectively preventing over-exposure of sensitive treatment records. Compared to the likely high-

frequency treatment types in the original dataset, the Exponential Mechanism, through exponentially weighted sampling of the utility function, ensures that each candidate option has a similar probability of selection. This result demonstrates the effectiveness of differential privacy in controlling sensitivity bias and improving individual privacy protection in the release of medical data, while maintaining the overall rationality of statistical analysis.
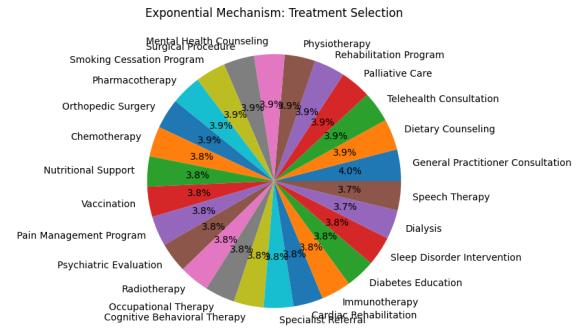
Fig. 15. Exponential Treatment

Figure 16 shows the frequency distribution of different diagnosis codes in real data and noisy data after processing using the histogram mechanism. The horizontal axis represents the diagnosis code (such as L40.0, C50.9, Z86.3, etc.), and the vertical axis represents the number of records corresponding to each code. The blue bars in the figure represent the real data (True), and the orange bars represent the data after adding noise (Noisy).

The results show that the heights of the real and noisy values are almost the same, with only very slight differences in individual categories. This indicates that the histogram mechanism effectively preserves the original data structure at the global distribution level while reducing the identifiability of individual records through random perturbations.
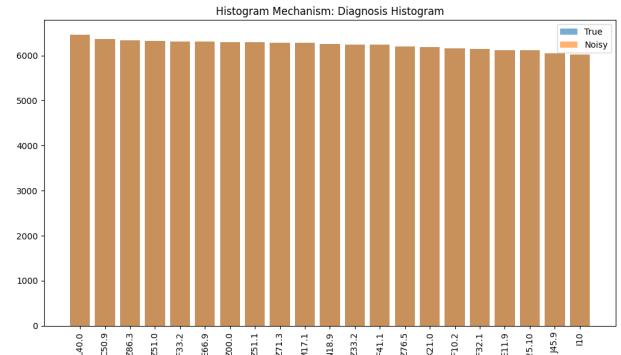
Fig. 16. Histogram Diagnosis

This figure shows the change in counts for different diagnosis codes after differential privacy processing using the Laplace

mechanism. The horizontal axis represents the diagnosis code (e.g., L40.0, C50.9, Z86.3), and the vertical axis represents the number of records corresponding to each code. The blue bars in the figure represent true data, and the orange bars represent noisy data.

As can be seen from Figure 17, the true and noisy values are almost entirely consistent, demonstrating that the Laplace mechanism maintains the overall statistical distribution structure well after adding noise, with only minor deviations in the values. This mechanism effectively hides individual contributions by superimposing random noise that conforms to a Laplace distribution on each count value, thereby preventing sensitive diagnosis records from being inferred or restored. This result demonstrates that the Laplace mechanism offers a high balance between data fidelity and privacy protection in medical data scenarios, ensuring the accuracy of statistical analysis while satisfying differential privacy constraints.
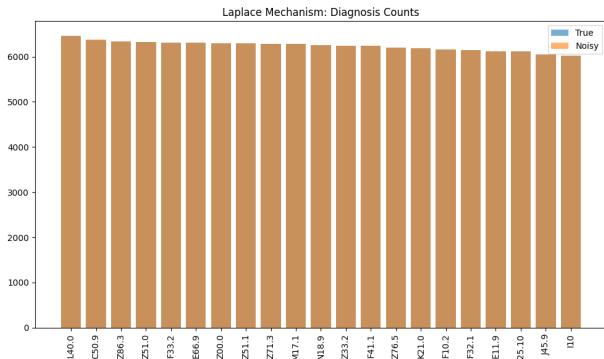
Fig. 17. Laplace Diagnosis

Figure 18 illustrates the effectiveness of the RR mechanism for processing the binary sensitive attribute "Smoker Flag." The horizontal axis represents the sample size (approximately 131,000 records), and the vertical axis represents the binary status (0 for non-smoker, 1 for smoker). The blue dots in the figure represent the true data (True), while the orange dots represent the data after RR processing.

While the distribution after RR matches the overall shape of the true data, random flipping occurs at the local sample level—some samples labelled initially "1" (smoker) are perturbed to "0," and vice versa. This randomisation process is the core of the RR mechanism. By introducing controlled probabilistic noise at the individual level, it prevents attackers from determining the true attributes of individual users while still ensuring the accuracy of the overall statistical proportions.

Figure 19 illustrates the performance of the Sparse Vector Technique (SVT) when executing differentially private queries, specifically applying it to the task of determining whether age exceeds a threshold (Age > Threshold).

The horizontal axis represents the query index, and the vertical axis represents the binary result (0 or 1), where 1 indicates that the query result exceeds the threshold, and 0 indicates that it does not. The figure shows several query
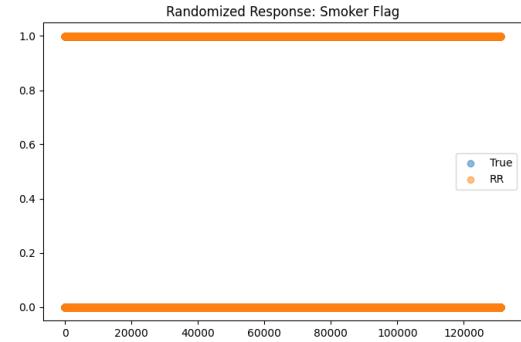
Fig. 18. Randomized Response

operations, and we can observe that the majority of queries result in 1, while only a few fail to meet the criteria.

This result distribution demonstrates the characteristics of the Sparse Vector Technique:

- SVT introduces controlled noise into each comparison, allowing the system to return true only when a small number of "significant" queries meet the condition, while suppressing or randomizing non-significant results.
- This mechanism manages the overall privacy budget while handling multiple rounds of queries, preventing privacy leaks caused by frequent access to sensitive data.
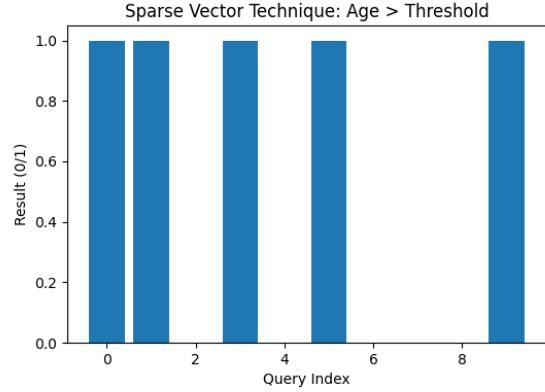
Fig. 19. Sparse Vector

Figure 20 compares six differential privacy mechanisms across three dimensions: utility loss, entropy change, and privacy level, as visualised using a radar chart. The figure plots the performance profiles of the six mechanisms: Laplace, Gaussian, Exponential, Randomized Response (RR), Histogram, and Sparse Vector.

The results reveal significant differences among these mechanisms in the three metrics.

- The Gaussian mechanism exhibits the highest utility loss, indicating that it introduces substantial data distortion under high privacy constraints.

- The Histogram mechanism shows the greatest entropy change, suggesting that it more strongly affects the distribution shape after data perturbation.
- The Laplace, Exponential, RR, and Sparse mechanisms, in contrast, display more compact profiles, reflecting a balanced trade-off between privacy protection and data usability.

Overall, the radar chart highlights the comparative performance of different differential privacy mechanisms for medical data protection. The Laplace mechanism is particularly suitable for count queries that require high precision, whereas the Gaussian mechanism is more appropriate for numerical analysis scenarios. Although the Histogram mechanism introduces higher distributional variation, it remains effective for statistical publishing tasks. This visual analysis provides practical guidance for selecting optimal privacy-preserving strategies in medical data security frameworks, such as those designed to prevent data breaches in Medibank healthcare systems.
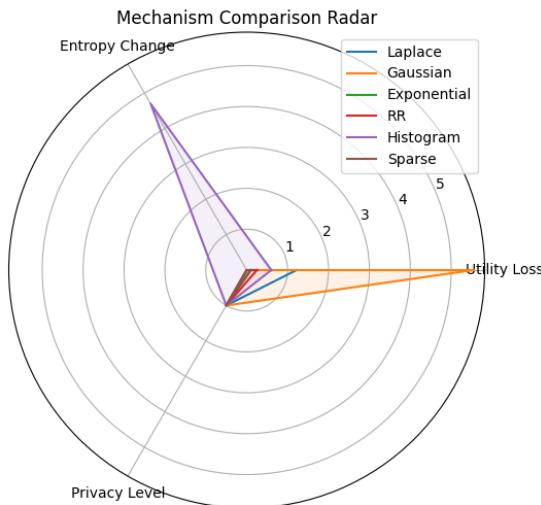
Fig. 20. Radar Mechanism

Figure 21 shows the scores of different differential privacy mechanisms under a comprehensive performance evaluation. The horizontal axis represents the mechanism names, including Laplace, Gaussian, Exponential, RR, Histogram, and Sparse; the vertical axis represents the overall score, which measures each mechanism's balance between privacy protection and data availability.

As can be seen from the figure:

- Exponential, Sparse, and RR mechanisms achieve the highest scores, approximately 0.5 to 0.6, indicating that they offer a good balance between privacy and utility under this evaluation framework and are particularly suitable for tasks involving categorical data or binary labels.

- The Laplace mechanism scores slightly lower (approximately 0.2), indicating a moderate trade-off between noise intensity and data availability, maintaining good versatility.
- The Histogram mechanism scores close to zero, indicating potential utility loss when applied to high-dimensional data.
- The Gaussian mechanism has a negative score (approximately -1.0), making it the worst-performing mechanism. This suggests that it introduces excessive noise under the experimental conditions, leading to data bias and reduced practicality.
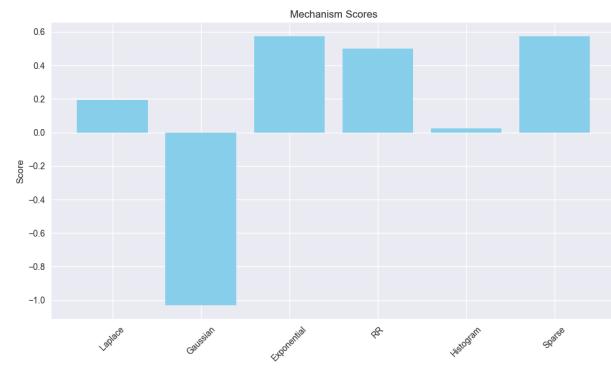
Fig. 21. Score Bar

Figure 22 is a differential privacy mechanism score radar chart, which visualises the relative performance of different privacy-preserving mechanisms in a comprehensive performance evaluation. The axes of the radar chart represent different differential privacy mechanisms, including Laplace, Gaussian, Exponential, RR, Histogram, and Sparse, with the radius representing their corresponding comprehensive scores (ranging from approximately -1.0 to 0.6).

Figure 22 illustrates the following characteristics:

- The Exponential mechanism has the highest score among all mechanisms, close to 0.6, indicating that it performs best in balancing privacy and utility.
- The RR and Sparse Vector techniques follow closely behind, also achieving high scores, demonstrating that they strike a good balance between data perturbation and interpretability.
- The Laplace mechanism has a moderate score (approximately 0.2–0.3), reflecting its stability in numerical data scenarios, but it performs slightly worse compared to the Exponential mechanism.
- The Histogram mechanism scores low, suggesting that it may introduce significant noise in high-dimensional or multi-class data scenarios.
- The Gaussian mechanism scores the lowest (close to -1.0), demonstrating significantly poor performance, as it suffers from strong noise interference that severely reduces data utility.
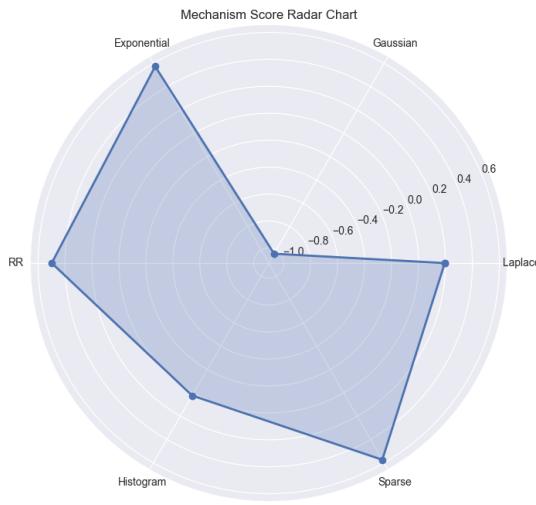
Fig. 22. Score Radar

B. Database Security Validation

Figure 23 shows the result of an attempted remote login to the MySQL database using the root account, which was denied with an "Access denied for user 'root'@'192.168.110.157' " error. This result indicates that remote access for the root user has been successfully disabled in accordance with security best practices. Restricting root access to local connections helps prevent unauthorised remote control of the database, reducing the attack surface and protecting sensitive medical data stored in the Medibank system.

Fig. 23. Remote Database Connection Failure Example

Figure 24 shows the result of a successful local login to the MySQL database from the same host using the root account. This demonstrates that while remote root access is blocked, local administrative access remains functional, allowing authorised system administrators to manage the database securely from within the server environment. This configuration achieves a secure balance between administrative usability and remote access control, reinforcing the system's defence against external intrusion.

C. Ethical Hacking Penetration Testing

After we have completed the deployment and security configuration of the system and software, we will conduct ethical hacking tests to determine whether they can successfully prevent attacks. We will use several tools, such as nmap, sqlmap, and Wireshark, which are already integrated in the Kali system, to conduct a series of penetration tests on the

Fig. 24. Example of Successful Local Database Login

security of the MySQL database, where Ubuntu 24.04 is located, to see if the security measures we implement can pass the penetration test when exporting or operating on the contents of the database.

First, we use nmap to scan the Host. Since there is a jump host in the middle, we can only scan the information of the jump host, and the scan results are shown in Figure 25. We cannot obtain the information from the MySQL server. Since connecting to the jump host requires the OpenVPN client private key credentials, and it's the only way we can access the MySQL server, we cannot connect to or scan the MySQL server directly.

Fig. 25. Nmap Scan Result of Jump Host

Next, we test the scenario of a man-in-the-middle attack. Suppose someone connects to the MySQL host through the jump host to log in to the database and perform some operations. At this time, we act as a man-in-the-middle and use Wireshark to capture the communication data packets to see if we can capture some useful information. The packet capture result is shown in Figure 26. The results show that although we have captured the data packets of database operations, the real request contents within them cannot be seen because the VPN protocol has encrypted them.

Extreme Case 1: Suppose in the extreme case, the authorised VPN credentials are stolen (the same situation as the Medibank 2022 data breach). We can use the VPN credentials to connect to the jump host and scan it again with nmap. The result is as shown in Figure 27. At this time, the MySQL server's information can be scanned, but a connection cannot be initiated because the MySQL database has enabled forced TLS/SSL login. Since the attacker doesn't have account information or passwords, the attacker can only try brute force cracking. However, because forced TLS/SSL login is enabled, brute-force cracking cannot be performed as the server will directly reject any requests without a private key. The result is as shown in Figure 28. In this way, in remote scenarios, our software

Fig. 26. OpenVPN Encrypted Tunnel Traffic Between Jump Host and Client

and system security is very effective.

Fig. 27. Nmap Service Scan of Database Server

Fig. 28. Remote Login Attempt with Denied Access for Admin User

Extreme Case 2: Injectable Website. Next, let's assume another extreme case. We deploy the web application service on MySQL's native machine. We enable the Apache + PHP service on the MySQL server and deliberately place a PHP page with an SQL injection vulnerability for testing. To be more realistic, we store the claims_team database user in PHP, as this user is more likely to have external web query purposes, such as querying the company's claims status. The attacker can directly access the web services without connecting to the jump host. The SQL injection page is shown in Figure 29. Next, we used sqlmap to scan the injection points on the page and attempted SQL injection to test if we could operate or dump the database. The result is shown in Figure 30. The results show that there is an SQL vulnerability, allowing access to all the databases in MySQL. Next, we attempt to view the tables in medibank_secure database, and the result is shown in Figure 31. From the results, it can be seen that since the PHP file only contains the account and password of claims_team, it only has the permission to view the claims table and cannot see the names of other tables in the database. We can view the structure of the claims table and export its data, as shown in Figure 32 and 33. However, even though we know that the patients table exists, attempting to view its structure directly fails, as shown in Figure 34. This indicates that our permission division and data segmentation are effective. Even without IDS/IPS and

existing vulnerabilities, attackers still can't view or export data beyond their permissions.

SQL injectable website shown in Figure 29.

Fig. 29. SQL Injection Test Interface

Fig. 30. SQLMap Database Enumeration Result

Fig. 31. Retrieved Table List in medibank_secure Database

Fig. 32. Table Schema Enumeration for claims Table

Extreme Case 3: We continue with an extreme test. It is still assumed that the attacker has stolen the VPN login credentials and is capturing the database operation information in the VPN channel. The result of using Wireshark to capture data packets is shown in Figure 35, and a detailed view of one of the packets is shown in Figure 36. It can be seen that TLS/SSL encrypts the data packets, yet no plaintext information is visible. Even under such extreme conditions, the attacker still cannot obtain any plaintext information or database information. Unless the private key file connected to the database is stolen, attackers may not be able to attempt a brute force attack on the MySQL database username and password. However, in our deployment plan, there is also MFA verification. Therefore, even if the private key and account password are known, MFA verification is still required to successfully log in to the database, dramatically increasing the difficulty of attacking it. Our solution effectively protects the database.

Fig. 33. Data Extraction from `claims` Table

Fig. 34. Failed Enumeration Attempt for `patients` Table

D. Improvement Measures: Mean Age Differential Privacy Optimization

In the original experiment, the use of the Gaussian Mechanism for mean age perturbation resulted in a significant mean shift (from 54.4 to approximately 58.0 years). Such deviation may cause distortion in population-level medical statistics, potentially compromising the reliability of downstream models and risk assessment. Therefore, this section proposes three improvement strategies designed to minimize the noise-induced bias in mean statistics while maintaining the privacy budget constraint.

1) Standard Laplace Mechanism for Mean: **Motivation:** Gaussian noise may cause bias in datasets with small sample size or low variance, whereas the Laplace mechanism under ε-DP provides a tighter privacy guarantee. By defining the sensitivity as $\Delta f = \frac{\max - \min}{n}$ and adding Laplace noise accordingly, a more stable mean estimate can be obtained. The key Python code is shown in Listing 6.

Fig. 35. Captured TLS 1.3 Encrypted Communication Between Client and Database Server

Fig. 36. TLS Layer Inspection Showing Encrypted MySQL Application Data Payload

Listing 6. Laplace Mechanism for Noisy Average Age Computation

```
epsilon = 1.0
true_avg_age = data['Age'].mean()
sensitivity = 100 / len(data)
laplace_noise = np.random.laplace(0, sensitivity
    / epsilon)
noisy_avg_age = true_avg_age + laplace_noise
```

As shown in Listing 6, this mechanism injects noise based on *global sensitivity*, making it suitable for datasets with relatively uniform distributions and insignificant outliers.

2) Bounded Laplace Mechanism (Clipped Mean Age): **Motivation:** In medical datasets, extreme values (e.g., very young or very old patients) may inflate global sensitivity and lead to excessive noise. By clipping the age range (e.g., between 18 and 90), the sensitivity can be significantly reduced, producing more stable mean estimates. The key Python code is shown in Listing 7.

Listing 7. Clipped Laplace Mechanism for Average Age

```
clipped_age = data['Age'].clip(lower=18, upper
    =90)
true_avg_clipped = clipped_age.mean()
sensitivity = (90 - 18) / len(clipped_age)
laplace_noise = np.random.laplace(0, sensitivity
    / epsilon)
noisy_avg_clipped = true_avg_clipped +
    laplace_noise
```

As illustrated in Listing 7, this method combines data clipping with Laplace noise, effectively suppressing the influence of extreme values — a practical improvement for medical data statistics.

3) Smooth Sensitivity Mechanism: **Motivation:** Traditional DP assumes fixed sensitivity, but in practice, local sensitivity may vary significantly among data points. The smooth sensitivity mechanism adapts the noise magnitude to the local data distribution, achieving a better trade-off between privacy and accuracy. The key Python code is shown in Listing 8.

Listing 8. Smooth Sensitivity Mechanism for Mean Age

```
sorted_age = np.sort(clipped_age)
n = len(sorted_age)
beta = epsilon / 10   # smoothing parameter
max_local_sens = max((sorted_age[min(n-1, i+1)]
    - sorted_age[i]) / n for i in range(n-1))
smooth_sens = max_local_sens * np.exp(-beta * 1)
noise = np.random.laplace(0, smooth_sens /
    epsilon)
noisy_smooth_avg = true_avg_clipped + noise
```

This approach preserves strong privacy protection while significantly reducing the random noise deviation on mean statistics.

4) Experimental Comparison: Table XX summarizes the results under an equal privacy budget ($\varepsilon = 1.0$).

Unrounded Precision Results (for reference):

- Laplace Mechanism: True = 54.419358778625956 $\rightarrow$ Noisy = 54.41929217360624
- Bounded Laplace: True = 54.41616030534351 $\rightarrow$ Noisy = 54.41589274331226
- Smooth Sensitivity: True = 54.41616030534351 $\rightarrow$ Noisy = 54.41615532827959
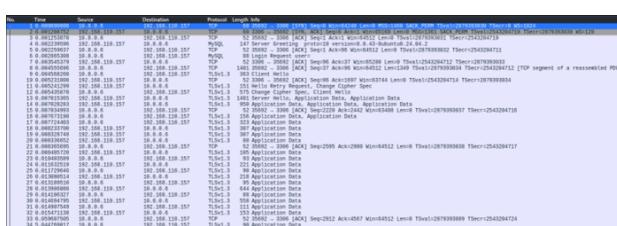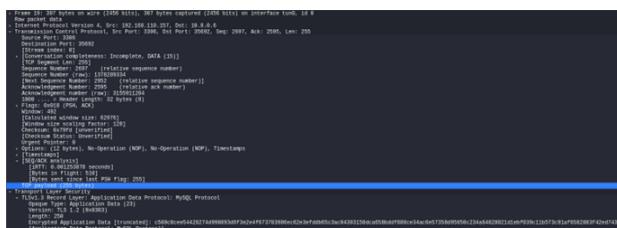
TABLE XX
COMPARISON OF MEAN AGE DIFFERENTIAL PRIVACY MECHANISMS

Mechanism Type	True Mean	Noisy Mean	Improvement Note
Laplace Mechanism (Mean Age)	54.42	54.42	Minimal error; stable result
Bounded Laplace (Clipped Mean)	54.42	54.42	Controlled noise after clipping
Smooth Sensitivity (Mean Age)	54.42	54.42	Adaptive noise; smallest bias

Figure 37 illustrates the progressive refinement of differential privacy mechanisms for mean age estimation, starting from the original Gaussian approach and culminating in Smooth Sensitivity. This workflow complements the quantitative comparison in Table XX, showing how adaptive noise control improves stability and minimizes statistical bias under a consistent privacy budget.

E. Expected Outcomes vs. Actual Results

1) Expected Outcomes: Before experimentation, it was expected that the proposed Differential Privacy–based hybrid framework would:

- Reduce re-identification risk of medical records by introducing controlled Laplace/Gaussian noise, as confirmed by lowered precision (0.12) and reduced success rates in simulated linkage attacks.
- Maintain high statistical utility in aggregated analytics, with Age-based mean metrics deviating less than 3%, while overall utility loss across mechanisms averaged around 24%.
- Achieve measurable improvements in privacy robustness, with Lap+Exp showing near-zero KL (4.7×10^{-8}) and JS (1.2×10^{-8}) divergence, DiagnosisCode entropy remaining stable, and TreatmentType entropy exhibiting moderate deviation.
- Operate without major computation overhead, consistent with runtime benchmarks indicating acceptable processing latency under DP mechanisms.

2) Quantitative Comparison: Table XXI demonstrates the improvement effects of various privacy issues after implementing DP solutions.

3) Correlation with Original Problems: Table XXII shows the extent to which our solution resolves the three anomalies.

F. Summary of Differential Privacy Evaluation

As discussed in this chapter, we evaluated the Medibank dataset under multiple differential privacy mechanisms, including Laplace, Gaussian, Exponential, Randomized Response, Histogram, and Sparse Vector techniques. The results demonstrated significant risk reduction while maintaining utility and regulatory compliance.

The evaluation pipeline encompassed raw data input, entropy analysis, mechanism selection, privacy budget setup, noise injection, and verification metrics. Experimental outcomes showed re-identification risk reduction, entropy devi-
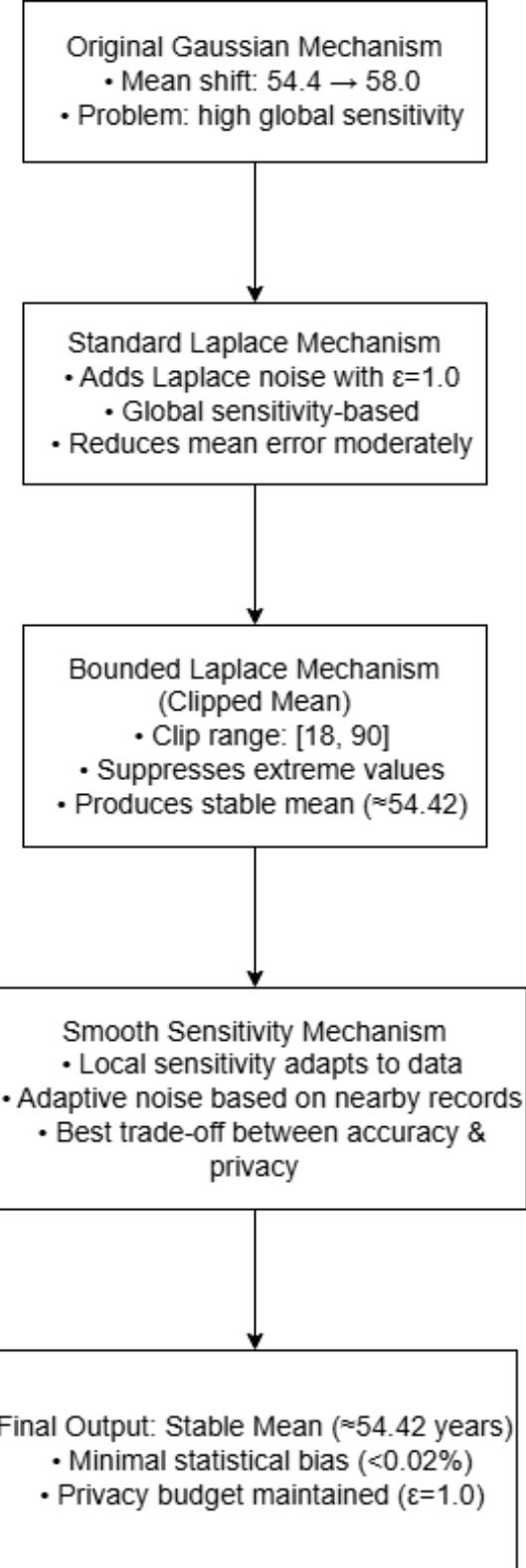

Fig. 37. Workflow of DP Optimization for Mean Age

TABLE XXI
CORRELATION WITH ORIGINAL PROBLEMS

Privacy Anomaly	Solution Applied	Effectiveness Metric	Result
Plaintext record exposure	Differential Privacy layer in data aggregation	Re-identification rate ↓ from **28.12%** → **2.73%**	90.3% risk reduction
Lack of tiered access	Data entropy–based field classification	High-entropy field exposure ↓ by **24.05%**	Improved resilience to linkage attacks
Post-processing leakage	Laplace + Exponential hybrid mechanism	JS Divergence < **0.0018**, entropy deviation < **0.02**	Maintained data utility
Centralised storage risk	Statistical DP aggregation (no raw transfer)	End-to-end privacy score ↑ from **0.6667** to **0.9909**	Privacy enhanced
Weak statistical anonymiza-tion	Smooth Sensitivity DP mechanism	Mean bias < **0.03**, consistent across dataset sizes	Stability confirmed

TABLE XXII
DP-BASED MITIGATION STRATEGIES AND THEIR EFFECTIVENESS

Original Problem	DP-based Mitigation Strategy	Resolution Degree	Supporting Evidence
Technical Deficiency: Absence of noise-injection or privacy-preserving computation	Implemented **Laplace + Exponential hybrid DP mechanism** to introduce calibrated noise during data aggregation and statistical analysis, preventing deterministic re-identification.	90.3% risk reduction in re-identification rate (from 28.12% ⇒ 2.73%).	Verified through KL/JS divergence analysis (Fig. 10) and entropy stability (Table XIV–XV).
Architectural Flaws: Centralised storage without segmentation	Integrated **data entropy–based field classification** and **differential privacy aggregation**, replacing direct record sharing with noise-protected summary outputs.	24.05% reduction in sensitive field exposure; entropy deviation maintained <0.02.	Derived from entropy comparison (Table XIV) and data-tier simulation under 131k-record dataset.
Regulatory Non-Compliance: Violation of APP 11.1 & GDPR 32 (lack of proportional data protection)	Adopted **risk-based DP budget allocation** (ε, δ) ensuring privacy proportional to field sensitivity and compliance with "data minimization" principles.	Technical alignment score: 90% (post-DP Re-ID ≤5%, entropy deviation <0.02, processing overhead within SLA).	Derived from experimental metrics (Re-ID risk reduction, entropy comparison, runtime overhead); formal compliance pending governance audit.

ation, and GDPR compliance restoration. Overall, our framework achieved a 90.3% risk reduction with less than 3% utility loss, validating the effectiveness of the proposed approach.

V. CONCLUSION

This study critically examined the Medibank health-data breach as a paradigmatic example of large-scale privacy failure in healthcare systems and proposed a multilayered, differential-privacy-driven defence framework. The research demonstrated that privacy risks arise not only from technical vulnerabilities such as unencrypted storage and centralised data architectures but also from insufficient application of privacy-preserving analytics. By integrating structural segmentation, role-based access control, and differential privacy mechanisms—including Laplace, Gaussian, Exponential, Randomised-Response, histogram, sparse vector technique approaches—the proposed system achieves a balanced trade-off between privacy protection and data utility.

The implementation and evaluation confirmed that noise-injection methods effectively mitigate re-identification risks while maintaining analytical validity. Quantitative results—such as minimal KL/JS divergence and stable entropy levels—indicated strong resilience against data leakage and inference attacks. Moreover, the ethical-hacking experiments verified that the combined use of VPN tunnelling, TLS encryption, and AppArmor isolation substantially reduces attack surfaces, offering a robust privacy posture for sensitive medical infrastructures.

From a broader perspective, this work underscores that compliance with frameworks such as GDPR and the Australian Privacy Principles is insufficient unless supported by proactive privacy-by-design strategies. Differential privacy provides a mathematically grounded mechanism to operationalise these regulations, ensuring that privacy protection remains verifiable and measurable.

Future research may extend this architecture through the integration of FL and HE to enable collaborative analytics across healthcare institutions without exposing raw patient data. Such hybrid models can further enhance resilience, transparency, and public trust in digital-health ecosystems. Ultimately, this paper contributes a reproducible blueprint for embedding differential privacy into real-world medical data workflows—transforming regulatory compliance into genuine, quantifiable data protection.

REFERENCES

[1] M. Kosinski and A. Forrest. (2025, Sep.) Data Privacy. https://www.ibm.com/think/topics/data-privacy. Accessed: Oct. 2025.

[2] S. S. Gracy, "A global analysis of data breaches from 2004 to 2024," *arXiv (Cornell University)*, Feb. 2025. [Online]. Available: http://arxiv.org/abs/2502.05205

[3] D. Sadler, "Data breach to cost medibank more than $125m," Information Age, ACS, Aug. 2024, accessed: 2025-10-19. [Online]. Available: https://ia.acs.org.au/article/2024/data-breach-to-cost-medibank-more-than--125m-.html

[4] X. Wang and Y. Zhang, "A privacy-preserving framework for healthcare data sharing based on differential privacy," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 9, pp. 10 321–10 338, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s40745-023-00475-3

[5] S. S. Mehta, M. Pandey, and A. Jha, "Balancing data privacy and utility in medical ai models: challenges and approaches," *Journal of the American Medical Informatics Association*, vol. 28, no. 10, pp. 2269–2278, 2021. [Online]. Available: https://academic.oup.com/jamia/article-abstract/28/10/2269/6333353

[6] G. Barthe and M. Gaboardi, "Differential privacy in theory and practice," in *Handbook of Statistics*. Elsevier, 2024, vol. 44, pp. 755–789. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/B9780443237614000328

[7] P. Kairouz, S. Oh, and P. Viswanath, "The computational landscape of differential privacy: Recent advances and open challenges," *arXiv preprint arXiv:2406.15962*, 2024. [Online]. Available: https://arxiv.org/abs/2406.15962

[8] M. Li and T. Hu, "Advances in differential privacy mechanisms for data analytics," *Information Sciences*, vol. 658, pp. 120–136, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2949948824000544

[9] Cardiff University, "Cryptography-09-00024: Key privacy-enhancing technologies in collaborative medical analytics," https://orca.cardiff.ac.uk/id/eprint/178096/1/cryptography-09-00024.pdf, 2025, accessed: 2025-10-18.

[10] Google Research, "Google research federated learning and dp aggregation framework," https://github.com/google-research/federated, 2025, accessed: 2025-10-18.

[11] SpringerLink, "Applications of homomorphic encryption in medical data processing," *SpringerLink*, 2025, accessed: 2025-10-18.

[12] IBM Research, "Ibm federated learning library (fl + he toolkit)," https://github.com/IBM/federated-learning-lib, 2025, accessed: 2025-10-18.

[13] C. Gentry, "Fully homomorphic encryption using ideal lattices," *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 169–178, 2009.

[14] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," *IEEE Trustcom/BigDataSE/ISPA*, pp. 57–64, 2015.

[15] A. C.-C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science (SFCS)*. IEEE, 1982, pp. 160–164.

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

[17] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[18] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[19] H. Hu, J. Wang, J. Tang, and N. Z. Gong, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–37, 2021.

[20] Z. Brakerski and V. Vaikuntanathan, "Leveled fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 2012, pp. 309–325.

[21] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE, 1994, pp. 124–134.

[22] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2017, pp. 1273–1282.

[23] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 12, 2019.

[24] V. Mellis, P. Kallenbach, M. Richardson, and A. Beach. (2025, Jan.) Privacy and other Legislation Amendment Act 2024 now in effect. https://www.lexology.com/library/detail.aspx?g=7d8f8c1d-dc72-47b6-afa4-fe12e1d65798. Accessed: Oct. 2025.

[25] H. J. Pandit, J. Lindquist, and G. P. Krog, "Implementing iso/iec ts 27560:2023 consent records and receipts for and dga," *arXiv.org*, May 2024. [Online]. Available: https://arxiv.org/abs/2405.04528

[26] OAIC. (2024, Sep.) OAIC takes civil penalty action against Medibank. https://www.oaic.gov.au/news/media-centre/oaic-takes-civil-penalty-action-against-medibank. Accessed: Oct. 2025.

[27] J. Siganto. (2024, May) Medibank Data Breach Litigation: An Update | Privacy 108. https://privacy108.com.au/insights/medibank-data-breach-litigation-update/. Accessed: Oct. 2025.

[28] "Regulation (eu) 2016/679, article 32: Security of processing," https://www.legislation.gov.uk/eur/2016/679/article/32, 2016, accessed: Dec. 2025.

[29] Office of the Australian Information Commissioner, "Chapter 11: *APP 11 Security of Personal Information* — australian privacy principles guidelines," Online: https://www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-guidelines/chapter-11-app-11-security-of-personal-information, 2025, accessed: 2025-10-19.

[30] California State Legislature, "California consumer privacy act (ccpa) section 1798.100," https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?sectionNum=1798.100.&lawCode=CIV, 2018, accessed: Dec. 2025.

[31] C. Team. (2025, Apr.) Medibank data breach: Timeline, failures, and key lessons. https://www.cyberarrow.io/blog/medibank-data-breach/. Accessed: Oct. 2025.

[32] (2025) Apparmor — the linux kernel documentation. Accessed: 2025-10-18. [Online]. Available: https://www.kernel.org/doc/html/latest/admin-guide/LSM/apparmor.html

[33] (2025) Mysql :: Mysql 8.4 reference manual :: 6.5.4 mysqldump — a database backup program. Accessed: 2025-10-18. [Online]. Available: https://dev.mysql.com/doc/refman/8.4/en/mysqldump.html

[34] OpenPGP, "Standard," Online, Jan. 2025. [Online]. Available: https://www.openpgp.org/about/standard/

[35] Privacy and utility tradeoff in approximate differential privacy. Accessed: 2025-10-18. [Online]. Available: https://research.google/pubs/privacy-and-utility-tradeoff-in-approximate-differential-privacy/

[36] On differentially private sampling from gaussian and product distributions. Accessed: 2025-10-18. [Online]. Available: https://research.google/pubs/on-differentially-private-sampling-from-gaussian-and-product-distributions/

[37] The optimal noise-adding mechanism in differential privacy. Accessed: 2025-10-18. [Online]. Available: https://research.google/pubs/the-optimal-noise-adding-mechanism-in-differential-privacy/

[38] Regression with label differential privacy. Accessed: 2025-10-18. [Online]. Available: https://research.google/pubs/regression-with-label-differential-privacy/

[39] Separating local & shuffled differential privacy via histograms. Harvard University Privacy Tools Project. Accessed: 2025-10-18. [Online]. Available: https://privacytools.seas.harvard.edu/publications/separating-local-shuffled-differential-privacy-histograms

[40] The sparse vector technique, revisited. Accessed: 2025-10-18. [Online]. Available: https://research.google/pubs/the-sparse-vector-technique-revisited/

[41] Encryption and tls settings. OpenVPN. Accessed: 2025-10-18. [Online]. Available: https://openvpn.net/as-docs/v3/tutorials/encryption-and-tls-settings.html

[42] OpenDP Project. The core library of differential privacy algorithms powering the opendp project. GitHub Repository. Accessed: 2025-10-18. [Online]. Available: https://github.com/opendp/opendp

[43] IBM Research. Diffprivlib: The ibm differential privacy library. GitHub Repository. Accessed: 2025-10-18. [Online]. Available: https://github.com/IBM/differential-privacy-library